

组合逻辑电路是数字逻辑电路中的一种电路类型。

本章首先讲述组合逻辑电路的结构特点、功能特点及功能的描述方法,然后着重介绍组合逻辑电路的分析方法和设计方法,接着着重介绍了加法器、数值比较器、编码器、译码器、数据分配器及数据选择器 6 种常用组合逻辑电路的电路组成、工作原理、功能描述方法及特点,以及常用典型集成 MSI 组合逻辑电路的功能、使用方法。

学习本章内容应具备的相关知识是:逻辑表达式、真值表、逻辑图及相互转换方法;逻辑函数的化简;各种逻辑门的逻辑功能。

3.1 组合逻辑电路概述

3.1.1 组合逻辑电路的特点

在数字系统中,按电路逻辑功能的不同,将电路分为组合逻辑电路和时序逻辑电路。

组合逻辑电路简称组合电路,结构示意图如图 3.1 所示。

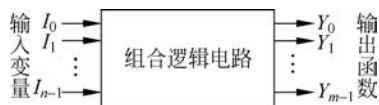


图 3.1 组合逻辑电路结构示意图

1. 组合逻辑电路的结构特点

在结构上,组合逻辑电路由门构成,且从输出端到输入端无反馈连接。

2. 组合逻辑电路的功能特点

在功能上,组合逻辑电路任一时刻的输出信号仅取决于该时刻的输入信号,而与输入信号作用前电路所处的状态无关。

输出函数与输入变量之间的逻辑关系可表示成如下函数关系式

$$Y_i = f_i(I_0, I_1, \dots, I_{n-1}) \quad (3.1)$$

3.1.2 组合逻辑电路逻辑功能的描述方法

真值表、卡诺图、逻辑表达式、逻辑图、波形图等方法均可用于描述组合逻辑电路的逻辑功能。

3.1.3 组合逻辑电路的分类

1. 按逻辑功能划分

组合逻辑电路可分为加法器、数值比较器、编码器、译码器、数据分配器、数据选择器等。

2. 按集成度划分

组合逻辑电路可分为 SSI、MSI、LSI、VLSI 等。

3.2 组合逻辑电路的分析方法和设计方法

3.2.1 组合逻辑电路的分析方法

1. 一般分析步骤

求解给定组合逻辑电路逻辑功能的过程称为组合逻辑电路的分析。组合逻辑电路分析的一般步骤如图 3.2 所示。

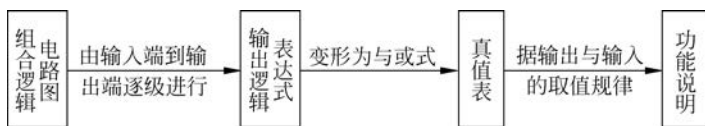


图 3.2 组合逻辑电路分析的一般步骤

有以下三点说明。

- (1) 分析的关键是各种功能描述方法的转换及根据真值表进行功能的正确说明。
- (2) 单输出组合逻辑电路由真值表中函数值为 1 时输入变量的取值规律进行功能说明。
- (3) 多输出组合逻辑电路要将几个输出综合在一起考虑与输入变量的取值关系进行功能说明。

2. 分析举例

【例 3.1】 试分析如图 3.3 所示的组合逻辑电路的逻辑功能。

【解】 如图 3.3 所示的组合逻辑电路由 4 个与非门构成, A 、 B 为 2 个输入变量, Y 为 1 个输出函数, 为单输出组合逻辑电路。

(1) 由逻辑图写出输出函数逻辑表达式:

$$\begin{aligned}
 Y &= \overline{A \cdot AB \cdot B \cdot AB} \\
 &= A \cdot \overline{AB} + B \cdot \overline{AB} \\
 &= A\overline{B} + \overline{A}B
 \end{aligned} \tag{3.2}$$

(2) 由逻辑表达式列出真值表, 如表 3.1 所示。

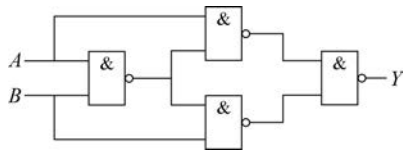


图 3.3 例 3.1 的组合逻辑电路

表 3.1 例 3.1 的真值表

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

列表方法: 输出函数逻辑表达式每个与项所覆盖的行填写 1 值, 重复时只填写一次 1 值, 剩余行填写 0 值。



视频讲解

(3) 由真值表进行功能说明:

由表 3.1 可知,当 A 、 B 输入变量取值相异时输出为 1; 当 A 、 B 输入变量取值相同时输出为 0。

电路实现异或功能。

【例 3.2】 试分析如图 3.4 所示的组合逻辑电路的逻辑功能。

【解】 如图 3.4 所示的组合逻辑电路由 4 个或非门构成, A 、 B 、 C 为 3 个输入变量, Y 为 1 个输出函数, 为单输出组合逻辑电路。

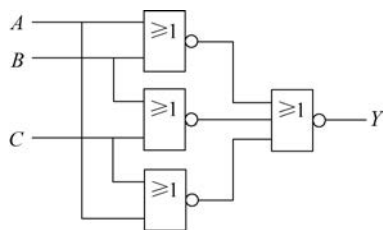


图 3.4 例 3.2 的组合逻辑电路

(1) 由逻辑图写出输出函数逻辑表达式:

$$\begin{aligned} Y &= \overline{A+B} + \overline{B+C} + \overline{A+C} \\ &= AB + BC + AC \end{aligned} \quad (3.3)$$

(2) 由逻辑表达式列出真值表, 如表 3.2 所示。

表 3.2 例 3.2 的真值表

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

(3) 由真值表进行功能说明:

由表 3.2 可知, 当 A 、 B 、 C 3 个输入变量的取值中 1 的个数占多数时, 输出 $Y=1$, 否则 $Y=0$ 。

电路实现三变量多数一致(多数表决)功能。

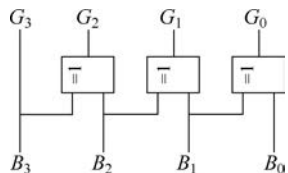


图 3.5 例 3.3 的组合逻辑电路

【例 3.3】 试分析如图 3.5 所示的组合逻辑电路的逻辑功能。

【解】 如图 3.5 所示的组合逻辑电路由 3 个异或门构成, B_3 、 B_2 、 B_1 、 B_0 为 4 个输入变量, G_3 、 G_2 、 G_1 、 G_0 为 4 个输出函数, 为多输出组合逻辑电路。

(1) 由逻辑图写出输出函数逻辑表达式:

$$\begin{cases} G_3 = B_3 \\ G_2 = B_3 \oplus B_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2 \\ G_1 = B_2 \oplus B_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1 \\ G_0 = B_1 \oplus B_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0 \end{cases} \quad (3.4)$$

(2) 由逻辑表达式(3.4)列出真值表, 如表 3.3 所示。

表 3.3 例 3.3 的真值表

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1

续表

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0

(3) 由真值表进行功能说明:

由表 3.3 可知,输入变量 $B_3B_2B_1B_0$ 的取值为 4 位二进制数的变化规律,输出函数 $G_3G_2G_1G_0$ 的取值为 4 位格雷码的变化规律。

电路实现将 4 位二进制代码转换成 4 位格雷码。



视频讲解

3.2.2 组合逻辑电路的设计方法

1. 一般设计步骤

根据给定的功能要求,求组合逻辑电路的过程称为组合逻辑电路的设计。

用 SSI 逻辑门设计组合逻辑电路时,要求所用门的数量最少,设计的一般步骤如图 3.6 所示。

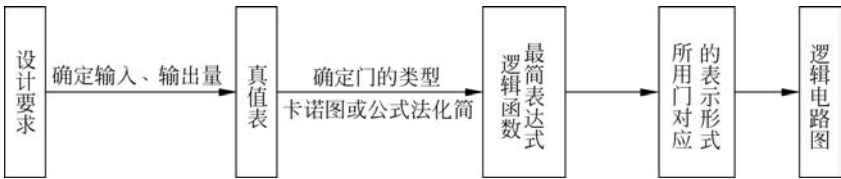


图 3.6 组合逻辑电路设计的一般步骤

有以下两点说明。

(1) 设计的关键是正确分析设计要求,确定有几个输入变量、几个输出函数,以及输出函数与输入变量之间的因果关系,必要时需规定 0、1 所表示的状态。

(2) 按门的类型进行函数化简,如用与门和或门、与非门实现,化简成最简与或式再变形;用或非门、与或非门等实现,先化简成反函数的最简与或式再变形。

2. 设计举例

【例 3.4】 试设计一个 1 位十进制数数值范围判别电路,十进制数用 8421 码表示,当输入的十进制数大于或等于 5 时,输出为 1,否则输出为 0。分别用与非门、或非门实现。

【解】 (1) 8421 码的 4 个输入变量用 $A、B、C、D$ 表示,一个输出函数用 Y 表示。

(2) 按输入的十进制数大于或等于 5 时输出为 1、否则输出为 0 的关系列出真值表,如表 3.4 所示,其中将非 8421 码按无关项处理。

表 3.4 例 3.4 的真值表

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

续表

A	B	C	D	Y	A	B	C	D	Y
0	0	1	0	0	1	0	1	0	×
0	0	1	1	0	1	0	1	1	×
0	1	0	0	0	1	1	0	0	×
0	1	0	1	1	1	1	0	1	×
0	1	1	0	1	1	1	1	0	×
0	1	1	1	1	1	1	1	1	×

(3) 用与非门实现的方案。

① 由真值表画出逻辑函数的卡诺图并圈 1 格化简,如图 3.7 所示,得最简与或逻辑表达式并变形为与非-与非式

$$Y = A + BC + BD = \overline{\overline{A} \cdot \overline{BC} \cdot \overline{BD}} \quad (3.5)$$

② 由逻辑表达式(3.5)画出用与非门实现的逻辑图,如图 3.8 所示。

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	×	×	×	×
10	1	1	×	×

图 3.7 例 3.4 的卡诺图及化简方案一

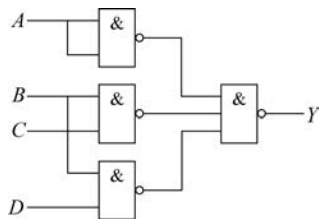


图 3.8 例 3.4 用与非门实现的逻辑图

(4) 用或非门实现的方案。

① 由真值表画出逻辑函数的卡诺图并圈 0 格化简,如图 3.9 所示,得反函数的最简逻辑表达式再变换为函数的或非-或非式

$$\bar{Y} = \bar{A}\bar{B} + \bar{A}\bar{C}\bar{D} \quad (3.6)$$

$$Y = \overline{\bar{A}\bar{B} + \bar{A}\bar{C}\bar{D}} = \overline{\bar{A} + B + \bar{A} + C + \bar{D}} \quad (3.7)$$

② 由逻辑表达式(3.7)画出用或非门实现的逻辑图,如图 3.10 所示。

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	×	×	×	×
10	1	1	×	×

图 3.9 例 3.4 的卡诺图及化简方案二

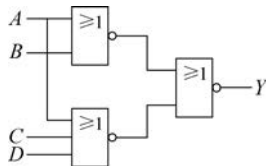


图 3.10 例 3.4 用或非门实现的逻辑图

【例 3.5】 试用最少的门设计一个三变量奇数判别电路,当三个输入变量的取值中有奇数个 1 时输出为 1,否则输出为 0。

【解】 (1) 三个输入变量用 A、B、C 表示,一个输出函数用 Y 表示。

(2) 按三个输入变量的取值中有奇数个 1 时输出为 1 的逻辑关系,列出真值表如表 3.5 所示。

表 3.5 例 3.5 的真值表

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

(3) 由真值表画出逻辑函数的卡诺图并化简,如图 3.11 所示,得最简逻辑表达式并变形为

$$\begin{aligned}
 Y &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\
 &= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC) \\
 &= \bar{A}(B \oplus C) + A \cdot \overline{B \oplus C} \\
 &= A \oplus (B \oplus C)
 \end{aligned} \quad (3.8)$$

(4) 由逻辑表达式(3.8)画出用异或门实现的逻辑图,如图 3.12 所示。

		BC			
		00	01	11	10
A	0	0	①	0	①
	1	①	0	①	0

图 3.11 例 3.5 的卡诺图及化简

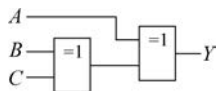


图 3.12 例 3.5 的逻辑图

【说明】 变量取值组合中有奇数个 1 或偶数个 1 的 2^{n-1} 个最小项构成的逻辑函数标准与或式(n 为变量个数),可用逻辑代数的基本公式转换为异或式并用异或门实现。

【例 3.6】 试用与非门和非门设计一个运算电路,输入为 2 位二进制数,输出为输入的平方。

【解】 (1) 用 A_1A_0 表示 2 位二进制数的输入变量;因最大输出数为 9,输出函数应为 4 位二进制数,用 $Y_3Y_2Y_1Y_0$ 表示。

(2) 按乘法运算的规则,列出真值表如表 3.6 所示。

表 3.6 例 3.6 的真值表

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

(3) 由真值表写出函数 Y_3 、 Y_2 、 Y_1 、 Y_0 的逻辑表达式,并化简变形得

$$\begin{cases}
 Y_3 = A_1A_0 = \overline{\overline{A_1A_0}} \\
 Y_2 = A_1\bar{A}_0 = \overline{\overline{A_1}\bar{A}_0} \\
 Y_1 = 0 \\
 Y_0 = \bar{A}_1A_0 + A_1A_0 = A_0(\bar{A}_1 + A_1) = A_0
 \end{cases} \quad (3.9)$$

(4) 由逻辑表达式(3.9)画出用与非门和非门实现的逻辑图如图 3.13 所示。

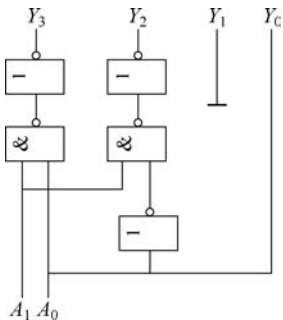


图 3.13 例 3.6 的逻辑图

【例 3.7】 如图 3.14 所示,在水箱中 A、B、C 三处安置三个水位检测元件,当水面高于检测元件时,检测元件输出高电平,水面低于检测元件时,检测元件输出低电平。试用与非

