

# 第四章 组合逻辑电路

## 4.1 概 述

### 主要要求:

- 掌握组合逻辑电路和时序逻辑电路的概念。
- 了解组合逻辑电路的特点与描述方法。

## 4.3.2 译码器

### 主要要求：

- 理解译码的概念。
- 掌握二进制译码器 **74HC138** 的逻辑功能和使用方法。
- 理解其他常用译码器的逻辑功能和使用方法。
- 掌握**用二进制译码器实现组合逻辑电路**的方法。

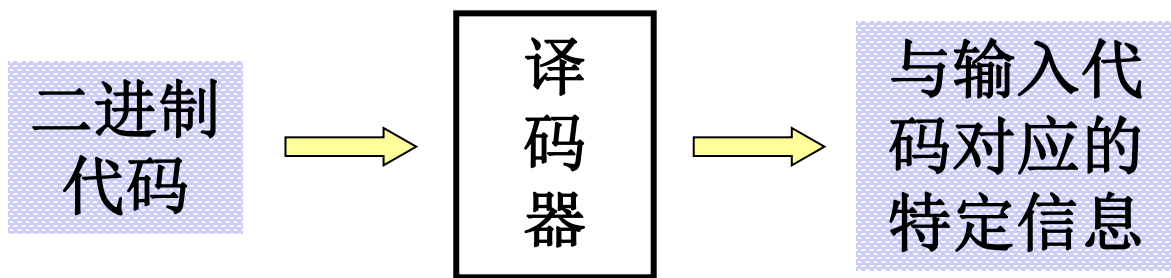
# 一、译码的概念与类型

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 (即 Decoder)

实现译码功能的电路



译码器

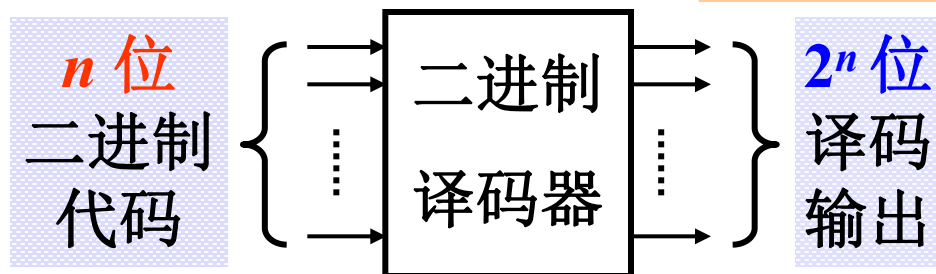
二进制译码器

二 - 十进制译码器

数码显示译码器

## 二、二进制译码器

将输入二进制代码译成相应输出信号的电路。



译码输入		译 码 输 出			
$A_1$	$A_0$	$Y_0$	$Y_1$	$Y_2$	$Y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

译码输出高电平有效

译码输入		译 码 输 出			
$A_1$	$A_0$	$Y'_0$	$Y'_1$	$Y'_2$	$Y'_3$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

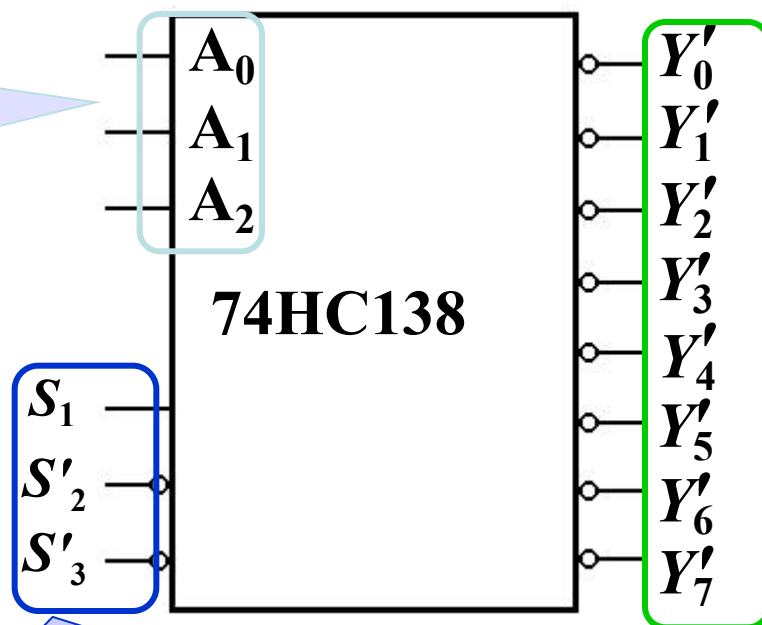
译码输出低电平有效

怎样写出输出端表达式

# (一) 3线—8线译码器 74HC138 简介

逻辑功能示意图

3 位二进制码输入端



8 个译码输出端  
低电平有效。

你能简单叙述  
一下它的逻辑  
功能吗？

使能端  $S_1$  高电平有效,  $S'_2$ 、 $S'_3$  低电平有效, 即当  $S_1 = 1$ ,  $S'_2 = S'_3 = 0$  时译码, 否则禁止译码。

# 74HC138的功能表:

禁止译码

允许译码器工作

输 入					输 出							
$S_1$	$S_2' + S_3'$	$A_2$	$A_1$	$A_0$	$Y_7'$	$Y_6'$	$Y_5'$	$Y_4'$	$Y_3'$	$Y_2'$	$Y_1'$	$Y_0'$
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

输出端逻辑式  
怎样写?

## 输出逻辑函数式

$$Y_0' = (A_2'A_1'A_0')' = m_0'$$

$$Y_1' = (A_2'A_1'A_0)' = m_1'$$

$$Y_2' = (A_2'A_1A_0')' = m_2'$$

$$Y_3' = (A_2'A_1A_0)' = m_3'$$

$$Y_4' = (A_2A_1'A_0')' = m_4'$$

$$Y_5' = (A_2A_1'A_0)' = m_5'$$

$$Y_6' = (A_2A_1A_0')' = m_6'$$

$$Y_7' = (A_2A_1A_0)' = m_7'$$

## 输出端一般表达式:

$$Y_i' = (S_1(S_2' + S_3')'m_i)'$$

其输出端能提供输入变量的全部最大项。

有什么实际意义？

## (二) 用二进制译码器实现组合逻辑函数

由于二进制译码器的输出端能提供输入变量的全部最小项，而任何组合逻辑函数都可以变换为最小项之和的标准式，因此用二进制译码器和门电路可实现任何组合逻辑函数。当译码器输出低电平有效时，多选用与非门；译码器输出高电平有效时，多选用或门。



★ [例] 试用译码器和门电路实现逻辑函数

$$Y = A'B'C + ABC' + C$$

解: (1) 根据逻辑函数选择译码器

选用 3 线 - 8 线译码器 74HC138,  
并令  $A_2 = A$ ,  $A_1 = B$ ,  $A_0 = C$ 。

(2) 将函数式变换为标准与 - 或式

$$\begin{aligned} Y &= A'B'C + ABC' + C \\ &= A'B'C + A'BC + AB'C + ABC' + ABC \\ &= m_1 + m_3 + m_5 + m_6 + m_7 \end{aligned}$$

