

第 4 章

组合逻辑电路的 分析与设计

逻辑电路

组合
逻辑电路

输出只取决于当前的输入。

门电路，不存在记忆元件。

时序
逻辑电路

输出取决于当前的输入和原来的状态。

组合电路、记忆元件。

• 4.1 组合逻辑电路手工分析

• 4.2 组合逻辑电路手工设计方法

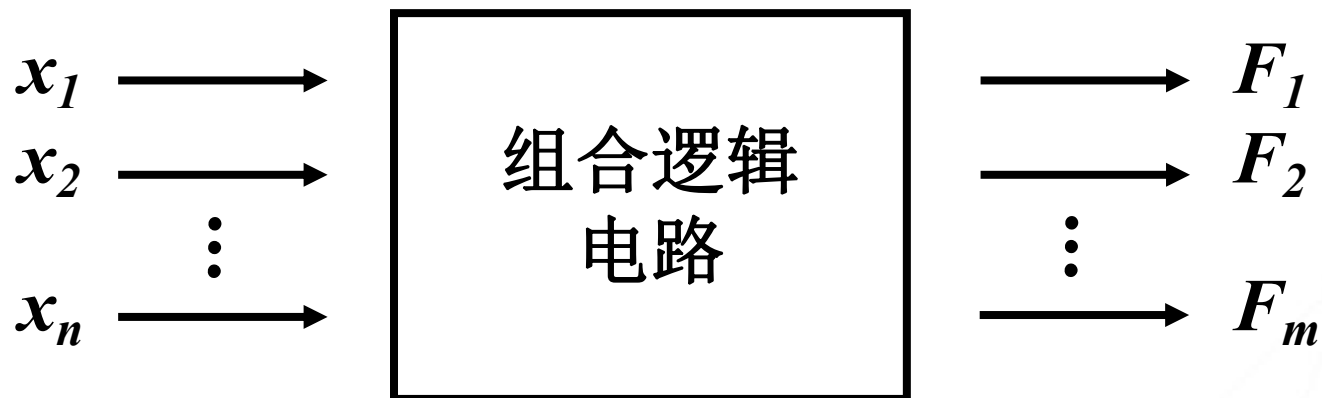
• 4.3 编码器

• 4.4 译码器

•

组合逻辑电路手工分析

1. 组合逻辑电路的定义



组合逻辑电路的框图

$$F_i = f_i(x_1, x_2, \dots, x_n) \quad (i=1, 2, \dots, m)$$

组合逻辑电路手工分析

1. 组合逻辑电路的定义

逻辑表达式：与或式，形式不惟一，通过变换可实现用不同门电路组成逻辑图。

真值表：具有惟一性。自动设计中，用真值表描述逻辑有利于计算机对逻辑的自动设计。

卡诺图：化简逻辑函数的主要工具。

逻辑图：一个逻辑表达式可用不同逻辑图实现，只反映逻辑功能，不反映电路的时序特性。

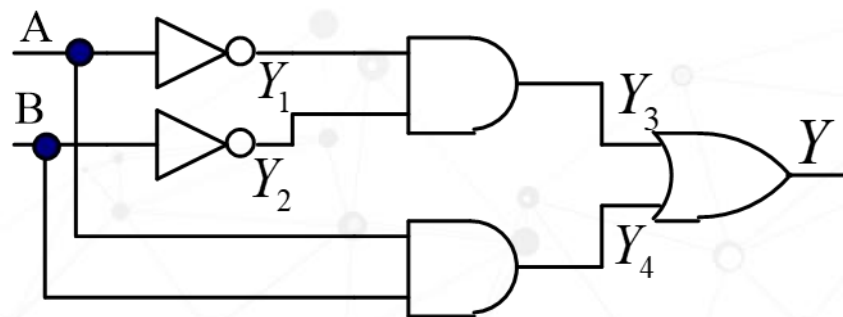
波形图：时序图，表示输出信号根据具体的逻辑关系在输入信号作用下随时间变化的规律。

组合逻辑电路手工分析

2. 组合逻辑电路的手工分析（步骤）

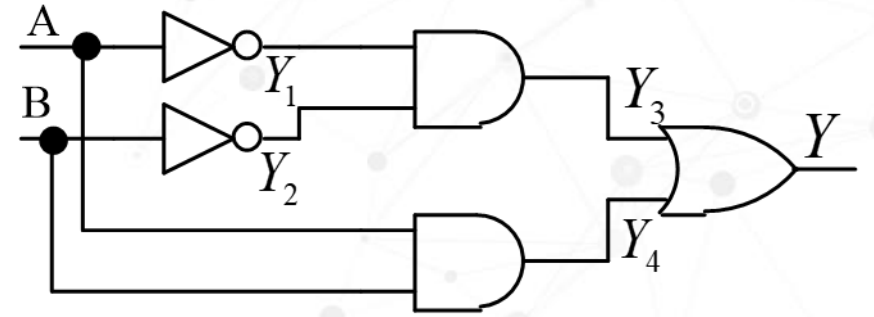


【例】 已知逻辑电路如图所示，分析该电路逻辑功能。



组合逻辑电路手工分析

3. 组合逻辑电路分析



解：（1）根据给定的逻辑电路，逐级写出各输出的逻辑函数表达式：

$$Y_1 = \bar{A} \quad Y_2 = \bar{B} \quad Y_3 = Y_1 \cdot Y_2 = \bar{A} \cdot \bar{B}$$

$$Y_4 = A \cdot B \quad Y = Y_3 + Y_4$$

（2）化简逻辑电路的输出函数表达式：

$$Y = Y_3 + Y_4 = \bar{A} \cdot \bar{B} + A \cdot B = \underline{A \odot B}$$

（3）列出真值表：

（4）逻辑功能：**同或**

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

组合逻辑电路手工分析

3. 组合逻辑电路分析

【例】已知逻辑电路如图所示，分析该电路的逻辑功能。

解：（1）列出逻辑函数表达式，并化简。

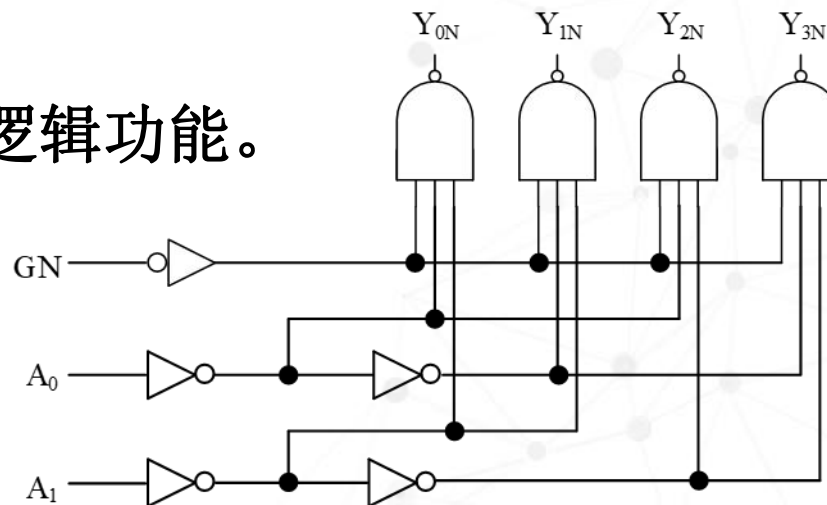
$$Y_{0N} = \overline{\overline{A_1} \overline{A_0}} \quad Y_{2N} = \overline{A_1 \overline{A_0}}$$

$$Y_{1N} = \overline{\overline{A_1} A_0} \quad Y_{3N} = \overline{A_1 A_0}$$

（2）列出真值表：

（3）逻辑功能：

2-4译码器



输入			输出			
GN	A ₁	A ₀	Y _{3N}	Y _{2N}	Y _{1N}	Y _{0N}
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

组合逻辑电路手工分析

3. 组合逻辑电路分析

【例】 已知逻辑电路如图所示，分析该电路逻辑功能。

解：（1）逻辑函数表达式，并化简。

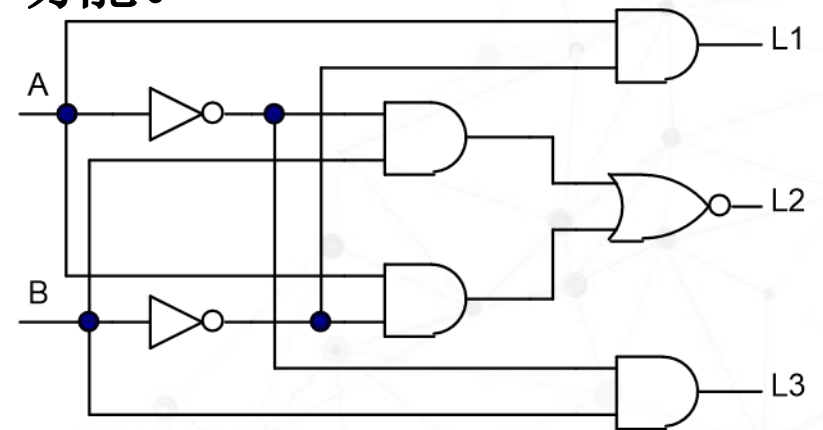
$$L_1 = A \cdot \bar{B} \qquad L_3 = \bar{A} \cdot B$$

$$\begin{aligned} L_2 &= \overline{AB + A\bar{B}} = \overline{AB} \cdot \overline{A\bar{B}} \\ &= (A + \bar{B})(\bar{A} + B) = AB + \bar{A}\bar{B} \end{aligned}$$

（2）列出真值表：

（3）逻辑功能：

一位二进制数比较器



A	B	L ₁	L ₂	L ₃
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

- 4.1 组合逻辑电路手工分析

- **4.2 组合逻辑电路手工设计方法**

- 4.3 编码器

- 4.4 译码器

- 4.5 数据选择器与数据分配器

-

组合逻辑电路手工设计方法

1. 组合逻辑电路手工设计的一般步骤

(1) 对实际逻辑问题进行**逻辑抽象**，确定输入、输出变量；分别对输入、输出变量的具体含义进行定义，然后根据输出与输入之间的逻辑关系列出真值表。

(2) 根据真值表写出相应的**逻辑函数表达式**；

(3) 将逻辑函数表达式**化简**，并转换成所需要的形式；(4) 根据最简逻辑函数表达式画出**逻辑电路图**。

组合逻辑电路手工设计方法

2. 组合逻辑电路的设计示例

【例】用与非门或者或非门设计一个表决电路。即设计一个A、B和C共三人的表决电路。当表决某个提案时，多数人同意，则提案通过；同时A具有否决权。若全票否决，也给出显示。

(1) 逻辑抽象、赋值，并建立真值表

(2) 列逻辑表达式

$$X = \sum m(5,6,7)$$

$$Y = m_0 = \overline{ABC} = \overline{A + B + C}$$

A	B	C	X	Y

组合逻辑电路手工设计方法

2. 组合逻辑电路的设计示例

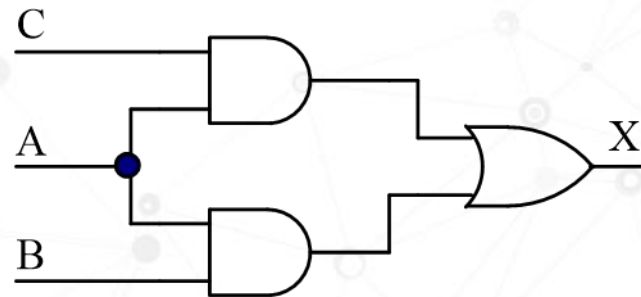
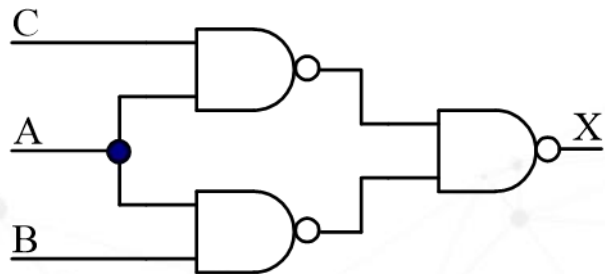
对X用卡诺图化简：

$$X = AB + AC$$

AB\C	00	01	11	10
0	0	0	1	0
1	0	0	1	1

(3) 变换表达式： $X = AB + AC = \overline{\overline{AB} + \overline{AC}} = \overline{\overline{AB} \cdot \overline{AC}}$

(4) 画出逻辑电路图（与非门），Y可用一个3输入或非门实现



用2输入与非门实现Y，需用几个与非门？

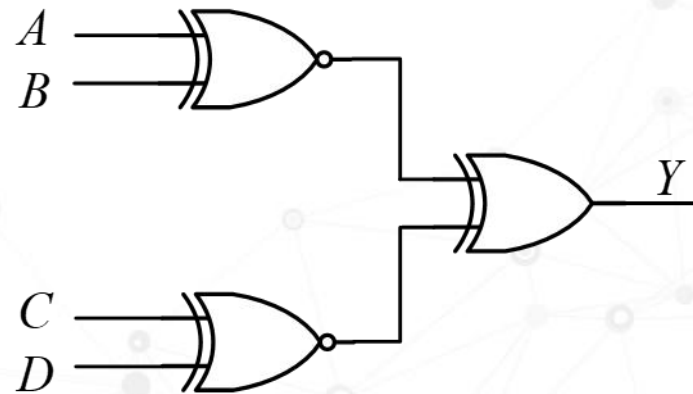
组合逻辑电路手工设计方法

2. 组合逻辑电路的设计示例

【例】某大厅有一盏灯和分布在不同位置的四个开关（A、B、C、D）。试利用四选一数据选择器为大厅设计一个电灯开关控制逻辑电路，使得人们可以在大厅的任何一个位置控制灯的亮或灭。

- (1) 逻辑抽象、列真值表
- (2) 列逻辑表达式

$$Y = \overline{A \oplus B \oplus C \oplus D}$$



输入				输出
A	B	C	D	Y
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	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	
1	1	1	1	

组合逻辑电路手工设计方法

2. 组合逻辑电路的设计示例

(1) 逻辑抽象

(2) 列真值表，写出逻辑表达式

$$Z = \overline{R}\overline{A}\overline{G} + \overline{R}AG + R\overline{A}\overline{G} + RAG + RAG$$

