

数字逻辑

——第四章 组合逻辑电路



4.1 组合逻辑电路分析

- 从电路结构看，组合逻辑电路具有两个特点：
 - 1) 由逻辑门电路组成，不包含任何记忆元件
 - 2) 信号是单向传输的，不存在任何反馈回路



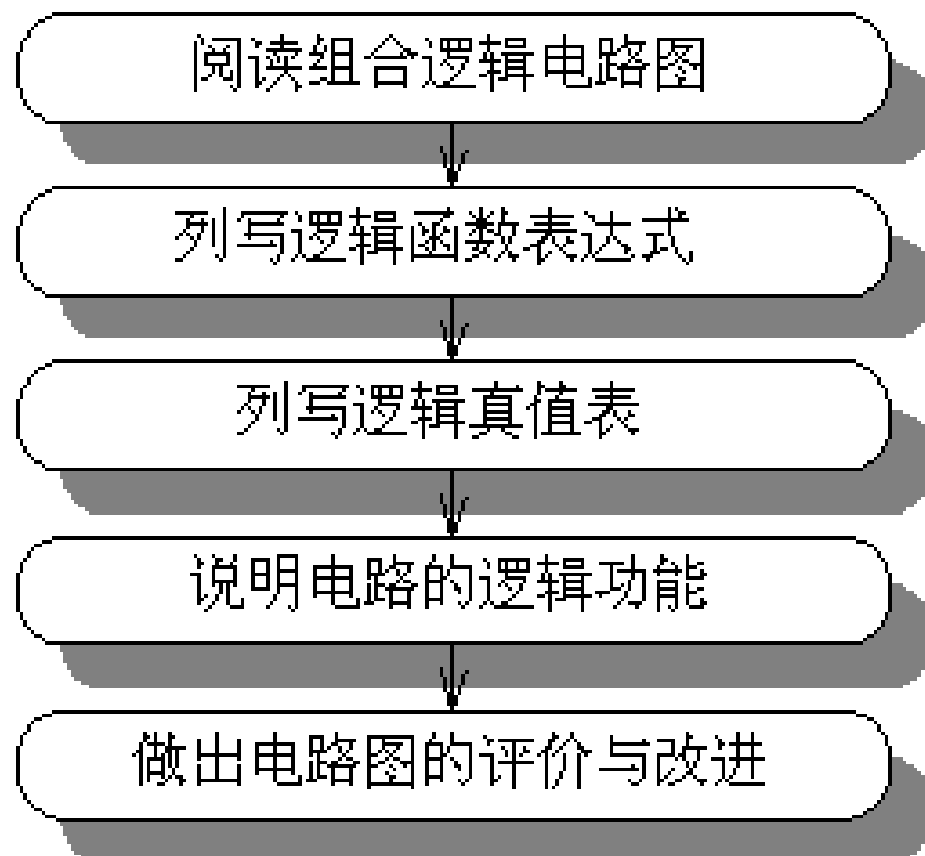
4.1 组合逻辑电路分析

- 什么是组合逻辑分析?
- [答]
- 找出组合逻辑电路的输入与输出关系，确定在什么样的输入取值组合下，对应的逻辑输出值有效，从而阐明组合逻辑电路的功能。



4.1 组合逻辑电路分析

■ 组合逻辑分析的步骤

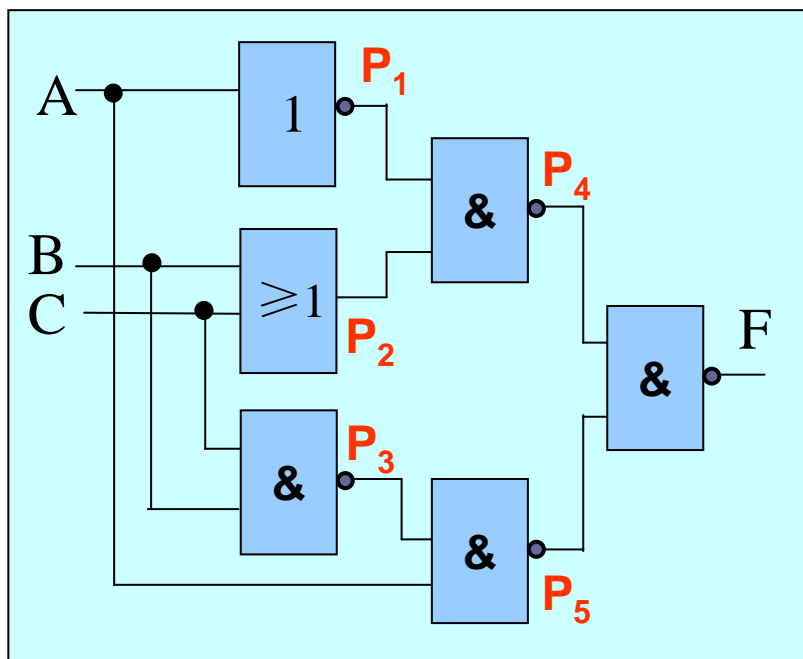




4.1 组合逻辑电路分析

■ 组合逻辑电路分析举例

例1：教材P98，例4.1



输入			输出
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

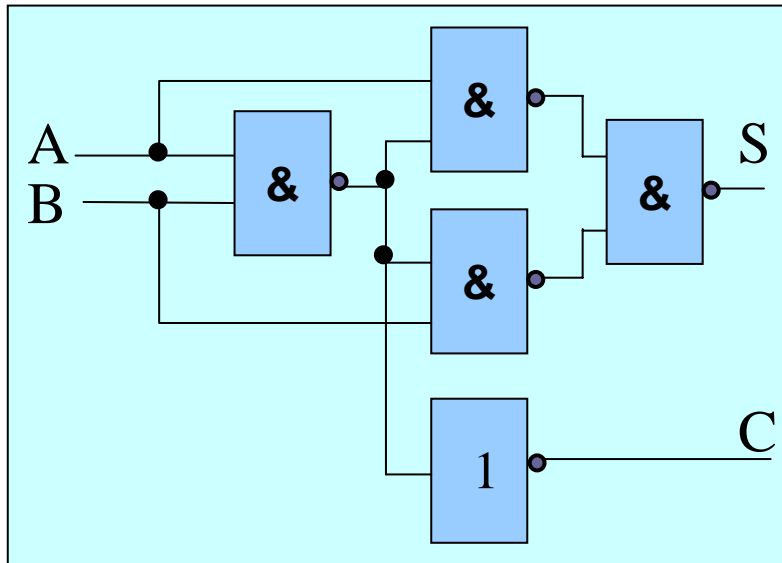
功能：不一致电路



4.1 组合逻辑电路分析

■ 组合逻辑电路分析举例

例2：教材P99，例4.2





4.1 组合逻辑电路分析

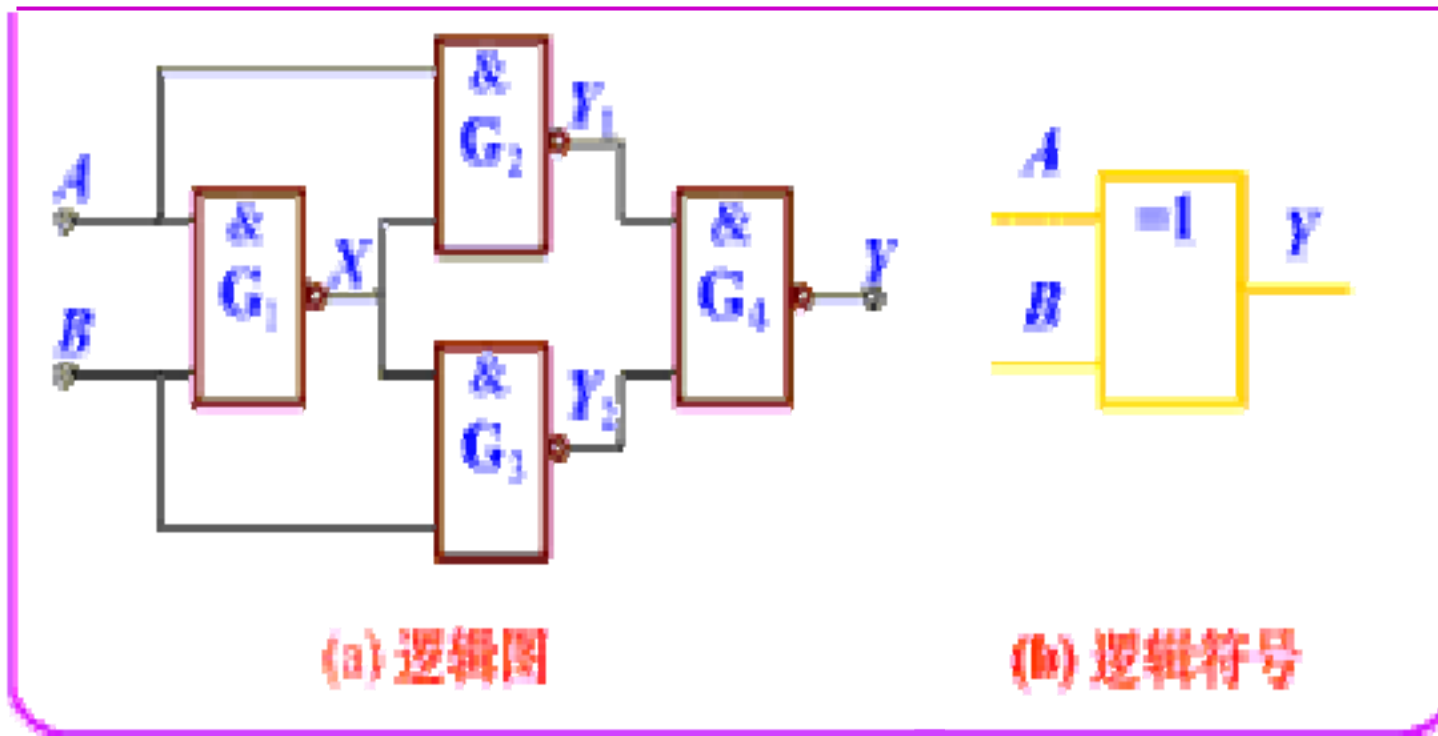
■ 组合逻辑电路分析举例

例3：CAI演示



4.1 组合逻辑电路分析

- 组合逻辑电路分析举例
- 例4:





4.1 组合逻辑电路分析

■ 组合逻辑电路分析举例

■ 例4:

- [解] (1) 由逻辑图写出逻辑式
逐级写出各个逻辑门的逻辑函数表达式并化简:

G1门 $X = \overline{AB}$

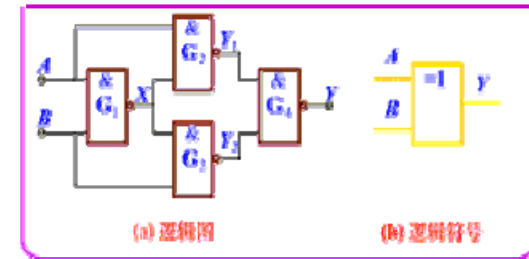
G2门 $Y_1 = \overline{AX} = \overline{AAB}$

G3门 $Y_2 = \overline{BX} = \overline{BAB}$

G4门 $Y = \overline{Y_1 Y_2} = \overline{\overline{AAB} \cdot \overline{BAB}} = \overline{\overline{AAB}} + \overline{\overline{BAB}}$

$$= A\overline{A}B + B\overline{A}B = A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B})$$

$$= A\overline{A} + A\overline{B} + B\overline{A} + B\overline{B} = A\overline{B} + B\overline{A}$$



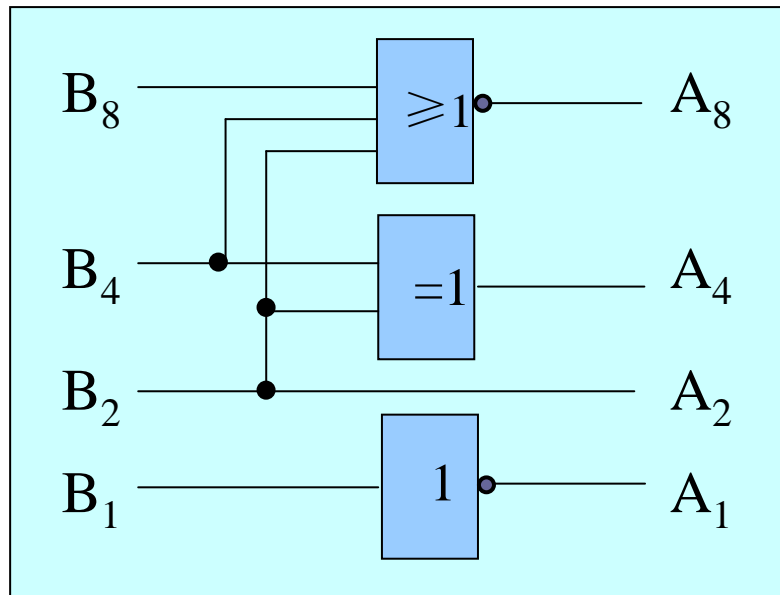
状态表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



组合逻辑电路分析举例

例5：分析如下逻辑电路。



列写表达式：

$$A_8 = \overline{B_8 + B_4 + B_2} = \overline{B_8} \overline{B_4} \overline{B_2}$$

$$A_4 = B_4 \oplus B_2$$

$$A_2 = B_2$$

$$A_1 = \overline{B_1}$$

这是一个8421BCD码对9求补器

B_8	B_4	B_2	B_1	A_8	A_4	A_2	A_1
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	1	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	0	1
1	1	0	1	0	1	0	0
1	1	1	0	0	0	1	1
1	1	1	1	0	0	1	0