(3) 根据译码器的输出有效电平确定需用的门电路

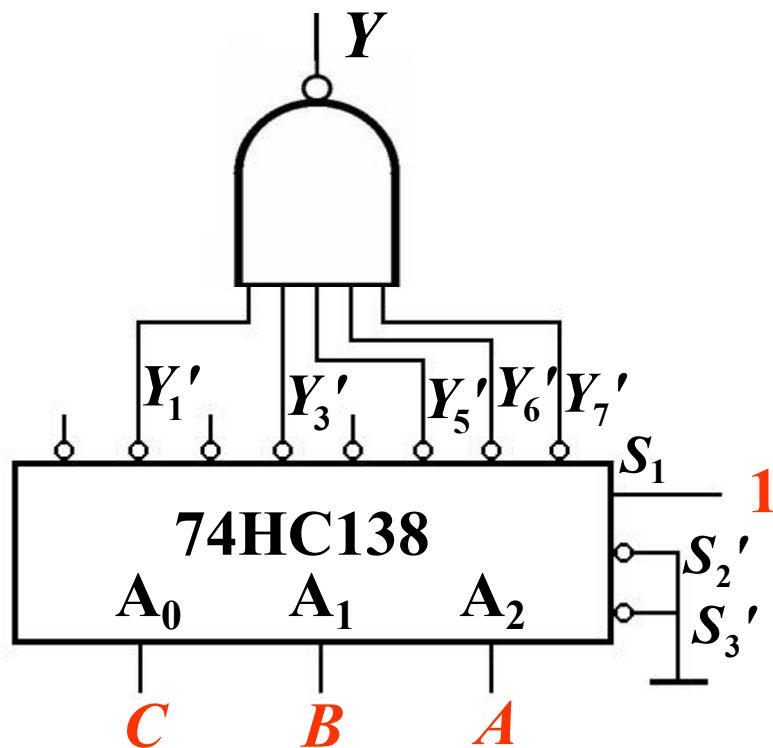
74HC138 输出低电平有效,  $Y_i' = m_i'$ ,  $i = 0 \sim 7$

因此, 将  $Y$  函数式变换为

$$Y = m_1 + m_3 + m_5 + m_6 + m_7$$
$$= (m_1' \cdot m_3' \cdot m_5' \cdot m_6' \cdot m_7')'$$
$$= (Y_1' \cdot Y_3' \cdot Y_5' \cdot Y_6' \cdot Y_7')'$$

采用 5 输入与非门, 其输入取自  $Y_1'$ 、 $Y_3'$ 、 $Y_5'$ 、 $Y_6'$  和  $Y_7'$ 。

(4) 画连线图



★ [例] 试用译码器实现全加器。

解：(1) 分析设计要求，列出真值表

设被加数为  $A_i$ ，加数为  $B_i$ ，低位进位数为  $C_{i-1}$ 。  
输出本位和为  $S_i$ ，向高位的进位数为  $C_i$ 。

列出全加器的真值表如下：

输 入			输 出	
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
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

(2) 根据真值表写函数式

$$\begin{cases} S_i = m_1 + m_2 + m_4 + m_7 \\ C_i = m_3 + m_5 + m_6 + m_7 \end{cases}$$

(3) 选择译码器

选用 3 线 - 8 线译码器  
74HC138。并令  $A_2 = A_i$ ，  
 $A_1 = B_i$ ， $A_0 = C_{i-1}$ 。

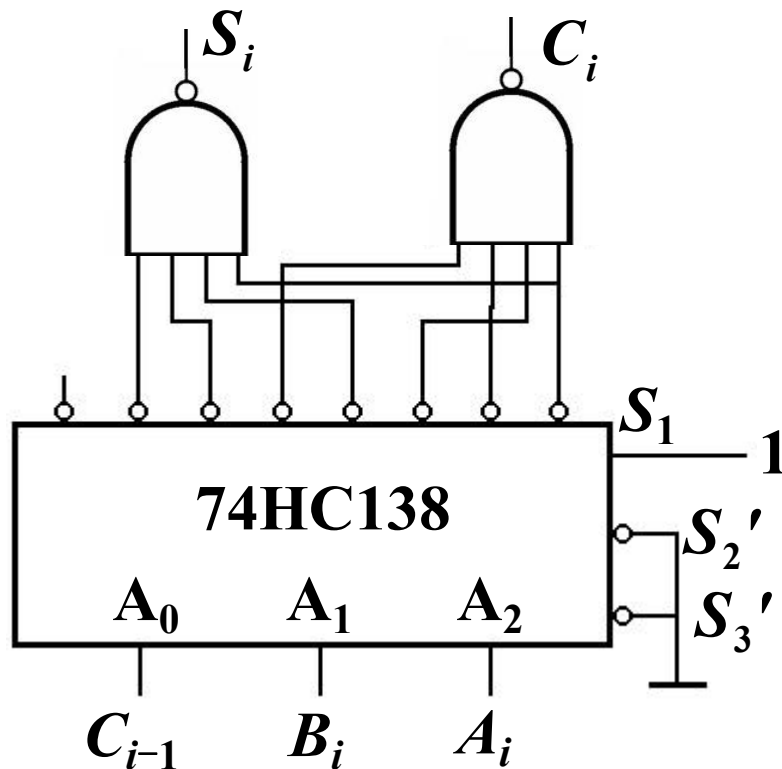
#### (4) 根据译码器的输出有效电平确定需用的门电路

74HC138 输出低电平有效,  $Y_i' = m_i'$ ,  $i = 0 \sim 7$

因此, 将函数式变换为

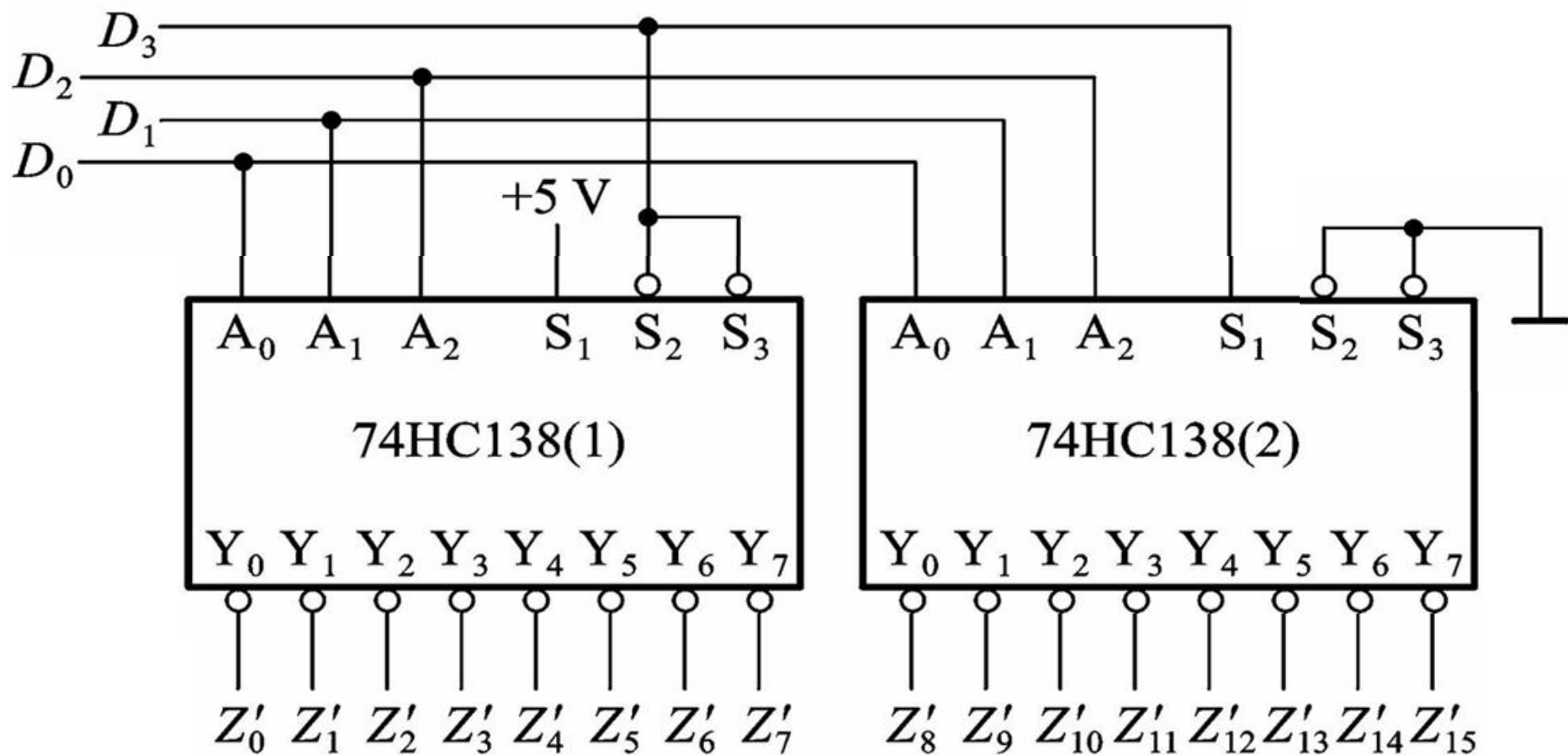
$$\begin{cases} S_i = m_1 + m_2 + m_4 + m_7 = (m_1' \cdot m_2' \cdot m_4' \cdot m_7')' \\ C_i = m_3 + m_5 + m_6 + m_7 = (m_3' \cdot m_5' \cdot m_6' \cdot m_7')' \end{cases}$$

