

第3章 组合逻辑电路

【内容概要】

数字系统的逻辑电路可分为两类：一类是组合逻辑电路，另一类是时序逻辑电路。本章介绍的组合逻辑电路，是指由各种门电路组合而成且无反馈的逻辑电路。

本章首先介绍组合逻辑电路中的基础门电路——CMOS 逻辑门电路和 TTL 逻辑门电路的内部结构、工作原理及其外部特性；其次，介绍组合逻辑电路的模型与特点，结合具体实例阐述组合逻辑电路分析的一般流程与方法，列举对常用组合逻辑电路进行分析的完整过程；再次，结合具体实例阐述组合逻辑电路设计的一般流程与方法、优化设计的常用技巧；最后，从电路稳定性的角度介绍组合逻辑中可能存在的险象，从电路应用的角度介绍加法器、编码器、译码器、多路选择器与分配器的设计及中规模集成电路产品的应用。

【学习目标】

通过学习本章内容，学生能了解 CMOS 逻辑门电路和 TTL 逻辑门电路的结构、工作原理和外部特性，利用基本逻辑运算和简单逻辑门电路，构建复合逻辑门电路；熟悉组合逻辑电路的定义、特点及功能，利用组合逻辑电路分析方法，对已知的逻辑电路图，推导出描述其逻辑特性的逻辑表达式并评述其逻辑功能，利用组合逻辑电路设计方法，对给定的逻辑命题，设计出能实现其功能的逻辑电路；能辨别出组合电路中存在的竞争—冒险现象；能分析常用组合逻辑电路（如加法器、编码器、译码器、多路选择器等）的组成原理，利用相应的中规模集成电路产品进行电路设计以实现更复杂的逻辑功能。

【思政元素】

在组合逻辑电路中，每个基本逻辑门电路仅能实现一个简单的逻辑功能，但根据逻辑代数的基本定理和运算法则将其组合连接，便可实现任意复杂的组合逻辑功能。本章教学中首先引入基本逻辑门和复合逻辑门的描述，然后在此基础上引入组合逻辑电路的概念，由此引导学生正确看待个人与集体的辩证关系，个人是组成集体的重要单元，而集体是个人力量的凝聚，具有更强大的功能。提醒读者应以集体目标为导向，在现有社会的运行机制下，充分发挥个人在集体中的作用，在提升个人综合实力的同时，增强集体凝聚力、竞争力和综合性创新能力，以更好地实现个人价值。

3.1 逻辑门电路简介

第2章讨论了逻辑代数的基本定律、基本公式以及逻辑函数的表示和化简方法，它们在进行逻辑电路分析和设计的理论基础。任何复杂的逻辑运算都是由“与”“或”“非”3种基本

逻辑运算组成的。因此,首先应该研究实现这3种基本逻辑运算的电路,在此基础上进一步构成各种复杂的逻辑运算电路。

用于实现基本逻辑运算和复合逻辑运算的单元逻辑电路通称为逻辑门电路,简称为“门”。为便于使用,逻辑门电路通常制作成集成电路。按其制作的半导体材料可分为TTL(transistor-transistor-logic)门电路和CMOS(complementary metal-oxide-semiconductor)门电路。TTL门电路的工作速度快、负载能力强,但功耗较大、集成度低;CMOS门电路的结构简单、集成度高、功耗低,但速度较慢、负载能力较弱。随着技术的进步,CMOS门电路的性能得到极大的提高,目前大规模、超大规模集成电路一般采用CMOS工艺制造。本节首先简要介绍CMOS逻辑门电路和TTL逻辑门电路的基本结构与工作原理;随后以CMOS门电路为例,介绍简单逻辑门电路、常用复合逻辑门电路以及逻辑门电路的主要外特性参数;最后阐述逻辑门电路中的正、负逻辑。



视频讲解

3.1.1 逻辑门电路的基本结构与工作原理

在二值数字逻辑中,逻辑变量的取值为0或1,对应数字电路中电子器件的“闭合”与“断开”两种状态。图3.1所示为开关电路示意图。当开关S断开时,输出电压 $v_O = V_{CC}$,输出高电平,表示逻辑1,如图3.1(a)所示;反之,当开关S闭合时,输出电压 $v_O = 0$,输出低电平,表示逻辑0,如图3.1(b)所示。

早期的开关由继电器构成,后来使用BJT或MOS管作为开关。BJT或MOS管相当于一个受控开关,当其工作在截止状态时,相当于开关断开,输出高电平;当其工作在饱和状态时,相当于开关闭合,输出低电平。

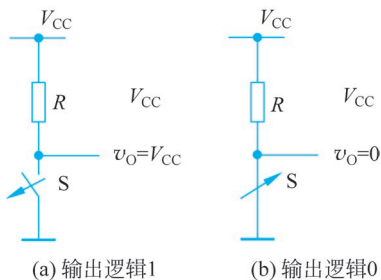


图 3.1 开关电路示意图

1. 基本 CMOS 逻辑门电路

按照导电载流子的不同,MOS管分为N沟道MOS(NMOS)管和P沟道MOS(PMOS)管。按照导电沟道形成机理的不同,MOS管又分为增强型和耗尽型。CMOS逻辑门电路以MOS管作为开关器件,由两种在导电性上互补的MOS管组成。本节以N沟道增强型MOS管为例,介绍其基本结构、工作原理与特性。

N沟道增强型MOS管的结构示意图及符号如图3.2所示。它是通过在P型硅衬底上用扩散法制作两个高掺杂浓度的N区,然后在P型硅表面生长一层很薄的二氧化硅绝缘层,

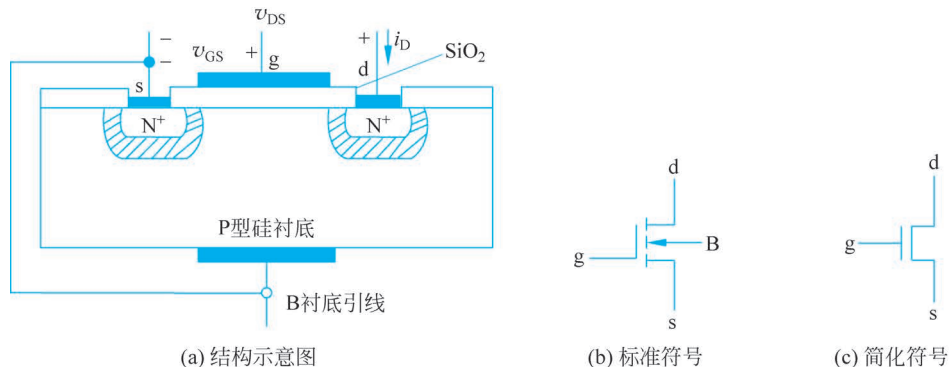


图 3.2 N沟道增强型MOS管

并在二氧化硅表面及两个 N 区各安置一个电极,形成栅极 g、源极 s 和漏极 d。通常将衬底与源极相连或接地电位,以防止有电流从衬底流入源极和导电沟道。

当栅极 g 和源极 s 之间所加电压 $v_{GS}=0$,则源极区、衬底和漏极区形成两个 PN 结背靠背串联,漏极 d 和源极 s 之间不导通, $i_D=0$ 。当栅极 g 和源极 s 之间加正向电压 $v_{GS} \geq V_T$ (V_T 为开启电压)时,栅极 g 和衬底之间形成足够强的电场,吸引衬底中的少数载流子(电子),使其聚集在栅极 g 下的衬底表面,形成 N 型反形层,该反形层就构成了漏极 d 和源极 s 之间的导电沟道。若此时漏极 d 和源极 s 之间加电压 v_{DS} ,将有漏极电流 i_D 产生,如图 3.3 所示。这种在 $v_{GS}=0$ 时不存在导电沟道, v_{GS} 必须增加到一定值时才形成导电沟道的场效应管,称为 N 沟道增强型 MOS 管。

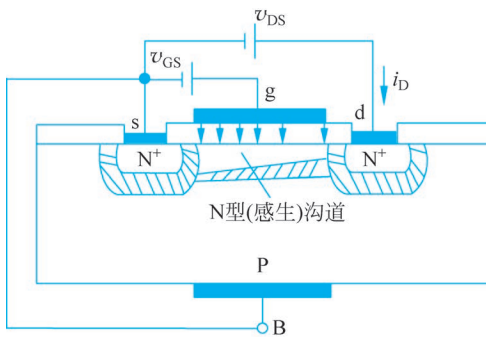


图 3.3 外加电压使 MOS 管形成导电沟道

MOS 管可视为二端口网络,输入端口电压为 v_{GS} ,输出端口电压为 v_{DS} 。当端口电压不同时,回路电流也将发生变化,相应的变化关系可由包含输出特性和转移特性的 $I-V$ 特性曲线反映,分别如图 3.4(a)、图 3.4(b)所示。

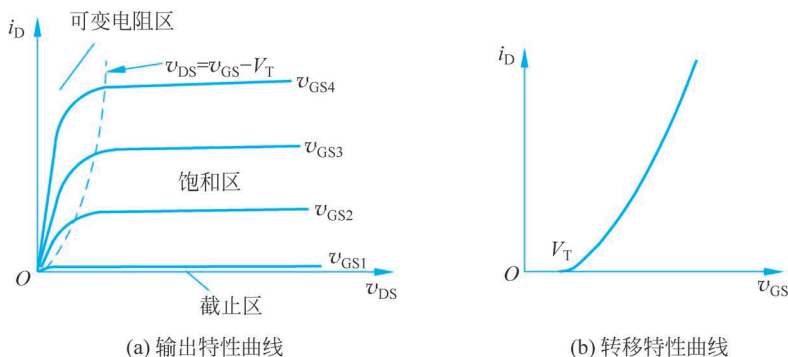


图 3.4 N 沟道增强型 MOS 管的 $I-V$ 特性曲线

输出特性曲线是指栅源电压 v_{GS} 一定的情况下,漏极电流 i_D 与漏源电压 v_{DS} 之间的关系。输出特性曲线分为 3 个工作区:截止区、饱和区和可变电阻区。

当 $v_{GS} < V_T$ 时,导电沟道尚未形成, $i_D=0$,漏—源间电阻很大,相当于断开,MOS 管处于截止工作状态,特性曲线的该区域称为截止区。

当 $v_{GS} \geq V_T$ 时,产生导电沟道,外加 v_{DS} 较小时, i_D 随 v_{DS} 呈线性增长。此时 MOS 管可视为一个受 v_{GS} 控制的可变电阻 R_{DS} , v_{GS} 越大,输出特性越倾斜,等效电阻越小。因此,该区域称为可变电阻区。

当 v_{DS} 继续增加,达到 $v_{DS} = v_{GS} - V_T$ 时,沟道在靠近漏极 d 处开始消失,称为预夹断。随着 v_{DS} 继续增加, i_D 几乎不再增加,此时的区域称为饱和区。因此,当 $v_{DS} < v_{GS} - V_T$ 时,N 沟道 MOS 管工作于可变电阻区;当 $v_{DS} \geq v_{GS} - V_T$ 时,MOS 管工作于饱和区。

转移特性曲线是指在漏源电压 v_{DS} 一定的条件下,栅源电压 v_{GS} 对漏极电流 i_D 的控制作用。当 MOS 管工作在饱和区时,由于 v_{DS} 对 i_D 的影响很小,所以不同 v_{DS} 所对应的转移特性曲线基本重合,可以用一条曲线来表示。这条曲线与横坐标的交点即为开启电压 V_T 。

与 N 沟道增强型 MOS 相反, P 沟道增强型 MOS 管是在 N 型衬底上制作两个高浓度的 P 区, 导电沟道为 P 型, 载流子为空穴, 其符号如图 3.5 所示。通常将衬底与源极相连, 或接电源。

为吸引空穴形成导电沟道, 栅极 g 接电源负极, 与衬底相连的源极 s 接电压的正极, 即 v_{GS} 为负值, 因此开启电压 V_T 也为负值。而 i_D 的实际方向为流出漏极 s , 与通常的假定方向相反。

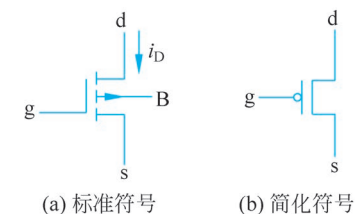


图 3.5 P 沟道增强型 MOS 管符号

N 沟道耗尽型 MOS 管的结构与增强型基本相同, 最大的区别是在生产过程中, 在二氧化硅绝缘层中掺入了大量正离子, 其符号如图 3.6 所示。当 $v_{GS}=0$ 时, 由于正离子的作用, 将电子吸引到栅极 g 下面的衬底表面形成 N 型沟道。当 $v_{GS}>0$ 时, 沟道变宽。在 v_{DS} 作用下, i_D 的数值更大。当 $v_{GS}<0$ 时, 沟道变窄, i_D 减小; 当 v_{GS} 达到某一负值 V_P 时, 沟道完全被夹断, 即使增大 v_{DS} , 也不会有漏极电流 i_D , MOS 管截止, V_P 称为夹断电压。

P 沟道耗尽型 MOS 管的结构与增强型基本相同, 但在二氧化硅绝缘层中掺入了大量负离子, 形成 P 型导电沟道, 其符号如图 3.7 所示。栅源电压 v_{GS} 可以为正、负或零值。当 $v_{GS}<0$ 时, i_D 增大; 当 $v_{GS}>0$ 时, i_D 减小; 当 v_{GS} 达到某一正值 V_P 时, 沟道完全被夹断, 即 MOS 管截止。

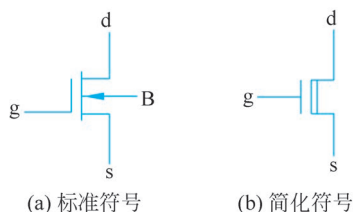


图 3.6 N 沟道耗尽型 MOS 管符号

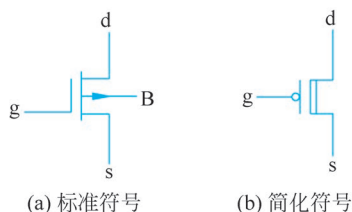


图 3.7 P 沟道耗尽型 MOS 管符号

用 N 沟道增强型 MOS 管替代图 3.1 所示的开关 S 构成的电路及其输出特性曲线如图 3.8 所示。MOS 管的作用对应于有触点开关 S 的“断开”和“闭合”。当输入为低电平(即 $v_I < V_T$) 时, MOS 管截止, 相当于开关“断开”, 输出为高电平; 当输入为高电平(即 $v_I > V_T$) 时, MOS 工作在可变电阻区, 相当于开关“闭合”, 输出为低电平。

