

第3章

组合逻辑电路

内 容 提 要

本章主要介绍组合逻辑电路的分析和设计方法,以及常用的各种中规模集成的组合逻辑电路,如编码器、译码器、数据分配器、数据选择器、数值比较器、奇偶校验电路和算术运算电路等,并介绍组合逻辑电路中的竞争-冒险现象及消除方法。

3.1 概述

数字系统中常用的各种数字部件按逻辑功能分为组合逻辑电路和时序逻辑电路两大类。组合逻辑电路的特点是任意时刻的输出仅取决于该时刻的输入信号组合,而与电路原来所处的状态无关。从电路结构上看,组合逻辑电路是由逻辑门组成的,无记忆元件,输出与输入之间无反馈。

任何一个多输入多输出的组合逻辑电路都可以用图 3.1.1 所示的结构框图来表示。

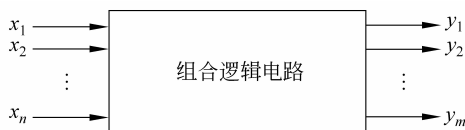


图 3.1.1 组合逻辑电路的框图

图中, $X=[x_1, x_2, \dots, x_n]$ 表示输入变量, $Y=[y_1, y_2, \dots, y_m]$ 表示输出变量。输出与输入之间可以用如下逻辑函数来描述:

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

或者写成向量函数的形式

$$Y = F(X)$$

逻辑函数表达式是表示组合逻辑电路的一种表示方法,此外,真值表、卡诺图和逻辑图等方法中的任何一种也可以表示组合逻辑电路的逻辑功能。

3.2 组合逻辑电路的分析方法和设计方法

3.2.1 组合电路的分析

组合逻辑电路的分析,即已知逻辑电路图,找出输出与输入之间的函数关系,并分析电路的逻辑功能。

组合逻辑电路的分析步骤如下:

- (1) 分别用符号标记各级门的输出端。
- (2) 从电路的输入到输出逐级写出逻辑函数式,最后得到整个电路的输出与输入关系的逻辑函数式。
- (3) 用卡诺图或公式化简法将逻辑函数化成最简形式。
- (4) 为使电路功能更加直观,列出逻辑函数真值表,分析电路逻辑功能。

【例 3.2.1】 试分析图 3.2.1 所示的电路的逻辑功能。

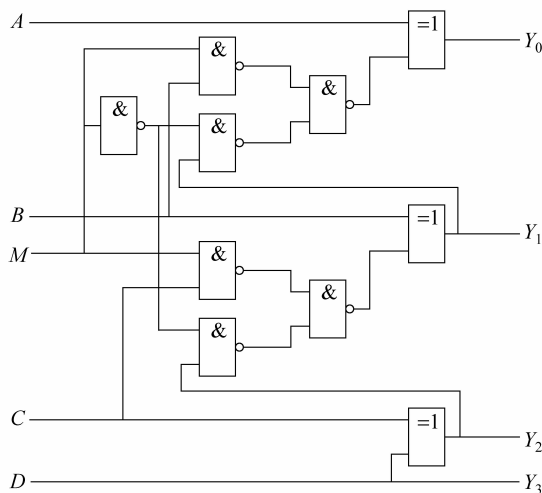


图 3.2.1 例 3.2.1 的电路

解: 根据逻辑图可写出 Y_0 、 Y_1 、 Y_2 、 Y_3 与 D 、 C 、 B 、 A 、 M 之间的逻辑函数式:

$$Y_3 = D$$

$$Y_2 = C \oplus D$$

$$Y_1 = B \oplus (MC + \overline{M}Y_2)$$

$$Y_0 = A \oplus (MB + \overline{M}Y_1)$$

当 $M=0$ 时:

$$Y_3 = D$$

$$Y_2 = C \oplus D$$

$$Y_1 = B \oplus C \oplus D$$

$$Y_0 = A \oplus B \oplus C \oplus D$$

当 $M=1$ 时：

$$Y_3 = D$$
$$Y_2 = C \oplus D$$
$$Y_1 = B \oplus C$$
$$Y_0 = A \oplus B$$

两种情况下的真值表如表 3.2.1 所示。

表 3.2.1 例 3.2.1 的真值表

M=1								M=0							
输 入				输 出				输 入				输 出			
D	C	B	A	Y ₃	Y ₂	Y ₁	Y ₀	D	C	B	A	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0
0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1
0	1	1	0	0	1	0	1	0	1	0	1	0	1	1	0
0	1	1	1	0	1	0	0	0	1	0	0	0	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0
1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	1
1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	0
1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1
1	1	0	0	1	0	1	0	1	0	1	0	1	1	0	0
1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0
1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	1

由真值表可知, $M=1$ 时完成将 8421BCD 码转换为格雷码; $M=0$ 时完成将格雷码转换为 8421BCD 码。

3.2.2 组合电路的设计

组合逻辑电路的设计与分析过程相反,其任务是根据给定的实际逻辑问题,设计出一个最简的逻辑电路图。这里所说的“最简”,是指电路中所用的器件个数最少,器件种类最少,而且器件之间的连线最少。

组合逻辑电路设计的一般步骤如下：

- (1) 根据设计题目要求,进行逻辑抽象,确定输入变量和输出变量及数目,明确输出变量和输入变量之间的逻辑关系。
- (2) 将输出变量和输入变量之间的逻辑关系(或因果关系)列成真值表。
- (3) 根据真值表写出逻辑函数,并用公式法和卡诺图方法将逻辑函数化简成最简表

达式。

(4) 选用小规模集成逻辑门电路、中规模的常用集成组合逻辑电路或可编程逻辑器件构成相应的逻辑函数。具体如何选择,应根据电路的具体要求和器件的资源情况来决定。

(5) 根据选择的器件,将逻辑函数转换成适当的形式。

① 在使用小规模集成门电路进行设计时,为获得最简单的设计结果,应把逻辑函数转换成最简形式,即器件数目和种类最少。因此通常把逻辑函数转换为与非-与非式或者与或非式,这样可以用与非门或者与或非门来实现。

② 在使用中规模组合逻辑电路设计电路时,需要将逻辑函数化成常用组合逻辑电路的逻辑函数式形式,具体做法将在下一节介绍。

③ 如果使用存储器或可编程逻辑器件来实现,具体做法将在第 7 章介绍。

(6) 根据化简或变换后的逻辑函数式,画出逻辑电路的逻辑图。

【例 3.2.2】 试设计一个供三人使用的表决逻辑电路。即三个人中,有两个或三个人表示同意,则表决通过,否则为不通过。

解: (1) 进行逻辑抽象。

用 A 、 B 、 C 表示每个人的表决结果,用 Y 表示三人的表决结果。因此, A 、 B 、 C 为输入逻辑变量, Y 为输出逻辑变量。用 1 表示表决人同意或表决通过,0 表示表决人不同意或表决不通过。

根据题意列出表 3.2.2 所示的逻辑真值表。

表 3.2.2 例 3.2.2 的逻辑真值表

输 入				输出	输 入				输出	输 入				输出	输 入				输出
A	B	C	Y		A	B	C	Y		A	B	C	Y		A	B	C	Y	
0	0	0	0		0	1	0	0		1	0	0	0		1	1	0	1	
0	0	1	0		0	1	1	1		1	0	1	1		1	1	1	1	

(2) 根据真值表,画出三变量逻辑函数卡诺图,如图 3.2.2 所示,化简后得到 Y 的逻辑函数表达式。

$$Y = AB + BC + AC \quad (3.2.1)$$

(3) 选定器件类型为小规模集成门电路。

(4) 根据式(3.2.1)画出逻辑电路图,得到图 3.2.3 电路。这里用到与门和或门。若用其他类型门电路来组成这个逻辑电路,应将最简与-或式化成相应的形式。

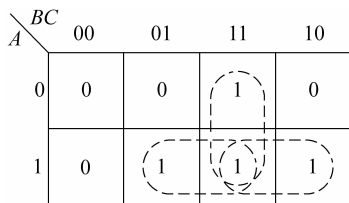


图 3.2.2 例 3.2.2 的卡诺图一

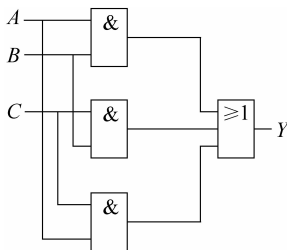


图 3.2.3 例 3.2.2 的逻辑图一

例如,要求用与非门实现这个逻辑电路时,应当将逻辑函数化成与非-与非表达式。

$$Y = \bar{Y} = \overline{AB + BC + AC} = \overline{AB} \cdot \overline{BC} \cdot \overline{AC} \tag{3.2.2}$$