输入变量			输出
R	A	G	Z
0	0	0	1
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

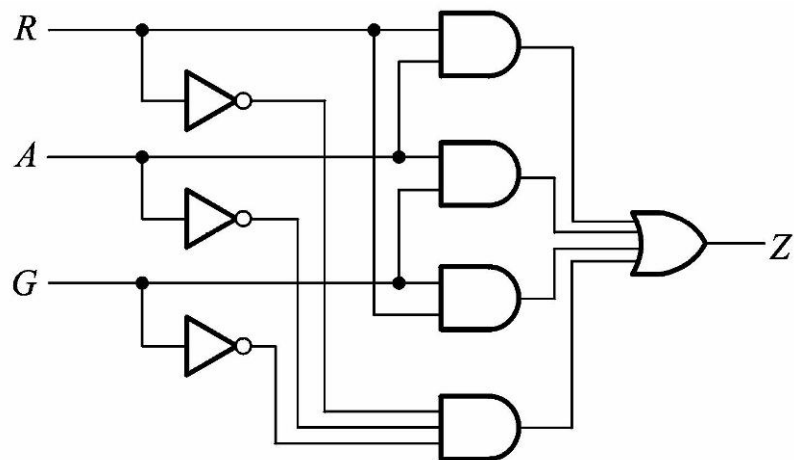
组合逻辑电路手工设计方法

2. 组合逻辑电路的设计示例

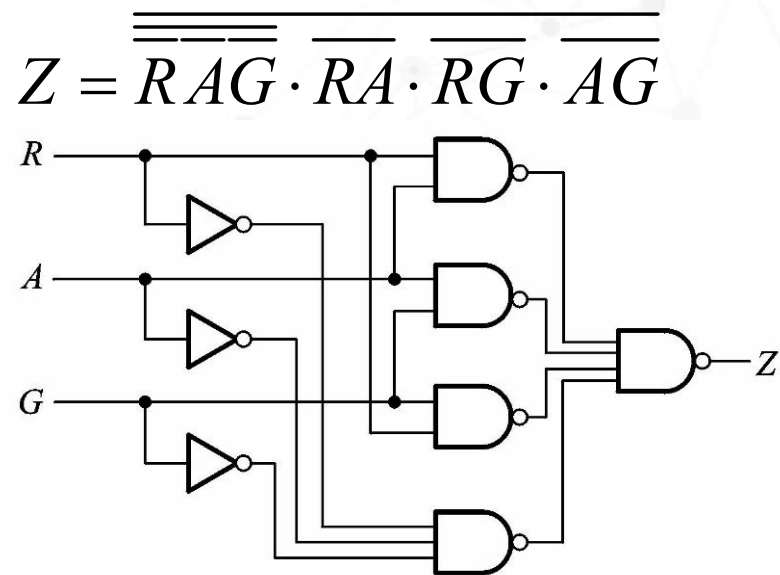
(3) 化简

$$Z = \overline{RAG} + RA + RG + AG$$

(4) 画出逻辑图



AG	00	01	11	10
R				
0	1	0	1	0
1	0	1	1	1



- 4.1 组合逻辑电路手工分析
- 4.2 组合逻辑电路手工设计方法

• **4.3 编码器**

- 4.4 译码器
- 4.5 数据选择器与数据分配器
- 4.6 加法器
-

编码器

1. 编码器的基本概念

生活中常用十进制数及文字、符号等表示事物。

编码器

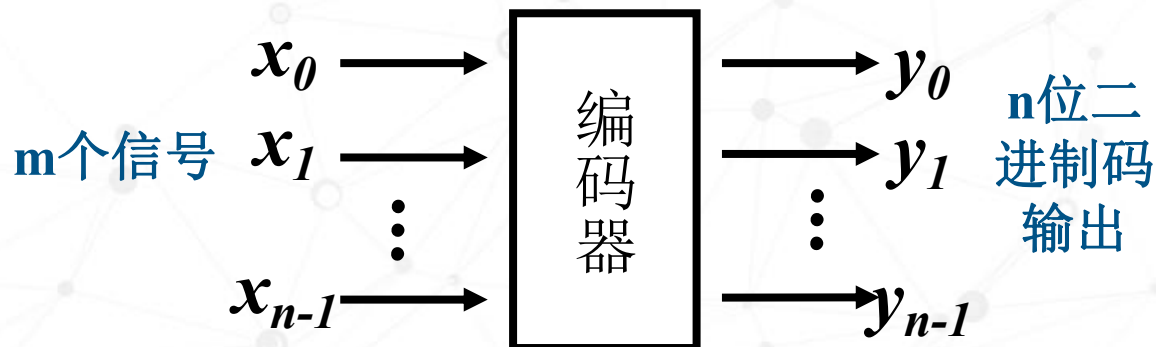


译码器



数字电路只能以二进制信号工作。

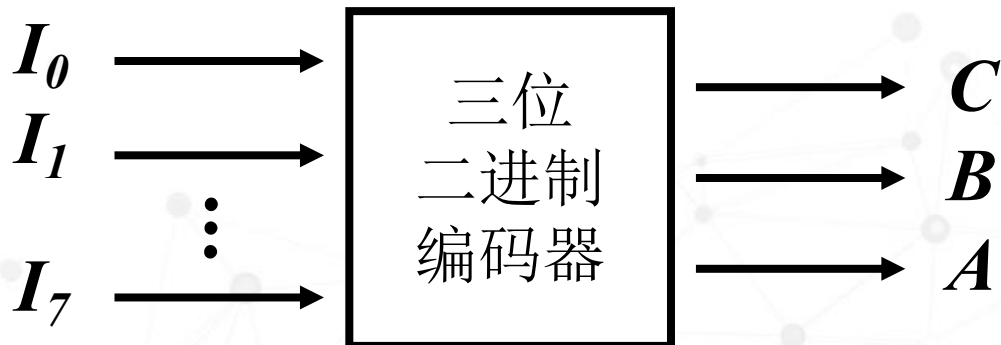
将数字、文字、符号或特定含义的信息表示成二进制代码的过程称为**编码**。
实现编码的逻辑电路，称为**编码器**。



编码器

2. 二进制普通编码器

任何时刻只允许输入一个有效编码请求信号，否则输出将发生混乱。



输入：八个信号（对象）
 $I_0 \sim I_7$ （二值量）

输出：三位二进制代码
 CBA

8-3线普通编码器

编码器

3位二进制编码器真值表

2. 二进制普通编码器

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	C	B	A
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	0	1
0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	1

表达式、电路图?

任何时刻只允许输入一个编码请求

其它输入取值组合不允许出现，为无关项

设输入信号为1表示对该输入进行编码。

编码器

$$C = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 I_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$
$$B = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 I_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$
$$A = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7$$

2. 二进制普通编码器

$$C = I_4 + I_5 + I_6 + I_7$$
$$= \overline{\overline{I_4 + I_5 + I_6 + I_7}}$$
$$= \overline{\bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7}$$

$$B = I_2 + I_3 + I_6 + I_7$$

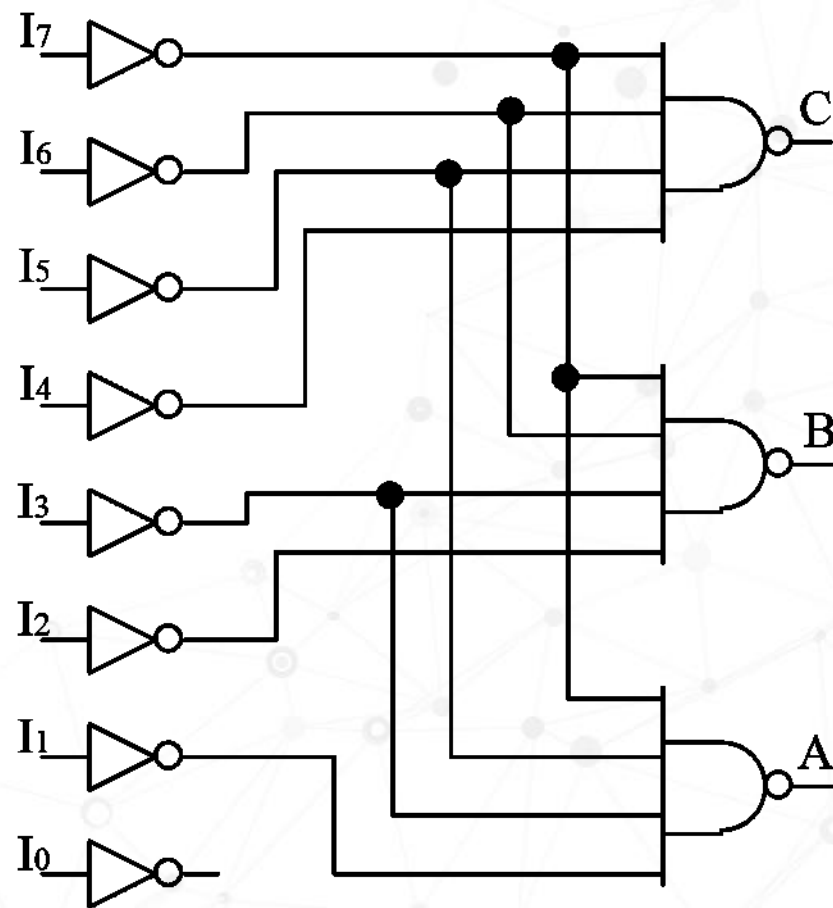
$$A = I_1 + I_3 + I_5 + I_7$$

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	C	B	A
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

编码器

2. 二进制普通编码器

$$\begin{aligned} C &= I_4 + I_5 + I_6 + I_7 & B &= I_2 + I_3 + I_6 + I_7 \\ &= \overline{\overline{I_4 + I_5 + I_6 + I_7}} & A &= I_1 + I_3 + I_5 + I_7 \\ &= \overline{\overline{I_4} \overline{I_5} \overline{I_6} \overline{I_7}} \end{aligned}$$



编码器

3. 二进制优先编码器

非工作状态

工作，且有
编码输入

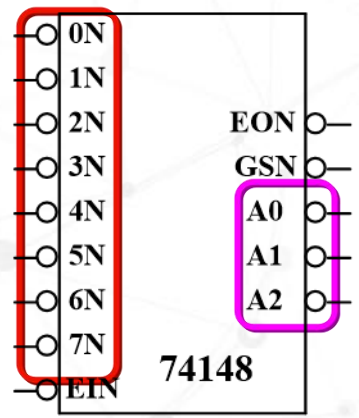
工作，但无有效
编码信号输入

输入								输出				
EIN	7N	6N	5N	4N	3N	2N	1N	GSN	EON	A2	A1	A0
1	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	0	1	1	1
0	0	×	×	×	×	×	×	0	1	0	0	0
0	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	0	×	×	×	×	0	1	0	1	0
0	1	1	1	0	×	×	×	0	1	0	1	1
0	1	1	1	1	0	×	×	0	1	1	0	0
0	1	1	1	1	1	0	×	0	1	1	0	1
0	1	1	1	1	1	0	0	0	1	1	1	0
0	1	1	1	1	1	0	0	0	1	1	1	1

1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

反码

特点: 允许同时输入两个以上的编码信号，但只对其中优先级最高的一个进行编码

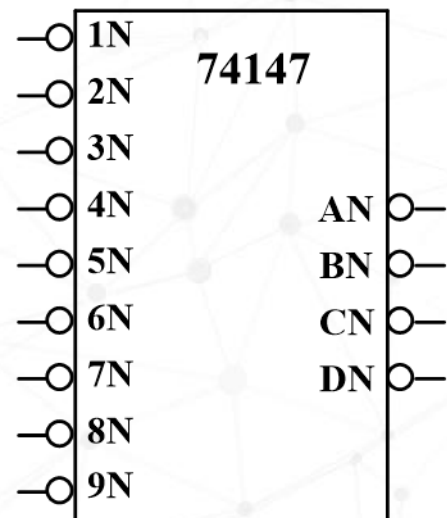


编码器

4. 二-十进制优先编码器

BCD优先编码器、10-4线优先编码器

输 入									输 出			
9N	8N	7N	6N	5N	4N	3N	2N	1N	DN	CN	BN	AN
1	1	1	1	1	1	1	1	1	1	1	1	1
0	×	×	×	×	×	×	×	×	0	1	1	0
1	0	×	×	×	×	×	×	×	0	1	1	1
1	1	0	×	×	×	×	×	×	1	0	0	0
1	1	1	0	×	×	×	×	×	1	0	0	1
1	1	1	1	0	×	×	×	×	1	0	1	0
1	1	1	1	1	0	×	×	×	1	0	1	1
1	1	1	1	1	1	0	×	×	1	1	0	0
1	1	1	1	1	1	1	0	×	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1	1	0



-
- 4.2 组合逻辑电路手工设计方法
- 4.3 编码器

• 4.4 译码器

- 4.5 数据选择器与数据分配器
- 4.6 加法器
- 4.7 比较器
-

译码器

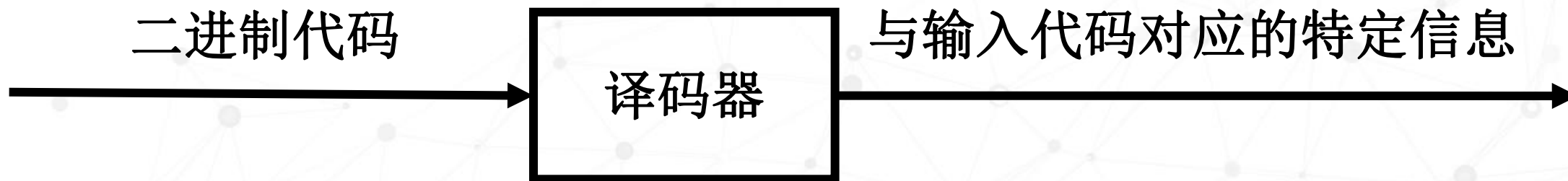
1. 译码器的概念

译码是编码的逆过程

将表示特定意义信息的二进制代码翻译出来

译码器(Decoder)

实现译码功能的电路

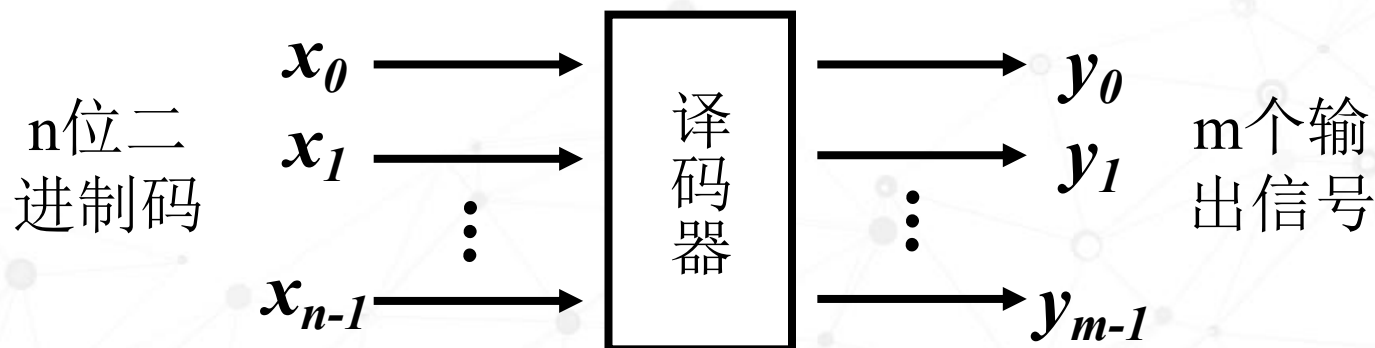


译码器

1. 译码器的概念

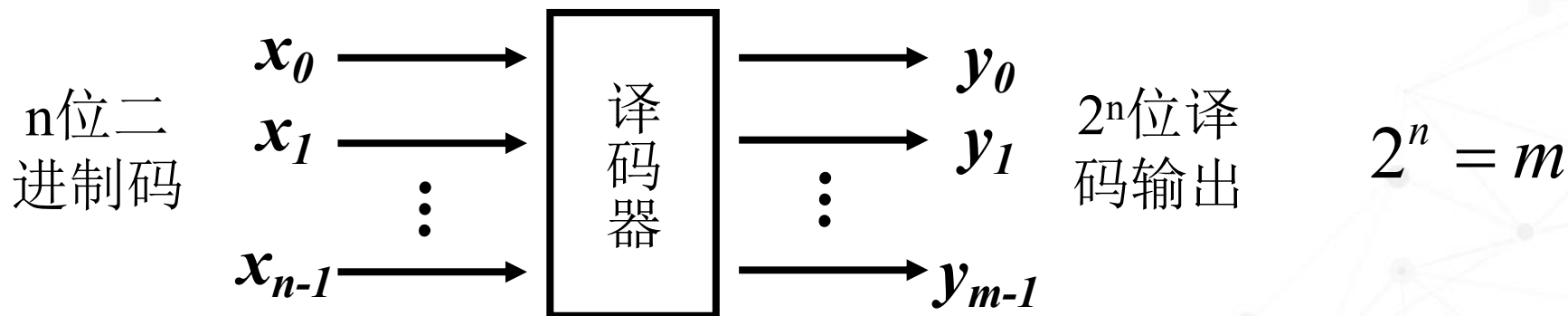
把代码状态的特定含义翻译出来的过程称为译码，实现译码操作的电路称为译码器。

译码器就是把一种代码转换成另一种代码的电路。



译码器

2. 二进制译码器



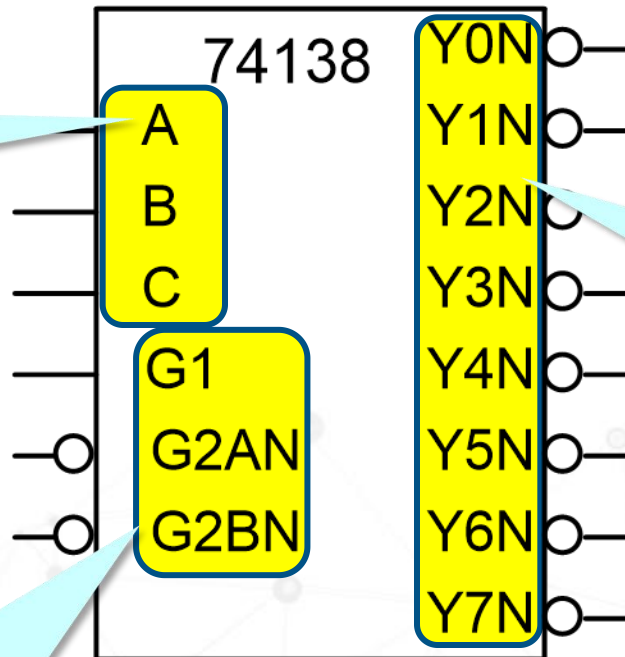
二进制译码器的输入端为 n 个，则输出端为 2^n 个，对应于输入代码的每一种状态， 2^n 个输出中只有一个为1（或为0），其余全为0（或为1）。

译码器

2. 二进制译码器

3-8线译码器74LS138

3位二进制码输入端，C为最高位。



8个译码输出端低电平有效。

使能端 G1 高电平有效，G2AN、G2BN 低电平有效，即当 $G1 = 1, G2AN + G2BN = 0$ 时译码，否则禁止译码。

译码器

2. 二进制译码器

74LS138真值表

G_1	$G_{2AN} + G_{2BN}$	C	B	A	Y_{0N}	Y_{1N}	Y_{2N}	Y_{3N}	Y_{4N}	Y_{5N}	Y_{6N}	Y_{7N}
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

$$Y_{0N} = \overline{\overline{C}\overline{B}\overline{A}} = \overline{m_0} \quad Y_{1N} = \overline{\overline{C}\overline{B}A} = \overline{m_1} \quad Y_{2N} = \overline{\overline{C}B\overline{A}} = \overline{m_2} \quad Y_{3N} = \overline{\overline{C}BA} = \overline{m_3}$$
$$Y_{4N} = \overline{C\overline{B}\overline{A}} = \overline{m_4} \quad Y_{5N} = \overline{C\overline{B}A} = \overline{m_5} \quad Y_{6N} = \overline{CB\overline{A}} = \overline{m_6} \quad Y_{7N} = \overline{CBA} = \overline{m_7}$$

$$Y_{iN} = \overline{m_i}$$

3. 用译码器实现逻辑函数

基本原理

$$Y_{iN} = \bar{m}_i$$

3位二进制译码器给出**3**变量的全部最小项(或最小项的非);

