

第5章

组合逻辑电路



知识导学



习题答案

在数字系统中,常用的各种逻辑电路按其结构、工作原理和逻辑功能可分为两大类:组合逻辑电路和时序逻辑电路。前面学过的门电路就属于最简单的组合逻辑电路。组合逻辑电路不但能独立完成各种复杂的逻辑功能,而且是时序逻辑电路的组成部分,它在数字系统中的应用十分广泛。

本章首先介绍组合逻辑电路的特点,然后介绍组合逻辑电路分析和设计的基本方法,最后介绍组合逻辑电路中的竞争-冒险。

5.1 组合逻辑电路的特点

由若干逻辑门组成的具有一组输入和一组输出的非记忆性逻辑电路,即为组合逻辑电路。它任意时刻产生的稳定输出值,仅取决于该时刻各输入值的组合,而与电路原来的状态无关,其一般结构框图如图 5.1 所示。

图 5.1 中, $X_0 \sim X_{m-1}$ 是电路的 m 个输入信号, $Y_0 \sim Y_{n-1}$ 是电路的 n 个输出信号。输出信号是输入信号的函数,表示为

$$Y_i = F_i(X_0, X_1, \dots, X_{m-1}), \quad i = 0, 1, \dots, n$$

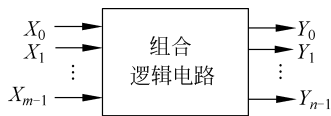


图 5.1 组合逻辑电路的一般结构

从电路结构上看,组合逻辑电路由逻辑门电路组

成,不包含任何记忆元件,且其输入与输出信号是单向传输的,不存在任何反馈回路。

描述组合逻辑电路逻辑功能的方法主要有逻辑表达式、真值表、卡诺图、逻辑图、波形图等。

5.2 组合逻辑电路的分析

分析组合逻辑电路是为了确定已知电路的逻辑功能,或检查电路设计是否合理。分析就是根据给定的逻辑图,找出输出信号与输入信号之间的关系,从而确定电路的逻辑功能。

5.2.1 组合逻辑电路的分析方法

尽管各种组合逻辑电路在功能上千差万别,但是它们的分析方法有共同之处。掌握了分析方法,就可以识别任何一个给定的组合逻辑电路的逻辑功能。



视频讲解

组合逻辑电路的分析过程如图 5.2 所示。

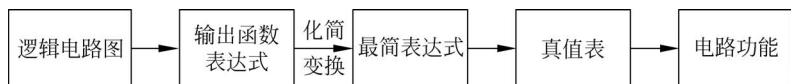


图 5.2 组合逻辑电路的分析过程

组合逻辑电路分析的一般步骤如下。

(1) 根据给定逻辑电路图写出输出与输入之间的逻辑函数表达式

为了确保写出的逻辑函数表达式正确无误,一般是在认清电路中所有逻辑器件和相互连线的基础上,从输入端开始往输出端逐级推导,直至得到所有与输入变量相关的输出函数表达式为止。用电路的输出函数表达式来表示电路的输出与输入之间的逻辑关系。

(2) 化简输出函数表达式

根据给定逻辑电路写出的输出函数表达式不一定是简表达式,为了简单、清晰地反映输入/输出之间的逻辑关系,应对逻辑函数表达式进行化简。化简的方法可以是公式化简法或卡诺图化简法。

(3) 列出输出函数真值表

有时,为了使电路的逻辑功能更加直观,还可以将输出函数表达式转换为真值表的形式,再对电路功能进行描述。因为有时候,通过逻辑函数表达式不能直观地看出电路的逻辑功能,这时就需要通过真值表来判断。

把全部输入组合代入输出函数最简表达式,计算得出输出结果,并以真值表的形式表示出来。真值表详尽地给出了输入/输出取值关系,它通过逻辑值直观地描述了电路的逻辑功能。

(4) 功能描述

根据真值表和化简后的输出函数表达式,概括出对电路逻辑功能的文字描述,并对原电路的设计方案进行评价,必要时提出改进意见和改进方案。

以上分析步骤是就一般情况而言的,实际应用中可根据问题的复杂程度和具体要求对上述步骤进行适当取舍。

5.2.2 组合逻辑电路的分析举例

【例 5-1】 试分析图 5.3 所示电路的逻辑功能。

解: (1) 根据逻辑电路图,写出输出函数表达式

根据电路中各逻辑门的功能,从输入端开始逐级写出函数表达式如下:

$$Y = \overline{AB} \cdot \overline{BC} \cdot \overline{AC}$$

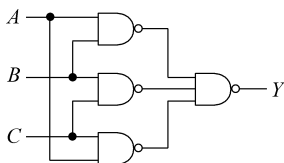


图 5.3 例 5-1 逻辑电路

(2) 化简输出函数表达式

用公式化简法对输出函数 Y 的表达式化简如下:

$$Y = AB + BC + AC$$

(3) 根据化简后的函数表达式列出真值表

该函数的真值表如表 5.1 所示。

表 5.1 例 5-1 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

(4) 功能描述

分析真值表后可以看出,该电路的三个输入变量中,只有两个及两个以上变量取值为 1 时,输出才为 1。因此,通常称该电路为“三人表决电路”。

【例 5-2】 试分析图 5.4 所示电路的逻辑功能。

解: (1) 根据逻辑电路图,写出输出函数表达式

根据电路中各逻辑门的功能,从输入端开始逐级写出函数表达式如下:

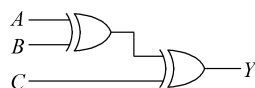


图 5.4 例 5-2 逻辑电路

$$Y = A \oplus B \oplus C$$

(2) 根据函数表达式列出真值表

该函数的真值表如表 5.2 所示。

表 5.2 例 5-2 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) 功能描述

分析真值表后可以看出,该电路的三个输入变量中,取值有奇数个 1 时, Y 为 1, 否则 Y 为 0。因此,该电路可用于检查 3 位二进制码的奇偶性,由于在输入的二进制码含有奇数个 1 时,输出有效信号,通常称该电路为“三变量奇校验电路”。

【例 5-3】 试分析图 5.5 所示电路的逻辑功能。

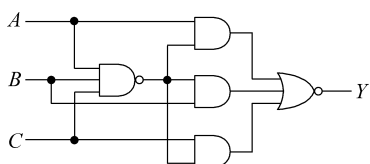


图 5.5 例 5-3 逻辑电路

解: (1) 根据逻辑电路图,写出输出函数表达式

根据电路中各逻辑门的功能,从输入端开始逐级写出函数表达式如下:

$$Y = A \cdot \overline{ABC} + B \cdot \overline{ABC} + C \cdot \overline{ABC}$$

(2) 化简输出函数表达式

用公式化简法对输出函数 Y 的表达式化简如下:

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

(3) 根据函数表达式列出真值表

该函数的真值表如表 5.3 所示。

表 5.3 例 5-3 真值表

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

(4) 功能描述

分析真值表后可以看出,该电路的三个输入变量取相同值时,Y 为 1,否则 Y 为 0。换句话说,当输入取值一致时输出为 1,不一致时输出为 0。可见,该电路具有检查输入信号是否一致的逻辑功能,一旦输出为 1,则表明输入一致。因此,通常称该电路为“一致性判定电路”。

由分析可知,该电路的设计方案并不是最简的。根据化简后的输出函数表达式,可画出实现给定功能的简化后的逻辑电路图,如图 5.6 所示。显然,它比原电路简单、清晰。

【例 5-4】 一个二输入端、二输出端的组合逻辑电路如图 5.7 所示,分析该电路的功能。

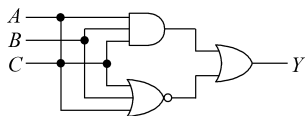


图 5.6 例 5-3 简化后的逻辑电路

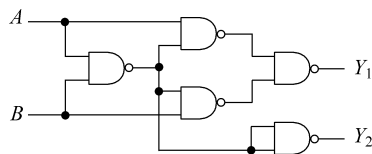


图 5.7 例 5-4 逻辑电路

解: (1) 根据逻辑电路图,写出输出函数表达式

$$Y_1 = \overline{\overline{AB} \cdot A \cdot \overline{AB} \cdot B}$$

$$Y_2 = \overline{\overline{AB}}$$

(2) 化简输出函数表达式

用公式化简法对输出函数化简如下:

$$\begin{aligned} Y_1 &= \overline{\overline{AB} \cdot A \cdot \overline{AB} \cdot B} \\ &= \overline{AB} \cdot A + \overline{AB} \cdot B \\ &= (\overline{A} + \overline{B}) \cdot A + (\overline{A} + \overline{B}) \cdot B \\ &= A\overline{B} + \overline{A}B \end{aligned}$$

$$\begin{aligned} Y_2 &= \overline{\overline{AB}} \\ &= AB \end{aligned}$$

(3) 根据化简后的函数表达式列出真值表

该函数的真值表如表 5.4 所示。

表 5.4 例 5-4 真值表

A	B	Y_1	Y_2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(4) 功能描述

分析真值表后可以看出,该电路的两个输入变量 A 、 B 都为 0 时,输出 Y_1 为 0, Y_2 也为 0;当 A 、 B 中有 1 个为 0 时,输出 Y_1 为 1, Y_2 为 0;当 A 、 B 都为 1 时,输出 Y_1 为 0, Y_2 为 1。这符合两个 1 位二进制数相加的原则,即 A 、 B 为两个加数, Y_1 是 A 、 B 相加的和(S), Y_2 是相加产生的进位(C_O)。该电路通常称作“半加器”,它能实现两个一位二进制数的加法运算。

【例 5-5】 分析图 5.8 所示组合逻辑电路。已知电路输入 $ABCD$ 为 8421 码,说明该电路的功能。

解: (1) 根据逻辑电路图,写出输出函数表达式

$$Y_1 = A \oplus (B \cdot (C + D))$$

$$Y_2 = B \oplus (C + D)$$

$$Y_3 = C \oplus \bar{D}$$

$$Y_4 = \bar{D}$$

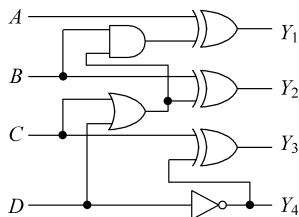


图 5.8 例 5-5 逻辑电路

(2) 根据函数表达式列出真值表

由于电路输入 $ABCD$ 为 8421 码,所以 $ABCD$ 只允许取值 0000~1001。根据所得输出函数表达式,可列出真值表如表 5.5 所示。

表 5.5 例 5-5 真值表

$ABCD$	$Y_1 Y_2 Y_3 Y_4$	$ABCD$	$Y_1 Y_2 Y_3 Y_4$
0000	0011	0101	1000
0001	0100	0110	1001
0010	0101	0111	1010
0011	0110	1000	1011
0100	0111	1001	1100

(3) 功能描述

分析真值表后可以看出,该电路的输出 $Y_1 Y_2 Y_3 Y_4$ 是一位十进制数的余 3 码,即该电路是一个将 8421 码转换成余 3 码的代码转换电路。

以上例子说明了组合逻辑电路分析的一般方法。从讨论过程可以看出,通过对电路进行分析,不仅可以找出电路输入、输出之间的关系,确定电路的逻辑功能,同时还能对某些设计不合理的电路进行改进和完善。

5.3 组合逻辑电路的设计

组合逻辑电路的设计就是根据实际问题所要求完成的逻辑功能,设计出在特定条件下



视频讲解

实现该功能的最简逻辑电路。所谓的“最简”，是希望所设计的电路使用的器件的数量最少、种类最少，器件之间的连线最少。器件的数量最少、种类最少，可以降低电路的复杂度和成本。器件之间的连线最少，可以提高电路的可靠性。

显然，电路设计是电路分析的逆过程。

5.3.1 组合逻辑电路的设计方法

组合逻辑电路的设计过程如图 5.9 所示。

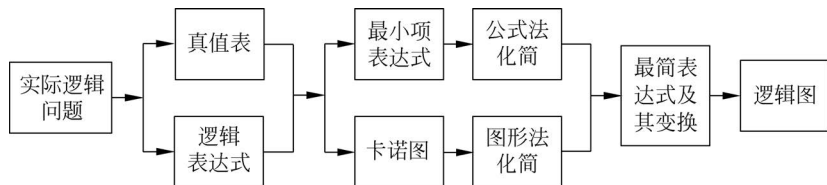


图 5.9 组合逻辑电路的设计过程

组合逻辑电路设计的一般步骤如下。

(1) 进行逻辑抽象

在很多情况下，实际问题所要求完成的逻辑功能都是用文字来描述的，逻辑抽象就是将文字描述的功能要求抽象为一种逻辑关系，即分析因果关系：确定输入变量与输出变量；对其进行状态赋值；将状态归类后，将因果关系用真值表或经过分析用逻辑表达式表示出来。这一步最为关键，因为如果分析错误，即逻辑抽象错误，所设计出的电路就不能达到设计的要求。

(2) 求出逻辑函数的最简表达式

基于小规模集成电路的组合逻辑电路设计是以门电路作为电路的基本单元，所以要以最简方案为目标，即应使使用的门电路的数目最少，而且门的输入端数目也最少。因此，需要求出逻辑函数的最简表达式。