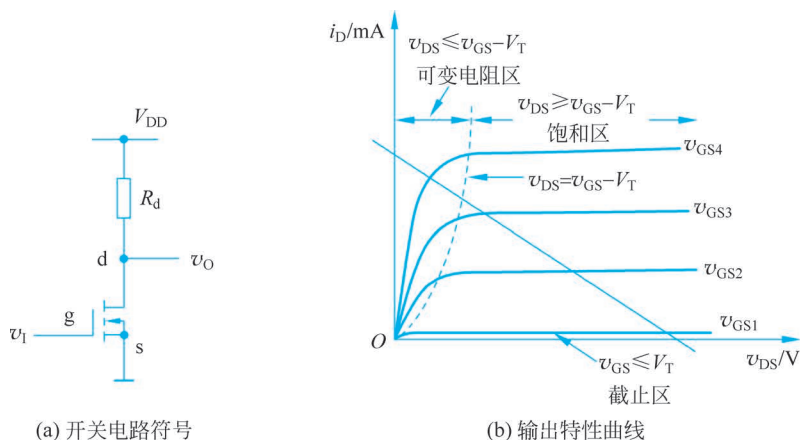


图 3.8 MOS 管开关电路

在图 3.8 所示的 MOS 管开关电路的输入端加一个理想的脉冲波形,如图 3.9 在状态转换之间有电容充、放电过程,输出波形上升沿、下降沿变得缓慢。

此外,当输入为高电平时,流过导通 NMOS 管的电流很大,图 3.8 中的电阻 R_d 起限流作用,但此时消耗在电阻 R_d 上的功率也很大。为了克服这个缺点,可用另一个 PMOS 管替代电阻 R_d ,构成 CMOS 反相器,如图 3.10 所示。两只 MOS 管的栅极连在一起作为输入端;它们的漏极连在一起作为输出端。

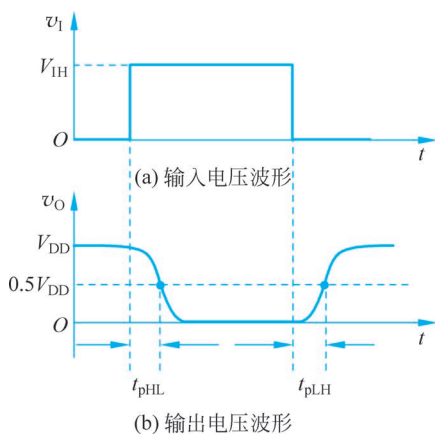


图 3.9 MOS 管开关电路电压波形

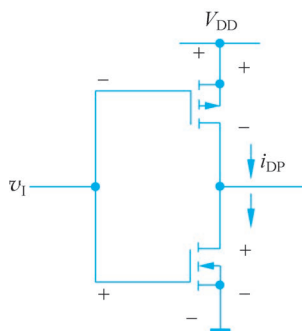
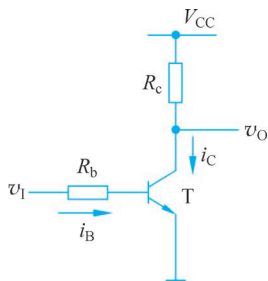


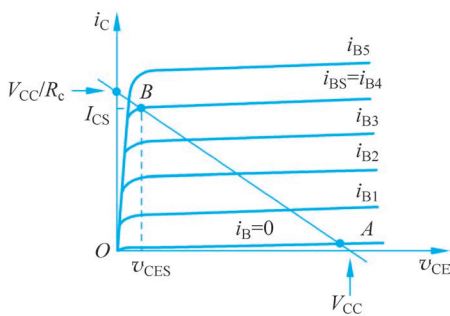
图 3.10 CMOS 反相器

2. 基本 TTL 逻辑门电路

由双极结型晶体管(bipolar junction transistor, BJT)构成的开关电路如图 3.11(a)所示,图中 BJT 为 NPN 型硅管。当输入为低电平时,例如 $v_I = 0V$ 时,BJT 的发射结为零偏($v_{BE} = 0V$),集电结为反向偏置($v_{BC} = 0V$),只有很少的漏电流流过 PN 结,故 $i_B \approx 0, i_C \approx 0, v_O = V_{CE} \approx V_{CC}$,对应于图 3.11(b)中的 A 点。这时集电极回路中的 c、e 极之间近似于开路,相当于开关断开一样,输出为高电平。BJT 处于截止的工作状态。



(a) BJT 管开关电路



(b) BJT 管输出特性

图 3.11 BJT 管开关电路及输出特性

当输入为高电平时,例如 $v_I = V_{CC}$ 时,调节电阻 R_b ,使 $i_B = V_{CC}/(\beta R_c)$,则 BJT 工作在图 3.11(b)中的 B 点,集电极电流 i_C 已接近最大值 V_{CC}/R_c ,称为集电极饱和电流 $I_{CS} \approx V_{CC}/R_c$,对应的基极电流称为基极临界饱和电流 $I_{BS} \approx V_{CC}/(\beta R_c)$ 。如果再增加 i_B ,则饱和程度加深,但 i_C 基本上保持在 I_{CS} 不再增加,集电极、发射极之间的电压 v_{CE} 为 $0.2 \sim 0.3V$,该电压称为 BJT 的饱和压降 v_{CES} ,它基本不随 i_B 增加而改变。由于 v_{CES} 很小,集电

极回路中的 c、e 极之间近似于短路,相当于开关闭合一样,输出为低电平。

BJT 的开关过程是在其饱和与截止两种状态之间的相互转换,也是内部电荷“建立”和“消散”的过程,需要一定的时间才能完成。当在图 3.11 所示的开关电路输入端加一个理想的脉冲波形,类似于图 3.9 中 MOS 管开关电路的电压波形,其输出电压 v_O 的变化会滞后于输入电压 v_I 的变化。

由于基本 BJT 反相器的动态性能不理想,可增加若干元器件构成 TTL 反相器以改善其动态性能,如图 3.12 所示。该电路由三部分组成, T_1 组成电路的输入级, T_3 、 T_4 和二极管 D 组成输出级, T_2 组成中间级作为输出级的驱动电路,将 T_2 的单端输入信号 v_{I2} 转换为互补的双端输出信号 v_{I3} 和 v_{I4} ,以驱动 T_3 和 T_4 ,最终实现反相器的逻辑关系。

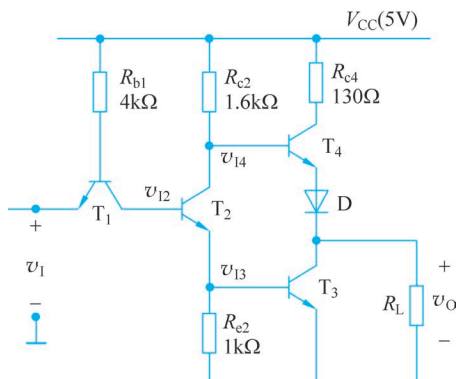


图 3.12 TTL 反相器的基本电路

3.1.2 简单逻辑门电路

简单逻辑门是指只有单一逻辑功能的门电路,如实现 3 种基本逻辑运算的“或”门、“与”门及“非”门电路,也称基本逻辑门。本节以 CMOS 门电路为例,讨论“非”门、“或”门、“与”门电路的基本工作原理。

1. “非”门电路

由 3.1.1 节可知,在 CMOS 门电路中,采用了两种在导电极性上互补的 MOS 半导体管。图 3.13(a)为用耗尽型 NMOS 管 T_5 和 PMOS 管 T_6 互补组成的 CMOS“非”门电路。 A 为输入端, F 为输出端。

当输入 A 为高电平时, T_6 截止, T_5 导通,结果输出端经 T_5 接“地”, F 为低电平,等效电路如图 3.13(b)所示。

当输入 A 为低电平时, T_5 截止, T_6 导通,结果电源经 T_6 传到输出端, F 为高电平,等效电路如图 3.13(c)所示。

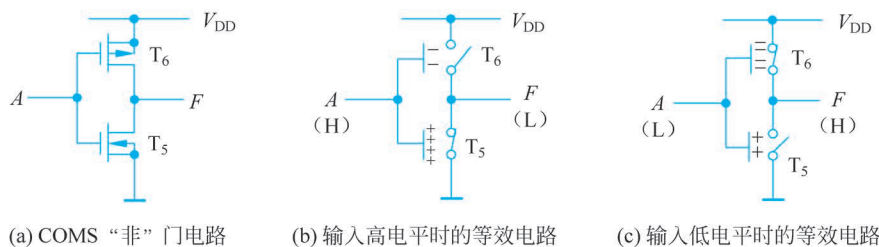


图 3.13 COMS“非”门电路及输入高电平和低电平时的等效电路

用 H 表示高电平,L 表示低电平,则上述输入和输出的电平关系如表 3.1 所示。若令 H 表示逻辑值“1”,L 表示逻辑值“0”,则表 3.1 可表示为表 3.2。表 3.2 所示即“非”运算的真值表。“非”门的逻辑表达式为 $F = \bar{A}$ 。

表 3.1 “非”门的输入、输出电位

A	F
L	H
H	L

表 3.2 “非”门的真值表

A	F
1	0
0	1

“非”门的输出总是输入的反相,故又称为反相器。

“非”门的逻辑符号如图 3.14 所示,其中,电路工作所需的电源和“地”是默认的,一般不予画出。图 3.14(a)为国内早期沿用的符号(SJ123.77 标准);图 3.14(b)为矩形轮廓图形符号(GB4728.12-85 标准规定的国标符号);图 3.14(c)为 IEEE 推荐符号(distinctive shape symbols,IEEE 1991 新版国际标准),在国外数字系统开发软件绘制的逻辑图中大都采用这一标准。本书将采用后两种标准。

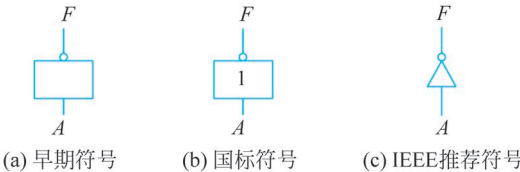


图 3.14 “非”门的逻辑符号

2. “或”门电路

实现“或”逻辑功能的电路称为“或”门。图 3.15(a)所示是 CMOS 结构组成的二输入“或”门电路。对比图 3.13(a)的“非”门电路可知, T_5 和 T_6 的接法完全相同,因此 T_5 和 T_6 在电路中等效为一个“非”门。该“非”门的输入为 P ,输出为 F ,故有 $F = \bar{P}$, A 、 B 是两个输入端。

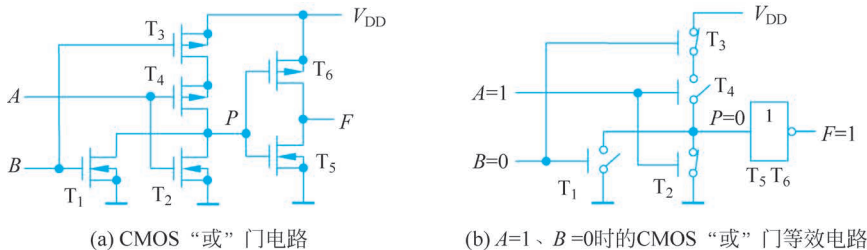


图 3.15 CMOS“或”门电路及 $A=1$ 、 $B=0$ 时的 CMOS“或”门等效电路

再来看 $T_1 \sim T_4$:一方面, T_1 和 T_3 组成互补结构,它们受输入 A 的控制,其中一个导通,另一个必然截止;而 T_2 和 T_4 组成互补结构,它们受输入 B 的控制;另一方面, T_1 和 T_2 为并联结构, T_3 和 T_4 为串联结构。

先分析 A 、 B 中至少有一个是高电平的情况。此时 T_3 和 T_4 中至少有一个截止,电源不可能传到 P 点; T_1 和 T_2 中至少有一个导通,使 P 点与“地”接通而成为低电平。于是, P 点的电平经 T_5 和 T_6 “非”运算后, F 为高电平。也就是说,只要 A 和 B 中至少有一个为高电平, F 就为高电平。图 3.15(b)所示是 A 为高电平、 B 为低电平时的等效电路图。

当 A 和 B 都是低电平时, T_3 和 T_4 都导通, T_1 和 T_2 都截止,于是电源电压 V_{DD} 经 T_3 和 T_4 传到 P 点,使 P 点为高电平,输出 F 为低电平。也就是说,只有 A 和 B 都为低电平时, F 才能为低电平。

图 3.15 中输入和输出的电平关系如表 3.3 所示。对应的真值表如表 3.4 所示。表 3.4 就是“或”运算的真值表,故图 3.15(a)所示的电路实现了“或”逻辑运算,称为“或”门。

表 3.3 “或”门的输入、输出电平

A	B	F
L	L	L
L	H	H
H	L	H
H	H	H

表 3.4 “或”门的真值表

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

图 3.16 为“或”门的逻辑符号。用逻辑函数表达的二输入“或”门逻辑功能如下：

$$F = A + B$$

集成电路“或”门的输入端可制成多个，相应逻辑符号中的输入端也可画成多个。

3. “与”门电路

实现“与”逻辑功能的电路称为“与”门。图 3.17 为 CMOS 二输入端“与”门电路。对比图 3.15(a)所示的“或”门电路可知， T_1 、 T_2 变为串联结构， T_3 、 T_4 变为并联结构，其余电路结构相同。因此，只有当 T_1 和 T_2 同时导通时， P 点才能为低电平， F 为高电平；否则 P 点为高电平， F 为低电平。也就是说，只有 A 、 B 同时为高电平时， F 才能为高电平；否则 F 为低电平。

“与”门电路的输入与输出电平关系和真值表分别如表 3.5 和表 3.6 所示。表 3.6 是“与”运算的真值表。故图 3.17 所示电路是实现“与”逻辑功能的电路，称为“与”门。

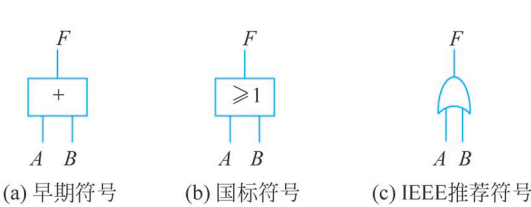


图 3.16 “或”门的逻辑符号

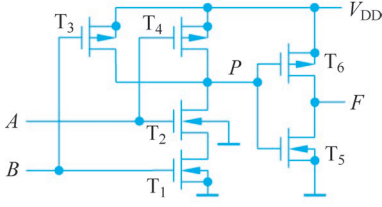


图 3.17 CMOS“与”门电路

表 3.5 “与”门的输入、输出电平

A	B	F
L	L	L
L	H	L
H	L	L
H	H	H

表 3.6 “与”门的真值表

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

“与”门的逻辑符号如图 3.18 所示，其逻辑功能可以用下列逻辑表达式表示：

$$F = AB$$

同样地，集成电路“与”门的输入端可以制成多个，逻辑符号中的输入端也可相应画成多个。

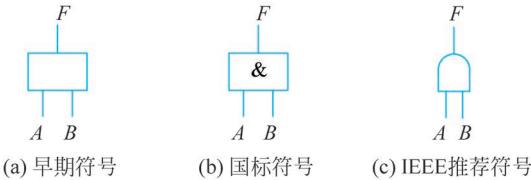


图 3.18 “与”门的逻辑符号



视频讲解

3.1.3 复合逻辑门电路

尽管“与”“或”“非”3种基本门电路可以实现各种逻辑功能,但在实际中大量存在两种或两种以上基本逻辑的复合运算。为使用方便,将经常遇到的复合运算制成集成门电路,称为复合逻辑门电路。如“与非”门、“或非”门、“与或非”门和“异或”门等。

