

第3章

组合逻辑电路

如果电路没有时序器件,在任何时刻的输出仅仅取决于该时刻的输入,与电路原来的状态无关,这样的数字逻辑电路称为组合逻辑电路。组合逻辑电路由逻辑门及其他组合逻辑器件组合而成,电路中的数字信号只单向传输,一般没有反馈电路,其输出 Y 与输入 X 之间的逻辑函数 $f()$ 可表示为 $Y_1 = f_1(X_1, X_2, \dots, X_n), Y_2 = f_2(X_1, X_2, \dots, X_n), \dots, Y_m = f_m(X_1, X_2, \dots, X_n)$, 如果要求数字电路输入输出逻辑关系与时间无关,则可用组合逻辑电路实现。

3.1 SSI 组合逻辑电路的分析



组合逻辑电路的基本单元是与、或、非 3 种逻辑门,与非、或非、与或非是复合逻辑门,这些集成门电路属于小规模集成电路,用这些器件构成的组合逻辑电路属于 SSI 组合逻辑电路。组合逻辑电路实际上是逻辑函数的电路实现,电路图、真值表、卡诺图、波形图都是描述逻辑电路的图表工具。

1. 组合逻辑电路的一般分析步骤

对于给定组合逻辑电路,我们常常要分析其功能,SSI 组合逻辑电路的分析可分为以下 4 步。

- (1) 从组合逻辑电路的输入至输出端,逐级写出逻辑函数表达式,最后得到电路的输出函数。
- (2) 用公式法或卡诺图法化简逻辑函数,得到最简逻辑表达式。
- (3) 将输入值不同取值代入最简逻辑函数表达式,计算出对应的函数值并将其填入真值表中,或者直接由最小项之和的标准表达式得到真值表。
- (4) 观察真值表,找出输出与输入之间的关系,用准确的语言描述电路的逻辑功能。

2. 组合逻辑电路的分析示例

例 3.1 分析图 3.1 所示的 SSI 逻辑电路,写出电路对应的逻辑函数,画出电路的输入输出波形图。

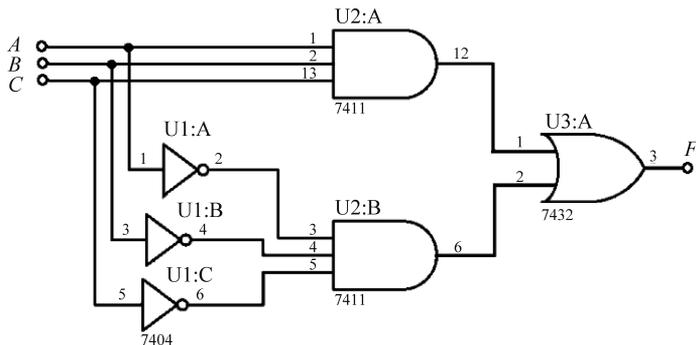


图 3.1 例 3.1 的逻辑电路图

解：(1) 写出电路对应的逻辑函数。

电路对应的逻辑函数为 $F(A, B, C) = ABC + \overline{A}\overline{B}\overline{C}$ ，已是最简函数，无须化简。

(2) 列出真值表，画出输入输出的波形。

逻辑函数为最小项之和的标准表达式，最小项为 ABC 与 $\overline{A}\overline{B}\overline{C}$ ，所以，仅当 $ABC = 111$ 或 $ABC = 000$ 时， $F = 1$ ；其他情况下， $F = 0$ 。因此，无须计算，直接得到真值表，如表 3.1 所示。

表 3.1 图 3.1 逻辑电路的真值表

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
F	1	0	0	0	0	0	0	1

根据真值表，可画出逻辑函数的输入输出波形图，如图 3.2 所示。

(3) 分析电路功能。

从真值表中，可看出当三输入变量完全相同时，即全为 1 或全为 0 时，函数值为 1；其他情况下，函数值为 0。因此，该电路是三变量一致性检测电路。

例 3.2 如图 3.3 所示的逻辑电路有两个输出端： F 、 G ，试写出电路对应的逻辑函数，并分析该电路的功能。

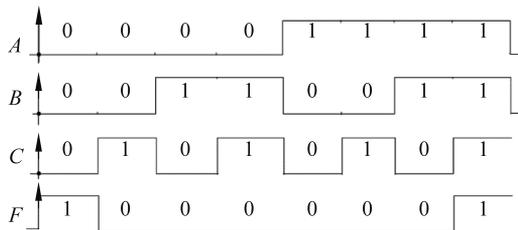


图 3.2 输入输出波形图

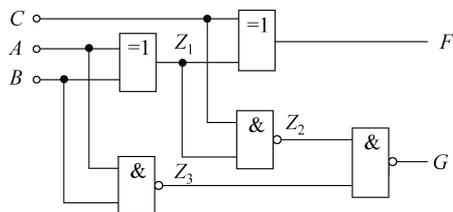


图 3.3 例 3.2 的逻辑电路图

解：图 3.3 所示的门电路采用国标符号，电路中含有“与非门”和“异或门”两种器件。

(1) 为了避免错误，应逐级写出逻辑表达式，最后写出输出函数 F 、 G 的表达式。 $Z_1 = A \oplus B$ ， $Z_3 = \overline{AB}$ ； $F(A, B, C) = C \oplus Z_1 = C \oplus (A \oplus B) = A \oplus B \oplus C$ ， $Z_2(A, B, C) = \overline{CZ_1} = \overline{C(A \oplus B)}$ ； $G(A, B, C) = \overline{Z_2 Z_3} = \overline{\overline{C(A \oplus B)} \overline{AB}}$ 。

(2) 对输出函数进行化简。

函数 $F(A, B, C)$ 无须化简， $F(A, B, C) = A \oplus B \oplus C$ ； $G(A, B, C) = \overline{\overline{Z_2 Z_3}} = \overline{\overline{C(A \oplus B)} \overline{AB}} = (A \oplus B)C + AB$ 。

(3) 列出输出函数 $F(A, B, C)$ 与 $G(A, B, C)$ 对应的真值表，如表 3.2 所示。

表 3.2 图 3.3 逻辑电路的真值表

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
F	0	1	1	0	1	0	0	1
G	0	0	0	1	0	1	1	1

(4) 分析电路功能。

由真值表可知:当输入信号(A,B,C)中1的个数为奇数个时,输出F为1,其他情况为0;当输入信号(A,B,C)中有两个或两个以上的1时,输出G为1,其他为0。因此,可认为A和B是被加数或加数,C是低位的进位数,F是带进位输入的一位二进制数加法的和,G是向高位的进位数,即加法的进位输出。可见,该电路是一个带进位输入与进位输出的一位二进制加法器,这种功能的器件称为全加器,该电路就是一位全加器。