由真值表可以直接写出逻辑函数的最小项表达式，然后用公式法化简为最简表达式；由真值表还可以画出逻辑函数的卡诺图，然后用图形法化简为最简表达式。

进行逻辑抽象得到的逻辑表达式如果不是最简，同样需要化简为最简表达式。

(3) 选择逻辑门类型并进行逻辑函数变换

在进行组合逻辑电路设计时，常常有指定用某种器件来实现的要求。在这种情况下，可以根据逻辑函数的运算规则，把化简后的逻辑表达式变换为满足选定的逻辑门对应的形式。例如，可以把最简与或式变换为与非式，满足全部用与非门来实现电路设计的要求。

(4) 画出逻辑电路图

根据变换后的逻辑表达式画出逻辑电路图。

以上设计步骤是就一般情况而言的，根据实际问题的难易程度和设计者的熟练程度，有时可跳过其中的某些步骤。在设计过程中可视具体情况灵活掌握。

5.3.2 组合逻辑电路的设计举例

【例 5-6】 设计一个监视交通信号灯状态的电路，以判断交通信号灯是否发生故障。

解：(1) 进行逻辑抽象

首先,确定输入变量与输出变量。假设用 R 、 Y 、 G 分别代表红、黄、绿交通信号灯的 3 个输入变量,用输出变量 E 表示信号灯故障的结果。则电路的功能为:如果 R 、 Y 、 G 信号灯出现故障,则输出 E 为 1。由此得到电路框图如图 5.10 所示。

接着,对输入变量与输出变量进行状态赋值。假设约定输入变量取值为 0 表示信号灯灭,取值为 1 表示信号灯亮;输出变量取值为 0 表示无故障,取值为 1 表示有故障。

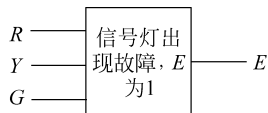


图 5.10 例 5-6 电路框图

然后,将状态进行归类。本例即确定什么是正常状态,什么是故障状态。经分析,正常的工作状态是 R 、 Y 、 G 信号灯中每次有且只有一个灯亮,函数 E 的值为 0。其他情况都为故障的状态,函数 E 的值为 1。根据状态分类,可列出该逻辑函数的真值表如表 5.6 所示。

表 5.6 例 5-6 真值表

R	Y	G	E	R	Y	G	E
0	0	0	1	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

(2) 求出逻辑函数的最简表达式

由真值表求逻辑函数的最简表达式有两种方法。

方法一是先基于真值表写出逻辑函数最小项表达式后,再利用公式法化简得到。

首先,对应每一个输出 E 为 1 时输入的取值,写一个与式。与式中,当输入变量取值为 1,写为原变量的形式;当输入变量取值为 0,写为非变量的形式。接着用或符号连接与式,得到逻辑函数标准与或表达式,即函数 E 的最小项表达式为

$$E = \bar{R}\bar{Y}\bar{G} + \bar{R}Y\bar{G} + R\bar{Y}\bar{G} + RY\bar{G} + RY\bar{G}$$

然后利用公式法化简,过程如下

$$\begin{aligned} E &= \bar{R}\bar{Y}\bar{G} + \bar{R}Y\bar{G} + R\bar{Y}\bar{G} + RY\bar{G} + RY\bar{G} \\ &= \bar{R}\bar{Y}\bar{G} + (\bar{R} + R)Y\bar{G} + R(\bar{Y} + Y)G + RY(\bar{G} + G) \\ &= \bar{R}\bar{Y}\bar{G} + Y\bar{G} + RG + RY \end{aligned}$$

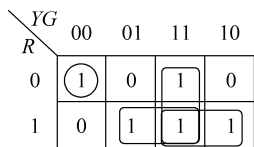


图 5.11 函数 E 的卡诺图

方法二是先基于真值表做出函数 E 的卡诺图,如图 5.11 所示。

用卡诺图化简后得到函数的最简与或表达式为

$$E = \bar{R}\bar{Y}\bar{G} + Y\bar{G} + RG + RY$$

(3) 选择逻辑门类型并进行逻辑函数变换

如果对设计的电路有选用特定的逻辑门类型的要求,则需要对逻辑函数进行形式变换。

按上述得到的逻辑函数 E 的最简式,如果选择非门和与或门,则无须再进行逻辑函数变换;而如果选择用集成电路来实现,则一般需要对表达式进行形式变换。

目前采用的小规模集成电路,大都是把几个相同类型的门电路封装在同一个集成芯片内。为提高集成芯片中逻辑门的利用率,可以减少集成芯片的使用数量。因此,在采用中小规模集成电路作为器件来设计逻辑电路时,尽量采用同一类型的逻辑门实现所要求的逻辑

功能。同时,考虑到提高电路工作速度的要求,通常选择输出端带取非功能的逻辑门(如与非门、或非门等),其速度要比输出端不带取非功能的逻辑门(如与门、或门等)快。

例如,在本例中,假定采用与非门实现给定功能的电路,则应将上述与或表达式变换成与非-与非形式。方法是利用还原律和反演规则(或摩根定律),即对与或表达式先进行两次取非运算,然后将最上面的非号保留,对下面的与非表达式应用反演规则,同时对取非运算应用重叠律后,变换为如下的与非-与非形式的表达式。

$$\begin{aligned} E &= \overline{RYG} + YG + RG + RY \\ &= \overline{\overline{\overline{RYG} + YG + RG + RY}} \\ &= \overline{\overline{RYG} \cdot \overline{YG} \cdot \overline{RG} \cdot \overline{RY}} \\ &= \overline{\overline{RR} \cdot \overline{YY} \cdot \overline{GG} \cdot \overline{YG} \cdot \overline{RG} \cdot \overline{RY}} \end{aligned}$$

如果需要其他形式,也都可以通过对表达式进行形式变换得到。

(4) 画出逻辑电路图

根据表达式就可以画出逻辑电路图。

例如,在本例中,由变换后的与非-与非表达式画出的逻辑电路图如图 5.12 所示。

可以看到,严格按照组合逻辑电路的设计过程,就可以把电路设计出来。而这里的每一步,在前面的布尔代数和门电路章节中都已经介绍过。

上述例子中输入变量较少,在进行逻辑抽象时,通过建立真值得到了相应的逻辑表达式。真值表的优点是规整、清晰,缺点是不方便,尤其当输入变量较多时十分麻烦。因此,当输入变量较多时,通常采用另一种方法,即通过分析问题,直接写出逻辑表达式。下面进行举例说明。