#### (5) 画连线图



### (三) 译码器的扩展

例如 两片 74LS138 组成的 4 线 - 16 线译码器。



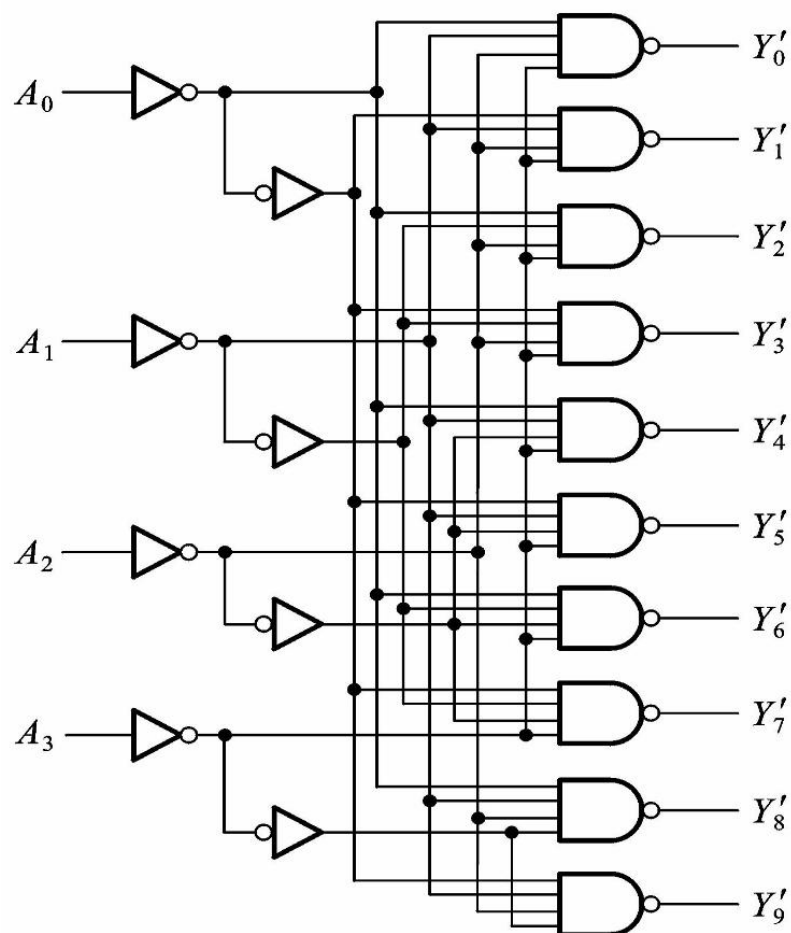
两片2线—4线译码器怎样扩展成3线—8线译码器？

4线—16线译码器能否取代3线—8线译码器？

### 三、二—十进制译码器

将 BCD 码的十组代码译成 0 ~ 9 十个对应输出信号的电路，又称 **4 线—10 线** 译码器。

8421BCD 码输入端，从高位到低位依次为  $A_3$ 、 $A_2$ 、 $A_1$  和  $A_0$ 。



10 个译码输出端，低电平 0 有效。

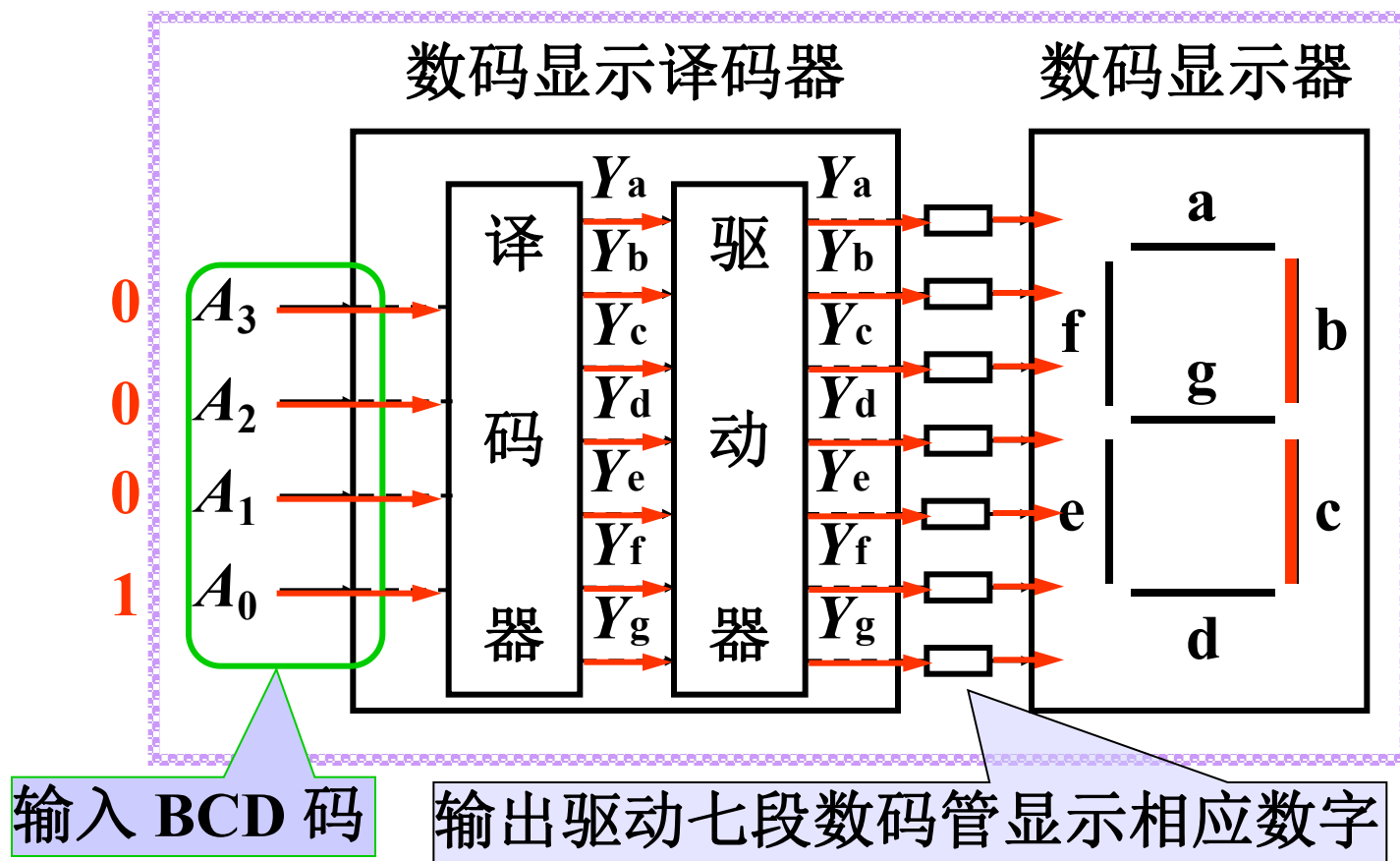
### 4 线-10 线译码器 74LS42 真值表

[illegible]

## 四、数码显示译码器

将输入的 BCD 码译成相应输出信号，以驱动显示器显示出相应数字的电路。

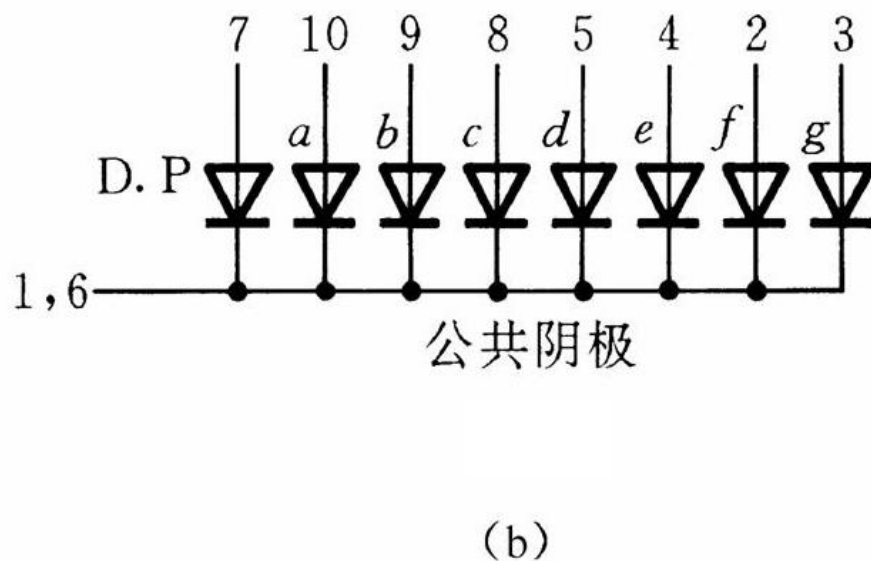
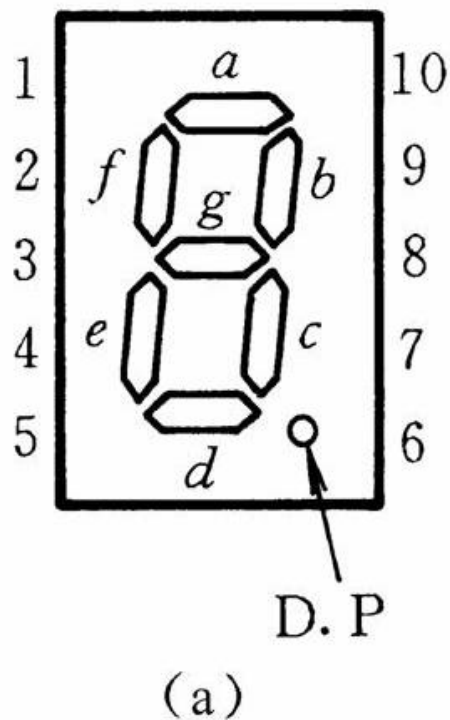
### (一) 数码显示译码器的结构和功能示意