3.2 SSI 组合逻辑电路的设计



用 SSI 组合逻辑器件设计电路实现给定的功能,这个过程称为 SSI 组合逻辑电路的设计。组合逻辑电路的设计和组合逻辑电路的分析互为逆过程。

组合逻辑电路设计的基本步骤如下。

(1) 根据给出的条件和最终要实现的功能进行逻辑抽象:设置输入和输出逻辑变量,每个逻辑变量具有0和1两种值,全部输入变量空间即为输入状态空间。

(2) 列出表明输入输出逻辑关系的真值表。在真值表左侧,列出所有输入状态,即输入变量的所有取值组合,依照题意,在真值表的右侧写出对应的输出逻辑值。

(3) 根据真值表写出输入输出的逻辑函数表达式,并进行化简,得到最简逻辑函数。

(4) 根据最简逻辑函数和给定器件,画出逻辑电路图。

例 3.3 设计一个三变量多数表决器。3人参加某提案表决,若多数同意,则提案获得通过;若少数同意,则提案被否决。要求列出真值表,化简逻辑函数,画出用与非门实现的电路图。

解: (1) 设置输入输出逻辑变量。

根据题目的要求,表决人的意见对应输入逻辑变量,用变量A、B、C表示;表决结果对应输出逻辑变量,用变量F表示。当输入为1时,表示同意提案,为0时表示否决。输出F为1时,提案获得通过;为0时提案被否决。

(2) 列出真值表,如表3.3所示。

表 3.3 三变量多数表决器的真值表

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

续表

C	0	1	0	1	0	1	0	1
F	0	0	0	1	0	1	1	1

(3) 写出逻辑函数并进行化简。

表 3.3 中,输出逻辑函数共有 4 种情况下值为 1,所以函数式有 4 个最小项,它们是 $011 \rightarrow \bar{A}BC$ 、 $101 \rightarrow A\bar{B}C$ 、 $110 \rightarrow ABC\bar{C}$ 、 $111 \rightarrow ABC$ 。逻辑函数 $F(A, B, C) = \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC = \sum m^3(3, 5, 6, 7)$, 利用卡诺图化简逻辑函数,如图 3.4 所示。

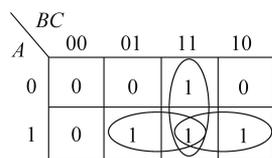
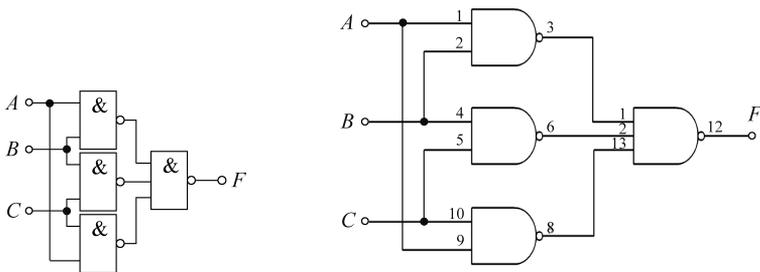


图 3.4 利用卡诺图化简逻辑函数

化简后, $F(A, B, C) = AB + BC + AC$ 。

(4) 对最简与或表达式进行两次非运算,转换成与非-与非表达式,即 $F(A, B, C) = AB + BC + AC = \overline{\overline{AB + BC + AC}} = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}$ 。

(5) 画出用与非门实现的逻辑电路图,如图 3.5(a)和图 3.5(b)所示分别是使用国标和国际标准的门电路符号绘制的电路图。



(a) 使用国标门电路符号绘制的电路图

(b) 使用国际标准门电路符号绘制的电路图

图 3.5 例 3.3 的逻辑电路图

例 3.4 球场照明亮度大于规定亮度后,球场才可举行比赛。某球场有甲、乙、丙、丁四盏照明灯,甲灯为高亮度灯,其余 3 盏灯均为普通亮度灯。当高亮度灯点亮时,只要再点亮一盏普通亮度灯,球场照明亮度就达标,可进行比赛;当高亮度灯熄灭时,3 盏普通亮度灯必须同时点亮,灯光才能满足比赛要求。设计逻辑电路,根据 4 盏照明灯的亮暗情况,判断球场灯光是否满足比赛的要求。如果可以举行比赛,则指示灯亮;若不能,则指示灯暗。要求列出真值表,化简逻辑函数,画出逻辑电路图。

解: (1) 设置输入输出变量,列出真值表。

设变量 A 、 B 、 C 、 D 分别表示甲、乙、丙、丁 4 盏灯的明暗情况,灯亮变量值为 1,灯暗为 0。若灯光满足球场比赛要求,则函数值 $F=1$;若不满足,则函数值 $F=0$ 。依题意得真值表,如表 3.4 所示。

表 3.4 真值表

A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
F	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1

(2) 画出卡诺图,如图 3.6 所示,将真值表填入卡诺图中,利用卡诺图化简逻辑函数。

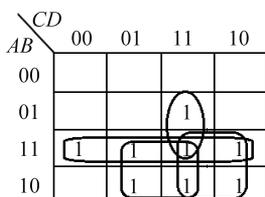


图 3.6 卡诺图

经化简,得到最简逻辑函数 $F(A, B, C, D) = AB + AC + AD + BCD$ 。

(3) 画出逻辑电路图,如图 3.7 所示。

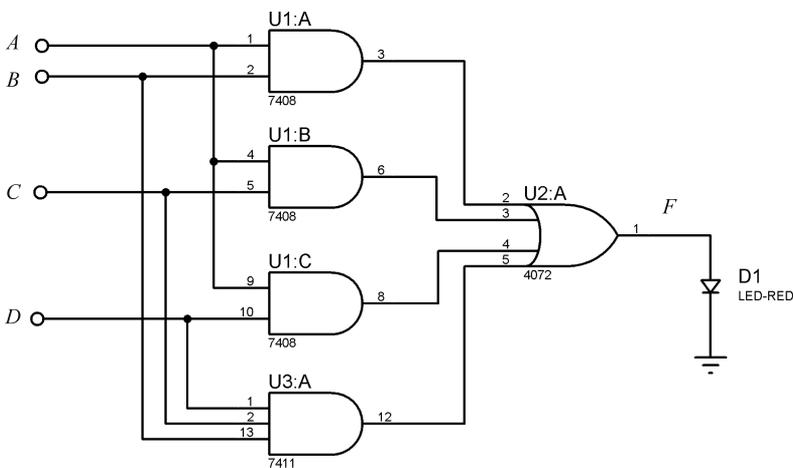


图 3.7 逻辑电路图 4

在例 3.4 题中,若用与非门设计电路,先将逻辑函数 $F(A, B, C, D) = AB + AC + AD + BCD$ 转换成与非-与非的形式: $F(A, B, C, D) = \overline{\overline{AB + AC + AD + BCD}} = \overline{\overline{AB} \cdot \overline{AC} \cdot \overline{AD} \cdot \overline{BCD}}$,然后画出用与非门实现的电路,如图 3.8 所示。