1. “与非”门

实现“与非”逻辑功能的门电路称为“与非”门。在图 3.17 所示的“与”门中, P 点的逻辑运算关系为 $P = \overline{AB}$ 。故去掉图 3.17 中的 T_5 和 T_6 ,将 P 点直接作为输出 F ,就是一个二输入端“与非”门电路。

“与非”门的逻辑符号如图 3.19 所示,注意输出端上有一个小圆圈,它表示“非”的意思。表 3.7 是二输入“与非”门的真值表,对应的逻辑表达式为 $F = \overline{AB}$ 。

表 3.7 “与非”门的真值表

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

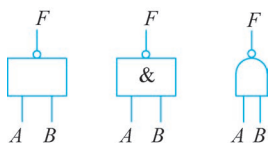


图 3.19 “与非”门的逻辑符号

集成电路“与非”门的输入端可以有多个,如 3 个、4 个等。

2. “或非”门

实现“或非”逻辑功能的门电路称为“或非”门。在图 3.15(a)所示的“或”门中, P 点的逻辑运算关系为 $P = \overline{A+B}$ 。故去掉图 3.15(a)中的 T_5 和 T_6 ,将 P 点直接作为输出 F ,就是一个二输入端“或非”门电路。

“或非”门的逻辑符号如图 3.20 所示,表 3.8 是二输入“或非”门的真值表,对应的逻辑表达式为 $F = \overline{A+B}$ 。

表 3.8 “或非”门的真值表

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

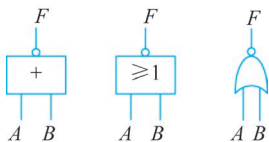


图 3.20 “或非”门的逻辑符号

集成电路“或非”门的输入端也可以有多个。

3. “与或非”门

“与”“或”“非”门 3 种运算的复合运算称为“与或非”运算,实现“与或非”逻辑功能的门电路称为“与或非”门。例如,下面是一个四变量“与或非”运算:

$$F = \overline{AB + CD}$$

实现上面的“与或非”运算的逻辑符号如图 3.21 所示。在实际电路中,“与或非”门的输入端个数、运

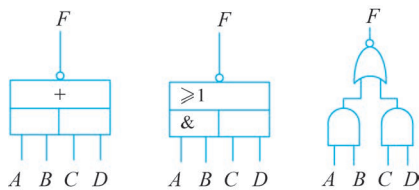


图 3.21 “与或非”门的逻辑符号

算形式将根据实际需要而定。因此,在画逻辑符号时应根据实际情况进行调整。

4. “异或”门 与“同或”门

在实际中,“异或”逻辑也是一种常用的复合逻辑,实现“异或”运算的门电路称为“异或”门,其逻辑符号如图 3.22 所示。图中, A 、 B 为输入端, F 为输出端。

“同或”门的逻辑符号如图 3.23 所示。由于“同或”实际上是“异或”之非,因此,“同或”逻辑也叫作“异或非”逻辑,其逻辑功能可用“异或”门和“非”门来实现,故“同或”门电路很少用到。

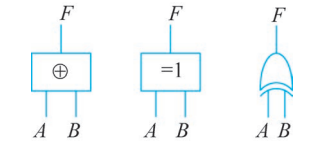


图 3.22 “异或”门的逻辑符号

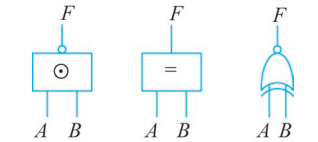


图 3.23 “同或”门的逻辑符号

5. 三态门

三态输出门(tri-state gate)简称三态门或 TS 门,是在典型门电路的基础上加控制端和控制电路构成的。前面描述的几种门电路的输出只有 2 种状态,要么为 0,要么为 1。但在三态输出门电路中,有 3 种输出状态:低阻抗的 0、1 状态和高阻抗状态。前 2 种状态与上述门电路相同,称为工作状态,第三种状态称为高阻态(或隔离状态)。CMOS 三态门的电路及逻辑符号如图 3.24 所示。此电路中, A 是输入端, F 是输出端, E 为输出允许端(或称为使能端)。由图 3.24 可知,2 个 CMOS 管的控制信号为

$$G_1 = \overline{EA}, \quad G_2 = \overline{E + A}$$

其真值表如表 3.9 所示。由表 3.9 可以看出下列规则。

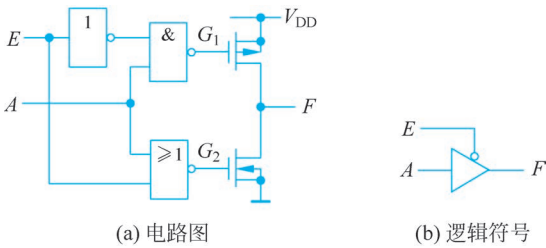


图 3.24 三态门电路及逻辑符号

表 3.9 “三态”门的真值表

E	A	G_1	G_2	F
0	0	1	1	0
0	1	0	0	1
1	0	1	0	高阻抗
1	1	1	0	高阻抗

(1) 当 $E=0$ 时,若 $A=0$,有 $G_1=G_2=1$,故上管截止,下管导通, $F=0$; 若 $A=1$,有 $G_1=G_2=0$,故上管导通,下管截止, $F=1$ 。总之,当 $E=0$ 时有 $F=A$,表示数据可以从输入端传向输出端。

(2) 当 $E=1$ 时,无论 A 为何值,总有 $G_1=1, G_2=0$ 。故上管和下管均为截止,输出端呈高阻态,输入端与输出端被隔离。

在数字系统中,三态门是常用的器件之一,常利用三态门实现数据的双向传输和多路数据切换。

3.1.4 逻辑门电路的主要外特性参数

门电路的外特性是指集成逻辑门对外表现的电气特性,在实际应用时必须认真对待。

当今,数字集成电路的品种繁多,制造工艺不尽相同。3.1.2节~3.1.3节中逻辑门的介绍仅限于CMOS工艺,此外,还有双极型工艺、PMOS工艺等。采用不同工艺制造的器件,虽然在实现的逻辑功能上是一致的,但在电气特性上却存在相当大的差异,各有特点。这里仅介绍门电路的几个主要外特性参数的定义,具体应用时请查阅相关技术手册。

1. 开门电平 V_{ON} 与关门电平 V_{OFF}

现以“非”门为例,说明开门电平与关门电平的定义。

开门电平 V_{ON} 是指使输出达到标准低电平时,应在输入端施加的最小电平值;关门电平 V_{OFF} 是指使输出达到标准高电平时,应在输入端施加的最大电平值。 V_{ON} 与 V_{OFF} 之间的差距越大,表示器件的抗干扰能力越强,但驱动器件正常工作所需的信号幅度越大。

上述定义对其他功能的逻辑门是类似的,不过输出电平的高低应根据该种门的逻辑关系而定。

2. 高电平输出 V_{OH} 与低电平输出 V_{OL}

仍以“非”门为例,说明高电平输出与低电平输出的定义。

高电平输出 V_{OH} 是指在规定负载条件下,输入端接低电平、输出端开路(逻辑1)时,器件输出的最小电压电平值;低电平输出 V_{OL} 是指在规定负载条件下,输入端接高电平、输出端是逻辑0时,器件输出的最大电压电平值。

3. 扇入系数 N_r

扇入系数 N_r 是指门电路允许的输入端数目,在器件制造时被确定。一般门电路的扇入系数为1~5,最多不超过8。例如,一个4输入端“与非”门,其 $N_r=4$ 。使用时,它最多允许有4个输入。如果要得到更多的输入端,则可用级联的方法实现。图3.25是用两个 $N_r=3$ 的“与”门和一个 $N_r=2$ 的“与非”门实现6输入“与非”运算。

如果有多余的输入端,则应在保证所需逻辑功能的前提下,将多余的输入端接“地”或接“高电平”。尤其是CMOS门的输入端必须这样处理,因为CMOS门的输入阻抗极高,悬空的输入引脚会感应空中的电磁干扰,导致电路无法正常工作。图3.26是对“与非”门和“或非”门的多余输入端的处理方法。为保证逻辑功能不变,应将多余的“与”运算输入端接高电平(电源“+”端),而多余的“或”运算输入端则应接低电平(“地”)。

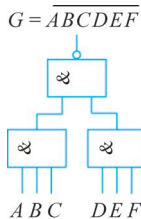


图 3.25 输入端扩展举例

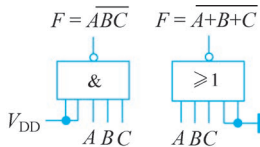


图 3.26 “与非”门和“或非”门的多余输入端的连接法

4. 扇出系数 N_c

实际应用中,门电路的输出端通常与其他门电路的输入端相连。一个门电路的输出端,最多能够驱动其他同类门电路的输入端个数,称为扇出系数。扇出系数实际上表示门的输出端带负载的能力。例如,若某“与非”门的扇出系数 $N_c=8$,表明它的输出端最多可驱动8个同类门输入端。

5. 平均时延 t_{PD}

信号通过实际逻辑门电路时都存在延迟。平均时延是指门电路的输出信号滞后于输入信号的平均时间。例如,对于“非”门电路,如果输入一个正极性的方波,则经过“非”门后,输出是一个延迟的负极性方波,如图 3.27 所示。从输入波形上升沿的 50%处,到输出波形下降沿的 50%处之间的时间间隔定义为前沿时延 t_{PLH} ,同理,定义 t_{PHL} 为后沿时延,它们的平均值称为平均时延:

$$t_{PD} = \frac{1}{2}(t_{PHL} + t_{PLH}) \tag{3-1}$$

平均时延是反映门电路工作速度的重要参数。

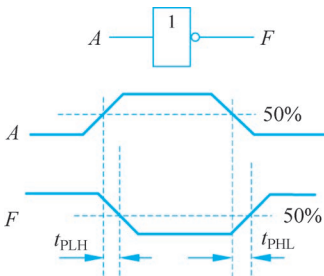


图 3.27 平均时延示意图



视频讲解

3.1.5 正逻辑与负逻辑

在上述门电路的讨论中,总是规定用高电平表示逻辑值“1”,用低电平表示逻辑值“0”,这种规定称为“正逻辑”。然而,这仅仅是一种人为的规定。如果反过来,用高电平表示逻辑值“0”,用低电平表示逻辑值“1”,即采用“负逻辑”,情况会怎样呢?

必须说明,上述两种规定都没有改变门电路内部的结构,因此其输入与输出的电位高低关系并未改变。我们关心的是,对于同一门电路,在两种规定下,所实现的逻辑功能是否相同。

以图 3.17 所示的电路为例,为方便比较,将其电平关系重绘于表 3.10 中。已知在正逻辑下,它是“与”门。再按负逻辑列出真值表,如表 3.11 所示。显然,此时的逻辑功能为“或”运算。因此,正逻辑下的“与”门是负逻辑下的“或”门。

表 3.10 图 3.17 的输入、输出电平关系

A	B	F
L	L	L
L	H	L
H	L	L
H	H	H

表 3.11 负逻辑下图 3.17 的真值表

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

类似地,可以分析图 3.15(a)所示的电路,在正逻辑下它是“或”门,但在负逻辑下它是“与”门;对于“非”门,不管是正逻辑还是负逻辑,它仍然是“非”门。

严格地讲,对于一个门电路,在未决定采用正逻辑还是负逻辑之前,就断言它是何逻辑运算关系是不妥的。然而,正逻辑符合人们的思维习惯,通常情况下,都约定用正逻辑下电路的逻辑功能来命名该电路的名称。在本书中,如无特别说明,均采用正逻辑。

为便于区分采用何种逻辑,在逻辑符号的输入端上加一个小圆圈表示负逻辑下的门电路符号。常用逻辑门的正逻辑和负逻辑符号如表 3.12 所示。

表 3.12 常用逻辑门的正逻辑和负逻辑符号

正 逻辑		负 逻辑	
逻辑符号	名 称	逻辑符号	名 称
	“或”门		“与”门
	“与”门		“或”门

续表

正 逻 辑		负 逻 辑	
逻辑符号	名 称	逻辑符号	名 称
	“与非”门		“或非”门
	“或非”门		“与非”门
	“异或”门		“同或”门

正、负逻辑的相互转换可用逻辑代数的有关定理实现,例如,正逻辑下的“与非”门,其表达式为

$$F = \overline{AB}$$

由摩根定理有

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

令 $Z = \overline{F}$, $X = \overline{A}$, $Y = \overline{B}$, 则

$$Z = \overline{X + Y} \tag{3-2}$$

将 X 、 Y 视为独立变量, Z 视为 X 、 Y 的函数, 则式(3-2)采用的是负逻辑。式(3-2)表明, 正逻辑下的“与非”门, 在负逻辑下则是“或非”门。

3.2 组合逻辑电路分析

3.2.1 组合逻辑电路的基本特点

组合逻辑电路主要由门电路构成。在电路中,任何时刻的输出仅取决于该时刻的输入信号,而与这一时刻输入信号作用前电路原来的状态没有任何关系,其电路模型可表示为图 3.28,该电路模型用函数式表示为式(3-3)。

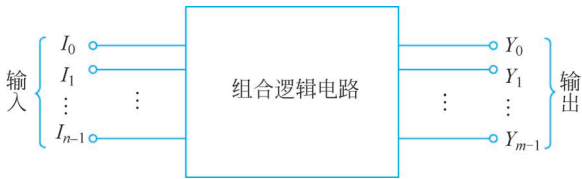


图 3.28 组合逻辑电路模型

$$\begin{cases} Y_0 = f_0(I_0, I_1, \dots, I_{n-1}) \\ Y_1 = f_1(I_0, I_1, \dots, I_{n-1}) \\ \vdots \\ Y_{m-1} = f_{m-1}(I_0, I_1, \dots, I_{n-1}) \end{cases} \tag{3-3}$$

组合逻辑电路的结构特点是由逻辑门构成的,不含记忆元件;输入信号是单向传输的,电路中含反馈回路。

根据电路输出端是一个还是多个,可将组合逻辑电路分为单输出组合逻辑电路和多输出组合逻辑电路两种类型。其功能可用逻辑函数表达式、真值表、时间波形图及逻辑图等进行描述。