...

n 位二进制译码器给出 **n** 变量的全部最小项(或最小项的非);

任意函数:

将 **n** 位二进制译码输出的最小项组合起来, 可获得任何形式的输入变量不大于 **n** 的组合函数

$$Y = \sum_{i=0}^{2^n-1} m_i \quad \text{或} \quad Y = \sum_{i=0}^{2^n-1} \bar{m}_i$$

注意地址高低位

3. 用译码器实现逻辑函数

【例】试用译码器74LS138和与非门实现逻辑函数： $F(A, B, C) = AB + BC$

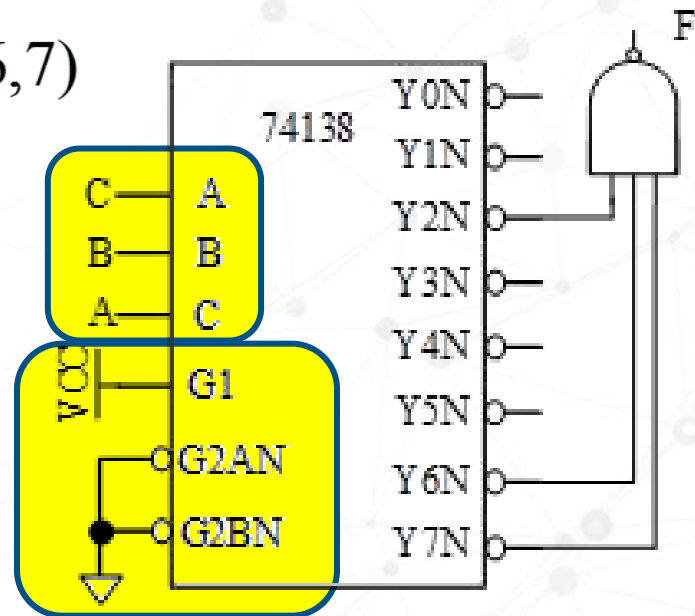
解：(1) 写出逻辑函数 F 的最小项之和形式：

$$\begin{aligned} F(A, B, C) &= AB + BC = AB(C + \bar{C}) + (A + \bar{A})BC \\ &= ABC + AB\bar{C} + \bar{A}BC + A\bar{B}C = \sum m(2, 3, 6, 7) \end{aligned}$$

(2) 转化为与非的形式：

$$\begin{aligned} F(A, B, C) &= m_2 + m_3 + m_6 + m_7 \\ &= \overline{\overline{m_2} \cdot \overline{m_3} \cdot \overline{m_6} \cdot \overline{m_7}} = \overline{Y_2 \cdot Y_3 \cdot Y_6 \cdot Y_7} \end{aligned}$$

(3) 连接电路。



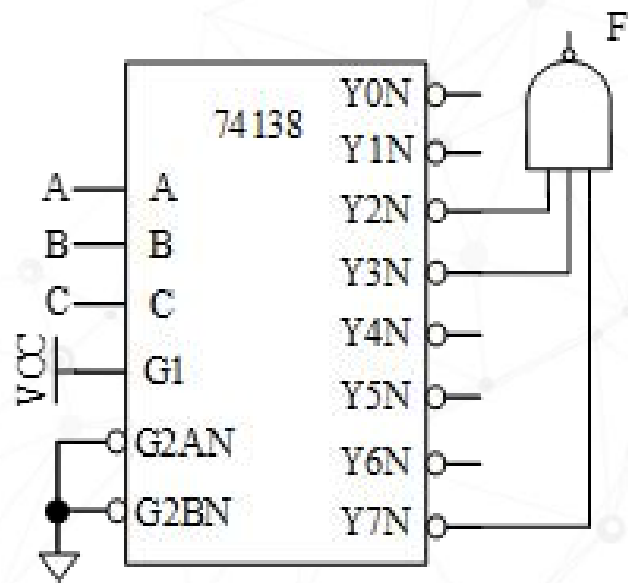
译码器

3. 用译码器实现逻辑函数



【例】试用译码器74LS138和与非门实现逻辑函数： $F(A, B, C) = AB + B\bar{C}$

解：
$$F(C, B, A) = AB + B\bar{C} = CBA + \bar{C}BA + \bar{C}B\bar{A}$$
$$= m_2 + m_3 + m_7 = \overline{\overline{m_2} \cdot \overline{m_3} \cdot \overline{m_7}} = \overline{Y_2 \cdot Y_3 \cdot Y_7}$$





3. 用译码器实现逻辑函数

【例】全减器的真值表如下，其中A为被减数，B为减数，BI为低位向本位的借位，D为本位差，BO为本位向高位的借位。用74LS138和与非门实现对应的逻辑电路。

解：写出输出最小项表达式，再转换成与非-与非式：

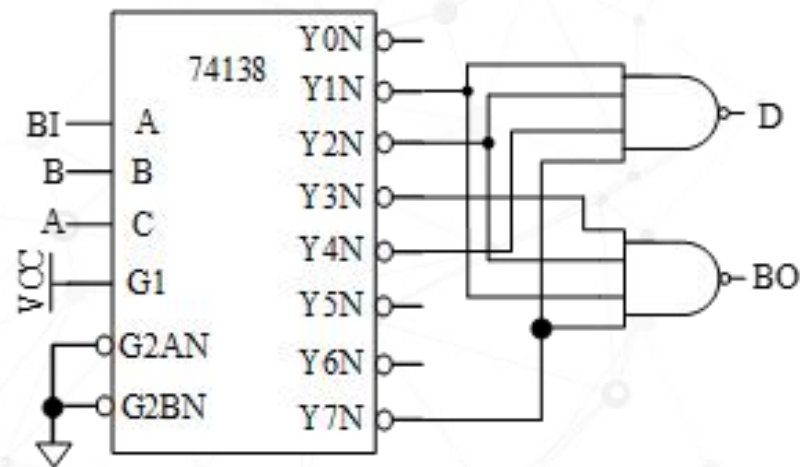
A	B	BI	D	BO
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = m_1 + m_2 + m_4 + m_7$$

$$= \overline{m_1 \cdot m_2 \cdot m_4 \cdot m_7} = \overline{Y_1 \cdot Y_2 \cdot Y_4 \cdot Y_7}$$

$$BO = m_1 + m_2 + m_3 + m_7$$

$$= \overline{m_1 \cdot m_2 \cdot m_3 \cdot m_7} = \overline{Y_1 \cdot Y_2 \cdot Y_3 \cdot Y_7}$$



5. 显示译码器

用来驱动各种显示器件，从而将用二进制代码表示的数字、文字、符号翻译成人们习惯的形式直观地显示出来的电路，称为**显示译码器**。

显示译码器的输出信号用以驱动显示器件，显示出十进制或十六进制数字及某些简单字符。

5. 显示译码器

(1) 七段数码显示器

七段数码显示器

发光二极管 (LED)

液晶显示器 (LCD)

由七段可发光的字段组合而成

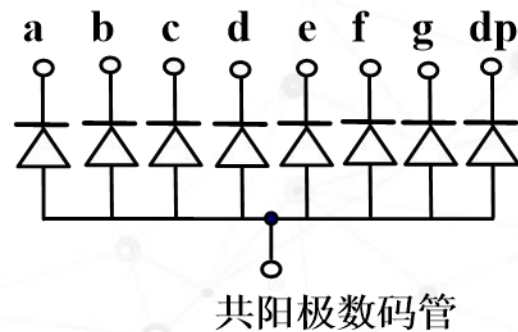
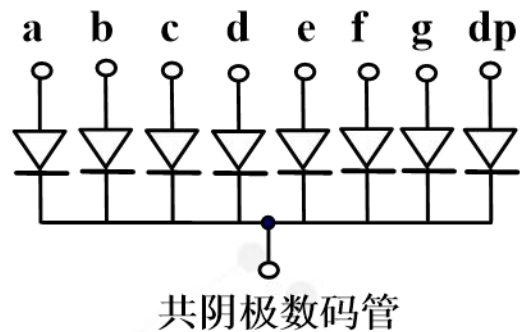
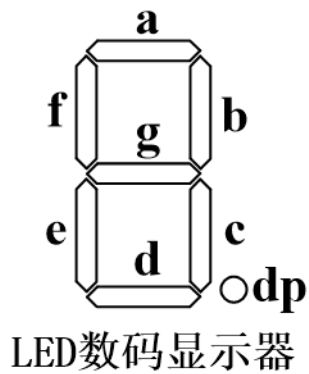
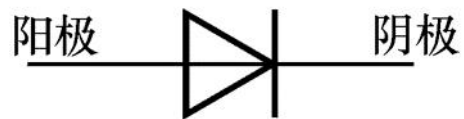
发光二极管的**优点**是工作电压较低，体积小，寿命长，工作可靠性高，响应速度快，亮度高。**缺点**是工作电流大，耗电大。

液晶显示器的主要**优点**是功耗极小，工作电压低。**缺点**是显示不够清晰，响应速度慢。

译码器

5. 显示译码器

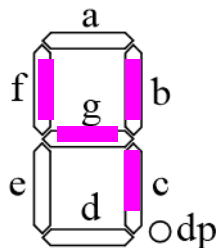
(1) 七段数码显示器



共阳极数码管段选码表

5. 显示译码器

(1) 七段数码显示器



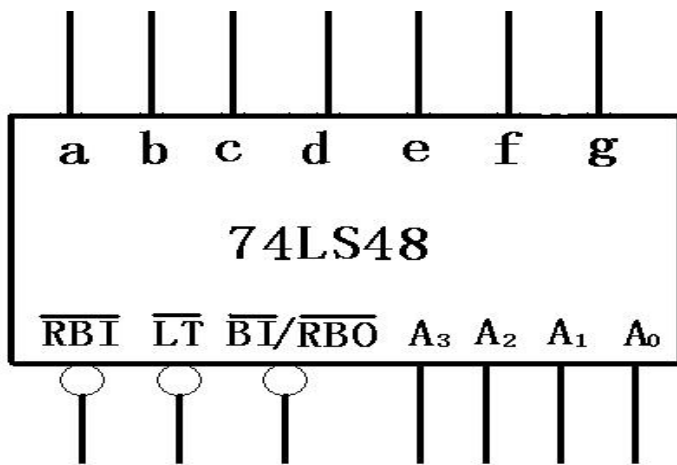
显示字符	dp	g	f	e	d	c	b	a	段选码
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H
A	1	0	0	0	1	0	0	0	88H
B	1	0	0	0	0	0	1	1	83H
C	1	1	0	0	0	1	1	0	C6H
D	1	0	1	0	0	0	0	1	A1H
E	1	0	0	0	0	1	1	0	86H
F	1	0	0	0	1	1	1	0	8EH

译码器

5. 显示译码器

(2) 七段显示译码器

驱动共阴极
数码显示器

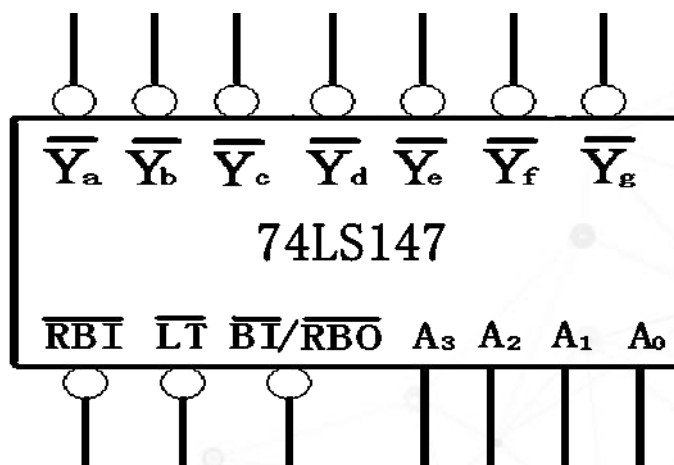


输出状态为：
高电平或低电平

↓
亮

↓
灭

驱动共阳极
数码显示器



输出状态为：
低电平或高阻态

↓
亮

↓
灭

5. 显示译码器

(3) 多数数码管动态扫描显示控制方法

静态显示控制：每一个译码器控制一个数码管。

优点：编程容易，控制容易，管理方便。

缺点：占用硬件电路资源和器件的端口资源较多。

动态扫描显示控制：一个显示译码器控制多个数码管，每个数码管的公共端由相应位选信号控制。

优点：节省驱动电路，降低功耗。

缺点：需通过电流驱动器、功率驱动器提高亮度。

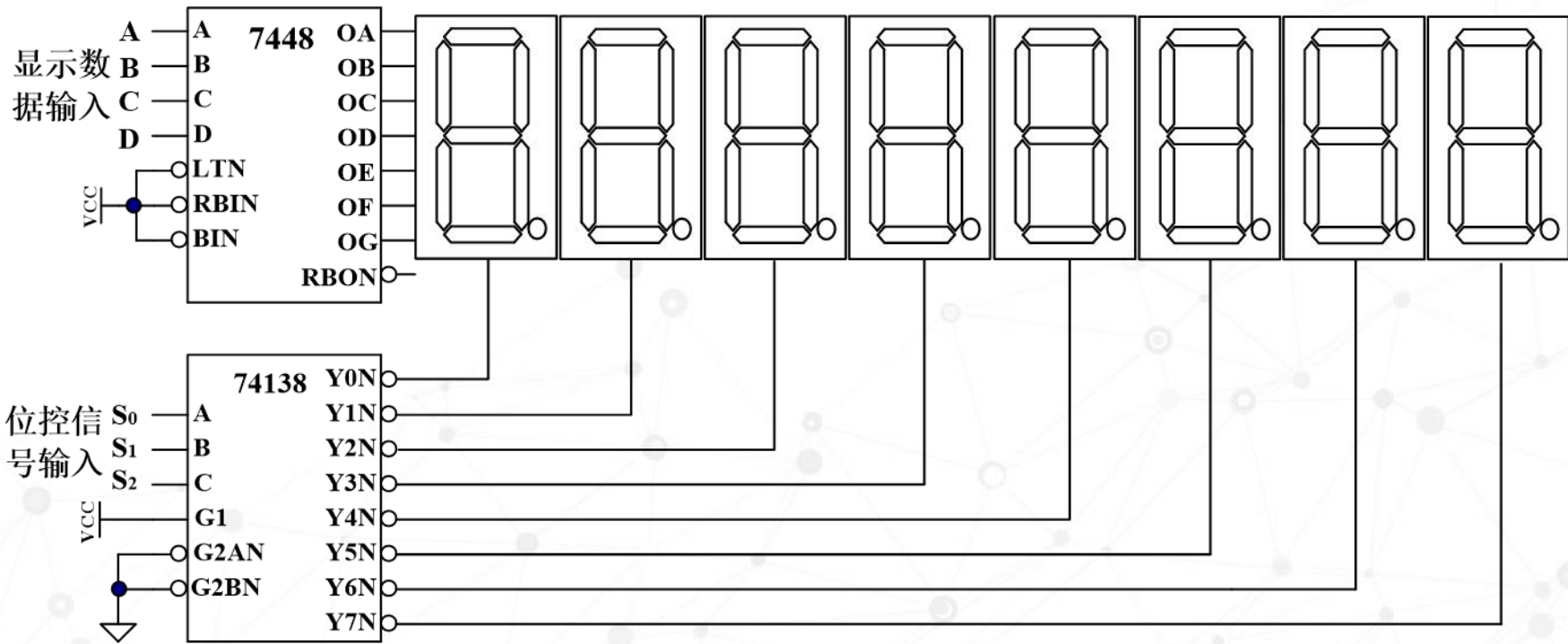
利用人的视觉延迟的特点

译码器

5. 显示译码器

(3) 多数数码管动态扫描显示控制方法

【例】用一片74LS48和74LS138实现八位数码管的八位十进制数显示。



-
- 4.3 编码器
- 4.4 译码器

- **4.5 数据选择器与数据分配器**

- 4.6 加法器
- 4.7 比较器
- 4.8 广义译码器概念
-

数据选择器与数据分配器

1. 数据选择器的基本概念

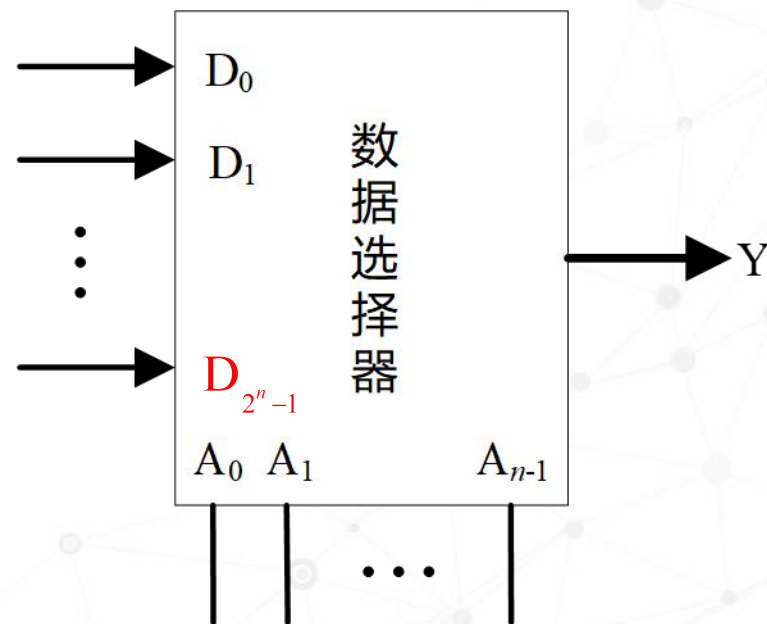
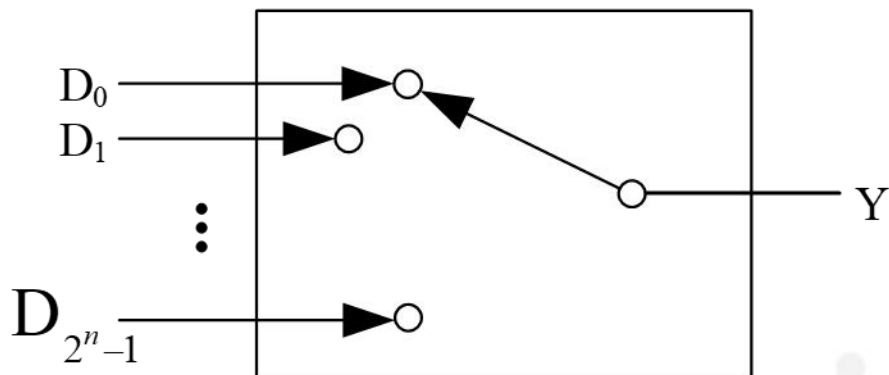
数据选择器：在多路数据传输中，根据地址码的要求，能把其中的一路信号挑选出来的电路就是数据选择器，又称多路选择器或多路开关。

