

数字电路可以分为组合逻辑电路和时序逻辑电路。组合逻辑电路是指电路任意时刻的输出状态只与该时刻的输入状态有关,而与该时刻之前的状态无任何关系,即组合逻辑电路不具有记忆功能,其输出与输入的关系具有即时性。而时序逻辑电路的输出不仅与当前时刻的输入状态有关,而且还与电路之前的状态有关,所以,时序逻辑电路具有记忆功能,后面章节将对其进行讨论。

3.1 逻辑门电路符号和外部特性

在数字电路中,逻辑运算的应用十分广泛,不同的逻辑运算在电路中都有自己不同的门电路符号,它们是组成各种数字电路的基本单元。本节将介绍基本逻辑门电路符号、复合逻辑门电路符号以及逻辑门电路的外部特性。

3.1.1 基本逻辑门电路符号

基本逻辑门电路指实现简单逻辑关系的电路,如与门、或门及非门。在逻辑电路中,输入和输出一般用高、低电平表示两种不同的状态。逻辑电路电平的高、低可用逻辑“0”和“1”表示,若用逻辑“1”表示高电平,逻辑“0”表示低电平,则称该体制为正逻辑体制;反之,若用逻辑“0”表示高电平,逻辑“1”表示低电平,则称为负逻辑体制。对于同一个电路,既可用正逻辑表示,也可用负逻辑表示。如“与”逻辑的功能表见表 3-1(a),H 表示高电平,L 表示低电平,分别进行正、负逻辑赋值,得到表 3-1(b)和表 3-1(c),观察发现,正逻辑下的与门,在负逻辑下却实现或逻辑运算。

表 3-1 “与”逻辑

(a) “与”逻辑的功能表					
A	B	Y	A	B	Y
L	L	L	H	L	L
L	H	L	H	H	H
(b) 正逻辑下“与”逻辑的真值表					
A	B	Y	A	B	Y
0	0	0	1	0	0
0	1	0	1	1	1

续表

(c) 负逻辑下“与”逻辑的真值表					
A	B	Y	A	B	Y
1	1	1	0	1	1
1	0	1	0	0	0

同理可知,正逻辑的或门在负逻辑中实现与运算;正逻辑的非门在负逻辑中仍然实现非运算。以后章节若无特别说明,均采用正逻辑体制。

1. 与逻辑门电路符号

与门。将实现“与”逻辑关系的电路称为与门电路。二输入端的“与”门对应的国际符号如图 3-1 所示。

2. 或逻辑门电路符号

或门。将实现“或”逻辑关系的电路称为或门电路。二输入端的“或”门对应的国际符号如图 3-2 所示。

3. 非逻辑门电路符号

非门。将实现“非”逻辑关系的电路称为非门电路,也称为反相器。“非”门对应的国际符号如图 3-3 所示。



图 3-1 二输入端的“与”门对应的国际符号



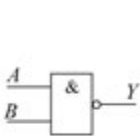
图 3-2 二输入端的“或”门对应的国际符号



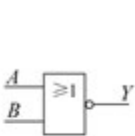
图 3-3 “非”门对应的国际符号

3.1.2 复合逻辑门电路符号

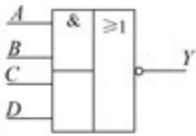
由 3 种基本逻辑运算可以组合成多种复合逻辑运算。“与非”“或非”“与或非”“异或”“同或”这几种复合逻辑门电路的国际符号如图 3-4 所示。



(a) “与非”门电路的国际符号



(b) “或非”门电路的国际符号



(c) “与或非”门电路的国际符号



(d) “异或”门电路的国际符号



(e) “同或”门电路的国际符号

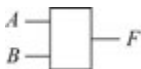
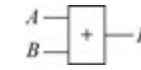
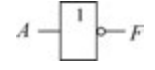
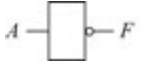
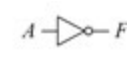
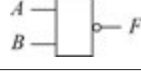
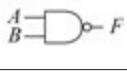
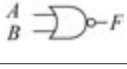
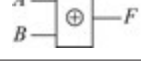
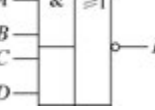
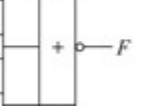
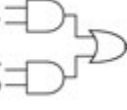
图 3-4 复合逻辑门电路的国际符号

注意: 本书给出的门电路符号均为中国国标符号,除此符号外,每种逻辑门电路还有自己的惯用符号和美国国标符号。

逻辑门电路是实现基本逻辑运算和常用逻辑运算的数字电路。表 3-2 给出了 3 种基本逻辑门和常用逻辑门电路的两种国标符号和惯用符号,更多的逻辑门电路符号请参考附

录 A。

表 3-2 逻辑门电路符号

名称	中国国标符号	惯用符号和美国国标符号	逻辑运算
与门		 	$F = AB$
或门		 	$F = A + B$
非门		 	$F = \bar{A}$
与非门		 	$F = \overline{AB}$
或非门		 	$F = \overline{A + B}$
异或门		 	$F = \bar{A}B + A\bar{B} = A \oplus B$
同或门		 	$F = \bar{A}\bar{B} + AB = A \odot B$
与或非门		 	$F = \overline{AB + CD}$

需要特别说明的是,逻辑门电路符号不能主观臆造。表 3-2 给出的符号都有具体的逻辑电路器件,在市场上是可以买到的。

3.1.3 逻辑门电路的外部特性

以 TTL 与非门为例,逻辑门电路的外部特性主要表现在以下几个主要参数。

(1) 输出高电平 U_{OH} : TTL 与非门的一个或几个输入为低电平时的输出电平。产品规范值 $U_{OH} \geq 2.4V$,标准高电平 $U_{SH} = 2.4V$ 。

(2) 高电平输出电流 I_{OH} : 输出为高电平时,提供给外接负载的最大输出电流,超过此值会使输出高电平下降。 I_{OH} 表示电路的拉电流负载能力。

(3) 输出低电平 U_{OL} : TTL 与非门的输入全为高电平时的输出电平。产品规范值 $U_{OL} \leq 0.4V$,标准低电平 $U_{SL} = 0.4V$ 。

(4) 低电平输出电流 I_{OL} : 输出为低电平时,外接负载的最大输出电流,超过此值会使输出低电平上升。 I_{OL} 表示电路的灌电流负载能力。

(5) 扇入系数 N_I : 一个门电路允许的输入端的最大数目。一般为 2~5,最多不超过 8。

(6) 扇出系数 N_O : 一个门电路的输出能带同类门的最大数目,它反映了门电路的带负载能力。一般 TTL 门电路的 $N_O \geq 8$,功率驱动门的 N_O 可达 25。

(7) 最大工作频率 f_{max} : 若超过此频率,电路就不能正常工作。

(8) 输入开门电平 U_{ON} : 在额定负载下使与非门的输出电平达到标准低电平 U_{SL} 的输入电平。它表示使与非门开通的最小输入电平。一般 TTL 门电路的 $U_{ON} \approx 1.8V$ 。

(9) 输入关门电平 U_{OFF} : 使与非门的输出电平达到标准高电平 U_{SH} 的输入电平。它表示使与非门关断所需的最大输入电平。一般 TTL 门电路的 $U_{OFF} \approx 0.8V$ 。

(10) 高电平输入电流 I_{IH} : 输入为高电平时的输入电流, 即当前级输出为高电平时, 本级输入电路造成的前级拉电流。

(11) 低电平输入电流 I_{IL} : 输入为低电平时的输出电流, 即当前级输出为低电平时, 本级输入电路造成的前级灌电流。

(12) 平均传输时间 t_{pd} : 信号通过与非门时所需的平均延迟时间。在工作频率较高的数字电路中, 信号经过多级传输后造成的时间延迟会影响电路的逻辑功能。

(13) 空载功耗: 与非门空载时电源总电流 I_{CC} 与电源电压 V_{CC} 的乘积。

3.2 组合逻辑电路的分析

组合逻辑电路可以有一个或多个输入端, 也可以有一个或多个输出端。图 3-5 给出了组合逻辑电路的示意框图。

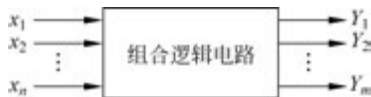


图 3-5 组合逻辑电路的示意框图

由此可看出, 数字信号在电路中是从输入端到输出端的单向传递, 函数表达式的形式如下。

$$\begin{cases} Y_1 = f_1(x_1, x_2, \dots, x_n) \\ Y_2 = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ Y_m = f_m(x_1, x_2, \dots, x_n) \end{cases}$$

对组合逻辑电路的研究, 主要从分析、设计及运用 3 方面进行。

- (1) 给出具体的组合逻辑电路, 对其进行分析并确定其逻辑功能。
- (2) 依据给出的具体需求, 设计出相关的组合逻辑电路, 完成所需功能。
- (3) 掌握常用中小规模器件的逻辑功能, 并能灵活运用于工程实践中。

组合逻辑电路的分析, 是指已知某一组合逻辑电路图, 通过列出其对应的逻辑表达式、真值表, 分析出该电路完成的功能。

3.2.1 组合逻辑电路的分析步骤

给定一个组合逻辑电路图时, 具体的分析步骤如下。

- (1) 根据逻辑电路图, 从输入端到输出端逐级写出逻辑表达式。
- (2) 利用公式法或卡诺图法, 对所得逻辑表达式进行化简, 得到最简逻辑表达式。
- (3) 根据得到的最简表达式列出真值表。
- (4) 依据真值表或最简表达式, 对逻辑电路进行分析, 确定其逻辑功能, 并进行文字描

述或进行电路改进。

3.2.2 组合逻辑电路的分析实例

下面通过几个实例,给出对组合逻辑电路具体的分析过程。

【例 3-1】 试分析图 3-6 所示的组合逻辑电路图,给出逻辑功能。

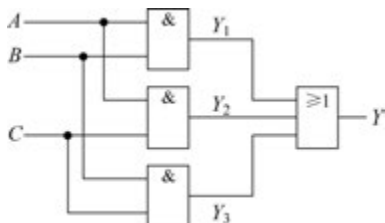


图 3-6 例 3-1 逻辑电路图

解: 由逻辑电路图逐级写出逻辑表达式。

$$\begin{cases} Y_1 = A \cdot B \\ Y_2 = A \cdot C \\ Y_3 = B \cdot C \\ Y = Y_1 + Y_2 + Y_3 \end{cases}$$

将 Y_1 、 Y_2 、 Y_3 的表达式代入 Y 的表达式中,可得 $Y = AB + AC + BC$ 。

因为以上表达式已经是最简表达式,所以可直接列出对应的真值表,见表 3-3。

表 3-3 例 3-1 真值表

输入			输出	输入			输出
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

由表 3-3 可看出,只有当 3 个输入变量中的两个或 3 个取值为 1 时,输出值才为 1,其他情况下输出均为 0。所以,该电路是一个多数表决电路,用来判断输入变量中是否有多数变量为 1。

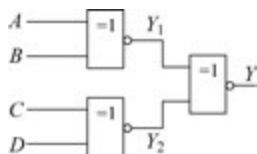


图 3-7 例 3-2 逻辑电路图

【例 3-2】 试分析图 3-7 所示的组合逻辑电路图,并给出逻辑功能。

解: 由电路图逐级写出逻辑表达式,并将 Y_1 和 Y_2 的表达式代入 Y 的表达式中,可得

$$\left. \begin{aligned} Y_1 &= \overline{A \oplus B} = AB + \overline{A} \overline{B} \\ Y_2 &= \overline{C \oplus D} = CD + \overline{C} \overline{D} \\ Y &= \overline{Y_1 \oplus Y_2} = Y_1 Y_2 + \overline{Y_1} \overline{Y_2} \end{aligned} \right\} \Rightarrow$$

$$\begin{aligned}
 Y &= Y_1 Y_2 + \overline{Y_1} \overline{Y_2} \\
 &= (AB + \overline{A} \overline{B})(CD + \overline{C} \overline{D}) + (\overline{A} B + A \overline{B})(\overline{C} D + C \overline{D}) \\
 &= ABCD + ABC\overline{D} + \overline{A} \overline{B} CD + \overline{A} \overline{B} C\overline{D} + \\
 &\quad A \overline{B} C \overline{D} + A \overline{B} C D + \overline{A} B C \overline{D} + \overline{A} B C D
 \end{aligned}$$

列出相应真值表,见表 3-4。

表 3-4 例 3-2 真值表

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

仔细观察表 3-4,分析可发现,当输入变量中有奇数个“1”时,电路的输出为“0”;反之,当输入有偶数个“1”时,电路的输出为“1”,所以该电路是一个奇偶校验器。

以上两个例子中的逻辑电路都只有一个输出变量,所以为单输出组合逻辑电路,反之,若组合逻辑电路有多个输出量,则称为多输出组合逻辑电路。下面给出一个多输出组合逻辑电路的例子。

【例 3-3】 试分析图 3-8 所示的两输出电路的逻辑功能。

解: 由逻辑电路图分别写出两个输出的逻辑表达式,并进行化简。

$$C = \overline{Y_1} = \overline{AB} = AB$$

$$S = \overline{Y_2} \cdot Y_3 = \overline{Y_1 A} \cdot Y_1 B = \overline{\overline{AB} \cdot A} \cdot \overline{AB} \cdot B = \overline{AB} \cdot \overline{AB} = \overline{AB} + \overline{AB} = A \oplus B$$

列出真值表,见表 3-5。

表 3-5 例 3-3 真值表

输 入		输 出	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

分析以上真值表可以看出,若 A、B 为两个二进制的加数,则 S 为这两个数的和,C 为两数相加向高位的进位,所以该电路可作为运算器中的基本单元电路——半加器。后面章

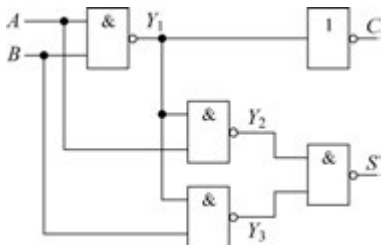


图 3-8 例 3-3 逻辑电路图

节将对加法器进行详细介绍。

3.3 组合逻辑电路的设计

3.3.1 组合逻辑电路的设计步骤

组合逻辑电路的设计过程与分析过程互逆,是根据所给逻辑问题设计出相关的逻辑电路,以满足逻辑功能要求的过程。一般来说,组合逻辑电路的设计按照以下步骤进行。

- (1) 分析逻辑问题,抽象出逻辑输入变量和输出变量。
- (2) 依据逻辑要求列出真值表。
- (3) 根据真值表列出逻辑函数表达式。
- (4) 对表达式进行化简或变形,使之与给定器件匹配。
- (5) 根据逻辑表达式做出逻辑电路图。