视频讲解

3.2.2 组合逻辑电路分析

组合逻辑电路分析是指对于已知的逻辑电路图,推导出描述其逻辑特性的逻辑表达式,进而评述其逻辑功能的过程。组合逻辑电路的分析广泛用于系统仿制、系统维修等领域,是学习、追踪最新技术的必备手段。

组合逻辑电路分析的方法,一般是根据给出的电路图,从输入端开始,根据器件的基本数字逻辑功能,逐级推导出输出逻辑函数表达式,再根据输出逻辑函数表达式列出真值表,从而确定逻辑电路的功能。通过分析还可以进一步评价电路设计方案的优劣,改进和完善电路的结构,结合实际需要,更换逻辑电路的某些器件。对设计优秀的方案进行分析,可以吸取设计思想,为分析和设计数字系统打下基础。

下面结合具体实例来说明组合逻辑电路分析的一般步骤。

【例 3-1】 给定逻辑电路如图 3.29 所示,分析其功能,并作出功能评价。

解: (1) 写出电路的逻辑表达式。

根据图 3.29 所示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。为方便起见,标出有关中间量,如图 3.30 中的 P_1 、 P_2 和 P_3 所示。于是有

$$\begin{aligned} P_1 &= \overline{AB}, & P_2 &= \overline{BC}, & P_3 &= \overline{AC} \\ F &= \overline{P_1 P_2 P_3} = \overline{\overline{AB} \overline{BC} \overline{AC}} \end{aligned} \quad (3-4)$$

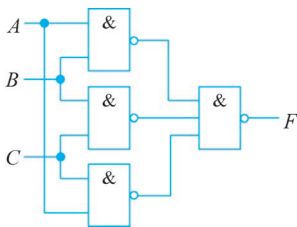


图 3.29 例 3-1 给定的逻辑电路图

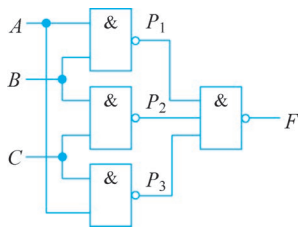


图 3.30 例 3-1 电路中的有关中间量

(2) 化简。

直接写出的逻辑函数往往不是最简的。为便于分析,需要将其化为最简“与或”表达式。用代数法化简式(3-4),有

$$\begin{aligned} F &= \overline{\overline{AB} \overline{BC} \overline{AC}} \\ &= \overline{\overline{AB}} + \overline{\overline{BC}} + \overline{\overline{AC}} \\ &= AB + BC + AC \end{aligned} \quad (3-5)$$

(3) 列出真值表。

真值表比逻辑表达式更容易看出电路的逻辑功能。由式(3-5)列出真值表,如表 3.13 所示。

(4) 分析电路的功能。

由表 3.13 可知,该电路仅当 A 、 B 和 C 中有两个或两个以上同时为 1 时,输出 F 的值为 1,其他情况下输出 F 均为 0。

表 3.13 例 3-1 的真值表

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

该电路的逻辑功能可结合具体使用场合来分析。例如,假设有 A 、 B 、 C 三个人对某事件进行表决,同意用“1”表示,不同意用“0”表示。表决结果用 F 表示, $F=1$ 表示该事件通过, $F=0$ 表示该事件未通过。则式(3-5)为一种多数表决逻辑电路。用卡诺图可以验证,该电路方案已经是最简的,不需要进一步化简。

例 3-1 详细地列出了分析逻辑电路的一般步骤,在实际应用中,可根据电路的复杂程度和分析者的熟练程度,对上述步骤进行适当取舍。

3.2.3 常用组合逻辑电路分析举例

1. 二进制加法器

在数字系统中最基本的运算是二进制加法。为说明二进制加法器的功能,下面先分析两个二进制数相加的过程。设有两个四位二进制数 a 、 b 相加,设 $a=1011$, $b=1011$,竖式演算过程如下:

$$\begin{array}{r}
 \quad \text{.....被加数} a \\
 +) \quad \text{.....加数} b \\
 \hline
 \quad \text{.....进位} c \\
 \hline
 \quad \text{.....和} s
 \end{array}$$

为了用逻辑运算实现算术运算,用逻辑 0 和 1 分别代表二进制数 0 和 1。

先看最低位。实现最低位算术加法运算的逻辑电路框图如图 3.31(a)所示。图 3.31(a)中, a_0 和 b_0 分别表示两个 1 位二进制被加数和加数, s_0 与 c_0 分别是相加产生的和与进位。这种能对两个 1 位二进制数进行相加而求得和及进位的逻辑电路称为二进制半加器,其逻辑符号如图 3.31(b)所示。半加器的两个输出 s_0 和 c_0 都是输入量 a_0 和 b_0 的函数。

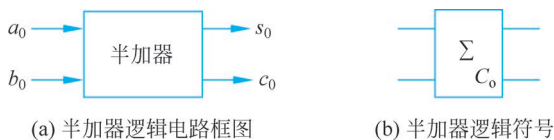


图 3.31 半加器逻辑电路框图及其逻辑符号

再看其他位的运算情况。如图 3.32(a)所示,参加相加的数除了 a_i 和 b_i ($i=1,2,3$) 两个加数位外,低位运算产生的进位 c_{i-1} 也参与加法运算,运算结果有和 s_i 及进位输出 c_i 。这种能对 3 个 1 位二进制数相加,求得和及进位的逻辑电路称为二进制全加器。其逻辑符号如图 3.32(b)所示,其中 C_i 为低位运算产生的进位输入, C_o 为本位运算产生的进位输出。

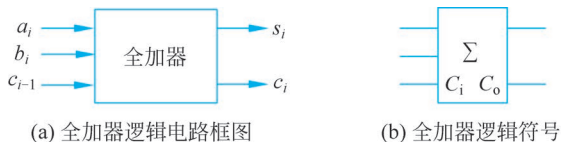


图 3.32 全加器逻辑电路框图及其逻辑符号

二进制全加器的逻辑功能齐全。不包含进位输入的二进制半加器电路更简单。以后将会看到,算术乘法电路中要用到大量的半加器。常见的集成电路加法器芯片有 7483(4 位二进制全加器)和 74283(4 位二进制超前进位全加器)。

【例 3-2】 图 3.33 所示为一个二进制半加器电路,试分析之。



视频讲解

解：由图 3.33 可写出两个输出的表达式如下。

$$S = A \oplus B, \quad C_o = AB \tag{3-6}$$

列出式(3-6)的真值表,如表 3.14 所示。

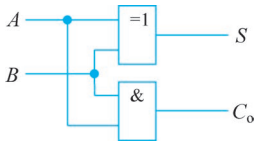


图 3.33 半加器电路图

表 3.14 半加器的真值表

A	B	S	C _o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1 位二进制数相加的算术运算规则为

$$0+0=00, \quad 0+1=01, \quad 1+0=01, \quad 1+1=10$$

对于表 3.14,显然,当把 A、B 作为两个加数,S 作为本位和,C_o 作为进位时,在正逻辑下,正好满足上述运算规则。因此,图 3.33 所示电路是一个二进制半加器。

【例 3-3】 图 3.34 所示为一个二进制全加器电路,试分析之。

解：由图 3.34,可写出电路的输出端 S 的逻辑表达式如下。

$$H = A \oplus B, \quad S = H \oplus C_i$$

因此有

$$S = A \oplus B \oplus C_i \tag{3-7}$$

输出 C_o 的逻辑表达式为

$$\begin{aligned} C_o &= \overline{\overline{AB} + \overline{HC_i}} \\ &= AB + (A \oplus B)C_i \\ &= AB + A\overline{B}C_i + \overline{A}BC_i \\ &= A(B + \overline{B}C_i) + B(A + \overline{A}C_i) \\ &= AB + (A + B)C_i \end{aligned} \tag{3-8}$$

由式(3-7)和式(3-8)可列出全加器电路的真值表,如表 3.15 所示。

表 3.15 全加器电路的真值表

C _i	B	A	S	C _o
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

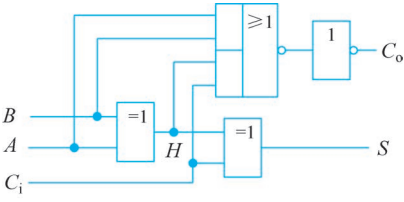


图 3.34 二进制全加器电路图

3 个 1 位二进制数算术相加的运算结果如下：

$$\begin{aligned} 0+0+0 &= 0, & 0+0+1 &= 1, & 0+1+0 &= 1, & 0+1+1 &= 10 \\ 1+0+0 &= 1, & 1+0+1 &= 10, & 1+1+0 &= 10, & 1+1+1 &= 11 \end{aligned}$$

对比表 3.15,当把 C_i 、 A 、 B 作为三个加数位, S 作为本位和, C_o 作为进位输出时,则在正逻辑下,满足二进制数的运算规则。因此,图 3.34 所示电路是一个二进制全加器。

2. 编码器与译码器

编码器与译码器是数字系统中最常用的组合逻辑部件。在数字系统中,往往需要改变原始数据的表示形式,以便存储、传输和处理数据,这一过程称为编码。例如,将二进制码变换为具有抗干扰能力的格雷码,能减少传输和处理时的误码;对图像、语音数据进行压缩,使数据量大幅减少,能降低传输和存储开销。译码则是将编码后的数据变换为原始数据的形式。无论编码器还是译码器,其逻辑功能均为将一种形式的码变换为另一种形式的码,是数字系统中广泛使用的多输入多输出组合逻辑部件。下文分析最基本的编/译码器电路。

1) 3-8 译码器

图 3.35 所示是一个由“与非”门组成的 3-8 译码器电路图,其中 CBA 为 3 位二进制码输入端, $F_7 \sim F_0$ 为 8 个输出端。3-8 译码器是 3 线至 8 线译码器的简称,其功能是将输入的 3 位二进制码译为 8 路输出。每路输出与一组二进制输入数据对应。根据图 3.35 可写出输出函数的逻辑表达式,见式(3-9)。3-8 译码器的真值表如表 3.16 所示。

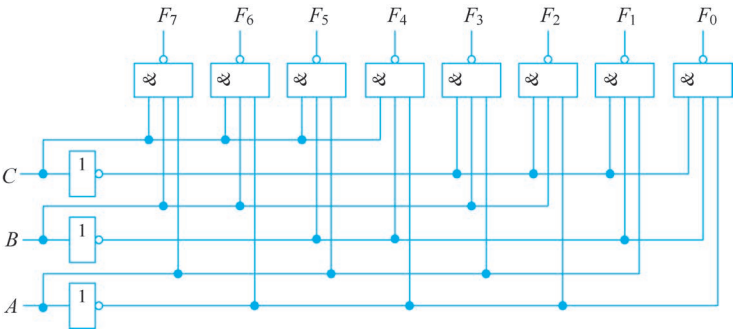


图 3.35 3-8 译码器电路图

$$\left. \begin{aligned} F_0 &= \overline{A}\overline{B}\overline{C} & F_1 &= \overline{A}\overline{B}C \\ F_2 &= \overline{A}B\overline{C} & F_3 &= \overline{A}BC \\ F_4 &= A\overline{B}\overline{C} & F_5 &= A\overline{B}C \\ F_6 &= AB\overline{C} & F_7 &= ABC \end{aligned} \right\} \quad (3-9)$$

表 3.16 3-8 译码器的真值表

<i>C</i>	<i>B</i>	<i>A</i>	<i>F</i> ₇	<i>F</i> ₆	<i>F</i> ₅	<i>F</i> ₄	<i>F</i> ₃	<i>F</i> ₂	<i>F</i> ₁	<i>F</i> ₀
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

3-8 译码器常用于地址译码、节拍分配等场合。例如,用 $F_7 \sim F_0$ 分别控制 8 个彩灯,当 $F_i (i=0,1,\cdots,7)$ 为 0 时彩灯点亮,否则彩灯熄灭。如果输入数据 CBA 每隔一段时间加 1,加到 8 时立即返回到 0,则 8 个彩灯将轮流点亮,形成流动效果。如果把 8 个彩灯换成能执行 8 种操作的逻辑单元,则可按顺序执行 8 种操作。3-8 译码器用途广泛,已制成集成电路芯片,如 74 系列的 74138 集成电路芯片。

2) 8421 码-格雷码编码器

8421 码是一种用 4 位二进制码表示一个十进制数的编码,4 个二进制位由高到低的权分别为 8、4、2、1。设 8421 码的 4 位二进制位为 $B_8B_4B_2B_1$,则其十进制数 N 为 $N=8\times B_8+4\times B_4+2\times B_2+1\times B_1$ 。

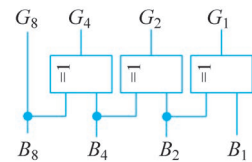


图 3.36 8421 码-格雷码编码器

图 3.36 所示是将 8421 码转换为格雷码的编码器电路。其中,输入量 $B_8B_4B_2B_1$ 为 8421 码,输出量 $G_8G_4G_2G_1$ 为格雷码。由图 3.36 可写出该编码器的输出逻辑表达式,如式(3-10)所示。

$$\left. \begin{aligned} G_8 &= B_8 \\ G_4 &= B_8 \oplus B_4 \\ G_2 &= B_4 \oplus B_2 \\ G_1 &= B_2 \oplus B_1 \end{aligned} \right\} \quad (3-10)$$

由式(3-10)列出真值表,如表 3.17 所示。表 3.17 中的输入 $B_8B_4B_2B_1$ 是 8421 码(0000~1001),输出 $G_8G_4G_2G_1$ 则是格雷码。

表 3.17 8421 码至格雷码的真值表

B_8	B_4	B_2	B_1	G_8	G_4	G_2	G_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

格雷码是一种具有一定抗误码能力的编码,格雷码的码组中任何两个相邻代码(或称码字)只有一位不同,这有利于减少干扰。数字电路中,信号在跳变时会产生尖峰脉冲干扰。例如,在一个与正弦波的幅度成正比的数据系列中,前后两个数据之差为 1 的情况会经常发生。显然,用二进制码表示这样的数据,两个相邻的数据可能有多位发生改变。例如,8421 码从 0111 增加到 1000,4 位都要发生改变。但对应的格雷码分别为 0100 和 1100,4 位中只有 G_8 位不同,这意味着尖峰干扰大幅减少。格雷码的这一特点常用于计算机系统的某些输入转换设备中,能有效地降低误码率。

3) 键盘编码器

图 3.37 所示是一个键盘编码器,能将某一个按键的输入信号编为相应的 8421 码。10 个按键分别代表十进制数 0~9,按下某一按键表示输入对应的十进制数,再由编码电路将其转换为对应的 4 位二进制码。