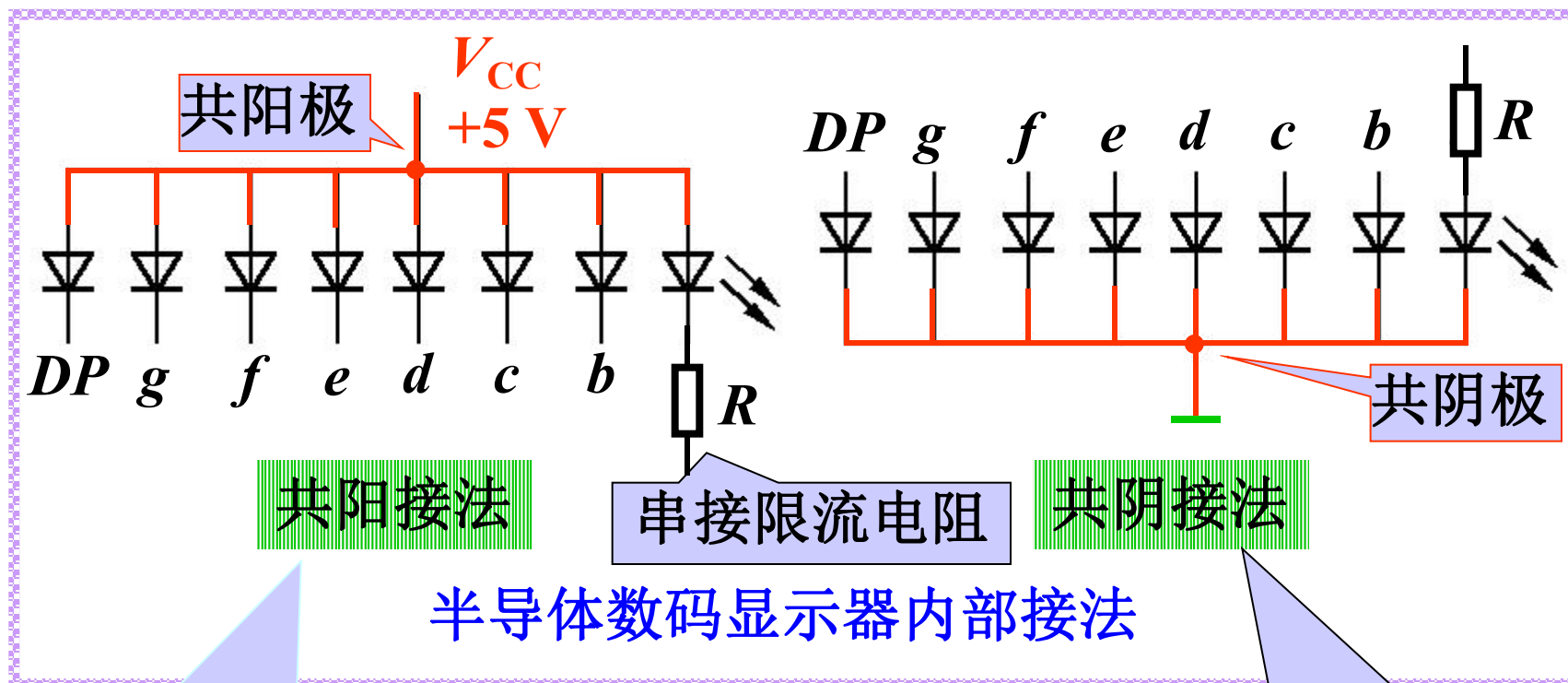




## 四、数码显示译码器

将输入的 BCD 码译成相应输出信号，  
七段字符显示以驱动显示器显示出相应数字的电路。





$a \sim g$  和  $DP$  为低电平时才能点亮相应发光段。

$a \sim g$  和  $DP$  为高电平时才能点亮相应发光段。

# • BCD七段字符显示译码器（代码转换器）7448

输 入					输 出							字形
数字	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>a</sub>	Y <sub>b</sub>	Y <sub>c</sub>	Y <sub>d</sub>	Y <sub>e</sub>	Y <sub>f</sub>	Y <sub>g</sub>	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	0	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	0	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	0	0	1	1	
10	1	0	1	0	0	0	0	1	1	0	1	
11	1	0	1	1	0	0	1	1	0	0	1	
12	1	1	0	0	0	1	0	0	0	1	1	
13	1	1	0	1	1	0	0	1	0	1	1	
14	1	1	1	0	0	0	0	1	1	1	1	
15	1	1	1	1	0	0	0	0	0	0	0	

# 真值表



# 卡诺图

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	0	1	1
	01	0	1	1	0
	11	0	1	0	0
	10	1	1	0	0

(a)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	1	1	1
	01	1	0	1	0
	11	1	0	0	0
	10	1	1	0	0