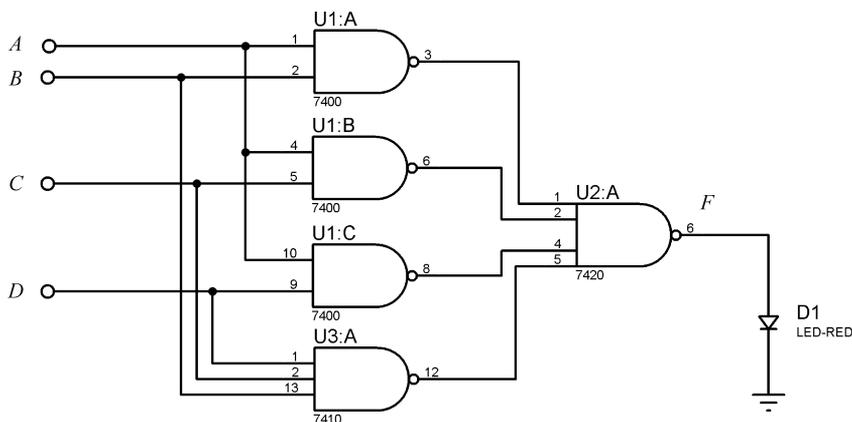


图 3.8 用与非门实现的电路图

3.3 常用 MSI 组合逻辑电路

为应用方便,将一些常用功能的逻辑电路制作成集成电路器件。常用的中规模集成电路度(medium scale integrated)的组合逻辑电路器件有编码器、译码器、数据选择器、数据分配器、数值比较器、加法器、算术逻辑运算单元等。

3.3.1 编码器

1. 编码原则



用若干位代码去标识特定意义的信号,这个过程就是编码。 n 位二进制代码可以表示 2^n 种不同的信息,若用 n 位编码,要能表示 M 个不同的信息, n 应满足 $2^n \geq M$ 。例如,101 键盘有 101 个按键,对按键编码时,若采用 6 位二进制编码, $2^6 = 64 < 101$,编码不足以表示 101 个按键;若采用 7 位编码, $2^7 = 128 > 101$,编码能够区分 101 个按键。因此,人们对 101 键盘编码时采用了 7 位二进制的 ASCII 码。

【思考】 如果要对 8 种信号进行编码,至少需要多少位二进制代码? 如果要对 9 种信号进行编码呢?

2. 普通编码器

编码器是一种实现编码功能的中规模组合逻辑器件,普通编码器任何时刻只允许输入一个有效信号,然后输出对应的二进制编码。3 位二进制普通编码器如图 3.9 所示。

图 3.9 所示的 3 位二进制普通编码器,输入为

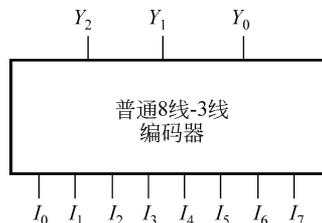


图 3.9 3 位二进制普通编码器

I_0, I_1, \dots, I_7 等 8 种需要编码的原始信息, 输出量用 Y_2, Y_1, Y_0 分别表示对应的 8 种编码 000、001、010、011、100、101、110、111。3 位二进制普通编码器的真值表如表 3.5 所示。

表 3.5 3 位二进制普通编码器的真值表

输入 编 码								输出 编 码		
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

表 3.5 中, 编码器输入信号 I_i 为高电平有效, 输出的编码 $Y_2 Y_1 Y_0$ 是原码。实际上, 常见的编码器一般输入信号为低电平有效, 输出的编码常常是对原码按位取反后的代码(类似反码)。根据表 3.5, 可以求得输出逻辑函数, 如下。

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

$$Y_1 = I_2 + I_3 + I_6 + I_7 = \overline{I_2} \overline{I_3} \overline{I_6} \overline{I_7}$$

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

如图 3.10(a)所示是用或门实现的电路图, 图 3.10(b)是用与非门实现的电路图。

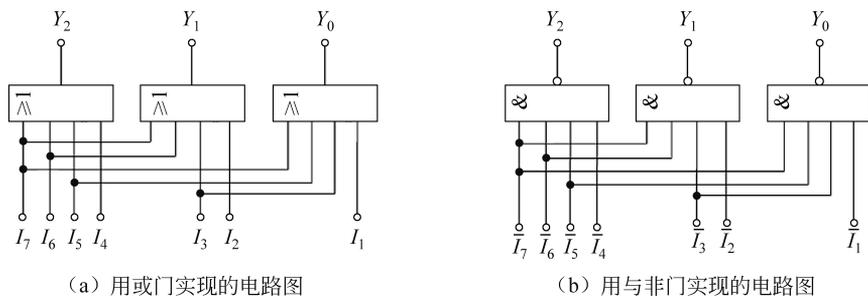


图 3.10 3 位二进制编码器的逻辑电路图

3. 优先编码器

在数字系统中, 当编码器允许同时有多个输入信号有效, 且每个输入信号有不同的

优先级别时,编码器只对其中优先权最高的一个输入信号进行编码,具有这样功能的编码器称为优先编码器。

1) 9线-4线优先编码器

74LS147是9线-4线的优先编码器,输出9个输入信号对应的4位二进制代码,芯片符号如图3.11所示。

74LS147属于中规模集成电路,图3.11中用符号 $\bar{I}_1 \sim \bar{I}_9$ 表示输入信号,上画线是为了提示输入低电平有效,输入端 \bar{I}_9 的优先级别最高, \bar{I}_1 的优先级别最低;输出端4位编码为 \overline{DCBA} , \bar{D} 为最高位, \bar{A} 位为最低位,上画线是为了提示编码输出是按位取反的代码。74LS147的输入和输出均是低电平有效,所以输入输出信号用带上画线符号表示,并不是表示信号必须进行非运算,才可作为电路的输入或输出,如果用不带上画线的符号表示输入输出信号,原则上也没有错,只是不能通过符号知晓芯片输入输出低电平有效的特点。

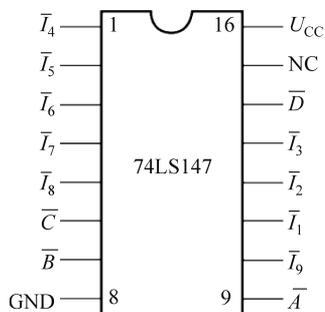


图 3.11 74LS147 芯片的引脚排列和常用符号

74LS147的真值表如表3.6所示,当无输入信号时,即全部输入为高电平: $\bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 \bar{I}_8 \bar{I}_9 = 111111111$,输出端将全部为高电平 $\overline{DCBA} = 1111$;当输入端 $\bar{I}_9 = 0$,表示 \bar{I}_9 有输入,由于 \bar{I}_9 优先级最高,此时,不论其他输入端是否有输入,即是否 $\bar{I}_{n \neq 9} = 0$,编码器输出“9”的代码,即1001按位取反后的代码 $\overline{DCBA} = 1001 = 0110$ 。