视频讲解

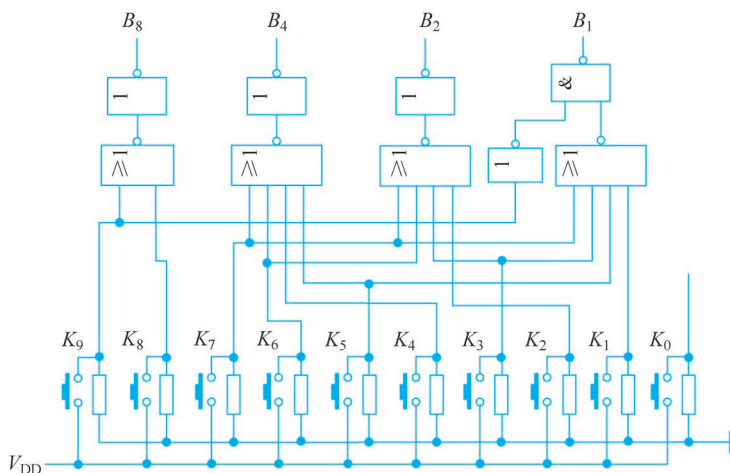


图 3.37 键盘编码器

图 3.37 中,按键未压下时,触点经电阻与地接通,向电路输入低电平。按键压下时,触点与电源 V_{DD} 接通,向电路输入高电平。设 $K_9 \sim K_0$ 为 10 个按键操作时对应的输入逻辑量,由图 3.37 可写出该编码器的输出逻辑表达式,如式(3-11)所示。

$$\left. \begin{aligned} B_8 &= \overline{\overline{K_8 + K_9}} = K_8 + K_9 \\ B_4 &= \overline{\overline{K_4 + K_5 + K_6 + K_7}} = K_4 + K_5 + K_6 + K_7 \\ B_2 &= \overline{\overline{K_2 + K_3 + K_6 + K_7}} = K_2 + K_3 + K_6 + K_7 \\ B_1 &= \overline{\overline{(K_1 + K_3 + K_5 + K_7)} K_9}} = K_1 + K_3 + K_5 + K_7 + K_9 \end{aligned} \right\} \quad (3-11)$$

由式(3-11)可列出键盘编码器的真值表,如表 3.18 所示。表 3.18 中的输出为 8421 码。图 3.37 中的 K_0 无论按下与否,电路的输出 $B_8 B_4 B_2 B_1$ 均为 0000。故 K_0 没有与图 3.37 中的任何一个门的输入端相连。

表 3.18 式(3-11)的真值表

K_9	K_8	K_7	K_6	K_5	K_4	K_3	K_2	K_1	K_0	B_8	B_4	B_2	B_1
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

当用户同时按下两个键时,电路的输出可能发生错误。例如,如果同时按下 K_7 和 K_8 , 输出为 1111,既不代表 7,也不代表 8。为避免“二义性”,定义多键同时按下时,取代表的十进制数较大的键为有效。于是, K_7 和 K_8 同时按下时,认定为仅有 K_8 按下,电路应输出 1000。实现这一编码功能的编码电路称为优先编码器,其电路如图 3.38 所示。由图 3.38 可以写出各输出的表达式,如式(3-12)所示。

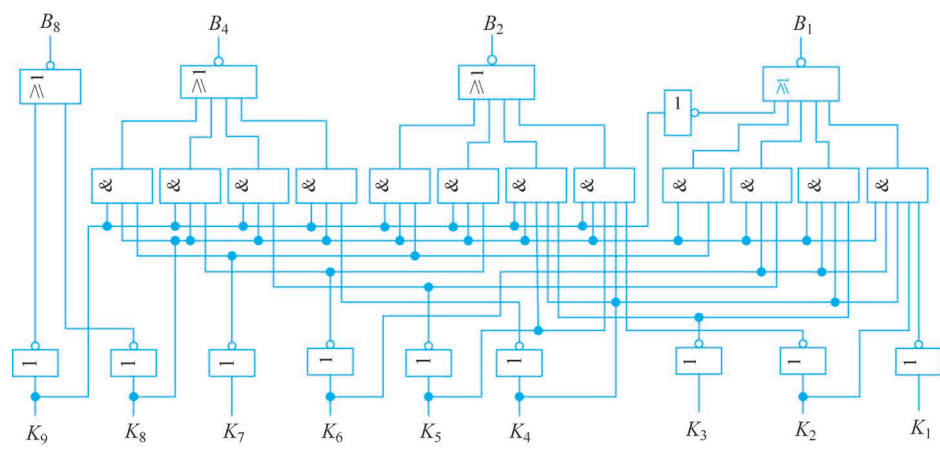


图 3.38 优先编码器电路图

$$\left. \begin{aligned} B_8 &= \overline{K_9} + K_8 \\ B_4 &= \overline{K_9} K_8 \overline{K_7} + K_9 K_8 \overline{K_6} + K_9 K_8 \overline{K_5} + K_9 K_8 \overline{K_4} \\ B_2 &= \overline{K_9} K_8 \overline{K_7} + K_9 K_8 \overline{K_6} + K_9 K_8 K_5 K_4 \overline{K_3} + K_9 K_8 K_5 K_4 \overline{K_2} \\ B_1 &= \overline{K_9} + K_8 \overline{K_7} + K_8 K_6 \overline{K_5} + K_8 K_6 K_4 \overline{K_3} + K_8 K_6 K_4 K_2 \overline{K_1} \end{aligned} \right\} \quad (3-12)$$

由式(3-12)可列出真值表,如表 3.19 所示。表 3.19 中的“Φ”为任意值,在 10 个输入 $K_9 \sim K_0$ 中,只要下标较大的为 1,则不管下标较小的为何值,均以下标较大者所代表的 8421 码作为输出,从而实现了优先编码。常见集成电路有 74147 和 74148,CMOS 定型产品 74HC147 和 74HC148,它们在逻辑功能上没有区别,只是电性能参数不同。

表 3.19 式(3-12)的真值表

K_9	K_8	K_7	K_6	K_5	K_4	K_3	K_2	K_1	K_0	B_8	B_4	B_2	B_1
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	Φ	0	0	0	1
0	0	0	0	0	0	0	1	Φ	Φ	0	0	1	0
0	0	0	0	0	0	1	Φ	Φ	Φ	0	0	1	1
0	0	0	0	0	1	Φ	Φ	Φ	Φ	0	1	0	0
0	0	0	0	1	Φ	Φ	Φ	Φ	Φ	0	1	0	1
0	0	0	1	Φ	Φ	Φ	Φ	Φ	Φ	0	1	1	0
0	0	1	Φ	Φ	Φ	Φ	Φ	Φ	Φ	0	1	1	1
0	1	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	1	0	0	0
1	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	1	0	0	1

4) 总线收发器

在计算机中,总线是各种数据的公共传输通道。总线收发器的功能是通过总线发送和

接收数据。图 3.39 所示为 8 位总线收发器的示意图,图 3.39 中 EN 为收发允许控制信号,EN=0 时,允许数据传输,EN=1 时,A、B 端呈高阻态,总线可用于其他部件之间的数据传输。DIR 为数据传输方向控制信号,DIR=0 时,总线上的数据可从 B 端传到 A 端,DIR=1 时,A 端的数据可传到总线上。常见总线收发器集成电路有 4 总线缓冲门 74125、8 总线缓冲门 74244、8 总线双向传送接收器 74245。

图 3.40 所示为 1 位总线收发器的逻辑电路。G₁、G₂ 为三态门,当要求数据从 B 端传到 A 端时,G₂ 门开通,G₁ 门呈高阻态;反之,当要求数据从 A 端传到 B 端时,G₁ 门开通,G₂ 门呈高阻态;当要求 A、B 端呈高阻态时,G₁、G₂ 门都不能开通;不允许 G₁、G₂ 门同时开通。该电路控制逻辑的任务就是将输入 DIR 和 EN 变换为控制 G₁、G₂ 门开通与否的输出信号。由图 3.40 可知输出信号为

$$G_1 = \overline{\overline{\text{EN}} \cdot \text{DIR}}, \quad G_2 = \overline{\overline{\text{EN}} \cdot \overline{\text{DIR}}}$$

(3-13)

由式(3-13)列出 G₁、G₂ 的控制逻辑真值表(见表 3.20)。

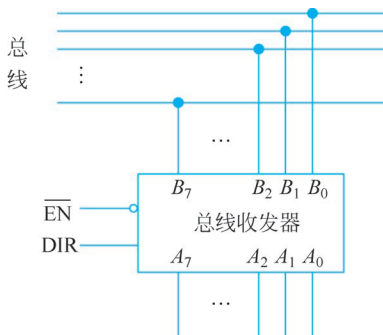


图 3.39 8 位总线收发器示意图

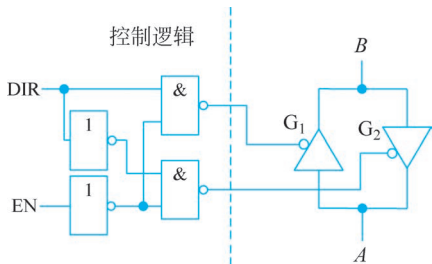


图 3.40 1 位总线收发器电路图

表 3.20 控制逻辑的真值表

输 入		输 出		功 能 说 明
EN	DIR	G ₁	G ₂	
0	0	1	0	G ₁ 门呈高阻态、G ₂ 门开通
0	1	0	1	G ₂ 门呈高阻态、G ₁ 门开通
1	0	1	1	G ₁ 、G ₂ 门呈高阻态
1	1	1	1	G ₁ 、G ₂ 门呈高阻态

由表 3.20 可以看出,该电路实现了所希望的功能。要实现多位数据的收发传输,只需将图 3.40 虚线右侧的电路重复多次,并共用控制逻辑的输出信号即可。图 3.41 为 8 总线双向传送接收器 74245 的内部逻辑电路图,其中 $\overline{\text{EA}}$ 表示低电平有效。

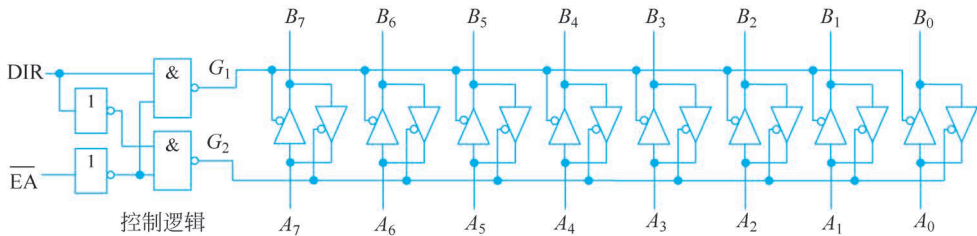


图 3.41 8 总线双向传送接收器 74245 的内部逻辑电路图



视频讲解

3.3 组合逻辑电路设计

组合逻辑电路分析是已知逻辑电路图,求出该电路能实现的功能。与此相反,组合逻辑电路设计则是根据给定的逻辑命题,设计出能实现其功能的逻辑电路。通常,逻辑命题是用文字表达的一个具有固定因果关系的事件。如果能推导出描述其功能的逻辑函数,就很容易用逻辑门实现该命题。因此,正确理解和分析逻辑命题,求出逻辑函数,是组合逻辑电路设计的最关键一步。

本节以一个简单逻辑命题为例,引入组合逻辑电路设计的步骤及方法。

【例 3-4】 设计一个判断 4 位二进制数是否大于 9 的电路,在计算机的运算器中,此功能用于将十六进制数调整为十进制数的操作中。

解: 第 1 步,分析命题,规划待设计电路的基本框架。

由命题可知,当前的结果仅与当前输入的数有关,与以前的输入和输出无关,这是一个典型的组合逻辑。要设计的电路需要 4 个输入端,分别记为 D 、 C 、 B 、 A ,用于输入 4 位二进制数;需要 1 个输出端,记为 F ,用于输出判断结果。

第 2 步,建立描述问题的逻辑函数。

首先,需要约定所有输入、输出量的值的含义。对于输入量 D 、 C 、 B 、 A ,令其分别代表 4 位二进制数由高到低的各位,该位为 1 表示输入高电平 1,为 0 表示输入低电平 0。令输出量 $F=1$ 时表示“大于”, $F=0$ 时表示“不大于”。由此,可列出描述问题的真值表,如表 3.21 所示。

由真值表 3.21 可写出输出函数 F 的最小项表达式如下。

$$F = \sum m(10, 11, 12, 13, 14, 15) \quad (3-14)$$

第 3 步,化简逻辑函数。

显然,式(3-14)可以直接用门电路实现,但所需的逻辑门较多,且需要为每个输入量提供反变量,因而线路复杂,成本高。因此,需要对 F 化简,求出最简表达式。

化简方法有公式法和卡诺图法。当输入变量不超过 4 个时,宜采用操作直观的卡诺图法。由表 3.21 作出卡诺图(见图 3.42)。

表 3.21 例 3-4 的真值表

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

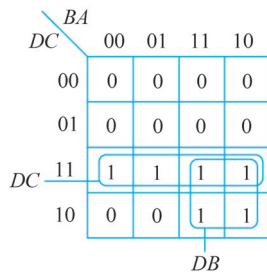


图 3.42 式(3-14)的卡诺图

在卡诺图上进行化简得到式(3-15),即

$$F = DC + DB \quad (3-15)$$

第4步,画出具体电路。

按式(3-15)画出电路,如图3.43所示。有时,希望用同类型的逻辑门来实现某逻辑功能。对式(3-15)进行如下变换:

$$F = \overline{\overline{DC}} + \overline{\overline{DB}} = \overline{\overline{DC} \overline{DB}} \quad (3-16)$$

按式(3-16)画出电路图,如图3.44所示。所需功能全部由“与非”门实现。注意,式(3-15)和式(3-16)中均不含变量A,即无论A取何值,均不影响判断结果,画图时不予考虑。

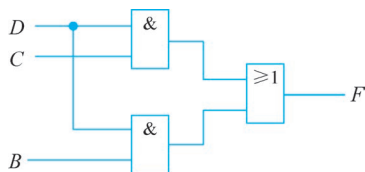


图 3.43 按式(3-15)画出电路

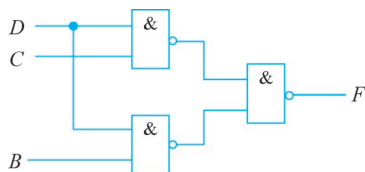


图 3.44 按式(3-16)画出电路

通过例3-4,可以归纳出组合逻辑电路设计的一般步骤及方法如下。

(1) 分析实际逻辑问题。

对命题进行全盘分析,特别注意不要遗漏细节。首先,判断问题是否为组合逻辑,如果不是组合逻辑,则不能用组合逻辑方法求解。其次,确定待设计的电路需要多少输入端、多少输出端,并明确各自的用途。

(2) 建立逻辑函数。

规定各输入和输出端上出现高电平或低电平时分别代表什么含义。对于二进制数,常用1代表高电平,0代表低电平。列出真值表,由真值表写出最小项之和(或最大项之积)形式的逻辑表达式。