4.2 组合逻辑电路设计

■ 组合逻辑电路设计方法

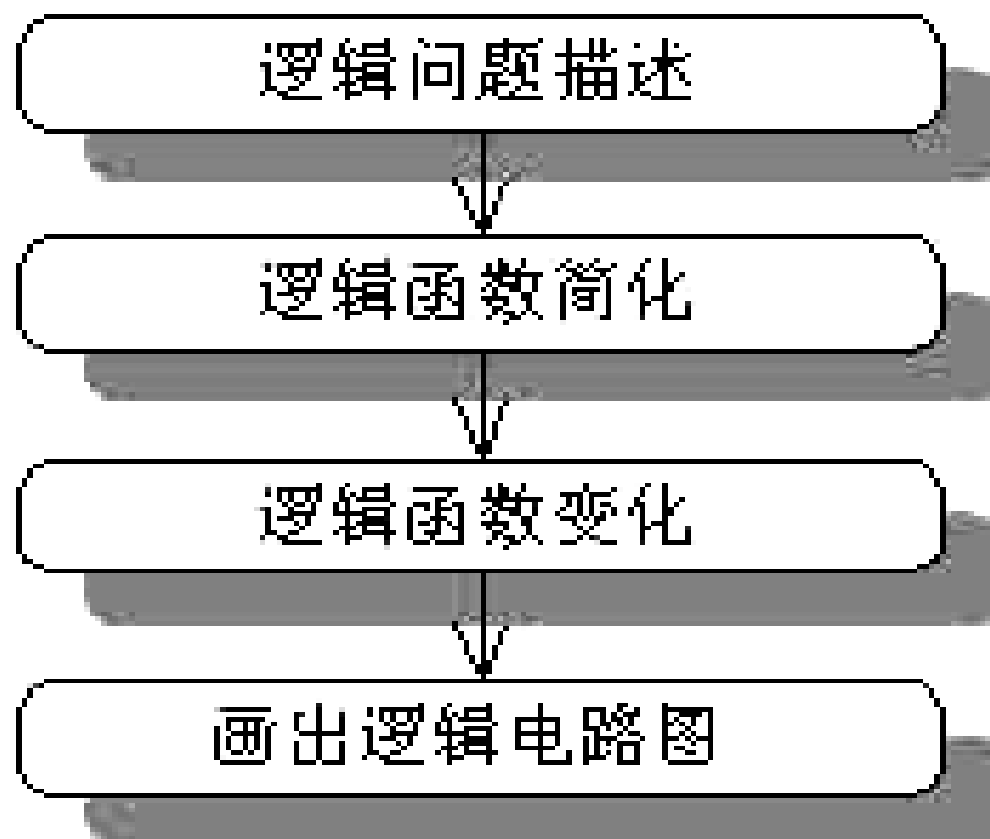
根据问题要求完成的逻辑功能，求出在特定条件下实现该功能的逻辑电路，这一过程称为逻辑设计，又叫逻辑综合。

■ 逻辑设计是逻辑分析的逆过程



4.1 组合逻辑电路分析

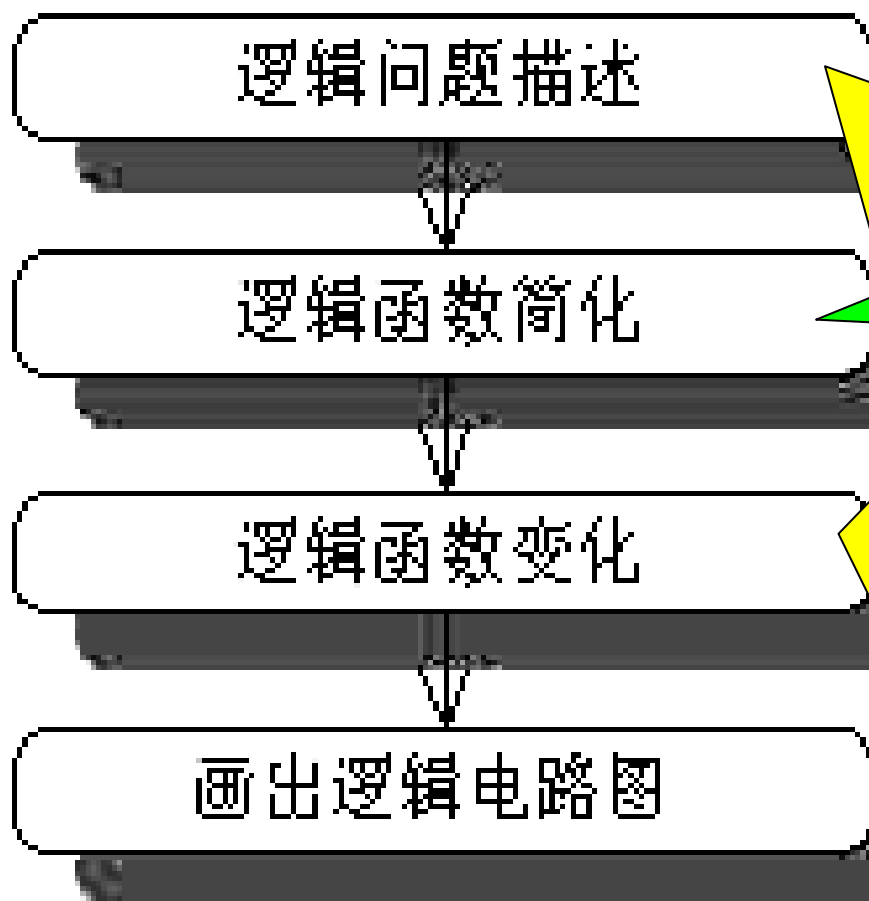
■ 组合逻辑电路设计方法





3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤



所谓逻辑函数的变换

所谓逻辑函数的变换，就是根据小规模标准门电路器件的类型，将最简与-或表达式变换为与非门（74LS00，74LS20）、或非门（74LS02）、与或非门（74LS51）来实现的形式。

以与或非门为例，其表达式。



3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤

逻辑问题描述

逻辑函数

所谓逻辑问题的描述，就是将文字描述的设计要求抽象为一个逻辑表达式。

通常的方法是：先界定输入输出逻辑变量的含义，建立真值表，再由真值表写出逻辑表达式。有些情况下，可由设计要求直接建立逻辑表达式。

例1:

某汽车驾驶员培训班进行结业考试，有三名评判员，其中A为主评判员，B、C为副评判员。在评判时按照少数服从多数的原则，但若主评判员认为合格，亦可通过。

试用与非门构成的逻辑电路实现此评判规则。



3 组合逻辑电路设计

例1：某汽车驾驶员培训班进行结业考试，有三名评判员，其中A为主评判员，B、C为副评判员。在评判时按照少数服从多数的原则，但若主评判员认为合格，亦可通过。试用与非门构成的逻辑电路实现此评判规则。

3.2 组合逻辑电路设计步骤

1) 逻辑问题的描述：

- 设：
- A为1时表示A同意合格，反之为不同意。
- B为1时表示B同意合格，反之为不同意。
- C为1时表示C同意合格，反之为不同意。
- Y为1时表示合格，反之为不合格。

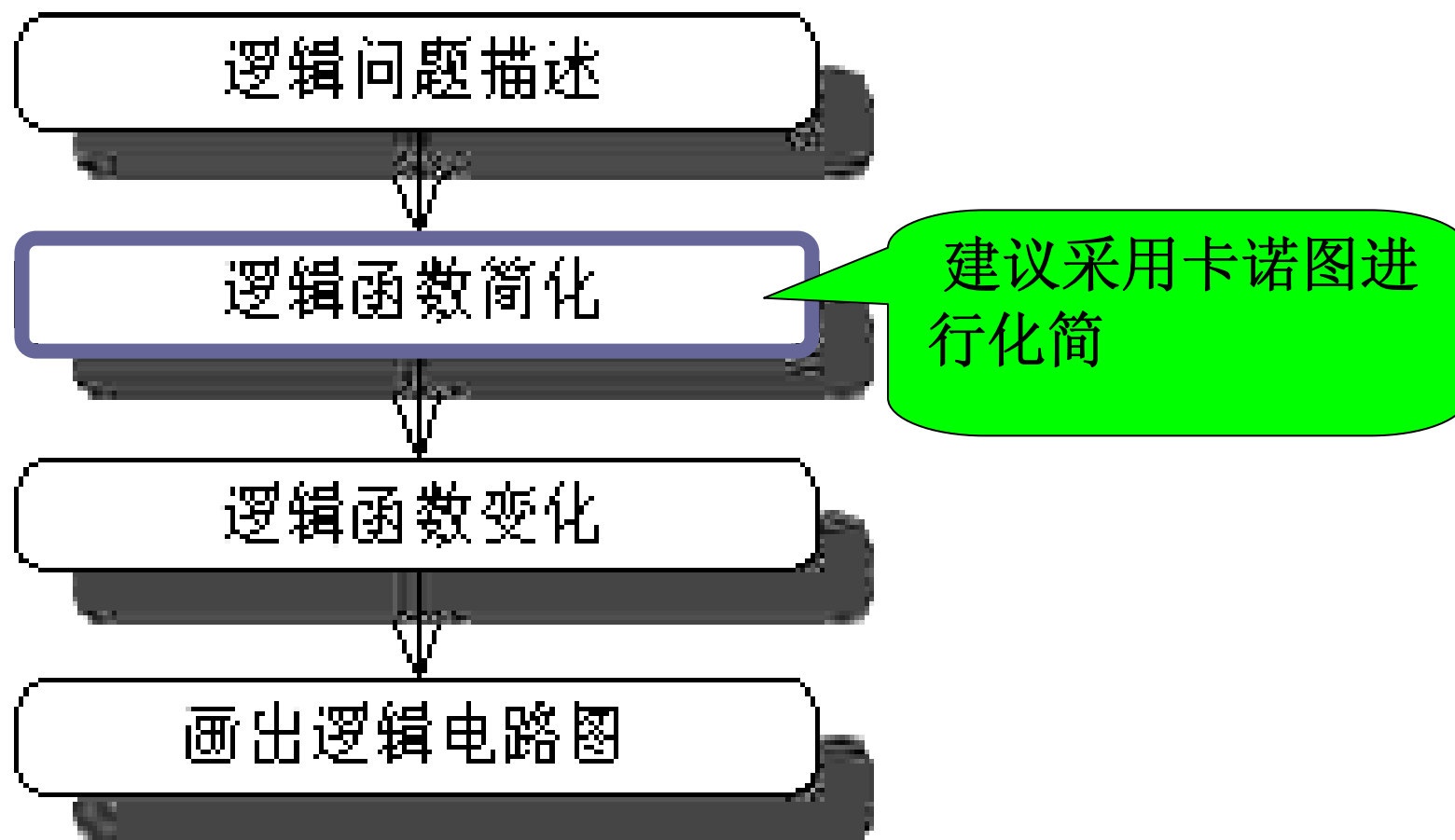
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$Y = \sum m^3(3,4,5,6,7)$



3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤



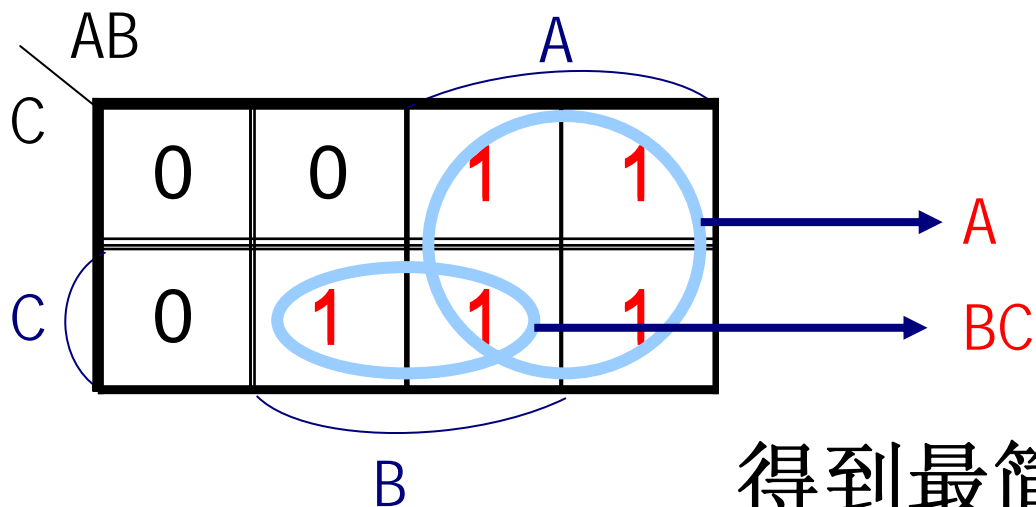


3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤

2) 逻辑函数化简:

$$Y = \sum m^3(3,4,5,6,7)$$



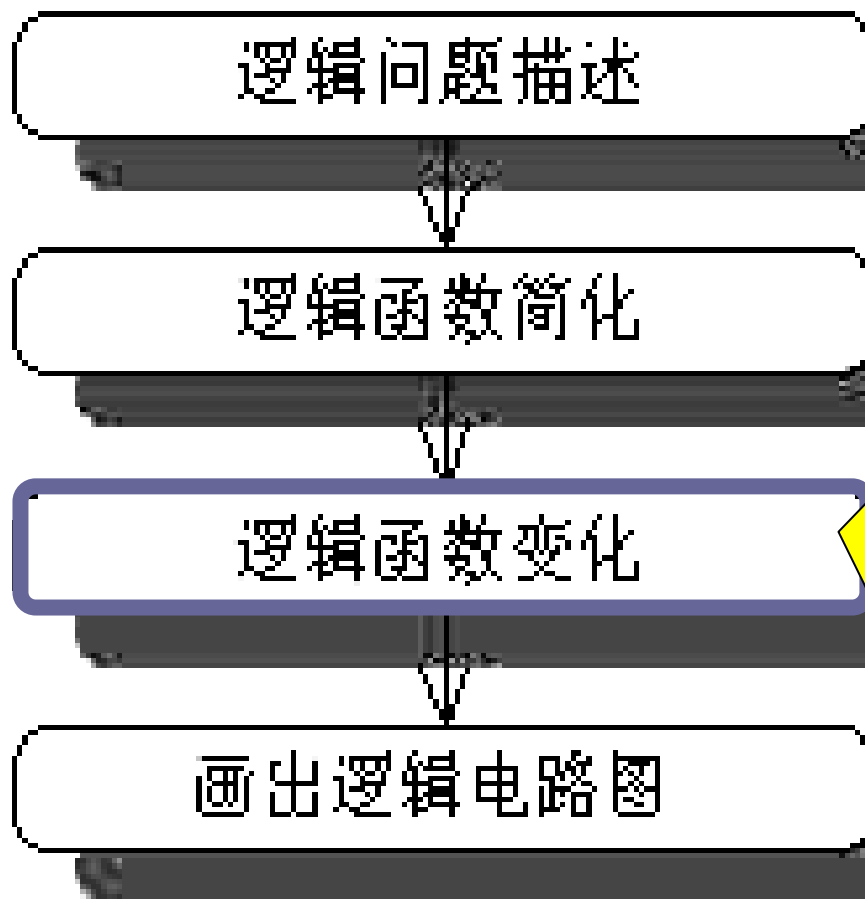
得到最简与-或式:

$$Y = A + BC$$



3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤



所谓逻辑函数的变换，就是根据小规模标准门电路器件的类型，将最简与-或表达式变换为与非门（74LS00，74LS20）、或非门（74LS02）、与或非门（74LS51）来实现的形式。



3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤

3) 逻辑函数变化（与非门实现）：

得到最简与-或式： $Y=A+BC$

结论：

- 原函数较简单时，对 F 两次求反可节省门电路；
- 反函数较简单时，对 \overline{F} 三次求反可节省门电路。但两次求反可获得较高的传输速度。

$$Y = A + BC = \overline{\overline{A + BC}} = \overline{\overline{A} \cdot \overline{BC}}$$