3.3.2 组合逻辑电路的设计实例

下面通过两个实例分别给出单输出组合逻辑电路和多输出组合逻辑电路的设计过程。

【例 3-4】 试设计一个裁判表决器。假设在某舞蹈考试的考场有 3 位考官,其中一位是主考官,两位是副考官。当考生完成表演后,考官按下自己面前的按钮决定该考生是否通过考试。若有两位或两位以上考官通过(其中一位必须是主考官),则表明考生通过考试。试用与非门实现电路,完成该表决器。

解:

- (1) 抽象输入变量与输出变量。

根据逻辑问题,有 3 位考官进行评分,所以设定 3 个输入变量 A 、 B 、 C 分别代表这 3 位考官,其中 A 表示主考官, B 和 C 表示副考官,取值为“1”时表示考官认为合格,取值为“0”时表示考官认为不合格;每位考生的考试结果只有两种,所以用 1 个输出变量 Y 表示, Y 取值为“1”时表示考试通过,取值为“0”时表示考试不通过。

- (2) 由给出的条件列出真值表,见表 3-6。

表 3-6 例 3-4 真值表

输 入			输出	输 入			输出
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	0	1	1	1	1

- (3) 由以上真值表写出逻辑函数表达式,并转换为最简与非表达式。

$$\begin{aligned}
 Y &= \overline{A}\overline{B}C + A\overline{B}C + ABC = (\overline{A}\overline{B}C + ABC) + (A\overline{B}C + ABC) \\
 &= AC + AB = \overline{\overline{AC}} + \overline{\overline{AB}} = \overline{\overline{AC} \cdot \overline{AB}}
 \end{aligned}$$

- (4) 画出逻辑电路图,如图 3-9 所示。

【例 3-5】 现有两个水泵 L 和 S 往某个深井内加水,其中水泵 L 的功率大于水泵 S 的功率,示意图如图 3-10 所示。当水位低于 A 点时,需要两个水泵一起工作;当水位位于 A 点和 B 点之间时,只要水泵 L 工作即可;当水位位于 B 点和 C 点之间时,只需要水泵 S 工作;若水位位于 C 点或 C 点以上时,水泵不需要工作。试用与非门设计一个控制电路控制这两个水泵的工作(输入仅提供原变量)。



图 3-9 例 3-4 逻辑电路图

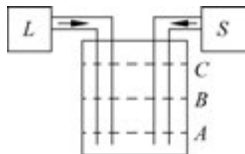


图 3-10 例 3-5 示意图

解: 设输入逻辑变量为 A 、 B 、 C , 输出变量为 L 、 S , 当水高于某一水位时, 取值为“1”, 否则为“0”; 水泵工作时, 取值为“1”, 否则为“0”。两个水泵工作时的逻辑取值如下。

当水位低于 A 点时, 有 $A=0, B=0, C=0$ 时, $L=1, S=1$ 。

当水位位于 A 点和 B 点之间时, 有 $A=1, B=0, C=0$ 时, $L=1, S=0$ 。

当水位位于 B 点和 C 点之间时, 有 $A=1, B=1, C=0$ 时, $L=0, S=1$ 。

若水位位于 C 点或 C 以上时, 有 $A=1, B=1, C=1$ 时, $L=0, S=0$ 。

除了以上情况外, 其他情况均不会出现, 可视为随意项。列出真值表, 见表 3-7。

表 3-7 例 3-5 真值表

输入			输出		输入			输出	
A	B	C	L	S	A	B	C	L	S
0	0	0	1	1	1	0	0	1	0
0	0	1	×	×	1	0	1	×	×
0	1	0	×	×	1	1	0	0	1
0	1	1	×	×	1	1	1	0	0

利用卡诺图(见图 3-11)进行化简, 可得

$$\begin{cases} L = \bar{B} \\ S = \bar{A} + BC \end{cases}$$

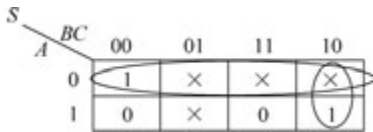
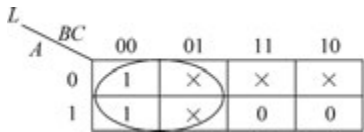


图 3-11 例 3-5 卡诺图

画出逻辑电路图, 如图 3-12 所示。

若要求用与非门实现, 则逻辑表达式变为

$$\begin{cases} L = \bar{B} = \overline{B \cdot 1} \\ S = \bar{A} + BC = \overline{\overline{\bar{A}} \cdot \overline{BC}} = \overline{\overline{A} \cdot \overline{BC}} = \overline{\overline{A} \cdot \overline{BC} \cdot 1} \end{cases}$$

根据逻辑函数表达式,画出相应逻辑电路图,如图 3-13 所示。

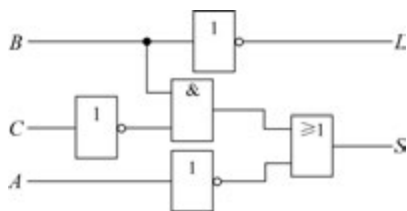


图 3-12 例 3-5 逻辑电路图



图 3-13 例 3-5 用与非门构造的逻辑电路图

3.4 中规模通用集成电路的逻辑设计

为了使用方便,可以将常用组合逻辑电路的设计标准化,制造成各类中小规模的集成电路芯片,集成电路芯片具有通用性强、扩展性好、兼容性好、功耗小、可靠性强、输入负载小等优点。采用中小规模集成电路构造数字系统,可以使数字系统的装配密度增大、结构简化、体积缩小、重量减轻、功耗降低、可靠性提高、设计实现和维护较容易,而且使用方便。本节将介绍几种常用的中规模通用集成电路。

3.4.1 加法器

所谓加法器,是指能够实现加法运算的电路。在计算机及其他数字系统中采用的是二进制的表示方法,不管是加、减运算,还是乘、除运算,均要转换为二进制的加法运算。所以,加法器是算术逻辑运算单元的基本逻辑电路。

1. 半加器

两个 1 位二进制数相加时,只考虑本位的相加,不考虑低位来的进位,这种相加被称为半加。能够实现半加功能的逻辑电路称为半加器(HA)。3.2 节中的例 3-3 给的是一个实现半加器的逻辑电路图,对图 3-8 进行改进,可得到用异或门以及与门构造的半加器。逻辑函数表达式如下:

$$\begin{cases} C = AB \\ S = A \oplus B \end{cases}$$

图 3-14 给出了改进后半加器的逻辑电路图和逻辑符号。

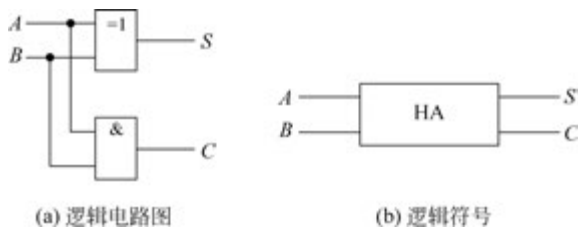


图 3-14 半加器的逻辑电路图与逻辑符号

2. 全加器

两个1位的二进制数相加时,除了考虑本位的相加外,还要考虑低位来的进位,这种相加被称为全加。能够实现全加功能的逻辑电路称为全加器(FA)。全加器的逻辑符号如图3-15所示,真值表见表3-8,其中, C_{n-1} 代表低位来的进位, A_n 和 B_n 代表本位的两个加数, S_n 代表本位和, C_n 代表向高位的进位。



图 3-15 全加器的逻辑符号

表 3-8 全加器真值表

输 入			输 出		输 入			输 出	
A_n	B_n	C_{n-1}	C_n	S_n	A_n	B_n	C_{n-1}	C_n	S_n
0	0	0	0	0	1	0	0	0	1
0	0	1	0	1	1	0	1	1	0
0	1	0	0	1	1	1	0	1	0
0	1	1	1	0	1	1	1	1	1

由真值表写出全加器本位和与进位信号的表达式,并进行变形可得

$$\begin{aligned}
 C_n &= \overline{A_n}B_nC_{n-1} + A_n\overline{B_n}C_{n-1} + A_nB_n\overline{C_{n-1}} + A_nB_nC_{n-1} \\
 &= (\overline{A_n}B_nC_{n-1} + A_n\overline{B_n}C_{n-1}) + (A_nB_n\overline{C_{n-1}} + A_nB_nC_{n-1}) \\
 &= (\overline{A_n}B_n + A_n\overline{B_n})C_{n-1} + A_nB_n(\overline{C_{n-1}} + C_{n-1}) \\
 &= (A_n \oplus B_n)C_{n-1} + A_nB_n \\
 S_n &= \overline{A_n}\overline{B_n}C_{n-1} + \overline{A_n}B_n\overline{C_{n-1}} + A_n\overline{B_n}\overline{C_{n-1}} + A_nB_nC_{n-1} \\
 &= (\overline{A_n}\overline{B_n}C_{n-1} + A_nB_nC_{n-1}) + (\overline{A_n}B_n\overline{C_{n-1}} + A_n\overline{B_n}\overline{C_{n-1}}) \\
 &= (\overline{A_n}\overline{B_n} + A_nB_n)C_{n-1} + (\overline{A_n}B_n + A_n\overline{B_n})\overline{C_{n-1}} \\
 &= (\overline{A_n} \oplus \overline{B_n})C_{n-1} + (A_n \oplus B_n)\overline{C_{n-1}} \\
 &= A_n \oplus B_n \oplus C_{n-1}
 \end{aligned}$$

根据逻辑表达式画出逻辑电路图,如图3-16所示。

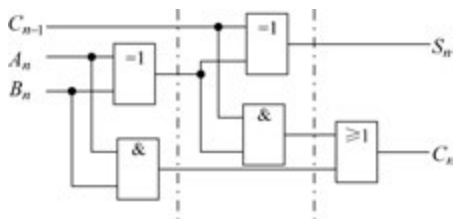


图 3-16 全加器逻辑电路图

仔细观察图3-16可发现,全加器可由两个半加器和一个或门构成,如图3-17所示。



图 3-17 半加器构成全加器

3. n 位加法器

若要实现 n 位二进制数的加法运算,可以采用串行结构或并行结构。下面以两个 4 位二进制数 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 相加为例,给出这两种不同的结构。

1) 串行进位加法器

4 位串行进位加法器原理图如图 3-18 所示,由 4 个全加器级联构成,低位全加器的进位输出与相邻的高位全加器的进位输入相连,各全加器的进位按照由低位向高位逐级串行传递,并形成进位链。

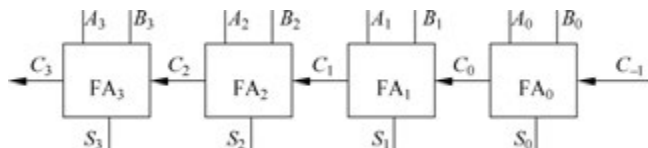


图 3-18 4 位串行进位加法器原理图

串行进位加法器具有电路简单的特点。又因为每一位相加的和都与本位进位输入有关,最高位只有在其他各低位全部相加并产生进位信号之后,才能产生最后的运算结果,所以运算速度较慢,而且位数越多,运算速度越低。

2) 超前进位加法器

超前进位加法器不必逐级传递进位信号,解决了串行进位加法器速度慢的问题。因为超前进位加法器可以根据输入信号同时形成各位向高位的进位,所以又被称为先行进位加法器、并行进位加法器。

4 位二进制数 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 相加,令 $P_i = A_i \oplus B_i$, $G_i = A_i B_i$,则第 i 位的进位信号为 $C_i = (A_i \oplus B_i)C_{i-1} + A_i B_i = P_i C_{i-1} + G_i$,各位相加产生的进位表达式如下。

$$C_0 = P_0 C_{-1} + G_0$$

$$C_1 = P_1 C_0 + G_1 = P_1 (P_0 C_{-1} + G_0) + G_1 = P_1 P_0 C_{-1} + P_1 G_0 + G_1$$

$$\begin{aligned} C_2 &= P_2 C_1 + G_2 = P_2 (P_1 P_0 C_{-1} + P_1 G_0 + G_1) + G_2 \\ &= P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2 \end{aligned}$$

$$\begin{aligned} C_3 &= P_3 C_2 + G_3 = P_3 (P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2) + G_3 \\ &= P_3 P_2 P_1 P_0 C_{-1} + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3 \end{aligned}$$

式中, P_i 为进位传递函数, G_i 为进位产生函数。由以上式子可以看出,各全加器的进位信号只与最低位的进位信号有关,所以,在输入两个加数及 C_{-1} 之后,可同时并行产生 $C_0 \sim C_3$,而不必像串行进位加法器需逐级传递进位信号。由基本门电路构成的超前进位加法器如图 3-19 所示。

中规模集成电路 74LS283 是 4 位超前进位全加器,其芯片引脚图如图 3-20 所示。图中, V_{CC} 接电源, GND 接地, $A_3 \sim A_0$ 与 $B_3 \sim B_0$ 分别输入参与运算的 4 位二进制数, C_3 与 $S_3 \sim S_0$ 输出最后的运算结果。若要完成更多位数的运算,可以将芯片级联扩展。

【例 3-6】 试用 74LS283 芯片实现 2 个 8 位数的相加运算。

解: 由于每片 74LS283 可实现 4 位二进制的运算,因此 8 位数的运算可用 2 片 74LS283 实现。将两个 8 位数的低 4 位送入低位芯片相加,高 4 位送入高位芯片相加,并将低位芯片的 C_3 接地,进位信号 C_3 接入高位芯片的 C_{-1} 。进行运算后,由高位芯片的 C_3 和两片芯片的 S 端输出结果。构造的电路图如图 3-21 所示。