在数据选择器中，要用地址输入信号来完成挑选数据任务。一个4选1的数据选择器，应有两个地址输入端，共有 $2^2=4$ 种组合，每种组合可选择对应的一路输入数据来输出；同理，8选1 数据选择器，应有3个地址输入端；16选1 数据选择器，应有4个地址输入端；依次类推。

数据分配器：在功能上正好与数据选择器相反，根据地址码的不同，将一路数据分配到相应的一个输出端上输出的电路。

数据选择器与数据分配器

1. 数据选择器的基本概念



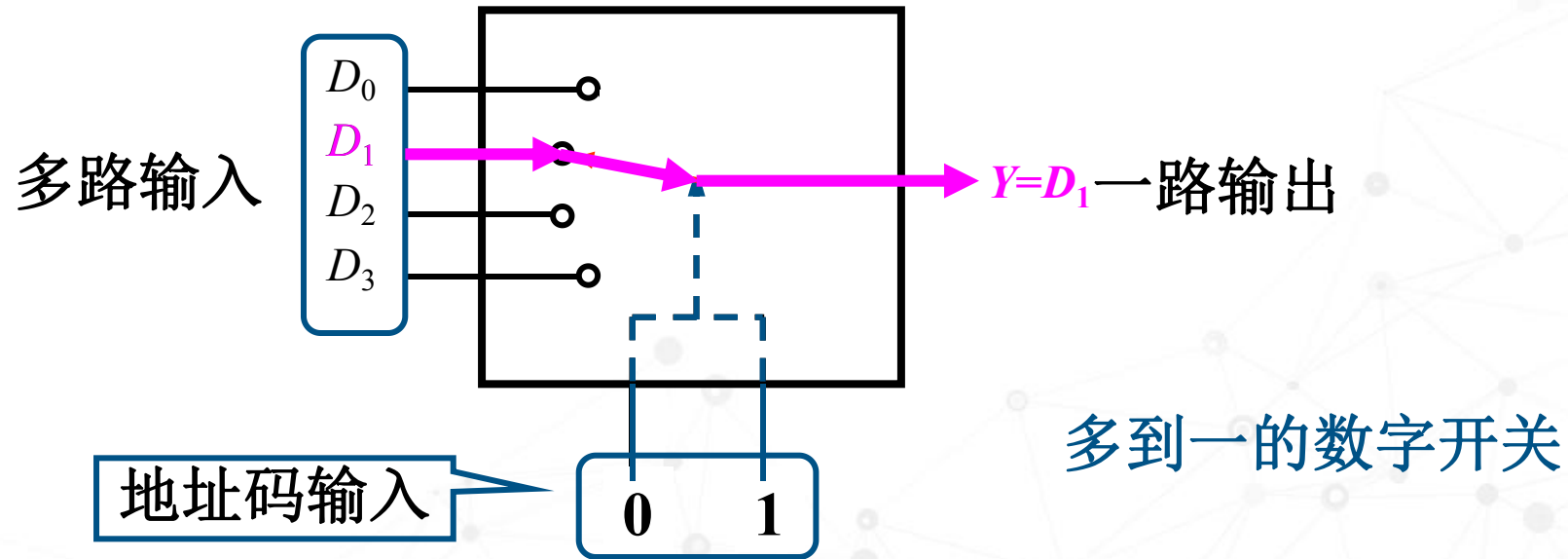
常用 2 选 1、4 选 1、8 选 1 和 16 选 1 等数据选择器

MUX

数据选择器与数据分配器

2. 集成数据选择器

4选1数据选择器工作示意图

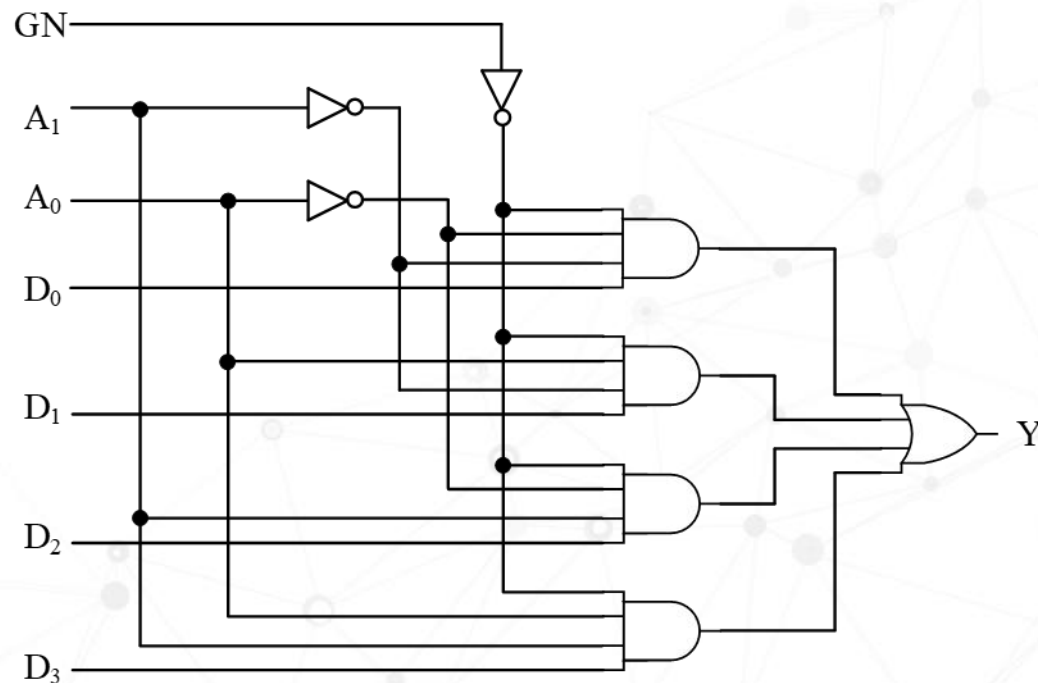
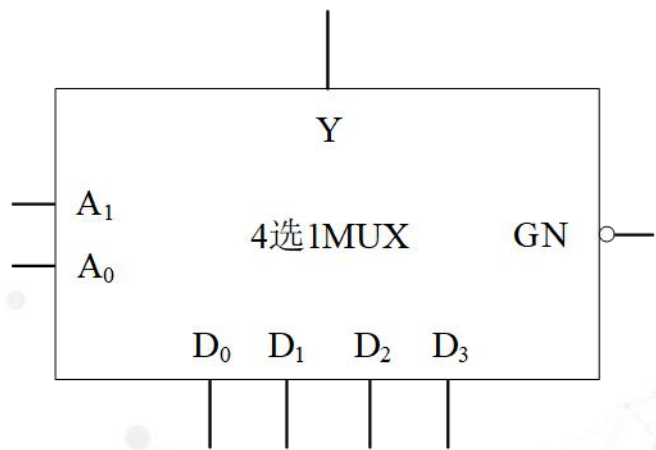


数据选择器的输入信号个数 M 与地址码个数 n 的关系为

$$M = 2^n$$

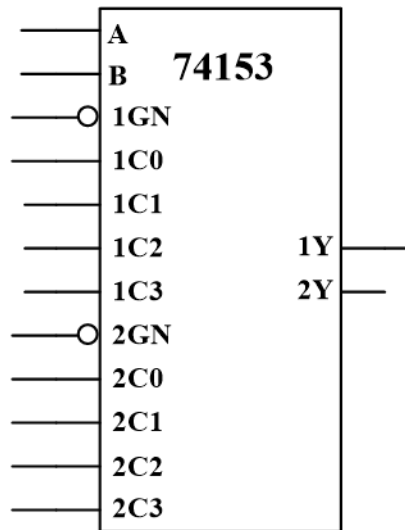
数据选择器与数据分配器

2. 集成数据选择器



数据选择器与数据分配器

2. 集成数据选择器



4选1数据选择器的真值表

输入			输出
A_1	A_0	GN	Y
\times	\times	1	0
0	0	0	D_0
0	1	0	D_1
1	0	0	D_2
1	1	0	D_3

输出Y提供了地址变量的全部最小项

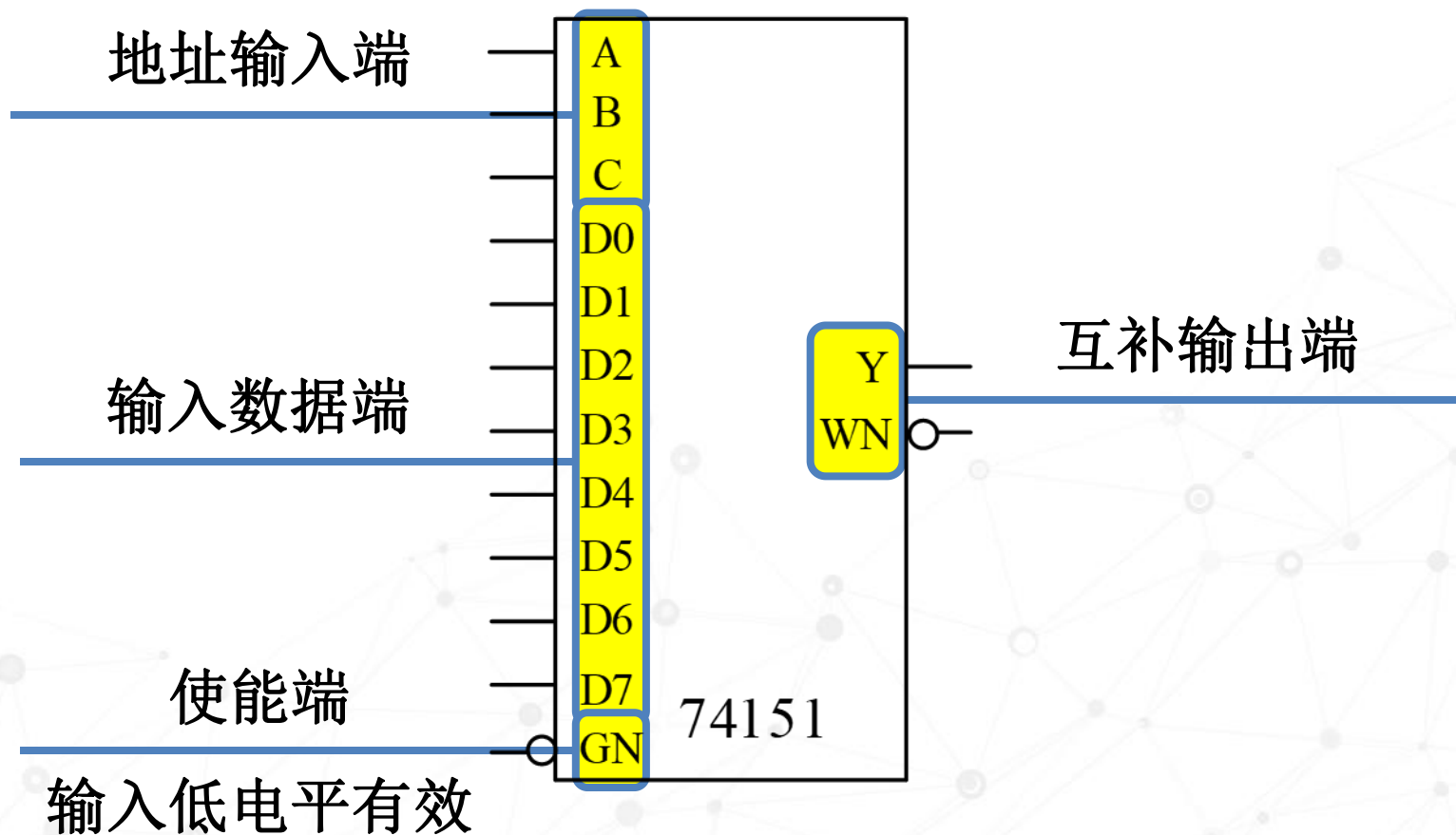
(1) $GN=1$ 时，数据选择器被禁止，无论地址变量取何值，输出Y总是等于0。

(2) $GN=0$ 时，
$$Y = \bar{A}_1\bar{A}_0D_0 + \bar{A}_1A_0D_1 + A_1\bar{A}_0D_2 + A_1A_0D_3 = \sum_{i=0}^3 m_i D_i$$

数据选择器与数据分配器

2. 集成数据选择器

8选1数据选择器



数据选择器与数据分配器

2. 集成数据选择器

(1) GN=1时，数据选择器被禁止，无论地址变量取何值，输出Y总是等于0。

(2) GN=0时，

$$Y = \overline{C}\overline{B}\overline{A}D_0 + \overline{C}\overline{B}AD_1 + \dots + CBAD_7 = \sum_{i=0}^7 m_i D_i$$