【例 5-7】 设计一个比较两个 4 位二进制数是否相等的数值比较器。

解: (1) 进行逻辑抽象

设两个 4 位二进制数分别为 $A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$, 比较结果用 Y 表示。当 $A = B$ 时, Y 为 1; 否则 Y 为 0。显然,这是一个有 8 个输入变量和 1 个输出变量的组合逻辑电路。电路框图如图 5.13 所示。

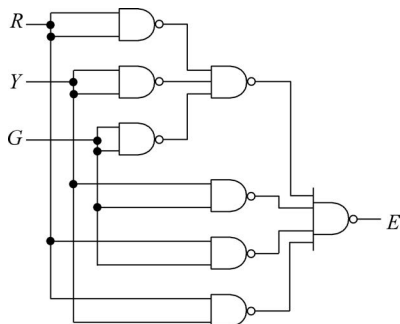


图 5.12 例 5-6 逻辑电路图



图 5.13 例 5-7 电路框图

由于二进制数 A 和 B 相等时,必须同时满足 $a_3 = b_3$ 、 $a_2 = b_2$ 、 $a_1 = b_1$ 、 $a_0 = b_0$, 而二进制数中, $a_i = b_i$ 只有 a_i 和 b_i 同时为 0 或者同时为 1 两种可能。而 a_i 和 b_i 同时为 0 或者同时为 1 时, $a_i \odot b_i = 1$ 。因此,该问题可用逻辑表达式描述如下:

$$Y = (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \odot b_1)(a_0 \odot b_0)$$

(2) 求出逻辑函数的最简表达式

上述逻辑函数表达式为同或-与表达式,不能化简。

(3) 选择逻辑门类型并进行逻辑函数变换

假定采用异或门和或非门实现给定功能,可将逻辑表达式作如下变换:

$$\begin{aligned} Y &= (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \odot b_1)(a_0 \odot b_0) \\ &= \overline{a_3 \oplus b_3} \overline{a_2 \oplus b_2} \overline{a_1 \oplus b_1} \overline{a_0 \oplus b_0} \\ &= \overline{\overline{a_3 \oplus b_3} \overline{a_2 \oplus b_2} \overline{a_1 \oplus b_1} \overline{a_0 \oplus b_0}} \\ &= \overline{(a_3 \oplus b_3) + (a_2 \oplus b_2) + (a_1 \oplus b_1) + (a_0 \oplus b_0)} \end{aligned}$$

(4) 画出逻辑电路图

根据变换后的逻辑表达式,可以画出逻辑电路图,如图 5.14 所示。

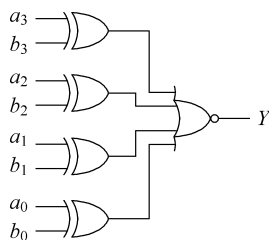


图 5.14 例 5-7 逻辑电路图

5.3.3 设计中实际问题的处理

上面介绍的是对组合逻辑电路进行设计的一般方法,然而,实际提出的设计 requirements 是多种多样的,可能存在某些需要考虑的实际问题。最常见的实际问题包括下面的两种情况。

1. 多输出变量的组合逻辑电路设计

只有一个输出端的逻辑电路的化简方法比较简单,只要将该逻辑函数化为最简,然后画出对应的逻辑图即可。而在设计组合逻辑电路的实际问题中,有同一组输入变量却有多个输出变量的应用更为普遍。多输出端逻辑电路的每个输出端都对应一个逻辑函数。进行多输出变量的组合逻辑电路设计时,不必一个逻辑函数对应一个逻辑电路,再对电路进行简单的拼接,因为这样不能保证逻辑电路整体最简。也就是说,各输出函数的最简表达式不要孤立地求出。因为各输出函数之间往往存在相互联系,具有某些共同的部分,因此,应该将它们当作一个整体,综合进行考虑,在逻辑函数化简时找出它们的共用项,从而在逻辑电路中共享逻辑门,以使电路整体结构最简。

【例 5-8】 设计一个设备工作异常检测电路,如图 5.15 所示,请用最少的逻辑门实现。其中,用红、黄指示灯 L_R 和 L_Y 表示三台设备 A、B、C 的工作情况:红灯亮表示有一台设备工作不正常;黄灯亮表示有两台设备工作不正常;红、黄灯全亮表示三台设备工作都不正常。

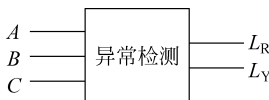


图 5.15 设备工作异常检测电路框图

解: 根据题意,列出异常检测电路的真值表如表 5.7 所示。

表 5.7 异常检测电路真值表

A	B	C	L_R	L_Y
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1

续表

A	B	C	L_R	L_Y
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

根据真值表写出输出函数的最小项表达式为

$$L_R = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$L_Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

对上述 L_R 表达式进行形式变换:

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

这样, L_R 函数只需两个异或门即可实现。

对 L_Y 表达式进行化简时, 如果不考虑共享逻辑门, 化简如下:

$$\begin{aligned} L_Y &= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC \\ &= (\bar{A}BC + ABC) + (A\bar{B}C + ABC) + (AB\bar{C} + ABC) \\ &= BC + AC + AB \end{aligned}$$

这样, L_Y 函数需要两个二输入与门和一个三输入或门才可实现。

如果选用与非门实现 L_R 函数, 变换如下:

$$L_Y = BC + AC + AB = \overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}} = \overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}}$$

这样, L_Y 函数也需要两个二输入与非门和一个三输入与非门才可实现。

为了尽可能减少电路中的逻辑门类型, 选用二输入与非门实现 L_Y 函数时, 继续对上述表达式作变换如下:

$$L_Y = \overline{\overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}}} = \overline{\overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}}}$$

这样, L_Y 函数需要六个二输入与非门才可实现。

可见, 如果不考虑共享逻辑门时, 本电路最少选用异或门和二输入与非门两种逻辑门类型, 最少需要两个异或门和六个二输入与非门加以实现。

考虑到 L_Y 函数可以与 L_R 函数共享异或门, 这样, 选用异或门和二输入与非门实现 L_R 函数, 可以对其作如下变换:

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

可见, 据此表达式实现 L_Y 函数时, 除与 L_R 函数共用一个异或门外, 额外只需三个二输入与非门即可实现。

根据化简变换后的输出函数表达式可画出相应的逻辑电路, 如图 5.16 所示。

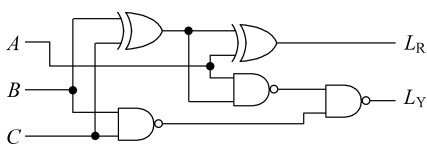


图 5.16 设备工作异常检测电路逻辑图

显然,通过找出两个函数的公共项,使它们共用同一个逻辑门,可使电路从整体上得到进一步简化。

2. 包含无关项的组合逻辑电路设计

前面给出的电路设计实例中,对于电路输入变量的任何一种取值组合,都有确定的输出函数值与之对应。换句话说,对于一个具有 n 个输入变量的组合逻辑电路,输出函数与 2^n 种输入取值组合均相关。但在某些实际问题中,常常由于输入变量之间存在的相互制约或问题的某种特殊限定等,使得输入变量的某些取值组合根本不会出现,或者虽然可能出现,但在这些输入取值组合下函数的值可以是任意的,对它为 1 还是为 0 并不关心。这类问题即为包含无关项(或任意项、约束项)的电路设计问题。

【例 5-9】 设计一个组合逻辑电路,将输入的 8421 码转换为 2421 码,试用与非门实现该电路。

解: 代码转换电路设计示意图如图 5.17 所示,它有四个输入端 A_3 、 A_2 、 A_1 、 A_0 和四个输出端 B_3 、 B_2 、 B_1 、 B_0 。



图 5.17 代码转换电路框图

代码转换真值表如表 5.8 所示,其中 1010~1111 不会在输入端出现,故作为约束项处理,并用“x”表示。

表 5.8 代码转换真值表

$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$
0000	0000	1000	1110
0001	0001	1001	1111
0010	0010	1010	xxxx
0011	0011	1011	xxxx
0100	0100	1100	xxxx
0101	1011	1101	xxxx
0110	1100	1110	xxxx
0111	1101	1111	xxxx

由真值表分别作出函数 B_3 、 B_2 、 B_1 、 B_0 的卡诺图,如图 5.18(a)、(b)、(c)、(d)所示。

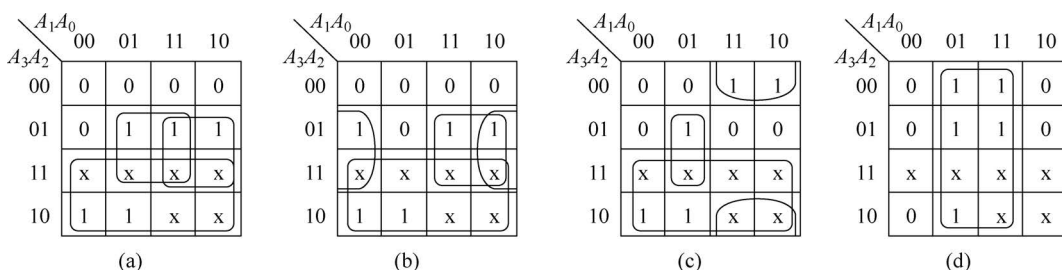


图 5.18 函数 B_3 、 B_2 、 B_1 、 B_0 的卡诺图

	A_1A_0		00	01	11	10
A_3A_2	00	0	0	0	0	0
	01	0	1	1	1	1
	11	x	x	x	x	x
	10	1	1	x	x	x

图 5.19 修改后的函数 B_3 的卡诺图

通过观察 B_3 、 B_2 、 B_1 、 B_0 的卡诺图中作圈的情况,为了使不同函数卡诺图中尽可能多地出现相同的圈,以使最后的逻辑电路中能够共用电路。所以修改函数 B_3 的卡诺图如图 5.19 所示。

用卡诺图化简后得到函数的最简与或表达式,并转换为与非-与非式:

$$B_3 = A_3 + A_2A_1 + A_2\overline{A_1}A_0 = \overline{\overline{A_3} \cdot \overline{A_2A_1} \cdot \overline{A_2\overline{A_1}A_0}}$$

$$B_2 = A_3 + A_2A_1 + A_2\overline{A_0} = \overline{\overline{A_3} \cdot \overline{A_2A_1} \cdot \overline{A_2\overline{A_0}}}$$

$$B_1 = A_3 + \overline{A_2}A_1 + A_2\overline{A_1}A_0 = \overline{\overline{A_3} \cdot \overline{\overline{A_2}A_1} \cdot \overline{A_2\overline{A_1}A_0}}$$

$$B_0 = A_0$$

根据上述表达式,画出电路的逻辑图,如图 5.20 所示。

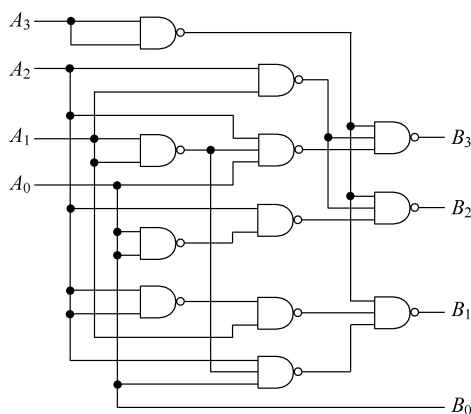


图 5.20 代码转换电路逻辑图

设计包含无关项的组合逻辑电路时,恰当地利用无关项进行函数化简,通常可使设计出来的电路更简单。

3. 带有使能端的组合逻辑电路设计

使能端(Enable, En)通常指的是控制信号输入端,又称允许输入端,也叫片选端(Chip Select, CS)。它是电路的一个输入信号,只有该信号有效,电路才能工作。通常情况下,信号名称采用原变量形式表示高电平有效,采用非变量形式表示低电平有效。

在组合逻辑电路的设计中,一般可以采用在电路的最后一级逻辑门的输入端加入使能控制信号实现使能端功能。下面举例说明。

【例 5-10】 设计一个密码锁,如图 5.21 所示。其中, A 、 B 、 C 、 D 是四位二进制代码输入端, \bar{E} 为密码输入使能端(当 $\bar{E}=0$ 时,表示确认密码)。每把锁有四位密码(设该锁的密码为 1011),若输入代码符合该锁密码且 $\bar{E}=0$ 使能时,送出一个开锁信号($Y_{\text{open}}=1$),用于开锁指示的指示亮;若输入代码不符合该锁密码且 $\bar{E}=0$ 使能时,送出报

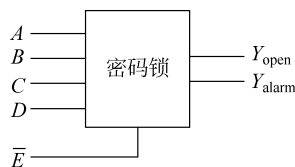


图 5.21 密码锁电路框图

警信号($Y_{\text{alarm}} = 1$),用于报警指示的灯亮;若 $\bar{E} = 1$,密码锁不工作,不送出任何信号。

解: 密码锁简化的真值表(也称功能表)如表 5.9 所示,其中,当 $\bar{E} = 1$ 时,输入端 A、B、C、D 无论状态是什么(用“x”表示),密码锁不工作,输出端信号状态为 0。