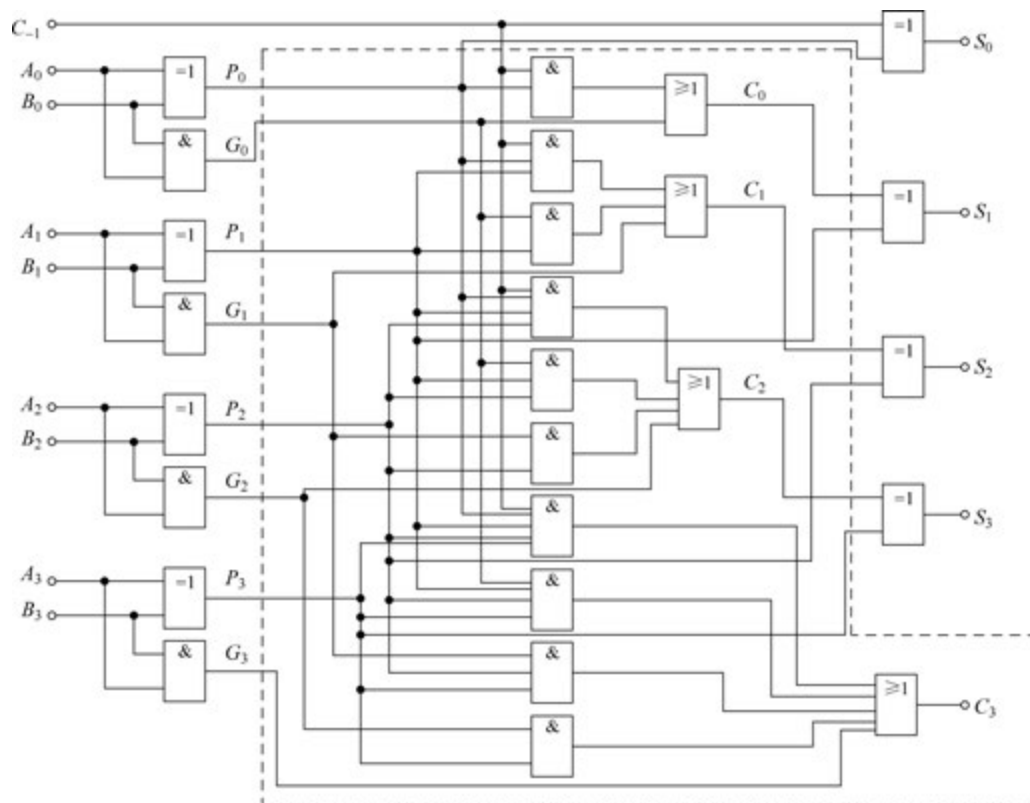


图 3-19 由基本门电路构成的超前进位加法器

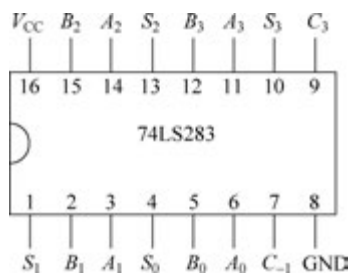


图 3-20 74LS283 芯片引脚图

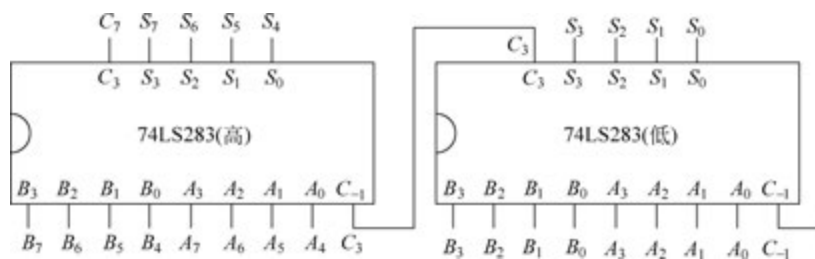


图 3-21 例 3-6 电路图

3.4.2 数值比较器

数值比较器是指用来比较两个二进制数大小的逻辑电路,其输入变量是两个 n 位二进制数 $A_{n-1}A_{n-2}\cdots A_1A_0$ 和 $B_{n-1}B_{n-2}\cdots B_1B_0$,输出变量有 $Y_{(A>B)}$ 、 $Y_{(A=B)}$ 、 $Y_{(A<B)}$ 3 个,任何时刻有且仅有一个输出变量有效。

1. 一位数值比较器

一位数值比较器的逻辑符号如图 3-22 所示。设 $A>B$ 时,有 $Y_{(A>B)}=1$; $A=B$ 时,有 $Y_{(A=B)}=1$; $A<B$ 时,有 $Y_{(A<B)}=1$,列出真值表,见表 3-9。

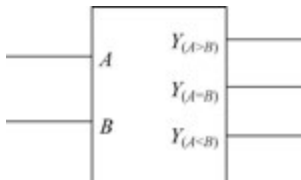


图 3-22 一位数值比较器的逻辑符号

表 3-9 一位数值比较器的真值表

输 入		输 出			输 入		输 出		
A	B	$Y_{(A<B)}$	$Y_{(A=B)}$	$Y_{(A>B)}$	A	B	$Y_{(A<B)}$	$Y_{(A=B)}$	$Y_{(A>B)}$
0	0	0	1	0	1	0	0	0	1
0	1	1	0	0	1	1	0	1	0

由表 3-9 可写出逻辑函数表达式:

$$\begin{cases} Y_{(A<B)} = \bar{A}B \\ Y_{(A=B)} = \bar{A}\bar{B} + AB = \overline{\bar{A}B + A\bar{B}} \\ Y_{(A>B)} = A\bar{B} \end{cases}$$

画出的逻辑电路图如图 3-23 所示。

2. 4 位数值比较器

两个 n 位二进制数 $A_{n-1}A_{n-2}\cdots A_1A_0$ 和 $B_{n-1}B_{n-2}\cdots B_1B_0$ 进行比较时,必须从高位开始比较。若在最高位有 $A_{n-1}>B_{n-1}$,则肯定有 $A>B$; 若有 $A_{n-1}<B_{n-1}$,则肯定有 $A<B$; 若有 $A_{n-1}=B_{n-1}$,则需要比较次高位,以此类推,可得出比较结果。

74LS85 是常用的 4 位数值比较器,其引脚图如图 3-24 所示,功能表见表 3-10。

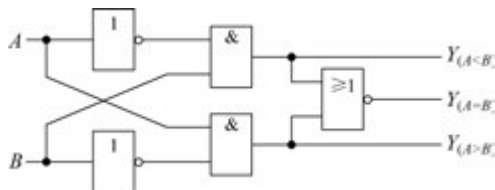


图 3-23 一位数值比较器逻辑电路图

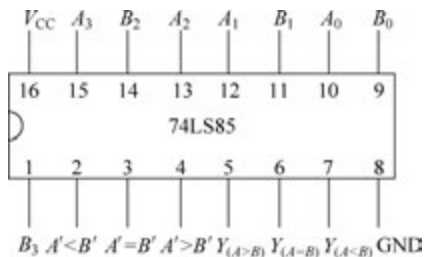


图 3-24 74LS85 引脚图

表 3-10 74LS85 功能表

比 较 输 入				级 联 输 入			输 出		
$A_3 \ B_3$	$A_2 \ B_2$	$A_1 \ B_1$	$A_0 \ B_0$	$A' > B'$	$A' < B'$	$A' = B'$	$Y_{(A > B)}$	$Y_{(A < B)}$	$Y_{(A = B)}$
$A_3 > B_3$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	1	0	0
$A_3 < B_3$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	0	1	0
$A_3 = B_3$	$A_2 > B_2$	$\times$	$\times$	$\times$	$\times$	$\times$	1	0	0
$A_3 = B_3$	$A_2 < B_2$	$\times$	$\times$	$\times$	$\times$	$\times$	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	$\times$	$\times$	$\times$	$\times$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	$\times$	$\times$	$\times$	$\times$	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	$\times$	$\times$	$\times$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	$\times$	$\times$	$\times$	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

74LS85 除了有两个 4 位数输入端,还有 3 个可用来扩展的级联输入端 $A' > B'$ 、 $A' < B'$ 和 $A' = B'$ 。由表 3-10 可以看出,数值比较器按照“从高位到低位,高位相等时比较低位”的原则工作。所以,当两个数各位均相等时,输出状态取决于级联输入的状态;若没有更低位参加运算时,级联输入端($A' > B'$)和($A' < B'$)接“0”、($A' = B'$)接“1”,才能产生两数相等的结果。读者可依据功能表列出逻辑表达式,用基本门做出逻辑电路图。

由 74LS85 可采用串联扩展或并联扩展的方法实现更多位数的数值比较器。图 3-25(a)给出了用串联扩展方法构造的 12 位数值比较器,图 3-25(b)给出了用并联扩展方法构造的 16 位数值比较器。

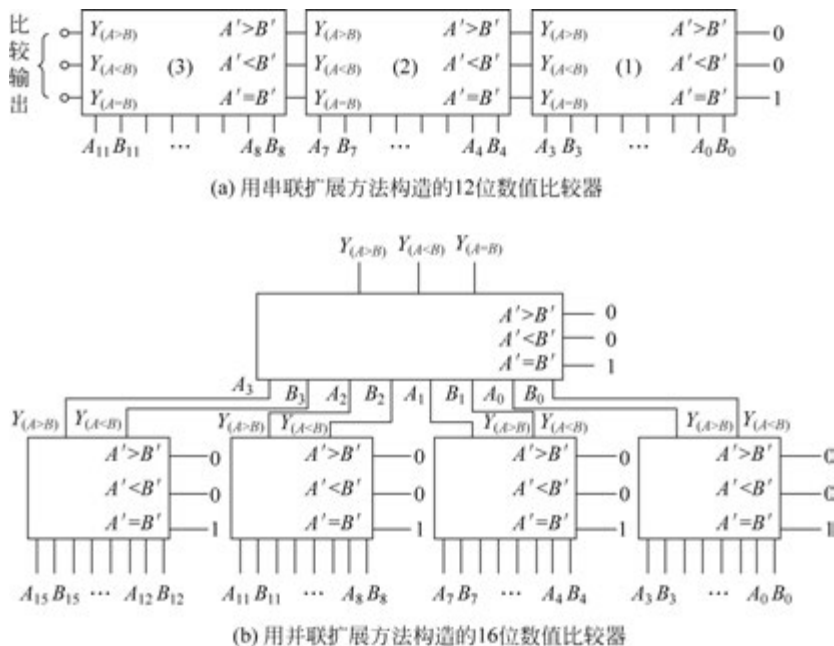


图 3-25 74LS85 构造多位数数值比较器

在图 3-25(a)中,芯片(1)送入的是 A 和 B 两个数的最低 4 位,所以其级联输入端($A' > B'$)和($A' < B'$)接“0”、($A' = B'$)接“1”。对于 12 位数的比较,若最高 4 位相等,则由中间 4 位的比较结果确定,所以芯片(2)的比较结果应该作为芯片(3)的条件,即芯片(2)的输出端分别接到芯片(3)的级联输入端。同理,若最高 8 位相等,则由最低 4 位的比较结果确定,所以芯片(1)的输出端应分别接到芯片(2)的级联输入端。

在图 3-25(b)中,将每 4 位作为一组进行比较,然后将比较结果再进行一次比较,最终得出结果。请读者自行分析其工作过程。

3.4.3 编码器和译码器

在日常生活中,常会遇到编码问题,如给每位学生编一固定的学号等。一般来说,用数字、文字或符号表示某一特定对象的过程称为编码,具有编码功能的逻辑电路被称为编码器,即将有特定意义的输入信息编成相应的若干位二进制代码输出的组合逻辑电路。

1. 编码器

1) 二进制编码器

在数字系统中,采用二进制进行编码,要表示的信息越多,所需的二进制位数也越多。



图 3-26 8 线-3 线编码器示意框图

1 位二进制代码有 $2(2^1)$ 种状态,2 位二进制代码有 $4(2^2)$ 种状态,以此类推, n 位二进制代码有 2^n 种状态。所谓二进制编码器,即用 n 位二进制代码对 2^n 个信号进行编码的电路。现在以 8 线-3 线编码器为例说明其工作原理。

如图 3-26 所示,8 线-3 线编码器用 3 位二进制数分别代表 8 个信号,8 个输入信号低电平有效,3 个输出信号高电平有效。任何一个时刻有且只能有一个输入信号有效,真值表见表 3-11。注意,框图中的小圆圈代表低电平有效,而不是“非”运算。

表 3-11 8 线-3 线编码器真值表

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

由表 3-11 可写出逻辑函数表达式:

$$\begin{cases} Y_2 = \overline{I_4} \overline{I_5} \overline{I_6} \overline{I_7} = I_4 + I_5 + I_6 + I_7 \\ Y_1 = \overline{I_2} \overline{I_3} \overline{I_6} \overline{I_7} = I_2 + I_3 + I_6 + I_7 \\ Y_0 = \overline{I_1} \overline{I_3} \overline{I_5} \overline{I_7} = I_1 + I_3 + I_5 + I_7 \end{cases}$$

根据表达式画出逻辑电路图。图 3-27 分别给出了由与非门以及或门构造的 8 线-3 线编码器。注意,用或门实现编码器时,输入量为高电平有效。

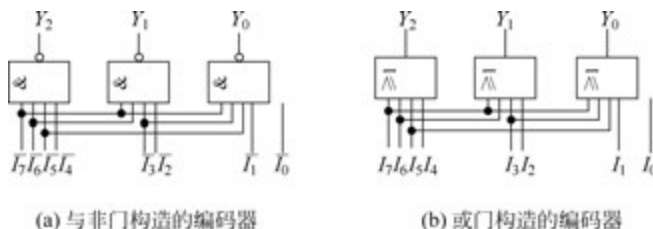


图 3-27 8 线-3 线编码器逻辑电路图

2) 优先编码器

若 8 线-3 线编码器在任何时刻有两个或两个以上有效信号同时输入,则输出就会发生混乱,优先编码器的提出解决了这一问题。

当多个有效信号同时输入时,优先编码器按照输入信号排定的优先顺序,只对其中优先级别最高的信号进行编码,所以,优先编码器具有单方面排斥的特性,即优先级别高的信号排斥优先级别低的信号。常用的集成器件有 8 线-3 线优先编码器 74LS148、10 线-4 线 8421BCD 优先编码器 74LS147 等。下面以 74LS148 为例,给出其工作原理。74LS148 芯片引脚图及示意图如图 3-28 所示。74LS148 真值表见表 3-12。

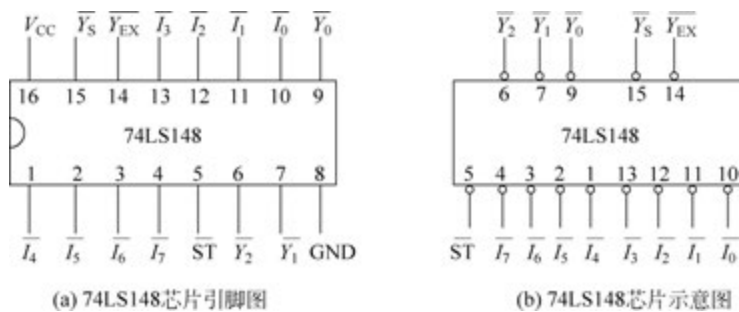


图 3-28 74LS148 芯片引脚图及示意图

表 3-12 74LS148 真值表

输 入									输 出				
$\overline{ST}$	$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_{EX}}$	$\overline{Y_S}$
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

续表

输 入									输 出				
$\overline{ST}$	$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_{EX}}$	$\overline{Y_S}$
0	×	×	×	×	×	0	1	1	0	1	0	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

74LS148 的输入信号和输出编码均为低电平有效, $\overline{I_7}$ 的优先级别最高, $\overline{I_6}$ 次之, 以此类推, $\overline{I_0}$ 最低。例如, 当 $\overline{I_7}$ 为 0 时, 不论其他输入是否有效, 输出均为“000”; 当 $\overline{I_5}$ 为“0”时, 只有当 $\overline{I_6}$ 和 $\overline{I_7}$ 为无效电平“1”时, 才输出“101”; 同理, 对于 $\overline{I_0}$ 来说, 只有其他输入信号均为无效信号“1”时, 才能输出“111”。