3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤

逻辑问题描述

逻辑函数简化

逻辑函数变化

画出逻辑电路图



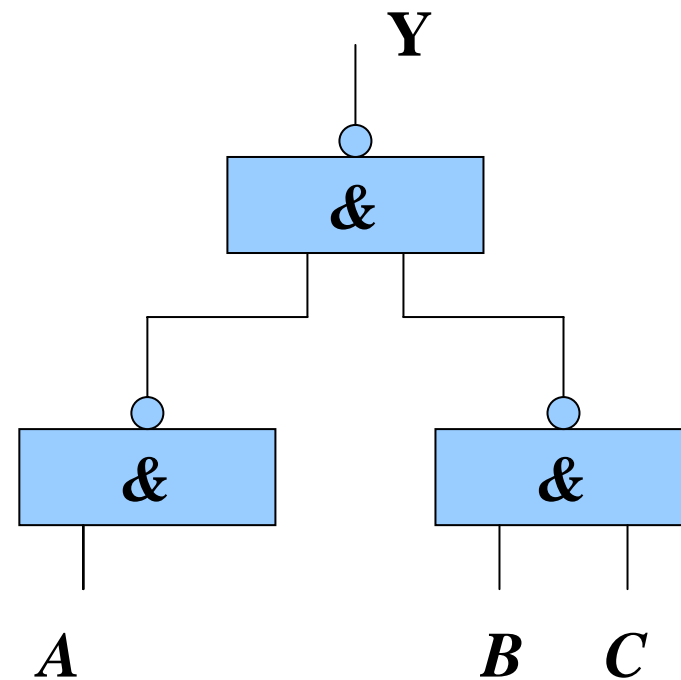
3 组合逻辑电路设计

3.2 组合逻辑电路设计步骤

例1:

4) 画出逻辑电路图:

$$Y = \overline{\overline{A} \cdot \overline{BC}}$$





4.2 组合逻辑电路设计

■ 组合逻辑电路设计方法

例1：设计一个3变量多数表决器

步骤：

- 1) 确定输入、输出变量，界定输入、输出变量含义，且列写真值表
- 2) 用卡诺图进行化简
- 3) 函数变换，用合适芯片实现
- 4) 画出逻辑电路图



4.2 组合逻辑电路设计

■ 组合逻辑电路设计方法

例2：设计一个含三台设备工作的故障显示器。
要求：

- a) 三台设备都正常工作，绿灯亮；
- b) 仅一台设备发生故障时，黄灯亮；
- c) 两台或两台以上设备同时发生故障时，红灯亮。

步骤：

- 1) 确定输入、输出变量，界定输入、输出变量含义，且列写真值表
- 2) 用卡诺图进行化简
- 3) 函数变换，用合适芯片实现
- 4) 画出逻辑电路图



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.1 包含无关条件的组合逻辑电路设计

- 在某些实际问题中，常常由于输入变量之间存在的相互制约或问题的某种特殊限定等，使得输入变量的某些取值根本不会出现，或者虽然可能出现，但对在这些输入取值组合下函数的值为1还是0并不关心。通常把这类问题称为包含无关条件的逻辑问题；



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.1 包含无关条件的组合逻辑电路设计

- 与这些输入取值组合对应的最小项称为**无关最小项**，简称**无关项**或者**任意项**；
- 描述这类问题的逻辑函数称为**包含无关条件的逻辑函数**。



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.1 包含无关条件的组合逻辑电路设计

例题：用与非门设计一个电路，判别8421码十进制的值 ≥ 5 。

■ CAI演示

步骤：

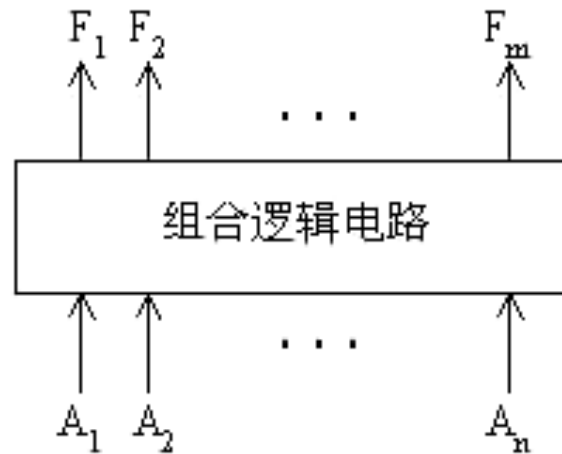
- 1) 确定输入、输出变量，界定输入、输出变量含义，且列写真值表
- 2) 用卡诺图进行化简
- 3) 函数变换，用合适芯片实现
- 4) 画出逻辑电路图



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.2 多输出函数的组合逻辑电路设计



特点：确定各输出函数的公用项，以使整个逻辑电路为最简，而不是片面地追求每个输出函数为最简。



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.2 多输出函数的组合逻辑电路设计

例4.9: 设计一个一位全加器。(P107)

(全加器是一个能对两个1位二进制数及来自低位的“进位”进行相加，产生本位“和”及向高位“进位”的逻辑电路。)

步骤:

- 1) 确定输入、输出变量，界定输入、输出变量含义，且列写真值表
- 2) 用卡诺图进行化简
- 3) 函数变换，用合适芯片实现
- 4) 画出逻辑电路图



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.3 无反变量提供的组合逻辑电路设计

■ 例4.10: P109

输入变量中无反变量时，用与非门实现如下逻辑函数：

$$F(A, B, C, D) = \overline{A}B + B\overline{C} + A\overline{B}C + AC\overline{D}$$



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.3 无反变量提供的组合逻辑电路设计

■ 例4.10: P109

解: (可简化电路)

$$\begin{aligned} F(A, B, C, D) &= \overline{A}B + B\overline{C} + A\overline{B}C + AC\overline{D} \\ &= B(\overline{A} + \overline{C}) + AC(\overline{B} + \overline{D}) \\ &= \overline{BAC} + \overline{ACBD} \\ &= \overline{\overline{BAC}} \bullet \overline{\overline{ACBD}} \end{aligned}$$



4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.4 考虑级数的组合逻辑电路设计

- 例：假定提供原、反变量，用与或非门和与非门实现逻辑函数

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

解：

- 1、用与非门实现
- 2、用与或非门及与非门实现
- 3、用一级与或非门实现



4.2 组合逻辑电路设计

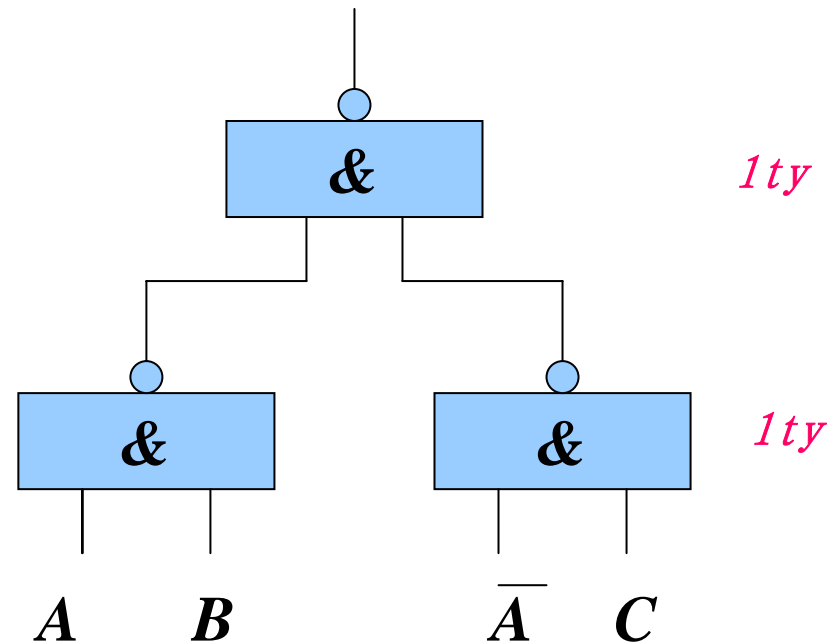
4.2.1 设计中几个实际问题的处理

4.2.1.4 考虑级数的组合逻辑电路设计

解：

1、用与非门实现

$$\begin{aligned} F &= AB + \bar{A}C \\ &= \overline{\overline{AB} \cdot \overline{\bar{A}C}} \end{aligned}$$





4.2 组合逻辑电路设计

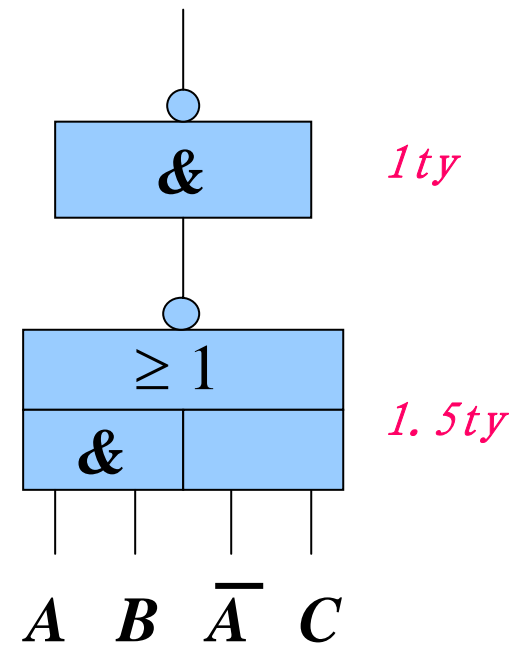
4.2.1 设计中几个实际问题的处理

4.2.1.4 考虑级数的组合逻辑电路设计

解：

2、用与或非门及与非门实现

$$\begin{aligned} F &= AB + \bar{A}C \\ &= \overline{\overline{AB + \bar{A}C}} \end{aligned}$$





4.2 组合逻辑电路设计

4.2.1 设计中几个实际问题的处理

4.2.1.4 考虑级数的组合逻辑电路设计

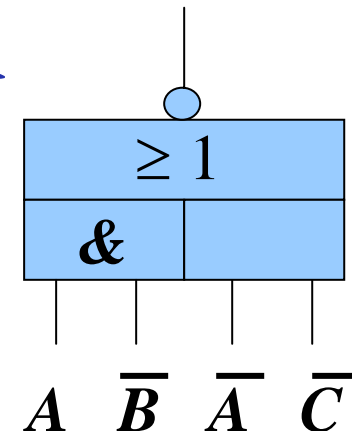
解：

3、用一级与或非门实现

思路：用卡诺图求出反函数

$$\overline{F} = A\overline{B} + \overline{A}\overline{C}$$

$$F = \overline{A\overline{B} + \overline{A}\overline{C}}$$



1.5ty



4.3 组合电路的险象

4.3.1 险象的产生

- 信号到达同一个门的时间有先有后，这种现象叫竞争
- 竞争产生险象，**CAI演示**
- 存在竞争现象的电路不一定都产生过渡干扰脉冲
- 电路中有竞争现象只是存在产生过渡干扰脉冲的危险而已，故称其为**竞争风险**。



4.2 组合逻辑电路设计

4.3 组合电路的险象

4.3.1 险象的消除

1. 加选通脉冲

- 该选通脉冲通常是 $P=0$ ，使电路处于封锁状态，只有在接收了输入信号并且电路达到了新的稳态之后，才有脉冲 $P=1$ ，允许电路输出。这就避免了竞争冒险的影响。引入选通脉冲的组合电路，输出信号只有在选通脉冲 $P=1$ 其间才有效。

2. 修改逻辑设计（增加冗余项法）

3. 增加惯性延时环节



4.3 常用的中规模组合逻辑构件

1. 加法器
2. 译码器
3. 数据选择器
4. 数码比较器

教材 第七章

P185



4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.1 半加器

4.3.1.2 全加器

4.3.1.3 串行加法器（时序逻辑中介绍）

4.3.1.4 并行加法器

- 1、并行输入串行进位加法器
- 2、并行输入并行进位加法器



4.3.1 加法器

4.3.1.1 半加器

只求本位和，不考虑低位的进位。实现半加操作的电路叫做半加器。

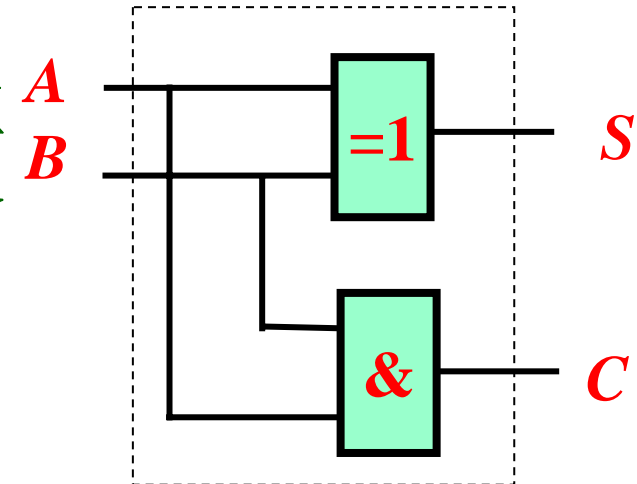
状态表

<i>A</i>	<i>B</i>	<i>C</i>	<i>S</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

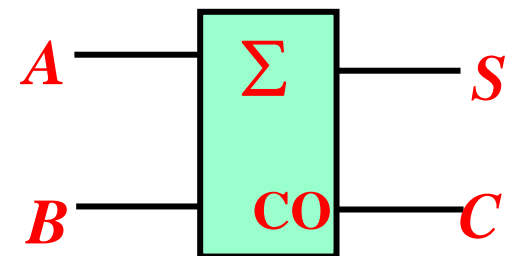
A、*B* 为两个加数；
C 为向高位的进位；
S 为半加和。

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = AB$$



半加器逻辑图



半加器逻辑符号



4.3.1.2 全加器

被加数、加数以及低位的进位三者相加称为“全加”，实现全加操作的电路叫做全加器。

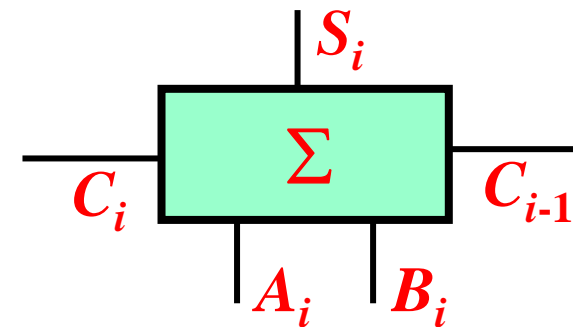
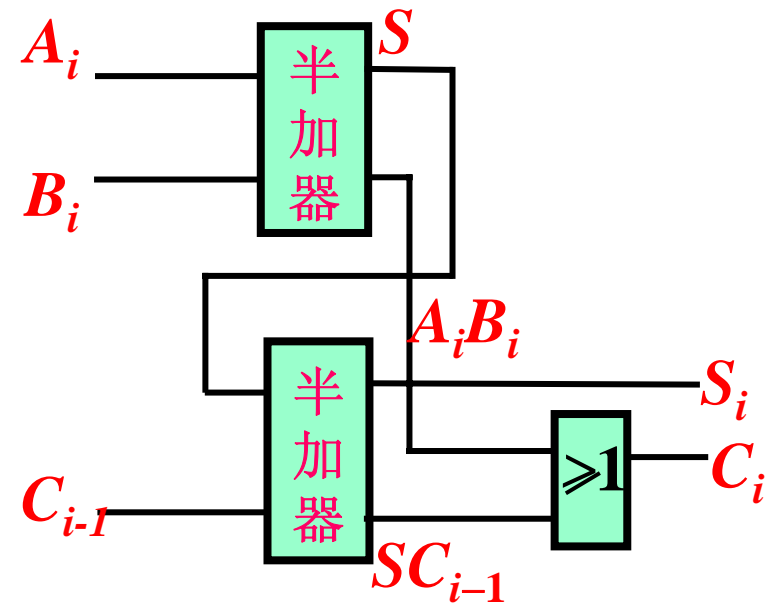
状态表