根据式(3.2.2)可以全部用与非门实现的逻辑电路图,如图 3.2.4 所示。

如果用与或非门实现这个逻辑电路,必须把式(3.2.2)化成最简的与-或-非表达式。在第 1 章讲过,可以圈卡诺图中的 0,然后求反而得到。圈 0 卡诺图如图 3.2.5 所示,得到式(3.2.3)的与-或-非式为

$$Y = \overline{AB + AC + BC} \tag{3.2.3}$$

按照式(3.2.3)画出用与-或-非门组成的逻辑电路如图 3.2.6 所示。

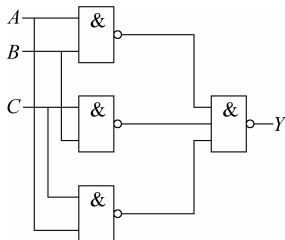


图 3.2.4 例 3.2.2 的逻辑图二

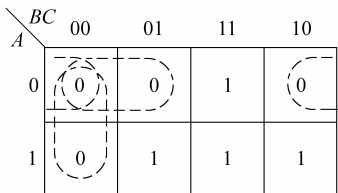


图 3.2.5 例 3.2.2 的卡诺图二

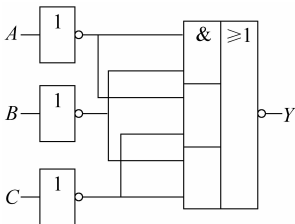


图 3.2.6 例 3.2.2 的逻辑图三

3.3 编码器和译码器

3.3.1 编码器

从广义上讲,将具有特定意义的信息(如文字、符号或数字)赋予相应的二进制代码的过程称为编码。用来实现编码操作的电路叫做编码器。根据被编码信号的不同特点和要求,编码器分为二进制编码器、二-十进制编码器和优先编码器等。

1. 二进制编码器

用 n 位二进制代码对 2^n 个信号进行编码的电路称为二进制编码器。对 N 个信号进行编码时,可用 $2^n \geq N$ 来确定需要使用的二进制代码的位数 n 。下面以 3 位二进制编码器为例说明二进制编码器的工作原理和设计过程。

3 位二进制编码器是将 $I_0、I_1、\cdots、I_7$ 共 8 个输入信号编成二进制代码。由于 $2^3 = 8$,输出可用 3 位二进制代码进行编码,所以该编码器有 8 个输入变量和 3 个输出变量,因此 3 位二进制编码器也称为 8 线-3 线编码器,如图 3.3.1 所示。

在某一时刻,该编码器只能对一个输入信号进行编码,即在编码器输入端,同一时刻不允许两个或两个以上的输入信号同时出现,也就是 $I_0、I_1、\cdots、I_7$ 互相排斥,所以该编码器真值表如表 3.3.1 所示。

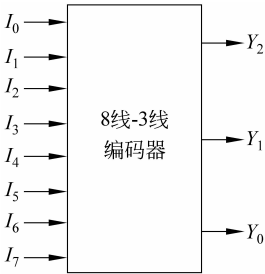


图 3.3.1 8 线-3 线编码器

表 3.3.1 8 线-3 线编码器真值表

输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_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

根据真值表和 $I_0, I_1, \dots, I_7$ 互相排斥的约束条件, 只要将使输出值为 1 的输入变量直接相加, 即可得到输出的最简与-或表达式。

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

$$Y_2 = \overline{I_4} \overline{I_5} \overline{I_6} \overline{I_7}$$

$$\text{与非-与非式 } Y_1 = \overline{I_2} \overline{I_3} \overline{I_6} \overline{I_7}$$

$$Y_0 = \overline{I_1} \overline{I_3} \overline{I_5} \overline{I_7}$$

图 3.3.2 是用与非门实现的 8 线-3 线编码器。当 $I_1 \sim I_7$ 均为 0 时, 电路输出为 I_0 的编码。

2. 二进制优先编码器

在实际应用中, 常常出现几个输入端同时加输入信号时, 编码器能够按照一定的优先次序, 对优先级最高的输入信号进行编码, 而不理睬级别低的信号。这种根据优先顺序进行编码的电路称为优先编码器。

图 3.3.3 给出了 8 线-3 线优先编码器(74LS148)的逻辑电路图、逻辑符号及外引线排列图。其真值表如表 3.3.2 所示。

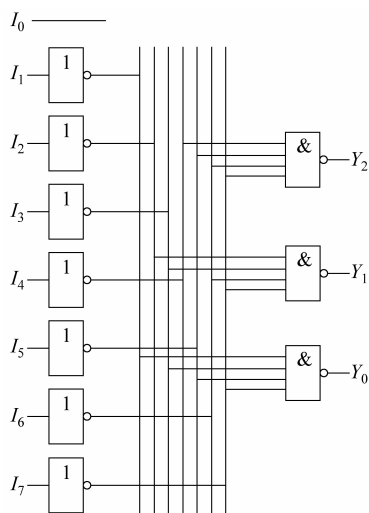


图 3.3.2 与非门实现 8 线-3 线编码器

表 3.3.2 8 线-3 线优先编码器真值表

输 入									输 出				
$\bar{S}$	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0	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	0	1	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1

续表

输 入									输 出				
$\bar{S}$	$\bar{I}_0$	$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$	$\bar{Y}_{EX}$	$\bar{Y}_S$
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

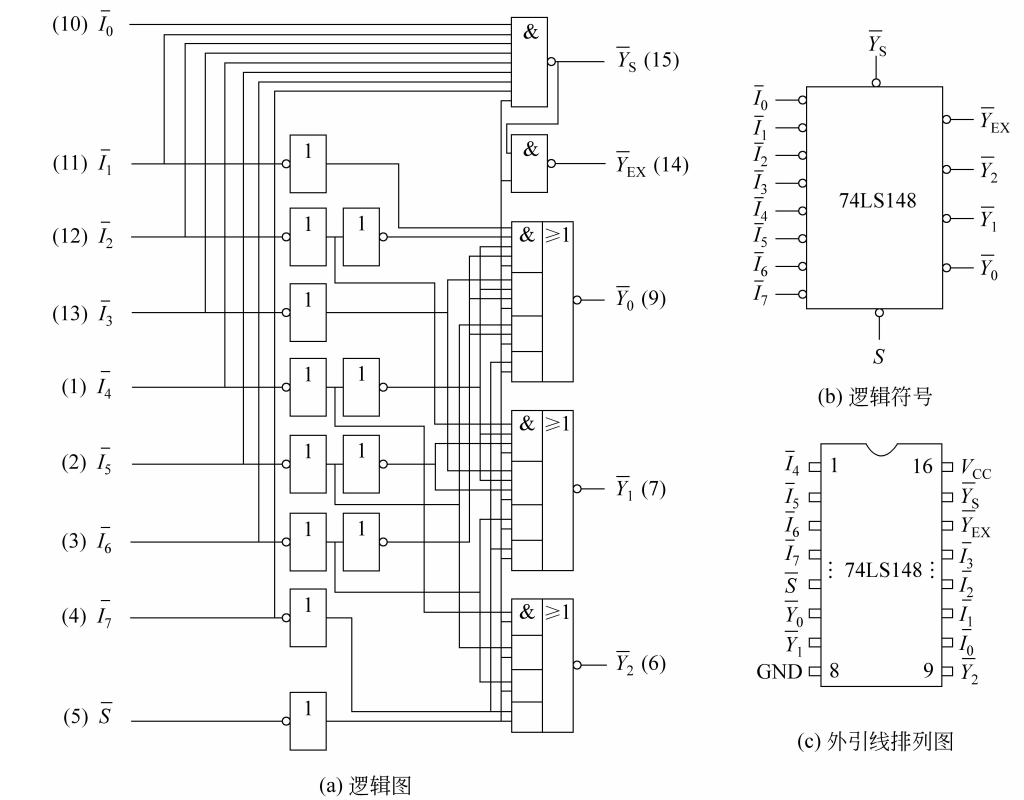


图 3.3.3 8 线-3 线优先编码器(74LS148)