为了便于扩展电路, 芯片还增加了使能输入端 $\overline{ST}$ 及优先扩展端 $\overline{Y_{EX}}$ 、 $\overline{Y_S}$ 。使能输入端 $\overline{ST}$ 低电平有效, 当 $\overline{ST}=1$ 时, 电路处于禁止状态, 禁止编码, 输出全部为“1”; 当 $\overline{ST}=0$ 时, 电路处于正常工作状态, 允许编码。电路处于工作状态下, 当 8 个输入信号均无效时, 有 $\overline{Y_S}=0$, 所以使能输出端 $\overline{Y_S}=0$ 表示“电路正常工作, 但是无有效信号输入”, 通常接至低位芯片的 $\overline{ST}$ 端, $\overline{Y_S}$ 与 $\overline{ST}$ 配合可以实现多级编码器之间的优先级别的控制。当至少有 1 个输入信号时, 有 $\overline{Y_{EX}}=0$, 因而扩展输出端 $\overline{Y_{EX}}$ 是控制标志, $\overline{Y_{EX}}=0$ 表示“电路工作, 且有有效信号输入”。

【例 3-7】 试用 74LS148 构造 16 线-4 线优先编码器。

解: 每片 74LS148 可对 8 个信号进行编码, 要对 16 个信号编码, 需要两片芯片。16 线-4 线优先编码器电路图如图 3-29 所示。

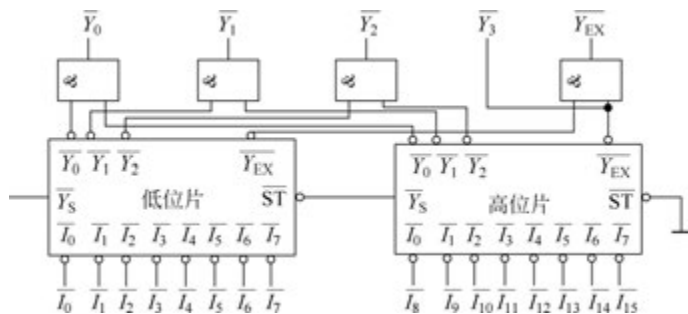


图 3-29 16 线-4 线优先编码器电路图

16 个输入信号均为低电平有效, 其中 $\overline{I_{15}}$ 优先级最高, $\overline{I_{14}}$ 次之, 以此类推, $\overline{I_0}$ 优先级最低。4 位输出为反码输出。若低位芯片有输入, 高位芯片无输入时, 如果输入有效信号为 $\overline{I_7}$, 则有高位芯片的 $\overline{Y_S}=0$, $\overline{Y_{EX}}=1$, 低位芯片的 $\overline{ST}=0$, 所以, 低位芯片工作, 高位芯片无有效信号输入, 输出 $\overline{Y_3} \overline{Y_2} \overline{Y_1} \overline{Y_0}=1000$ (反码), 即 0111(原码)。反之, 若高位芯片有输入, 低位芯片无输入时, 如果输入有效信号为 $\overline{I_{15}}$, 则有高位芯片的 $\overline{Y_S}=1$, $\overline{Y_{EX}}=0$, 低位芯片的 $\overline{ST}=1$, 所以低位芯片禁止, 高位芯片工作, 输出 $\overline{Y_3} \overline{Y_2} \overline{Y_1} \overline{Y_0}=0000$ (反码), 即 1111(原码)。

3) 二-十进制编码器

二-十进制编码器是将十进制数的 10 个数码 0~9 或其他 10 个信息转换为 8421BCD 码的逻辑电路,也称为 8421BCD 编码器。该编码器有 10 个输入端和 4 个输出端,即 10 线-4 线编码器。表 3-13 给出了二-十进制编码器的真值表。

表 3-13 二-十进制编码器的真值表

输 入	输 出			
十进制数(I)	Y_3	Y_2	Y_1	Y_0
0 (I_0)	0	0	0	0
1 (I_1)	0	0	0	1
2 (I_2)	0	0	1	0
3 (I_3)	0	0	1	1
4 (I_4)	0	1	0	0
5 (I_5)	0	1	0	1
6 (I_6)	0	1	1	0
7 (I_7)	0	1	1	1
8 (I_8)	1	0	0	0
9 (I_9)	1	0	0	1

由表 3-13 列出逻辑函数表达式:

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

根据表达式画出逻辑电路图,如图 3-30 所示。

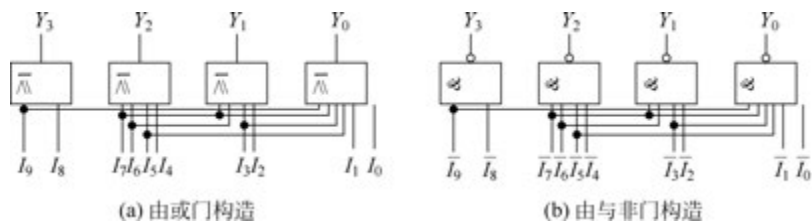


图 3-30 二-十进制编码器逻辑电路图

2. 译码器

译码是编码的逆过程,是将具有特定含义的一组代码“翻译”出来的过程,即把一种代码转换为另一种代码。能完成译码功能的电路称为译码器。译码器的种类很多,但其工作原理和分析设计方法大同小异。下面介绍广泛应用的 3 种译码器——二进制译码器、二-十进制译码器和显示译码器。

1) 二进制译码器

如图 3-31 所示的二进制译码器方框图,输入端为 n 个,输出端为 2^n 个, n 位代码的每种取值对应一个输出



图 3-31 二进制译码器方框图

变量,且 2^n 个输出中只有一个为1(或为0),其余全为0(或为1)。由于二进制译码器可以译出输入变量的全部状态,故又称为变量译码器。例如,3位二进制译码器有3个输入端和8个输出端,故称为3线-8线译码器,其真值表见表3-14。

表 3-14 3线-8线译码器真值表

输 入			输 出							
A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

由表3-4列出逻辑表达式:

$$\begin{cases} Y_0 = \overline{A_2} \overline{A_1} \overline{A_0} & Y_1 = \overline{A_2} \overline{A_1} A_0 \\ Y_2 = \overline{A_2} A_1 \overline{A_0} & Y_3 = \overline{A_2} A_1 A_0 \\ Y_4 = A_2 \overline{A_1} \overline{A_0} & Y_5 = A_2 \overline{A_1} A_0 \\ Y_6 = A_2 A_1 \overline{A_0} & Y_7 = A_2 A_1 A_0 \end{cases}$$

由上式可以看出,译码器的逻辑表达式均为输入信号的“与”运算,所以其逻辑图是由与门构成的阵列。3线-8线译码器的逻辑电路图如图3-32所示,若将电路中的与门用与非门替换,则可得到低电平有效的3线-8线译码器。常用的中规模器件有74LS138。图3-33给出了74LS138的逻辑符号,表3-15给出了其功能表。

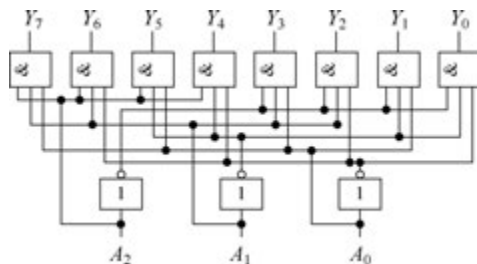


图 3-32 3线-8线译码器的逻辑电路图

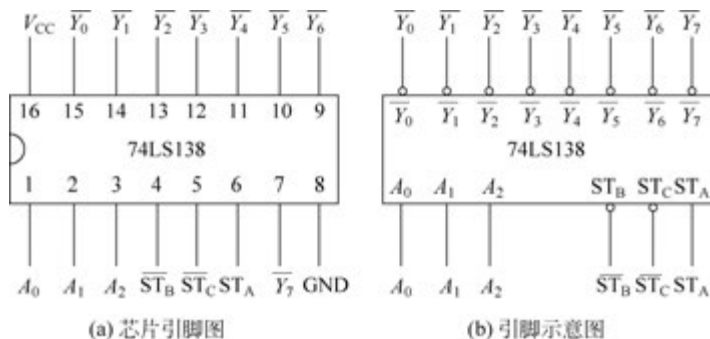


图 3-33 74LS138 的逻辑符号

表 3-15 74LS138 功能表

输 入					输 出							
$\overline{ST_A}$	$\overline{ST_B} + \overline{ST_C}$	A_2	A_1	A_0	$\overline{Y_7}$	$\overline{Y_6}$	$\overline{Y_5}$	$\overline{Y_4}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	×	×	×	×	1	1	1	1	1	1	1	1
1	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

由表 3-15 可看到,74LS138 除了 3 个编码输入端外,还有 3 个可用来扩展或级联芯片的片选控制端 $\overline{ST_A}$ 、 $\overline{ST_B}$ 和 $\overline{ST_C}$ 。只有当 $\overline{ST_A}=1$ 、 $\overline{ST_B}=0$ 和 $\overline{ST_C}=0$ 时,译码器才处于译码状态,输出端输出有效的低电平;否则,译码器禁止,所有的输出端都被封锁在高电平。

【例 3-8】 试用 74LS138 构造 4 线-16 线译码器。

解: 4 线-16 线译码器的输出表达式为

$$\begin{aligned}
 \overline{Y_0} &= \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0} & \overline{Y_1} &= \overline{A_3} \overline{A_2} \overline{A_1} A_0 & \overline{Y_2} &= \overline{A_3} \overline{A_2} A_1 \overline{A_0} & \overline{Y_3} &= \overline{A_3} \overline{A_2} A_1 A_0 \\
 \overline{Y_4} &= \overline{A_3} A_2 \overline{A_1} \overline{A_0} & \overline{Y_5} &= \overline{A_3} A_2 \overline{A_1} A_0 & \overline{Y_6} &= \overline{A_3} A_2 A_1 \overline{A_0} & \overline{Y_7} &= \overline{A_3} A_2 A_1 A_0 \\
 \overline{Y_8} &= A_3 \overline{A_2} \overline{A_1} \overline{A_0} & \overline{Y_9} &= A_3 \overline{A_2} \overline{A_1} A_0 & \overline{Y_{10}} &= A_3 \overline{A_2} A_1 \overline{A_0} & \overline{Y_{11}} &= A_3 \overline{A_2} A_1 A_0 \\
 \overline{Y_{12}} &= A_3 A_2 \overline{A_1} \overline{A_0} & \overline{Y_{13}} &= A_3 A_2 \overline{A_1} A_0 & \overline{Y_{14}} &= A_3 A_2 A_1 \overline{A_0} & \overline{Y_{15}} &= A_3 A_2 A_1 A_0
 \end{aligned}$$

该译码器需要 4 个代码输入端,而 74LS138 只有 A_2 、 A_1 和 A_0 共 3 个,所以需要将 74LS138 的一个扩展端作为 A_3 。如图 3-34 所示,将低位芯片的 $\overline{ST_B}$ 作为 A_3 的输入端,并令 $\overline{ST_A}=1$ 、 $\overline{ST_C}=0$ 。将高位芯片的 $\overline{ST_A}$ 作为 A_3 的输入端,并将其 $\overline{ST_B}$ 和 $\overline{ST_C}$ 作为使能端,当 $\overline{ST_B}=0$ 、 $\overline{ST_C}=0$ 时,电路处于译码状态。

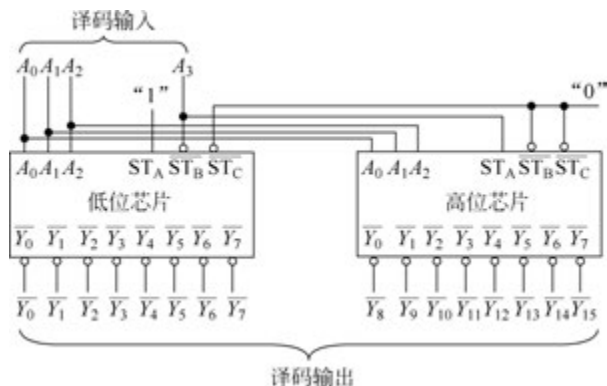


图 3-34 例 3-8 4 线-16 线译码器电路图

当输入编码为“0000~0111”时,有 $A_3=0$,所以低位芯片的 $ST_A=1, \overline{ST_B}=\overline{ST_C}=0$,而高位芯片的 $ST_A=0$,因此低位芯片工作,高位芯片禁止,译出的是低 8 个输出信号的其中一个。反之,当输入编码是“1000~1111”时,有 $A_3=1$,所以高位芯片的 $ST_A=1, \overline{ST_B}=\overline{ST_C}=0$,而低位芯片的 $\overline{ST_B}=1$,因此高位芯片工作,低位芯片禁止,译出的是高 8 个输出信号的其中一个。

2) 二-十进制译码器

二-十进制译码器是把二-十进制代码“翻译”成 10 个十进制数字信号的电路,其输入为十进制数的 4 位二进制编码(即 BCD 码),分别用 $A_3、A_2、A_1、A_0$ 表示;输出的是与 10 个十进制数字对应的 10 个信号,用 $\overline{Y_9} \sim \overline{Y_0}$ 表示。由于二-十进制译码器有 4 个输入端、10 个输出端,因此又称为 4 线-10 线译码器,其工作原理与 3 线-8 线译码器类似,不同的是,4 位编码组成的 16 种状态中,“1010~1111”没有对应的输出端,所以这 6 组编码被称为伪码。当输入伪码时,10 个输出端均处于无效状态,所以该译码器具有拒绝伪码的功能。常用的器件有 74LS42,其真值表见表 3-16,低电平输出有效。74LS42 芯片引脚图及示意图如图 3-35 所示。

表 3-16 74LS42 真值表

序号	输 入				输 出									
	A_3	A_2	A_1	A_0	$\overline{Y_9}$	$\overline{Y_8}$	$\overline{Y_7}$	$\overline{Y_6}$	$\overline{Y_5}$	$\overline{Y_4}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	0	1
2	0	0	1	0	1	1	1	1	1	1	1	0	1	1
3	0	0	1	1	1	1	1	1	1	1	0	1	1	1
4	0	1	0	0	1	1	1	1	1	0	1	1	1	1
5	0	1	0	1	1	1	1	1	0	1	1	1	1	1
6	0	1	1	0	1	1	1	0	1	1	1	1	1	1
7	0	1	1	1	1	1	0	1	1	1	1	1	1	1
8	1	0	0	0	1	0	1	1	1	1	1	1	1	1
9	1	0	0	1	0	1	1	1	1	1	1	1	1	1
伪码	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