(3) 化简逻辑函数。

根据题意,选择合适的化简方法,求出逻辑函数的最简表达式。

(4) 画出电路图。

根据实际需要选择合适的逻辑门,对最简表达式进行适当变换,按变换后的逻辑表达式画出电路图。

3.4 设计方法的灵活运用

3.3节中介绍的组合逻辑电路设计步骤直观、规范,具有一般性。但是,实际中的问题是千变万化的,目前,计算机部件的逻辑规模巨大、逻辑关系极为复杂。设计者应根据问题的复杂程度和积累的经验,采取灵活的方法。数字逻辑电路的设计目标是:在保证功能和性能要求的前提下,使用尽可能少的逻辑元件,以利于降低硬件成本和系统功耗。

3.4.1 逻辑代数法

在建立逻辑表达式时,灵活地运用逻辑表达式的形式,往往会简化设计难度,使设计结果更合理。



视频讲解

【例 3-5】 设计一个数值比较器,能比较两个 2 位二进制正整数的大小。

解: 第 1 步,分析实际逻辑问题,规划电路框架。

设两个 2 位二进制正整数为 $X=X_1X_2, Y=Y_1Y_2$, 比较结果用一个输出端 Z 指示。当 $X \geq Y$ 时,输出 Z 为高电平 1,否则 Z 为低电平 0。待设计的电路有 4 个输入端、1 个输出端。

第 2 步,建立逻辑函数。

根据题意列真值表,如表 3.22 所示。由表 3.22 可见, Z 为 0 的项较少,故按最大项之积形式写出的逻辑表达式较简单:

$$Z = \prod M(1,2,3,6,7,11) \quad (3-17)$$

第 3 步,化简逻辑函数。

按照上述思路,我们感兴趣的是最小项取值为 0 的方格,故在画卡诺图时,取值为 1 的最小项方格不用标注,如图 3.45 所示。然而,按 0 方格合并得到的项是 Z 的反函数,即

$$\bar{Z} = \bar{X}_1Y_1 + \bar{X}_1\bar{X}_2Y_2 + \bar{X}_2Y_1Y_2 \quad (3-18)$$

表 3.22 例 3-5 的真值表

X_1	X_2	Y_1	Y_2	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

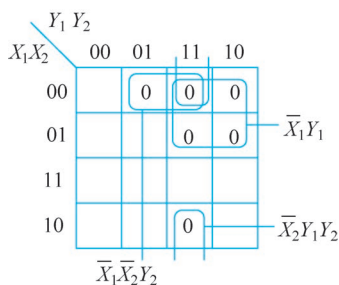


图 3.45 例 3-5 的卡诺图

对式(3-18)两边取反,就能得到 Z 的原函数:

$$Z = \overline{\bar{X}_1Y_1 + \bar{X}_1\bar{X}_2Y_2 + \bar{X}_2Y_1Y_2} \quad (3-19)$$

或

$$Z = \overline{\bar{X}_1Y_1} \overline{\bar{X}_1\bar{X}_2Y_2} \overline{\bar{X}_2Y_1Y_2} \quad (3-20)$$

$$Z = (X_1 + \bar{Y}_1)(X_1 + X_2 + \bar{Y}_2)(X_2 + \bar{Y}_1 + \bar{Y}_2) \quad (3-21)$$

第 4 步,画出电路图。

式(3-19)~式(3-21)都可由相应的门电路实现,如图 3.46 所示。究竟采用哪一个方案,要根据具体情况而定。当使用小规模集成逻辑门器件时,希望某一局部功能尽量集中在同一个芯片中,以缩短连接导线,减少导线间的干扰和传输时延。目前,标准小规模逻辑门器件大都将同一类型的几个门制作在同一个芯片中,此时,应采用图 3.46(b)所示的电路形式。然而,目前广泛采用可编程逻辑器件(programmable logic device, PLD)来实现数字系

统。即在一个芯片中有大量的、不同类型的逻辑门可用,设计者不必强调门的类型一致,应着重考虑电路运行的稳定性、可靠性和速度,力求用较少的门达到设计目标。

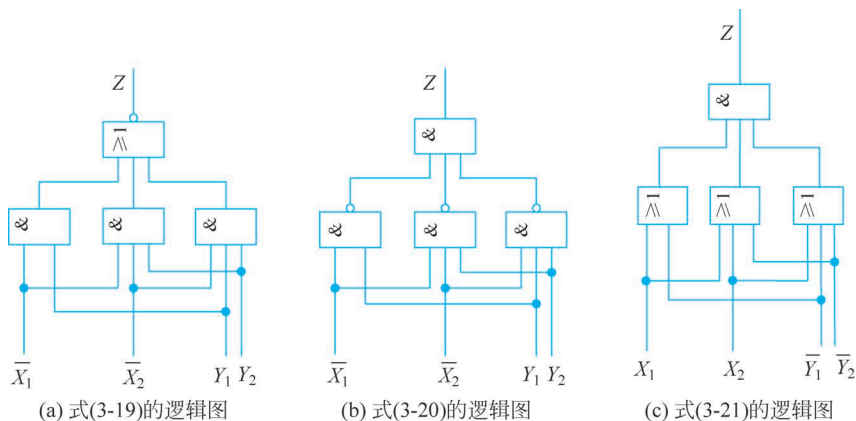


图 3.46 用 3 种形式实现例 3-5

由图 3.46 可以看出,输入量采用了反变量。在很多场合,信号源能同时提供原变量和反变量。但是,当信号源不能提供这些反变量时,不得不使用“非”门来求反。这样做一是需要的门数增加,二是门的插入会引入传输时延,导致 X 和 Y 信号传输到终端的耗时不一致,使电路的稳定性降低。在某些情况下,对逻辑函数进行适当变换可减少输入反变量。

例如,函数 $F = \overline{A}B + B\overline{C} + A\overline{B}C$, 虽然已经是最简“与非”表达式,但在实现时,各变量都需要反变量。对 F 进行变换后,变换后的表达式(3-22)就不存在反变量了。

$$\begin{aligned}
 F &= B(\overline{A} + \overline{C}) + AC\overline{B} \\
 &= B(\overline{A} + \overline{B} + \overline{C}) + AC(\overline{A} + \overline{B} + \overline{C}) \\
 &= \overline{BABC} + AC\overline{ABC}
 \end{aligned} \quad (3-22)$$

3.4.2 利用无关项简化设计

设一个组合逻辑有 m 个输入量,这些输入量的取值共有 2^m 个。在某些实际问题中,有些取值根本不会出现,或即使出现了也不予关心。利用这一现象,可简化逻辑设计。

【例 3-6】 水箱水位高度指示器如图 3.47 所示。 D 、 C 、 B 、 A 是 4 个探测针,各探测针间的距离均为 1m。当水与某针接触时,该针上产生低电平,否则该针上产生高电平。设计一个组合逻辑,以 D 、 C 、 B 、 A 为输入量,输出高度值 $Y = Y_2Y_1Y_0$ (3 位二进制数)。

解: 第 1 步,分析实际逻辑问题,规划电路框架。

要设计的电路有 4 个输入端、3 个输出端,如图 3.48 所示。这是一个多输出组合逻辑电路,在逻辑设计的实际问题中大量存在。

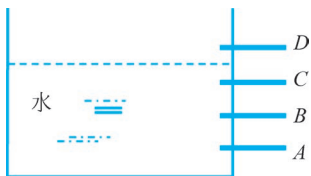


图 3.47 水箱水位高度指示器示意图



图 3.48 水位高度指示逻辑的电路框架

现在研究输入量的取值范围,4 个输入量可以表示的值有 16 种,但在本问题中只有 5 种可能的取值,如表 3.23 所示。即 $DCBA = \{0000,1000,1100,1110,1111\}$ 。其余的值不会出现,将其作为无关项考虑。

表 3.23 水位高度与输入、输出量之间的关系

水位高度/m	输 入				输 出		
	D	C	B	A	Y ₂	Y ₁	Y ₀
4	0	0	0	0	1	0	0
3	1	0	0	0	0	1	1
2	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0

第 2 步,建立逻辑函数。
对于多输出组合逻辑,在真值表中应将全部输出量一一列出。对于无关项,输出量可任意指定,用 Φ 标记,真值表如表 3.24 所示。

表 3.24 例 3-6 的真值表

D	C	B	A	Y ₂	Y ₁	Y ₀
0	0	0	0	1	0	0
0	0	0	1	Φ	Φ	Φ
0	0	1	0	Φ	Φ	Φ
0	0	1	1	Φ	Φ	Φ
0	1	0	0	Φ	Φ	Φ
0	1	0	1	Φ	Φ	Φ
0	1	1	0	Φ	Φ	Φ
0	1	1	1	Φ	Φ	Φ
1	0	0	0	0	1	1
1	0	0	1	Φ	Φ	Φ
1	0	1	0	Φ	Φ	Φ
1	0	1	1	Φ	Φ	Φ
1	1	0	0	0	1	0
1	1	0	1	Φ	Φ	Φ
1	1	1	0	0	0	1
1	1	1	1	0	0	0

由真值表得出各输出量的逻辑表达式如下:

$$\left. \begin{aligned} Y_2 &= m_0 + \sum \Phi(1,2,3,4,5,6,7,9,10,11,13) \\ Y_1 &= m_8 + m_{12} + \sum \Phi(1,2,3,4,5,6,7,9,10,11,13) \\ Y_0 &= m_8 + m_{14} + \sum \Phi(1,2,3,4,5,6,7,9,10,11,13) \end{aligned} \right\} \quad (3-23)$$

式(3-23)中,用 Φ 表示的最小项值是无关项。
第 3 步,化简逻辑函数。
为求出全部输出的逻辑表达式,对每一输出都要作出对应的卡诺图,如图 3.49 所示。在合并方格时, Φ 既可视作 0,也可视为 1,怎样对化简有利就怎样确定。

化简结果为

$$\left. \begin{aligned} Y_2 &= \overline{D} \\ Y_1 &= D\overline{B} \\ Y_0 &= B\overline{A} + D\overline{C} = \overline{\overline{B}\overline{A}}\overline{\overline{D}\overline{C}} \end{aligned} \right\} \quad (3-24)$$

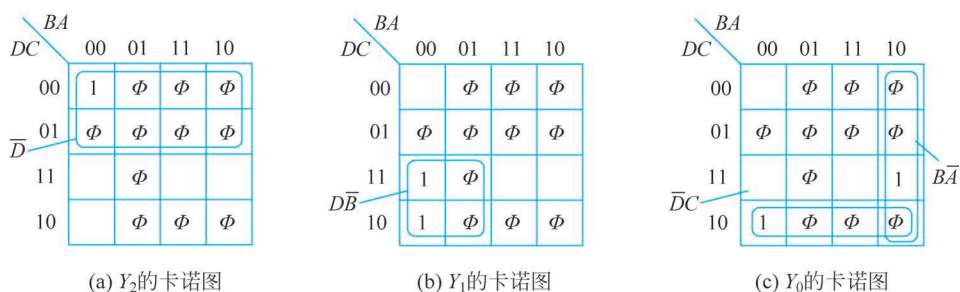


图 3.49 例 3-6 的卡诺图

第 4 步,画出电路图。

由式(3-24)画出电路图,如图 3.50 所示。

必须指出,利用无关项简化设计,得到正确输出的前提条件是输入量不出现异常值。但实际情况是复杂的,例如图 3.47 中的探测针被腐蚀、引线脱落或电磁干扰等,都可能会导致输入量异常。例如,当探测针 B 的引线脱落,水位达到 4m 时的输入为 $DCBA = 1101$,代入式(3.23)得 $Y_2Y_1Y_0 = 010$,即水位错误地指示为 2m。因此,设计者应认真分析问题的起因及后果,谨慎使用。

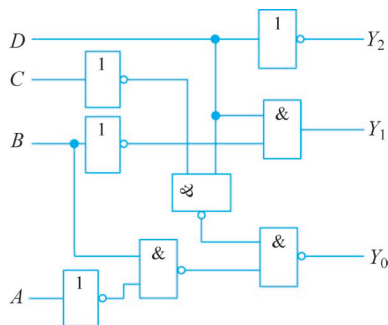


图 3.50 例 3-6 的电路图

3.4.3 分析设计法

由真值表建立给定问题的逻辑表达式,对全部输入、输出变量进行了无遗漏的枚举,是一种规范、直观的方法。但是,当输入变量较多时,列出完整的真值表是一件十分麻烦的事,而且化简也相当困难。在实际中,很多问题具有明显的规律性,对其加以分解,找出其中的基本操作步骤,对各步骤用逻辑电路予以实现,再把它们有机地结合为一个整体,是解决问题的有效方法。下面举例说明。

【例 3-7】 设计一个乘法器,实现两个 2 位二进制数相乘。

解: 乘法器是计算机的运算器中实现算术乘法运算的重要逻辑电路。如果用逻辑意义上的“1”和“0”分别表示算术意义上的数值 1 和 0,用“ $\times$ ”表示算术意义上的“乘”,“ $\cdot$ ”表示逻辑意义上的“与”,则 1 位二进制数算术乘法运算与逻辑运算的对应关系为

算术运算	对应的逻辑运算
$0 \times 0 = 0$	$0 \cdot 0 = 0$
$0 \times 1 = 0$	$0 \cdot 1 = 0$
$1 \times 0 = 0$	$0 \cdot 1 = 0$
$1 \times 1 = 1$	$1 \cdot 1 = 1$

这说明,1 位二进制数的算术乘对应于逻辑“与”。多位二进制数的代数乘又是怎样的呢?现以两个 2 位二进制数相乘为例进行说明。

设两个乘数分别为 A_1A_0 和 B_1B_0 。两个 2 位二进制数相乘,其积最多为 4 位,设为 $M_3M_2M_1M_0$ 。用手工演算乘法的过程如图 3.51 所示。仿照手工演算过程,可画出逻辑电

路图,如图 3.52 所示。图 3.52 中上部的 4 个“与”门用于实现第 1 步和第 2 步运算时产生的 4 个“与”项; 2 个半加器用于实现第 3 步运算。

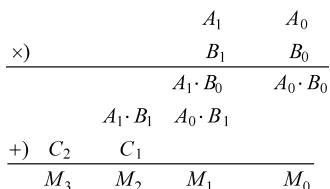


图 3.51 手工演算乘法的过程

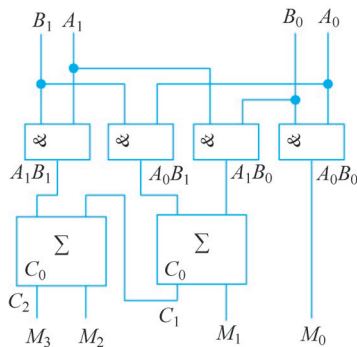


图 3.52 2 位二进制算术乘法电路

上述计算中的“+”表示算术意义上的“加”,乘法演算过程如下。

第 1 步: 用 B_0 去“ $\times$ ” A_0 , 其乘积为 $A_0 \cdot B_0$; 接着又用 B_0 去“ $\times$ ” A_1 , 其乘积为 $A_1 \cdot B_0$ 。