由逻辑电路图可得到输出表达式为

$$\begin{aligned}\bar{Y}_2 &= \overline{(I_4 + I_5 + I_6 + I_7)S} \\ \bar{Y}_1 &= \overline{(I_7 + I_6 + I_3\bar{I}_4\bar{I}_5 + I_2\bar{I}_4\bar{I}_5)S} \\ \bar{Y}_0 &= \overline{(I_7 + I_5\bar{I}_6 + I_3\bar{I}_4\bar{I}_6 + I_1\bar{I}_2\bar{I}_4\bar{I}_6)S} \\ \bar{Y}_S &= \overline{\bar{I}_0\bar{I}_1\bar{I}_2\bar{I}_3\bar{I}_4\bar{I}_5\bar{I}_6\bar{I}_7S} \\ \bar{Y}_{EX} &= \overline{(I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7)S}\end{aligned}$$

由逻辑图和逻辑符号可以看出， $\bar{I}_0 \sim \bar{I}_7$ 为输入信号， $\bar{Y}_2$ 、 $\bar{Y}_1$ 和 $\bar{Y}_0$ 为输出，低电平有效。为了便于扩展，8 线-3 线优先编码器还设置了 3 个附加控制端： $\bar{S}$ 为选通输入端，低电平有效， $\bar{Y}_S$ 为选通输出端， $\bar{Y}_{EX}$ 为优先扩展输出端。

由表 3.3.2 所示的真值表可以看出， $\bar{S}=1$ 时编码器不工作，编码器输出全为 1。

只有 $\bar{S}=0$ 时编码器正常工作,又分两种情况:

(1) 所有输入全为 1 时,即无输入信号要求编码,编码器输出 $\bar{Y}_2\bar{Y}_1\bar{Y}_0=111$ 且 $\bar{Y}_{EX}=1, Y_S=0$ 。

(2) 当 $\bar{I}_0\sim\bar{I}_7$ 至少有一个为 0 时,编码器按输入的优先级进行编码, $\bar{I}_7$ 的优先权最高, $\bar{I}_6$ 次之, $\bar{I}_0$ 最低。例如,当 $\bar{I}_7\bar{I}_6=11$, 而 $\bar{I}_5=0$ 时,不管其他输入为何值,都对 $\bar{I}_5$ 进行编码,输出 $\bar{Y}_2\bar{Y}_1\bar{Y}_0=010$, 而 $\bar{Y}_{EX}=0, \bar{Y}_S=1$ 。

一片 8 线-3 线优先编码器 74LS148 只具有 8 级优先编码功能,利用选通输入端 $\bar{S}$ 、选通输出端 $\bar{Y}_S$ 和优先扩展输出端 $\bar{Y}_{EX}$,可以实现多级优先编码。下面结合一个例子说明 $\bar{Y}_{EX}$ 和 $\bar{Y}_S$ 信号实现电路扩展的方法。

【例 3.3.1】 试用两片 74LS148 实现 16 线-4 线优先编码器。要求 16 个输入端为 $\bar{A}_{15}\sim\bar{A}_0$, 4 个输出端为 $Z_3\sim Z_0$, 其中 $\bar{A}_{15}$ 优先权最高, $\bar{A}_0$ 优先权最低。

解: 将两片 74LS148 串行连接,如图 3.3.4 所示。片 1 的输入端作为低 8 位 $\bar{A}_7\sim\bar{A}_0$ 输入,片 1 输出 $\bar{Y}_S$ 作为 16 线-4 线优先编码器的选通输入端;片 2 的输入端作为高 8 位 $\bar{A}_{15}\sim\bar{A}_8$ 输入,片 2 的输出 $\bar{Y}_S$ 接片 1 的 $\bar{S}$ 输入端,片 2 的 $\bar{Y}_{EX}$ 经反相器输出作为 Z_3 ;两片的 $\bar{Y}_2, \bar{Y}_1, \bar{Y}_0$ 经与非门输出 Z_2, Z_1, Z_0 , 两片的 $\bar{Y}_{EX}$ 经与非门输出作为电路总的优先扩展输出端 Z_{EX} , 片 2 的 $\bar{S}$ 端接地。当片 2 输入 $\bar{A}_9=0$ 时,则片 2 的 $\bar{Y}_{EX}=0, \bar{Y}_S=1$, 片 2 进行编码,由于片 1 的 $\bar{S}=1$, 不进行编码,电路输出 $Z_3Z_2Z_1Z_0=1001$ 。当片 2 输入全为 1 时,即 $\bar{A}_{15}\sim\bar{A}_8$ 没有编码输入信号,片 2 的输出 $\bar{Y}_2\bar{Y}_1\bar{Y}_0=111, \bar{Y}_{EX}=1, \bar{Y}_S=0$, 使片 1 的 $\bar{S}=0$, 片 1 可以进行编码。若 $\bar{A}_6=0$, 则片 1 的 $\bar{Y}_2\bar{Y}_1\bar{Y}_0=001$, 电路的总输出 $Z_3Z_2Z_1Z_0=0110$ 。

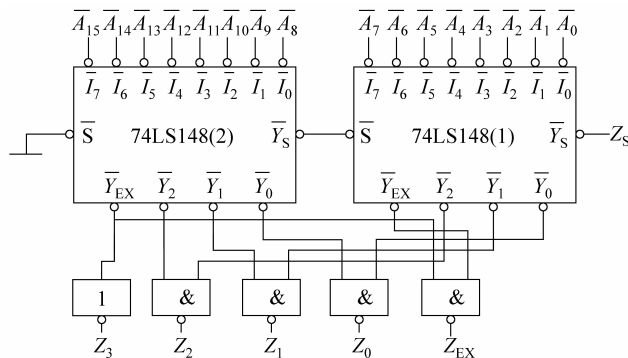


图 3.3.4 16 线-4 线优先编码器

3. BCD 码优先编码器

将 $\bar{I}_0\sim\bar{I}_9$ 10 个信号编成 10 个 BCD 码,其中 10 个输入信号中 $\bar{I}_9$ 的优先权最高, $\bar{I}_0$ 的优先权最低。图 3.3.5 是 BCD 码优先编码器(74LS147)逻辑电路图、逻辑符号及外引线排列图。

由逻辑电路图可得到输出表达式为

$$\bar{Y}_3 = \bar{I}_8 + \bar{I}_9$$

$$\bar{Y}_2 = \bar{I}_4\bar{I}_8\bar{I}_9 + \bar{I}_5\bar{I}_8\bar{I}_9 + \bar{I}_6\bar{I}_8\bar{I}_9 + \bar{I}_7\bar{I}_8\bar{I}_9$$

$$\bar{Y}_1 = \bar{I}_2\bar{I}_4\bar{I}_5\bar{I}_8\bar{I}_9 + \bar{I}_3\bar{I}_4\bar{I}_5\bar{I}_8\bar{I}_9 + \bar{I}_6\bar{I}_8\bar{I}_9 + \bar{I}_7\bar{I}_8\bar{I}_9$$

$$\bar{Y}_0 = \bar{I}_1\bar{I}_2\bar{I}_4\bar{I}_5\bar{I}_8\bar{I}_9 + \bar{I}_3\bar{I}_4\bar{I}_5\bar{I}_8\bar{I}_9 + \bar{I}_5\bar{I}_6\bar{I}_8\bar{I}_9 + \bar{I}_7\bar{I}_8\bar{I}_9 + \bar{I}_9$$