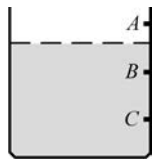


图 3.14 例 3.7 的水箱示意图

门设计一个水位状态显示电路。要求：当水面在 A、B 之间的正常状态时，仅绿灯 G 亮；水面在 B、C 间或 A 以上的异常状态时，仅黄灯 Y 亮；水面在 C 以下的危险状态时，仅红灯 R 亮。

【解】 (1) 状态赋值。检测元件 A、B、C 输出高电平用逻辑 1 表示，输出低电平用逻辑 0 表示。指示灯 G、Y、R 亮用逻辑 1 表示，不亮用逻辑 0 表示。

(2) 根据题意，列出真值表如表 3.7 所示。其中，将不可能出现的状态所对应的输入取值组合 010、100、101、110 按无关项处理。

表 3.7 例 3.7 的真值表

A	B	C	G	Y	R
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	×	×	×
0	1	1	1	0	0
1	0	0	×	×	×
1	0	1	×	×	×
1	1	0	×	×	×
1	1	1	0	1	0

(3) 由真值表画出逻辑函数的卡诺图并圈 1 格化简，如图 3.15 所示，得最简与或逻辑表达式并变形为与非-与非式。

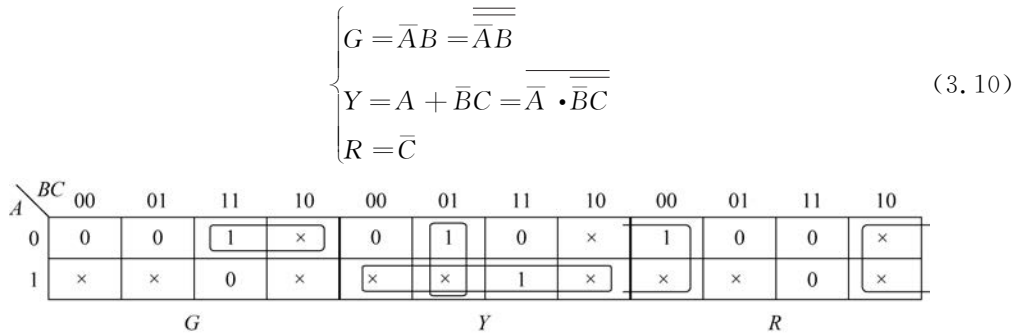


图 3.15 例 3.7 的卡诺图及化简

(4) 由逻辑表达式(3.10)画出用与非门实现的逻辑图，如图 3.16 所示。

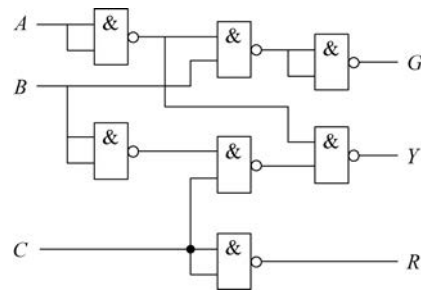


图 3.16 例 3.7 的逻辑图

3.3 常用组合逻辑电路

3.3.1 加法器



视频讲解

在数字系统中,加法器是运算器中的一种典型运算电路。

1 位全加器是组成加法器的基础,而 1 位半加器是全加器的基础。下面分别介绍 1 位半加器、1 位全加器及加法器的原理。

1. 1 位半加器

不考虑来自低位的进位数,仅对被加数和加数两个 1 位二进制数进行算术相加的运算称为半加运算。实现半加运算的电路称为 1 位半加器,简称半加器。

半加器的逻辑设计如下。

(1) 输入变量为被加数 A_i 、加数 B_i ；输出函数为本位和数 S_i 和向高位的进位数 C_{i+1} 。

(2) 按算术加法运算的规则,列出真值表如表 3.8 所示。

表 3.8 半加器的真值表

A_i	B_i	S_i	C_{i+1}	A_i	B_i	S_i	C_{i+1}
0	0	0	0	1	0	1	0
0	1	1	0	1	1	0	1

(3) 由真值表写出函数 S_i, C_{i+1} 最简逻辑表达式并变形得

$$\begin{cases} S_i = \bar{A}_i B_i + A_i \bar{B}_i = A_i \oplus B_i \\ C_{i+1} = A_i B_i \end{cases} \quad (3.11)$$

(4) 由逻辑表达式(3.11)画出用异或门和与门实现的逻辑图,如图 3.17 所示。半加器的图形符号如图 3.18 所示。

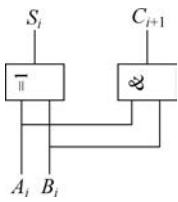


图 3.17 半加器的逻辑图

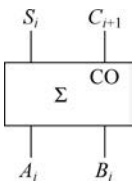


图 3.18 半加器的图形符号

2. 1 位全加器

对被加数、加数及来自低位的进位数三个 1 位二进制数进行算术相加的运算称为全加运算。实现全加运算的电路称为 1 位全加器,简称全加器。

全加器的逻辑设计如下。

(1) 输入变量为被加数 A_i 、加数 B_i 和来自低位的进位数 C_i ；输出函数为本位和数 S_i 和向高位的进位数 C_{i+1} 。

(2) 按算术加法运算的规则,列出真值表如表 3.9 所示。

表 3.9 全加器的真值表

A_i	B_i	C_i	S_i	C_{i+1}
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

(3) 由真值表画出逻辑函数的卡诺图并化简,如图 3.19 所示,得最简逻辑表达式并变形为

$$\begin{cases} S_i = \bar{A}_i \bar{B}_i C_i + \bar{A}_i B_i \bar{C}_i + A_i \bar{B}_i \bar{C}_i + A_i B_i C_i = A_i \oplus B_i \oplus C_i \\ C_{i+1} = A_i B_i + A_i C_i + B_i C_i = \overline{\bar{A}_i \bar{B}_i + \bar{A}_i \bar{C}_i + \bar{B}_i \bar{C}_i} \end{cases} \quad (3.12)$$

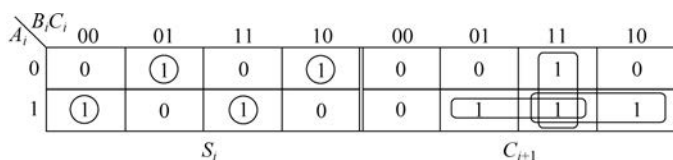


图 3.19 全加器的卡诺图及化简

(4) 由逻辑表达式(3.12)画出用异或门、与或非门及非门实现的逻辑图,如图 3.20 所示。全加器的图形符号如图 3.21 所示。

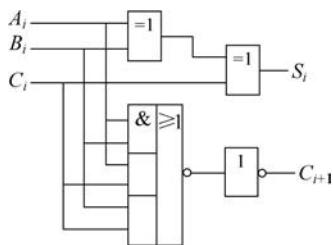


图 3.20 全加器的逻辑图

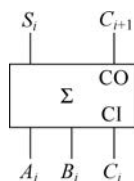


图 3.21 全加器的图形符号

3. 加法器

实现对多位二进制数进行算术加法运算的电路称为加法器。按进位方式不同,又分为串行进位加法器和并行进位加法器两种。

1) 串行进位加法器

串行进位加法器(ripple-carry adder)由多个全加器构成,低位全加器的进位输出端连接相邻高位全加器的进位输入端,进位信号由低位到高位逐位传递,称为串行进位或行波进位。串行进位加法器的特点是结构简单、运算速度慢。图 3.22 为 4 位串行进位加法器的逻辑图。

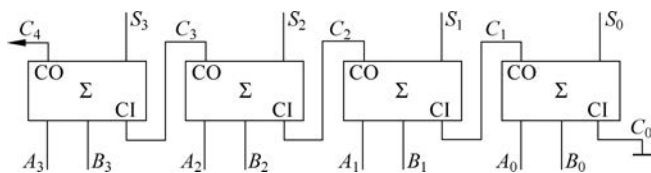


图 3.22 4 位串行进位加法器的逻辑图

2) 并行进位加法器

并行进位亦称超前进位,各级的进位可同时产生,即通过逻辑电路事先确定各位的进位为 0 或 1,高位运算时不必等待低位的进位,各位运算同时进行。并行进位加法器(carry look-ahead adder)的特点是结构复杂、运算速度快。

74LS283 为集成 4 位并行进位加法器,图形符号如图 3.23 所示。其中, $A_3 A_2 A_1 A_0$ 为数 A 输入端, $B_3 B_2 B_1 B_0$ 为数 B 输入端, CI 为进位输入端, $S_3 S_2 S_1 S_0$ 为和输出端, CO 为进位输出端。运算关系为

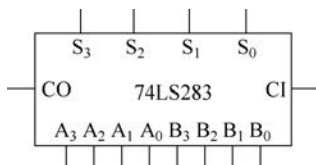


图 3.23 74LS283 的图形符号

$$S_3 S_2 S_1 S_0 = A_3 A_2 A_1 A_0 + B_3 B_2 B_1 B_0 + CI \quad (3.13)$$

并产生 CO 。

3.3.2 数值比较器

数值比较器是用来比较两个相同位数二进制数大小、相等关系的逻辑电路。

1. 1 位数值比较器

1 位数值比较器的逻辑设计如下。

(1) 输入变量为两个相比较的 1 位二进制数 A_i 、 B_i ; 输出函数为比较的结果: $Y_{i(A>B)}$ 表示 $A_i > B_i$ 的比较结果, $Y_{i(A=B)}$ 表示 $A_i = B_i$ 的比较结果, $Y_{i(A<B)}$ 表示 $A_i < B_i$ 的比较结果。

(2) 按数的大小、相等关系,列出真值表如表 3.10 所示。

(3) 由真值表写出各输出函数最简逻辑表达式并变形得

$$\begin{cases} Y_{i(A>B)} = A_i \bar{B}_i \\ Y_{i(A<B)} = \bar{A}_i B_i \\ Y_{i(A=B)} = \bar{A}_i \bar{B}_i + A_i B_i = \overline{A_i \bar{B}_i + \bar{A}_i B_i} = \overline{Y_{i(A>B)} + Y_{i(A<B)}} \end{cases} \quad (3.14)$$

(4) 由逻辑表达式(3.14)画出用非门、与门及或非门实现的逻辑图,如图 3.24 所示。

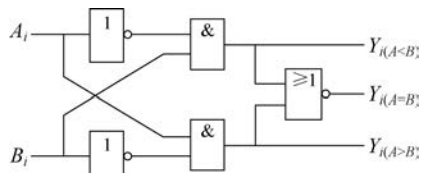


图 3.24 1 位数值比较器的逻辑图

表 3.10 1 位数值比较器的真值表

A_i	B_i	$Y_{i(A>B)}$	$Y_{i(A=B)}$	$Y_{i(A<B)}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



视频讲解

2. 集成 4 位数值比较器

多位数值比较的原理是从高位开始比较,高位能确定大小关系时不用比较低位,高位相等时比较次高位,以此类推。各位都相等时两个数才相等。

74LS85 是设置级联输入端的集成 4 位数值比较器,图形符号如图 3.25 所示。其中, $A_3A_2A_1A_0$ 为数 A 输入端; $B_3B_2B_1B_0$ 为数 B 输入端; $A>B$ 、 $A=B$ 、 $A<B$ 为级联输入端,是比 A_0 、 B_0 更低位的比较结果; $Y_{(A>B)}$ 、 $Y_{(A=B)}$ 、 $Y_{(A<B)}$ 为比较结果输出端。

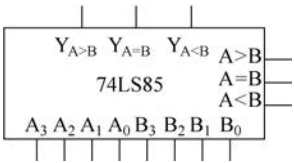


图 3.25 74LS85 的图形符号

74LS85 的真值表如表 3.11 所示。

表 3.11 集成 4 位数值比较器 74LS85 的真值表

比 较 输 入				级 联 输 入			输 出		
A_3B_3	A_2B_2	A_1B_1	A_0B_0	$A>B$	$A<B$	$A=B$	$Y_{A>B}$	$Y_{A<B}$	$Y_{A=B}$
$A_3>B_3$	×	×	×	×	×	×	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

【说明】 按从高位开始比较,高位能确定大小关系时不用比较低位,高位相等时比较次高位的原理简化列写真值表。特点是既压缩了真值表的规模,又直观地表示了数值比较情况。

多片 74LS85 可级联扩大数的比较范围。级联的连接规则是:高位片的级联输入端接低位片的输出端;最低位片的级联输入端接常数: $A>B$ 端、 $A<B$ 端分别接 0, $A=B$ 端接 1。

图 3.26 为用 2 片集成 4 位数值比较器 74LS85 组成 8 位数值比较器的逻辑图,根据表 3.11 可以很容易地理解工作原理,这里不再叙述。

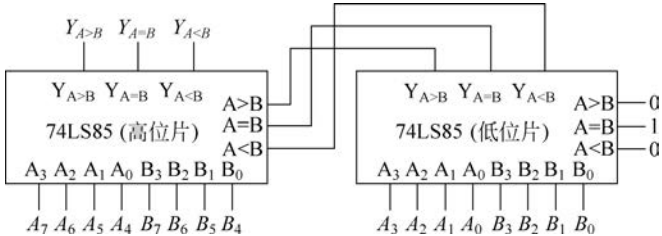


图 3.26 2 片 74LS85 组成 8 位数值比较器的逻辑图

3.3.3 编码器

用按一定规律排列的 0 和 1 二进制数作代码表示十进制数的过程称为编码。实现编码



视频讲解

的电路称为编码器。编码器有二进制编码器、二-十进制编码器和优先编码器等。如编码器有 8 个输入端和 3 个输出端,称为 8 线-3 线编码器;如编码器有 10 个输入端和 4 个输出端,称为 10 线-4 线编码器,以此类推。