表 3.6 编码器 74LS147 的真值表

\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{D}	\bar{C}	\bar{B}	\bar{A}
×	×	×	×	×	×	×	×	×	1	1	1	1
×	×	×	×	×	×	×	×	0	0	1	1	0
×	×	×	×	×	×	×	0	1	0	1	1	1
×	×	×	×	×	×	0	1	1	1	0	0	0
×	×	×	×	0	1	1	1	1	1	0	0	1
×	×	×	0	1	1	1	1	1	1	0	1	0
×	×	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

2) 8线-3线优先编码器 74LS148

74LS148芯片是一种8线-3线优先编码器。74LS148芯片的常用符号如图3.12所

示,图中 $\bar{I}_1 \sim \bar{I}_7$ 为8个输入信号,优先级别依次递增, $\bar{Y}_2 \bar{Y}_1 \bar{Y}_0$ 为输出的编码信号, \bar{S} 为使能输入端, \overline{OE} 和 \overline{GS} 共同指示芯片的工作状态。当 \bar{S} 为低电平,且输入信号 $\bar{I}_1 \sim \bar{I}_7$ 有低电平时,芯片进行编码工作,输出3位二进制数按位取反后的代码 $\bar{Y}_2 \bar{Y}_1 \bar{Y}_0$,此时,状态信号 $\overline{OEGS} = 01$,表明允许芯片工作,且芯片有了有效的输入信号,芯片正常工作,输出了编码信号。 \overline{OE} 和 \overline{GS} 信号可用于多片74LS148级联以扩展编码位数,两片74LS148通过 \overline{OE} 和 \overline{GS} 的恰当连接,可以扩展为16线-4线优先编码器。

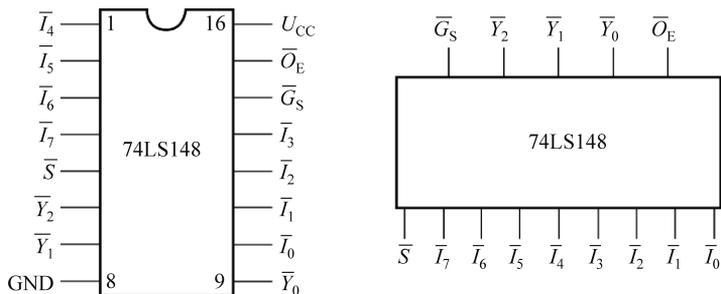


图 3.12 74LS148 芯片的常用符号

编码器 74LS148 的真值表如表 3.7 所示。

表 3.7 编码器 74LS148 的真值表

\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	\overline{GS}	\overline{OE}
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	0	0	0	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

3.3.2 译码器

根据输入的 n 位二进制代码,在 2^n 个输出端口的对应的一路获得输出,完成这样功能的器件称为译码器。例如,如果 3 线-8 线译码器输入 3 位二进制代码 011,那么译码器

0~7号输出端口中的3号输出端获得输出。如果输出高电平有效,3号输出逻辑1的电平;如果输出低电平有效,3号输出逻辑0的电平。显然,译码器和编码器的功能正好相反。若译码器输入二进制代码有 n 位,则有 2^n 个输出端口。

译码器也是一种常用的中规模组合逻辑电路,在数字系统中,译码器不仅在代码转换与数码显示中得到广泛的应用,还常常用于数据分配、存储器寻址、信号控制等场合。根据功能差异,译码器可分为变量译码器、显示译码器和代码变换译码器。本节主要介绍变量译码器和显示译码器的外部工作特性和应用。

1. 变量译码器



74LS138是一种16只引脚的3线-8线变量译码器芯片,芯片符号如图3.13所示,第16脚是电源端,第8脚是接地端, A_2 、 A_1 、 A_0 是3位二进制代码输入端, $\bar{Y}_7 \sim \bar{Y}_0$ 是8个译码输出端口, G_1 、 \bar{G}_{2A} 、 \bar{G}_{2B} 是芯片的使能信号。

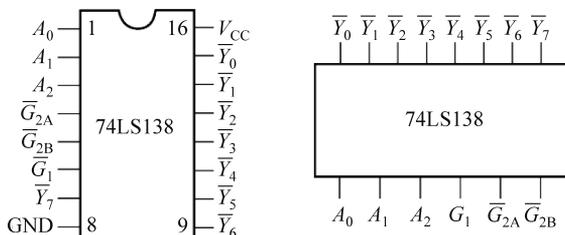


图 3.13 74LS138 芯片的引脚和常用符号

当使能信号 $G_1 = 1$, $\bar{G}_{2A} = \bar{G}_{2B} = 0$ 时,芯片进行译码工作,根据输入代码 $A_2A_1A_0$,相应输出端口 \bar{Y}_i 产生低电平;若使能信号为其他取值,则74LS138输出端全部高电平,表示无输出。使能信号 G_1 、 \bar{G}_{2A} 、 \bar{G}_{2B} 也可用于多片74LS138级联扩展功能,例如,通过使能信号恰当地连接,两片74LS138级联后可扩展成4线-16线译码器。

译码器74LS138的真值表如表3.8所示。

表 3.8 译码器74LS138的真值表

G_1	$\bar{G}_{2A} + \bar{G}_{2B}$	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
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

续表

G_1	$\bar{G}_{2A} + \bar{G}_{2B}$	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	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

74LS138 输出低电平有效,即输出端为低电平,表示产生输出。74LS138 各输出函数如下。

$$\begin{aligned}\bar{Y}_0 &= \overline{A_2 A_1 A_0} = \bar{m}_0 \\ \bar{Y}_1 &= \overline{A_2 A_1 A_0} = \bar{m}_1 \\ &\vdots \\ \bar{Y}_7 &= \overline{A_2 A_1 A_0} = \bar{m}_7\end{aligned}$$

每个输出函数 \bar{Y}_i 是输入变量 A_2, A_1, A_0 的一个最小项的反函数: $\bar{Y}_i = \overline{m_i(A_2, A_1, A_0)}$ 。

例 3.5 试用译码器 74LS138 实现逻辑函数 $F(A, B, C) = \bar{A}B + \bar{B}C + A\bar{C}$ 。

解: (1) 通过配项,求得函数 F 的最小项之和表达式为:

$$F(A, B, C) = \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + A\bar{B}\bar{C} = \sum m^3(1, 2, 3, 4, 5, 6)。$$

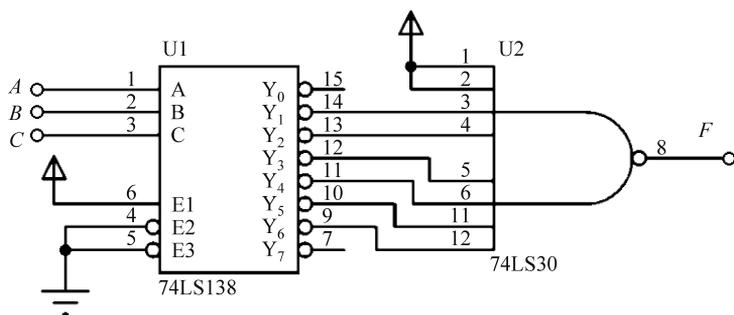
(2) 将函数 F 的最小项之和表达式转换成与非-与非的形式,如下。

$$\begin{aligned}F(A, B, C) &= \overline{\overline{\bar{A}BC + \bar{A}\bar{B}C + A\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + A\bar{B}\bar{C}}} \\ &= \overline{\bar{A}BC \cdot \bar{A}\bar{B}C \cdot A\bar{B}C \cdot \bar{A}\bar{B}\bar{C} \cdot ABC \cdot A\bar{B}\bar{C}} \\ &= \overline{\bar{Y}_3 \cdot \bar{Y}_2 \cdot \bar{Y}_5 \cdot \bar{Y}_1 \cdot \bar{Y}_6 \cdot \bar{Y}_4}\end{aligned}$$