(b)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	1	1	0
	01	1	1	1	1
	11	0	0	0	0
	10	1	1	1	0

(c)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	0	1	1
	01	0	1	0	1
	11	0	1	0	1
	10	1	0	1	1

(d)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	0	0	1
	01	0	0	0	1
	11	0	0	0	1
	10	1	0	0	1

(e)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	1	0	0	0
	01	1	1	0	1
	11	1	1	0	1
	10	1	1	0	0

(f)

$A_1A_2$					
$A_3A_2$		00	01	11	10
	00	0	0	1	1
	01	1	1	0	1
	11	1	1	0	1
	10	1	1	1	1

(g)

# BCD—七段显示译码器7448的逻辑图

$$Y_a = (A_3' A_2' A_1' A_0 + A_3 A_1 + A_2 A_0)'$$

$$Y_b = (A_3 A_1 + A_2 A_1 A_0' + A_2 A_1' A_0)'$$

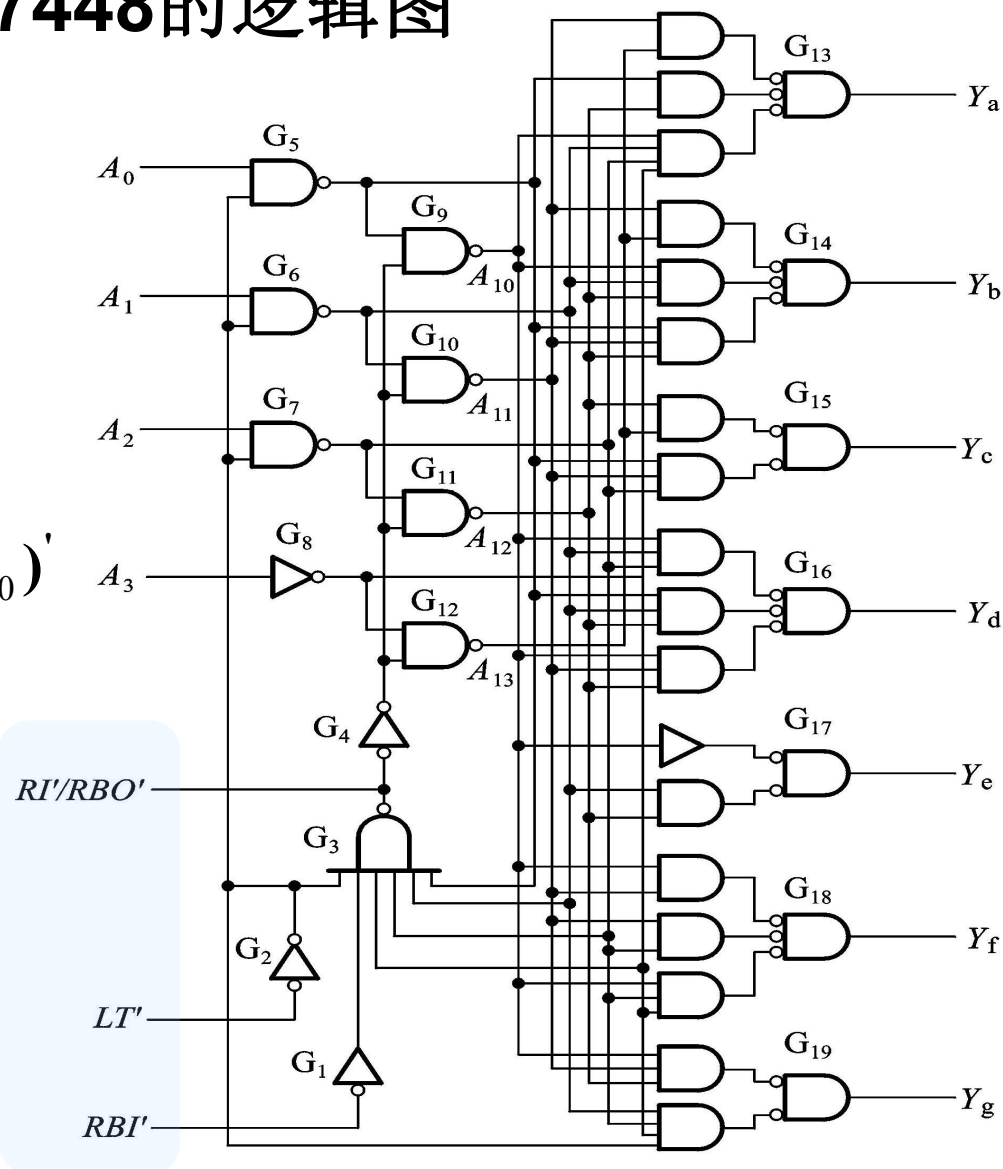
$$Y_c = (A_3 A_2 + A_2' A_1 A_0')'$$

$$Y_d = (A_2 A_1 A_0 + A_2 A_1' A_0' + A_2' A_1' A_0)'$$

$$Y_e = (A_2 A_1' + A_0)'$$

$$Y_f = (A_3' A_2' A_0 + A_2' A_1 + A_1 A_0)'$$

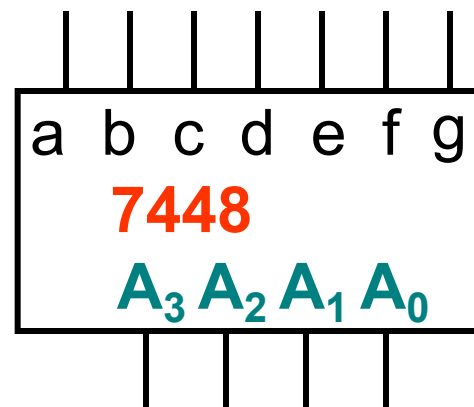
$$Y_g = (A_3' A_2' A_1' + A_2 A_1 A_0)'$$