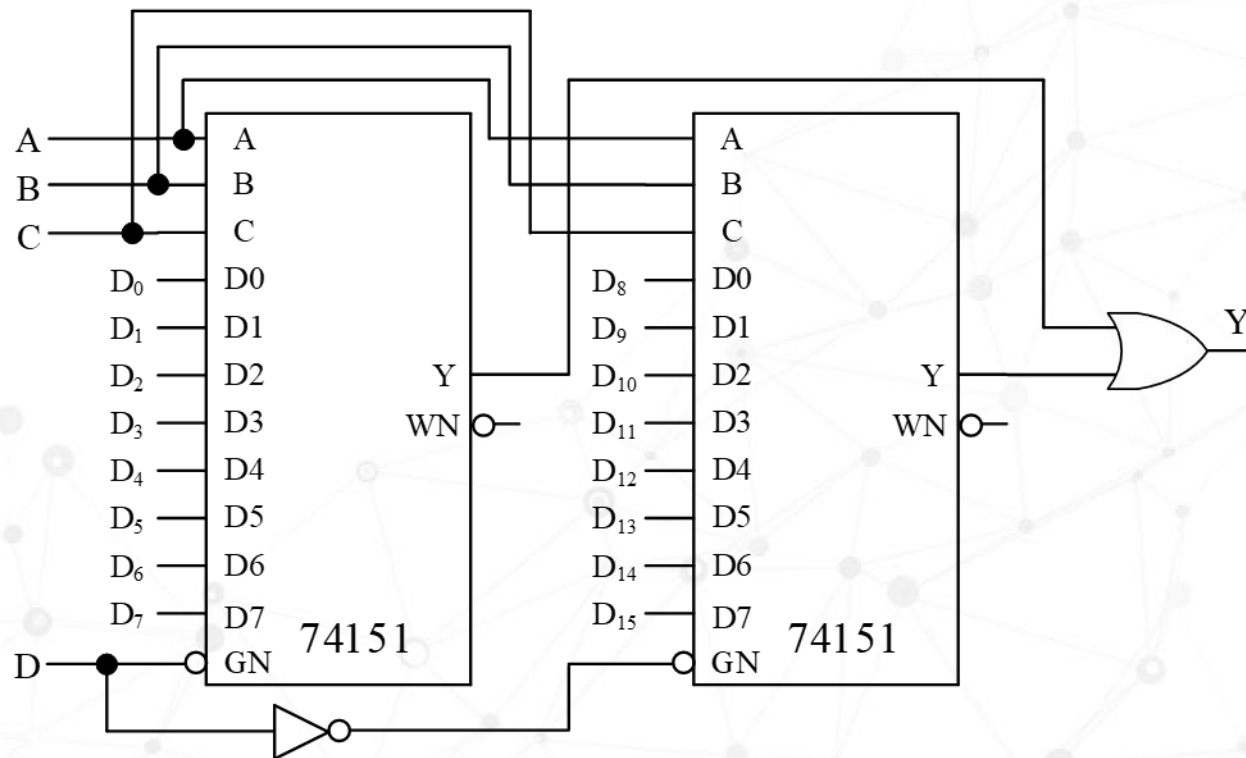
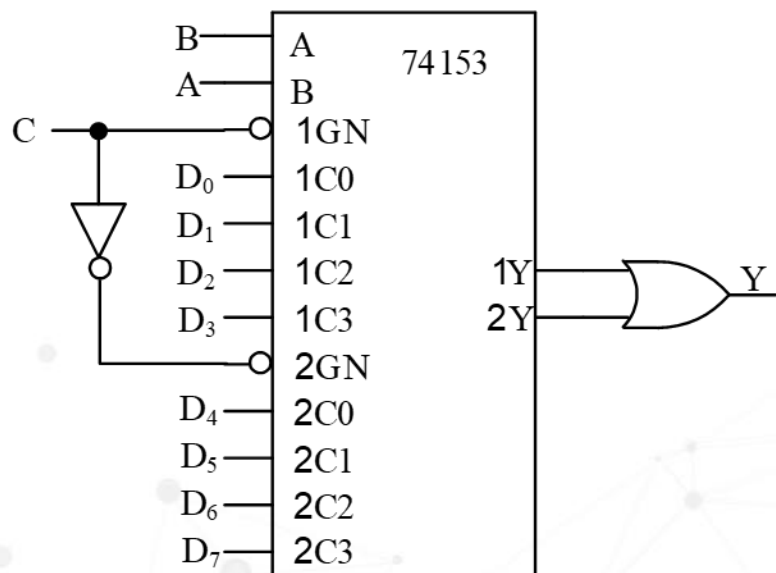
输出Y提供了地址变量的全部最小项

输 入				输 出	
GN	C	B	A	Y	W
1	×	×	×	0	1
0	0	0	0	D_0	\overline{D}_0
0	0	0	1	D_1	\overline{D}_1
0	0	1	0	D_2	\overline{D}_2
0	0	1	1	D_3	\overline{D}_3
0	1	0	0	D_4	\overline{D}_4
0	1	0	1	D_5	\overline{D}_5
0	1	1	0	D_6	\overline{D}_6
0	1	1	1	D_7	\overline{D}_7

数据选择器与数据分配器

2. 集成数据选择器

译码器的扩展



数据选择器与数据分配器

注意地址高低位

3. 用数据选择器实现逻辑函数

【例】用8选1数据选择器74LS151实现逻辑函数： $F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC$

解：(1) 将F变成最小项之和的形式：

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC(B + \bar{B}) = m_0 + m_3 + m_5 + m_7$$

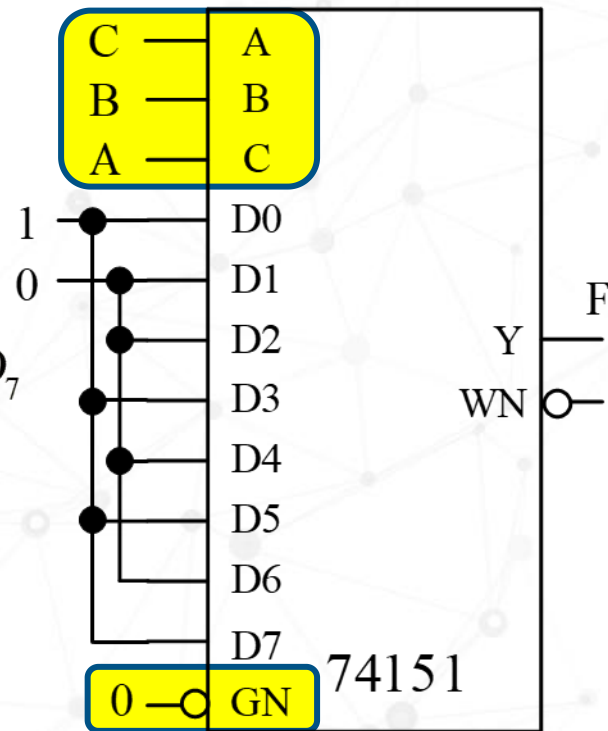
(2) 写出数据选择器的输出表达式

$$Y = \sum_{i=0}^7 m_i D_i = m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7$$

比较F和Y得：

$$D_0 = 1; D_1 = 0; D_2 = 0; D_3 = 1;$$

$$D_4 = 0; D_5 = 1; D_6 = 0; D_7 = 1;$$



数据选择器与数据分配器

3. 用数据选择器实现逻辑函数

【例】用8选1数据选择器74LS151实现逻辑函数： $F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC$

解：(1) 将F变成最小项之和的形式：

$$F = \bar{C}\bar{B}\bar{A} + CB\bar{A} + C(B+\bar{B}) \quad A=m_0+m_5+m_6+m_7$$

(2) 写出数据选择器的输出表达式

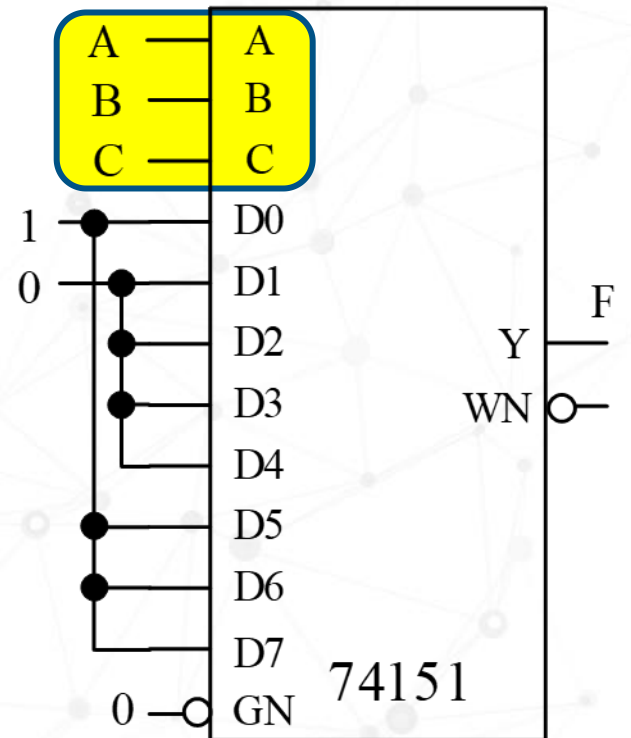
$$Y = \sum_{i=0}^7 m_i D_i = m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7$$

比较F和Y得：

$$D_0 = 1; D_1 = 0; D_2 = 0; D_3 = 0;$$

$$D_4 = 0; D_5 = 1; D_6 = 1; D_7 = 1;$$

注意地址高低位



数据选择器与数据分配器

3. 用数据选择器实现逻辑函数

【例】用1/2个双4选1数据选择器74LS153实现逻辑函数： $F = AB + AC$

解：(1) 将F变成最小项之和的形式：

$$F = AB + AC = AB(C + \bar{C}) + AC(B + \bar{B}) = ABC + ABC\bar{C} + A\bar{B}C$$

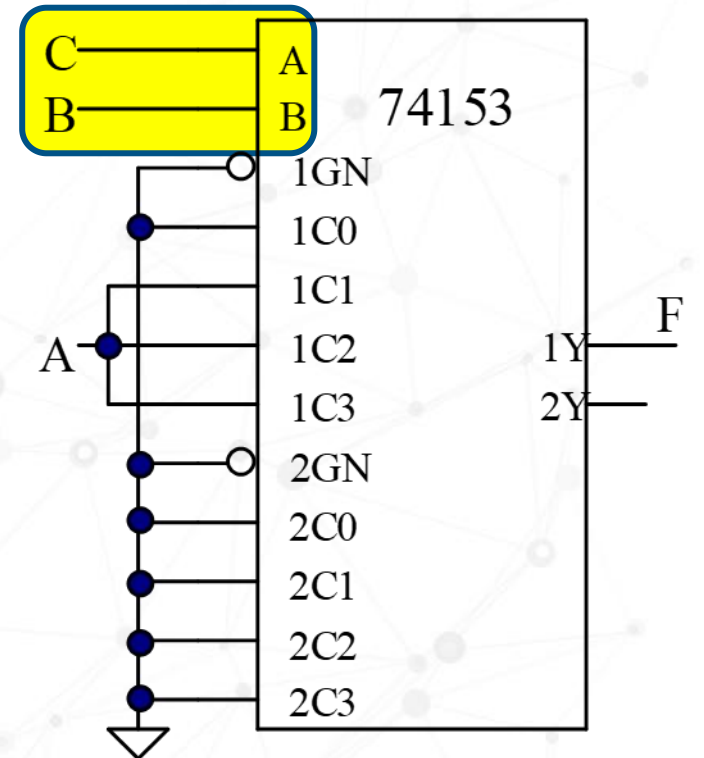
(2) 当B接74LS153的高位地址，C接74LS153的低位地址时，F可改写成：

$$\begin{aligned} F &= 0 \cdot \bar{B}\bar{C} + A \cdot \bar{B}C + A \cdot B\bar{C} + A \cdot BC \\ &= m_0 \cdot 0 + m_1 \cdot A + m_2 \cdot A + m_3 \cdot A \end{aligned}$$

要使得 $1Y=F$ ，则有：

$$1C_0=0; 1C_1=A; 1C_2=A; 1C_3=A;$$

注意地址高低位



数据选择器与数据分配器

注意地址高低位

3. 用数据选择器实现逻辑函数

【例】用74LS151实现逻辑函数： $F = \sum m(0, 2, 5, 6, 10, 12, 13, 14, 15)$

解:(1)F的卡诺图 (2)选定D作为图记变量，F降一维。

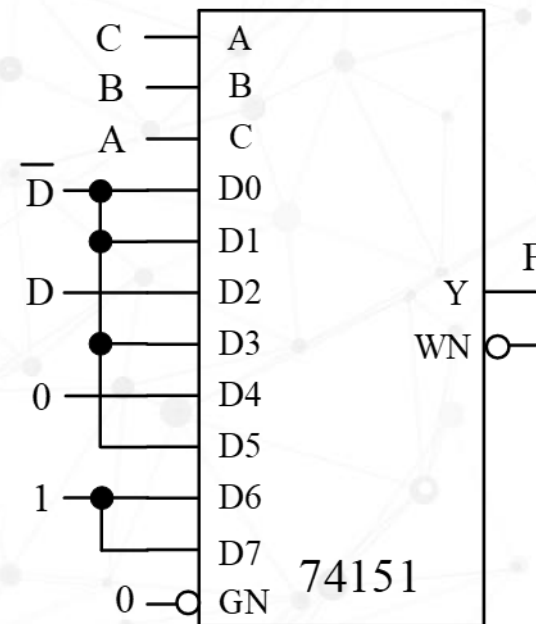
(3)74LS151地址端 $CBA = ABC$.

F	CD		AB			
	00	01	11	10		
00	1	0	0	1		
01	0	1	0	1		
11	1	1	1	1		
10	0	0	0	1		

降维D

F	C		AB	
	0	1		
00	\bar{D}	\bar{D}		
01	D	\bar{D}		
11	1	1		
10	0	\bar{D}		

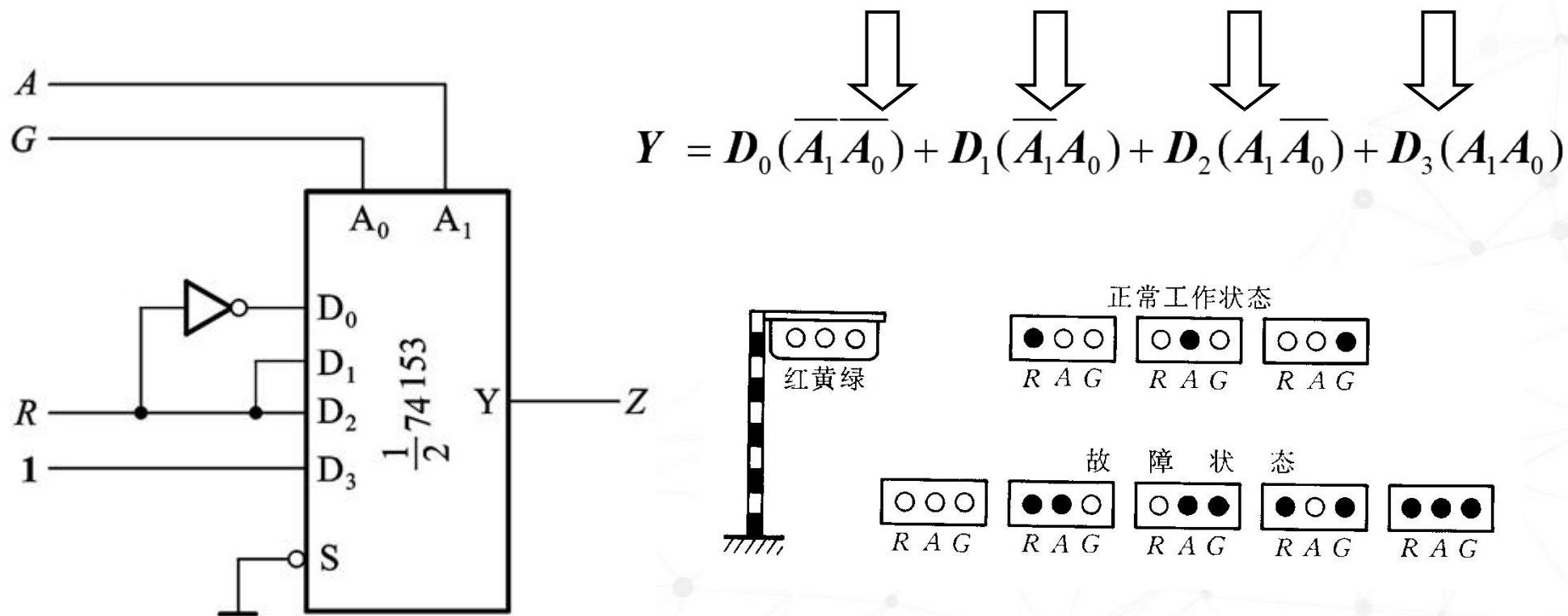
Y	C		AB	
	0	1		
00	D ₀	D ₁		
01	D ₂	D ₃		
11	D ₆	D ₇		
10	D ₄	D ₅		



$$D_0 = D_1 = D_3 = D_5 = \bar{D}; D_2 = D; D_4 = 0; D_6 = D_7 = 1;$$

数据选择器与数据分配器

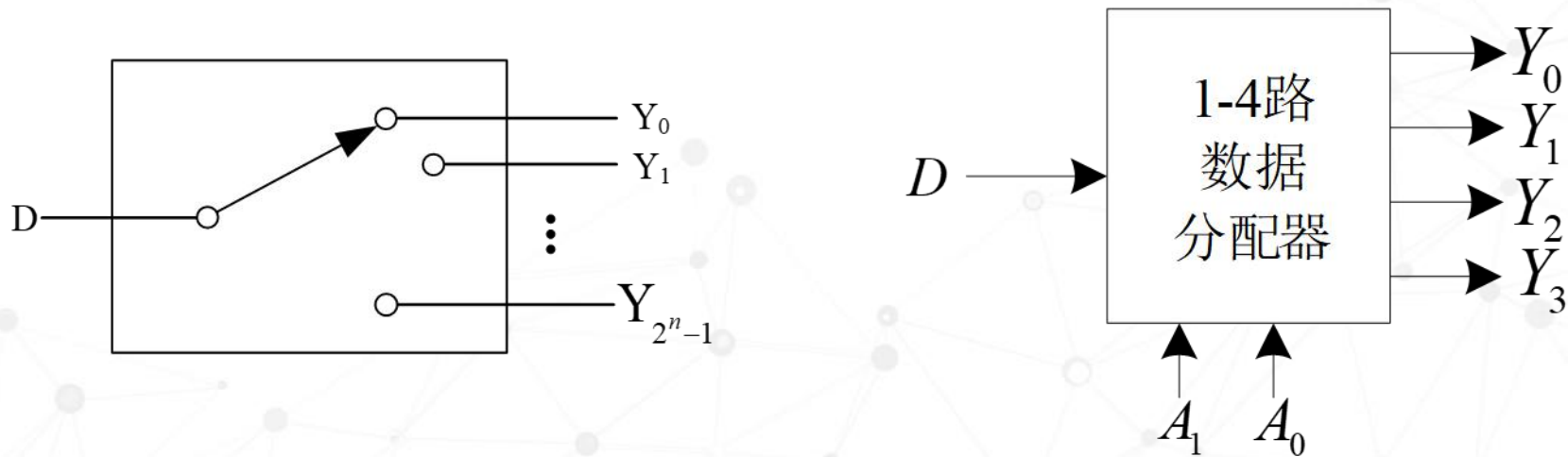
【例】用74LS153实现： $Z = \overline{R}\overline{A}\overline{G} + \overline{R}A\overline{G} + R\overline{A}\overline{G} + RA\overline{G} + RAG$