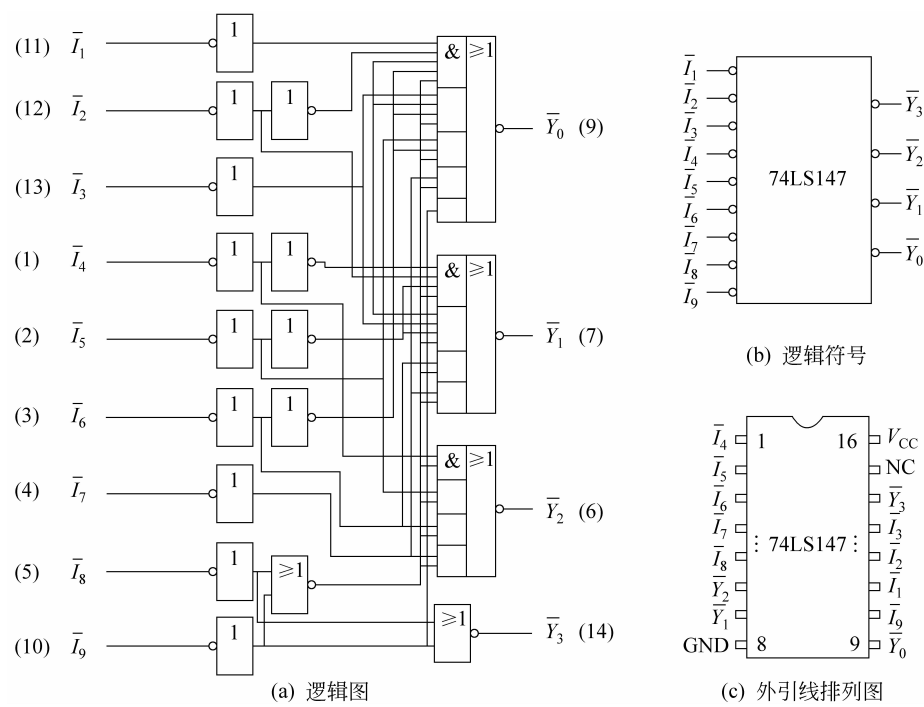


图 3.3.5 BCD 码优先编码器(74LS147)

其真值表如表 3.3.3 所示。从表中可以看出,编码器的输出以 BCD 码的反码形式给出。

表 3.3.3 BCD 码优先编码器真值表

输 入										输 出			
$\bar{I}_0$	$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{I}_8$	$\bar{I}_9$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$	$\bar{Y}_0$
×	×	×	×	×	×	×	×	×	0	0	1	1	0
×	×	×	×	×	×	×	×	0	1	0	1	1	1
×	×	×	×	×	×	×	0	1	1	1	0	0	0
×	×	×	×	×	×	0	1	1	1	1	0	0	1
×	×	×	×	×	0	1	1	1	1	1	0	1	0
×	×	×	×	0	1	1	1	1	1	1	0	1	1
×	×	×	0	1	1	1	1	1	1	1	1	0	0
×	×	0	1	1	1	1	1	1	1	1	1	0	1
×	0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1

3.3.2 译码器

编码的逆过程(把表示特定意义的信息代码翻译出来的过程)称为译码,实现译码操作的电路称为译码器。译码器分为 3 类,一是二进制译码器,也称最小项译码器,有 3 线-

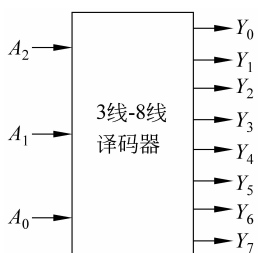


图 3.3.6 3线-8线译码器

8线、4线-16线译码器等；二是码制转换译码器，有8421BCD码转换十进制译码器、余3码转换十进制译码器等；三是显示译码器，用来驱动各类显示器，如发光二极管、液晶数码管等。

1. 二进制译码器

二进制译码器输入为二进制码，输出为与输入代码一一对应的高、低电平信号。三位二进制译码器示意图如图3.3.6所示。

真值表列于表3.3.4中， $\bar{Y}_0 \sim \bar{Y}_7$ 是相互独立的8个信号，分别对应于 A_2 、 A_1 、 A_0 的8种组合状态，输出低电平有效。

表 3.3.4 3线-8线译码器真值表

输 入			输 出							
A_2	A_1	A_0	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$
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

由表可直接得出，每个输出函数是输入变量的一个最小项为：

$$\begin{aligned}\bar{Y}_0 &= \overline{A_2 A_1 A_0} & \bar{Y}_1 &= \overline{A_2 \bar{A}_1 A_0} \\ \bar{Y}_2 &= \overline{A_2 A_1 \bar{A}_0} & \bar{Y}_3 &= \overline{A_2 \bar{A}_1 \bar{A}_0} \\ \bar{Y}_4 &= \overline{A_2 \bar{A}_1 A_0} & \bar{Y}_5 &= \overline{A_2 A_1 A_0} \\ \bar{Y}_6 &= \overline{A_2 A_1 \bar{A}_0} & \bar{Y}_7 &= \overline{A_2 \bar{A}_1 A_0}\end{aligned}$$

由表达式可以看出，对应每个输入状态，仅有一个输出为0，其余为1。例如 $A_2 A_1 A_0 = 101$ 时，仅 $\bar{Y}_5 = 0$ ，即 $\bar{Y}_5$ 是输入二进制码101的译码输出，所以这种译码器也称为最小项译码器。

图3.3.7(a)是用与非门实现的3线-8线译码器74LS138。输入为3位二进制数 A_2 、 A_1 、 A_0 ，输出有 $\bar{Y}_0 \sim \bar{Y}_7$ 共8个信号，分别对应输入的共8种组合。另外，74LS138有3个附加控制端 S_1 、 $\bar{S}_2$ 和 $\bar{S}_3$ ，只有当 $S_1 = 1$ 、 $\bar{S}_2 = \bar{S}_3 = 0$ 时，译码器处于工作状态；否则，译码器不实现译码，也就是说不管输入为任何值，8个输出信号均为1，如表3.3.5所示。

表 3.3.5 3线-8线译码器74LS138真值表

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
0	×	×	×	×	1	1	1	1	1	1	1	1

续表

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	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

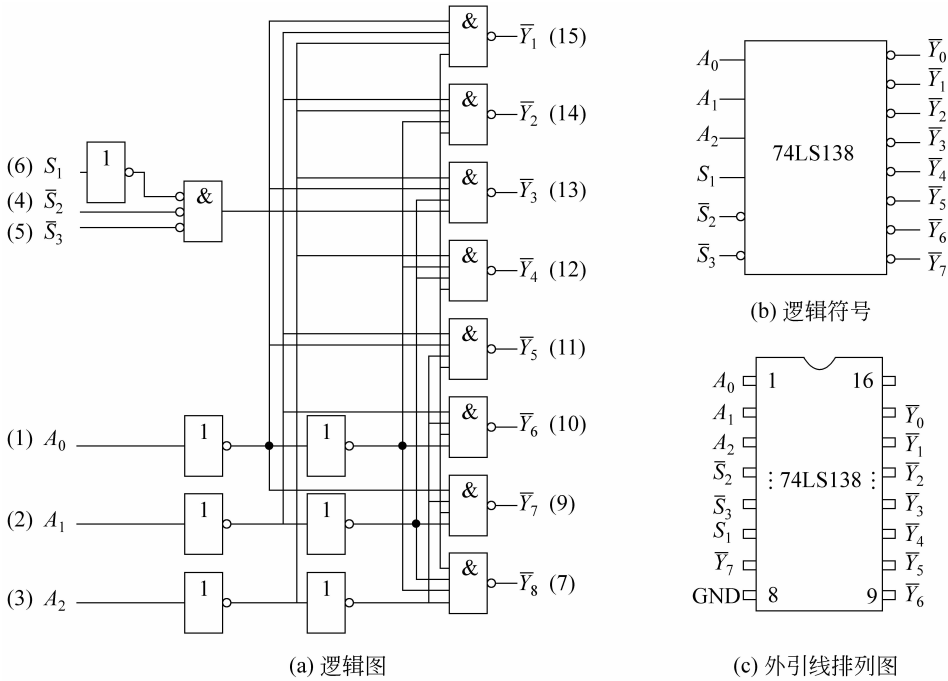


图 3.3.7 3 线-8 线译码器 (74LS138)