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

C_{i-1} : 来自低位的进位

C_i : 向高位的进位

根据半加器的原理，写出全加器的函数表达式



全加器逻辑符号



4.3 常用的中规模组合逻辑构件

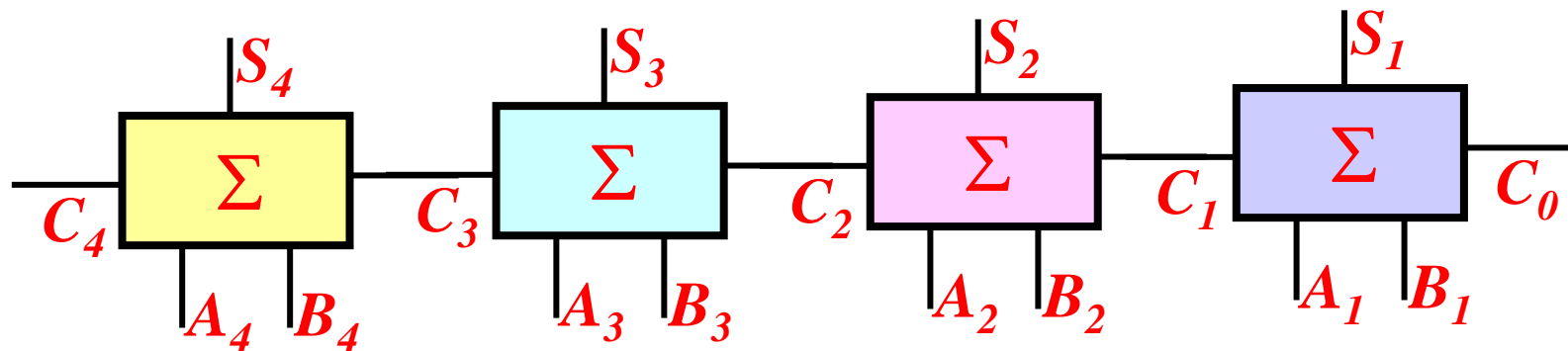
4.3.1 加法器

4.3.1.4 并行加法器

1、并行输入串行进位加法器

例：设计4位全加器

解：用4个全加器实现。



缺点：速度慢，需等待进位。



4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

2、并行输入并行进位加法器 (74LS283, T693)

并行进位又称为先行进位，超前进位，

例：设计4位全加器

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

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

特点：提高电路复杂度，换取较高运算速度。



4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

2、并行输入并行进位加 (74LS283, T693)

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

$$C_i = A_i B_i + (A_i + B_i) C_{i-1} = G_i + P_i C_{i-1}$$

$$C_1 = G_1 + P_1 \cdot C_0$$

$$\begin{aligned} C_2 &= G_2 + P_2 \cdot C_1 \\ &= G_2 + P_2 \cdot (G_1 + P_1 \cdot C_0) \end{aligned}$$

$$\begin{aligned} C_3 &= G_3 + P_3 \cdot C_2 \\ &= G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot C_0)) \end{aligned}$$

$$C_4 = \dots\dots$$

特点：提高电路复杂度，换取较高运算速度。

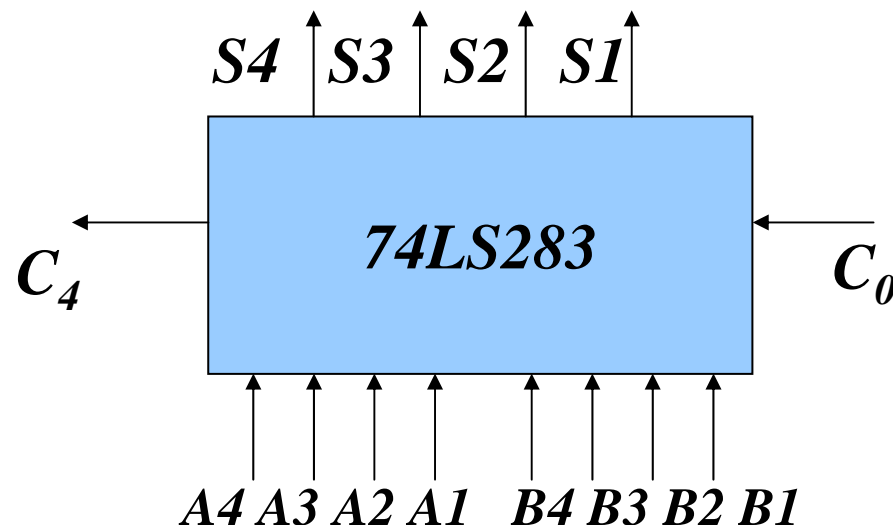


4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

2、并行输入并行进位加法器逻辑符号 (74LS283,T693)





4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

2、并行输入并行进位加法器 (74LS283,T693)

■ 74LS283的应用

例:

1. 将8421码转换成余3码
2. 将余3码转换成8421码(补码、反码)
3. 将以上1、2综合。



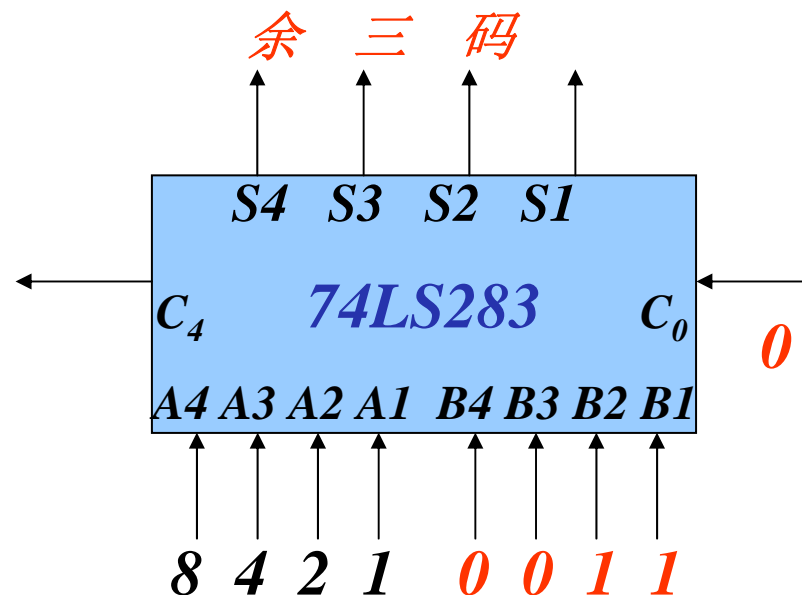
4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

■ 74LS283的应用

例：1、将8421码转换成余3码





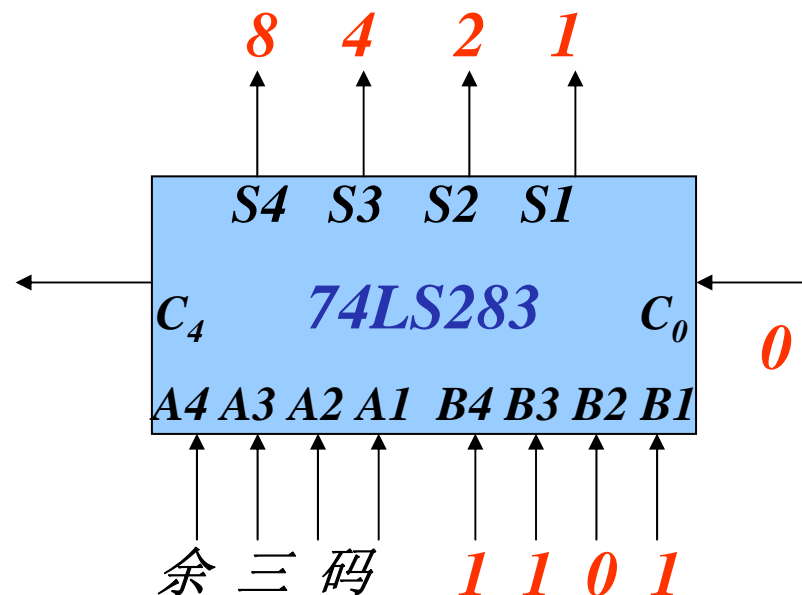
4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

■ 74LS283的应用

例： 2、将余3码转换成8421码(补码)





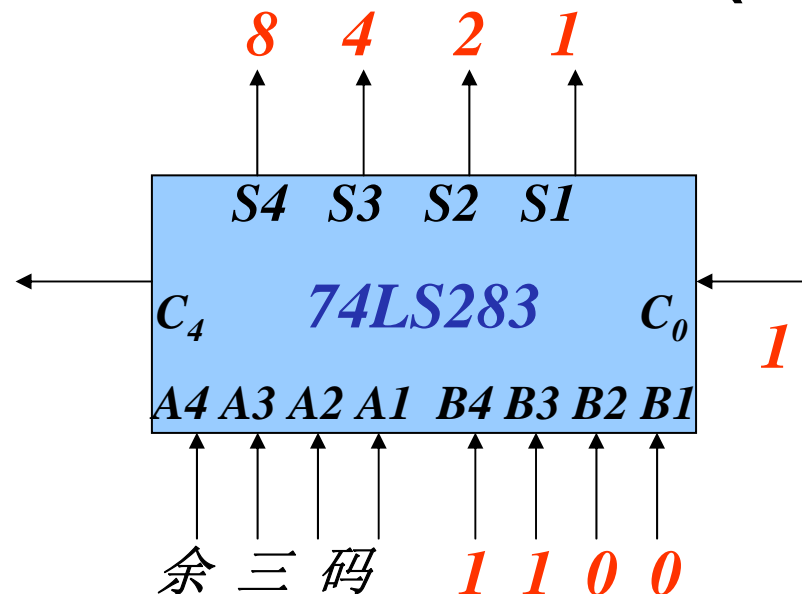
4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器

■ 74LS283的应用

例： 2、将余3码转换成8421码(反码)



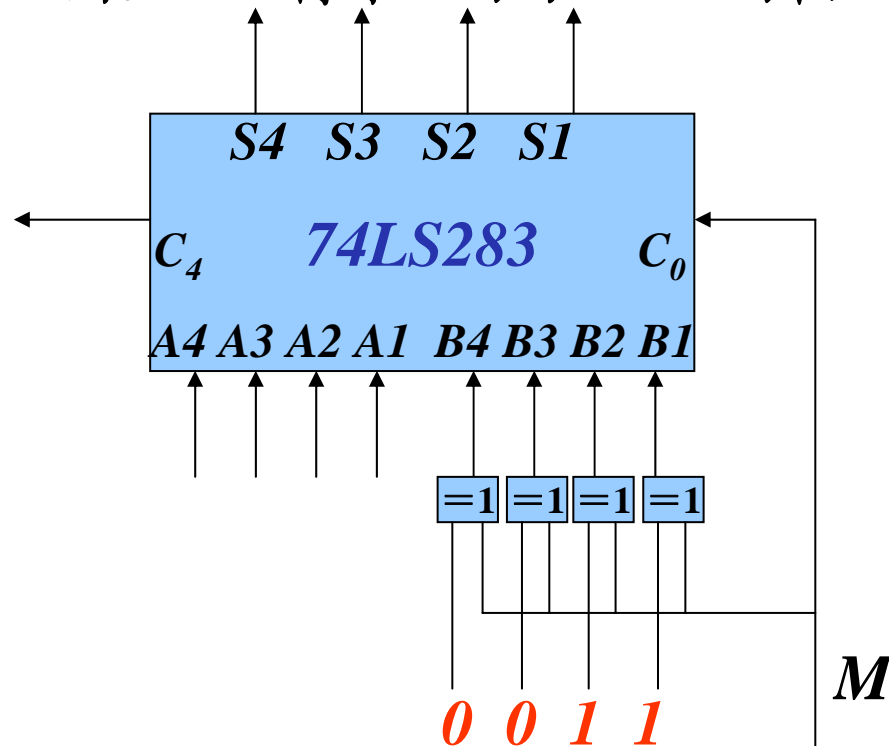


4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器（74LS283的应用）

例：3、将余3码与8421码相互转换（解1）



分析:

- 当 $M=0$ 时, 8421 码转化为余三码
- 当 $M=1$ 时, 余三码转化成 8421 码

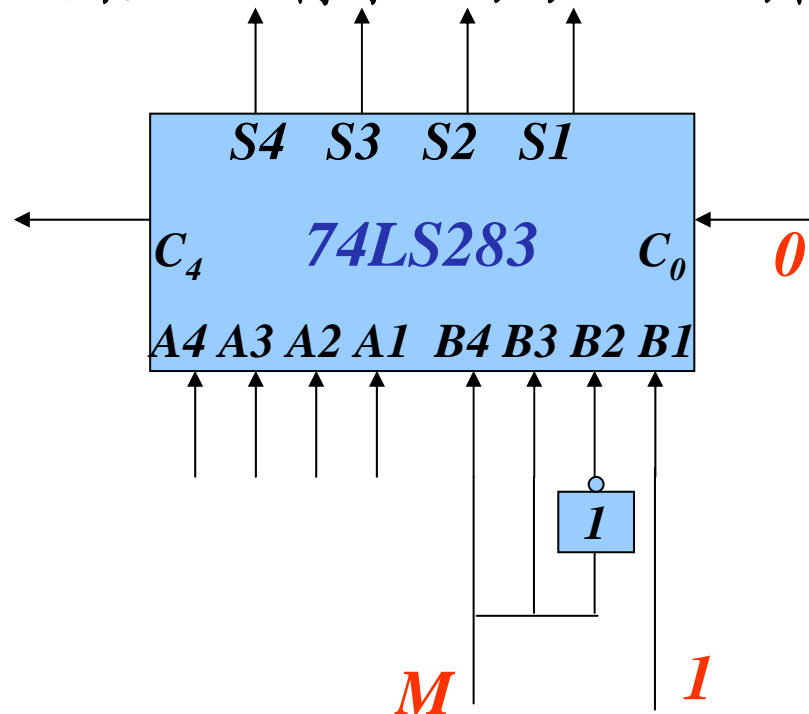


4.3 常用的中规模组合逻辑构件

4.3.1 加法器

4.3.1.4 并行加法器（74LS283的应用）

例：3、将余3码与8421码相互转换（解2）



分析：

- 当 $M=0$ 时，8421码转化为余三码
- 当 $M=1$ 时，余三码转化成8421码





4.3 常用的中规模组合逻辑构件

4.3.2 译码器

- 对具有特定含义的输入代码进行“翻译”，将其转换成相应的输出信号。
- 它的输入是二进制的代码，输出是一组高低电平信号，每输入一组不同的代码，只有一个输出端呈现有效信号，故通常称为多一译码器。