数据选择器与数据分配器

4. 数据分配器

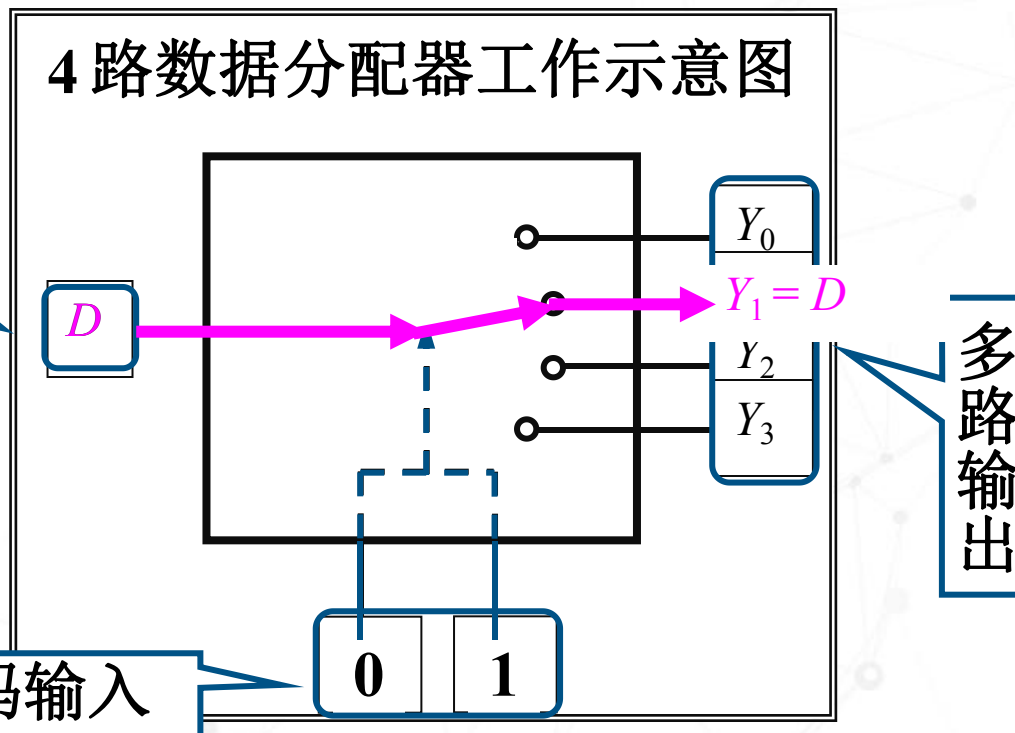
又称多路分配器（DEMUX），功能与数据选择器相反，是将单路输入数据按 n 位地址分配到 2^n 路数据输出端的某一路输出上。



数据选择器与数据分配器

4. 数据分配器

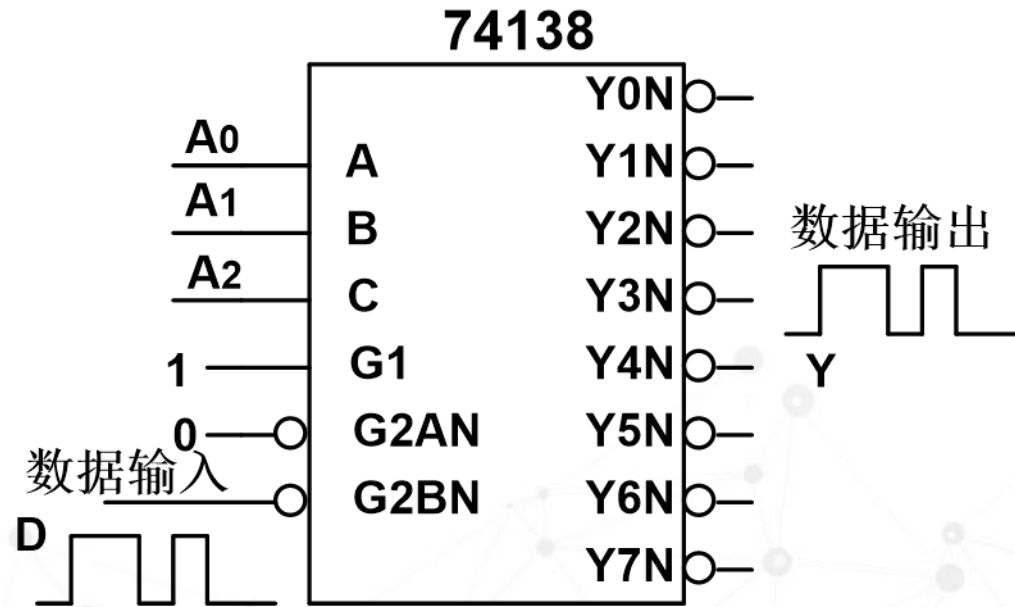
输入			输出			
GN	A ₁	A ₀	Y _{3N}	Y _{2N}	Y _{1N}	Y _{0N}
1	×	×	1	1	1	1
0	0	0	1	1	1	D
0	0	1	1	1	D	1
0	1	0	1	D	1	1
0	1	1	D	1	1	1



数据选择器与数据分配器

4. 数据分配器

用74LS138构成1-8路数据分配器



$$Y_i = G1(\overline{G2AN} + \overline{G2BN}) m_i$$

G2BN=0时，74138工作(输出低有效),则输出CBA选定的通道数据(结果为)；

G2BN=1，74138禁止，所有输出端都输出**1**。即实现把数据**D**分配到输出端。

-
- 4.4 译码器
- 4.5 数据选择器与数据分配器

• 4.6 加法器

- 4.7 比较器
- 4.8 广义译码器概念
- 4.9 可编程逻辑器件的结构与原理
-

加法器

1. 半加器

将输入的两个1位二进制数A和B相加，产生和数S及进位数CO。

输入		输出	
A	B	CO	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

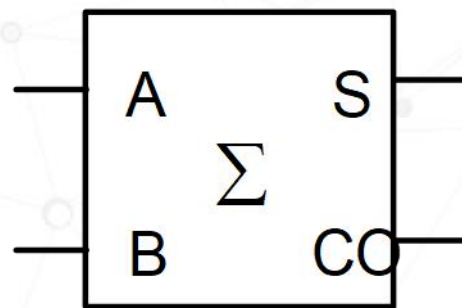
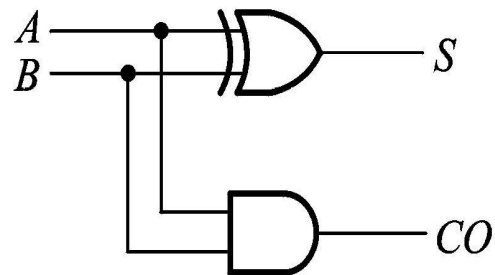
加数

向高位的进位

本位的和

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

$$CO = AB$$



加法器

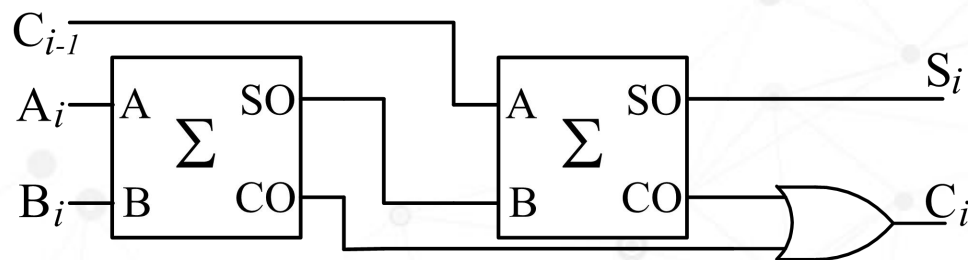
2. 全加器 将两个1位二进制数及来自低位的进位相加

输入			输出	
A_i	B_i	C_{i-1}	C_i	S_i
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



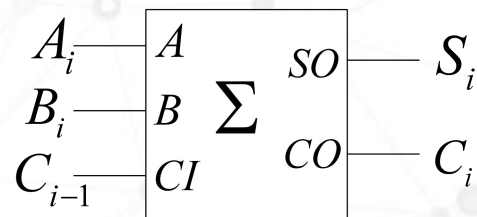
$$S_i = \sum m(1,2,4,7)$$

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



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

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



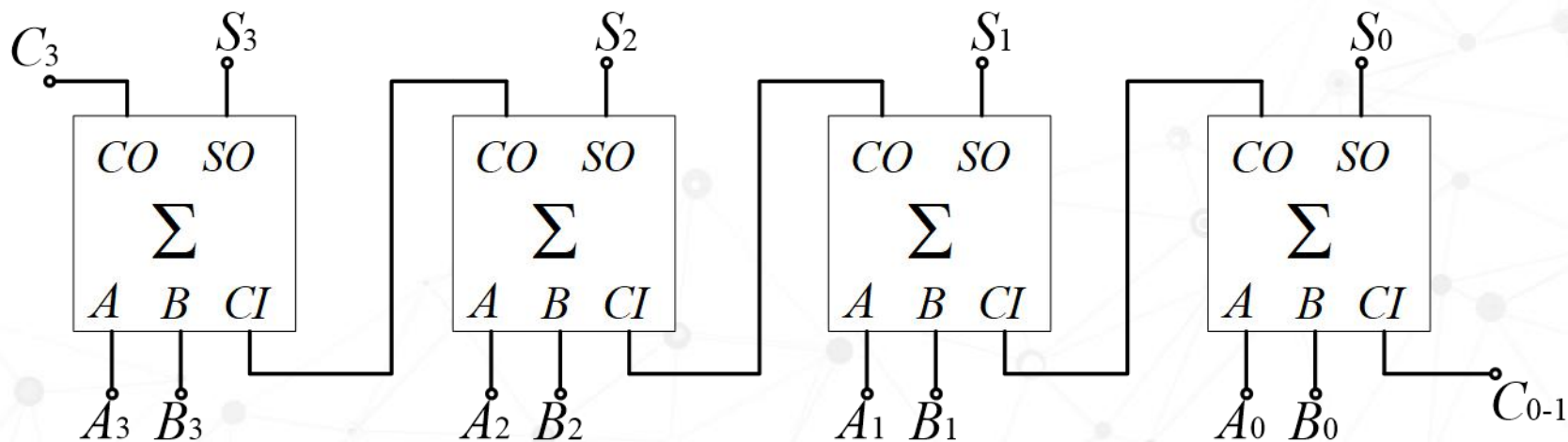
加法器

3. 多位加法器

(1) 串行进位加法器

优点：简单
缺点：慢

构成：把n位全加器串联起来，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。



加法器

3. 多位加法器

(2) 超前进位加法器

提高工作速度的途径：设法减小进位信号的传递时间

进位表达式：

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

$$G_i = A_i B_i \quad \text{--进位产生项}$$

$$P_i = A_i + B_i \quad \text{--进位传输项}$$

$$C_i = G_i + P_i C_{i-1}$$

$$= G_i + P_i [G_{i-1} + P_{i-1} C_{i-2}]$$

$$= G_i + P_i G_{i-1} + P_i P_{i-1} [G_{i-2} + P_{i-2} C_{i-3}]$$

⋮

$$= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \cdots + P_i P_{i-1} \cdots P_1 G_0 + P_i P_{i-1} \cdots P_0 C_{-1}$$

和位表达式：

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

优点：快，每1位的和及最后的进位基本同时产生。

缺点：电路复杂。

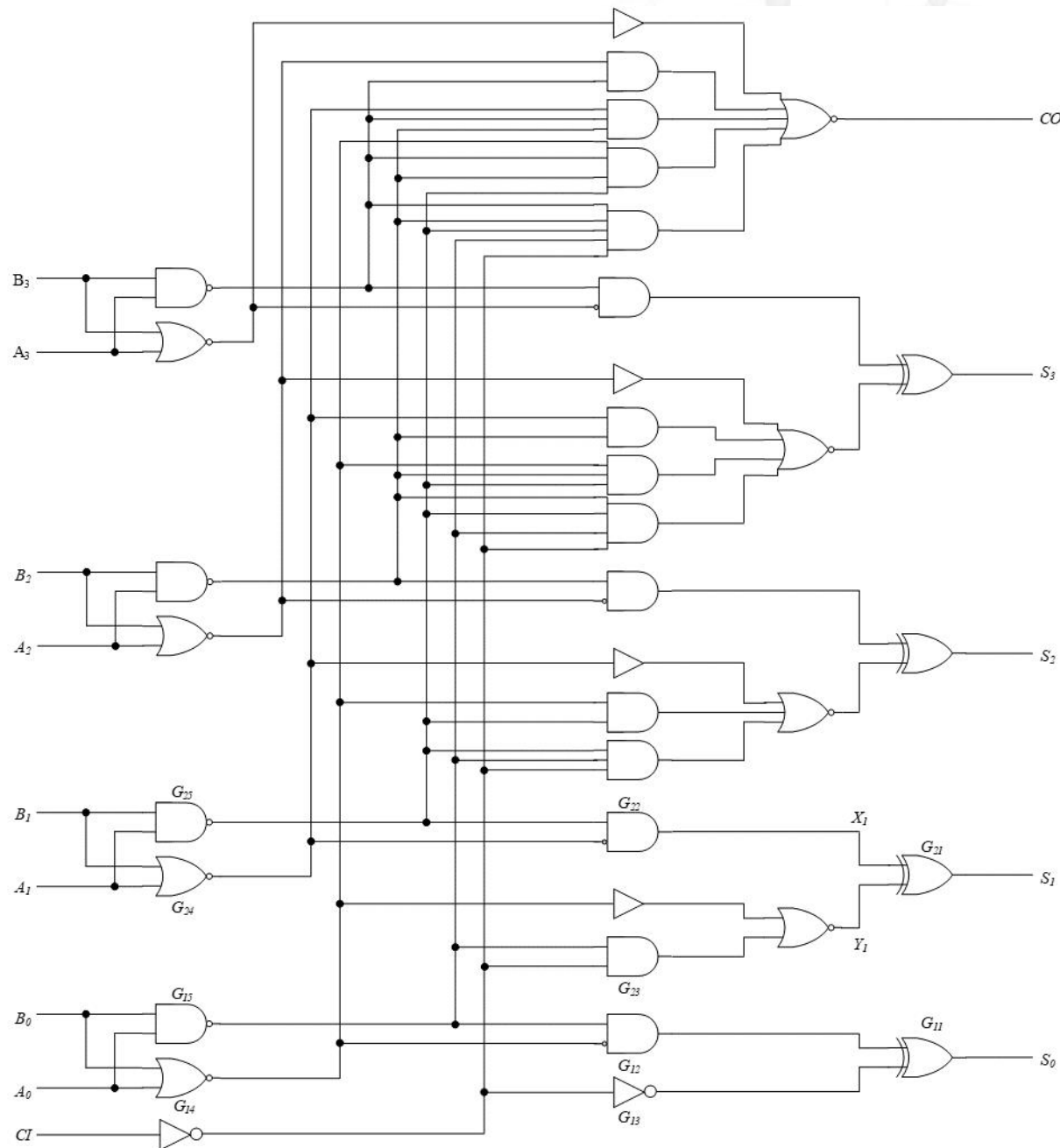
加法器

3. 多位加法器

(2) 超前进位加法器

优点：快，每1位的和及最后的进位基本同时产生。

缺点：电路复杂。

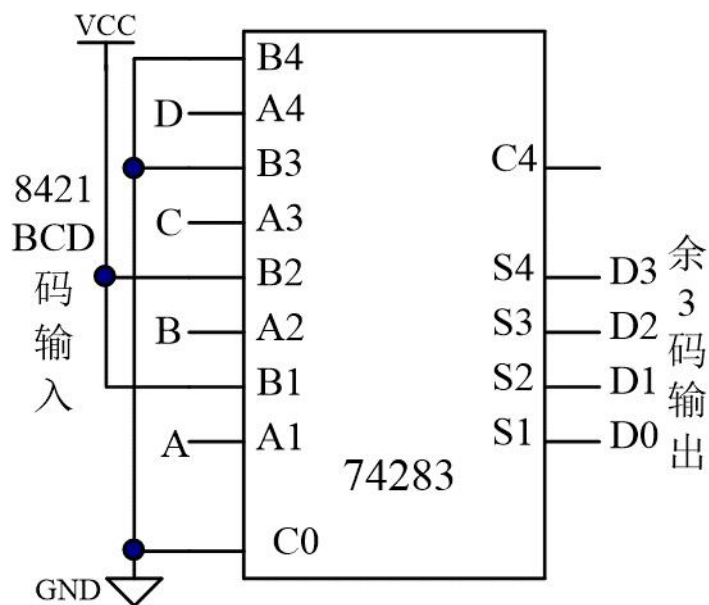


加法器

3. 多位加法器

(2) 超前进位加法器

【例】用4位二进制全加器74LS283完成1位8421 BCD码余3码的转换。



输入				输出			
D	C	B	A	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

-
- 4.5 数据选择器与数据分配器
- 4.6 加法器

• **4.7 比较器**

- 4.8 广义译码器概念
- 4.9 可编程逻辑器件的结构与原理
- 4.10 组合电路的竞争与冒险

比较器

1. 1位数值比较器

两个1位二进制数 A, B 比较有三种结果

A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$F_{A>B} = A\bar{B} \quad F_{A<B} = \bar{A}B \quad F_{A=B} = \bar{A}\bar{B} + AB = \overline{A\bar{B} + \bar{A}B}$$