其中, Y_i 为 A, B, C 的最小项,即 $Y_i = m_i(A, B, C)$,将 A, B, C 接入 74LS138 的输入端,74LS138 的 8 个输出就是最小项 $m_i(A, B, C)$ 的反函数,选取 $F(A, B, C)$ 的 6 个输出: $\bar{Y}_1 = \overline{\bar{A}\bar{B}C}, \bar{Y}_2 = \overline{\bar{A}BC}, \bar{Y}_3 = \overline{A\bar{B}C}, \bar{Y}_4 = \overline{\bar{A}\bar{B}\bar{C}}, \bar{Y}_5 = \overline{ABC}, \bar{Y}_6 = \overline{A\bar{B}\bar{C}}$,然后进行与非运算 $\overline{\bar{Y}_1 \cdot \bar{Y}_2 \cdot \bar{Y}_3 \cdot \bar{Y}_4 \cdot \bar{Y}_5 \cdot \bar{Y}_6}$,便实现了函数 $F(A, B, C) = \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + A\bar{B}\bar{C}$ 。用 74LS138 实现逻辑函数 F 的电路图如图 3.14 所示,图中使用了 8 输入端与非门 74LS30,多余的 2 个输入引脚接高电平,便成了 6 输入端与非门。

$$F(A, B, C) = \overline{1 \cdot 1 \cdot \bar{Y}_1 \cdot \bar{Y}_2 \cdot \bar{Y}_3 \cdot \bar{Y}_4 \cdot \bar{Y}_5 \cdot \bar{Y}_6} = \overline{\bar{Y}_1 \cdot \bar{Y}_2 \cdot \bar{Y}_3 \cdot \bar{Y}_4 \cdot \bar{Y}_5 \cdot \bar{Y}_6}$$

在图 3.14 中,使用 1 片中规模集成电路器件 74LS138 和 1 片 8 输入端与非门 74LS30,就实现了逻辑函数 $F(A, B, C) = \bar{A}B + \bar{B}C + A\bar{C}$ 。如果电路全部使用与、或、非、与非、或非等小规模集成电路器件,至少需要 3 片芯片才可实现逻辑函数 $F(A, B, C)$,可见,与小规模集成电路器件相比,使用中规模集成电路器件进行电路设计,可以节约器件的数量,电路更简化,意味着不仅降低了制作成本,又提高了电路的可靠性。

图 3.14 用 74LS138 实现逻辑函数 F 的电路图

2. 显示译码器



在数字系统中,信息常常需要通过终端显示出来,便于人们观察,因此,需要将表示数字、文字、符号等信息的二进制代码翻译成显示器的显示代码,使显示器直观地显示数字、文字、符号等信息,供人们查看。显示器有多种,如笔划段型数码管、点阵型显示器、液晶显示(liquid crystal display, LCD)器等,每种显示器控制显示的方式不同。显示器耗能较大,常常需要驱动电路才能正常显示,显示译码器具有译码和驱动双重功能。

1) 半导体显示器

发光二极管外加正向电压时,可以将电能转换成光能,即发光。笔划段型数码管将小型发光二极管封装成数码的笔划段,每段用一个发光二极管控制其显示,七段发光二极管组成“日”字形,用来显示数码 0~9,例如,要显示数码“4”,就要点亮发光二极管的 b、c、f、g 段,熄灭 a、d、e 段,一个七段数码管显示一位数码,八段数码管显示带小数点的数码,如图 3.15(a)所示。点阵型显示器由行列矩阵式的可控发光点组成,每个发光点封装了一个发光二极管,由行列式矩阵的发光点阵来显示字符和图形,如图 3.15(b)所示。

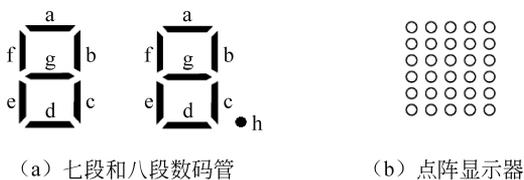


图 3.15 笔划段型数码管和点阵型显示器示意图

笔划段型数码管分为共阴极和共阳极两种,如图 3.16 所示,共阴极数码管器件内部的发光二极管阴极端连在一起,引出为数码管的公共端(com 引脚),因此,应用时,共阴极数码管 com 引脚要接地,这样数码管才可能点亮。共阴极数码管的输入信号连接到发光二极管的阳极,共阴极数码管的哪段需要点亮,该段的输入端就施加高电平,若输入低电平,该段就不亮。

共阳极数码管正好相反,数码管内部的发光二极管阳极端连在一起,引出为数码管的公共端(com 引脚)。应用时,共阳极数码管公共端必须接高电平,共阳极数码管各段

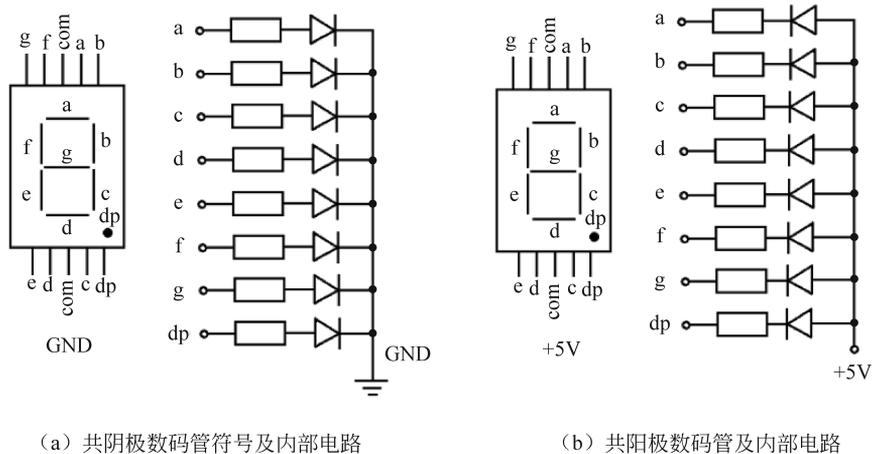


图 3.16 笔划段型八段数码管

输入端连接发光二极管的阴极,因此,共阳极数码管的哪一段要点亮,哪一段的输入端就施加低电平,若输入高电平,则该段就不亮。

例 3.6 使用七段共阴极数码管显示数字“3”,试问该数码管的公共端应接什么电平? 输入信号 a、b、c、d、e、f、g 应接什么电平? 如果换成七段共阳极数码管显示数字“3”,则公共端应接什么电平? 输入信号 a、b、c、d、e、f、g 又应接什么电平?