1. 二进制编码器

用 n 位二进制代码对 2^n 个十进制数进行编码的电路称为二进制编码器。

二进制编码器的特点是,输入变量有约束,任一时刻只允许一个输入信号有效,即输入变量互相排斥。

3 位二进制编码器的逻辑设计如下。

(1) 输入变量为 0 ~ 7 八个十进制数,用 $I_0 \sim I_7$ 表示,1 输入有效;输出函数为 3 位二进制代码,用 $Y_2 \sim Y_0$ 表示。

(2) 用二进制数表示十进制数,列出真值表如表 3.12 所示。

表 3.12 3 位二进制编码器的真值表

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

【说明】 由输入变量互斥的约束特点,用一行表示对一个有效的输入信号进行编码的方法简化列写真值表。特点是既压缩了真值表的规模,又直观地表示了编码情况。

(3) 由真值表写出函数 Y_2 、 Y_1 、 Y_0 的最简逻辑表达式。

变量个数较多时无法用卡诺图化简。变量互斥时加入约束条件使用公式法对最小项进行简化,分析如下。如

$$\begin{aligned}
 \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7 &= \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7 + I_0 I_7 + I_1 I_7 + I_2 I_7 + I_3 I_7 + I_4 I_7 + I_5 I_7 + I_6 I_7 \\
 &= I_7 (\bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 + I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6) \\
 &= I_7
 \end{aligned} \quad (3.15)$$

所以有

$$\begin{cases} Y_2 = I_4 + I_5 + I_6 + I_7 \\ Y_1 = I_2 + I_3 + I_6 + I_7 \\ Y_0 = I_1 + I_3 + I_5 + I_7 \end{cases} \quad (3.16)$$

(4) 由逻辑表达式(3.16)画出用或门实现的逻辑图,如图 3.27 所示。

【说明】 对 I_0 的编码是隐含的,当输入变量 $I_1 \sim I_7$ 均无效为 0 时,输出 $Y_2 Y_1 Y_0 = 000$,就是 I_0 的编码。

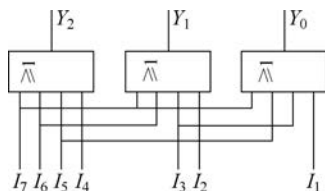


图 3.27 3 位二进制编码器的逻辑图

2. 二-十进制编码器

用4位二进制代码对0~9一位十进制数进行编码的电路称为二-十进制编码器。

8421码二-十进制编码器的逻辑设计如下。

(1) 输入变量为0~9十个十进制数,用 $I_0 \sim I_9$ 表示,1输入有效,任一时刻只允许一个输入信号有效;输出函数为4位8421码,用 $Y_3 \sim Y_0$ 表示。

(2) 按8421码的编码规则列出简化形式的真值表,如表3.13所示。

表 3.13 8421 码编码器的真值表

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	Y_3	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

(3) 由真值表写出函数 Y_3 、 Y_2 、 Y_1 、 Y_0 的最简逻辑表达式

$$\begin{cases} Y_3 = I_8 + I_9 \\ Y_2 = I_4 + I_5 + I_6 + I_7 \\ Y_1 = I_2 + I_3 + I_6 + I_7 \\ Y_0 = I_1 + I_3 + I_5 + I_7 + I_9 \end{cases} \quad (3.17)$$

(4) 由逻辑表达式(3.17)画出用或门实现的逻辑图,如图3.28所示。

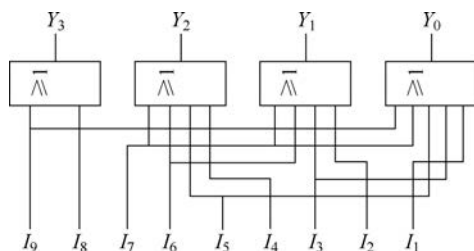


图 3.28 8421 码编码器的逻辑图

【说明】 对 I_0 的编码是隐含的,当输入变量 $I_1 \sim I_9$ 均无效为0时,输出 $Y_3 Y_2 Y_1 Y_0 = 0000$,就是 I_0 的编码。

3. 优先编码器

优先编码器(priority encoder)允许同时输入两个及两个以上编码信号,但只对其中一个优先级最高的输入信号进行编码。

74LS148是8线-3线集成二进制优先编码器,图形符号如图3.29所示。其中, $\overline{EI}$ 为选通输入端,0有

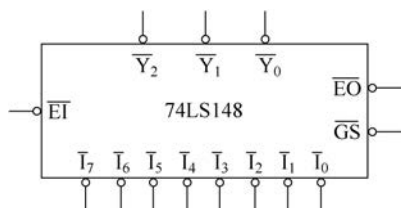


图 3.29 74LS148 的图形符号

效； $\overline{EO}$ 为选通输出端； $\overline{GS}$ 为扩展输出端； $\overline{I}_0 \sim \overline{I}_7$ 为编码信号输入端，0 有效，大数优先级高； $\overline{Y}_2 \sim \overline{Y}_0$ 为输出端，反码输出。注意， $\overline{ET}$ 、 $\overline{EO}$ 、 $\overline{GS}$ 、 $\overline{I}_i$ 中的横线“ $\overline{}$ ”表示 0 有效，不是非号。

74LS148 简化形式的真值表如表 3.14 所示。

表 3.14 8 线-3 线二进制优先编码器 74LS148 的真值表

$\overline{EI}$	$\overline{I}_0$	$\overline{I}_1$	$\overline{I}_2$	$\overline{I}_3$	$\overline{I}_4$	$\overline{I}_5$	$\overline{I}_6$	$\overline{I}_7$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$	$\overline{EO}$	$\overline{GS}$
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	×	×	×	×	×	×	×	0	0	0	0	1	0
0	×	×	×	×	×	×	0	1	0	0	1	1	0
0	×	×	×	×	×	0	1	1	0	1	0	1	0
0	×	×	×	×	0	1	1	1	0	1	1	1	0
0	×	×	×	0	1	1	1	1	1	0	0	1	0
0	×	×	0	1	1	1	1	1	1	0	1	1	0
0	×	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

图 3.30 为用 2 片 8 线-3 线集成二进制优先编码器组成 16 线-4 线二进制优先编码器的逻辑图。根据表 3.14 可以很容易地理解工作原理，这里不再叙述。

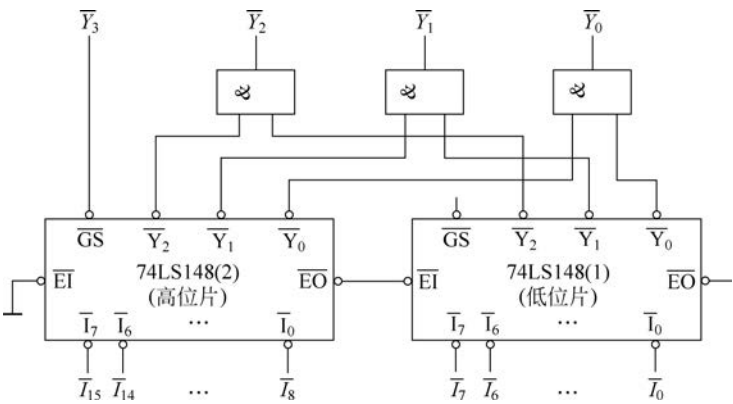


图 3.30 2 片 74LS148 组成 16 线-4 线二进制优先编码器

3.3.4 译码器

将代码译成表示十进制数输出信号的过程称为译码。实现译码的电路称为译码器 (decoder)。译码器有二进制译码器、二十进制译码器和显示译码器等。

译码器的输入信号为二进制代码，输出信号为对应输入二进制代码的十进制数输出信号。输出信号的形式有直接表示十进制数的高、低电平信号及通过显示器件表示十进制输出的显示控制信号。如译码器有 3 个输入端和 8 个输出端，称为 3 线-8 线译码器；如译码器



视频讲解

有 4 个输入端和 10 个输出端,称为 4 线-10 线译码器,以此类推。

1. 二进制译码器

将 n 位二进制代码译成 2^n 个十进制数的译码器称为二进制译码器。

二进制译码器特点是:每输入一组代码,多个输出端中仅一个输出端有信号输出。

1) 2 位二进制译码器

2 位二进制译码器的逻辑设计如下。

(1) 输入变量为 2 位二进制代码,用 A_1 、 A_0 表示;输出函数为 4 个十进制数,用 $Y_3 \sim Y_0$ 表示,1 输出有效。

(2) 按二进制代码和十进制数的对应关系,列出真值表如表 3.15 所示。

表 3.15 2 位二进制译码器的真值表

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

(3) 由真值表写出函数 Y_3 、 Y_2 、 Y_1 、 Y_0 的最简逻辑表达式

$$\begin{cases} Y_3 = A_1 A_0 \\ Y_2 = A_1 \bar{A}_0 \\ Y_1 = \bar{A}_1 A_0 \\ Y_0 = \bar{A}_1 \bar{A}_0 \end{cases} \quad (3.18)$$

(4) 由逻辑表达式(3.18)画出用与门、非门实现的逻辑图如图 3.31 所示。

2 位二进制译码器也称为 2 线-4 线译码器。

2) 集成 3 线-8 线译码器

74LS138 是设置选通控制端的集成 3 线-8 线译码器,图形符号如图 3.32 所示。其中, S_1 、 $\bar{S}_2$ 、 $\bar{S}_3$ 为选通控制端; A_2 、 A_1 、 A_0 为代码输入端,或称为地址输入端; $\bar{Y}_0 \sim \bar{Y}_7$ 为输出端,0 输出有效。注意, $\bar{S}_i$ 、 $\bar{Y}_i$ 中的横线“ $\bar{}$ ”表示 0 有效,不是非号。

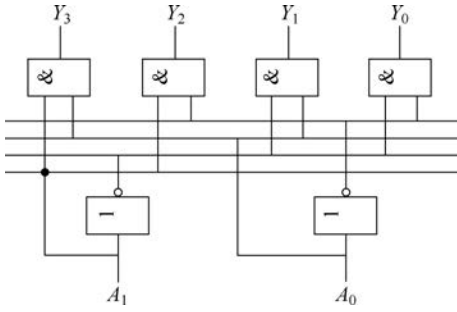


图 3.31 2 位二进制译码器的逻辑图

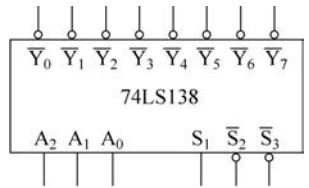


图 3.32 74LS138 的图形符号

74LS138 的真值表如表 3.16 所示。由真值表可写出选通控制端 $S_1 = 1$ 、 $\bar{S}_2 = \bar{S}_3 = 0$ 时的输出逻辑表达式为

$$\left\{ \begin{array}{l} \bar{Y}_0 = \overline{A_2 A_1 A_0} \\ \bar{Y}_1 = \overline{A_2 A_1 A_0} \\ \bar{Y}_2 = \overline{A_2 A_1 A_0} \\ \bar{Y}_3 = \overline{A_2 A_1 A_0} \\ \bar{Y}_4 = \overline{A_2 A_1 A_0} \\ \bar{Y}_5 = \overline{A_2 A_1 A_0} \\ \bar{Y}_6 = \overline{A_2 A_1 A_0} \\ \bar{Y}_7 = \overline{A_2 A_1 A_0} \end{array} \right. \quad (3.19)$$

输出逻辑表达式可表示成一般形式

$$\bar{Y}_i = \bar{m}_i \quad (i = 0 \sim 7) \quad (3.20)$$

其中, m_i 是地址输入变量 A_2 、 A_1 、 A_0 构成的最小项。译码的结果是地址输入变量构成最小项的反函数。

表 3.16 3 线-8 线译码器 74LS138 的真值表

S_1	$\bar{S}_2 + \bar{S}_3$	A_2	A_1	A_0	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$
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

当选通控制端不是 $S_1 = 1$ 、 $\bar{S}_2 = \bar{S}_3 = 0$ 时, 译码器禁止译码, 各输出端 $\bar{Y}_0 \sim \bar{Y}_7$ 均输出 1。

3) 集成译码器的扩展

74LS138 选通控制端不仅可以控制译码器的工作状态, 还可以进行译码范围扩展。

图 3.33 为用 2 片 74LS138 扩展构成 4 线-16 线译码器的一种连接方案。其中, 4 位二进制代码的最高位 A_3 控制各片的部分选通控制端, 4 位二进制代码的低 3 位 A_2 、 A_1 、 A_0 接至各片的片内代码输入端。 $\bar{S}$ 为所构成 4 线-16 线译码器的选通控制端。

$\bar{S} = 0$ 时的译码过程为: $A_3 A_2 A_1 A_0$ 的变化范围为 0000~0111 时, 片 1 工作译码, $\bar{Y}_0 \sim \bar{Y}_7$ 端有输出; 而片 2 禁止译码, $\bar{Y}_8 \sim \bar{Y}_{15}$ 端均输出 1。 $A_3 A_2 A_1 A_0$ 的变化范围为 1000~1111 时, 片 2 工作译码, $\bar{Y}_8 \sim \bar{Y}_{15}$ 端有输出; 而片 1 禁止译码, $\bar{Y}_0 \sim \bar{Y}_7$ 端均输出 1。

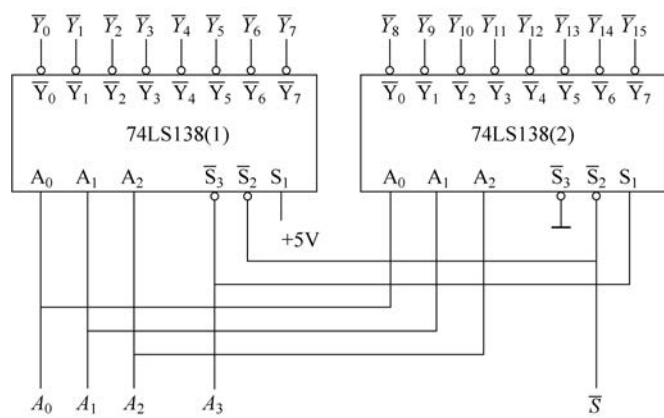


图 3.33 2 片 74LS138 组成 4 线-16 线译码器的逻辑图

2. 二-十进制译码器

将 4 位 BCD 码译成 10 个十进制数的译码器称为二-十进制译码器,亦称为 4 线-10 线译码器。

74LS42 是 8421 码集成二-十进制译码器,图形符号如图 3.34 所示。其中, A_3 、 A_2 、 A_1 、 A_0 为代码输入端,或称为地址输入端; $\bar{Y}_0 \sim \bar{Y}_9$ 为输出端,0 输出有效。