附加控制端 S_1 、 $\bar{S}_2$ 和 $\bar{S}_3$ 也叫做片选输入端,利用片选的作用可以将多片连接起来以扩展译码器的功能。图 3.3.7(b)、(c)分别为 74LS138 的逻辑符号和外引线排列图。

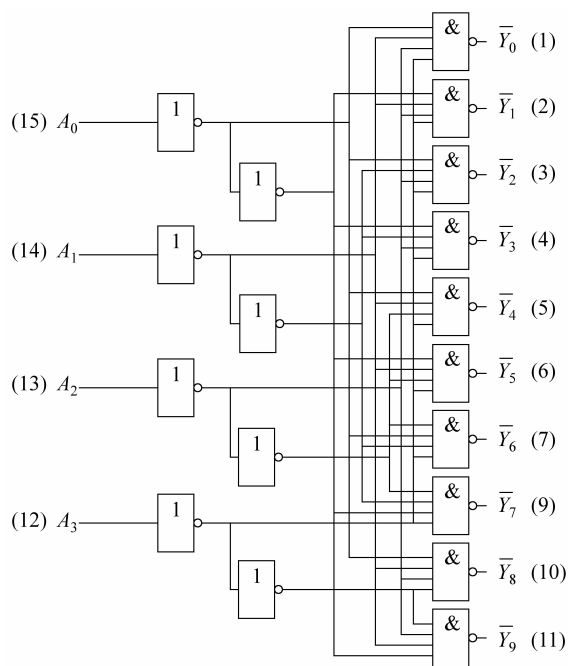
2. 二-十进制译码器

二-十进制译码器是码制转换译码器的一种,将输入 BCD 码的 10 个代码翻译成十进制代码 0~9 的逻辑电路。

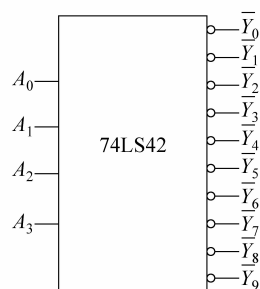
图 3.3.8(a)是二-十进制译码器 74LS42 的逻辑电路图,图 3.3.8(b)、(c)分别为 74LS42 的逻辑符号和外引线排列图。其真值表如表 3.3.6 所示。

表 3.3.6 二-十进制译码器 74LS42 真值表

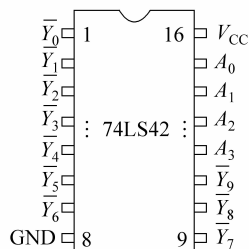
输 入				输 出									
A_3	A_2	A_1	A_0	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$	$\bar{Y}_8$	$\bar{Y}_9$
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	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
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1



(a) 逻辑图



(b) 逻辑符号



(c) 外引线排列图

图 3.3.8 二-十进制译码器(74LS42)

根据真值表可写出

$$\begin{aligned}
\bar{Y}_0 &= \overline{\bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0} & \bar{Y}_1 &= \overline{\bar{A}_3 \bar{A}_2 \bar{A}_1 A_0} \\
\bar{Y}_2 &= \overline{\bar{A}_3 \bar{A}_2 A_1 \bar{A}_0} & \bar{Y}_3 &= \overline{\bar{A}_3 \bar{A}_2 A_1 A_0} \\
\bar{Y}_4 &= \overline{\bar{A}_3 A_2 \bar{A}_1 \bar{A}_0} & \bar{Y}_5 &= \overline{\bar{A}_3 A_2 \bar{A}_1 A_0} \\
\bar{Y}_6 &= \overline{\bar{A}_3 A_2 A_1 \bar{A}_0} & \bar{Y}_7 &= \overline{\bar{A}_3 A_2 A_1 A_0} \\
\bar{Y}_8 &= \overline{A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0} & \bar{Y}_9 &= \overline{A_3 \bar{A}_2 \bar{A}_1 A_0}
\end{aligned}$$

4 个输入端为 A_3 、 A_2 、 A_1 和 A_0 ，用于输入 8421BCD 码，10 个输出端 $\bar{Y}_0 \sim \bar{Y}_9$ 对应十进制数 0~9，输出低电平有效。对应 BCD 码以外的伪码（即 1010~1111 共 6 个代码）作为输入时，译码器拒绝翻译，输出均无低电平，所以这个电路具有拒绝伪码的功能。

3. 显示译码器

在数字系统中，经常需要将数字、文字和符号的二进制编码翻译成人们习惯的形式直观地显示出来，供人们读取或监视系统的工作情况。能够把二进制代码翻译并显示出来的电路叫做显示译码器，它包括译码驱动电路和数码显示器两部分。

1) 数码显示器

(1) 显示器分类及特点。

常用的数码显示器有两种，一种是发光二极管(LED)显示器，其优点是清晰悦目、工作电压低(1.5~3V)、体积小、寿命长、可靠性高等，而且响应时间短(1~100ns)，颜色丰富(有红、绿、黄等颜色)，亮度高。其缺点是工作电流比较大，每段的工作电流在 10mA 左右。

另一种是液晶(LCD)显示器，其特点是驱动电压低(在 1V 以下可以工作)、工作电流非常小、功耗极小(1μW 以下)，配合 CMOS 电路可以组成微功耗系统。其缺点是亮度差、响应速度低(在 10~200ms 范围)，这限制了它在快速系统中的应用。

最常用的显示译码器是能驱动七段数码管的 BCD 七段显示译码器。下面主要以发光二极管作为显示器件的七段数码显示器。

(2) LED 显示器。

发光二极管使用的材料与普通的硅二极管和锗二极管不同，有磷砷化镓、磷化镓或砷化镓等几种，而且杂质浓度很高。当 PN 结外加正向电压时，P 区多数载流子空穴和 N 区多数载流子电子在扩散过程中复合，同时释放出能量，发出一定波长的光。发光二极管发出的光线波长与磷和砷的比例有关，含磷的比例越大波长越短，同时发光效率越低。目前生产的磷砷化镓发光二极管(如 BS201、BS211 等)发出的光线波长在 6500Å 左右，呈橙红色。

七段半导体数码管是将七个发光二极管按一定的方式连接在一起，每段为一个发光二极管，七段分别为 a、b、c、d、e、f、g，显示某个字形时，相应段的发光二极管发光。在 BS201 等一些数码管的右下角增设一个小数点，其形成所谓的八段数码管，其外形图如图 3.3.9(a)所示。

按连接方式不同，八段半导体数码管分为共阴极和共阳极两种。共阴极是指 BS201 八段发光二极管的阴极连接在一起，每个发光二极管的阳极经限流电阻接到显示译码器输出端(译码器输出高电平有效)，如图 3.3.9(b)所示。而共阳极是指 BS201 八段发光二极管的阳极连接在一起，每个发光二极管的阴极经限流电阻接到显示译码器输出端(译码器输出低电平有效)，如图 3.3.9(c)所示。改变限流电阻大小，可改变二极管中的电流大小，从而控制发光亮度。

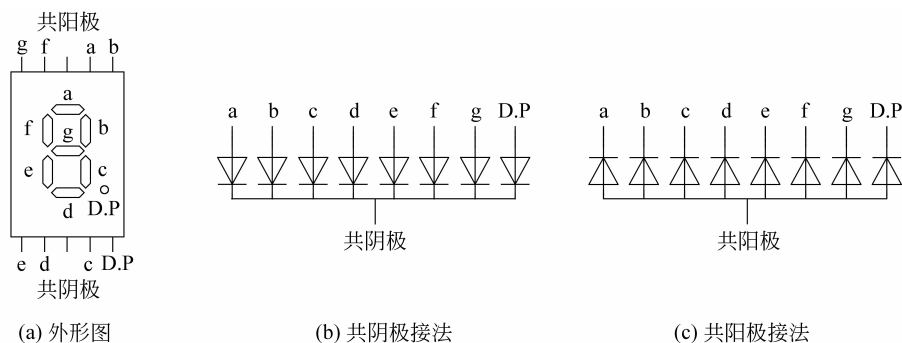


图 3.3.9 八段半导体数码管

2) BCD 七段显示译码器

半导体数码管和液晶显示器都可以用 TTL 或 CMOS 集成电路直接驱动。为了使七段数码管显示 0~9 十个数字,需要使用 BCD 七段译码器将 BCD 码翻译成数码管所要求的驱动信号。中规模 BCD 七段译码器的种类很多,下面以配合半导体数码管 BS201A 工作的 4 线-7 线译码器/驱动器 7448 为例加以介绍。