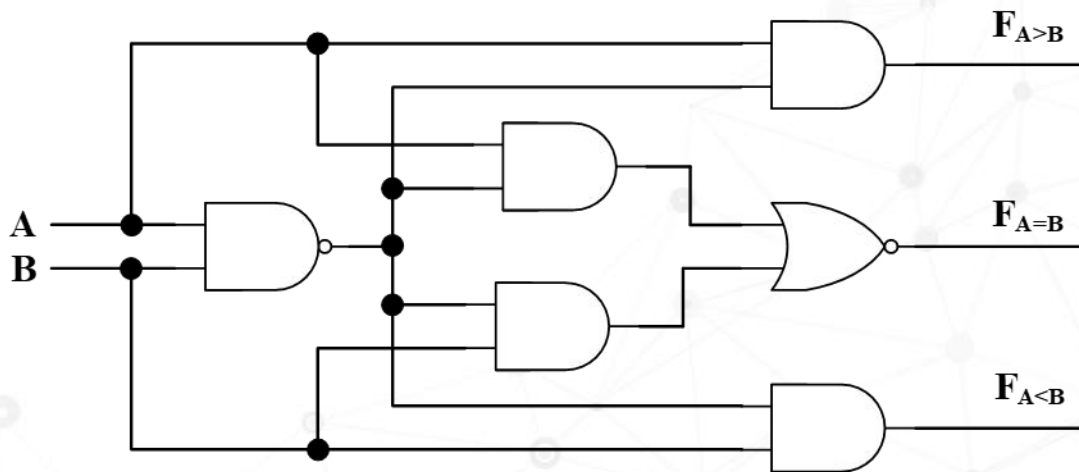
比较器

1. 1位数值比较器

$$F_{A>B} = A\bar{B} = \overline{A\bar{A}B}$$

$$F_{A<B} = \bar{A}B = \overline{B\bar{A}B}$$

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



比较器

2. 集成数值比较器 74LS85

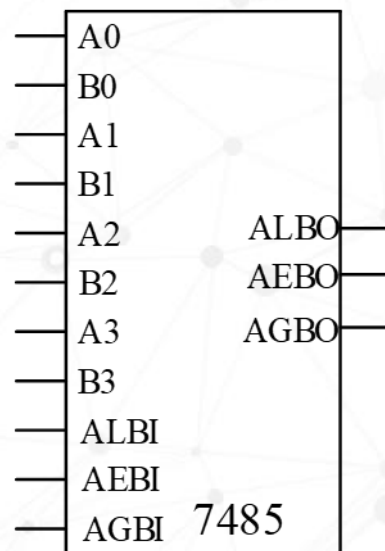
比较输入				级联输入			输出		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$A' > B'$	$A' < B'$	$A' = B'$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	×	×	×	×	×	×	1	0	0
$A_3 < B_3$	×	×	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 > B_2$	×	×	×	×	×	1	0	0
$A_3 = B_3$	$A_2 < B_2$	×	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	×	×	×	×	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	×	×	×	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1

$$F_{A>B} = A_3 \overline{B_3} + \overline{A_3 \oplus B_3} A_2 \overline{B_2} + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \overline{B_1} + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \oplus B_1 A_0 \overline{B_0} + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \oplus B_1 A_0 \oplus B_0 \cdot (A' > B')$$

$$F_{A<B} = \overline{A_3} B_3 + \overline{A_3 \oplus B_3} A_2 B_2 + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 B_1 + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \oplus B_1 A_0 B_0 + \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \oplus B_1 A_0 \oplus B_0 \cdot (A' < B')$$

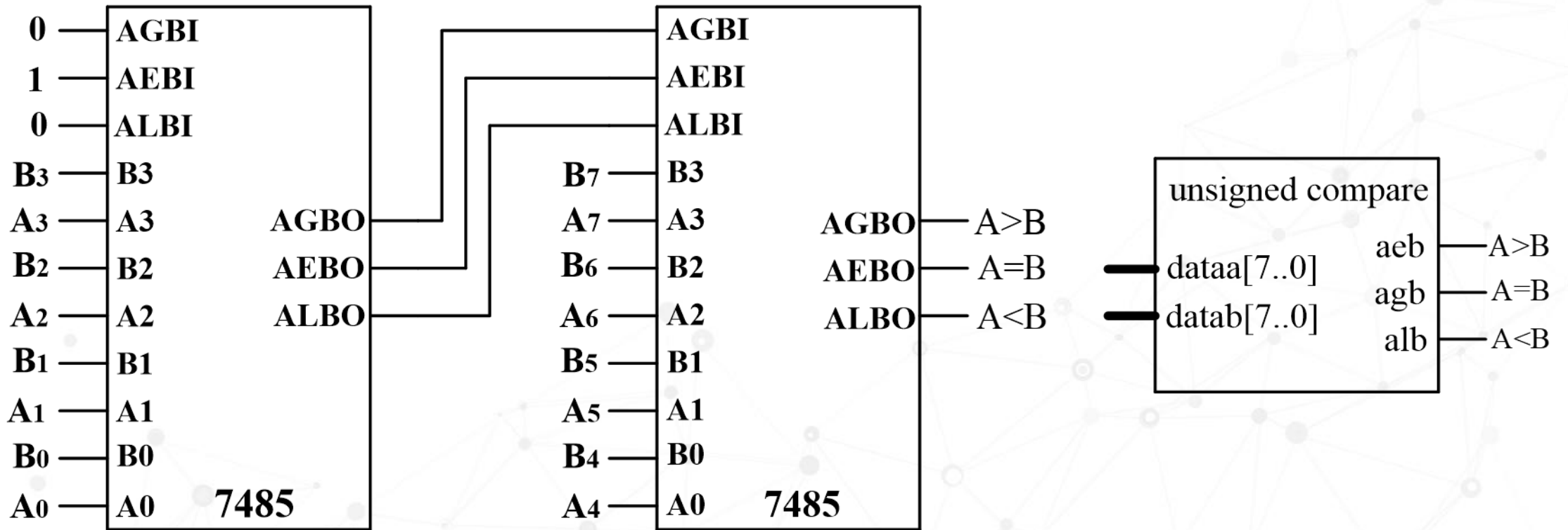
$$F_{A=B} = \overline{A_3 \oplus B_3} A_2 \oplus B_2 A_1 \oplus B_1 A_0 \oplus B_0 \cdot (A' = B')$$

原理：从高位比起，只有高位相等，才比较下一位。



比较器

3. 集成数值比较器应用举例 { 串联 并联

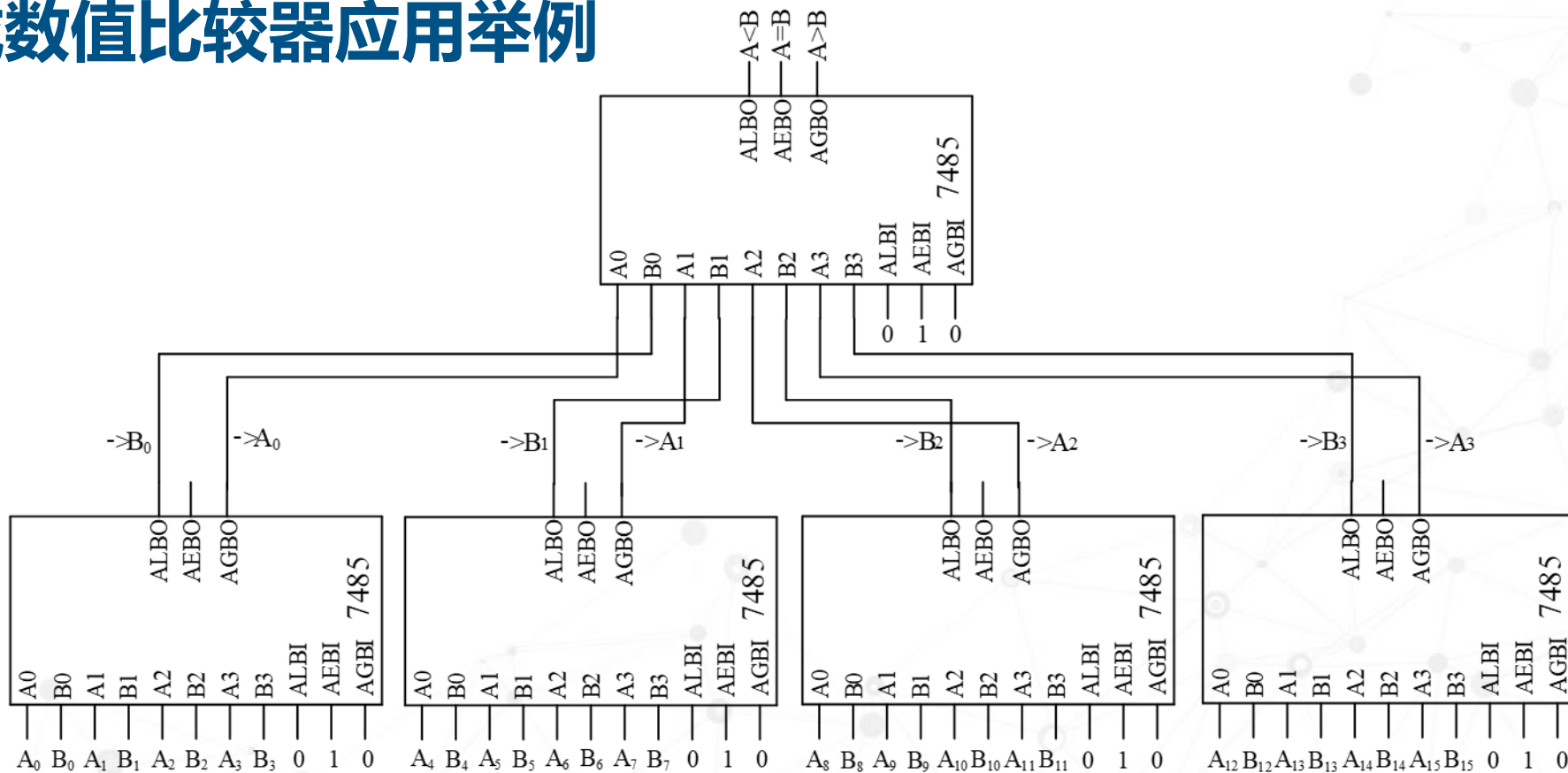


2片7485串联扩展为8位比较器

参数可设置无符号8位比较器

比较器

3. 集成数值比较器应用举例



5片四位二进制数值比较器并联扩展成的16位比较器

-
- 4.5 数据选择器与数据分配器
- 4.6 加法器
- 4.7 比较器

• 4.8 广义译码器概念

- 4.9 可编程逻辑器件的结构与原理
- 4.10 组合电路的竞争与冒险

广义译码器概念

各类组合运算器

分配器

比较器

多路选择器

加法器

各类编码器

各类译码器

.....