表 5.9 密码锁功能表

\bar{E}	ABCD	Y_{open}	Y_{alarm}	\bar{E}	ABCD	Y_{open}	Y_{alarm}
0	0000	0	1	0	1001	0	1
0	0001	0	1	0	1010	0	1
0	0010	0	1	0	1011	0	1
0	0011	0	1	0	1100	0	1
0	0100	0	1	0	1101	1	0
0	0101	0	1	0	1110	0	1
0	0110	0	1	0	1111	0	1
0	0111	0	1	1	xxxx	0	0
0	1000	0	1				

不考虑 \bar{E} 时,根据功能表写出输出函数表达式为

$$Y_{\text{open}} = ABC\bar{D}$$

$$Y_{\text{alarm}} = \overline{Y_{\text{open}}}$$

当 $\bar{E} = 0$ 时,上述表达式有效,输出为 1;当 $\bar{E} = 1$ 时,上述表达式无效,输出为 0。因此,考虑 \bar{E} 时,得到输出函数表达式为

$$Y_{\text{open}} = \bar{E}ABC\bar{D}$$

$$Y_{\text{alarm}} = \overline{\bar{E}ABC\bar{D}}$$

为了说明使能信号的引入,此处不对输出函数进行形式变换,根据上述表达式,直接画出电路的逻辑图,如图 5.22 所示。

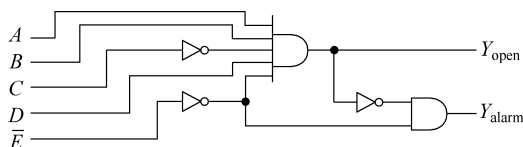


图 5.22 密码锁电路逻辑图

在中规模集成电路中,使能端被普遍应用,它既可以是输入,也可以是输出,一般用来扩展中规模集成电路的功能。当进行输入使能时,使能信号也可以由三态门的使能信号来提供。

4. 带有输入缓冲的组合逻辑电路设计

组合逻辑电路的功能中有些是专用的,有些是通用的。一些通用的组合逻辑电路一般是做成一个电路芯片,以整体的形式用于数字系统的设计中。从上述举例的设计电路中可以看到,电路的一个输入信号大多需要同时驱动片内很多逻辑门的输入。而如果该电路的输入信号是由前序电路的逻辑门输出信号提供时,这就导致其负载数目不是一个,而是多个。无形中降低了输入信号的驱动能力,增加了前面电路的负担。一般情况下,希望将一个电路芯片看作一个负载,这样,就需要在组合逻辑电路设计时,增加输入缓冲电路部分。输

入缓冲一般可以用两级反相器来实现。下面举例说明。



图 5.23 2-4 译码器电路框图

【例 5-11】 设计一个 2-4 译码器,如图 5.23 所示。其中, D_1 、 D_0 是 2 位二进制代码代表的信号输入端, $\overline{Y_0}$ 、 $\overline{Y_1}$ 、 $\overline{Y_2}$ 、 $\overline{Y_3}$ 是 2 位二进制代码代表的 4 个信号输出端,低电平有效。当 $D_1D_0=00$ 时, $\overline{Y_0}$ 有效;当 $D_1D_0=01$ 时, $\overline{Y_1}$ 有效;当 $D_1D_0=10$ 时, $\overline{Y_2}$ 有效;当 $D_1D_0=11$ 时, $\overline{Y_3}$ 有效。

解: 根据题意,列出 2-4 译码器的真值表如表 5.10 所示。

表 5.10 2-4 译码器真值表

D_1	D_0	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

由真值表得到输出函数表达式如下:

$$\overline{Y_0} = \overline{D_1 D_0}$$

$$\overline{Y_1} = \overline{D_1} D_0$$

$$\overline{Y_2} = D_1 \overline{D_0}$$

$$\overline{Y_3} = D_1 D_0$$

根据上述表达式,画出电路的逻辑图,如图 5.24 所示。

由图 5.24 可见,输入信号 D_1 和 D_0 的负载分别有三个。

加入输入缓冲电路后的逻辑图如图 5.25 所示。

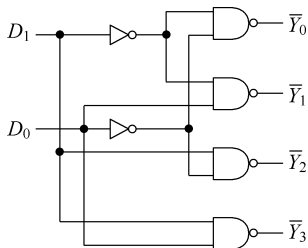


图 5.24 2-4 译码器电路逻辑图

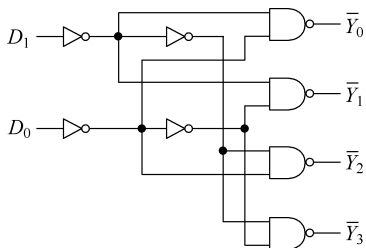


图 5.25 带有输入缓冲的 2-4 译码器电路逻辑图

由图 5.25 可见,加入输入缓冲电路后,输入信号 D_1 和 D_0 的负载分别只有一个。



视频讲解

5.4 组合逻辑电路中的竞争-冒险

前面讨论组合逻辑电路时,只研究了输入信号和输出信号稳定状态之间的关系,而没有考虑信号在传输中的延迟问题。实际上,当输入信号通过任何一个逻辑部件时,都会有延迟,这就使得当电路所有输入信号达到稳定状态时,输出信号并不是立即达到稳定状态。同时,延迟也使输出端可能出现不是理想条件下的结果,甚至会产生一些错误。

对于组合逻辑电路来说,当两个或多个输入信号同时向相反方向发生变化时,由于逻辑部件的传输延迟不同而造成信号传输过程中的竞争,可能会在输出端产生短暂的尖峰错误信号,这种尖峰就是组合逻辑电路中的竞争-冒险。虽然竞争-冒险是暂时的,信号稳定后会消失,但这种尖峰信号对一些边沿敏感的器件或电路(如触发器、计数器等)会引起误操作,使电路工作的可靠性下降。

5.4.1 竞争-冒险的产生

当一个逻辑门的两个输入端的信号同时向相反方向变化时,从变化开始到稳定状态所需的时间不同,称为竞争。两个输入端可以是不同变量所产生的信号,但取值的变化方向是相反的。也可以是在一定条件下,门电路输出端的逻辑表达式化简成两个互补信号相与或者相或。逻辑门因输入端的竞争而导致输出产生不应有的尖峰干扰脉冲(又称过渡干扰脉冲)的现象,称为冒险。冒险是一种瞬态现象,它暂时性地破坏正常逻辑关系,一旦瞬态过程结束,即可恢复正常逻辑关系。下面举两个例子说明这一现象。