第 2 步: 用 B_1 分别去“ $\times$ ” A_0 和 A_1 , 分别得到积 $A_0 \cdot B_1$ 和 $A_1 \cdot B_1$ 。

第 3 步: 做“+”运算。结果: $M_0 = A_0 B_0$; M_1 是两个 1 位二进制数 $B_0 A_1$ 和 $B_1 A_0$ 相“+”, 可用半加器实现, 且产生进位 C_1 ; 类似地, M_2 也是两个 1 位二进制数 ($B_1 A_1$ 和 C_1) 相“+”的“和”, 也用半加器实现, 且产生进位 C_2 ; $M_3 = C_2$ 。

3.5 组合逻辑电路的险象

信号经过逻辑门会产生时延。不仅如此, 信号经过导线也会产生时延。时延的大小与信号经历的门数及导线的尺寸有关。因此, 输入信号经过不同的途径到达输出端需要的时间也不同。这一因素不仅会使数字系统的工作速度降低, 使信号的波形参数变坏, 而且还会在电路中产生所谓“竞争—冒险”现象, 严重时甚至使系统无法正常工作。在高速数字电路中, 尤其不可忽视这个问题。

3.5.1 险象的产生与分类

什么是险象? 下面通过具体电路进行讨论。设有如下逻辑函数:

$$F = AB + \bar{A}C \quad (3-25)$$

用“与非”门实现的电路如图 3.53(a)所示。设输入信号 $B=1, C=1, A$ 在 t_1 时刻由 1 跳变到 0。

先讨论门没有时延的理想工作情况, 如图 3.53(b)所示。在 A 下跳的同时, q, p 信号上跳, s 信号下跳。 F 信号是 q, s 信号“与非”运算的结果, 在任何时刻, q, s 中总有一个为 0, 故 F 恒为 1。因已假设跳变不需要时间, 故 t_1 时刻 F 的值是什么不予考虑。

上述理想情况在实际中并不存在。实际的逻辑门都有时延, 记为 t_{PD} 。因此, 考虑 t_{PD} 后的工作波形如图 3.53(c)所示。 q 和 p 信号分别来自于 G_2 和 G_1 门, 要等到 t_2 时刻才能跳变为 1; 从 A 到 s 经历了 G_1 和 G_3 两个门, 则 s 信号要等到 t_3 时刻才能跳变为 0。 q, s “与非”运算的结果将在 $t_2 \sim t_3$ 时间段为 0, 此结果再经 G_4 延时 t_{PD} 后才能输出, 故 F 信号

在 $t_3 \sim t_4$ 时间段为 0。这与理想的情况不一致。以此作为驱动信号会导致后续逻辑电路产生误动作,其后果往往是严重的。

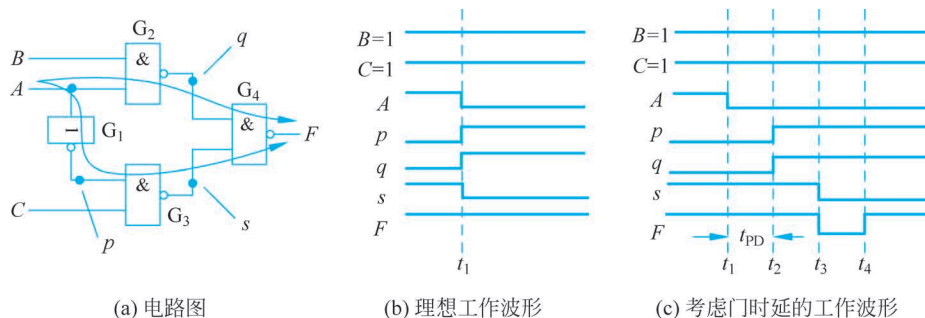


图 3.53 具有险象的逻辑电路及工作波形分析

这种现象产生的原因是, A 经过两条不同的途径传向输出端,在到达的时间上出现竞争。因两条途径的时延不同,结果输出短暂的错误脉冲,通常称为“毛刺”,如图 3.53(c)中带有阴影的脉冲。这种现象称为组合逻辑的竞争-冒险现象,简称“险象”。

按险象脉冲的极性,可将险象分为“0”型险象与“1”型险象。若险象脉冲为负极性脉冲,则称为“0”型险象;反之,若险象脉冲为正极性脉冲,则称为“1”型险象。图 3.53(c)中的险象为“0”型险象。

按输入变化前后,“正常的输出”是否应该变化,可将险象分为静态险象和动态险象。若输出本应静止不变,但险象使输出发生了不应有的短暂变化,则称为静态险象;反之,在输出应该变化的情况下出现了险象,则称为动态险象。图 3.53(c)中的险象为静态险象。

由以上两种分类方法,可组合成 4 种险象,如图 3.54 所示。

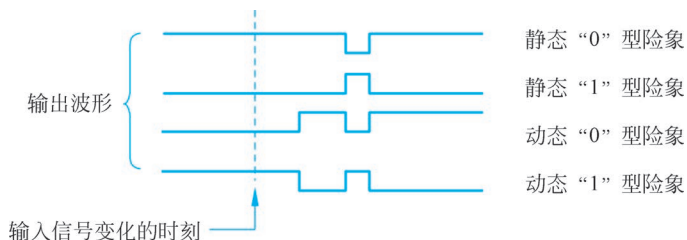


图 3.54 4 种组合险象示意图

3.5.2 险象的判断与消除

为了消除险象,需要从逻辑功能的描述形式上判断险象是否存在。描述逻辑功能的形式有代数表达式、真值表和卡诺图,下面研究如何由代数表达式和卡诺图来判断险象并讨论其消除方法。

1. 用代数法判断及消除险象

研究式(3-25),仍然令 $B=1$ 、 $C=1$ 保持不变,考察由于 A 变化产生的险象。将 $B=1$ 、 $C=1$ 代入式(3-25),有

$$F = A \cdot 1 + \bar{A} \cdot 1 = A + \bar{A} \quad (3-26)$$

式(3-26)表明,当不考虑门的时延时, $F=1$,与理想的结果一致。但若考虑“非”门的时



视频讲解

延,则 A 及 $\bar{A}$ 在跳变点不能对齐,于是产生了险象。对式(3-26)进行如下变换:

$$F = A + \bar{A} = \bar{A}A \quad (3-27)$$

式(3-26)及式(3-27)中出现 $x + \bar{x}$ 或 $x \cdot \bar{x}$ 形式的项,这样的项会产生险象。这一结论具有一般性。由此,可以得到判断险象的方法如下。

对于逻辑表达式 $F(x_n, \dots, x_i, \dots, x_1)$, 考察 $x_i (i=n, n-1, \dots, 1)$ 变化、其他量不变时是否产生险象。须将其他量的固定值代入式 $F(x_n, \dots, x_i, \dots, x_1)$ 中,若得到的表达式含有形如 $x_i + \bar{x}_i$ 或 $x_i \bar{x}_i$ 形式的项,则该逻辑表达式可能产生险象。

【例 3-8】 判断函数 $F = AB + \bar{A}C + A\bar{C}$ 描述的逻辑电路是否可能产生险象。

解: 该函数含有 3 个变量,其中 A 和 C 同时存在原变量及反变量,它们的原变量及反变量分别出现在不同的最小项中。即 A 、 C 将通过不同的时延途径,最终由“或”门汇集起来。故电路中存在竞争,但能否产生险象尚须进一步判断。

(1) 考察变量 A 。让 B 、 C 取不同的值,求 F 的表达形式,如表 3.25 所示。当 $B=1$ 、 $C=1$ 时, $F = A + \bar{A}$, 电路产生险象。

(2) 考察变量 C 。让 A 、 B 取不同的值,求 F 的表达形式,如表 3.26 所示。无论 A 、 B 取何值,电路均不产生险象。

表 3.25 考察变量 A

B	C	F	险象?
0	0	A	
0	1	$\bar{A}$	
1	0	$A + A$	
1	1	$A + \bar{A}$	√

表 3.26 考察变量 C

A	B	F	险象?
0	0	C	
0	1	C	
1	0	$\bar{C}$	
1	1	1	

例 3-8 说明,竞争并非一定产生险象。产生险象的竞争称为**临界竞争**,不产生险象的竞争称为**非临界竞争**。

如何消除险象? 当 $BC=11$ 时本应有 $F=1$,但此时却有 $F = A + \bar{A}$ 。因此若在函数中增加一个冗余项 BC ,使函数表达式成为

$$F = AB + \bar{A}C + A\bar{C} + BC \quad (3-28)$$

当 $B=1$ 、 $C=1$ 时,式(3-28)中的 BC 项恒为 1,无论其他项如何变化,恒有 $F=1$,则就不再产生险象了。

2. 用卡诺图法判断及消除险象

利用卡诺图可以更加直观地判断险象,并找出消除险象的方法。

仍以例 3-8 为例,画出卡诺图,如图 3.55 所示。在图 3.55 中,当 BC 不变时,能产生 A 及 $\bar{A}$ 的、值为 1 的最小项分别落在两个“相切”的卡诺圈①、②内;当 AB 不变时,能产生 C 及 $\bar{C}$ 的、值为 1 的最小项落在同一个卡诺圈②中。这说明,相切的卡诺圈会产生险象,这一结论具有一般性。现增加卡诺圈④(见虚线圈),使①、②“连通”,即增加一个冗余项 BC ,于是险象得以消除。最终得到图 3.56 所示的电路。

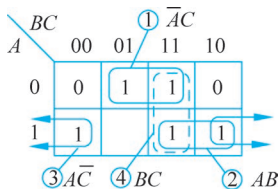


图 3.55 例 3-8 的卡诺图

【例 3-9】 逻辑函数为 $F = DC\bar{B} + \bar{D}BA + DB\bar{A}$ ，试用卡诺图法消除电路中存在的险象。

解：画出给定函数的卡诺图(见图 3.57)。

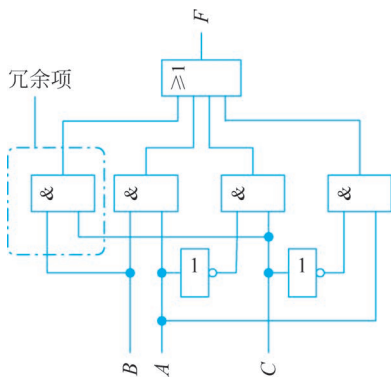


图 3.56 例 3-8 的最终电路

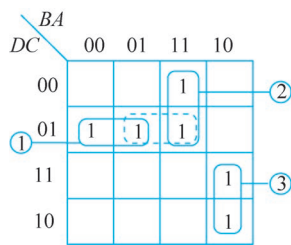


图 3.57 例 3-9 的卡诺图

图 3.57 中卡诺圈①、②相切，因此增加一个卡诺圈(见虚线圈)，使①、②连通。得到消除险象的函数表达式，如式(3-29)所示，式(3-29)中最后一项为冗余项。

$$F = DC\bar{B} + \bar{D}BA + DB\bar{A} + \bar{D}CA \quad (3-29)$$

3. 用选通法避开险象

前面的讨论仅局限于多个输入量中的某一个发生变化而产生的险象，在很多情况下，多个输入量会“同时”变化。这里的“同时”只是一种理想状态。若考虑导线的时延、元件参数的离散性等因素，希望“同时”变化的信号实际上总会有先有后；实际信号的变化率也不可能为无穷大。对于一个复杂的数字系统，要完全消除险象是非常困难的，或者要付出高昂的代价。

险象只是一种暂态过程，待电路进入稳态后，输出量即恢复成正确值。因此，使用一个选通脉冲，对稳态下的输出量取样，就能避开险象，获得正确的输出。例如，在图 3.53(a)中的输出端增加一个选通门，如图 3.58(a)所示。在取样脉冲有效($T=1$)时，选通门的输出与 F 一致，可作为电路的输出使用；当 $T=0$ 时， $G=0$ ，不予采用。只要取样脉冲有效期间发生在暂态期过后的稳态期 $t_5 \sim t_6$ ，就能保证在 $t_5 \sim t_6$ 期间得到正确的输出。

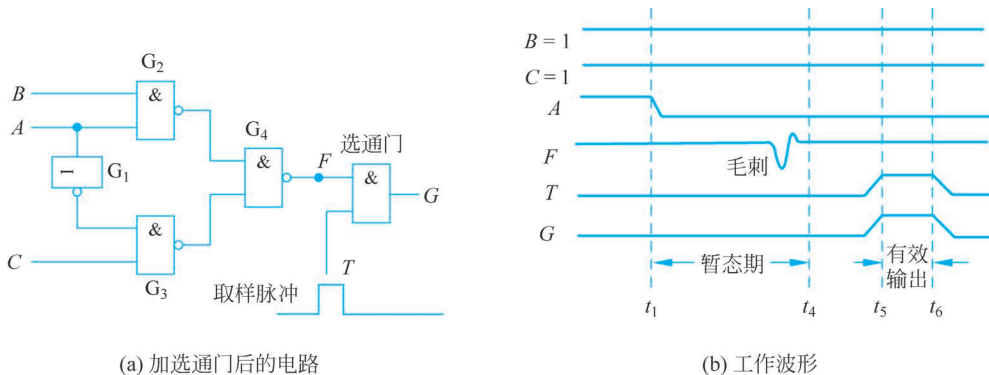


图 3.58 用选通法避开险象

3.6 常用的组合逻辑电路设计

3.6.1 8421 码加法器

8421 码加法器是实现十进制数相加的逻辑电路。8421 码用 4 位二进制数表示 1 位十进制数(0~9),4 个二进制位能表示 16 个编码,但 8421 码只利用了其中的 0000~1010 这 10 个编码,其余 6 个编码为非法编码。尽管 8421 码利用率不高,但因人们习惯了十进制,所以 8421 码加法器也是一种常用的逻辑电路。

8421 码加法器与 4 位二进制数加法运算电路不同。8421 码加法器是 2 个十进制数相加,和大于 9 时应产生进位。下面先研究 8421 码的加法运算规律,然后举例说明其设计方法。

设参与相加的量为被加数 X 、加数 Y 及来自低位 8421 码加法器的进位 C_{-1} 。设 X 、 Y 及 C_{-1} 按十进制相加,产生的和为 Z ,进位为 W 。 X 、 Y 、 Z 均为 8421 码。