3) 显示译码器

在数字系统或装置中,若要将数字量直观地显示出来,必须使用数码显示电路。数码显示电路由数码显示器和显示译码器电路构成。

(1) 数码显示器。

数码显示器即数码管,是用来显示数字、文字或符号的器件,一般有 3 种显示方式——字形重叠式、点矩阵式和分段式。目前,应用最普遍的是七段分段式显示,如测试仪的显示

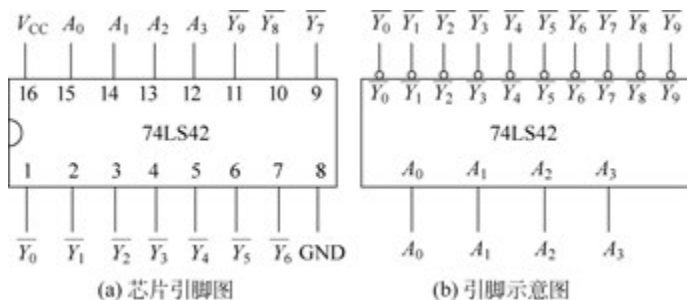


图 3-35 74LS42 芯片引脚图及示意图

屏幕,如图 3-36 所示。

七段显示器示意图如图 3-37(a)所示, $a \sim g$ 这 7 个发光段(h 为小数点)可组成不同的数字或符号。数码管有共阴极(见图 3-37(b))和共阳极(见图 3-37(c))两种,共阴极的数码显示器,公共阴极接地,某段的输入信号为高电平时,对应发光二极管被点亮;共阳极的数码显示器,公共阳极接高电平,某段的输入信号为低电平时,对应发光二极管被点亮。常用共阴极数码管 TS547 显示字形见表 3-17。



图 3-36 显示实例

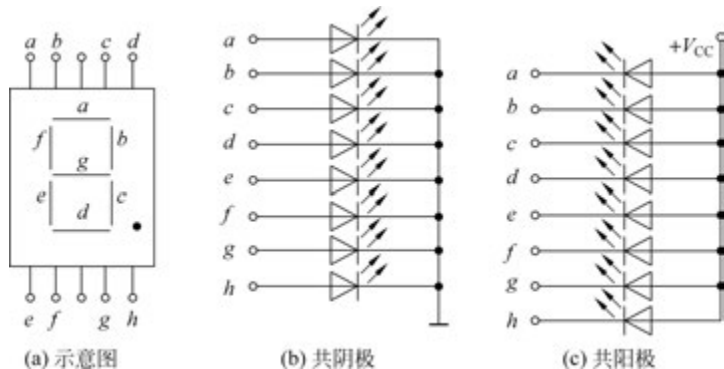


图 3-37 七段显示器示意图及构造

表 3-17 常用共阴极数码管 TS547 显示字形

输入信号							显示字形
a	b	c	d	e	f	g	
1	1	1	1	1	1	0	┌
0	1	1	0	0	0	0	├
1	1	0	1	1	0	1	┤
1	1	1	1	0	0	1	┘
0	1	1	0	0	1	1	└
1	0	1	1	0	1	1	┐
0	0	1	1	1	1	1	┌
1	1	1	0	0	0	0	├
1	1	1	1	1	1	1	┤
1	1	1	0	0	1	1	┘

(2) 显示译码器。

数码显示器需要与显示译码/驱动器配合,才能实现其显示功能。能够驱动各种显示器件,将用二进制代码表示的数字、文字、符号翻译成人们习惯的形式直观地显示出来的电路为显示译码器。七段显示译码器的输入为4位8421BCD码,输出为 $a \sim g$ 7个信号,这7个信号分别驱动七段显示器的7个发光段,所以也称为4线-7段译码器。常用的器件有74LS48,其芯片引脚图如图3-38所示,功能表见表3-18。

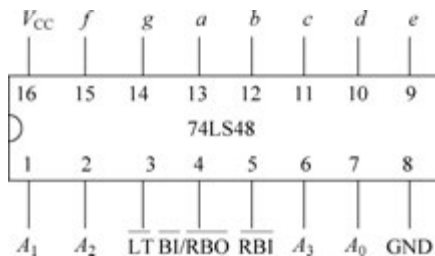


图 3-38 74LS48 芯片引脚图

表 3-18 74LS48 功能表

功能或 十进制数	输 入						输 出						
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	A_3	A_2	A_1	A_0	$\overline{\text{BI/RBO}}$	a	b	c	d	e	f g
$\overline{\text{LT}}$ (试灯)	0	×	×	×	×	×	1	1	1	1	1	1	1
$\overline{\text{RBI}}$ (动态灭零)	1	0	0	0	0	0	0	0	0	0	0	0	0
$\overline{\text{BI/RBO}}$ (灭灯)	×	×	×	×	×	×	0(输入)	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	0
1	1	×	0	0	0	1	1	0	1	1	0	0	0
2	1	×	0	0	1	0	1	1	1	0	1	1	0
3	1	×	0	0	1	1	1	1	1	1	0	0	1
4	1	×	0	1	0	0	1	0	1	1	0	0	1
5	1	×	0	1	0	1	1	1	0	1	1	0	1
6	1	×	0	1	1	0	1	0	0	1	1	1	1
7	1	×	0	1	1	1	1	1	1	1	0	0	0
8	1	×	1	0	0	0	1	1	1	1	1	1	1
9	1	×	1	0	0	1	1	1	1	1	0	0	1
10	1	×	1	0	1	0	1	0	0	0	1	1	0
11	1	×	1	0	1	1	1	0	0	1	1	0	0
12	1	×	1	1	0	0	1	0	1	0	0	0	1
13	1	×	1	1	0	1	1	1	0	0	1	0	1
14	1	×	1	1	1	0	1	0	0	0	1	1	1
15	1	×	1	1	1	1	1	0	0	0	0	0	0

由表3-18可看出,74LS48辅助控制信号 $\overline{\text{LT}}$ 、 $\overline{\text{RBI}}$ 和 $\overline{\text{BI/RBO}}$ 的作用如下。

(1) 试灯信号 $\overline{\text{LT}}$: 低电平有效,用来检测各发光段是否正常发光。当 $\overline{\text{LT}}=0$ 时,74LS48所驱动的七段显示器各发光段全部被点亮,与输入信号无关。电路要正常显示,则必须使 $\overline{\text{LT}}=1$ 。

(2) 动态灭零信号 $\overline{\text{RBI}}$: 低电平有效。当 $\overline{\text{LT}}=1$ 、 $\overline{\text{RBI}}=0$ 且输入信号 $A_3A_2A_1A_0$ 为“0000”时, 该位“0”不显示, 即“0”字被熄灭; 当输入信号不全为 0 时, 该位正常显示, 所以该信号可以用来消去无效的零。例如, 数字“0606”中的第一个“0”不需要显示, 可将 $\overline{\text{RBI}}$ 接地进行灭零。

(3) $\overline{\text{BI/RBO}}$ 是一个低电平有效的复合信号, 既可以作为输入端, 也可以作为输出端。 $\overline{\text{BI}}$ 为灭灯输入信号, 当 $\overline{\text{BI}}=0$ 时, 不论输入信号状态如何, 显示管全部熄灭, 与试灯信号正好相反, 所以可以作为是否显示的控制端。与其联结的信号 $\overline{\text{RBO}}$ 为动态灭零信号, 当 $\overline{\text{LT}}=1$ 、 $\overline{\text{RBI}}=0$ 且输入信号为“0000”时, 动态灭零信号 $\overline{\text{RBO}}$ 有效, 输出“0”, 其他情况输出为“1”。当有多位数字显示时, 可用此端进行连接。

一位数码的显示译码电路如图 3-39(a) 所示, 74LS48 的 7 个输出端分别与数码显示管的相应的输入端相连, 辅助控制端 $\overline{\text{LT}}$ 、 $\overline{\text{RBI}}$ 和 $\overline{\text{BI/RBO}}$ 均输入无效信号。可动态灭零的多位数码的显示译码电路示意图如图 3-39(b) 所示, 小数点位置固定, 有 4 位整数, 2 位小数。每位数码的显示译码电路构成与图 3-39(a) 类似, 不同的是, 在整数部分将高位数码译码器的 $\overline{\text{BI/RBO}}$ 与低位数码译码器的 $\overline{\text{RBI}}$ 相连, 而在小数部分要将低位数码译码器的 $\overline{\text{BI/RBO}}$ 与高位数码译码器的 $\overline{\text{RBI}}$ 相连。

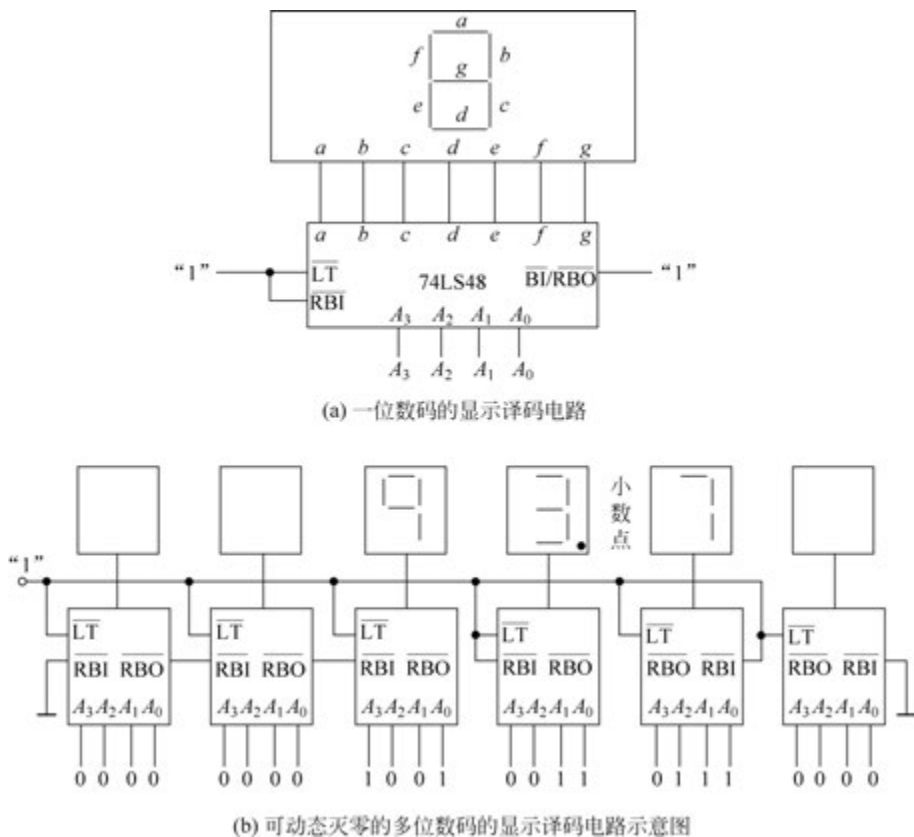


图 3-39 显示译码电路

3.4.4 数据选择器和数据分配器

1. 数据选择器

数据选择器也称为多路选择开关或多路调制器,由地址译码器和多路数字开关组成,其功能是在地址选择信号的作用下,从多个数据输入通道中选择某一个通道的数据由输出端送出。数据选择器方框图如图 3-40 所示,有 n 个地址输入端, 2^n 个数据输入端,通常称为 2^n 选 1 数据选择器。

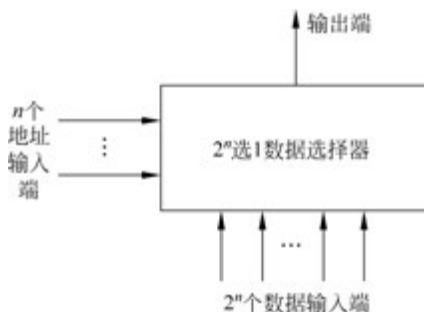


图 3-40 数据选择器方框图

1) 四选一数据选择器

四选一数据选择器是从 4 个数据通道中选择其中一个通道的数据输出。因为有 4 个数据通道,所以需要 4 个不同的地址选择信号控制每个通道,因此,地址输入端必须有 $2(4=2^2)$ 个。四选一数据选择器真值表见表 3-19, D 为数据输入端, A_1 、 A_0 为地址选择信号输入端, Y 为输出端。

表 3-19 四选一数据选择器真值表

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

由表 3-19 写出逻辑函数表达式:

$$\begin{aligned}
 Y &= D_0 \overline{A_1} \overline{A_0} + D_1 \overline{A_1} A_0 + D_2 A_1 \overline{A_0} + D_3 A_1 A_0 \\
 &= \sum_{i=0}^3 D_i m_i
 \end{aligned}$$

其中, m_i 为地址信号 A_1 、 A_0 对应的最小项符号。四选一数据选择器的逻辑电路图如图 3-41 所示。

2) 集成数据选择器

典型的集成数据选择器有双四选一数据选择器 74LS153、八选一数据选择器 74LS151。

双四选一数据选择器 74LS153 内含两个完全相同的四选一数据选择器。74LS153 芯片引脚图如图 3-42 所示,其功能表见表 3-20, $1D$ 和 $2D$ 分别代表两个数据选择器的 4 个数据输入端, $1Y$ 和 $2Y$ 分别代表两个数据选择器的输出端, $A_1 A_0$ 为两个四选一数据选择器共用的一组地址选择端,使能端 $\overline{1S}$ 和 $\overline{2S}$ 分别控制两个四选一数据选择器,便于控制电路工作及扩展,低电平有效,即 $\overline{S}=0$ 时,选择器工作, $\overline{S}=1$ 时,选择器禁止。

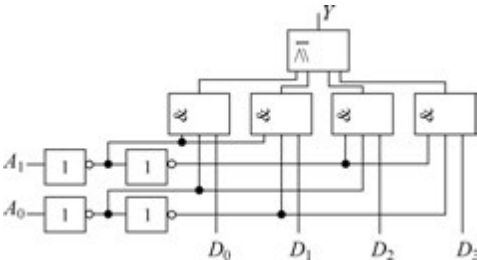


图 3-41 四选一数据选择器的逻辑电路图

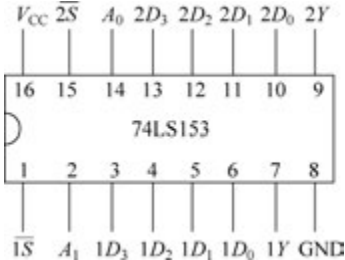


图 3-42 74LS153 芯片引脚图

表 3-20 74LS153 功能表

输 入					输 出	
$\overline{1S}/\overline{2S}$	$2D$	$1D$	A_1	A_0	$2Y$	$1Y$
1	×	×	×	×	0	0
0	$2D_0$	$1D_0$	0	0	$2D_0$	$1D_0$
0	$2D_1$	$1D_1$	0	1	$2D_1$	$1D_1$
0	$2D_2$	$1D_2$	1	0	$2D_2$	$1D_2$
0	$2D_3$	$1D_3$	1	1	$2D_3$	$1D_3$