例如,图 5.26 所示的电路,其输出表达式为

$$Y = \overline{AB} = A + B$$

在稳态情况下,当输入信号 A 和 B 中只要有一个为 1 时,即 \bar{A} 和 \bar{B} 中只要有一个为 0 时,与非门 G_3 的输出始终为 1。如果信号 A 、 B 分别经非门 G_1 、 G_2 送入与非门 G_3 时的变化同时发生,则能满足要求。若信号 A 和 B 同时向相反方向变化,即 A 从 1 变为 0, B 从 0 变为 1,由于非门 G_1 、 G_2 电路的延迟差异或信号传输延迟差异,致使 \bar{B} 从 1 变为 0 的时刻,滞后 \bar{A} 从 0 变为 1 的时刻。因此,在很短的时间间隔内,与非门 G_3 的两个输入均为 1,其输出就会出现一个低电平窄脉冲(干扰脉冲),如图 5.27 所示。

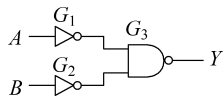


图 5.26 竞争-冒险电路举例一

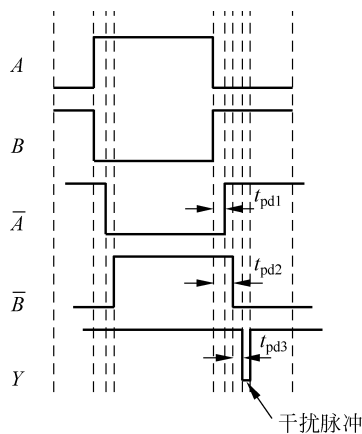


图 5.27 竞争-冒险电路举例一输出波形图

图 5.27 中, t_{pd1} 、 t_{pd2} 、 t_{pd3} 分别表示逻辑门 G_1 、 G_2 、 G_3 的延迟时间(图中没有考虑其他传输延迟因素)。

再如图 5.28 所示的电路,其输出表达式为

$$Y = \overline{\overline{AB}AC} = AB + \bar{A}C$$

假设输入信号 $B=C=1$, 将 $B、C$ 的值代入上述函数表达式, 得

$$Y = A + \bar{A}$$

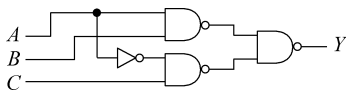


图 5.28 竞争-冒险电路举例二

此时, 同样出现最后一级逻辑门的两个输入信号同时向相反方向变化的情况。即在输出应该为 1 的情况下, 可能会产生一个瞬态为 0 的信号。因此, 电路产生了竞争-冒险。

5.4.2 竞争-冒险的消除

对于一个具体的数字逻辑电路, 由于各种因素的随机性, 判断它是否存在竞争-冒险是十分困难的。然而, 由于竞争-冒险对一些边沿敏感的器件或电路会引起误操作, 降低电路工作的可靠性。因此, 必须采取相应的措施消除组合逻辑电路中的竞争-冒险。常用于消除竞争-冒险的方法有增加冗余项法、增加选通脉冲法等。

1. 增加冗余项法

增加冗余项的方法是, 通过在函数表达式中“或”上多余的与项或者“与”上多余的或项, 使原函数不可能在某种条件下转化成 $X + \bar{X}$ 或者 $X \cdot \bar{X}$ 的形式, 从而消除可能产生的竞争-冒险。

【例 5-12】 用增加冗余项的方法消除图 5.28 所示电路中可能产生的竞争-冒险。

解: 根据函数表达式

$$Y = AB + \bar{A}C$$

当 $B=C=1$ 时, 输入信号 A 的变化使电路输出可能产生竞争-冒险。解决这一问题的思路是如何保证当 $B=C=1$ 时, 使输出保持为 1。显然, 若函数表达式中包含与项 BC , 则可达到这一目的。为此, 根据布尔代数中的包含律, 在上述表达式中增加与项 BC , 可以将表达式形式变换为

$$Y = AB + \bar{A}C = AB + \bar{A}C + BC$$

可见, 与项 BC 是上述函数的一个冗余项。增加冗余项后的逻辑电路如图 5.29 所示。该电路不再产生竞争-冒险。

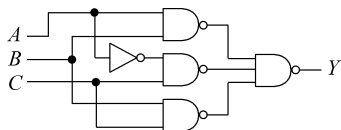


图 5.29 增加冗余项消除竞争-冒险的电路逻辑图

2. 增加选通脉冲法

用增加冗余项的方法可以消除一些竞争-冒险现象。但是, 这种方法的适用范围是有限的。用增加选通脉冲的方法消除竞争-冒险则具有通用性。

由于组合逻辑电路中的竞争-冒险总是发生在输入信号发生变化的瞬间, 且竞争-冒险总是以窄脉冲的形式输出, 因此, 为了避开竞争-冒险, 可以增加一个选通脉冲来对输出可能产生尖峰干扰脉冲的门电路加以控制。在选通脉冲到来之前, 选通控制线的电平使输出门被关闭, 使竞争-冒险脉冲无法输出。只有在输入信号转换完成并稳定后, 才引入选通脉冲使输出门被开启, 使电路送出稳定输出信号。

脉冲有正和负之分。如果脉冲跃变后的值比初始值高, 为正脉冲, 如图 5.30(a) 所示;

反之,则为负脉冲,如图 5.30(b)所示。

用增加选通脉冲的方法消除图 5.26 所示电路中可能产生的竞争-冒险的电路,如图 5.31 所示。

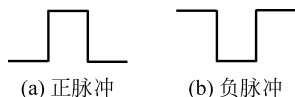


图 5.30 正脉冲与负脉冲

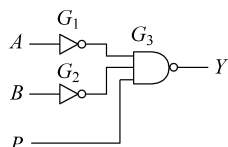


图 5.31 增加选通脉冲消除竞争-冒险的电路逻辑图

如图 5.31 所示,如果输出门是与门或与非门,则选通信号是一个正脉冲,因为正脉冲(高电平)到来时,它们的输出才是有效信号;没有选通脉冲时(低电平),电路没有输出。选通脉冲是等到输入信号稳定后才出现的,这样可以避免竞争-冒险,如图 5.32 所示。