## 七段译码器

$A_3$ 、 $A_2$ 、 $A_1$ 、 $A_0$ : BCD码输入信号。

$a \sim g$ : 译码输出, 高电平有效。



$\overline{RI}/\overline{RBO}$  灭灯输入/灭零输出信号

(1) 灭灯输入。低电平时, 数码管全灭;

(2) 灭零输出信号。  $\overline{RBI} = 0$  时,  $\overline{RBO} = 0$ 。

$\overline{LT}$  : 试灯信号输入。当  $\overline{LT} = 0$  且  $\overline{RI} = 1$  (无效) 时, 不论  $A_3 \sim A_0$  状态如何,  $a \sim g$  七段全亮。

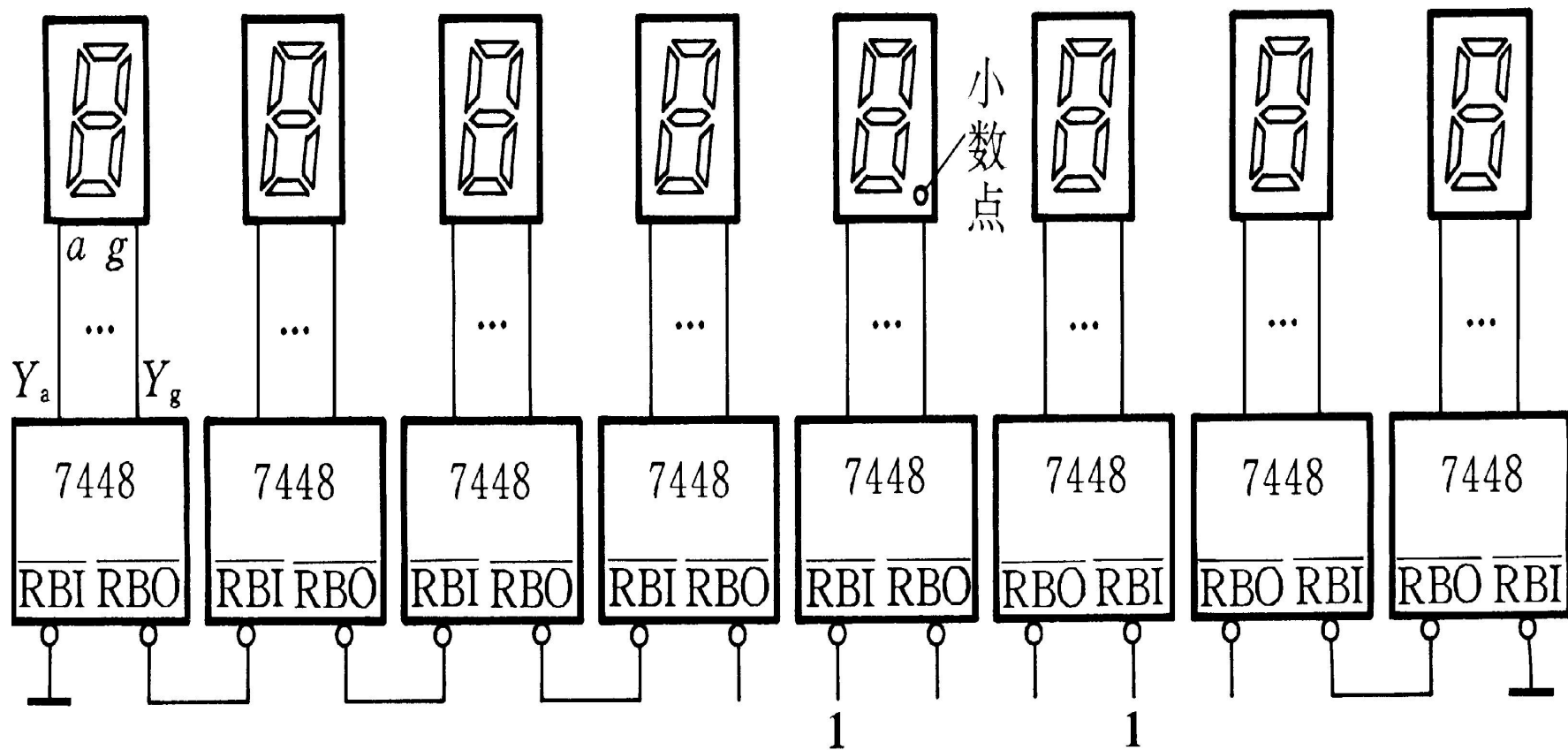
$\overline{RBI}$  : 灭零输入信号 (不显示 0, 其它数码正常显示)。

$\overline{RBI} = 0$  (  $\overline{LT} = 1$  ) 时, 不显示数码 0。

$\overline{RBO}$  作为灭零输出。

$\overline{\text{LT}}$	$\overline{\text{RBI}}$	$\text{A}_3\text{A}_2\text{A}_1\text{A}_0$	$\overline{\text{RI/RBO}}$	abcdefg	显示字型
1	1	0 0 0 0	1	111110	
1	X	0 0 0 1	1	0110000	
1	X	0 0 1 0	1	1101101	
1	X	0 0 1 1	1	1111001	
1	X	0 1 0 0	1	0110011	
1	X	0 1 0 1	1	1011011	
1	X	0 1 1 0	1	0011111	
1	X	0 1 1 1	1	1110000	
1	X	1 0 0 0	1	1111111	
1	X	1 0 0 1	1	1110011	
X	X	x x x x	0	0000000	数码管的所有发光段均熄灭
1	0	0 0 0 0	0	0000000	灭零，数码管0灭
0	X	x x x x	1	1111111	

# 有灭零控制的8位数码显示系统

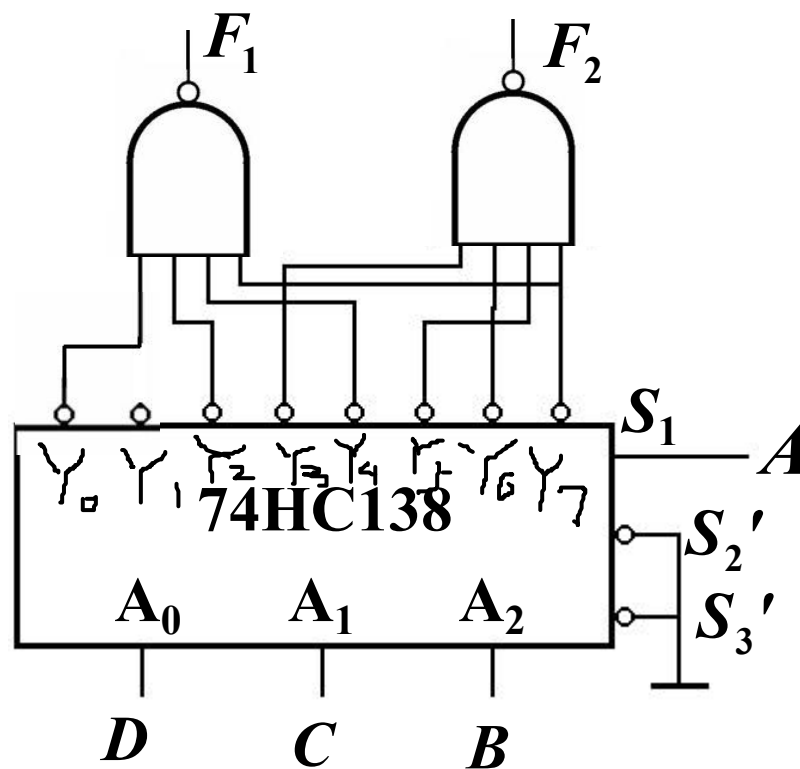


## 灭零控制

- 整数部分：最高位是0，而且灭掉以后，输出次高位的  $\overline{RBI}$  输入信号  $\overline{RBO}$  作为
- 小数部分：最低位是0，而且灭掉以后，输出为次低位的  $\overline{RBI}$  输入信号  $\overline{RBO}$  作