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

功能：将 n 位二进制输入变量译为 2^n 个输出函数，且每个输出函数对应于 n 个输入变量的一个最小项。

- 在使能输入端为有效电平时，对应每一组输入代码，仅一个输出端为有效电平，其余为无效电平。



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

有效电平可以是高电平也可以是低电平。

- 高电平有效：称高电平译码或“1”译中
- 低电平有效：称低电平译码或“0”译中

注：课程中都是“0”译中，即输出只有一个“0”，其他都为“1”



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

● 常用标准组件:

- ① **2—4线译码器**
- ② **3—8线译码器**
- ③ **4—16线译码器**
- ④ **4—10线译码器**



4.3 常用的中规模组合逻辑构件

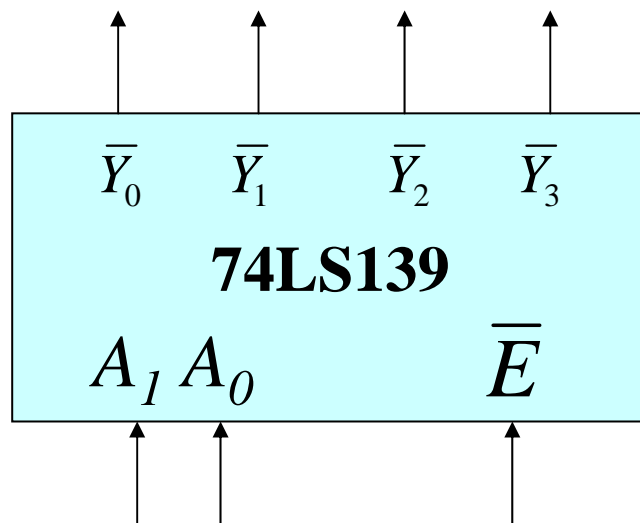
4.3.2 译码器

4.3.2.1 变量译码器

● 常用标准组件:

① 2—4线译码器

输 入	输 出			
$A \ B$	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3
0 0	0	1	1	1
0 1	1	0	1	1
1 0	1	1	0	1
1 1	1	1	1	0



通用表达式:

$$\bar{Y}_i = \bar{m}_i$$



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

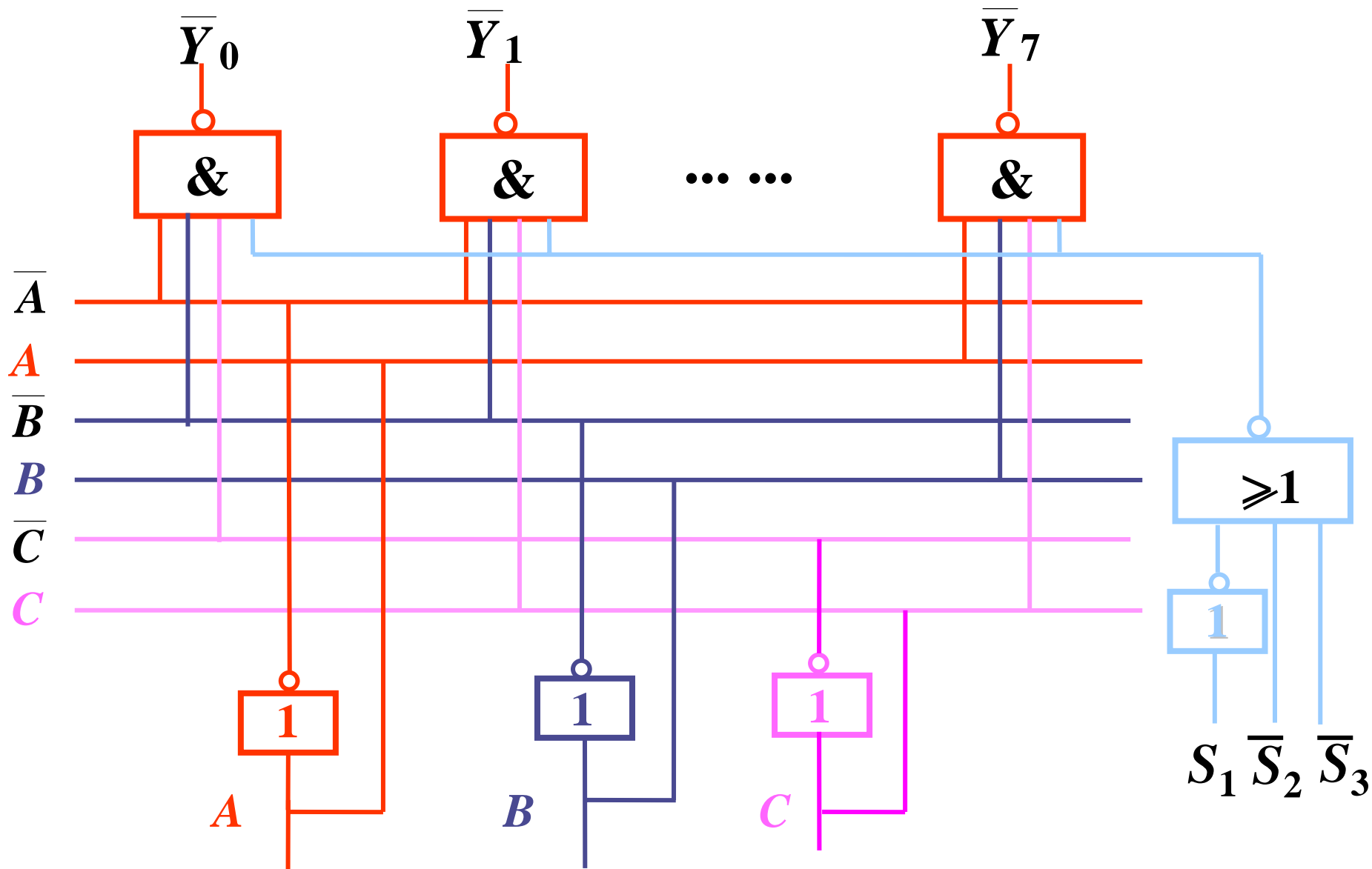
4.3.2.1 变量译码器

1、逻辑结构（**74LS138**为例）

74LS138是**3-8**译码器

4.3 常用的中规模组合逻辑构件

1) 3—8译码器逻辑图



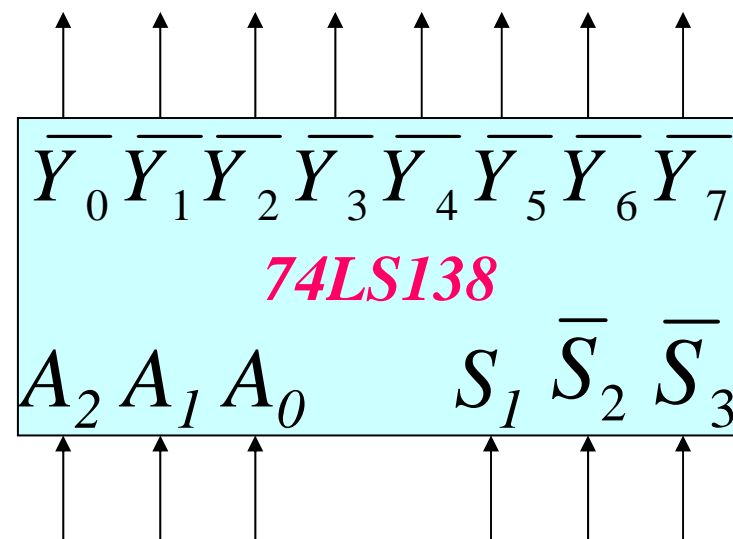


4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2) 3—8译码器符号 P189



■ CAI演示



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2) 含义

数据输入： **A,B,C**（需要进行译码的输入信号三个变量,有**8**种组合。）

译码输出： $\bar{Y}_0 \sim \bar{Y}_7$:对应一组输入代码组合,输出中只有一个是低电平。

- 例如 **ABC=000**时, $\bar{Y}_0 = 0$, 其余为高
- **ABC=111**时, $\bar{Y}_7 = 0$, 其余为高。



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2) 74LS138含义

控制输入： $S_1, \bar{S}_2, \bar{S}_3$ ：使能输入端，

当 $S_1=1, \bar{S}_2 + \bar{S}_3 = 0$ 时，允许译码器工作。

即有且仅有一个输出为**0**（低电平），其余都是**1**。



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2) 3—8译码器真值表 P189

译码器的状态表

输 入	输 出							
$A B C$	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	\overline{Y}_4	\overline{Y}_5	\overline{Y}_6	\overline{Y}_7
0 0 0	0	1	1	1	1	1	1	1
0 0 1	1	0	1	1	1	1	1	1
0 1 0	1	1	0	1	1	1	1	1
0 1 1	1	1	1	0	1	1	1	1
1 0 0	1	1	1	1	0	1	1	1
1 0 1	1	1	1	1	1	0	1	1
1 1 0	1	1	1	1	1	1	0	1
1 1 1	1	1	1	1	1	1	1	0

译码器逻辑式

$$\overline{Y}_0 = \overline{A}\overline{B}\overline{C}$$

$$\overline{Y}_1 = \overline{A}\overline{B}C$$

$$\overline{Y}_2 = \overline{A}B\overline{C}$$

$$\overline{Y}_3 = \overline{A}BC$$

...

通用表达式:

$$\overline{Y}_i = \overline{m}_i$$



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2、典型应用

1) 扩展功能

例1：用**3—8**译码器构成**6—64**译码器

例2：用两个**3—8**译码器构成一个**4—16**译码器

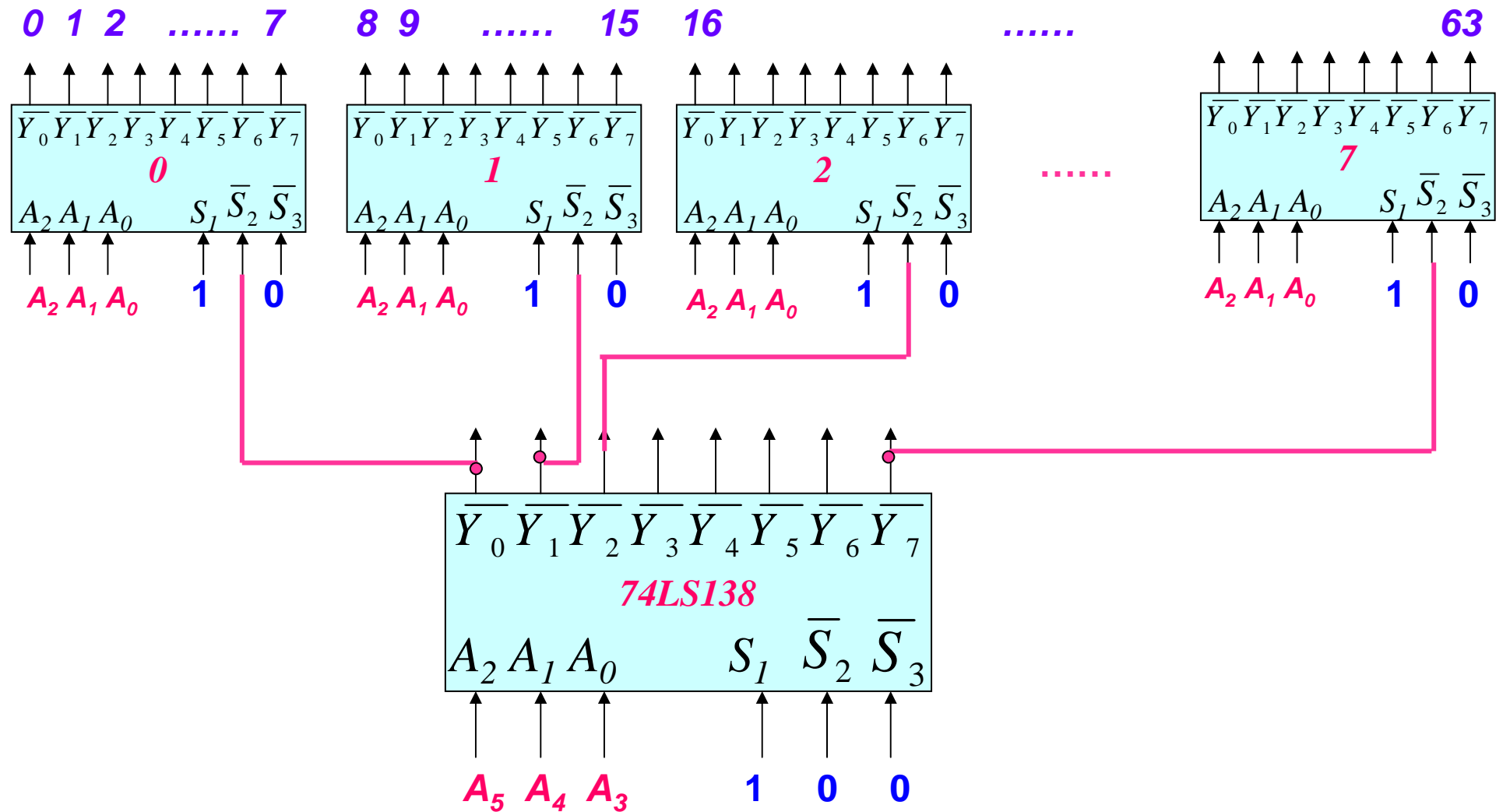
思考：

例3：用两个**2—4**译码器构成一个**3—8**译码器

例4：用**2—4**及**3—8**译码器构成**5—32**译码器

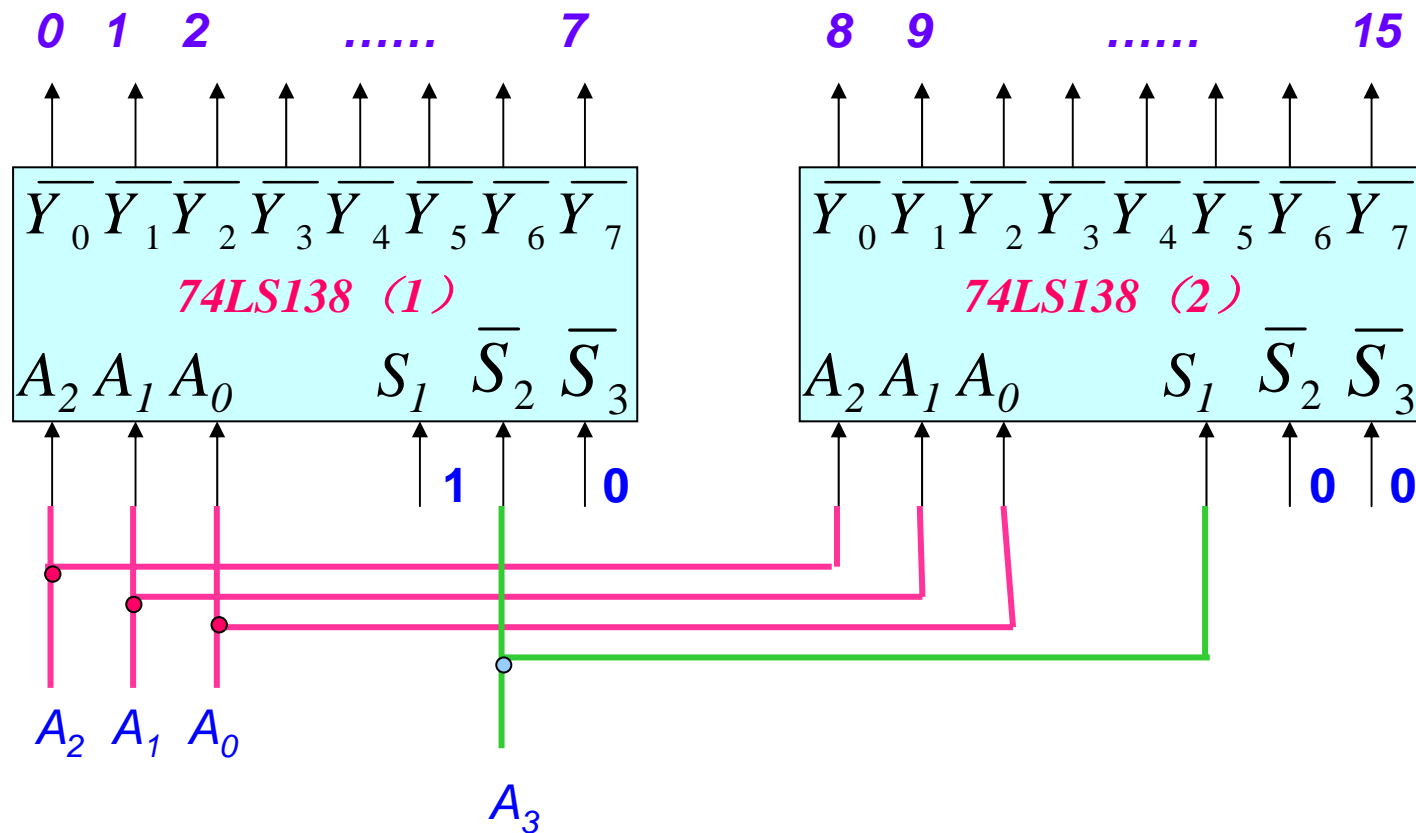


例1：用3—8译码器构成6—64译码器





例2：用两个3—8译码器构成一个4—16译码器





4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2、典型应用

1) 扩展功能

例1：用**3—8**译码器构成**6—64**译码器

例2：用两个**3—8**译码器构成一个**4—16**译码器

思考：

例3：用两个**2—4**译码器构成一个**3—8**译码器

例4：用**2—4**及**3—8**译码器构成**5—32**译码器



4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.1 变量译码器

2、典型应用

2) 构成函数发生器

例1：用**74LS138**构成一位全加器

$$S_i = \sum m^3(1,2,4,7)$$

$$C_i = \sum m^3(3,5,6,7)$$

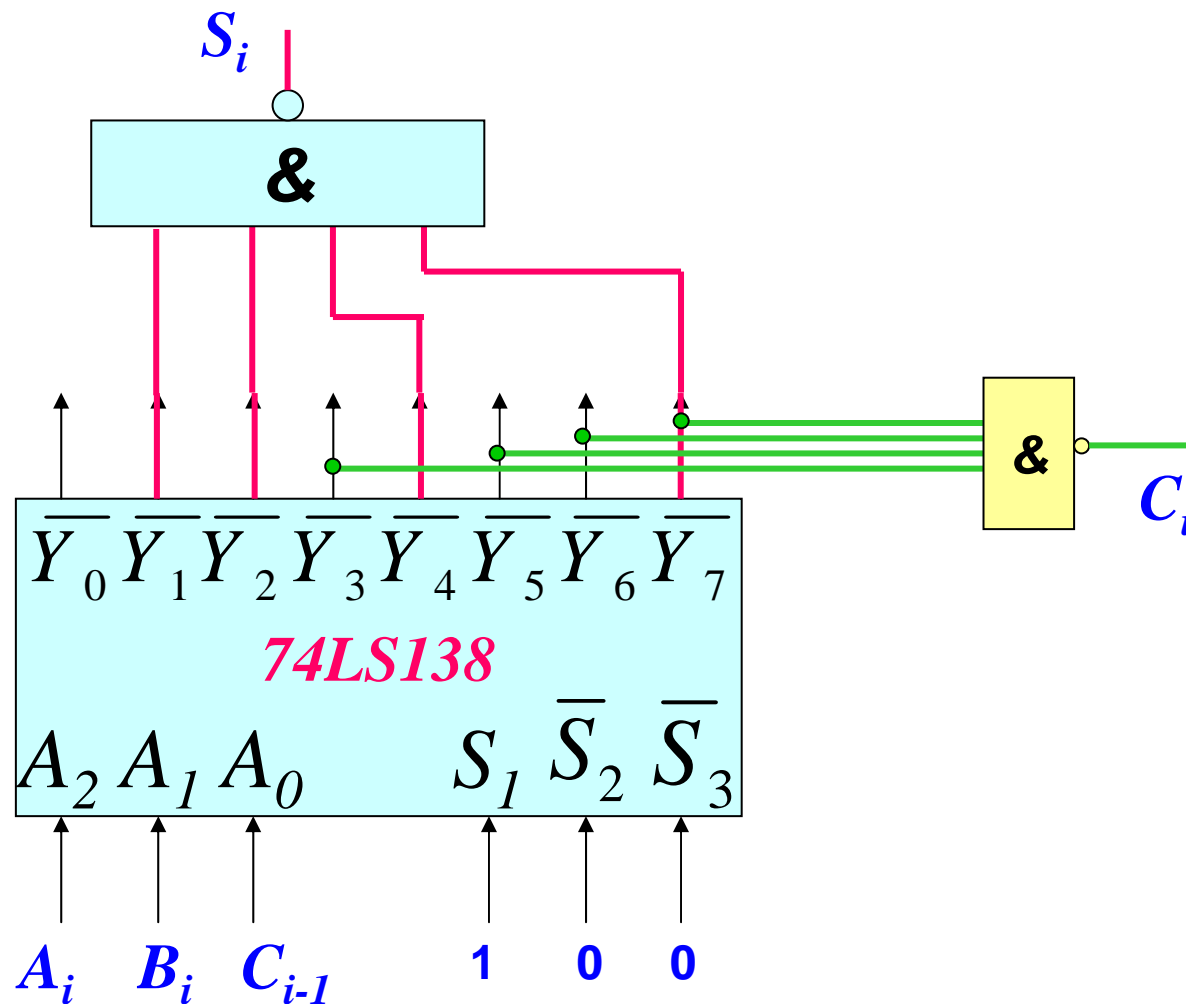


2) 构成函数发生器

例1: 用**74LS138**构成一位全加器

$$S_i = \sum m^3(1, 2, 4, 7)$$

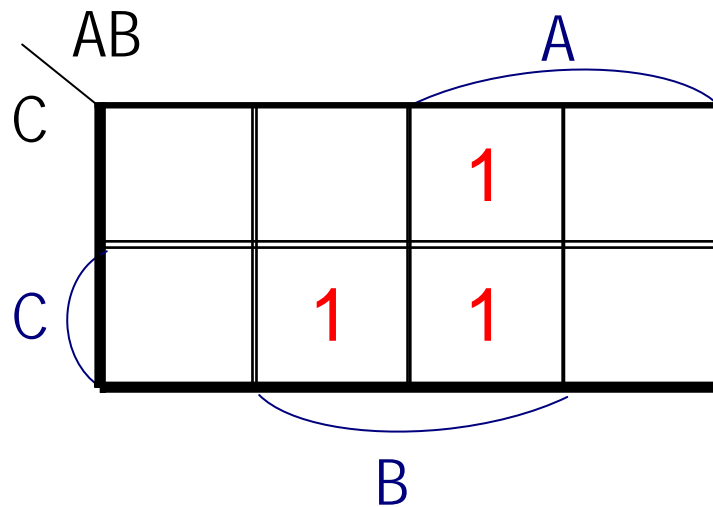
$$C_i = \sum m^3(3, 5, 6, 7)$$



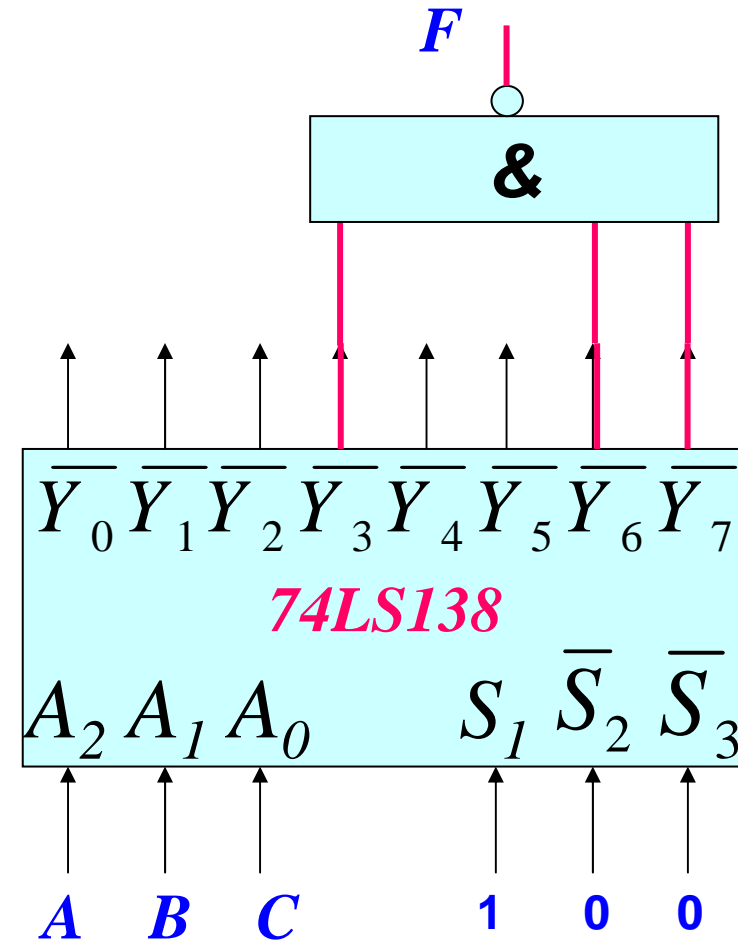


2) 构成函数发生器

例2: 用**74LS138**实现函数 $F = AB + BC$



$$F = m_3 + m_6 + m_7$$

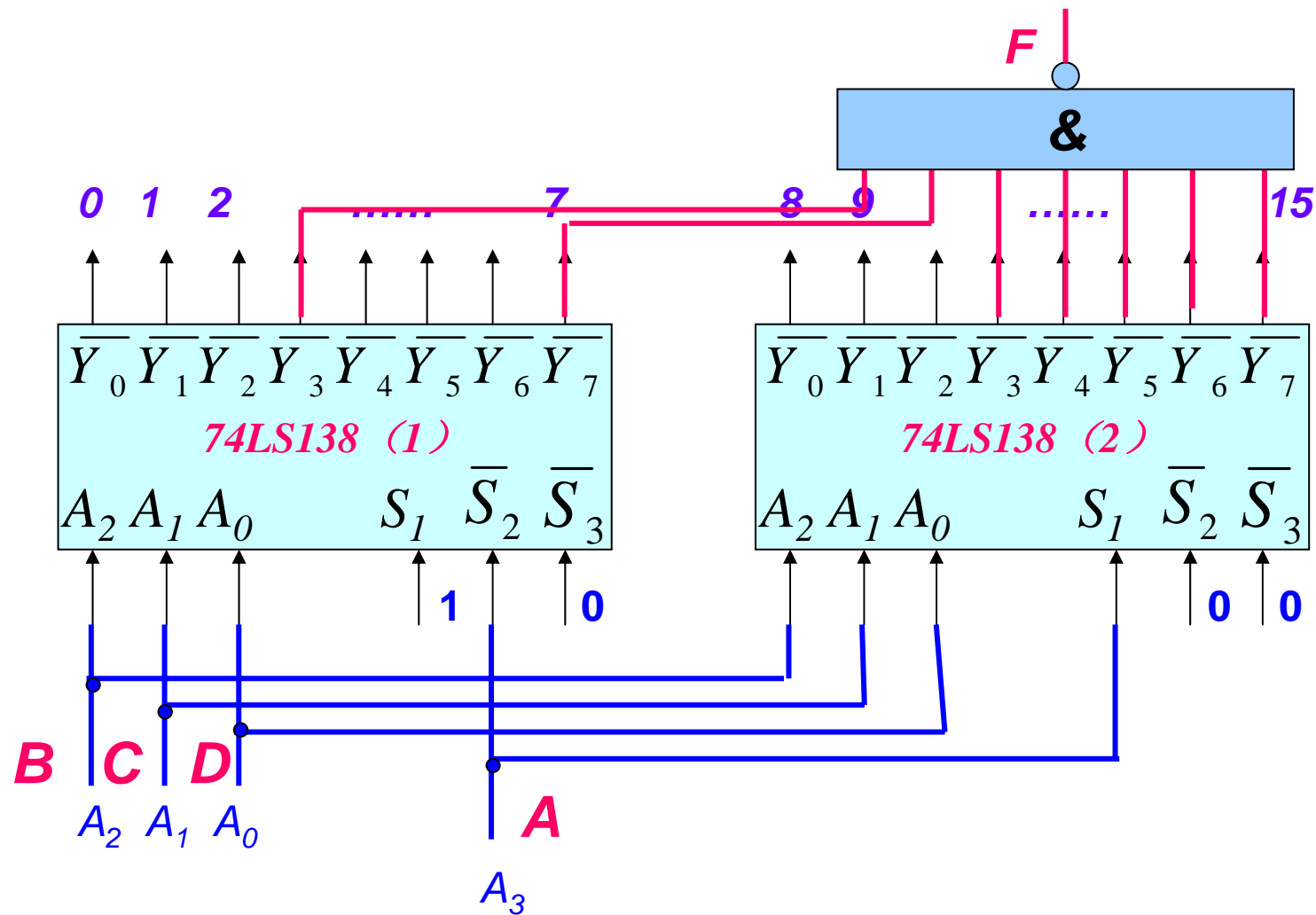


理论上一块**74LS138**可以实现**3**变量的任意函数



例3：用74LS138实现函数

$$F = AB + CD = \sum m^4(3,7,11,12,13,14,15)$$





4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.2 码制变换译码器

- 将一种代码形式转换成另一种代码形式的译码器。
- **二—十进制译码器**（又称作**4线—10线译码器**），**功能是将4位BCD码的10组代码翻译成10个十进制数字符号对应的输出信号。**