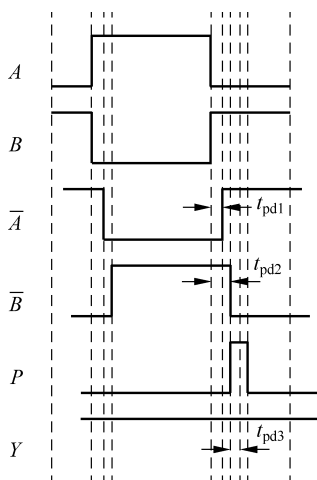


图 5.32 竞争-冒险电路举例一输出波形图

如果输出门是或门或者或非门,则选通信号是一个负脉冲,因为负脉冲(低电平)到来时,或门或者或非门的输出才是有效信号;没有选通脉冲时(高电平),电路没有输出。

习题

5.1 组合逻辑电路图如图 5.33 所示,写出输出函数的与或表达式,并列其真值表。

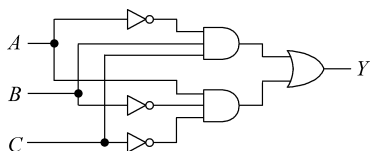


图 5.33 习题 5.1 逻辑电路图

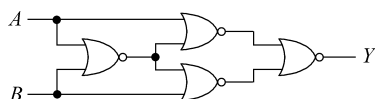


图 5.34 习题 5.2 逻辑电路图

5.3 组合逻辑电路图如图 5.35 所示,写出输出函数表达式,列出真值表,并说明电路的逻辑功能。

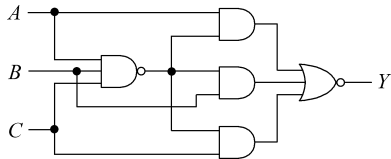


图 5.35 习题 5.3 逻辑电路图

5.4 组合逻辑电路图如图 5.36 所示,写出输出函数表达式,列出真值表,并说明电路的逻辑功能。

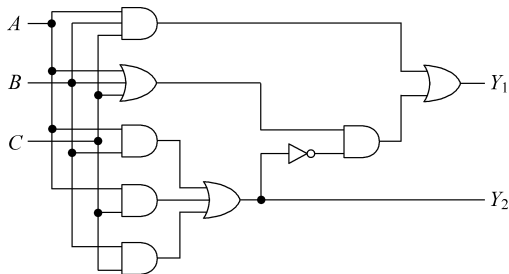


图 5.36 习题 5.4 逻辑电路图

5.5 组合逻辑电路图如图 5.37 所示,写出输出函数表达式,列出真值表,并说明电路的逻辑功能。

5.6 组合逻辑电路图如图 5.38 所示,写出输出函数表达式,列出真值表,并说明电路的逻辑功能。

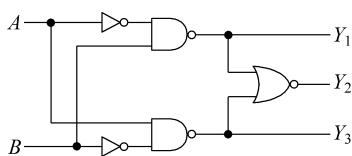


图 5.37 习题 5.5 逻辑电路图

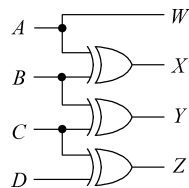


图 5.38 习题 5.6 逻辑电路图

5.7 用与非门设计一个 4 变量的多数表决器电路。当输入变量中有 3 个或 3 个以上为 1 时,则输出为 1,否则输出为 0。

5.8 设计一个数值比较电路,对输入的两个 2 位二进制数 $A = A_1A_0$, $B = B_1B_0$ 进行比较。当 $A > B$ 时,输出 $Y = 1$,否则输出 $Y = 0$ 。

5.9 用与非门设计一个代码转换电路,将 1 位十进制数的余 3 码转换成 2421 码。

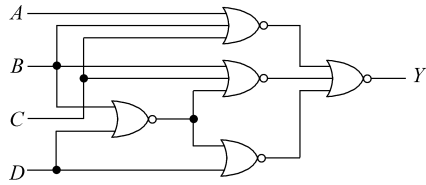
5.10 用与非门设计一个四舍五入电路。该电路输入为 1 位十进制数的 8421 码,当其值大于或等于 5 时,输出 $Y = 1$,否则输出 $Y = 0$ 。

5.11 设计一个偶检测电路,检测 4 位二进制码中 1 的个数是否为偶数。若为偶数个 1,则输出为 1,否则输出为 0。

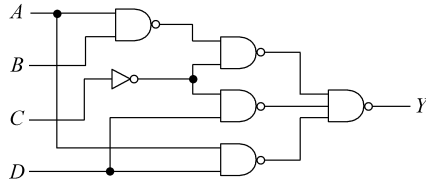
5.12 有一火灾报警系统,设有烟感、温感和紫外光感 3 种类型的火灾探测器。为了防止误报警,只有当其中两种或两种以上类型的探测器发出火灾探测信号时,报警系统才产生

报警控制信号。试用与非门设计一个产生报警控制信号的电路。

5.13 判断图 5.39 所示各逻辑电路是否存在竞争-冒险,说明原因。



(a)



(b)

图 5.39 习题 5.13 逻辑电路图

5.14 下列函数描述的电路是否可能产生竞争-冒险?如果能产生,说明是在什么情况下产生,并用增加冗余项的方法消除。

(1) $Y_1 = AB + AC\bar{C} + \bar{C}D$

(2) $Y_2 = AB + \bar{A}CD + BC$

(3) $Y_3 = (A + \bar{B})(\bar{A} + \bar{C})$

第6章

常用的集成组合逻辑电路



知识导学



习题答案

在数字系统设计中,有些逻辑电路是经常或大量使用的,为了使用方便,一般把这些逻辑电路制成中、小规模集成电路产品。在组合逻辑电路中,常用的中规模集成组合逻辑电路产品有加法器、数值比较器、编码器、译码器、多路选择器、奇偶校验器等。这些中规模集成组合逻辑器件的功能虽然比小规模集成电路强,但也不像大规模集成电路那样功能专一化,这些器件产品的品种虽然不少,但也不可能完全符合使用者的要求,这就需要将多片级联以扩展其功能。

本章分别介绍这些组合逻辑器件的电路结构、工作原理和级联扩展或使用方法。



视频讲解

6.1 加法器

数字系统的基本任务之一是进行算术运算。在数字系统中,加、减、乘、除运算均可利用加法器来实现,所以加法器是数字系统中最基本的运算单元电路。

6.1.1 半加器

两个1位二进制数相加而不考虑来自低位进位的加法运算称为半加,实现半加运算的电路称为半加器(Half Adder, HA)。半加器的逻辑符号如图6.1所示。

图6.1中, A 、 B 是两个1位二进制加数的输入端, S 是两个数相加后的和数输出端, C_o 是向高位的进位输出端,真值表如表6.1所示。

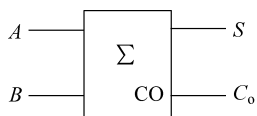


图6.1 半加器的逻辑符号

表6.1 半加器的真值表

A	B	S	C_o	A	B	S	C_o
0	0	0	0	1	0	1	0
0	1	1	0	1	1	0	1

按照分析方法,由表6.1可以写出半加器输出端的逻辑函数表达式:

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$C_o = AB$$

根据半加器的逻辑函数表达式,可画出其逻辑电路如图6.2所示。