以 $A_3A_2A_1A_0$ 表示显示译码器的输入的 BCD 码,以 $Y_a \sim Y_g$ 表示七段半导体数码管的驱动信号。假设译码器输出高电平有效,即输出为 1 时相应段的发光二极管发光。按照图 3.3.9(a)所示字形,可列出显示译码器真值表,如表 3.3.7 所示。表中除列出了 BCD 码的 10 个状态,还规定了 1010~1111 这 6 个状态的显示字形。

表 3.3.7 BCD 七段显示译码器真值表

输 入					输 出							字形
序号	A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	0	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	0	0	1	1	9
10	1	0	1	0	0	0	0	1	1	0	1	C
11	1	0	1	1	0	0	1	1	0	0	1	3
12	1	1	0	0	0	1	0	0	0	1	1	U
13	1	1	0	1	1	0	0	1	0	1	1	E
14	1	1	1	0	0	0	0	1	1	1	1	7
15	1	1	1	1	0	0	0	0	0	0	0	

从得到的真值表,用卡诺图化简可得到 $Y_a \sim Y_g$ 的函数表达式:

$$Y_a = \overline{A_3}\overline{A_2}\overline{A_1}A_0 + A_2\overline{A_0} + A_3A_1$$

$$Y_b = \overline{A_2}\overline{A_1}A_0 + A_2A_1\overline{A_0} + A_3A_1$$

$$Y_c = \overline{A_2}A_1\overline{A_0} + A_3A_2$$

$$Y_d = \overline{A_2}\overline{A_1}A_0 + A_2A_1A_0 + A_2\overline{A_1}\overline{A_0}$$

$$Y_e = A_2\overline{A_1} + A_0$$

$$Y_f = A_1A_0 + \overline{A_3}\overline{A_2}A_0 + \overline{A_2}A_1$$

$$Y_g = \overline{A_2}A_1A_0 + \overline{A_3}\overline{A_2}\overline{A_1}$$

根据逻辑函数式可画出 BCD 七段显示译码器 7448 的逻辑电路图,如图 3.3.10(a)所示,图 3.3.10(b)和图 3.3.10(c)分别为 7448 的逻辑符号和外部引线排列图。

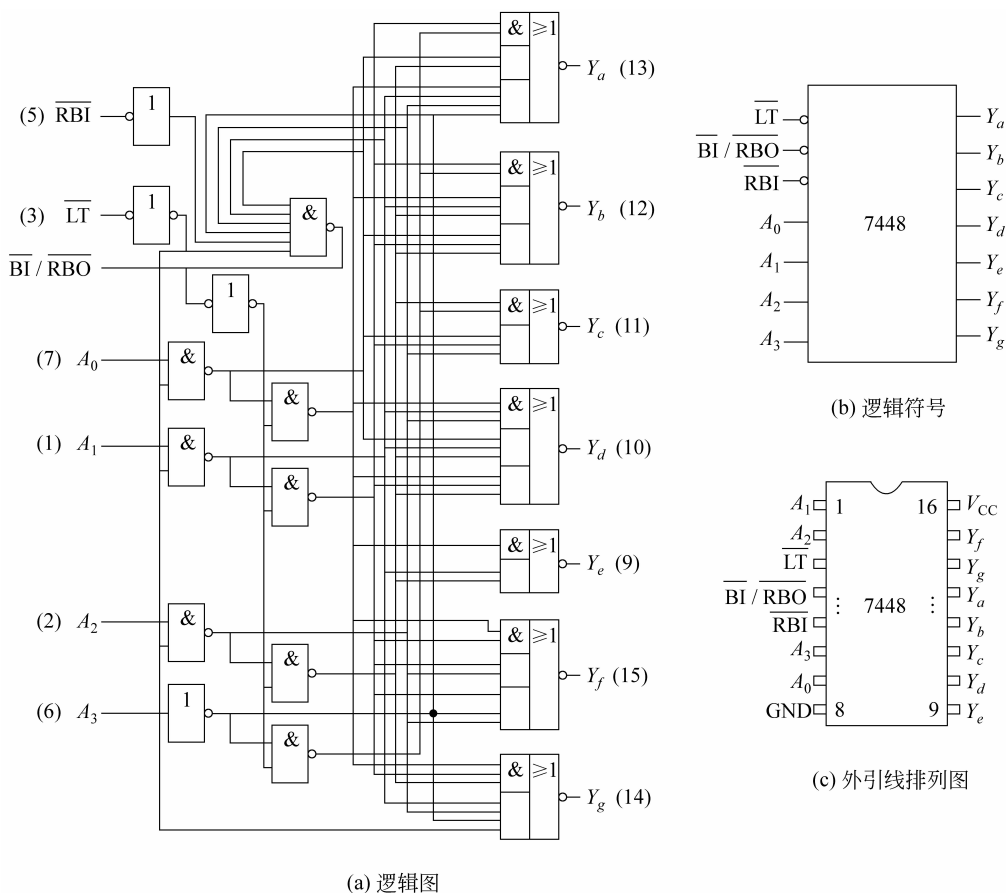


图 3.3.10 BCD 七段显示译码器 7448

另外,7448 逻辑电路中增加了附加控制电路。下面介绍其功能和用法。

(1) 灯测试输入端 $\overline{LT}$ 。

当 $\overline{LT}=0$ 时, $\overline{Y_a} \sim \overline{Y_g}$ 均输出高电平,七段半导体数码管全部点亮,显示 8 字形,用来测试数码管的好坏。当 $\overline{LT}=1$ 时显示译码器按输入 BCD 码正常显示。

(2) 灭零输入端 $\overline{\text{RBI}}$ 。

当 $\overline{\text{RBI}}=0$ 时,若输入端 $A_3A_2A_1A_0=0000$,则 $\overline{Y}_a \sim \overline{Y}_g$ 均输出低电平,实现灭零;若输入端为其他的 BCD 码,则正常显示。设置灭零输入端 $\overline{\text{RBI}}$ 的目的是为了把不希望显示的零熄灭。例如有一个 4 位的数码管显示电路显示“03.40”时前后两位的 0 是多余的,可以在对应位的灭零输入端加入灭零信号,即使 $\overline{\text{RBI}}=0$,则只显示出“3.4”。对不需要灭零的位则应使 $\overline{\text{RBI}}=1$ 。

(3) 灭灯输入/灭零输出端 $\overline{\text{BI}}/\overline{\text{RBO}}$ 。

当 $\overline{\text{BI}}/\overline{\text{RBO}}$ 作为输入端使用时,称为灭灯输入端。若 $\overline{\text{BI}}=0$,则无论输入为何种状态, $\overline{Y}_a \sim \overline{Y}_g$ 输出均为 0,七段半导体数码管全部熄灭,可用来控制是否显示。若 $\overline{\text{BI}}=1$ 时,正常译码显示。当 $\overline{\text{BI}}/\overline{\text{RBO}}$ 作为输出端使用时,称为灭零输出端,其表达式为

$$\overline{\text{RBO}} = \overline{\text{LT}} A_3 A_2 A_1 A_0 \overline{\text{RBI}}$$

由此可知,当 $A_3A_2A_1A_0=0000$ 而且有灭零输入信号 ($\overline{\text{RBI}}=0$) 和 $\overline{\text{LT}}=1$ 时, $\overline{\text{RBO}}=0$,该信号既可以使本位灭零 ($\overline{\text{RBI}}=0$),又同时输出低电平信号 ($\overline{\text{RBO}}=0$),为相邻位灭零提供条件。这样可以消去多位数显示中前后不必要的零。