解: 要显示数字“3”,数码管的 e、f 段应不亮,a、b、c、d、g 段要点亮,如图 3.17 所示。

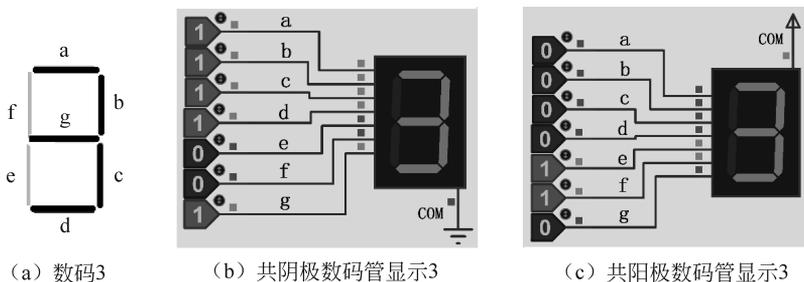


图 3.17 数码管数码 3 的显示

使用共阴极数码管时,公共端应接地,笔端输入高电平点亮,输入 abcdefg 电平应为 1111001;使用共阳极数码管时,公共端应接高电平,笔端低电平点亮,输入信号 abcdefg 电平应为 0000110。

发光二极管导通电阻在几欧到几百欧之间,小型发光二极管正常发光的额定工作电流一般为 20mA 左右,红色发光二极管的压降一般为 1.8~2.2V,黄色发光二极管的压降一般为 1.8~2.0V,其他颜色的工作电压为 3V 左右。因此,无论是共阴极还是共阳极数码管,应用时,各段输入端应串联一个限流电阻,以防电流过大,烧毁发光二极管,但是,限流电阻也不可过大,以免电流过小,发光二极管显示过于暗淡或不发光。

例 3.7 图 3.18 中,发光二极管 D_1 的压降为 $2V$,工作电流为 $10\sim 20mA$,如果电路电压 $U_1=3.6V$,试问限流电阻 R_1 的阻值范围为多少?若电路电压 $U_1=5V$,限流电阻 R_1 的阻值范围又为多少?

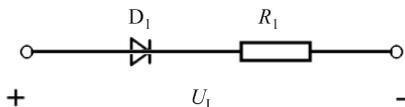


图 3.18 发光二极管应用电路

解: 当 $U_1=3.6V$ 时, R_1 的最大值 $R_{1MAX}=(3.6V-2V)/10mA=160\Omega$, R_1 的最小值 $R_{1MIN}=(3.6V-2V)/20mA=80\Omega$, 因此, 当 $U_1=3.6V$ 时, 限流电阻 R_1 阻值范围应在 $80\sim 160\Omega$ 。

当 $U_1=5V$ 时, R_1 的最大值 $R_{1MAX}=(5V-2V)/10mA=300\Omega$, R_1 的最小值 $R_{1MIN}=(5V-2V)/20mA=150\Omega$ 。因此, 当 $U_1=5V$ 时, 限流电阻 R_1 阻值范围应在 $150\sim 300\Omega$ 。

2) 液晶显示器

当某些晶体的温度介于两个熔点之间时, 呈现出液晶状态, 此时既有液体的流动性, 也不失晶体的某些特性, 如分子结构排列整齐有序、透明度好, 但是, 其透明度和颜色可以随电场、磁场、光、温度等外界条件的变化而变化。例如, 在极微小的电场作用下, 液晶分子结构能在极短的时间内改变排序状态, 从而影响透明度。若将电场施加在液晶不同部位, 液晶通过对光线阻挡或偏转, 依靠穿透的光线, 能够显示出字形和图案。液晶显示器件本身并不发光, 如果处于黑暗中, 液晶不会显示任何图形。

液晶显示器是一种平板薄型显示器件, 其驱动电压很低, 工作电流极小, 与 CMOS 电路组合起来可组成低功耗显示系统, 广泛应用于仪器、仪表及电子产品的显示器中。

3) 显示译码器

显示译码器是既有译码功能, 也有显示驱动功能的器件, 可以将 4 位 8421 码转译成数码管的各段显示代码, 同时提供较大的电流, 以驱动数码管显示。七段显示译码器有 7448、7449、74247、74248、72249、74347、CD4511、CD4543 等中规模集成电路芯片, 其中, 7448 和 CD4511 是共阴极数码管常用的七段显示译码器。

如图 3.19 所示是七段显示译码器 74LS48 芯片引脚图, 74LS48 的 4 位 8421 码输入端分别为 A_3 、 A_2 、 A_1 、 A_0 。其中, A_3 为最高位, A_0 为最低位, 输出的七段显示代码为

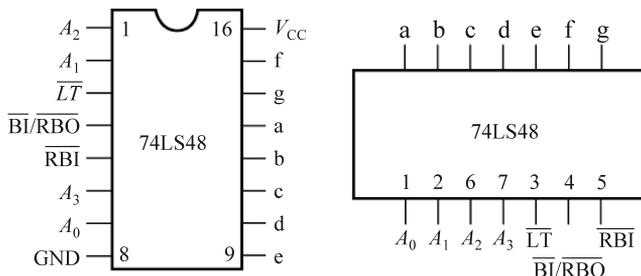


图 3.19 74LS48 芯片引脚图

a~g,该芯片内部的输出电路有上拉电阻,可以直接驱动共阴极数码管,芯片还有3个使能端: \overline{LT} 、 $\overline{BI/RBO}$ 、 \overline{RBI} 。显示译码器 74LS48 的真值表如表 3.9 所示。

表 3.9 显示译码器 7448 的真值表

\overline{LT}	\overline{RBI}	$\overline{BI/RBO}$	A_4	A_3	A_2	A_1	a	b	c	d	e	f	g	显示字符
1	1	1	0	0	0	0	1	1	1	1	1	1	0	显示 0
1	1	1	0	0	0	1	0	1	1	0	0	0	0	显示 1
1	1	1	0	0	1	0	1	1	0	1	1	0	1	显示 2
1	1	1	0	0	1	1	1	1	1	1	0	0	1	显示 3
1	1	1	0	1	0	0	0	1	1	0	0	1	1	显示 4
1	1	1	0	1	0	1	1	0	1	1	0	1	1	显示 5
1	1	1	0	1	1	0	0	0	1	1	1	1	1	显示 6
1	1	1	0	1	1	1	1	1	1	0	0	0	0	显示 7
1	1	1	1	0	0	0	1	1	1	1	1	1	1	显示 8
1	1	1	1	0	0	1	1	1	1	0	0	1	1	显示 9
1	1	1	1	0	1	0	0	0	0	1	1	0	1	显示 □
1	1	1	1	0	1	1	0	0	1	1	0	0	1	显示 □
1	1	1	1	1	0	0	0	1	0	0	0	1	1	显示 U
1	1	1	1	1	0	1	1	0	0	0	1	0	1	显示 E
1	1	1	1	1	1	0	0	0	0	1	1	1	1	显示 E
1	1	1	1	1	1	1	0	0	0	0	0	0	0	无显示
0	×	×	×	×	×	×	1	1	1	1	1	1	1	显示 8