八选一数据选择器 74LS151 的芯片引脚图如图 3-43 所示，其功能表见表 3-21，有 3 个地址选择端，一对互补的输出 Y 和 $\bar{Y}$ ，与 74LS153 一样，使能端 $\bar{S}$ 低电平有效，当 $\bar{S}=0$ 时，数据选择器工作， $\bar{S}=1$ 时，数据选择器被禁止。

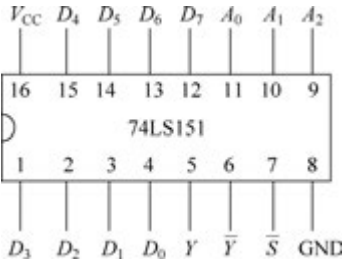


图 3-43 74LS151 芯片引脚图

表 3-21 74LS151 功能表

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

将两片八选一数据选择器 74LS151 相连,可得到十六选一数据选择器,如图 3-44 所示。高位芯片用来选通 $D_8 \sim D_{15}$ 8 个数据通道,低位芯片用来选通 $D_0 \sim D_7$ 8 个数据通道。十六选一数据选择器需要 4 个地址选择端,而 74LS151 只有 3 个地址选择端,所以需通过使能端再构造一个地址选择端。电路正常工作情况下,两个数据选择器中只有一个处于工作状态,所以可将两个使能端用非门连接作为地址选择信号 A_3 。如地址选择信号为“0111”时,有 $A_3=0, \overline{S_1}=0, \overline{S_2}=1$,所以低位芯片工作,高位芯片禁止,输出低位芯片 D_7 通道的数据。若地址选择信号为“1111”时,有 $A_3=1, \overline{S_1}=1, \overline{S_2}=0$,所以高位芯片工作,低位芯片禁止,输出高位芯片 D_7 通道的数据,即 D_{15} 的数据。

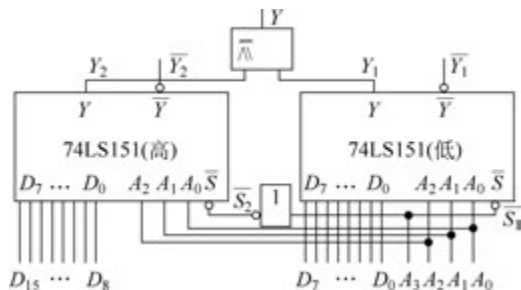


图 3-44 由 74LS151 构造的十六选一数据选择器

2. 数据分配器

数据分配器也称为多路分配器,其功能正好与数据选择器相反,可以根据需要将输入的数据从多个输出端中的任何一个输出,相当于单刀多掷开关。图 3-45 给出了数据选择器和数据分配器的功能示意图,从图中可清楚地看出二者的联系。真值表见表 3-22。



图 3-45 数据选择器和数据分配器的功能示意图

表 3-22 数据分配器真值表

D	输 入		输 出			
	A_1	A_0	Y_3	Y_2	Y_1	Y_0
	0	0	0	0	0	D
	0	1	0	0	D	0
	1	0	0	D	0	0
	1	1	D	0	0	0

由表 3-22 可写出逻辑函数表达式:

$$Y_0 = \overline{D} \overline{A_1} \overline{A_0} \quad Y_1 = \overline{D} \overline{A_1} A_0 \quad Y_2 = D \overline{A_1} \overline{A_0} \quad Y_3 = D \overline{A_1} A_0$$

一般地,数据分配器的功能可由译码器实现。图 3-46 给出了由 3 线-8 线译码器 74LS138 实现 1 路-8 路数据分配器的电路图,将 $\overline{ST_A}$ 和 $\overline{ST_C}$ 分别置“1”和“0”, $\overline{ST_B}$ 作为数

据输入端, A_2 、 A_1 、 A_0 作为地址信号输入端, $\overline{Y_0} \sim \overline{Y_7}$ 为输出端。当 $A_2 A_1 A_0$ 为“000”时, 选通输出端 $\overline{Y_0}$, 当输入数据 $D=0$ 时, 即 $\overline{ST_B}=0$, 译码器正常工作, 则有 $\overline{Y_0}=0$; 当输入数据 $D=1$ 时, 即 $\overline{ST_B}=1$, 译码器禁止, 则有 $\overline{Y_0}=1$, 由上可看出 $\overline{Y_0}$ 的输出值与 $\overline{ST_B}$ 的输入一致, 所以实现了数据分配器的功能。

将数据分配器和数据选择器结合在一起可以构成多路数据分时传送系统, 如图 3-47 所示。



图 3-46 1 路-8 路数据分配器

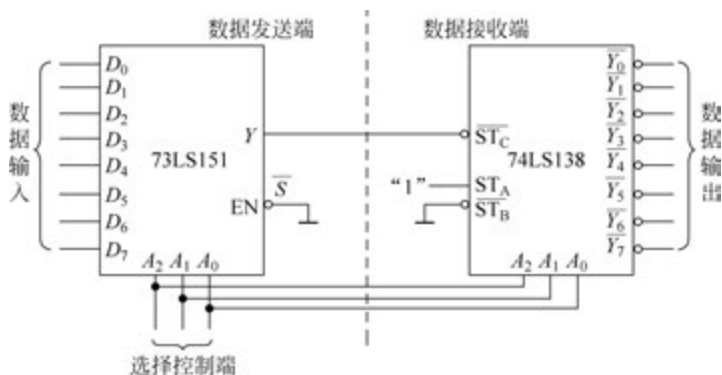


图 3-47 多路数据分时传送系统

3.4.5 综合应用实例

集成电路的集成度高、性能可靠、成本低, 用中大规模的集成电路构造的数字系统具有体积小、可靠性高等优点。下面给出几个用加法器、译码器、数据选择器设计组合逻辑电路的实例。

【例 3-9】 (1) 用全加器实现代码转换电路, 将 8421BCD 码转换为余 3 码; (2) 用全加器实现并行加法/减法器。

解: (1) 因为将 8421BCD 码加 3 (即 0011) 后, 即可得到余 3 码, 所以在加数 A 端送入 8421BCD 码, B 端置为“0011”, 经过全加器相加得到的输出即余 3 码。8421BCD 码转换为余 3 码电路图如图 3-48(a) 所示。

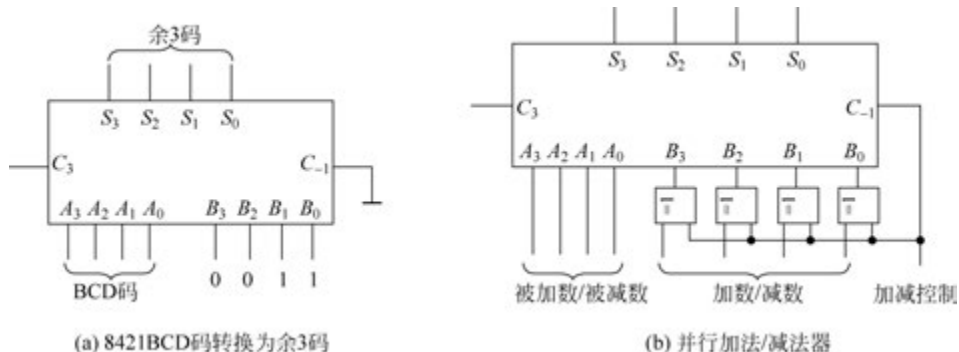


图 3-48 例 3-9 电路图

(2) 因为电路要同时实现加法运算和减法运算,所以用 C_{-1} 端作为加减控制端, A 端作为被加数(或被减数)输入端, B 端作为加数(或减数)输入端。当 $C_{-1}=0$ 时,有 $B \oplus 0 = B$,所以 B 端输入原码,电路完成 $A+B$ 的运算,即加法运算;当 $C_{-1}=1$ 时,有 $B \oplus 1 = \bar{B}$,所以 B 端输入反码,电路完成 $A + \bar{B} + 1$ 的运算, $A + \bar{B} + 1 = A - B$,即实现减法运算。并行加法/减法器电路图如图 3-48(b)所示。

【例 3-10】 利用 4 位数值比较器和逻辑门设计输血指示器,若绿灯亮,则表明血型匹配,可以输血;若红灯亮,则表明血型不匹配。

解: 一般来说,血型有 A、B、AB 及 O 型 4 种,所以每种血型需要用两位二进制代码表示,用“00”表示 O 型血,“01”“10”及“11”分别表示 A、B、AB 型血。假设 AB、CD 分别为输送和接受的血型, G 表示绿灯, R 表示红灯,真值表见表 3-23。

表 3-23 例 3-10 真值表

A	B	C	D	G	R	A	B	C	D	G	R
0	0	0	0	1	0	1	0	0	0	0	1
0	0	0	1	1	0	1	0	0	1	0	1
0	0	1	0	1	0	1	0	1	0	1	0
0	0	1	1	1	0	1	0	1	1	1	0
0	1	0	0	0	1	1	1	0	0	0	1
0	1	0	1	1	0	1	1	0	1	0	1
0	1	1	0	0	1	1	1	1	0	0	1
0	1	1	1	1	0	1	1	1	1	1	0

经分析可知,只要满足血型相同、输送 O 型血或者接受方是 AB 型血 3 个条件之一,就可以输血。

(1) 血型相同,即 $AB=CD$,可通过数值比较器对两位二进制数进行比较,若二者相等,则输出 $G=1$ 。

(2) 输送 O 型血,即 $AB=00$,则输出 $G=1$,所以有 $G = \bar{A} \cdot \bar{B} = \overline{A+B}$,可用或非门实现。

(3) 接受方是 AB 型血,即 $CD=11$,则输出 $G=1$,所以有 $G=CD$,可用与门实现。

将以上 3 部分电路的输出用或门相连即可构造出输血指示电路,如图 3-49 所示。

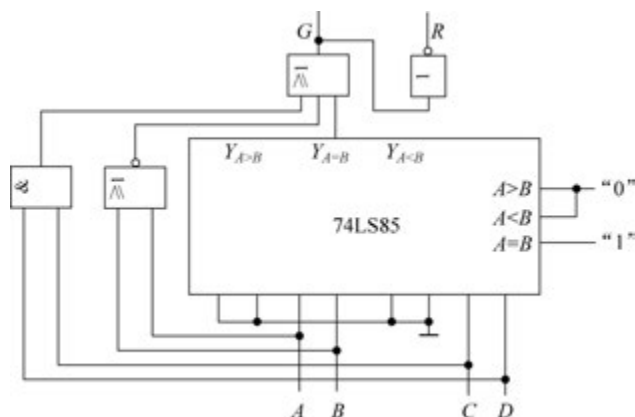


图 3-49 例 3-10 电路图

【例 3-11】 试用 3 线-8 线译码器 74LS138 以及与非门实现一位全加器。

解：由译码器的输出表达式可看出，译码器的输出均为最小项，所以 n 位译码器实际上是一个 n 变量的最小项输出器，又因为逻辑表达式都可以写成最小项表达式，所以任何组合逻辑函数都可以利用译码器实现。具体步骤如下。

(1) 写出函数的最小项表达式。因为 74LS138 是低电平有效，所以将函数表达式变换为与非-与非形式。

$$\begin{cases} S_i(A_i, B_i, C_{i-1}) = \sum m(1, 2, 4, 7) = \overline{m_1} \overline{m_2} \overline{m_4} \overline{m_7} \\ C_i(A_i, B_i, C_{i-1}) = \sum m(3, 5, 6, 7) = \overline{m_3} \overline{m_5} \overline{m_6} \overline{m_7} \end{cases}$$

(2) 确定 74LS138 输入变量，令 $A_2 = A_i, A_1 = B_i, A_0 = C_{i-1}$ 。

(3) 画出逻辑电路图，如图 3-50 所示。

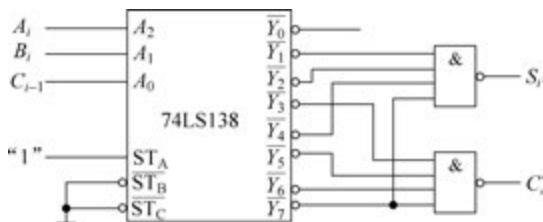


图 3-50 例 3-11 电路图

【例 3-12】 试用数据选择器 74LS153 实现逻辑函数 $L = \overline{A}\overline{B}C + \overline{A}B\overline{C} + AB$ 。

解：因为数据选择器的逻辑表达式具有标准与或表达式形式，提供地址变量的全部最小项，而且 D_i 可以当作一个变量对待，所以利用数据选择器的输入 D_i 选择地址变量组成的最小项 m_i ，可以实现任何所需的组合逻辑函数。 n 个地址变量的数据选择器在不增加门电路的情况下，最多可实现 $n+1$ 个变量的函数。如 3 个输入变量的逻辑函数可以用 2 个地址变量的 74LS153 实现。下面给出具体步骤。

(1) 确定逻辑函数的变量与数据选择器的变量间的对应关系，令 $A = A_1, B = A_0, C$ 由 $D_0 \sim D_3$ 确定，数据选择器的输出 Y 即函数的输出 L 。

(2) 写出逻辑函数 L 的标准与或式及数据选择器的表达式 Y ，对比二者确定 C 。

$$\begin{aligned} L &= \overline{A}\overline{B}C + \overline{A}B\overline{C} + AB = \overline{A}\overline{B}C + \overline{A}B\overline{C} + ABC + AB\overline{C} \\ &= \overline{A_1}\overline{A_0}C + \overline{A_1}A_0\overline{C} + A_1A_0C + A_1A_0\overline{C} = \overline{A_1}\overline{A_0}C + \overline{A_1}A_0\overline{C} + A_1A_0(C + \overline{C}) \end{aligned}$$

$$Y = D_0\overline{A_1}\overline{A_0} + D_1\overline{A_1}A_0 + D_2A_1\overline{A_0} + D_3A_1A_0$$

比较 L 和 Y ，可得

$$D_0 = C \quad D_1 = \overline{C} \quad D_2 = 0 \quad D_3 = 1$$

(3) 画出逻辑电路图，如图 3-51 所示。

【例 3-13】 试用数据选择器 74LS153 实现逻辑函数：

$$L(A, B, C, D) = \sum m(1, 3, 5, 9, 11)$$

解：由前可知，在不增加门电路的情况下， n 个地址变量的数据选择器最多可实现 $n+1$ 个变量的函数。若考虑门电路，则可以实现更多变量的函数。

逻辑函数 L 有 4 个变量，而 74LS153 只有两个地址选择端，令 $A = A_1, B = A_0$ ，则 C 和

D 都要由 $D_0 \sim D_3$ 确定。

$$\begin{aligned}
 L(A, B, C, D) &= \sum m(1, 3, 5, 9, 11) \\
 &= \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD \\
 &= \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}B(\bar{C}D + CD) \\
 &= \bar{A}\bar{B}D + \bar{A}B\bar{C}D + \bar{A}BD
 \end{aligned}$$