74LS42 的真值表如表 3.17 所示。由真值表可看出,74LS42 对 8421 码以外的 6 个伪码 1010~1111 拒绝翻译,当伪码输入时, $\bar{Y}_0 \sim \bar{Y}_9$ 均输出为 1,不会出现误译码。

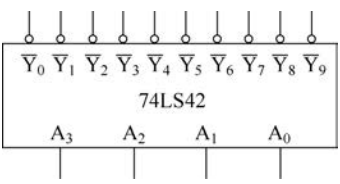


图 3.34 74LS42 的图形符号

表 3.17 4 线-10 线译码器 74LS42 的真值表

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	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

由真值表可写出输出逻辑表达式为

$$\begin{cases} \overline{Y}_0 = \overline{A_3 A_2 A_1 A_0} \\ \overline{Y}_1 = \overline{A_3 A_2 A_1 A_0} \\ \vdots \\ \overline{Y}_9 = \overline{A_3 A_2 A_1 A_0} \end{cases} \quad (3.21)$$

输出逻辑表达式可表示成一般形式

$$\overline{Y}_i = \overline{m}_i \quad (i = 0 \sim 9) \quad (3.22)$$

其中, m_i 是地址输入变量 A_3 、 A_2 、 A_1 、 A_0 构成的最小项。译码的结果是地址输入变量构成最小项的反函数。

由表 3.17 可以看出, 当 74LS42 的输出 $\overline{Y}_8$ 、 $\overline{Y}_9$ 不使用, 并将 A_3 作为选通控制端时, 构成 0 输出有效的 3 线-8 线译码器。

3. 显示译码器

在数字系统中, 经常需要将二进制编码翻译成人们习惯的十进制数形式, 直观地显示出来, 七段字符显示器是常见的数码显示器件。显示译码器由七段字符显示器和译码器构成。

1) 七段字符显示器

七段字符显示器又称七段数码显示器, 由七段可发光的线段构成, 利用发光段的不同组合方式显示“0~9”十进制数, 如图 3.35 所示。



图 3.35 七段字符显示器及显示的数字

按发光原理不同, 显示器件分为 LED(Light Emitting Diode)显示器(半导体显示器)和 LCD(Liquid Crystal Display)显示器(液晶显示器)。

(1) LED 显示器。

用发光二极管(LED)组成字形显示数字, 每个 LED 为一个发光段, 有共阳极和共阴极两种接法, 如图 3.36 所示。

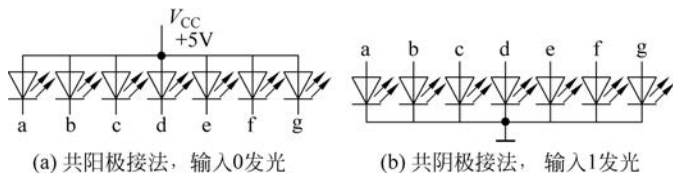


图 3.36 七段 LED 的两种接法

(2) LCD 显示器。

液晶是一种既具有液体的流动性又具有晶体光学特性的有机化合物。它的透明度和显示的颜色受外电场的控制。依靠在外界电场作用下产生的光电效应, 调制外界光线使液晶不同部位显现出反差, 显示出字形, 如图 3.37 所示。

2) 4 线-七段译码器/驱动器

74LS248 是集成 4 线-七段译码器/驱动器, 1 输出有效, 用于驱动共阴极 LED 数码显示

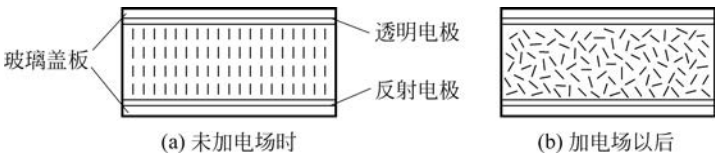


图 3.37 液晶的显示原理

器,图形符号如图 3.38 所示。

其中, $A_3 \sim A_0$ 为译码地址输入端, $\overline{LT}$ 为灯测试输入端(0 有效); $\overline{RBI}$ 为脉冲消隐输入端(0 有效); $\overline{BI}/\overline{RBO}$ 为消隐输入(0 有效)/脉冲消隐输出(0 有效); $Y_a \sim Y_g$ 为控制七段 LED 的输出端。

74LS248 的真值表如表 3.18 所示。

表 3.18 4 线-七段译码器/驱动器 74LS248 的真值表

十进制 或功能	输 入						$\overline{BI}/\overline{RBO}$	输 出						
	$\overline{LT}$	$\overline{RBI}$	A_3	A_2	A_1	A_0		Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0
1	1	×	0	0	0	1	1	0	1	1	0	0	0	0
2	1	×	0	0	1	0	1	1	1	0	1	1	0	1
3	1	×	0	0	1	1	1	1	1	1	1	0	0	1
4	1	×	0	1	0	0	1	0	1	1	0	0	1	1
5	1	×	0	1	0	1	1	1	0	1	1	0	1	1
6	1	×	0	1	1	0	1	1	0	1	1	1	1	1
7	1	×	0	1	1	1	1	1	1	1	0	0	0	0
8	1	×	1	0	0	0	1	1	1	1	1	1	1	1
9	1	×	1	0	0	1	1	1	1	1	1	0	1	1
10	1	×	1	0	1	0	1	0	0	0	1	1	0	1
11	1	×	1	0	1	1	1	0	0	1	1	0	0	1
12	1	×	1	1	0	0	1	0	1	0	0	0	1	1
13	1	×	1	1	0	1	1	1	0	0	1	0	1	1
14	1	×	1	1	1	0	1	0	0	0	1	1	1	1
15	1	×	1	1	1	1	1	0	0	0	0	0	0	0
消隐	×	×	×	×	×	×	0	0	0	0	0	0	0	0
脉冲消隐	1	0	0	0	0	0	0	0	0	0	0	0	0	0
灯测试	0	×	×	×	×	×	1	1	1	1	1	1	1	1

用 74LS248 控制 1 位 LED 显示电路如图 3.39 所示。

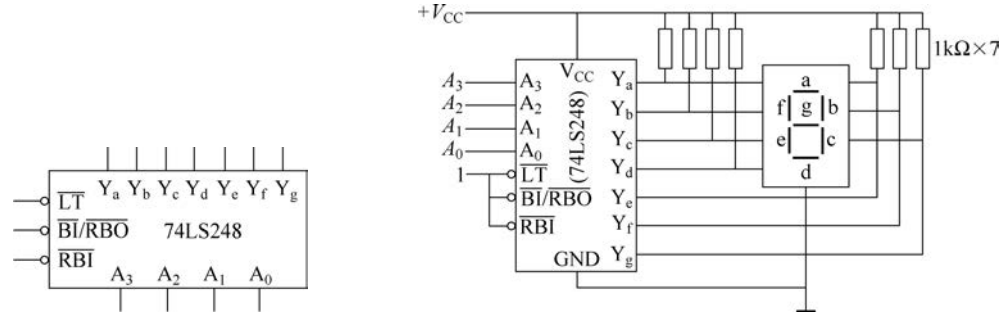


图 3.38 74LS248 的图形符号

图 3.39 74LS248 控制 1 位 LED 显示电路

3.3.5 数据分配器

在分配控制信号(也称地址信号)的作用下,将一个数据分配到多个输出端中的某个输出端的逻辑电路称为数据分配器。数据分配器有1个数据输入端、 n 个分配控制信号和 2^n 个数据输出端。功能示意图如图3.40所示。

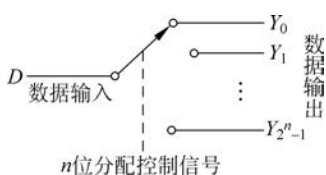


图 3.40 数据分配器的示意图

1 路-4 路数据分配器的逻辑设计如下。

(1) 输入变量及输出函数。1 个数据信号,用 D 表示; 2 个分配控制信号,用 A_1 、 A_0 表示; 4 个输出函数信号,用 Y_0 、 Y_1 、 Y_2 、 Y_3 表示。

(2) 按控制分配关系,列出真值表如表 3.19 所示。

表 3.19 1 路-4 路数据分配器的真值表

A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

【说明】 由数据分配关系,用一行表示将数据分配到一个输出端的方法简化列写成引入变量真值表。特点是既压缩了真值表的规模,又直观地表示了数据分配情况。

(3) 由真值表写出输出函数 Y_0 、 Y_1 、 Y_2 、 Y_3 的最简逻辑表达式:

$$\begin{cases} Y_0 = \bar{A}_1 \bar{A}_0 D \\ Y_1 = \bar{A}_1 A_0 D \\ Y_2 = A_1 \bar{A}_0 D \\ Y_3 = A_1 A_0 D \end{cases} \quad (3.23)$$

(4) 由逻辑表达式(3.23)画出用与门、非门实现的逻辑图如图 3.41 所示。

图 3.42 为用集成 3 线-8 线译码器 74LS138 构成的 1 路-8 路数据分配器。其中, D 为数据输入端, $A_2 \sim A_0$ 为分配控制信号, $\bar{Y}_0 \sim \bar{Y}_7$ 为数据输出端。根据表 3.16 可以很容易地理解工作原理,这里不再叙述。

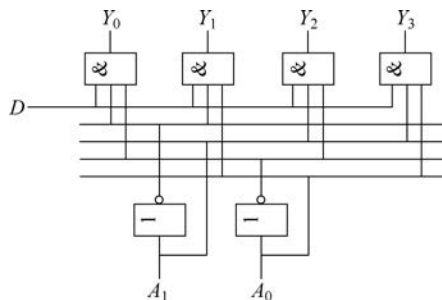


图 3.41 1 路-4 路数据分配器的逻辑图

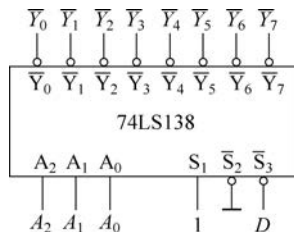


图 3.42 74LS138 构成的 1 路-8 路数据分配器

3.3.6 数据选择器

在选择控制信号(也称地址输入信号)的作用下,从多个输入数据中选择一个作为输出



视频讲解



视频讲解

信号的逻辑电路称为数据选择器(Multiplexer,MUX)。

数据选择器有 2^n 个数据输入端、 n 个选择控制信号和 1 个数据输出端。功能示意图如图 3.43 所示。

1. 4 选 1 数据选择器

4 选 1 数据选择器的逻辑设计如下。

(1) 输入变量及输出函数。4 个数据输入信号,用 D_0 、 D_1 、 D_2 、 D_3 表示; 2 个选择控制信号,用 A_1 、 A_0 表示; 1 个输出函数,用 Y 表示。

(2) 按选择控制关系,列出真值表如表 3.20 所示。

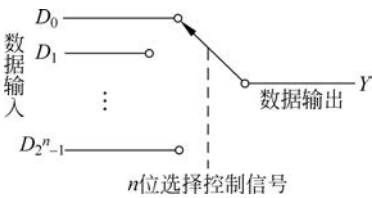


图 3.43 数据选择器示意图

表 3.20 4 选 1 数据选择器的真值表

A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

【说明】 由数据选择关系,用一行表示选择一个输入数据的方法简化列写成引入变量真值表。特点是既压缩了真值表的规模,又直观地表示了数据选择情况。

(3) 由真值表写出函数 Y 的最简逻辑表达式:

$$Y = \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3 \tag{3.24}$$

(4) 由逻辑表达式(3.24)画出用与或非门、非门实现的逻辑图如图 3.44 所示。

2. 集成数据选择器

1) 集成 4 选 1 数据选择器

74LS153 是集成双 4 选 1 数据选择器,芯片内部有 2 个相同的 4 选 1 数据选择器,各有独立的选通控制端、数据输入端、输出端和有公用的地址输入端(选择控制端)。

74LS153 的图形符号如图 3.45 所示。其中, $\bar{S}$ 为选通控制端, A_1 、 A_0 为地址输入端, D_0 、 D_1 、 D_2 、 D_3 为数据输入端, Y 为输出端。注意, $\bar{S}$ 中的横线“ $\bar{}$ ”,表示 0 有效,不是非号。

74LS153 的真值表如表 3.21 所示。

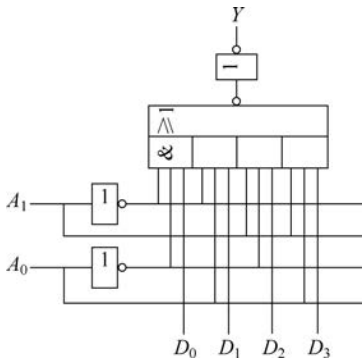


图 3.44 4 选 1 数据选择器的逻辑图

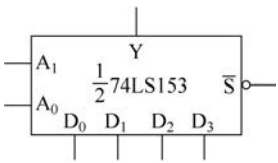


图 3.45 74LS153 的图形符号

表 3.21 74LS153 的真值表

$\bar{S}$	A_1	A_0	Y
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

由真值表得 74LS153 的逻辑表达式为

$$Y = [\bar{A}_1 \bar{A}_0 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3] \bar{S} \quad (3.25)$$

即 $\bar{S}=0$ 时

$$\begin{aligned} Y &= \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3 \\ &= \sum_{i=0}^3 m_i \cdot D_i \end{aligned} \quad (3.26)$$