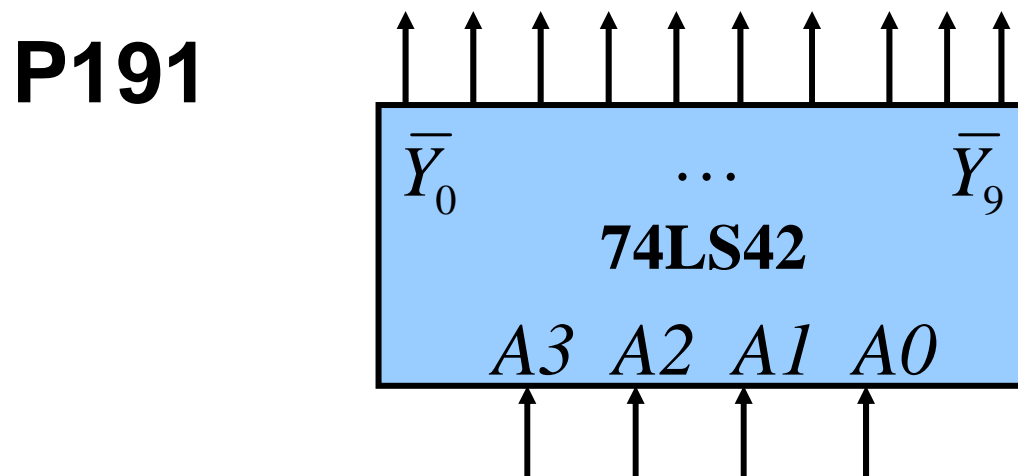


4.3 常用的中规模组合逻辑构件

4.3.2 译码器

4.3.2.2 码制变换译码器

■ 二—十进制译码器逻辑符号 (T331)



■ 逻辑表达式分析 (分析 \overline{Y}_0 与 \overline{Y}_9)



4.3.2.2 码制变换译码器

例：用二—十进制译码器构成3—8译码器
提示：真值表，P191

[illegible]



4.3.2 译码器

4.3.2.3. 数字显示译码器——七段显示译码器

七段显示译码器的功能是把 **8421** 二—十进制代码译成对应于数码管的七个字段信号，驱动数码管显示出相应的十进制数码。

74LS47输出为低电平有效，即输出为**0**时，对应字段点亮；输出为**1**时对应字段熄灭。该译码器能驱动七段显示器显示**0~15**共**16**个数字的字形。

74LS47 译码器有四个输入端 A_0, A_1, A_2, A_3 和七个输出端 $\overline{a} \overline{b} \overline{c} \overline{d} \overline{e} \overline{f} \overline{g}$ 。



4.3.2 译码器

4.3.2.3. 数字显示译码器——七段显示译码器

三个输入控制端：

\overline{LTI} ：测试输入端，用来检查显示管的七段是否都能正常工作。当 $\overline{BI} = 1$ 且 $\overline{LTI} = 0$ 时，七段均点亮，数码管显示 8。

\overline{BI} ：灭灯输入端，当它等于零时，数码管各段均熄灭。

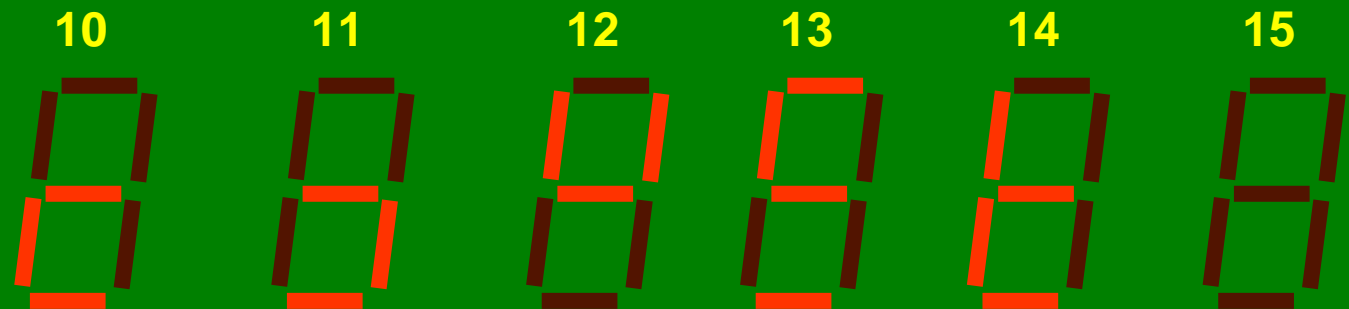
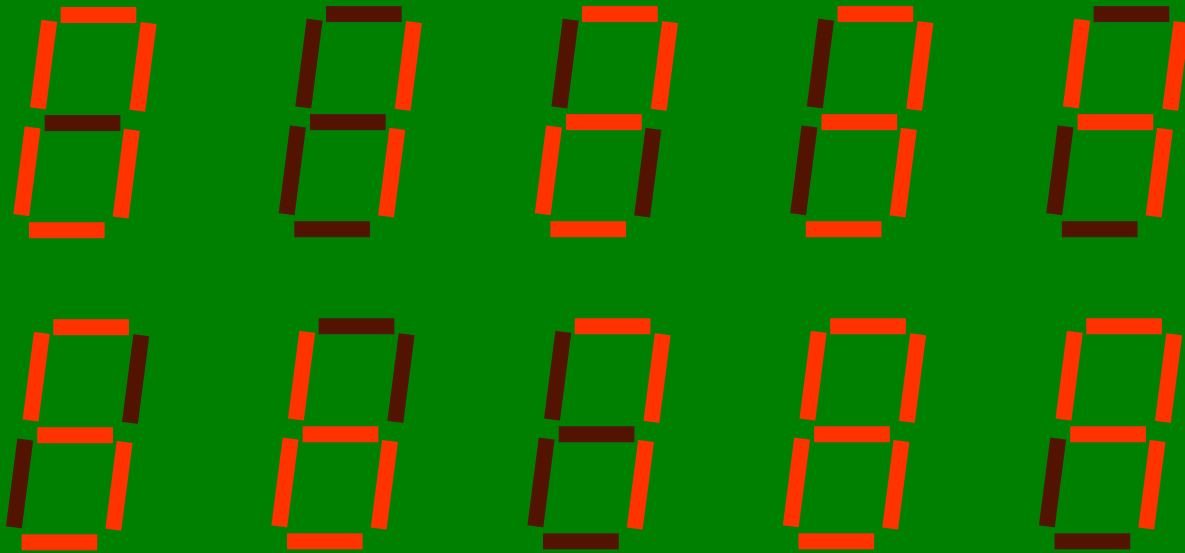
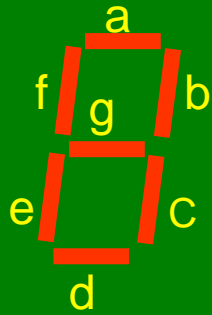
\overline{RBI} ：灭零输入端，当 $\overline{BI} = 0$ ， $\overline{LTI} = 1$ ， $\overline{RBI} = 0$ ，只有当 $A_3 \sim A_0$ 均为零，数码管七段均熄灭。用来消除无效 0。



74LS247 七段字形显示译码器的状态表

$A_3A_2A_1A_0$	\bar{a}	\bar{b}	\bar{c}	\bar{d}	\bar{e}	\bar{f}	\bar{g}	显示 字形
0 0 0 0	0	0	0	0	0	0	1	
0 0 0 1	1	0	0	1	1	1	1	
			⋮					
1 0 0 0	0	0	0	0	0	0	0	
1 0 0 1	0	0	0	1	0	0	0	

七段显示器件结构

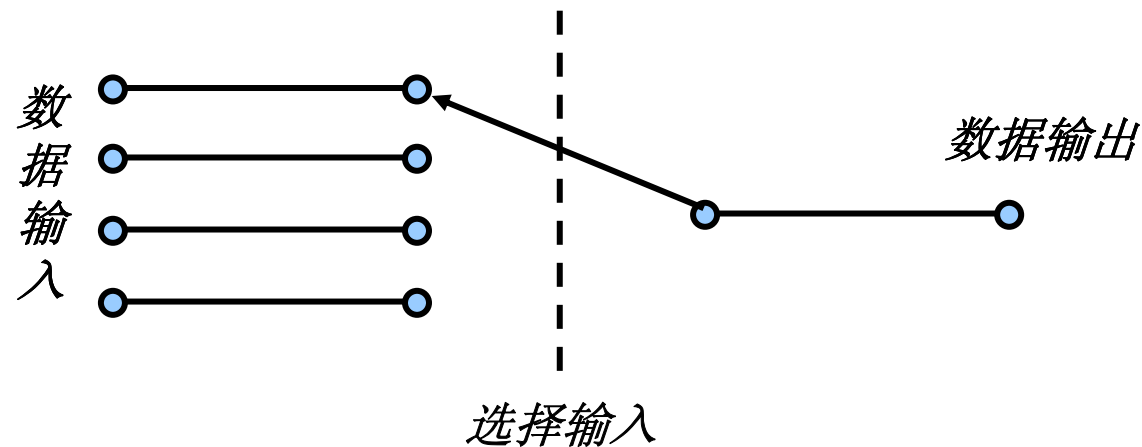




4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

- 又称多路选择器或多路开关，数据选择器是多路输入，选择一路输出的组合逻辑构件。



四选一



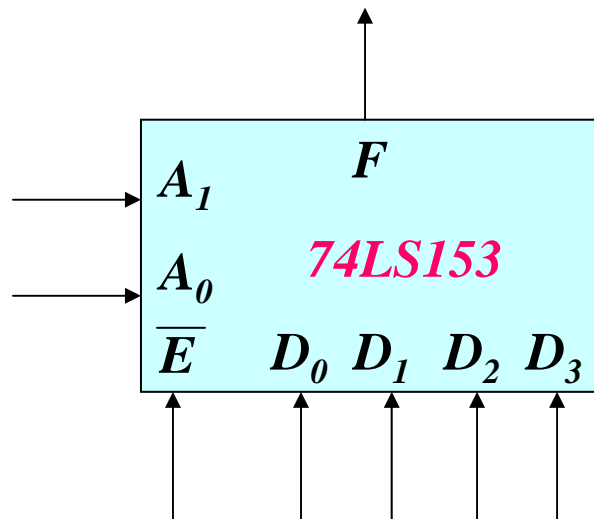
4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.1 逻辑功能

1. 四选一数据选择器

惯用符号:





4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.1 逻辑功能

1. 四选一数据选择器

- **74LS153**是四选一多路开关，即一个芯片内集成了两个同样结构的四选一多路开关。

逻辑图：四个输入端，一个输出端

表达式:
$$F = \overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3 = \sum_{i=0}^3 m_i D_i$$



4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.1 逻辑功能

1. 四选一数据选择器

功能表:

选择输入		数据输入				输出控	输出
A1	A0	D ₀	D ₁	D ₂	D ₃	$\overline{\text{E}}$	F
×	×	×	×	×	×	1	0
0	0	D ₀	×	×	×	0	D ₀
0	1	×	D ₁	×	×	0	D ₁
1	0	×	×	D ₂	×	0	D ₂
1	1	×	×	×	D ₃	0	D ₃

为0时工作,
为1时静止



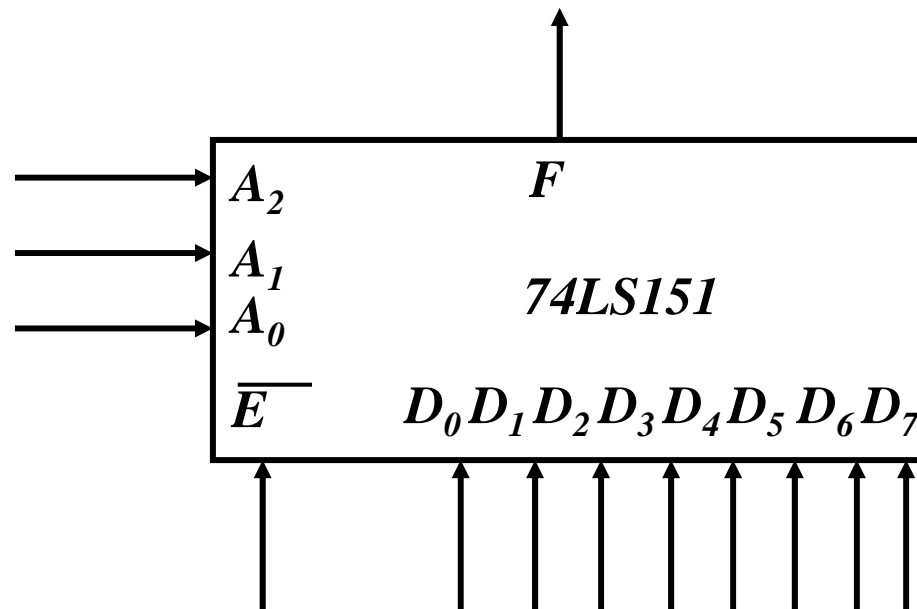
4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.1 逻辑功能

2. 八选一数据选择器

惯用符号:





4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.1 逻辑功能

2. 八选一数据选择器

表达式:

$$F = \sum_{i=0}^7 m_i D_i$$

功能表:

\bar{E}	A_2	A_1	A_0	F
1	×	×	×	0
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7



4.3 常用的中规模组合逻辑构件

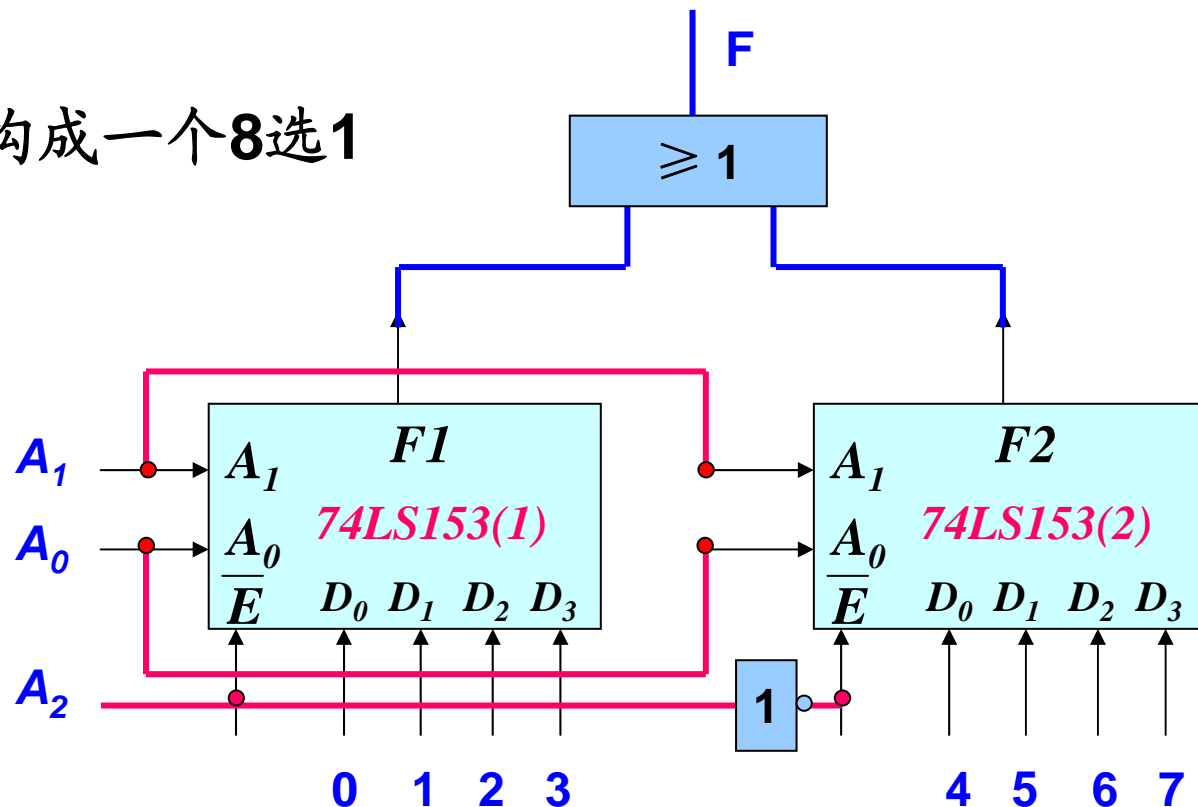
4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