用 7448 可以直接驱动半导体数码管 BS201,其接线图如图 3.3.11 所示。图中流过发光二极管的电流由电源电压经上拉电阻 R 提供,选取合适的电阻值使电流大于数码管所需要的电流。

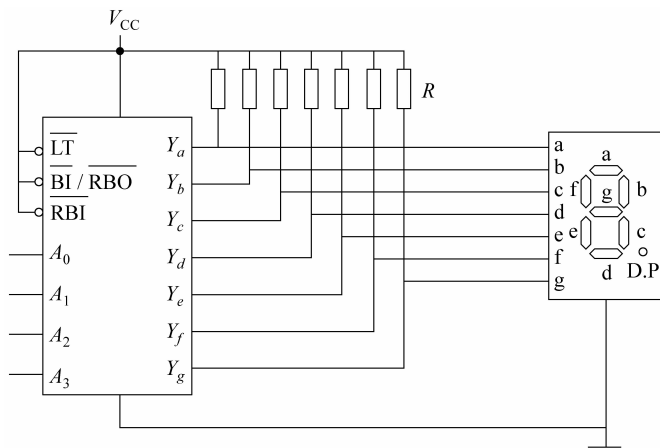


图 3.3.11 7448 驱动七段数码管接线图

4. 译码器的应用举例

1) 3 线-8 线译码器 74LS138 应用

【例 3.3.2】 利用两片 74LS138 组成 4 线-16 线译码器,将 4 位输入的二进制代码 $A、B、C、D$ 译成 16 个独立的低电平信号 $\overline{Z}_0 \sim \overline{Z}_{15}$,其中 A 为最高位。

解: 由 74LS138 逻辑符号可知,译码器有 3 个地址输入端 $A_2、A_1$ 和 A_0 可作为 4 线-16 线译码器低 3 位 $B、C、D$,再利用 3 个附加控制端 $S_1、\overline{S}_2$ 和 $\overline{S}_3$ 进行合理组合,构成第 4 个地址输入端 A ,图 3.3.12 给出了两片 74LS138 扩展成 4 线-16 线译码器的电路。

当 $A=0$ 时,片 1 的 $\overline{S}_2=0$ 允许译码,其输出取决于输入变量 $B、C、D$;而片 2 的 $S_1=$

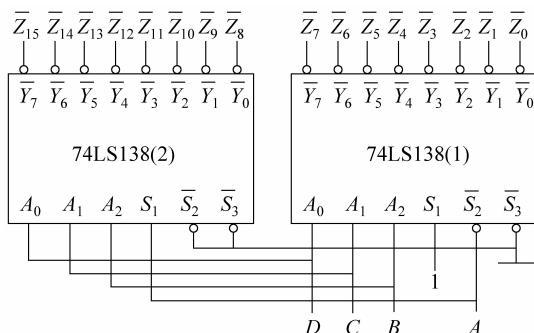


图 3.3.12 例 3.3.2 的电路(4 线-16 线译码器)

0 禁止译码,其输出均为 1。当 $A=1$ 时,片 1 的 $\bar{S}_2=1$ 禁止译码,其输出均为 1;而片 2 的 $S_1=1$ 允许译码,其输出由 B, C, D 决定。这样接成的 4 线-16 线译码器电路 3 个附加控制端分别为片 1 的 S_1 ,片 2 的 $\bar{S}_2, \bar{S}_3$,利用它们可以再接成 5 线-32 线译码器。

【例 3.3.3】 试画出用 74LS138 和门电路实现如下多输出逻辑函数。

$$\begin{cases} Z_1 = AC \\ Z_2 = \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB \\ Z_3 = ABC\bar{C} + BC \end{cases}$$

解: 首先将逻辑函数化成最小项之和形式,得到

$$\begin{cases} Z_1 = AC = ABC + A\bar{B}C = m_5 + m_7 \\ Z_2 = \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB = \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC + A\bar{B}C = m_1 + m_4 + m_6 + m_7 \\ Z_3 = ABC\bar{C} + BC = ABC\bar{C} + ABC\bar{C} + A\bar{B}C + \bar{A}BC = m_2 + m_6 \end{cases}$$

将上式整理成与非-与非形式:

$$\begin{cases} Z_1 = \overline{\bar{m}_5 \bar{m}_7} \\ Z_2 = \overline{\bar{m}_1 \bar{m}_4 \bar{m}_6 \bar{m}_7} \\ Z_3 = \overline{\bar{m}_2 \bar{m}_6} \end{cases}$$

由于 74LS138 为最小项译码器,每一个输出 $\bar{Y}_i = \bar{m}_i$,因此只要在输出端增加三个与非门即可实现 $Z_1 \sim Z_3$ 的逻辑电路,如图 3.3.13 所示。

另外,在微机系统中,译码器常用作存储器或输入、输出接口芯片的地址译码器。 n 位地址线可以寻址 2^n 个存储单元。

2) 显示译码器 7448 灭零功能的应用

【例 3.3.4】 试用译码驱动电路 7448 和数码管实现多位数码显示系统。

解: 将灭零输入端和灭零输出端配合使用,可以实现多位数码显示器整数前和小数后的灭零控制,其连接方法如图 3.3.14 所示。图中接法如下:整数部分的高位 $\overline{RBO}$ 和低位的 $\overline{RBI}$ 相连,最高位 $\overline{RBI}$ 接 0;小数部分的低位 $\overline{RBO}$ 和高位的 $\overline{RBI}$ 相连,最低位 $\overline{RBI}$ 接 0,最高位 $\overline{RBI}$ 接 1;小数点位 $\overline{RBI}$ 接 1。这样整数部分只有高位为 0,而且被熄灭的情况下,低位

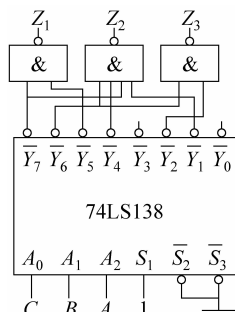


图 3.3.13 例 3.3.3 的电路

才有灭零输入信号;小数部分只有低位为 0,而且被熄灭的情况下,高位才有灭零输入信号,从而实现了多位十进制数码的灭零控制。

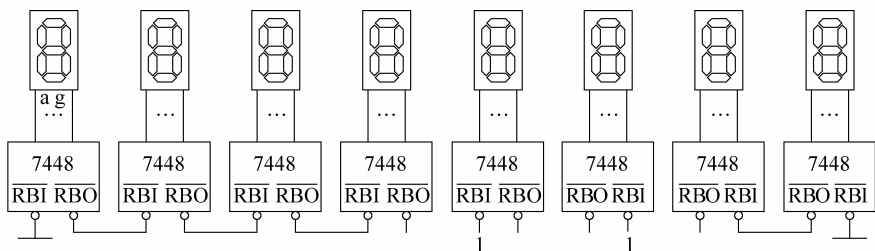


图 3.3.14 例 3.3.4 的电路(灭零功能的多位数码显示器)

3.4 数据选择器和分配器

3.4.1 数据选择器

在数字系统中,通常需要从多路数据中选择一路进行传输,执行这种功能的电路称为数据选择器(Multiplexer),也称多路开关或多路选择器,简称 MUX。图 3.4.1 为四选一数据选择器功能示意图, $D_0 \sim D_3$ 是数据输入端, A_1 、 A_0 是数据选择控制端又称地址输入端。四选一数据选择器真值表如表 3.4.1 所示。

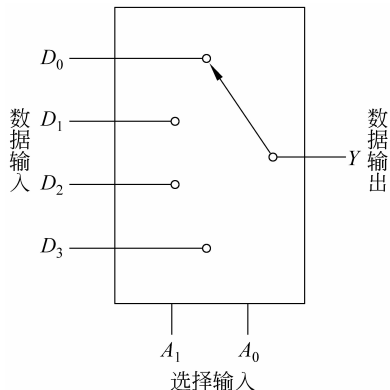


图 3.4.1 四选一数据选择器示意图

表 3.4.1 四选一数据选择器真值表

输 入		输 出
A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

由表 3.4.1 可得到输出变量的逻辑函数表达式为

$$Y = (\bar{A}_1 \bar{A}_0) D_0 + (\bar{A}_1 A_0) D_1 + (A_1 \bar{A}_0) D_2 + (A_1 A_0) D_3$$