其中, m_i 是以 A_1 、 A_0 为变量构成的最小项。

$\bar{S}=1$ 时

$$Y = 0 \quad (3.27)$$

2) 集成 8 选 1 数据选择器

74LS151 是集成 8 选 1 数据选择器, 图形符号如图 3.46 所示。其中, $\bar{S}$ 为选通控制端, A_2 、 A_1 、 A_0 为地址输入端, $D_0 \sim D_7$ 为数据输入端, Y 为输出端, $\bar{Y}$ 为反码输出端。注意, $\bar{S}$ 中的横线“ $-$ ”, 表示 0 有效, 不是非号。

74LS151 的真值表如表 3.22 所示。

表 3.22 74LS151 的真值表

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

由真值表得 74LS151 的逻辑表达式为

$$\begin{aligned} Y &= [\bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \bar{A}_2 A_1 \bar{A}_0 D_2 + \bar{A}_2 A_1 A_0 D_3 + \\ &\quad A_2 \bar{A}_1 \bar{A}_0 D_4 + A_2 \bar{A}_1 A_0 D_5 + A_2 A_1 \bar{A}_0 D_6 + A_2 A_1 A_0 D_7] \bar{S} \end{aligned} \quad (3.28)$$

即 $\bar{S}=0$ 时

$$\begin{aligned} Y &= \bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \bar{A}_2 A_1 \bar{A}_0 D_2 + \bar{A}_2 A_1 A_0 D_3 + \\ &\quad A_2 \bar{A}_1 \bar{A}_0 D_4 + A_2 \bar{A}_1 A_0 D_5 + A_2 A_1 \bar{A}_0 D_6 + A_2 A_1 A_0 D_7 \\ &= \sum_{i=0}^7 m_i \cdot D_i \end{aligned} \quad (3.29)$$

其中, m_i 是以 A_2 、 A_1 、 A_0 为变量构成的最小项。

$\bar{S}=1$ 时

$$Y = 0 \quad (3.30)$$

反码输出的逻辑表达式为

$$\bar{Y} = [\bar{A}_2 \bar{A}_1 \bar{A}_0 \bar{D}_0 + \bar{A}_2 \bar{A}_1 A_0 \bar{D}_1 + \bar{A}_2 A_1 \bar{A}_0 \bar{D}_2 + \bar{A}_2 A_1 A_0 \bar{D}_3 +$$

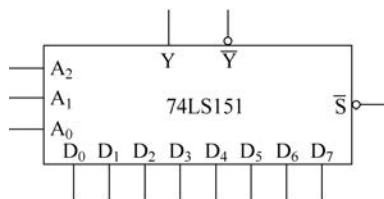


图 3.46 74LS151 的图形符号

$$A_2 \bar{A}_1 \bar{A}_0 \bar{D}_4 + A_2 \bar{A}_1 A_0 \bar{D}_5 + A_2 A_1 \bar{A}_0 \bar{D}_6 + A_2 A_1 A_0 \bar{D}_7] \bar{S} + \bar{S} \quad (3.31)$$

3) 集成数据选择器的扩展

(1) 使用选通控制端扩展。

图 3.47 为 2 片 8 选 1 数据选择器 74LS151 使用选通控制端扩展构成 16 选 1 数据选择器的逻辑电路图。

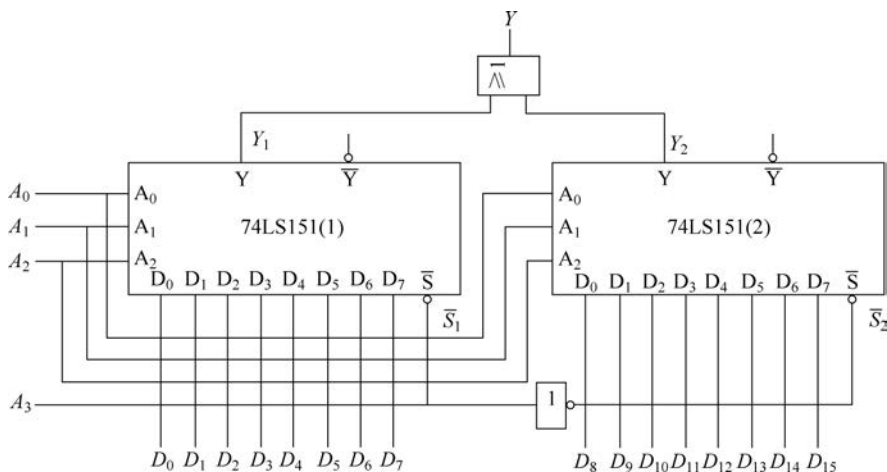


图 3.47 2 片 8 选 1 数据选择器 74LS151 构成 16 选 1 数据选择器

工作原理为：最高位地址输入信号 A₃ 控制 2 片的选通控制端。当 A₃=0 时，使 S₁=0、S₂=1，片(1)工作、片(2)禁止，Y=Y₁，由 A₂A₁A₀ 从 D₀~D₇ 中选择一个输入信号作输出；当 A₃=1 时，使 S₁=1、S₂=0，片(2)工作、片(1)禁止，Y=Y₂，由 A₂A₁A₀ 从 D₈~D₁₅ 中选择一个输入信号作输出。

(2) 使用两级选择扩展。

图 3.48 为用 4 选 1 数据选择器 74LS153 通过两级选择扩展构成 8 选 1 数据选择器的逻辑电路图。

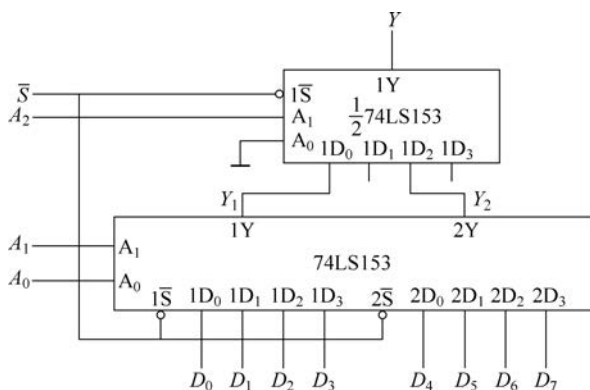


图 3.48 4 选 1 数据选择器通过两级选择扩展构成 8 选 1 数据选择器

工作原理为：当 A₂=0 时，Y=Y₁，由 A₁A₀ 从 D₀~D₃ 中选择一个输入信号作输出；当 A₂=1 时，Y=Y₂，由 A₁A₀ 从 D₄~D₇ 中选择一个输入信号作输出。

3.4 用中规模集成组合逻辑器件实现组合逻辑函数

利用某些组合 MSI 逻辑器件的特定逻辑功能,可以实现一般组合逻辑函数,从而扩展专用集成电路的应用范围。下面介绍用二进制译码器、数据选择器及加法器实现组合逻辑函数的方法。

3.4.1 用二进制译码器实现组合逻辑函数

1. 基本原理

集成二进制译码器 74LS138 在选通控制端 $S_1=1, \bar{S}_2=\bar{S}_3=0$ 时处于工作状态,输出逻辑表达式的一般形式为

$$\bar{Y}_i = \bar{m}_i \quad (i=0 \sim 7) \quad (3.32)$$

其中, m_i 是地址输入变量 A_2, A_1, A_0 构成的最小项。式(3.32)表明二进制译码器 74LS138 处于工作状态时的输出函数为地址输入变量构成最小项的反函数。

任何逻辑函数都可表示成最小项之和形式并变换成与非-与非式,即最小项反函数的与非式。由二进制译码器产生最小项的反函数,附加与非门将某些最小项的反函数进行与非运算,可实现一般组合逻辑函数。

2. 用二进制译码器实现组合逻辑函数的一般步骤

一般步骤如图 3.49 所示。

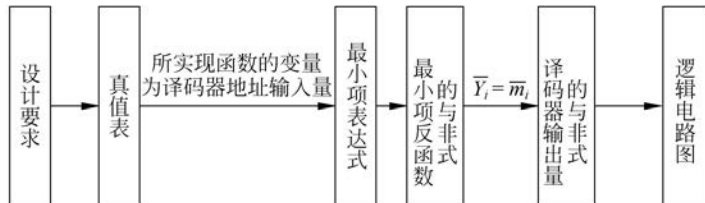


图 3.49 用二进制译码器实现组合逻辑函数的一般步骤

有以下两点说明。

- (1) 单个译码器使用时所实现函数的变量个数不多于 3 个。
- (2) 既适用于单输出也适用于多输出组合逻辑函数的实现。

3. 设计举例

【例 3.8】 试用 3 线-8 线译码器 74LS138 附加门电路设计一个 1 位全加器。

【解】 (1) 输入变量为被加数 A_i 、加数 B_i 和来自低位的进位数 C_i ；输出函数为本位和数 S_i 和向高位的进位数 C_{i+1} 。

令所实现函数的输入变量为译码器的地址输入变量,即令

$$\begin{cases} A_2 = A_i \\ A_1 = B_i \\ A_0 = C_i \end{cases} \quad (3.33)$$

- (2) 列出真值表如表 3.9 所示。

- (3) 由真值表写出标准与或式并变形为最小项反函数的与非式及译码器输出量的与



视频讲解

非式

$$\begin{cases}
 S_i = \bar{A}_i \bar{B}_i C_i + \bar{A}_i B_i \bar{C}_i + A_i \bar{B}_i \bar{C}_i + A_i B_i C_i \\
 = m_1 + m_2 + m_4 + m_7 \\
 = \overline{\bar{m}_1 \bar{m}_2 \bar{m}_4 \bar{m}_7} = \overline{\bar{Y}_1 \bar{Y}_2 \bar{Y}_4 \bar{Y}_7} \\
 C_{i+1} = \bar{A}_i B_i C_i + A_i \bar{B}_i C_i + A_i B_i \bar{C}_i + A_i B_i C_i \\
 = m_3 + m_5 + m_6 + m_7 \\
 = \overline{\bar{m}_3 \bar{m}_5 \bar{m}_6 \bar{m}_7} = \overline{\bar{Y}_3 \bar{Y}_5 \bar{Y}_6 \bar{Y}_7}
 \end{cases} \quad (3.34)$$

(4) 由逻辑表达式(3.33)和式(3.34)画出用 74LS138 及与非门实现的逻辑图如图 3.50 所示。

4. 分析举例

对用集成译码器构成的组合逻辑电路进行分析,关键是要清楚译码器本身的逻辑关系及信号的传输关系。

【例 3.9】 试分析如图 3.51 所示的组合逻辑电路的逻辑功能。

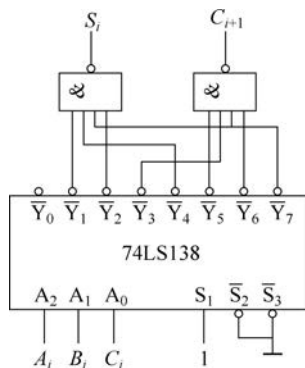


图 3.50 例 3.8 的逻辑图

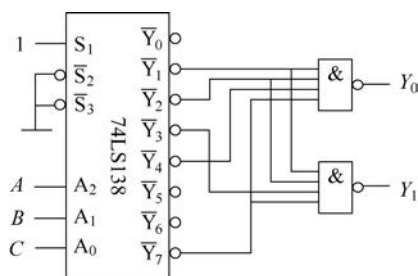


图 3.51 例 3.9 的组合逻辑电路

【解】 如图 3.51 所示的组合逻辑电路由 1 个 3 线-8 线译码器 74LS138 及 2 个与非门构成, A 、 B 、 C 为 3 个输入变量, Y_0 、 Y_1 为 2 个输出函数, 为多输出组合逻辑电路。

(1) 由逻辑图写出输出函数逻辑表达式

$$\begin{aligned}
 Y_0 &= \bar{Y}_1 \bar{Y}_2 \bar{Y}_4 = \overline{\bar{m}_1 \bar{m}_2 \bar{m}_4 \bar{m}_7} \\
 &= m_1 + m_2 + m_4 + m_7 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC
 \end{aligned} \quad (3.35)$$

$$\begin{aligned}
 Y_1 &= \bar{Y}_1 \bar{Y}_2 \bar{Y}_3 \bar{Y}_7 = \overline{\bar{m}_1 \bar{m}_2 \bar{m}_3 \bar{m}_7} \\
 &= m_1 + m_2 + m_3 + m_7 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC
 \end{aligned} \quad (3.36)$$

(2) 由逻辑表达式(3.35)和式(3.36)列出真值表如表 3.23 所示。

表 3.23 例 3.9 真值表

A	B	C	Y_0	Y_1	A	B	C	Y_0	Y_1
0	0	0	0	0	1	0	0	1	0
0	0	1	1	1	1	0	1	0	0

续表

A	B	C	Y_0	Y_1	A	B	C	Y_0	Y_1
0	1	0	1	1	1	1	0	0	0
0	1	1	0	1	1	1	1	1	1

(3) 由真值表做功能说明:

由表 3.23 可知,该电路完成 1 位全减逻辑功能。 A 、 B 、 C 分别为被减数、减数及来自低位的借位数, Y_0 为本位差数, Y_1 为向高位的借位数。

3.4.2 用数据选择器实现组合逻辑函数

1. 基本原理

集成数据选择器 74LS153、74LS151 在选通控制端 $\bar{S}=0$ 时处于工作状态,输出逻辑表达式的一般形式为

$$Y = \sum m_i \cdot D_i \quad (3.37)$$

其中, m_i 是地址输入变量 A_1 、 A_0 或 A_2 、 A_1 、 A_0 构成的最小项。

任何逻辑函数都可表示为与或表达式。若数据选择器的地址输入量为所实现函数的变量或部分变量,各数据输入端为常量 0、1 或另外的部分变量,可实现一般组合逻辑函数。

2. 用数据选择器实现组合逻辑函数的一般步骤

用数据选择器实现组合逻辑函数的方法是使所实现函数的表示形式与数据选择器的表示形式相一致,通过对比确定数据选择器各数据输入端的逻辑表达式。通常有真值表法、代数法和卡诺图法等。