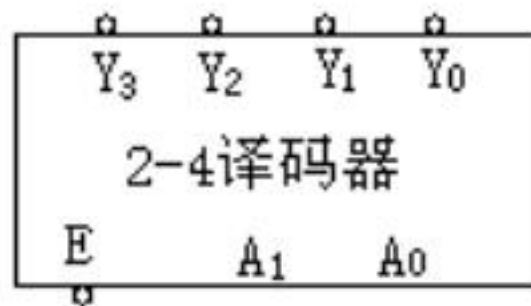
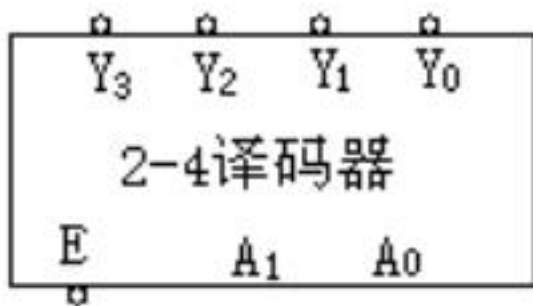


讨论： 写出下面电路中 $F_1$ 的最简与或式， $F_2$ 的最简或与式。



讨论：使用如图所示电路中的译码器和与门实现逻辑函数。

$$F(A, B, C) = B'C + BC' + AC$$



### 4.3.3 数据选择器

#### 主要要求:

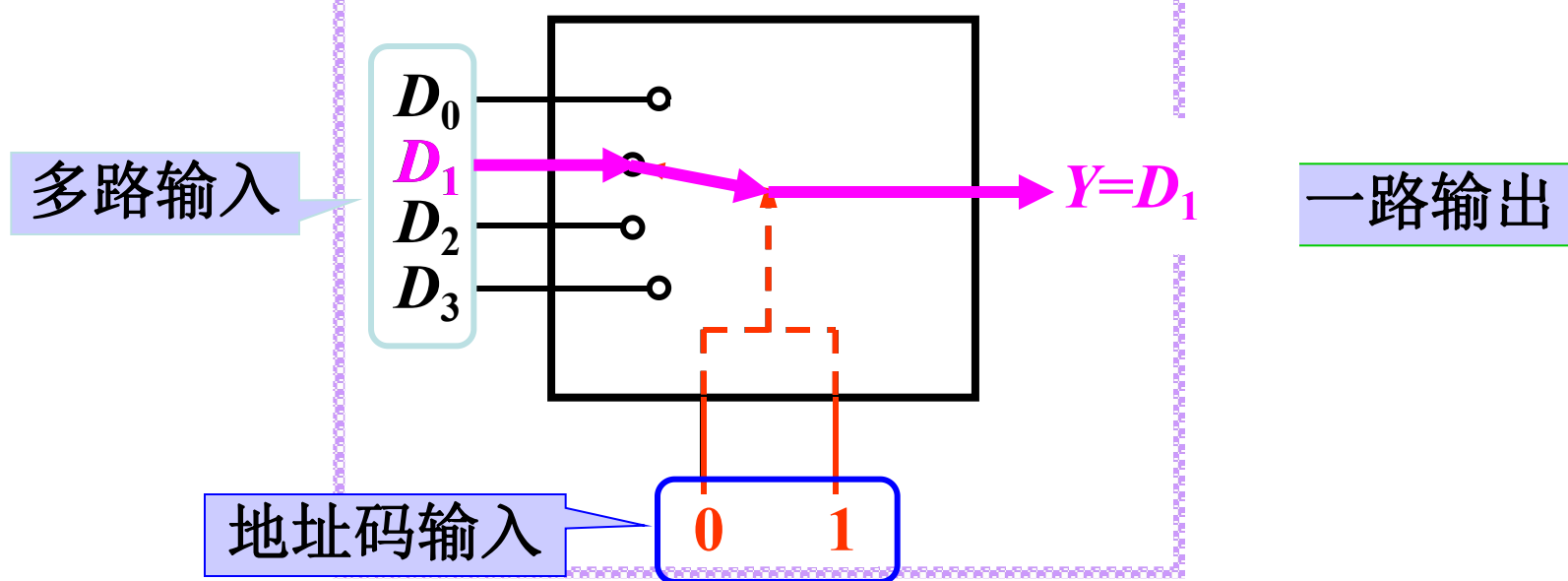
- 理解数据选择器和数据分配器的作用。
- 理解常用数据选择器的逻辑功能及其使用。
- 掌握用数据选择器实现组合逻辑电路的方法。