将上式与数据选择器的表达式相比,可得: $D_0 = D$ 、 $D_1 = \bar{C}D$ 、 $D_2 = D$ 、 $D_3 = 0$ 。

画出电路图,如图 3-52 所示。

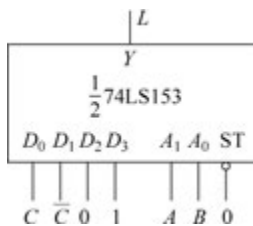


图 3-51 例 3-12 逻辑电路图

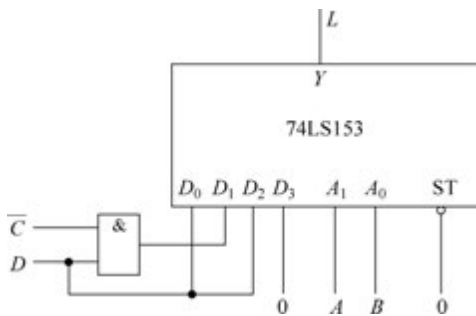


图 3-52 例 3-13 电路图

3.5 组合逻辑电路的险象

之前,我们是在电路稳定工作且假定门电路都处于理想状态的情况下对组合逻辑电路进行分析和设计的。然而,在实际中,所有的逻辑门都存在传输延迟时间,信号经导线传输时也需要一定时间,所以在组合逻辑电路中,当输入信号的状态改变时,输出端可能会出现不正常的干扰信号,使电路产生错误的输出,这种现象称为竞争冒险。

3.5.1 险象的产生

当输入信号经过多条路径传送后到达同一逻辑门的输入时,由于传送路径不同,导致信号到达该门的输入时间不一致的现象称为竞争。竞争分为临界竞争和非临界竞争。所谓临界竞争是指产生错误输出的竞争,而非临界竞争是指不会产生错误输出的竞争。临界竞争产生的错误输出会引起后级电路的错误输出,这种现象被称为冒险或险象。

图 3-53(a)给出了两个互补信号相与的电路图和波形图。因为非门的传输延迟,从波形图可以看出, $\bar{A}$ 信号滞后于 A 信号,所以在很短的时间间隔内, $\bar{A}$ 和 A 会同时出现高电平,两个互补信号相与时,应该有 $Y_1 = A \cdot \bar{A} = 0$ 为持续低电平,但却出现了高电平的干扰信号,这种现象即险象。同理,图 3-53(b)中两个互补信号相或时,应该有 $Y_2 = A + \bar{A} = 1$,但由于非门的传输延迟,两个互补信号在短时间内会同时为低电平,所以出现输出为低电平窄脉冲的情况。

冒险现象主要是由门电路的延迟时间产生的。对于低速运转的系统,出现的窄脉冲不会引起错误,但是,对于高速工作的数字系统来说,不正常的干扰信号会导致系统错误,无法

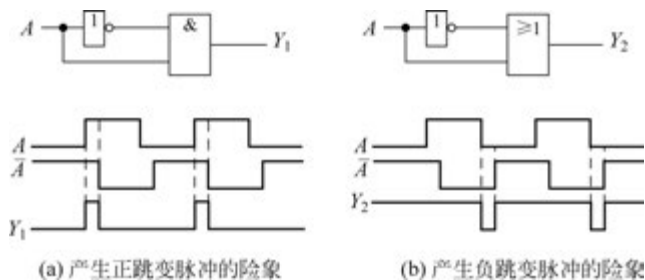


图 3-53 冒险产生原因

正常工作,所以必须克服这一现象。

3.5.2 险象的分类

依据输入信号变化前后,输出是否相同,险象可分为静态险象和动态险象。静态险象是指在输入变化而输出不应该发生变化的情况下,输出端产生瞬间的错误。动态险象是指在输入变化而输出应该发生变化的情况下,输出端在变化过程中产生瞬间的错误。组合电路中的动态险象一般由静态险象引起,如果消除了电路中的静态险象,也就消除了动态险象。

依据错误脉冲的极性,险象可分为“1”型险象和“0”型险象:若错误输出信号为正脉冲,则为“1”型险象;若错误输出信号为负脉冲,则为“0”型险象。静态险象、动态险象、“1”型险象及“0”型险象的组合情况如图 3-54 所示。

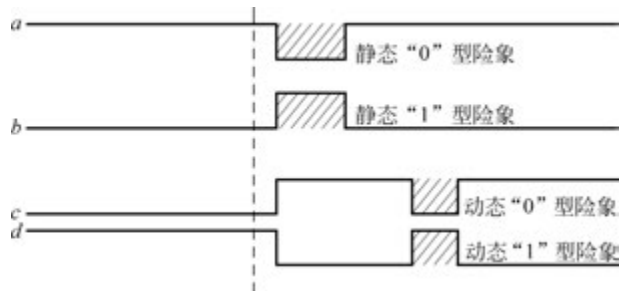


图 3-54 险象分类示意图

例如,波形 a 中的输出信号应该为持续高电平,但是,当输入信号发生变化时(虚线表示变化时刻),输出信号产生了瞬间的错误,且错误信号为低电平,所示该险象为静态“0”型险象。

3.5.3 险象的判断

1. 代数法判断

代数法判断是指通过逻辑函数表达式判断电路中是否存在险象。在逻辑函数表达式中,若某个变量(如变量 A)同时以原变量和反变量两种形式出现,则其具备竞争条件,此时将其余变量置“0”或置“1”后,如果该表达式变为 $Y = A \cdot \bar{A}$,则产生“1”型险象;如果表达式变为 $Y = A + \bar{A}$,则产生“0”型险象。

【例 3-14】 试判断图 3-55 所示电路是否存在险象。

解: 该电路图对应的逻辑表达式为 $Y = A\bar{B} + BC$, 因为表达式中变量 B 的原变量和反变量同时出现, 所以具备竞争条件。又因为当 $A = C = 1$ 时, 有 $Y = \bar{B} + B$, 所以该电路中存在竞争冒险。

2. 卡诺图法判断

将逻辑函数填入卡诺图, 圈出卡诺圈, 若存在相切但不相交的卡诺圈, 则逻辑函数存在竞争冒险。将例 3-14 中的逻辑函数填入图 3-56 中, 并圈出卡诺圈, 从图中可看到两个卡诺圈相切但不相交, 所以存在险象。

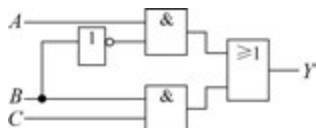


图 3-55 例 3-14 电路图

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

图 3-56 例 3-14 卡诺图

3. 其他方法判断

除了采用代数法和卡诺图法判断有无险象外, 还可以采用实验法及使用计算机辅助分析手段进行判断。

采用实验法比较直观、可靠, 将所有可能出现的状态加至电路输入端, 观察输出端是否出现不正常的干扰信号即可。使用计算机辅助手段是指通过在计算机上模拟数字逻辑电路的工作情况, 以此检查是否存在竞争冒险现象。

3.5.4 险象的消除

1. 消除互补变量

可通过消去互补变量消除险象。例如, 函数 $Y = (A + B)(\bar{B} + C)$, 当 $A = C = 0$ 时, 有 $Y = B \cdot \bar{B}$, 存在险象。将函数展开并消去互补变量后, 有 $Y = A\bar{B} + AC + BC$, 逻辑电路中不会存在竞争冒险现象。

2. 增加冗余项

保证原函数不变, 增加冗余项消除险象。例如, 在例 3-14 的卡诺图中增加一个卡诺圈(图 3-57 中的虚线圈), 表达式将变为 $Y = A\bar{B} + AC + BC$, 逻辑电路中的险象消除。因为表达式中的 AC 对于函数来说是多余的, 所以称为冗余项。

3. 利用滤波电容

因为险象产生的干扰脉冲非常窄, 所以可在电路的输出端并联一个小容量($4 \sim 20\text{pF}$)的电容, 以此消除险象。利用滤波电容消除险象如图 3-58 所示。

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

图 3-57 增加冗余卡诺圈的卡诺图

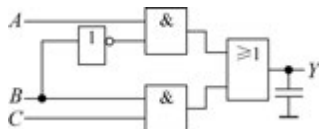


图 3-58 利用滤波电容消除险象

3.6 本章小结

组合逻辑电路是指电路的任意时刻的输出状态只与该时刻的输入状态有关,而与该时刻之前的状态无任何关系,输出与输入的关系具有即时性。电路由逻辑门电路构成,不含任何存储元件。

组合逻辑电路研究的两个主要问题是分析与设计。电路的分析与设计为互逆的过程,所以掌握好分析方法尤为重要。组合逻辑电路除了采用基本的逻辑门实现外,还可以采用中规模集成器件实现。常用的中规模集成器件有全加器、编码器、译码器、数据选择器、数据分配器等,本章着重介绍了这些器件的基本工作原理、逻辑功能、使用方法及应用举例,最后简要介绍了组合逻辑电路中的竞争冒险现象及消除冒险现象的常用方法。

3.7 习题和自测题

习题(答案见附录 D)

1. 某组合逻辑电路有 4 个输入端和 1 个输出端,当输入信号中没有“1”输入或者输入奇数个“1”时,输出信号为“1”。试列出真值表,并写出其最简与或表达式。
2. 分析图 3-59 中的两个电路,写出逻辑表达式,并说明电路功能。

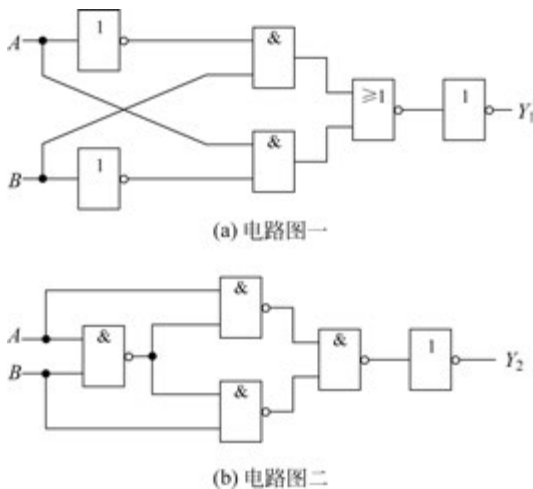


图 3-59 习题 2

3. 已知某组合逻辑电路的输入 A 、 B 、 C 和输出 F 的波形如图 3-60 所示,不计逻辑门的延时,试写出 F 的最简与或式。
4. 试分析图 3-61 所示的逻辑电路图,说明输出端与输入端的关系。其中, A 、 B 为数据输入端, S_1 、 S_2 为控制输入端, Y 为输出端。
5. 试用两级最少的与非门组成与图 3-62 所示电路具有相同逻辑功能的电路。
6. 试设计一个“逻辑不一致”电路。要求当 4 个输入变量取值不一致时,输出为“1”;当 4 个输入变量取值一致时,输出为“0”。

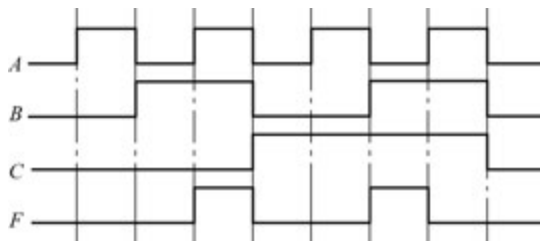


图 3-60 习题 3

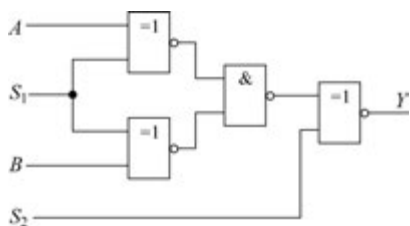


图 3-61 习题 4

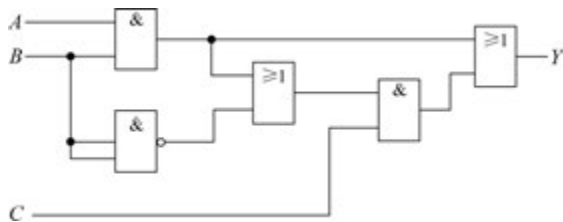


图 3-62 习题 5

7. 某电路的输入 X 及输出 Y 均为 3 位二进制数,要求: X 值不能大于 6; 当 X 大于或等于 0,且小于或等于 3 时,有 $Y=X$; 当 X 大于或等于 4,且小于或等于 6 时,有 $Y=X+1$ 。试用两级最少的与非门设计实现该逻辑电路。

8. 试用与非门设计一个可以完成以下 4 个功能的多功能逻辑电路。

(1) $F=AB$; (2) $F=\overline{A \oplus B}$; (3) $F=A+B$; (4) $F=\overline{AB}$

9. 试设计一个判偶电路: 当输入偶数个“1”时,输出为“1”; 否则输出“0”。假设该电路有 4 个输入端。

10. 试用与、或、非逻辑门实现两位二进制数的比较电路,输出比较结果。

11. 试用 4 位二进制加法器设计一个代码转换器,将余 3 代码转换为 8421BCD 码。

12. 试用一片 4 位比较器构成一个数值范围指示器,其输入变量为 8421BCD 码表示的一位十进制数 X 。当 $X \geq 5$ 时,指示器输出 1; 否则输出 0。

13. 试用四选一数据选择器实现下列函数。

(1) $F=(A+\overline{B})(\overline{B}+C)$

(2) $F=B\overline{C}+\overline{A}\overline{C}D+A\overline{C}D+\overline{A}\overline{B}CD+A\overline{B}C\overline{D}$

(3) $F(A,B,C,D) = \sum m(0,2,3,4,5,7) + \sum d(8,10,15)$

14. 试用八选一数据选择器实现 13 题中的函数。

15. 试用八选一数据选择器实现全加器的功能。

16. 试用 74LS138 和相应的逻辑门实现下列逻辑函数,画出逻辑电路图。

(1) $F=\overline{A}\overline{C}+B\overline{C}+\overline{A}\overline{B}C$

(2) $F(A,B,C) = \sum m(0,2,4,5,7)$

17. 试用 74LS138 译码器设计一个控制电路对 3 台设备进行控制,当设备出现故障时,由不同的指示灯指示。当设备正常工作时,指示灯绿灯亮;当有一台设备故障时,指示灯红灯亮;当有两台设备故障时,指示灯黄灯亮;当 3 台设备都故障时,指示灯红灯、黄灯、绿灯都亮。

18. 试用 74LS138 和逻辑门实现以下多输出函数,并画出电路图。

$$\begin{cases} F_1 = \overline{A}C + B\overline{C} \\ F_2 = \overline{A}BC + A\overline{B}C + AC \end{cases}$$