真值表法是使实现函数的真值表和数据选择器的真值表有一致的形式;代数法是使实现函数的逻辑表达式和数据选择器的逻辑表达式有一致的形式;卡诺图法是使实现函数的卡诺图和数据选择器的卡诺图有一致的形式。

图 3.52 为真值表法实现组合逻辑函数的一般步骤。

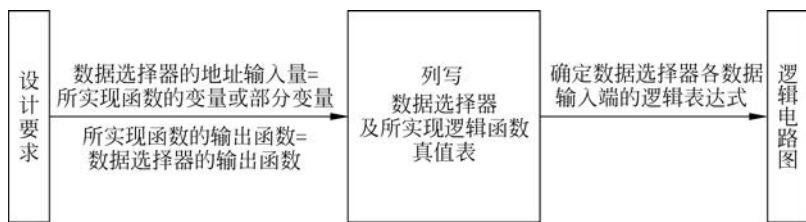


图 3.52 用数据选择器实现组合逻辑函数的一般步骤

有以下两点说明。

(1) 数据选择器一般适用于单输出组合逻辑函数的实现。

(2) 数据选择器有 n 位地址、函数有 k 个变量时,一般 n 和 k 的关系为 $n=k-1$ 或 $n=k$ 。

3. 设计举例

【例 3.10】 用 4 选 1 数据选择器 74LS153 实现逻辑函数 $F=AB+BC+AC$ 。



视频讲解

【解】 1) 真值表法

(1) 选择所实现逻辑函数的 A 、 B 变量为数据选择器的地址输入变量,即令

$$\begin{cases} A_1 = A \\ A_0 = B \end{cases} \quad (3.38)$$

从数据选择器的输出端输出所实现的逻辑函数,即令

$$F = Y \quad (3.39)$$

(2) 以 A 、 B 为变量列出 4 选 1 数据选择器及逻辑函数 F 的真值表,如表 3.24 所示。

表 3.24 例 3.10 的真值表

A	B	Y	F
0	0	D_0	0
0	1	D_1	C
1	0	D_2	C
1	1	D_3	1

(3) 由真值表确定数据选择器各数据输入端的表达式为

$$\begin{cases} D_0 = 0 \\ D_1 = D_2 = C \\ D_3 = 1 \end{cases} \quad (3.40)$$

(4) 由逻辑表达式(3.38)、式(3.39)和式(3.40)画出用 74LS153 实现的逻辑图如图 3.53 所示。

2) 代数法

(1) 选择所实现逻辑函数的 A 、 B 变量为数据选择器的地址输入变量,即令

$$\begin{cases} A_1 = A \\ A_0 = B \end{cases} \quad (3.41)$$

从数据选择器的输出端输出所实现的逻辑函数,即令

$$F = Y \quad (3.42)$$

(2) 写出 4 选 1 数据选择器 74LS153 的输出逻辑表达式

$$Y = \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3 \quad (3.43)$$

(3) 将函数 F 的表达式变形为和 4 选 1 数据选择器的输出逻辑表达式相一致的形式

$$\begin{aligned} F &= AB + BC + AC \\ &= AB + (\bar{A} + A)BC + A(\bar{B} + B)C \\ &= AB + \bar{A}BC + ABC + A\bar{B}C + ABC \\ &= AB(1 + C + C) + \bar{A}BC + A\bar{B}C \\ &= \bar{A}\bar{B} \cdot 0 + \bar{A}B \cdot C + A\bar{B} \cdot C + AB \cdot 1 \end{aligned} \quad (3.44)$$

(4) 比较式(3.43)和式(3.44),确定数据选择器各数据输入端的表达式为

$$\begin{cases} D_0 = 0 \\ D_1 = D_2 = C \\ D_3 = 1 \end{cases} \quad (3.45)$$

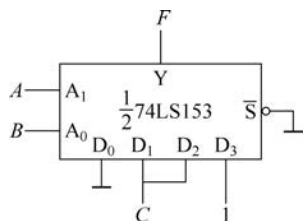


图 3.53 例 3.10 的逻辑图

(5) 由逻辑表达式(3.41)、式(3.42)和式(3.45)画出用 74LS153 实现的逻辑图如图 3.53 所示。

【例 3.11】 用 8 选 1 数据选择器 74LS151 设计一个三变量奇数判别电路,当三个输入变量的取值中有奇数个 1 时输出为 1。

【解】 (1) 三个输入变量用 A 、 B 、 C 表示,一个输出函数用 F 表示。

选择所实现逻辑函数的 A 、 B 、 C 变量为数据选择器的地址输入变量,即令

$$\begin{cases} A_2 = A \\ A_1 = B \\ A_0 = C \end{cases} \quad (3.46)$$

从数据选择器的输出端输出所实现的逻辑函数,即令

$$F = Y \quad (3.47)$$

(2) 以 A 、 B 、 C 为变量列出 8 选 1 数据选择器及逻辑函数 F 的真值表,如表 3.25 所示。

(3) 由真值表确定数据选择器各数据输入端的表达式为

$$\begin{cases} D_0 = D_3 = D_5 = D_6 = 0 \\ D_1 = D_2 = D_4 = D_7 = 1 \end{cases} \quad (3.48)$$

(4) 由逻辑表达式(3.46)、式(3.47)和式(3.48)画出用 74LS153 实现的逻辑图如图 3.54 所示。

表 3.25 例 3.11 的真值表

A	B	C	Y	F
0	0	0	D_0	0
0	0	1	D_1	1
0	1	0	D_2	1
0	1	1	D_3	0
1	0	0	D_4	1
1	0	1	D_5	0
1	1	0	D_6	0
1	1	1	D_7	1

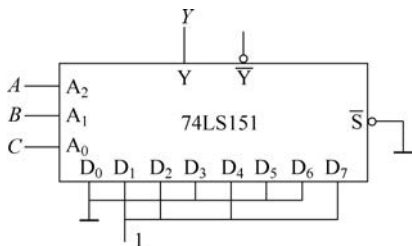


图 3.54 例 3.11 的逻辑图

4. 分析举例

对用数据选择器构成的组合逻辑电路进行分析,关键是要清楚数据选择器本身的逻辑关系及信号的传输关系。

【例 3.12】 试分析如图 3.55 所示的组合逻辑电路的逻辑功能。

【解】 如图 3.55 所示的组合逻辑电路由 3 个 4 选 1 数据选择器构成, A 、 B 、 C 、 D 为 4 个输入变量, Y 为 1 个输出函数,为单输出组合逻辑电路。

(1) 由逻辑图写出输出函数逻辑表达式为

$$\begin{aligned} Y &= \bar{A}Y_0 + AY_1 \\ &= \bar{A}(\bar{B}\bar{C} \cdot 0 + \bar{B}C \cdot D + B\bar{C} \cdot 1 + BC \cdot 1) + A(\bar{B} \cdot \bar{C} \cdot 1 + \bar{B}C \cdot 1 + B\bar{C} \cdot \bar{D} + BC \cdot 0) \\ &= \bar{A}\bar{B}CD + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}\bar{D} \end{aligned} \quad (3.49)$$

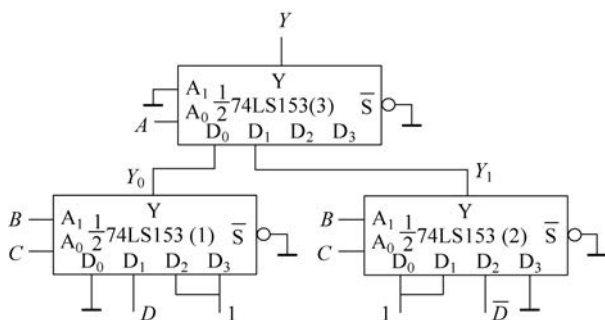


图 3.55 例 3.12 的组合逻辑电路

(2) 由逻辑表达式(3.49)列出真值表如表 3.26 所示。

表 3.26 例 3.12 的真值表

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

(3) 由真值表做功能说明:

由表 3.26 可知,当输入变量 A 、 B 、 C 、 D 的取值为 0011~1100 时输出 $Y=1$, 否则输出 $Y=0$ 。而 0011~1100 为余 3 码。

该电路为余 3 码检测电路。

3.4.3 用加法器实现组合逻辑函数

加法器适合设计函数关系为输出函数=输入变量+常量的组合逻辑电路。其中,输入变量与常数为算术加运算。常用于实现 8421 码和余 3 码的转换,转换关系式为

$$\text{余 3 码} = 8421 \text{ 码} + 0011 \quad (3.50)$$

$$8421 \text{ 码} = \text{余 3 码} + 1101 \quad (3.51)$$

【例 3.13】 试用 4 位加法器 74LS283 设计一个将 8421 码转换为余 3 码的代码转换电路。

【解】 输入变量为 4 位 8421 码,用 D 、 C 、 B 、 A 表示;输出函数为 4 位余 3 码,用 W 、 X 、 Y 、 Z 表示。

转换关系为

$$WXYZ = DCBA + 0011 \quad (3.52)$$

画出逻辑图如图 3.56 所示。

【例 3.14】 试用 4 位加法器 74LS283 设计一个将余 3 码转换为 8421 码的代码转换电路。

【解】 输入变量为 4 位余 3 码,用 $W、X、Y、Z$ 表示; 输出函数为 4 位 8421 码,用 $D、C、B、A$ 表示。

转换关系为

$$DCBA = WXYZ + 1101 \quad (3.53)$$

画出逻辑图如图 3.57 所示。

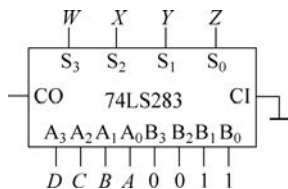


图 3.56 例 3.13 的逻辑图

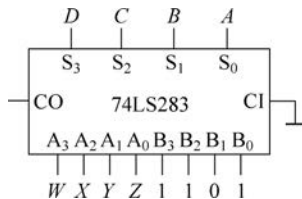


图 3.57 例 3.14 的逻辑图

3.5 组合逻辑电路的竞争冒险

3.5.1 产生竞争冒险的原因

1. 竞争冒险的概念

在组合逻辑电路中,由于逻辑门存在传输时间,使两个互反的变量经不同的路径到达同一点有先有后,这种现象称为竞争。因竞争所产生的错误输出称为冒险。

2. 1 型冒险分析

图 3.58 给出了与门产生竞争的逻辑电路及输入、输出波形。

输出逻辑表达式为

$$Y = A \cdot \bar{A} \quad (3.54)$$

由于非门存在传输时间,使到达与门输入端的信号 $\bar{A}$ 滞后于 A , $\bar{A}$ 和 A 为有竞争的变量。稳态时无论 $A=0$ 、 $A=1$ 或 A 由 1 跳变为 0 时,由式(3.54)有正确输出 $Y=0$ 。当 A 由 0 跳变为 1 时,由式(3.54)有输出 $Y=1$,为错误输出,称为 1 型冒险。

3. 0 型冒险分析

图 3.59 给出了或门产生竞争的逻辑电路及输入输出波形。

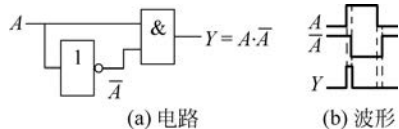


图 3.58 与门竞争冒险

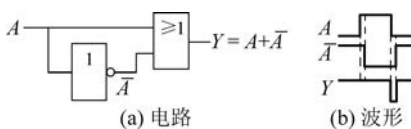


图 3.59 或门竞争冒险

输出逻辑表达式为

$$Y = A + \bar{A} \quad (3.55)$$

由于非门存在传输时间,使到达或门输入端的信号 $\bar{A}$ 滞后于 A , $\bar{A}$ 和 A 为有竞争的变量。稳态时无论 $A=0$ 、 $A=1$ 或 A 由 0 跳变为 1 时,由式(3.55)有正确输出 $Y=1$ 。当 A 由 1 跳变为 0 时,由式(3.55)有输出 $Y=0$,为错误输出,称为 0 型冒险。



视频讲解

3.5.2 检查竞争冒险的方法

1. 代数检查法

逻辑函数表达式中同时含有某一变量的原、反变量存在冒险的可能性。令其他变量为常数 0 或 1 时,若简化为 $Y=A \cdot \bar{A}$ 形式,则存在 1 型冒险;若简化为 $Y=A+\bar{A}$ 形式,则存在 0 型冒险。

【例 3.15】 一个组合逻辑电路的输出逻辑表达式为 $Y=AB+\bar{A}C$,检查竞争冒险情况。

【解】 当 $B=1、C=1$ 时, $Y=A+\bar{A}$,存在 0 型冒险。

【例 3.16】 一个组合逻辑电路的输出逻辑表达式为 $Y=(A+B)(\bar{B}+C)$,检查竞争冒险情况。

【解】 当 $A=0、C=0$ 时, $Y=B \cdot \bar{B}$,存在 1 型冒险。

2. 卡诺图检查法

逻辑函数的卡诺图中,若两个化简的包围圈相切,则存在冒险。

例 3.15 和例 3.16 中各逻辑函数的卡诺图及化简的包围圈如图 3.60 所示,各卡诺图中的两个包围圈均相切,存在冒险。

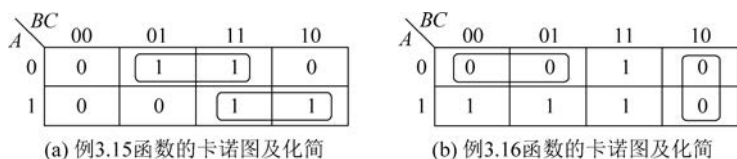


图 3.60 包围圈相切存在冒险

3.5.3 消除竞争冒险的方法

1. 修改逻辑设计,增加冗余项

例 3.15 中的逻辑表达式为 $Y=AB+\bar{A}C$,当 $B=1、C=1$ 时存在 0 型冒险。若增加冗余项 BC ,表达式变为 $Y=AB+\bar{A}C+BC$,则当 $B=1、C=1$ 时 $Y=1$,消除了冒险。