中规模集成电路数据选择器按其所用半导体材料和制造工艺,可分为 CMOS 数据选择器和 TTL 数据选择器;按内部结构的不同,可分为单通道数据选择器和双通道数据选择器两种。下面主要讨论单通道的八选一数据选择器 74LS151 和双通道的双四选一数据选择器 74LS153 的工作原理。

1. 八选一数据选择器 74LS151

图 3.4.2(a)为 TTL 中规模集成电路 74LS151 型八选一数据选择器的逻辑图,它通过给定不同的地址代码,即 $A_2A_1A_0$ 的状态,从 8 个输入数据 $D_0 \sim D_7$ 中选出一个,并送至输出端。 $\bar{S}$ 是附加控制端,用于控制电路工作状态和扩展功能, $\bar{Y}$ 为 Y 的互补输出端。其真值表如表 3.4.2 所示。

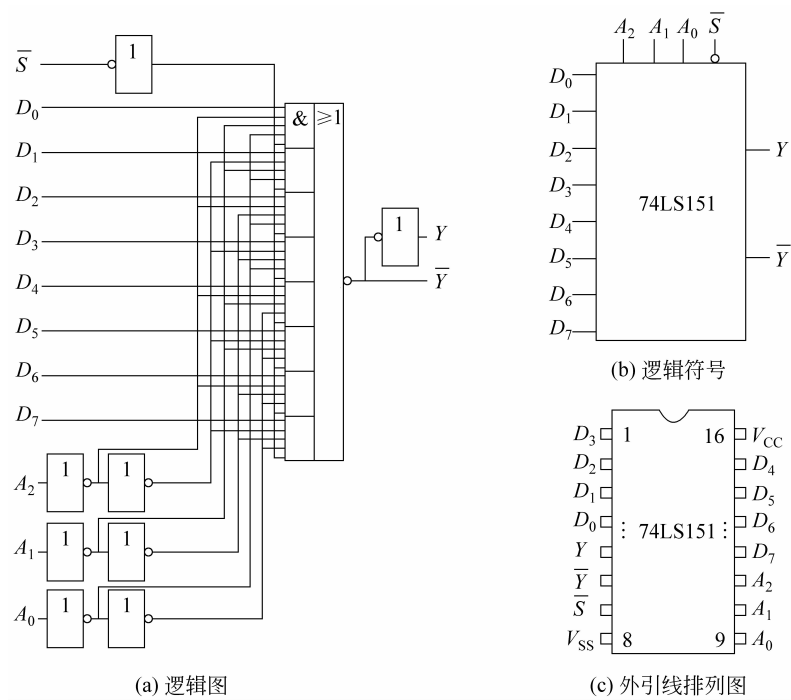


图 3.4.2 74LS151 型八选一数据选择器

表 3.4.2 74LS151 数据选择器真值表

控制端	输 入			输 出		控制端	输 入			输 出	
$\bar{S}$	A_2	A_1	A_0	Y	$\bar{Y}$	$\bar{S}$	A_2	A_1	A_0	Y	$\bar{Y}$
1	×	×	×	0	1	0	1	0	0	D_4	$\bar{D}_4$
0	0	0	0	D_0	$\bar{D}_0$	0	1	0	1	D_5	$\bar{D}_5$
0	0	0	1	D_1	$\bar{D}_1$	0	1	1	0	D_6	$\bar{D}_6$
0	0	1	0	D_2	$\bar{D}_2$	0	1	1	1	D_7	$\bar{D}_7$
0	0	1	1	D_3	$\bar{D}_3$						

74LS151 的逻辑符号和外引线排列图如图 3.4.2(b)和图 3.4.2(c)所示。

由表 3.4.2 可知,74LS151 数据选择器输出变量逻辑表达式为

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

2. 双四选一数据选择器 74LS153

图 3.4.3(a)为 TTL 中规模集成电路 74LS153 型双四选一数据选择器的逻辑图,

图 3.4.3(b) 为其逻辑符号。 A_1 、 A_0 为两个公用的选择输入端,两个附加控制端 $\bar{S}_1$ 、 $\bar{S}_2$ 各自独立,低电平控制数据输出,两个输出端 Y_1 、 Y_2 也是互相独立的。其真值表如表 3.4.3 所示。

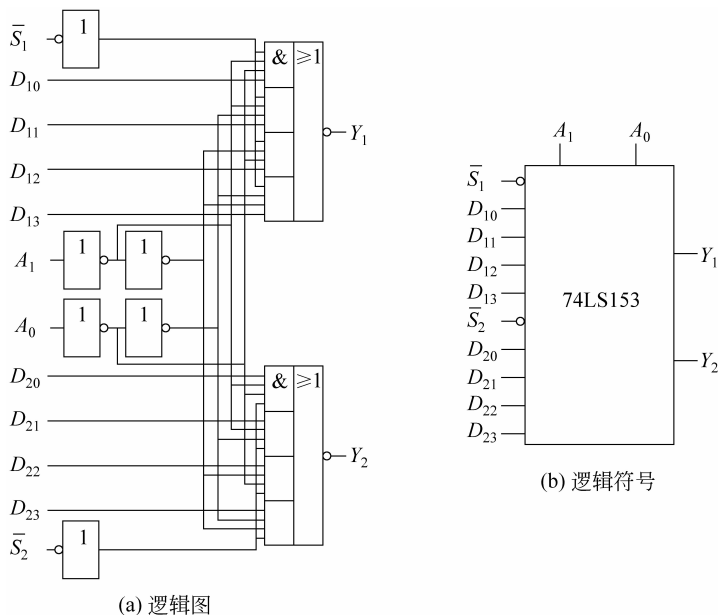


图 3.4.3 74LS153 型双四选一数据选择器

表 3.4.3 74LS153 数据选择器真值表

控制端		输入		输出		控制端		输入		输出	
$\bar{S}_1$	$\bar{S}_2$	A_1	A_0	Y_1	Y_2	$\bar{S}_1$	$\bar{S}_2$	A_1	A_0	Y_1	Y_2
1	1	×	×	0	0	0	0	1	0	D_{12}	D_{22}
0	0	0	0	D_{10}	D_{20}	0	0	1	1	D_{13}	D_{23}
0	0	0	1	D_{11}	D_{21}						

输出逻辑表达式可写成

$$Y_1 = [(\bar{A}_1 \bar{A}_0) D_{10} + (\bar{A}_1 A_0) D_{11} + (A_1 \bar{A}_0) D_{12} + (A_1 A_0) D_{13}] S_1$$

$$Y_2 = [(\bar{A}_1 \bar{A}_0) D_{20} + (\bar{A}_1 A_0) D_{21} + (A_1 \bar{A}_0) D_{22} + (A_1 A_0) D_{23}] S_2$$

在 CMOS 集成电路中经常用传输门组成数据选择器。图 3.4.4 中给出了双四选一数据选择器 CC14539 的逻辑电路结构,地址输入 $A_1 A_0$ 是公用的,两组的数据输入、输出和控制端都是互相独立的。通过 $A_1 A_0$ 不同状态选通不同的传输门,从而控制哪一个数据输入至输出端。例如,当 $A_1 A_0 = 10$ 时,传输门 TG_1 、 TG_3 、 TG_7 和 TG_9 导通,同时 TG_6 和 TG_{12} 也导通,只有 D_{12} 、 D_{22} 两个输入端数据通过传输门分别到达输出端 Y_1 和 Y_2 。CC14539 的逻辑符号如图 3.4.3(b) 所示。

由于集成电路受到电路芯片面积和外部封装大小的限制,目前生产的中规模数据选择器的最大数据通道为 16。当有较多的数据源需要选择时,可以用多片小容量的数据选择器组合来进行容量的扩展。