19. 判断下列各电路逻辑函数是否存在逻辑冒险。若存在,试用修改逻辑设计的方法进行消除。

(1) $F = A\overline{B} + \overline{A}B$

(2) $F = \overline{A}D + A\overline{B} + A\overline{C}\overline{D}$

(3) $F = (A+C)(\overline{A}+B)$

(4) $F = \overline{\overline{A+B} + \overline{A+B}}$

自测题(答案见附录 D)

一、单选题

- 输出 $F = B\overline{C} + A\overline{B}C$ 的逻辑电路由()组成。
 - 两个与门和一个或门
 - 两个与门、一个或门和两个非门
 - 两个或门、一个与门和两个非门
 - 两个与门、一个或门和一个非门
- 与运算可以由()产生。
 - 两个与非门
 - 3 个与非门
 - 一个或非门
 - 两个或非门
- 或运算可以由()产生。
 - 两个或非门
 - 3 个与非门
 - 4 个与非门
 - 答案 A、B 都正确
- 一个半加器的特点是()。
 - 有 2 个输入和 2 个输出
 - 有 3 个输入和 2 个输出
 - 有 2 个输入和 3 个输出
 - 有 2 个输入和 1 个输出
- 一位全加器的特点是()。
 - 有 2 个输入和 2 个输出
 - 有 3 个输入和 2 个输出
 - 有 2 个输入和 3 个输出
 - 有 2 个输入和 1 个输出
- 一个全加器的输入为 $A_i=1$ 、 $B_i=1$ 、 $C_{in}=1$, 则其输出为()。
 - $S_i=0, C_{out}=0$
 - $S_i=0, C_{out}=1$
 - $S_i=1, C_{out}=0$
 - $S_i=1, C_{out}=1$
- 一个 4 位并行加法器可以用于()。
 - 两个 2 位二进制数相加
 - 两个 4 位二进制数相加
 - 顺序相加 4 个位
 - 一次 4 位相加
- 如果将一个 4 位并行加法器扩展为一个 8 位加法器, 则必须()。
 - 使用 4 个 4 位不相连的加法器
 - 使用两个 4 位加法器, 并将其中一个加法器的和输出与另一个加法器的进位输入相连
 - 使用 8 个 4 位不相连的加法器
 - 使用两个 4 位加法器, 并将其中一个加法器的进位输出与另一个加法器的进位输入相连

9. 如果一个比较器 74LS85 的输入为 $A=1011, B=1001$, 则输出为()。
- A. $F_{A>B}=0, F_{A<B}=0, F_{A=B}=1$ B. $F_{A>B}=0, F_{A<B}=1, F_{A=B}=0$
 C. $F_{A>B}=1, F_{A<B}=0, F_{A=B}=0$ D. $F_{A>B}=1, F_{A<B}=1, F_{A=B}=0$
10. 一个低输出有效的 4/16 译码器, 当 $\overline{Y}_{12}=0$ 时, 译码器的输入是()。
- A. $A_3A_2A_1A_0=1010$ B. $A_3A_2A_1A_0=1110$
 C. $A_3A_2A_1A_0=1100$ D. $A_3A_2A_1A_0=0100$
11. 一个共阴极七段数码管的译码电路, 当输入为 0101 时, 则其七段输出 a, b, c, d, e, f, g 的值分别等于()。
- A. 1011011 B. 1011111 C. 0100100 D. 0100000
12. 如果一个高位优先的 8 线-3 线优先编码器, 输出为高电平有效, 当其 4 个输入 I_0, I_2, I_5, I_7 都为有效电平时, 则优先编码器的输出编码为()。
- A. 101 B. 100 C. 111 D. 000
13. 数据选择器与()基本相同。
- A. 多路复用器 B. 多路分配器 C. 编码器 D. 译码器
14. 一个八选一数据选择器, 有()。
- A. 1 个数据输入、8 个数据输出和 3 个地址输入
 B. 8 个数据输入、1 个数据输出和 3 个地址输入
 C. 3 个数据输入、1 个数据输出和 8 个地址输入
 D. 1 个数据输入、3 个数据输出和 8 个地址输入
15. 某电路如图 3-63 所示, 74LS151 是八选一数据选择器。该电路的逻辑功能是()。
- A. $F = \sum m(4, 8, 9, 13)$ B. $F = \sum m(6, 8, 9, 13)$
 C. $F = \sum m(6, 7, 8, 9, 12, 15)$ D. $F = \sum m(6, 8, 13, 14)$
16. 32 路数据分配器的地址输入(选择控制)端有()个。
- A. 3 B. 4 C. 5 D. 16
17. 由一片 74LS283(4 位超前进位加法器)构成的某电路如图 3-64 所示, 若输入信号 A 的取值为 0101, 则输出信号 S 的取值为()。
- A. 0010 B. 1001 C. 0111 D. 1000

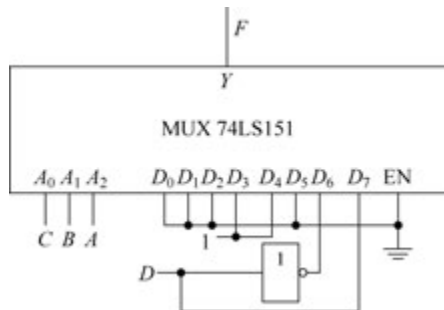


图 3-63 单选题 15

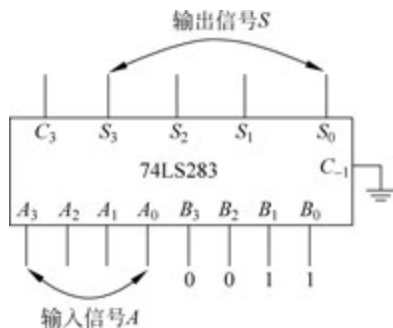


图 3-64 单选题 17

18. 由 74LS138 译码器和逻辑门构成的某电路如图 3-65 所示, 电路的输入信号 A 、 B 、 C 和输出信号 Z 之间的逻辑函数(写成最小项表达式)为()。

A. $Z = \sum m(1, 2, 4, 7)$

B. $Z = \sum m(0, 3, 5, 6)$

C. $Z = \sum m(1, 3, 4, 6)$

D. $Z = \sum m(0, 2, 5, 7)$

19. 某组合电路的输入信号 A 、 B 、 C 和输出信号 F 的波形关系如图 3-66 所示, 其函数表达式为()。

A. $F = \sum m(1, 2, 5, 6)$

B. $F = \sum m(0, 3, 4, 7)$

C. $F = \sum m(1, 4, 5, 6)$

D. $F = \sum m(0, 2, 3, 7)$

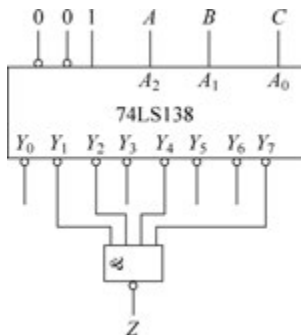


图 3-65 单选题 18

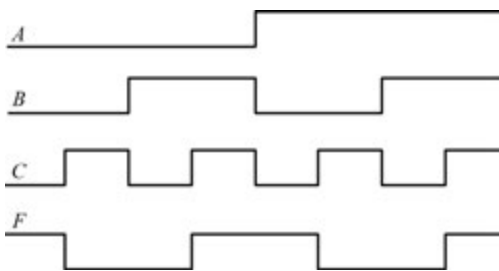


图 3-66 单选题 19

20. 某电路如图 3-67 所示, 该电路的逻辑表达式为()。

A. $F = \overline{AABC} + \overline{BABC} + \overline{CABC}$

B. $F = \overline{\overline{A}ABC} + \overline{\overline{B}ABC} + \overline{\overline{C}ABC}$

C. $F = \overline{ABC} + ABC$

D. $F = \overline{A+B+C} + ABC$

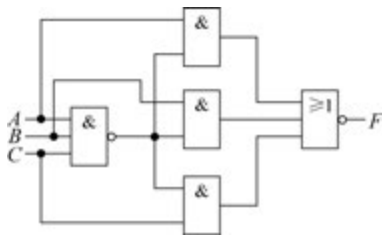


图 3-67 单选题 20

21. 用四选一数据选择器实现函数 $Y = A_1A_0 + \overline{A_1}A_0$, 应使()。

A. $D_0 = D_2 = 0, D_1 = D_3 = 1$

B. $D_0 = D_2 = 1, D_1 = D_3 = 0$

C. $D_0 = D_1 = 0, D_2 = D_3 = 1$

D. $D_0 = D_1 = 1, D_2 = D_3 = 0$

22. 4 线-10 线译码器, 输入信号端有()个。

A. 10

B. 2

C. 3

D. 4

23. 下列各函数相等, 其中无冒险现象的逻辑函数是()。

A. $F = AC + \overline{BC} + \overline{CD}$

B. $F = \overline{CD} + \overline{BC} + \overline{ACD}$

C. $F = AC + \overline{BC} + \overline{CD} + \overline{BD} + \overline{AB}$

D. $F = AC + \overline{CD} + \overline{BCD} + \overline{BD}$

24. 用低电平输出有效的译码器实现组合逻辑电路时, 还需要()。

- A. 与非门 B. 或非门 C. 与门 D. 或门
25. 逻辑函数 $F = \overline{A}C + AB + \overline{B}\overline{C}$, 当变量的取值为()时, 不出现冒险现象。
 A. $B=C=1$ B. $B=C=0$
 C. $A=1, C=0$ D. $A=0, B=0$
26. 在 8 线-3 线优先编码器(74LS148)中, 8 条数据输入线 $\overline{I_0} \cdots \overline{I_7}$ 同时有效时, 优先级最高为 $\overline{I_7}$ 线, 则输出线 $\overline{Y_2} \overline{Y_1} \overline{Y_0}$ 的值应是()。
 A. 000 B. 010 C. 101 D. 111
27. ()不是消除竞争冒险的措施。
 A. 接入滤波电路 B. 修改逻辑设计
 C. 加入选通脉冲 D. 屏蔽输入信号的尖峰干扰
28. 采用 4 位比较器(74LS85)对两个 4 位数比较时, 先比较()位。
 A. 最低 B. 最高 C. 次高 D. 次低
29. 用 8 线-3 线译码器 74LS138 和辅助门电路实现逻辑函数 $Y = A_2 + \overline{A_2} \overline{A_1}$, 应()。
 A. 用与非门, $Y = \overline{Y_0} \overline{Y_1} \overline{Y_4} \overline{Y_5} \overline{Y_6} \overline{Y_7}$
 B. 用与门, $Y = \overline{Y_2} \overline{Y_3}$
 C. 用或门, $Y = \overline{Y_2} + \overline{Y_3}$
 D. 用或门, $Y = \overline{Y_0} + \overline{Y_1} + \overline{Y_4} + \overline{Y_5} + \overline{Y_6} + \overline{Y_7}$
30. 串行加法器的进位信号采用()传递, 而并行加法器的进位信号采用()传递。
 A. 超前, 逐位 B. 逐位, 逐位 C. 逐位, 超前 D. 超前, 超前
31. 数字逻辑的二值逻辑中不能直接表示的是()。
 A. 事情的真和假 B. 灯泡的亮和灭
 C. 几何中的曲线和折线 D. 电路中电平的高与低
32. 在下列逻辑电路中, 不是组合逻辑电路的是()。
 A. 数据分配器 B. 二进制译码器
 C. 计数器 D. 比较器
33. 设计一个组合逻辑电路, 其输入是 3 位二进制数 $B = B_2 B_1 B_0$, 输出是 $Y = 2B$, 则输出变量有()个。
 A. 2 B. 4 C. 6 D. 8
34. 要设计实现任何所需的组合逻辑函数, 可以使用下列集成器件中的()。
 A. 半加器 B. 译码器 C. 编码器 D. 数值比较器
35. 组合逻辑电路的输出仅与()有关。
 A. 过去的输入 B. 当前的输入
 C. 电源电压 D. 内部的存储状态
36. 在组合逻辑电路中, 若要使输出 F 为 1, 则输入变量 A, B, C 必须满足的关系是()。
 A. $A=B=C=1$
 B. $A+B+C=1$ (此处的“+”表示逻辑或)
 C. $A=B$ 且 $C=0$

- D. $A \cdot B \cdot C = 1$ (此处的“ $\cdot$ ”表示逻辑与)
37. 编码器的主要功能是将()转换为()。
- A. 高电平信号,低电平信号
B. 特定的十进制数或有效电平,一组二进制代码
C. 一组二进制代码,特定的十进制数
D. 低电平信号,高电平信号
38. 多路选择器(MUX)是一种()电路。
- A. 反馈 B. 时序 C. 延时 D. 组合
39. 在组合逻辑电路设计中,为了消除竞争冒险现象,可以采取的措施不包括()。
- A. 选通脉冲宽度大于信号传输延迟时间
B. 接入滤波电容
C. 增加输入变量的数目
D. 修改设计,使输出端信号变化沿变得缓慢
40. 下列电路中,()的输出与输入变量之间的逻辑关系是固定的,不随时间变化。
- A. 组合逻辑电路 B. 时序逻辑电路
C. 存储逻辑电路 D. 反馈逻辑电路

二、判断题

1. 与非门不能用来产生与的功能。 ()
2. 或非门不能用来产生与的功能。 ()
3. 非或逻辑等价于与非。 ()
4. 一个比较器可以确定两个二进制数是否相等。 ()
5. 编码器基本是实现译码器操作的逆过程。 ()
6. 数据选择器是一个逻辑电路,它允许数字信息从单一的信号线传送到几条线路上。 ()
7. 加法器只能实现加法运算,不能实现其他功能。 ()
8. 将有特定意义的输入信息编成相应的若干位二进制代码输出的组合逻辑电路是编码器。 ()
9. 组合逻辑电路中,有冒险就一定有竞争,有竞争也就一定有冒险。 ()
10. 超前进位加法器是将各级加的进位预先进行运算,再同步求各级的和,以此减小进位的传输延迟。 ()
11. 若编码器中有 50 个编码对象,则要求输出二进制代码的位数为 5 位。 ()
12. 组合逻辑电路有记忆功能。 ()
13. 数字电路中用 1 和 0 分别表示两种状态,二者无大小之分。 ()
14. 4 个全加器可以组成一个串行进位的 4 位数加法器。 ()
15. 数据选择器和译码器的输出可以提供变量的所有最小项,因此可以用它们设计实现任何组合逻辑函数。 ()
16. 在组合逻辑电路中,竞争产生的原因主要是电路中使用了不同型号的元器件。 ()

17. 卡诺图是一种用于化简布尔表达式的图形工具,也适用于组合逻辑电路的设计和分析。 ()
18. 在组合逻辑电路中,所有的门电路(如与门、或门、非门等)都是可逆的。 ()
19. 组合逻辑电路中的冒险现象只存在于输出端,与输入端无关。 ()
20. 在组合逻辑电路中,竞争冒险现象是由于输入信号同时变化引起的。 ()