例 3.16 中的逻辑表达式为 $Y=(A+B)(\bar{B}+C)$,当 $A=0、C=0$ 时存在 1 型冒险。若增加冗余项 $(A+C)$,表达式变为 $Y=(A+B)(\bar{B}+C)(A+C)$,当 $A=0、C=0$ 时 $Y=0$,消除了冒险。

在逻辑函数卡诺图中,增加冗余项的方法就是在两个包围圈相切处再增加一个包围圈。例 3.15 和例 3.16 的卡诺图及包围圈的修改如图 3.61 所示,虚线包围圈为所增加的包围圈。

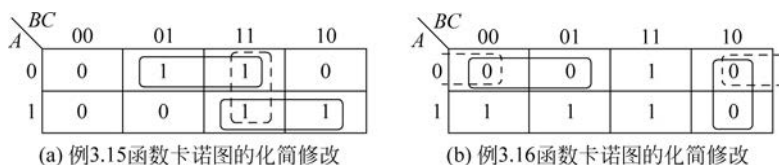


图 3.61 增加冗余项消除冒险

2. 引入选通脉冲

在组合逻辑电路输出端的输入端增加一个选通脉冲信号,当有冒险脉冲时将输出级封锁,使冒险脉冲不能输出,而当冒险脉冲消失后选通脉冲允许正常输出。

3. 接入滤波电容

冒险脉冲是一个数十纳秒的窄脉冲,在电路的输出端接入一个几百皮法的小电容,可将窄脉冲的幅度削弱至门的阈值电平以下,从而消除冒险。

3.6 组合逻辑电路 Multisim 仿真示例

1. 1 位半加器逻辑功能的 Multisim 仿真

(1) 在 Multisim 中构建 1 位半加器逻辑功能仿真电路,如图 3.62 所示。其中,1 位半加器由异或门 74LS86N 和与门 74LS08N 组成,开关用于控制输入状态,开关接 +5V 为逻辑 1 输入、接地为逻辑 0 输入,指示灯用于确定输出状态,灯亮为逻辑 1 输出、不亮为逻辑 0 输出。

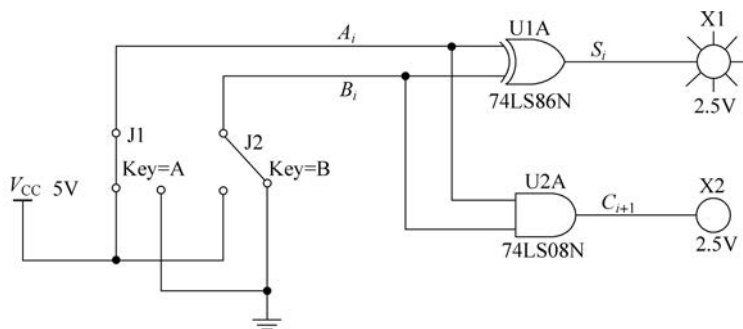


图 3.62 1 位半加器逻辑功能仿真电路

(2) 仿真运行分析。

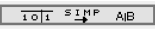
改变开关状态使 $A_i B_i$ 分别为 00、01、10、11 四种组合状态,指示灯显示的状态 $S_i C_{i+1}$ 分别为 00、10、10、01,符合半加运算的规律。

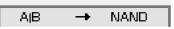
亦可用字信号发生器提供所需的输入信号;用逻辑分析仪或四踪示波器以波形形式显示输入、输出信号。

2. 用 Multisim 仿真设计组合逻辑电路

用 Multisim 仿真设计例 3.4 所要求的 1 位十进制数数值范围判别电路,十进制数用 8421 码表示,当输入的十进制数大于或等于 5 时输出为 1,否则输出为 0。

(1) 在虚拟仪表栏中选取逻辑转换仪,打开面板在真值表区单击 A、B、C、D 四个逻辑变量,在真值表区输出变量列中填入函数值,建立真值表如图 3.63 所示。

(2) 单击逻辑转换仪面板上“真值表→简化逻辑表达式”按钮 ,在逻辑表达式栏中得到相应的最简逻辑表达式,如图 3.63 面板底部表达式栏所示。

(3) 单击“逻辑表达式→与非门逻辑电路”按钮 ,在工作平台的电路工作区得到 2 输入端与非门构成的逻辑电路,如图 3.64 所示。



视频讲解



视频讲解

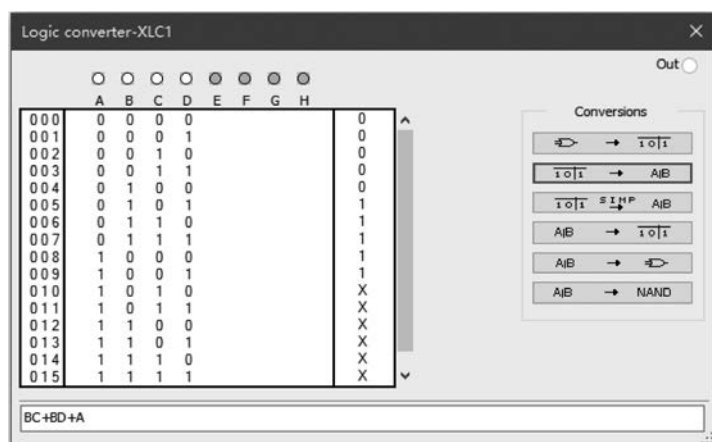


图 3.63 逻辑转换仪中建立的真值表

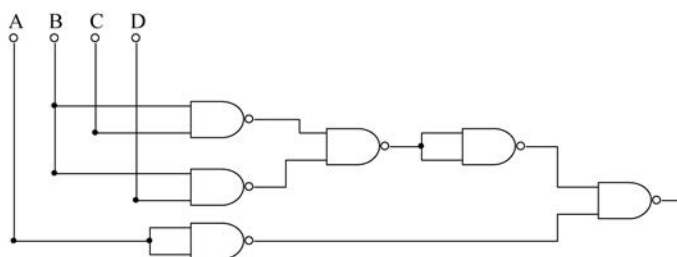


图 3.64 仿真设计的数值范围判别电路

3. 集成数据选择器逻辑功能的 Multisim 仿真

(1) 在 Multisim 中构建数据选择器逻辑功能仿真电路如图 3.65 所示。其中, 4 选 1 集成数据选择器选用 74LS153N, 字信号发生器用于产生数据选择器的 $D_0 \sim D_3$ 数据输入变量, 逻辑分析仪用于显示数据选择器的 $D_0 \sim D_3$ 数据输入变量信号及 Y 输出函数信号的波形, 数据选择器的选择控制端 A_1 、 A_0 的输入状态由开关控制, 选通控制端 $\bar{S}=0$ 处于工作状态。

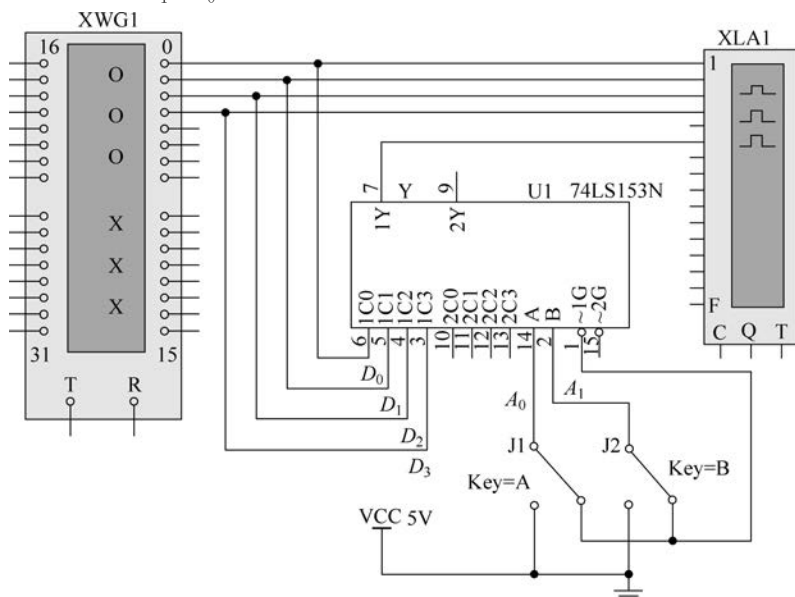


图 3.65 数据选择器逻辑功能仿真电路

(2) 设计字信号发生器输出信号时,字组内容应反映数据选择器不同数据输入端的输入情况,输入波形设计和字组数据如图 3.66 所示。

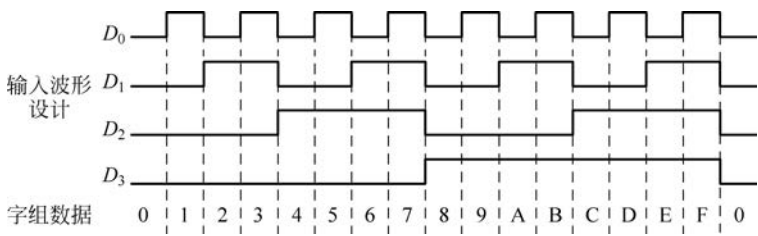


图 3.66 输入波形设计及字组数据

在字信号发生器的数据栏内以十六进制依次输入各字组数据 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F、0,并对最后一个字数据进行末地址设置,完成所有字组数据的设置。

(3) 仿真运行分析。

$A_1A_0=00$ 时逻辑分析仪显示的波形如图 3.67 所示,由上至下依次为 $D_0\sim D_3$ 数据输入变量及输出函数 Y 的波形。

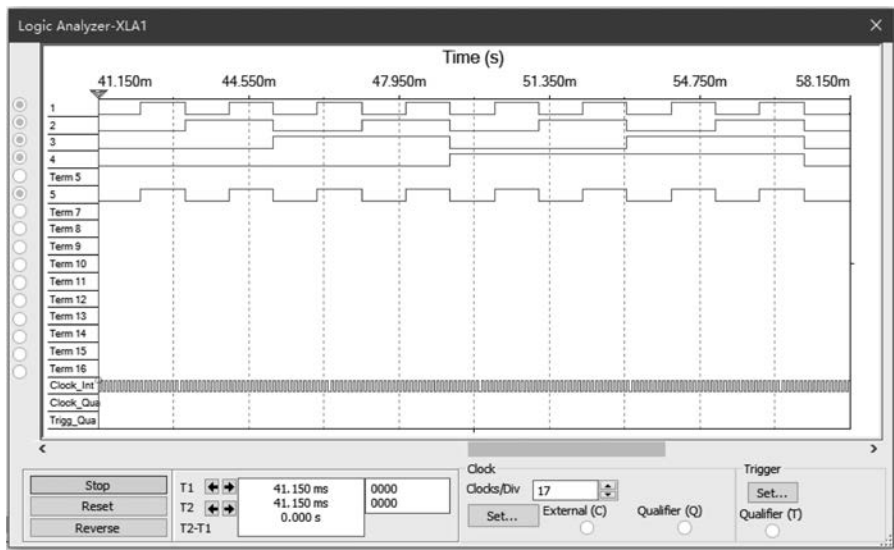


图 3.67 $A_1A_0=00$ 时的仿真实波形

由图 3.67 可知, Y 输出函数的波形和 D_0 输入变量的波形相同,实现了 $A_1A_0=00$ 时选择 D_0 作输出。

再将 A_1A_0 分别置于 01、10、11,可得到选择 D_1 、 D_2 、 D_3 作输出的仿真波形。

本章小结

本章主要介绍了组合逻辑电路的特点、组合逻辑电路的分析方法和设计方法、常用组合逻辑电路、用中规模集成组合逻辑器件实现组合逻辑函数的方法及组合逻辑电路的竞争冒险。

(1) 组合逻辑电路的结构特点是由门构成且从输出端到输入端无反馈连接。组合逻辑电路的功能特点是任一时刻的输出信号仅取决于该时刻的输入信号,而与输入信号作用前

电路所处的状态无关。

(2) 常用组合逻辑电路有加法器、数值比较器、编码器、译码器、数据分配器和数据选择器。要注意理解概念,掌握集成器件的图形符号、逻辑表达式等功能。

(3) 组合逻辑电路的分析通常采用逐级推导的方法写出输出逻辑表达式,推导表达式时关键要清楚逻辑门、集成器件的功能,然后通过列真值表等方式反映电路输出函数与输入变量之间的逻辑关系。

(4) 进行组合逻辑电路设计,所用组合器件不同,设计方法也有所不同。用门电路设计时,化简求解和门类型对应的最简逻辑表达式;用译码器、数据选择器等中规模组合器件设计时,将逻辑表达式变换为适合该器件的相应形式;加法器只适于输出等于输入算术加常数的特殊设计。

(5) 对于组合逻辑电路的竞争冒险,主要清楚什么条件下冒险现象可能出现。

自测题

1. 单项选择题

- (1) 组合逻辑电路在结构上()。
 - A. 由门构成且无反馈
 - B. 由门构成可以有反馈
 - C. 含有记忆元件
 - D. 以上均正确
- (2) 用异或门实现逻辑函数 $Y = \overline{A \oplus B \oplus C}$, 当只提供原变量时所用异或门的最少数量为()。
 - A. 1 个
 - B. 2 个
 - C. 3 个
 - D. 4 个
- (3) 某逻辑函数的最简表达式为 $Y = A\bar{B} + \bar{A}B$, 在只有原变量没有反变量的条件下实现电路时共需要的门电路为()。
 - A. 3 种类型 5 个
 - B. 3 种类型 4 个
 - C. 2 种类型 4 个
 - D. 2 种类型 3 个
- (4) 可以有多个输入信号同时有效的编码器是()。
 - A. 二进制编码器
 - B. 二-十进制编码器
 - C. 优先编码器
 - D. 8421 码编码器
- (5) 二进制编码器的功能是()。
 - A. 将若干个输入信号转换成其他的输出信号
 - B. 将某个控制信息转换成给定的二进制数
 - C. 将 $N=2^n$ 个输入信号变成 n 位二进制代码
 - D. 将 n 位二进制代码变成 $N=2^n$ 个十进制数
- (6) 用 3 线-8 线译码器芯片 74LS138 扩展实现一个 4 线-16 线译码器, 需要 74LS138 芯片的数量为()。
 - A. 2 片
 - B. 4 片
 - C. 8 片
 - D. 16 片
- (7) 在设计 8421 码译码器时, 可以作为无关项在设计中加以利用的伪码为 0000~1111 十六种状态的()。
 - A. 前 6 个
 - B. 后 6 个