# 一、数据选择器和数据分配器的作用

**数据选择器：**根据地址码的要求，从多路输入信号中选择其中一路输出的电路。

又称多路选择器 (Multiplexer, 简称MUX) 或多路开关。

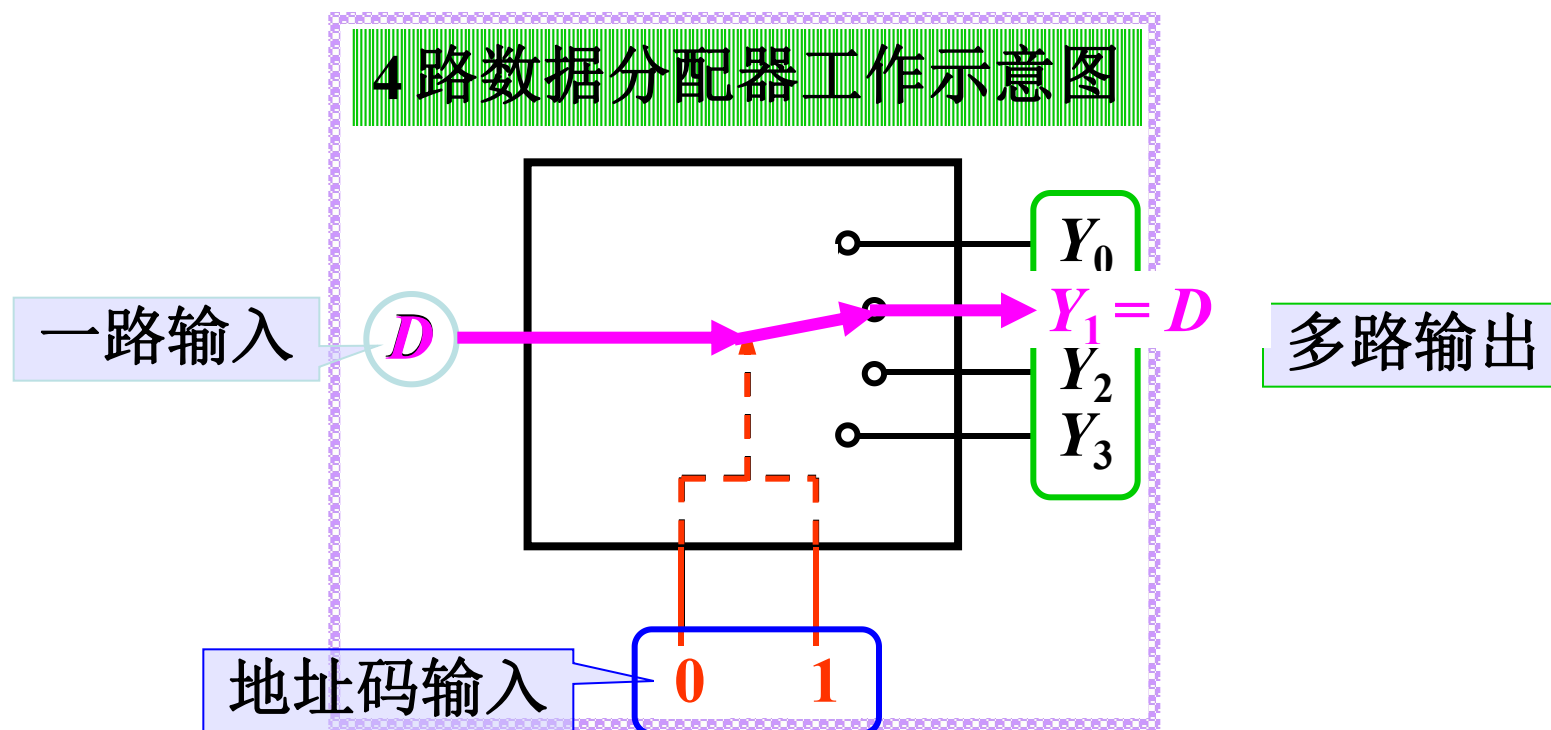
4 选 1 数据选择器工作示意图



数据选择器的输入信号个数  $N$  与地址码个数  $n$  的关系为  $N = 2^n$

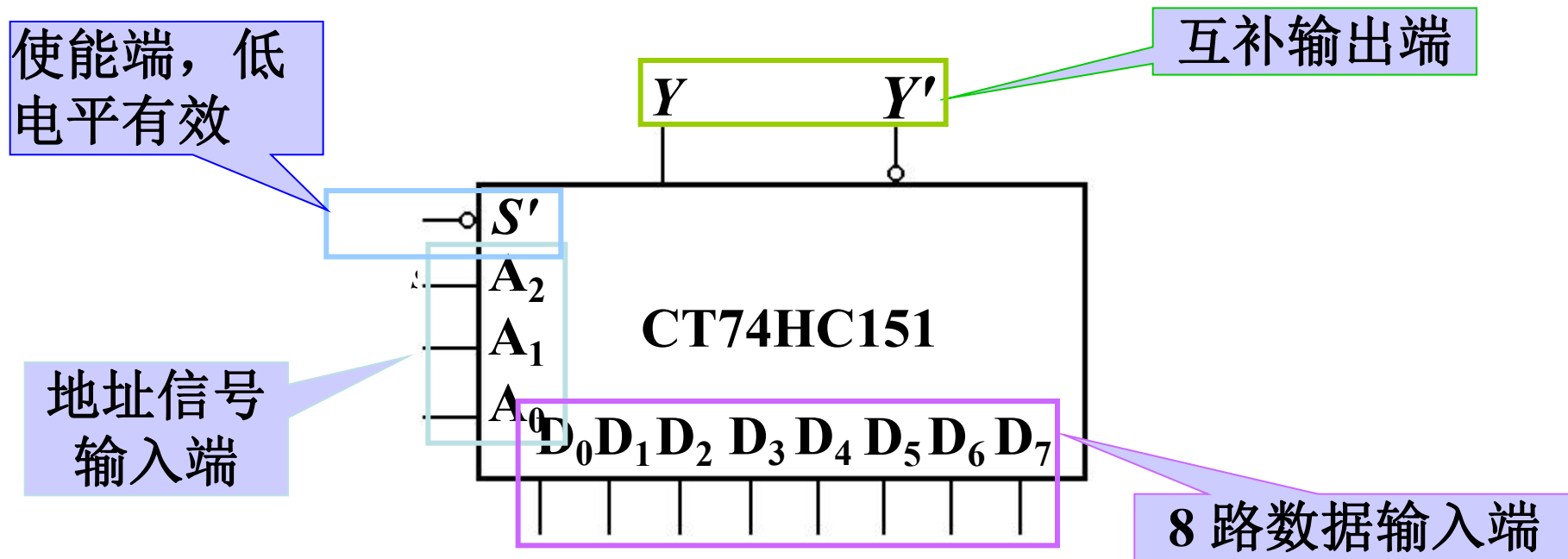
**数据分配器：**根据地址码的要求，将一路数据分配到指定输出通道上去的电路。

**Demultiplexer, 简称DMUX**



## 二、数据选择器的逻辑功能及其使用

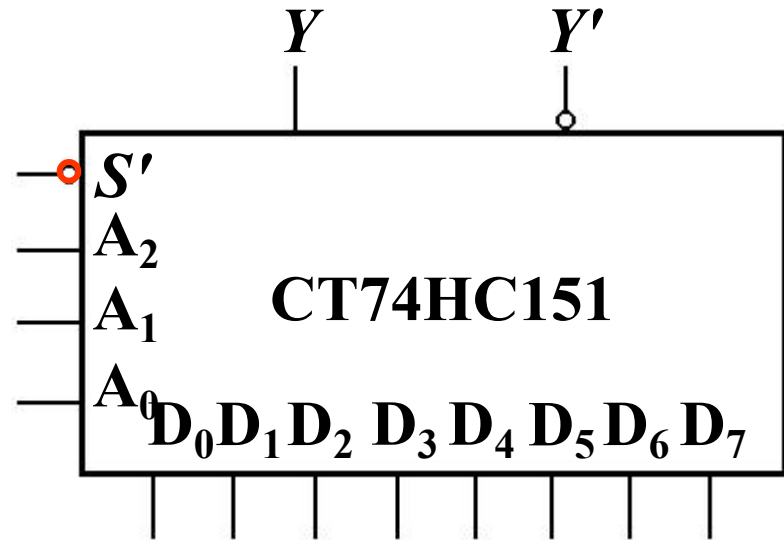
### 8 选 1 数据选择器 CT74HC151



CT74HC151的逻辑功能示意图

# 8 选 1 数据选择器 CT74HC151 真值表

输 入				输 出
$S'$	$A_2$	$A_1$	$A_0$	$Y$
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	$D_7$



## CT74HC151 输出函数表达式

输 入				输 出
$S'$	$A_2$	$A_1$	$A_0$	$Y$
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	$D_7$

$$\begin{aligned}
 Y &= A_2'A_1'A_0'D_0 + A_2'A_1'A_0D_1 + \\
 &\quad A_2'A_1A_0'D_2 + A_2'A_1A_0D_3 + \\
 &\quad A_2A_1'A_0'D_4 + A_2A_1'A_0D_5 + \\
 &\quad A_2A_1A_0'D_6 + A_2A_1A_0D_7 \\
 &= m_0D_0 + m_1D_1 + m_2D_2 + m_3D_3 + \\
 &\quad m_4D_4 + m_5D_5 + m_6D_6 + m_7D_7
 \end{aligned}$$

其输出端能提供地址  
输入变量的全部最小项。

能实现函数  
发生器吗？  
怎样实现？