组合逻辑真值表的
表达输入输出关系的

广义译码器

传统的组合逻辑设计流程

现代的自动设计流程



-

- 4.6 加法器

- 4.7 比较器

- 4.8 广义译码器概念

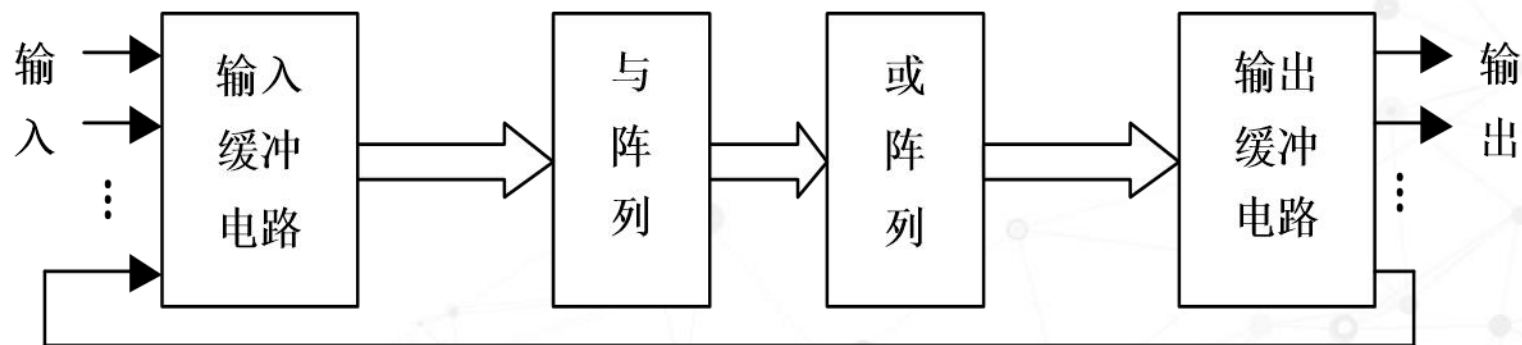
- **4.9 可编程逻辑器件的结构与原理**

- 4.10 组合电路的竞争与冒险

可编程逻辑器件的结构与原理

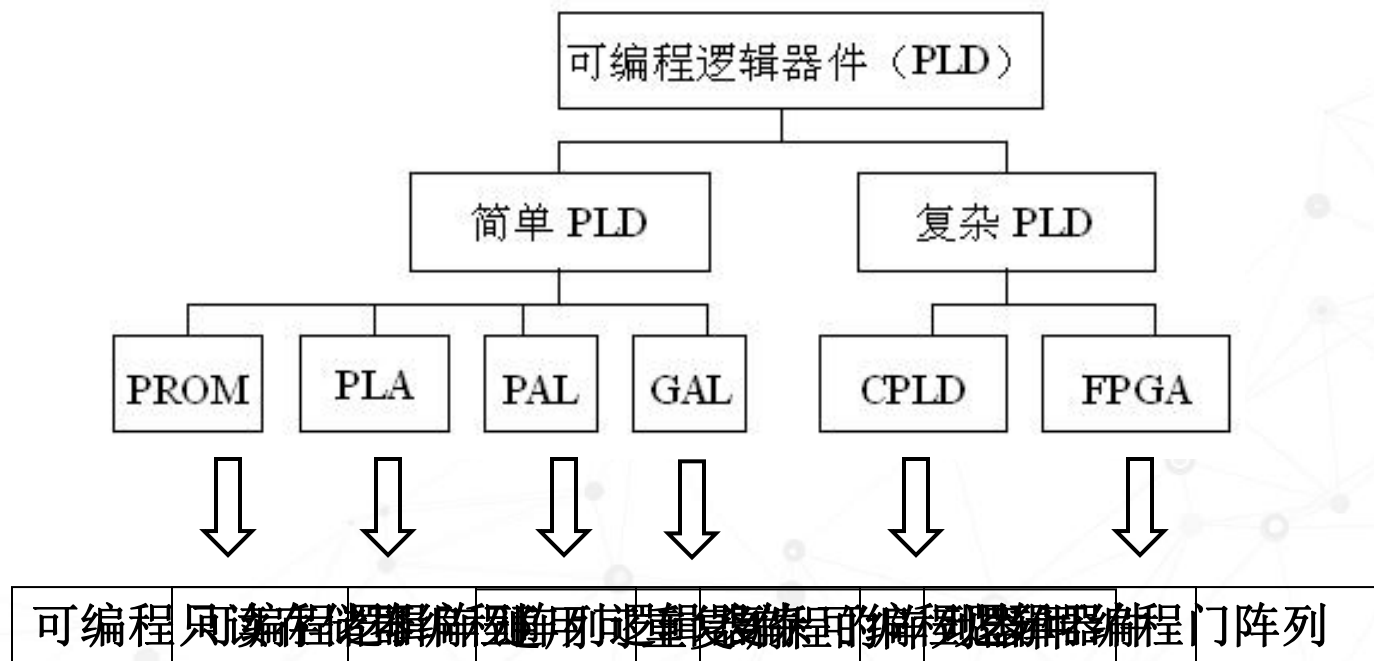
1. PLD概述

基本PLD器件的原理结构图



可编程逻辑器件的结构与原理

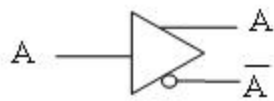
2. 简单PLD结构与工作原理



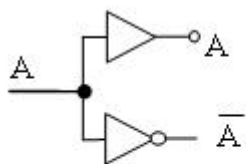
可编程逻辑器件的结构与原理

2. 简单PLD结构与工作原理

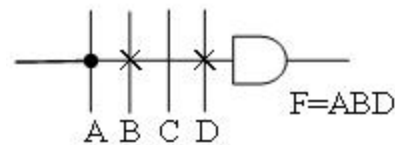
(1) 电路符号表示



PLD 的互补缓冲器



PLD 的互补输入



PLD 中与阵列表示



PLD 中或阵列的表示



未连接



固定连接



可编程连接

阵列线连接表示

可编程逻辑器件的结构与原理

2. 简单PLD结构与工作原理

(2) PROM

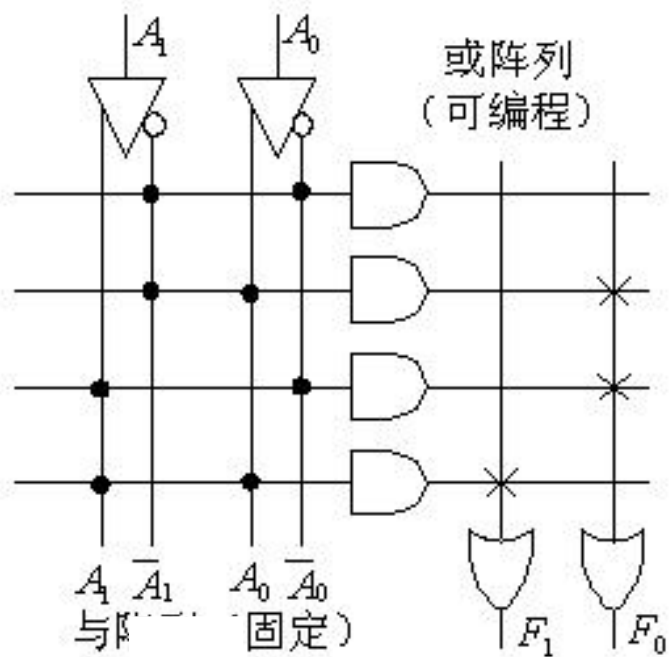


图 4.48 PROM 表达的 PLD 阵列图

$$F_0 = A_0 \overline{A_1} + \overline{A_0} A_1$$
$$F_1 = A_1 A_0$$

即

$$F_0 = A_0 \oplus A_1$$
$$F_1 = A_0 A_1$$

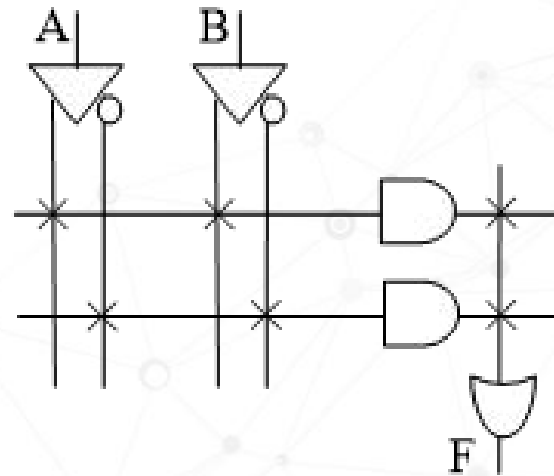
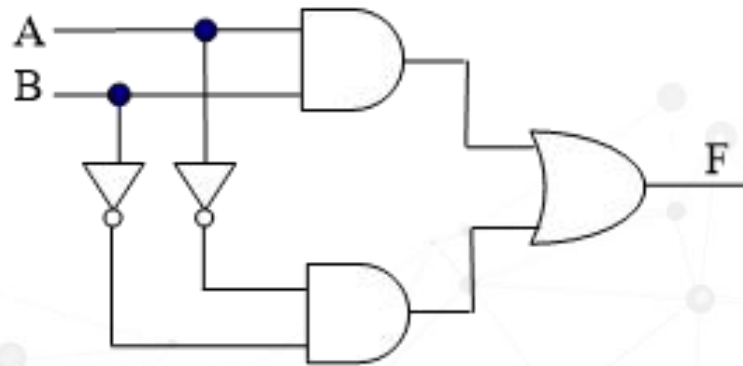
可编程逻辑器件的结构与原理

2. 简单PLD结构与工作原理

(2) PROM

【例】下图给出了函数F的逻辑图，假设与阵列和或阵列都是可编程的，试画出相应的PLD结构图。

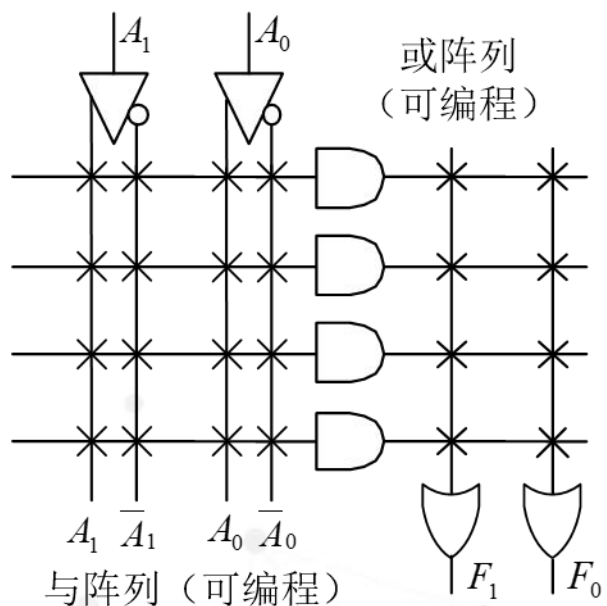
解： $F = AB + \bar{A}\bar{B}$



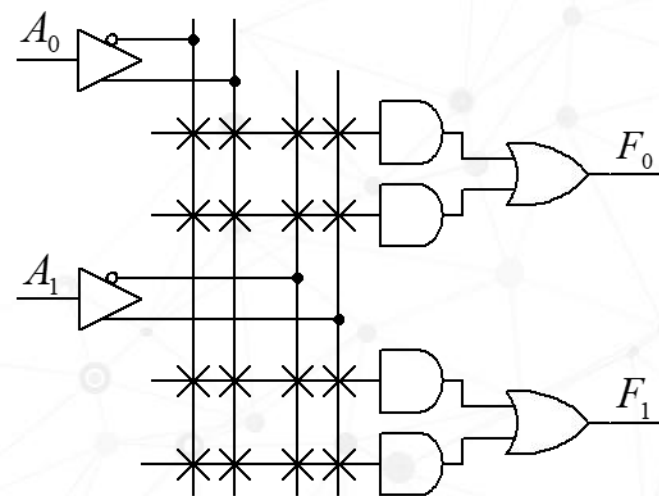
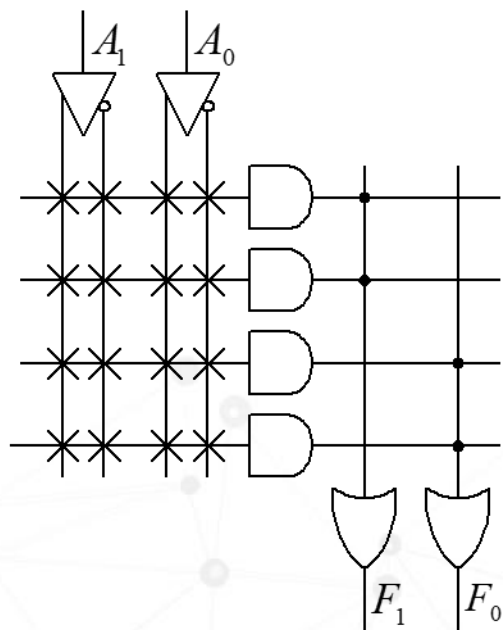
可编程逻辑器件的结构与原理

2. 简单PLD结构与工作原理

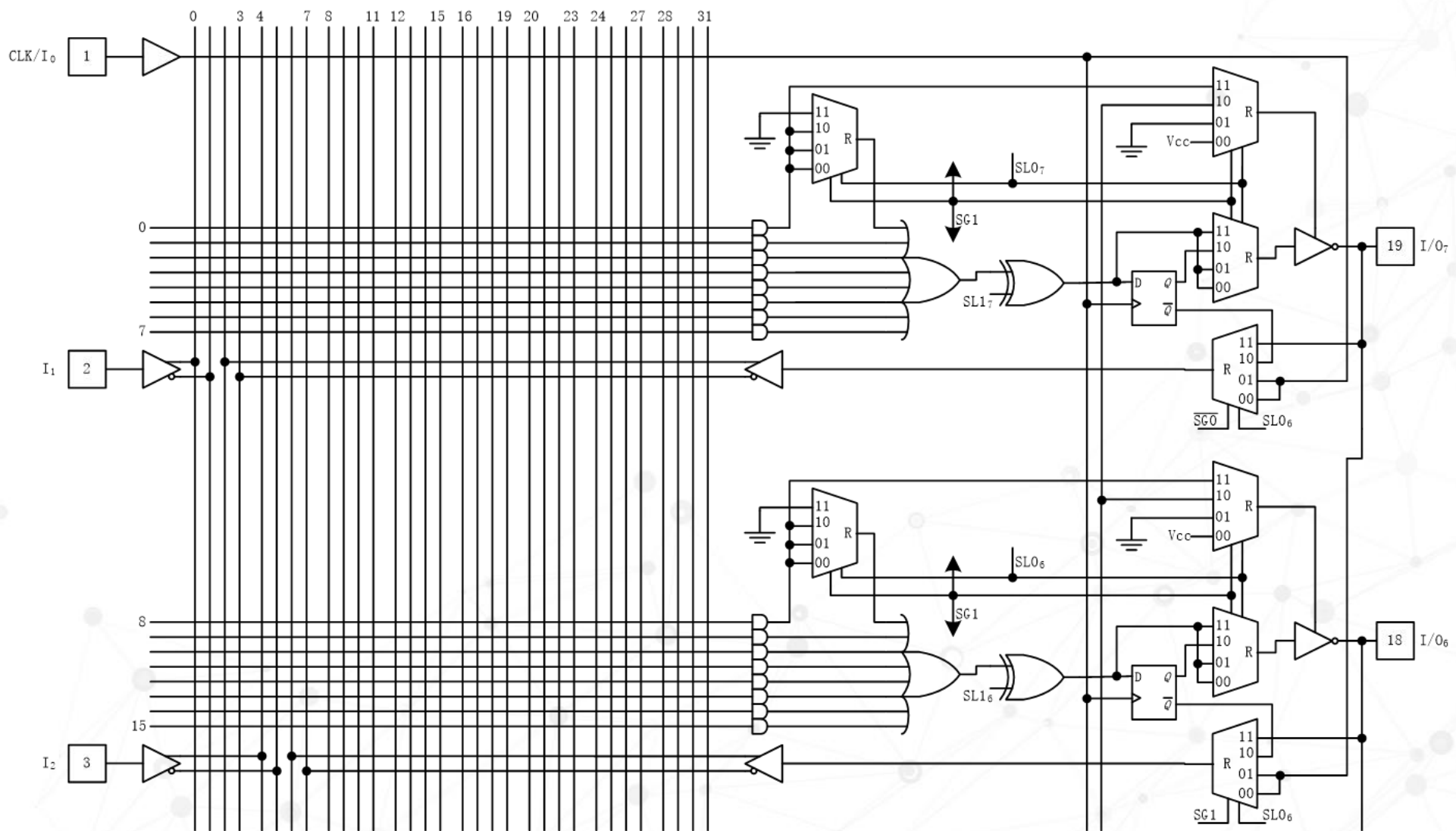
(3) PLA



(4) PAL



可编程逻辑器件的结构与原理



-
- 4.7 比较器
- 4.8 广义译码器概念
- 4.9 可编程逻辑器件的结构与原理
- **4.10 组合电路的竞争与冒险**

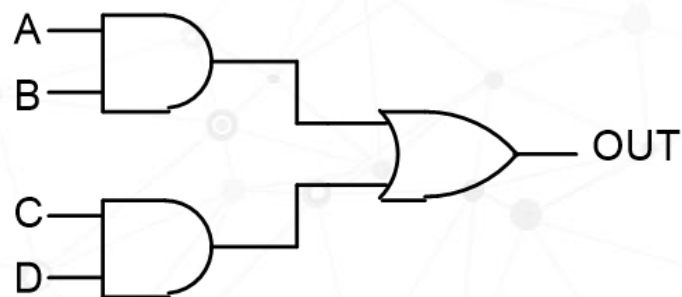
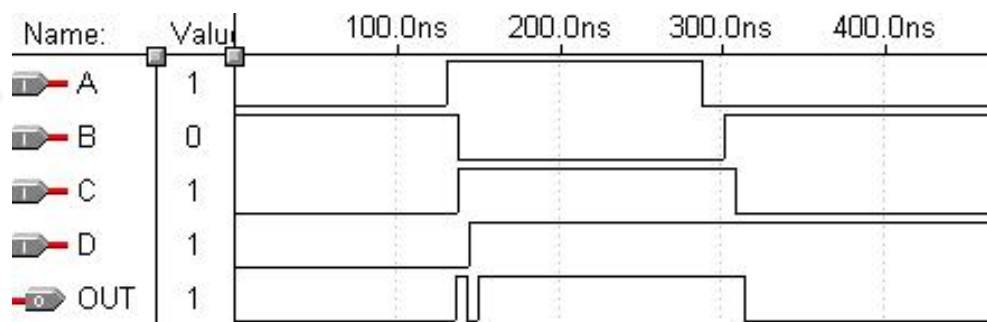
组合电路的竞争与冒险

一、竞争

两个输入“同时向相反的逻辑电平变化”，称存在“竞争”

二、竞争-冒险

因“竞争”而可能在输出产生尖峰脉冲的现象，称为“竞争-冒险”。



组合电路的竞争与冒险

1. 逻辑冒险现象的判断

(1) 代数法

判断是否存在 $A + \bar{A}$ 或 $A\bar{A}$ 的形式

【例】某逻辑函数表达式为 $F = A\bar{B} + BC$ ，试判断该逻辑电路是否可能产生冒险现象。

解：表达式中B以原变量和反变量的形式出现。

假设输入变量A=C=1，将A、C的值代入表达式，得 $F = \bar{B} + B$ ，理论上无论B为何值，该函数表达式F的值恒为1。当B发生变化时，可能使电路产生冒险现象。

组合电路的竞争与冒险

1. 逻辑冒险现象的判断

(2) 卡诺图法

观察是否存在某两个圈“相切”的情况

【例】某逻辑函数表达式为 $F=A\bar{B}+BC$ ，试判断该逻辑电路是否可能产生冒险现象。

BC	00	01	11	10
0	m_0	m_1	m_3 1	m_2
1	m_4 1	m_5 1	m_7 1	m_6

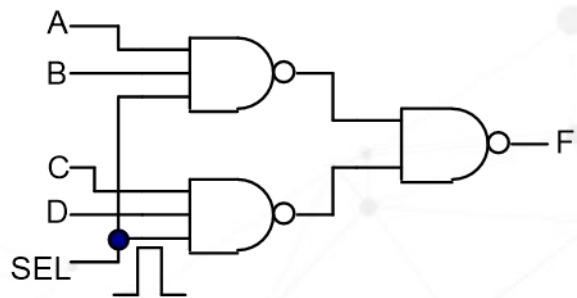
组合电路的竞争与冒险

2. 冒险现象的解决办法

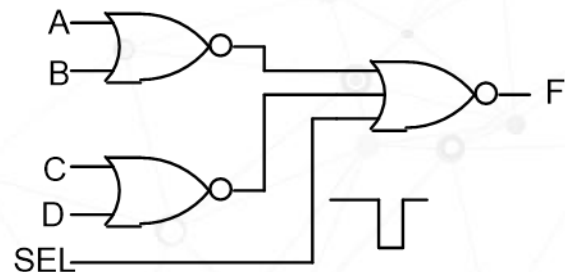
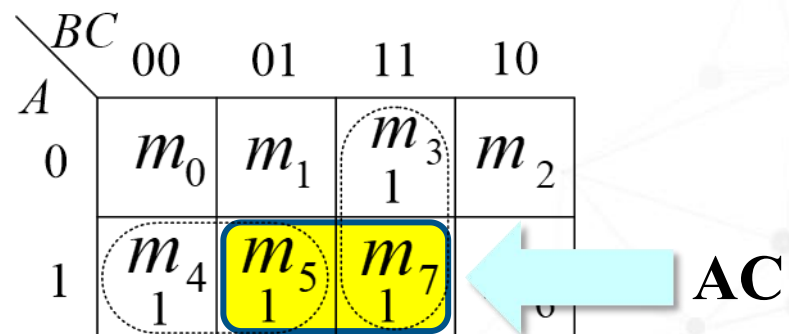
(1) 增加冗余项

$$F = A\bar{B} + BC + AC$$

(2) 选通法



选通脉冲SEL采用高电平有效的电路



选通脉冲SEL采用低电平有效的电路

The background features a complex network of white lines connecting various nodes, overlaid on a pattern of semi-transparent, overlapping hexagons in shades of blue, purple, and gold. The overall aesthetic is clean, modern, and technical.

本章 完