C. 前 3 个和后 3 个

D. 中间 6 个

(8) 4 选 1 数据选择器的地址输入量为 A_1, A_0 , 数据输入量为 D_3, D_2, D_1, D_0 , 若使输出 $Y = D_2$, 则应使地址输入 $A_1 A_0 = ()$ 。

A. 00

B. 01

C. 10

D. 11

(9) 一组合逻辑电路的输出逻辑表达式为 $Y_1 = A \oplus B \oplus C, Y_2 = \bar{A}B + \bar{A}C + BC$, 该电路是()。

A. 1 位半加器

B. 1 位全加器

C. 1 位全减器

D. 以上均不对

(10) 当 $B = C = 1$ 时, 函数式 $Y = \bar{A}B + AC$ 存在()。

A. 0 型冒险

B. 1 型冒险

C. 0 型、1 型冒险都有

D. 无竞争冒险

2. 填空题

(1) 在数字逻辑电路中, 任一时刻的输出信号只取决于该时刻的输入信号, 而与输入信号作用前电路所处的状态无关的逻辑电路属于_____逻辑电路。

(2) 某逻辑函数的最简表达式为 $Y = \bar{A}\bar{B} + \bar{A}B$, 在只提供原变量的条件下若用与非门来实现, 则需要双输入端与非门电路的个数为_____。

(3) 二进制编码器、二十进制编码器、优先编码器中, 对输入信号没有约束的是_____。

(4) 二进制译码器的特点是, 每输入一组代码多个输出端中仅_____个输出端有信号输出。

(5) 二进制译码器的每一组输入组合对应一个输出端, 所以 n 位二进制代码输入的译码器有_____个输出端。

(6) 0 输出有效的 3 线-8 线译码器 74LS138 若使输出 $\bar{Y}_3 = 0$, 则代码输入量 $A_2 A_1 A_0$ 应为_____。

(7) 实现两个一位二进制数相加但不考虑低位来的进位, 产生一位和值及一位进位值的加法器称为_____。

(8) 1 位半加器的输入为 A_i, B_i , 其和输出逻辑表达式 $S_i =$ _____, 进位输出逻辑表达式 $C_{i+1} =$ _____。

(9) 当 1 位全加器的输入 $A_i = 1, B_i = 0, C_i = 1$ 时, 其和输出 $S_i =$ _____, 进位输出 $C_{i+1} =$ _____。

(10) 用 8 选 1 数据选择器芯片 74LS151 扩展实现一个 64 选 1 数据选择器, 所需 74LS151 芯片数为_____。

习题

1. 分析如图 3.68 所示的组合逻辑电路, 写出逻辑表达式, 列出真值表, 说明逻辑功能。
2. 分析如图 3.69 所示的组合逻辑电路, 写出逻辑表达式, 列出真值表, 说明逻辑功能。
3. 分析如图 3.70 所示的组合逻辑电路, 说明 A, B, C 有哪些取值组合使 Y 为 1。
4. 分析如图 3.71 所示的组合逻辑电路, 写出逻辑表达式, 列出真值表, 说明逻辑功能。

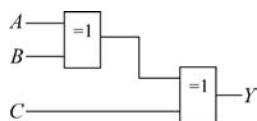


图 3.68

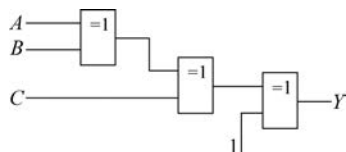


图 3.69

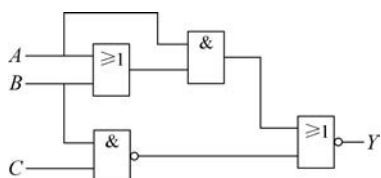


图 3.70

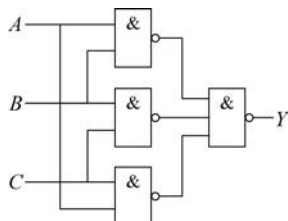


图 3.71

5. 分析如图 3.72 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。
 6. 分析如图 3.73 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。

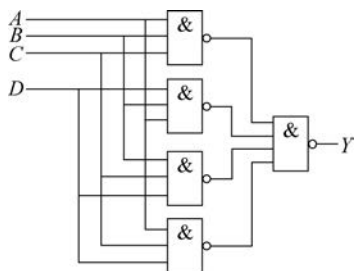


图 3.72

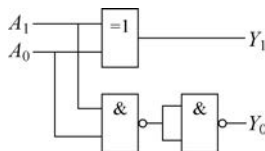


图 3.73

7. 分析如图 3.74 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。
 8. 分析如图 3.75 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。

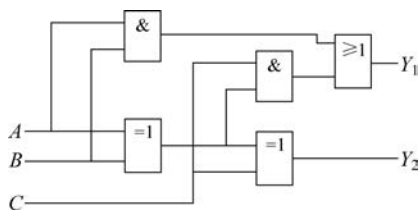


图 3.74

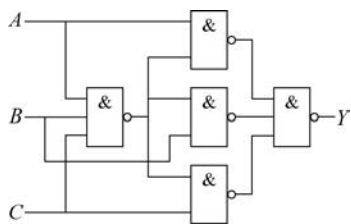


图 3.75

9. 分析如图 3.76 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能,电路 0 输出有效。
 10. 分析如图 3.77 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。
 11. 设计一个三变量多数一致电路,当输入变量 A、B、C 中 1 的个数占多数时输出 Y 为 1,输入为其他状态时输出 Y 为 0。分别用与门和或门、与非门、或非门实现。

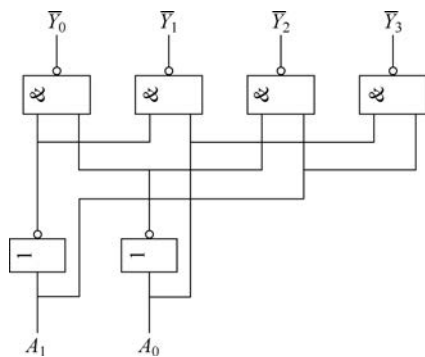


图 3.76

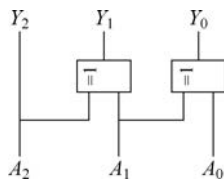


图 3.77

12. 设计一个代码转换电路,输入为 A_3 、 A_2 、 A_1 、 A_0 四位二进制代码,输出为 Y_3 、 Y_2 、 Y_1 、 Y_0 四位格雷码,用最少量的门实现。

13. 设计一个有 A 、 B 两个输入变量的数值比较器,当 $A > B$ 时输出 $Y_1 Y_0 = 10$,当 $A = B$ 时输出 $Y_1 Y_0 = 11$,当 $A < B$ 时输出 $Y_1 Y_0 = 01$ 。

14. 某组合逻辑电路输出 Y_2 、 Y_1 、 Y_0 与输入 A 、 B 、 C 逻辑关系如图 3.78 所示,试用与非门和非门设计该组合逻辑电路。

15. 试设计一个用与非门实现的“四舍五入”逻辑电路,输入 A 、 B 、 C 、 D 按 8421 码编码,当输入数值大于或等于 5 时输出 $Y = 1$,否则 $Y = 0$ 。

16. 试设计一个乘法运算电路,输入是被乘数 $A_1 A_0$ 、乘数 $B_1 B_0$ 两个 2 位二进制数,输出是两者的乘积 $Y_3 Y_2 Y_1 Y_0$ 。

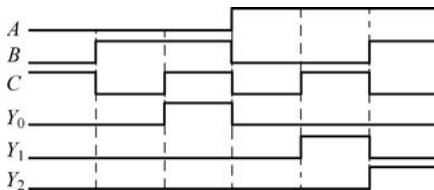


图 3.78

17. 分析如图 3.79 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。

18. 分析如图 3.80 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。

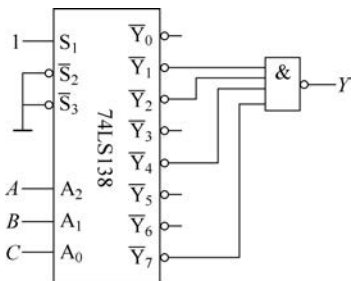


图 3.79

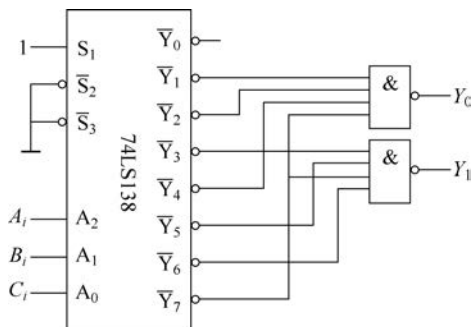


图 3.80

19. 分析如图 3.81 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑功能。

20. 分析如图 3.82 所示的组合逻辑电路,写出逻辑表达式,列出真值表,说明逻辑

功能。

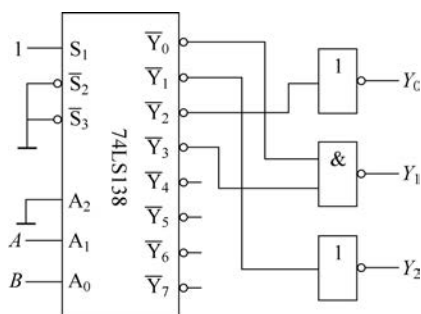


图 3.81

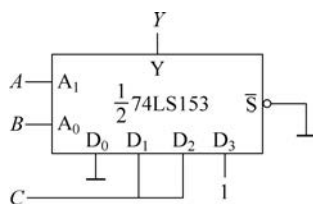


图 3.82

21. 分析如图 3.83 所示的组合逻辑电路, 写出逻辑表达式, 列出真值表, 说明逻辑功能。

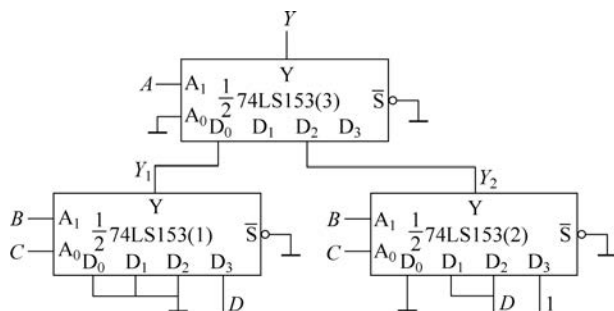


图 3.83

22. 试用 2 片 3 线-8 线译码器 74LS138 构成一个 4 线-16 线译码器。

23. 试用 3 线-8 线译码器 74LS138 和与非门实现如下多输出逻辑函数。

$$\begin{cases} Y_1 = AC \\ Y_2 = \bar{A}\bar{B}C + A\bar{B}\bar{C} + BC \\ Y_3 = \bar{B}\bar{C} + ABC \end{cases}$$

24. 试用 3 线-8 线译码器 74LS138 和必要的门电路设计一个 1 位全减器电路。

25. 试用 3 线-8 线译码器 74LS138 和必要的门电路设计一个三变量奇偶检测电路, 当输入变量中有偶数个 1 和全 0 时输出为 1, 否则输出为 0。

26. 试用 4 选 1 数据选择器 74LS153 采用两级选择方式构成 16 选 1 数据选择器。

27. 试用 4 选 1 数据选择器 74LS153 实现如下逻辑函数。

$$F(A, B, C) = \sum m(3, 5, 6, 7)$$

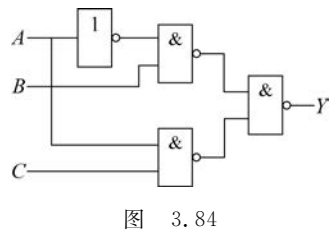
28. 试用 4 选 1 数据选择器 74LS153 设计一个三变量奇偶检测电路, 当输入变量中有偶数个 1 和全 0 时输出为 1, 否则输出为 0。

29. 试用 8 选 1 数据选择器和必要的门设计一个函数发生器电路, 其功能如表 3.27 所示。

30. 试检查如图 3.84 所示的组合逻辑电路是否存在竞争冒险。若存在,用增加冗余项的方法进行消除,并画出无竞争冒险的逻辑电路。

表 3.27 功能表

S_1	S_0	F
0	0	$A \cdot B$
0	1	$A + B$
1	0	$A \oplus B$
1	1	$\overline{A}$



Multisim 仿真练习题

- 1. 用 74LS00 四 2 输入与非门构建仿真电路,对例 3.1 中的逻辑电路进行 Multisim 仿真分析。
- 2. 用 74LS86 四 2 输入异或门构建仿真电路,对例 3.3 中的逻辑电路进行 Multisim 仿真分析。
- 3. 在 Multisim 中用 74LS138 构建 3 线-8 线译码器逻辑功能验证的仿真电路。
- 4. 用 3 线-8 线译码器 74LS138、74LS40 双 4 输入与非门构建仿真电路,对例 3.8 中的 1 位全加器进行 Multisim 仿真分析。
- 5. 用 4 位加法器 74LS283 构建仿真电路,对例 3.13、例 3.14 的 8421 码、余 3 码转换电路进行 Multisim 仿真分析。
- 6. 用 74LS00 四 2 输入与非门、74LS04 六反相器构建仿真电路,对习题 9 的电路进行 Multisim 仿真分析。
- 7. 用 3 线-8 线译码器 74LS138、74LS00 四 2 输入与非门及 74LS04 六反相器构建仿真电路,对习题 19 的电路进行 Multisim 仿真分析。
- 8. 用 74LS153 双 4 选 1 数据选择器构建仿真电路,对习题 20 的电路进行 Multisim 仿真分析。
- 9. 在 Multisim 中用与非门完成习题 11 的组合逻辑电路仿真设计。
- 10. 在 Multisim 中完成 1 位全减器电路的仿真设计。