1、扩展功能

例：用2个4选1构成一个8选1

方法一：





4.3 常用的中规模组合逻辑构件

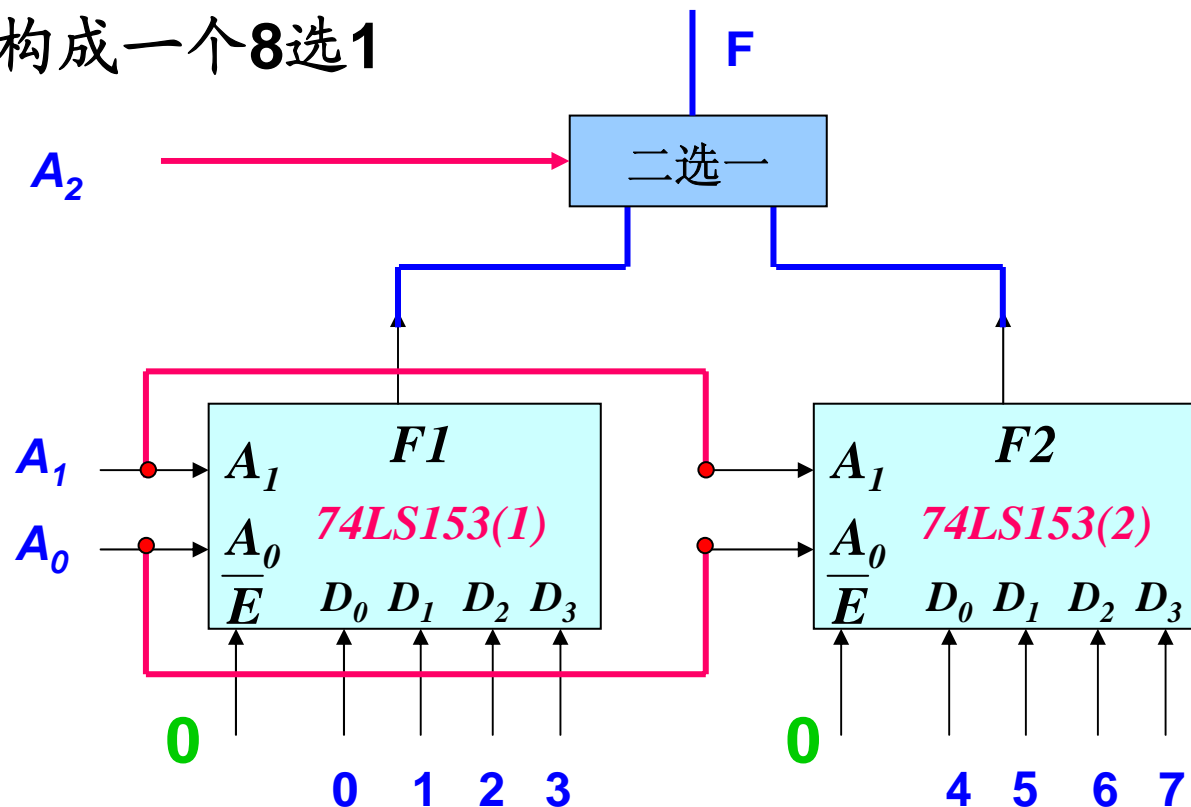
4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

1、扩展功能

例：用2个4选1构成一个8选1

方法二：





4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

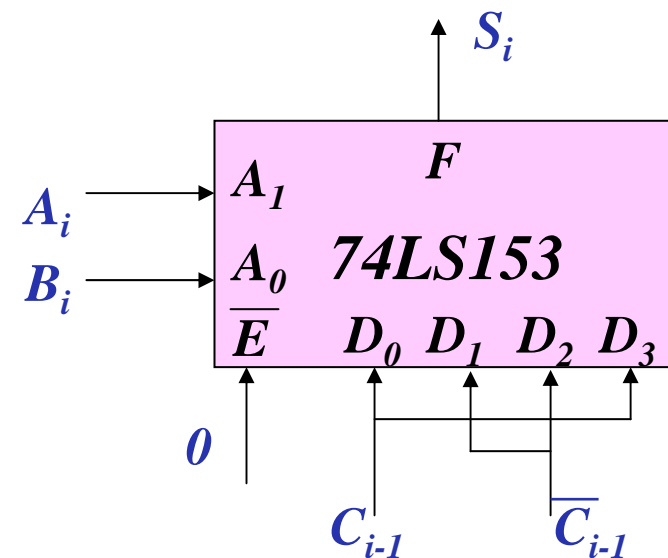
2、构造函数发生器 (重点)

例：用4选1构造一位全加器

分析：

$$S_i = \sum m^3(1,2,4,7)$$

$$C_i = \sum m^3(3,5,6,7)$$





4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

2、构造函数发生器 (重点)

例：实现函数 $F = AB + A\bar{C}$

方法一：代数法

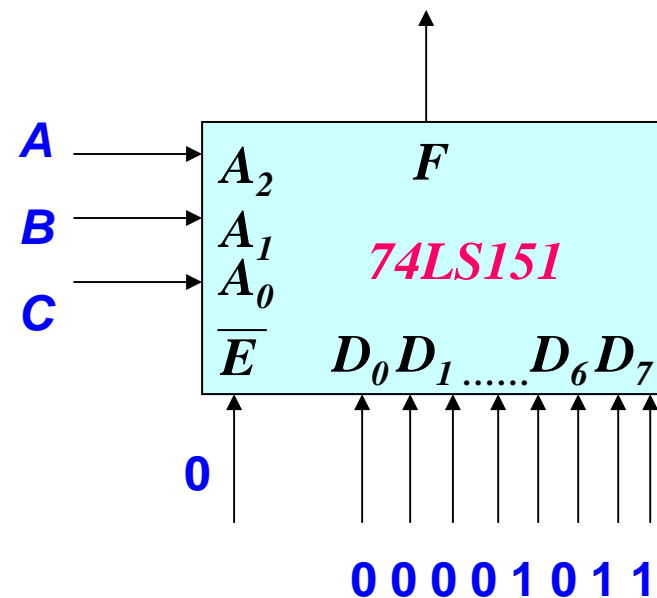
解：化成最小项表达式

$$F = \sum m^3(6,7,4)$$

1) 用8选1实现

分析：8选1表达式：

$$F = \sum_{i=0}^7 m_i D_i$$



解： $D_6 = D_7 = D_4 = 1$
 $D_0 = D_1 = D_2 = D_3 = D_5 = 0$



4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

2、构造函数发生器 (重点)

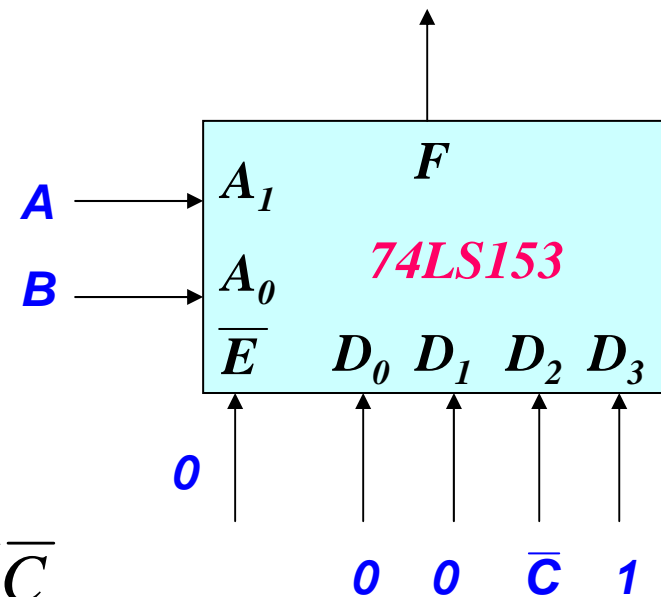
例：实现函数 $F = AB + A\bar{C}$

方法一：代数法

2) 用4选1实现

分析：

$$\begin{aligned} F &= AB + A\bar{C} \\ &= \underline{AB} + \underline{AB}\bar{C} + \underline{A\bar{B}}\bar{C} \end{aligned}$$



将AB找出来，当作地址输入



4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

2、构造函数发生器 (重点)

例：实现函数 $F = AB + A\bar{C}$

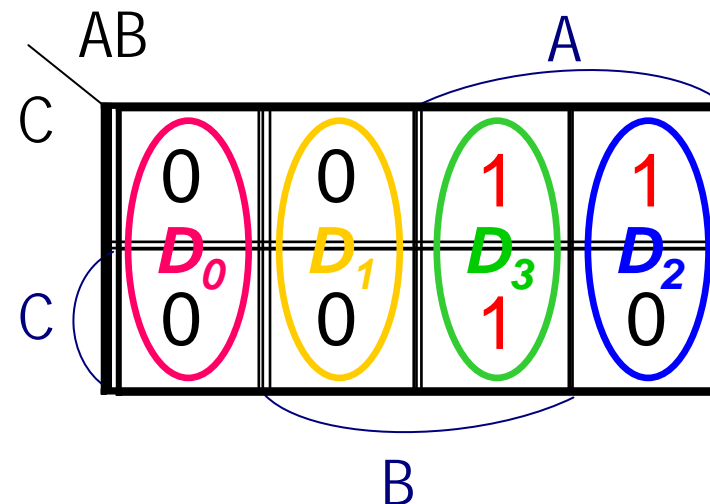
方法二：卡诺图法

1) 用8选1实现

解：三变量卡诺图

2) 用4选1实现

解：划分子卡诺图





4.3 常用的中规模组合逻辑构件

4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

2、构造函数发生器 (重点)

例：用4选1实现4变量逻辑函数的功能，函数式为

$$F(A, B, C, D) = \sum m^4(1, 2, 4, 9, 10, 11, 12, 14, 15)$$

方法一：代数法

先利用卡诺图化简，再找出作为地址输入的变量

方法二：卡诺图法

四变量卡诺图中划分子卡诺图

方法三：功能表法

先利用卡诺图化简，再列写真值表



4.3 常用的中规模组合逻辑构件

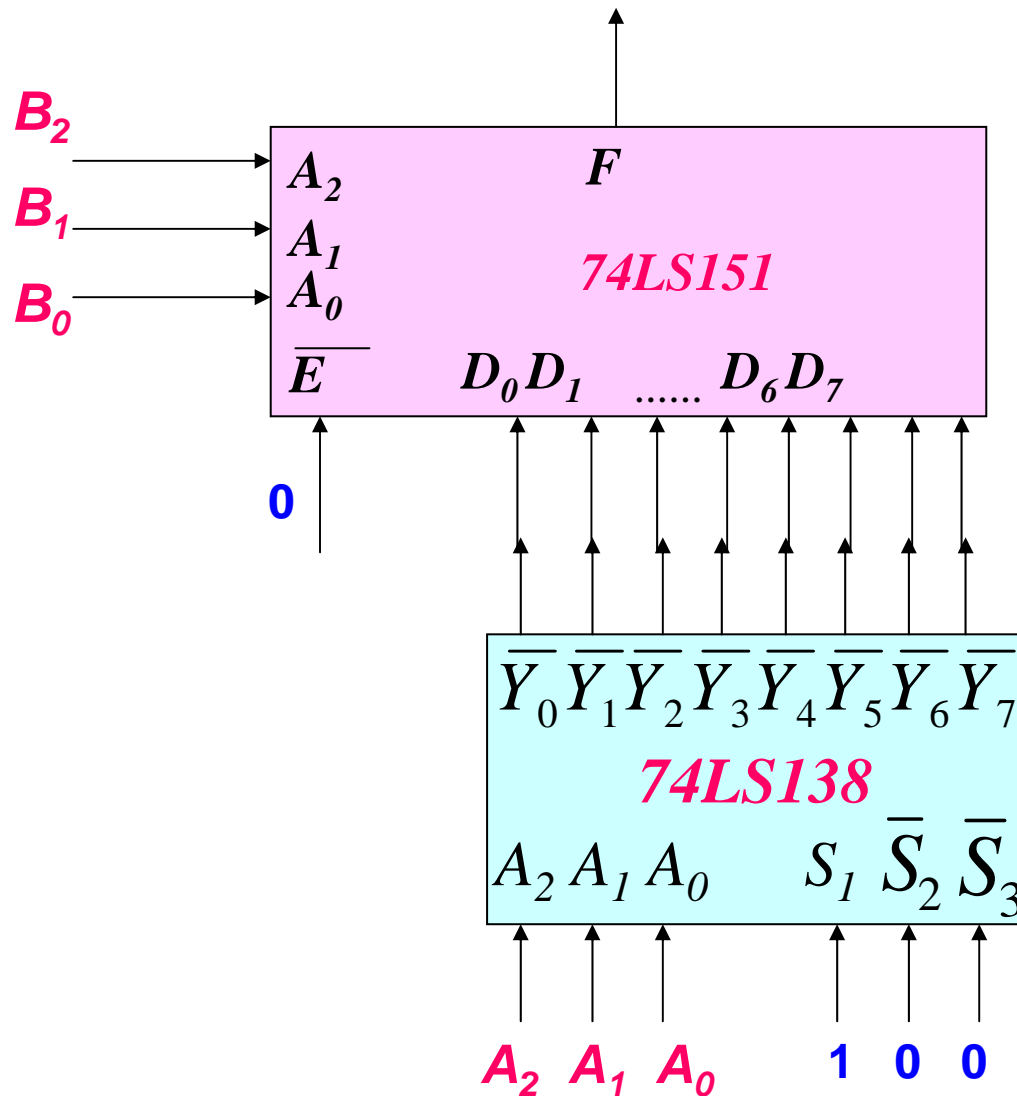
4.3.4 数据选择器 (MUX)

4.3.4.2 典型应用

例3、用3-8译码器和8路选择器构造一个3位二进制数等值比较器。

解：设比较三位二进制数A和B，

结论：当F=0时，
A=B。





作业一:

- 预习：第七章，第一节

- 习题四

4.2 4.3 4.4 4.8 4.9



作业二:

- 预习: 触发器
- 习题四 P226

7.1 7.4 7.5 7.8