在正常工作状态下, \overline{LT} 、 $\overline{BI/RBO}$ 、 \overline{RBI} 接高电平,在 $A_3A_2A_1A_0$ 端输入 4 位 8421 码,74LS48 输出 7 段共阴极数码管的显示代码,数码管显示相应的十进制数码。74LS48 和共阴极数码管的连接方式如图 3.20 所示。

74LS48 各引脚的功能如下。

(1) 试灯信号 \overline{LT} : 此信号低电平有效,当 $\overline{LT}=0$ 时,不论输入 $A_3A_2A_1A_0$ 取何值,输出端 $abcdefg=111111$,数码管显示数字“8”,此项功能用于测试数码管是否每一段都能够点亮,是否有损坏段。正常工作时, \overline{LT} 应为高电平。

(2) 灭灯信号 \overline{BI} : 当 $\overline{LT}=1$, $\overline{BI}=0$ 时,不论其他输入端为何种电平,全部输出端 $abcdefg=000000$,数码管灭灯,无显示。

(3) 条件灭零信号 \overline{RBI} : 当 $\overline{LT}=\overline{BI}=1$, $\overline{RBI}=0$ 时,若输入 $A_3A_2A_1A_0=0000$, $abcdefg=000000$,数码管无显示;若 $A_3A_2A_1A_0 \neq 0000$,芯片正常工作,数码管显示相应的数码。

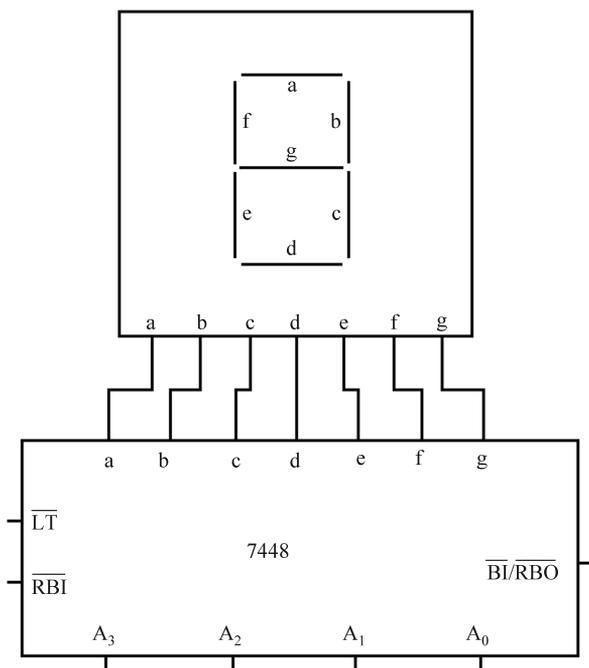


图 3.20 74LS48 和共阴极数码管的连接方式

(4) 灭零输出信号 \overline{RBO} : 若 $\overline{RB1}=0$ 且输入 $A_3A_2A_1A_0=0000$, \overline{RBO} 输出为 0, 表示输入为零, 符合灭零条件, 零被成功灭掉, 零不予显示。

如图 3.21 所示是使用显示译码器 7448 构成的多位数字显示系统。整数最高位 7448 的 $\overline{RB1}$ 与小数最低位的 $\overline{RB1}$ 均接逻辑 0, 若整数最高位或小数最低位输入 $A_3A_2A_1A_0=0000$, 数码管将不显示零, 如数据 012.340 最前和最后的零不应显示。小数点前的个位数字不管是 0 还是 1 都要显示, 如数据 0.12 与 10.23 在小数点前的 0 必须显示, 因此显示个位数的 7448 芯片 $\overline{RB1}$ 应设置为 1, 表明个位零不能灭掉。

多位显示系统中, 还需将高位 7448 芯片的 \overline{RBO} 与相邻低位 7448 的 $\overline{RB1}$ 连接。若高位 7448 芯片的 $\overline{RBO}=0$, 表明高位数 $A_3A_2A_1A_0=0000$, 且条件灭零有效 $\overline{RB1}=0$, 高位数据零被成功灭掉, 此时, 相邻低位若是零, 也应被灭掉。例如, 0012.34 的第一位零被灭掉后, 第二位零也应被灭掉, 因此相邻低位 7448 芯片应 $\overline{RB1}=0$, 此时, 高位 7448 芯片的 $\overline{RBO}=\overline{RB1}=0$ 。若高位 7448 的 $\overline{RBO}=1$, 则表明高位输入 $A_3A_2A_1A_0 \neq 0000$, 或虽然输入是零但不灭掉显示, 此时, 相邻低位若是零, 也必须显示, 低位 7448 芯片应设置 $\overline{RB1} \neq 0$, 此时, 高位 7448 芯片的 $\overline{RBO}=\overline{RB1}=1$ 。例如, 数据 1002.34 中的第一个零要显示出来, 第一个零对应 7448 芯片 $\overline{RBO}=1$, 与之相邻的第二个零也不能被灭掉, 第二个零对应的 7448 芯片应设置 $\overline{RB1}=1$, 满足高位 7448 芯片的 $\overline{RBO}=\overline{RB1}$ 。故高位 7448 芯片的 \overline{RBO} 始终应与相邻低位 7448 芯片的 $\overline{RB1}$ 相等, 因此要将它们相连。

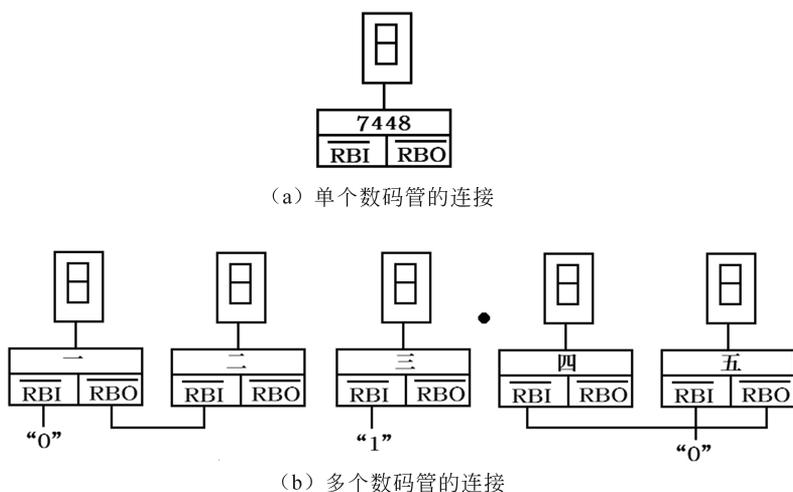


图 3.21 多位数字显示系统

3.3.3 数据选择器



1. 数据选择器的功能

从多路数据中选择一路数据进行输出的数字器件称为数据选择器,数据选择器也可以看作是多选一的电子开关。如图 3.22 所示为“四选一”的数据选择器的功能示意图,输入的 4 路数据为 $I_0 I_1 I_2 I_3$,地址信号为 $A_1 A_0$,使能信号为 \bar{G} 。当使能信号有效时,即 $\bar{G} = 0$,从 4 路数据 $I_0 I_1 I_2 I_3$ 中选择地址信号 $A_1 A_0$ 指定的那路信号输出,如地址信号 $A_1 A_0 = 00$,输出 $Y = I_0$;地址信号 $A_1 A_0 = 11$,输出 $Y = I_3$ 。

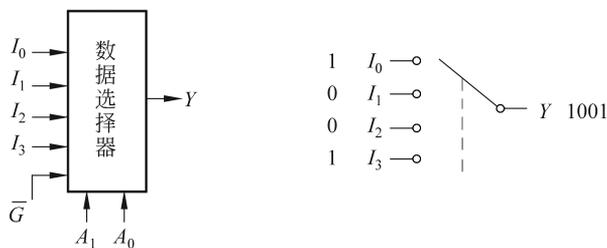


图 3.22 数据选择器的功能示意图

数据选择器也是常用的 MSI 组合逻辑器件。在远距离传输多位数据时,为了降低线路成本,常常将多路数据一位一位地发送到一条线路上传输,这时就需要数据选择器将并行数据转换成串行数据进行发送,如图 3.22 所示,“四选一”数据选择器输入并行数据 $I_3 I_2 I_1 I_0 = 1001$,若每隔 Δt 时间,地址变化 1 次,发送 1bit 数据,经过 $4\Delta t$ 时间后,输出端 Y 得到串行数据 1001。

2. 数据选择器的输出函数

74HC153 与 74HC151 分别是“四选一”和“八选一”的数据选择器,它们均属于中规模集成电路。74HC153 片内有两个“四选一”的数据选择器,分别由使能信号 $\overline{1E}$ 和 $\overline{2E}$ 控制,“八选一”的数据选择器 74HC151 有一对互补的输出 Y 和 \overline{Y} 。

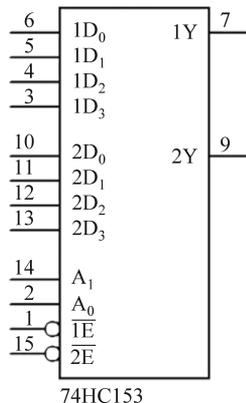


图 3.23 “四选一”的数据选择器 74HC153

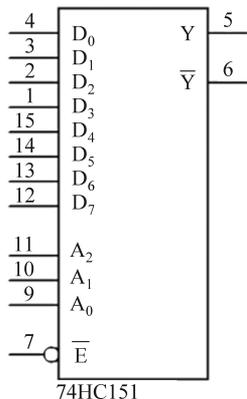


图 3.24 “八选一”的数据选择器 74HC151

“四选一”数据选择器 74153 的真值表如表 3.10 所示,从表 3.10(a)可看出,由地址信号 A_1A_0 决定选择哪一路输入信号进行输出,如 $A_1A_0=00$,输出信号 $Y=D_0$, Y 与 D_1 、 D_2 、 D_3 无关;又如 $A_1A_0=11$,输出信号 $Y=D_3$, Y 与 D_0 、 D_1 、 D_2 无关。表 3.10(b)是简化后的真值表,它与表 3.10(a)等价。

表 3.10 “四选一”数据选择器 74153 的真值表

(a) 功能表格式 1							(b) 功能表格式 2			
D_3	D_2	D_1	D_0	A_1	A_0	Y	D	A_1	A_0	Y
×	×	×	0	0	0	0	D_0	0	0	D_0
×	×	×	1	0	0	1	D_1	0	1	D_1
×	×	0	×	0	1	0	D_2	1	0	D_2
×	×	1	×	0	1	1	D_3	1	1	D_3
×	0	×	×	1	0	0				
×	1	×	×	1	0	1				
0	×	×	×	1	1	0				
1	×	×	×	1	1	1				

根据功能表 3.10,可以得到 74153 输出逻辑函数 Y ,其值为各地址信号的最小项与对应数据形成的与项,再相或的结果。依据输出函数表达式可得芯片 74153 的内部电路

图,如图 3.25 所示,可以看出具有开关功能的数据选择器,其内部电路也是由基本的门电路组成的。

$$\begin{aligned}
 Y &= f(D_3, D_2, D_1, D_0, A_1, A_0) \\
 &= \bar{A}_1 \bar{A}_0 \cdot D_0 + \bar{A}_1 A_0 \cdot D_1 + A_1 \bar{A}_0 \cdot D_2 + A_1 A_0 \cdot D_3 \\
 &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3
 \end{aligned}$$

同理可得,可得到“八选一”数据选择器 74151 的输出函数。

$$\begin{aligned}
 Y &= f(D_7, D_6, D_5, D_4, D_3, D_2, D_1, D_0, A_2, A_1, A_0) \\
 &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7 \\
 &= \bar{A}_2 \bar{A}_1 \bar{A}_0 \cdot D_0 + \bar{A}_2 \bar{A}_1 A_0 \cdot D_1 + \bar{A}_2 A_1 \bar{A}_0 \cdot D_2 + \bar{A}_2 A_1 A_0 \cdot D_3 + \\
 &\quad A_2 \bar{A}_1 \bar{A}_0 \cdot D_4 + A_2 \bar{A}_1 A_0 \cdot D_5 + A_2 A_1 \bar{A}_0 \cdot D_6 + A_2 A_1 A_0 \cdot D_7 \\
 &= \sum_{i=0}^7 m_i(A_2, A_1, A_0) \cdot D_i
 \end{aligned}$$

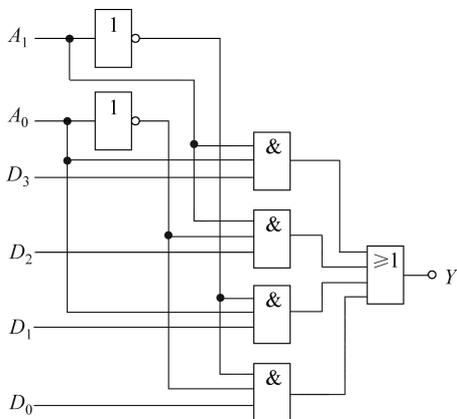


图 3.25 “四选一”数据选择器 74153 电路结构

例 3.8 分别使用“八选一”和“四选一”数据选择器 74HC151、74HC153 设计三变量多数判决器。

解: ① 设三变量分别为 A 、 B 、 C , 判决结果为 F , 依题意得到表 3.11 所示的真值表。

表 3.11 真值表

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
F	0	0	0	1	0	1	1	1

由真值表可得到三变量多数判决器对应的逻辑函数: $F(A, B, C) = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ 。