先将 X 、 Y 及 C_{-1} 按二进制相加,得到的和记为 S 。显然,若 $S \leq 9$,则 S 本身就是 8421 码, S 的值与期望的 Z 值一致,进位 W 应为 0;但是,当 $S > 9$ 时, S 不再是 8421 码。此时,须对 S 进行修正,取 S 的低 4 位按二进制加 6,丢弃进位,就能得到期望的 Z 值,而此时进位 W 应为 1。现举例演算如下。

(1) 设 $X=3, Y=5, C_{-1}=1$,则 $S=X+Y+C_{-1}=9$ 。因 $S \leq 9$,故 S 的值就是 Z 值,且 W 为 0。演算过程为

$$\begin{array}{r}
 0101 \quad \cdots \cdots X \\
 0011 \quad \cdots \cdots Y \\
 + \quad 1 \quad \cdots \cdots C_{-1} \\
 \hline
 1001 \quad \cdots \cdots S
 \end{array}
 \xrightarrow{S \leq 9} \text{结果 } Z=S, W=0$$

(2) 设 $X=5, Y=9, C_{-1}=1$,则 $S=X+Y+C_{-1}=15$ 。因 $S > 9$,故 S 的值不是 Z 值,须对 S 进行加 6 修正,而 W 应为 1。演算过程为

$$\begin{array}{r}
 0101 \quad \cdots \cdots X \\
 1001 \quad \cdots \cdots Y \\
 + \quad 1 \quad \cdots \cdots C_{-1} \\
 \hline
 1111 \quad \cdots \cdots S
 \end{array}
 \xrightarrow{S > 9}
 \begin{array}{r}
 1111 \quad \cdots \cdots S \text{ 的低4位} \\
 0110 \quad \cdots \cdots 6 \\
 + \\
 \hline
 10101 \quad \cdots \cdots Z
 \end{array}
 \xrightarrow{\text{丢弃进位}} \text{结果 } \begin{matrix} Z=0101 \\ W=1 \end{matrix}$$

【例 3-10】 设计 8421 码加法器。

解: 第 1 步,分析实际逻辑问题,规划电路框架。

按上述思路,电路的框架如图 3.59 所示。图 3.59 中, C_3 是 X 、 Y 及 C_1 按二进制相加产生的进位。“4 位二进制加法器”已在 3.2.3 节进行了详细讨论,因此本例的重点是“加 6 修正”电路的设计。

现在分析“加 6 修正”电路的功能:①应能判断 $C_3S_3S_2S_1S_0$ 是否大于 9,以决定是“加 6”还是“加 0”;②要有一个二进制加法器,被加数为 $C_3S_3S_2S_1S_0$,加数为 6 或 0。因为②所述及的仍然是二进制加法器,故完成设计的关键归结为实现①述及的功能,即“ >9 判断逻辑”,其电路框架如图 3.60 所示。该电路有 5 个输入端,有一个判断结果输出端 R 。

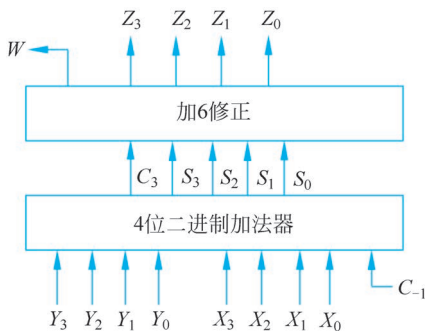


图 3.59 1 位 8421 码加法器电路框图

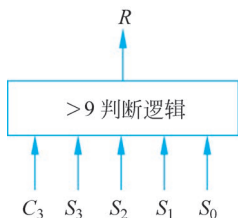


图 3.60 >9 判断电路框图

第 2 步,建立逻辑函数。

现在建立如图 3.60 所示电路的逻辑表达式。约定当输入量 $C_3S_3S_2S_1S_0 > 9$ 时,输出 R 为 1。据此列出真值表,见表 3.27。注意:表 3.27 中只列出了输入为 $0 \sim 19$ 的情况,这是因为两个 1 位十进制数及进位 C_3 相加,其和不会超过 19。

表 3.27 判断>9 逻辑的真值表

输 入						输出 R	输 入						输出 R
十进制数	C_3	S_3	S_2	S_1	S_0		十进制数	C_3	S_3	S_2	S_1	S_0	
0	0	0	0	0	0	0	10	0	1	0	1	0	1
1	0	0	0	0	1	0	11	0	1	0	1	1	1
2	0	0	0	1	0	0	12	0	1	1	0	0	1
3	0	0	0	1	1	0	13	0	1	1	0	1	1
4	0	0	1	0	0	0	14	0	1	1	1	0	1
5	0	0	1	0	1	0	15	0	1	1	1	1	1
6	0	0	1	1	0	0	16	1	0	0	0	0	1
7	0	0	1	1	1	0	17	1	0	0	0	1	1
8	0	1	0	0	0	0	18	1	0	0	1	0	1
9	0	1	0	0	1	0	19	1	0	0	1	1	1

因有 5 个输入量,不使用卡诺图化简。这里结合分析法得出 R 的逻辑表达式。观察表 3.27,当输入大于 15 时, $C_3=1$,反之 $C_3=0$ 。因此有

$$R = \overbrace{C_3}^{\substack{\uparrow \\ \text{保证输入大于 15 时 } R=1}} + \overbrace{\bar{C}_3 \cdot \sum m(10,11,12,13,14,15)}^{\substack{\uparrow \\ \text{从 } S_3S_2S_1S_0 \text{ 中提取使 } R=1 \text{ 的项}}} \quad (3-30)$$

第 3 步,化简逻辑函数。

由卡诺图化简式(3.30)中的 $\sum m(10,11,12,13,14,15)$ 项。作卡诺图,如图 3.61 所示。

结合式(3-30),得到化简结果为式(3-31)。

$$\begin{aligned} R &= C_3 + \bar{C}_3 (S_3S_2 + S_3S_1) \\ &= C_3 + S_3S_2 + S_3S_1 \\ &= \overline{\overline{C_3} \overline{S_3S_2} \overline{S_3S_1}} \end{aligned} \quad (3-31)$$

第4步,画出电路图。

现在的问题是,如何根据 R 的值产生进位 W 及加数 6 或 0。由表 3.27 可知, R 的值与 W 的值一致。而加数 6 对应的二进制数为 0110,故只需将 0110 中为“1”的位用 R 的值代替、为“0”的位接为固定低电平即可解决问题。最终得到的电路如图 3.62 所示。

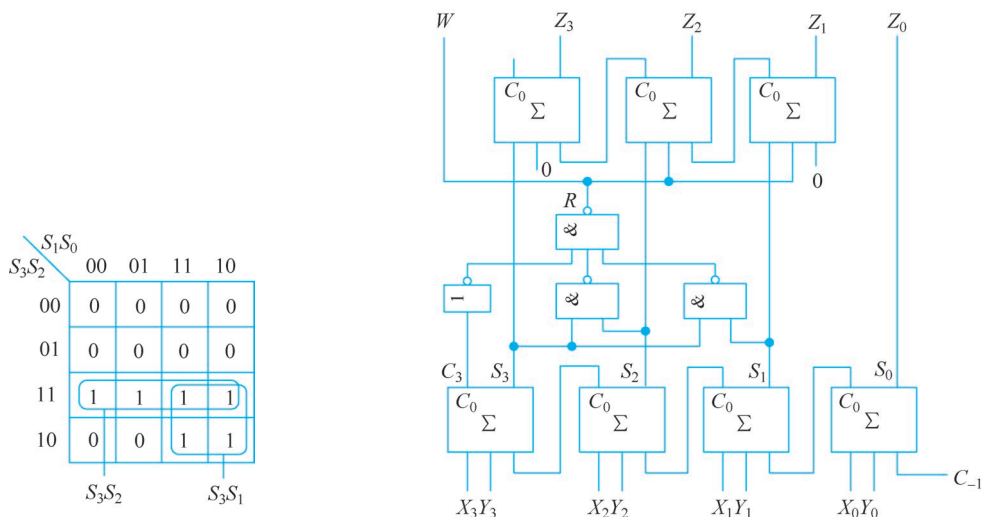


图 3.61 $\sum m(10, 11, 12, 13, 14, 15)$ 的卡诺图

图 3.62 8421 码加法器的电路图

3.6.2 数码管显示译码器

在一些电子设备中,需要将 8421 码代表的十进制数显示在数码管上,如图 3.63 所示。数码管内的各个笔画段由 LED(发光二极管)制成。每个 LED 均有一个阳极和一个阴极。对于共阴数码管,各个 LED 的阴极全部连在一起,接地,阳极由外部驱动,故驱动信号为高电平有效。当某 LED 的阳极接高电平、阴极接地时,该 LED 就会发光。共阳数码管则相反,使用时必须注意。图 3.63 中所使用的数码管是共阴数码管。

7 段译码器逻辑电路的功能是将一位 8421 BCD 码译为驱动数码管各电极的 7 个输出量 $a \sim g$ 。输入量 $DCBA$ 是 8421 码, $a \sim g$ 是 7 个输出端,分别与数码管上的对应笔画段相连(见图 3.63)。在 $a \sim g$ 中,输出为 1 的能使对应的笔画段发光,否则对应的笔画段熄灭。例如,要使数码管显示“0”字形,则 g 段不亮,其他段都亮,即要求 $abcdefg = 1111110$ 。 h 是小数点,另用一条专线驱动,不参加译码。由此可得到 7 段译码器逻辑的真值表,如表 3.28 所示。

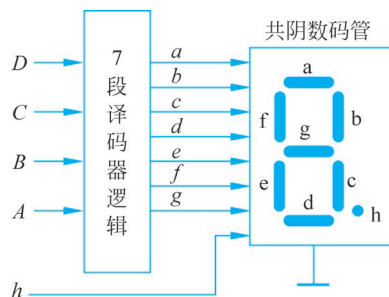


图 3.63 数码管显示译码器电路框图

对 7 个输出分别作出卡诺图,如图 3.64 所示。为简化设计,这里用卡诺圈圈定为 0 的最大项,并充分利用了无关项。注意图 3.64(e)中出现的相切卡诺圈,已被另一卡诺圈连通,险象被消除。

表 3.28 7 段译码器逻辑的真值表

输 入					输 出						
十 进 制 数	8421 码				<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>							
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

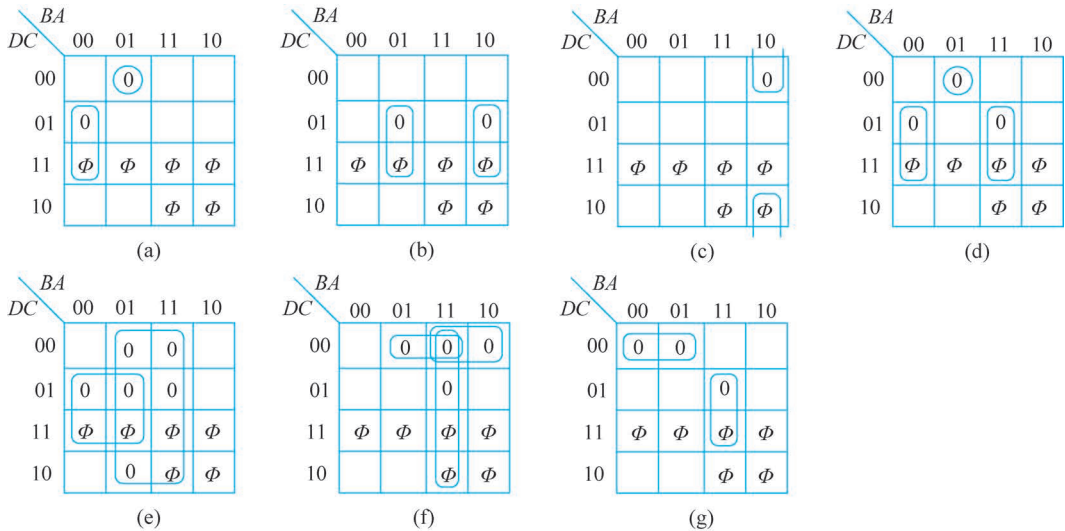


图 3.64 译码逻辑的卡诺图

化简结果如下：

$$\left. \begin{aligned} a &= \overline{DCBA} + \overline{CBA} = \overline{DCBA} \overline{CBA} \\ b &= \overline{CBA} + \overline{CBA} = \overline{CBA} \overline{CBA} \\ c &= \overline{CBA} \\ d &= \overline{DCBA} + \overline{CBA} + \overline{CBA} = \overline{DCBA} \overline{CBA} \overline{CBA} \\ e &= \overline{CB} + A = \overline{CBA} \\ f &= \overline{BA} + \overline{DCA} + \overline{DCB} = \overline{BA} \overline{DCA} \overline{DCB} \\ g &= \overline{DCB} + \overline{CBA} = \overline{DCB} \overline{CBA} \end{aligned} \right\} \quad (3-32)$$

电路实现见图 3.65。图 3.65 中共用了式(3-32)中重复出现的最大项,节省了“非与”门数量。如果输入能提供反变量,则 4 个“非”门也可以省去。

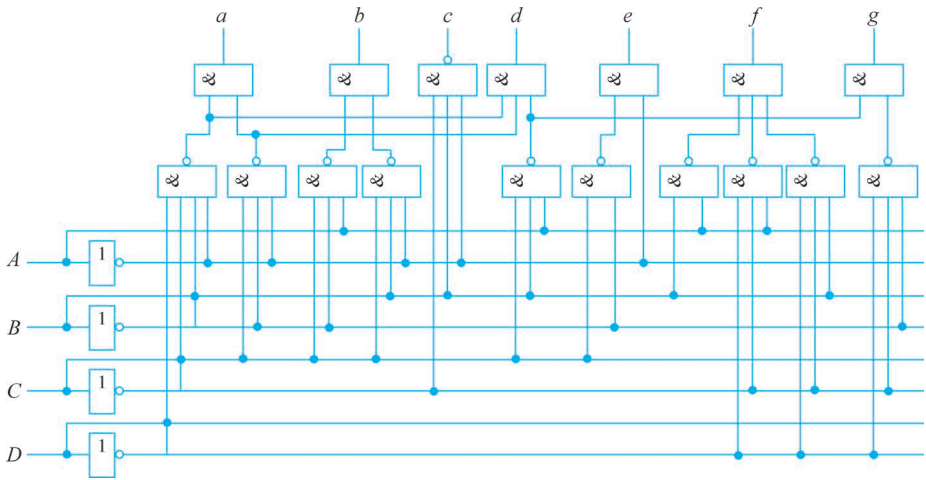


图 3.65 7 段数码管译码逻辑的电路实现

由于利用了无关项,当输入不是 8421 码时,显示结果或不是正常的数字形状,或是不希望出现的数字。例如,输入 $DCBA=1111$ 时,由式(3-32)算出: $abcdefg=1110000$,结果显示数字“7”,为克服此缺点建议采用第 9.2 节所提供的设计方法。

7 段译码器有成品出售,如 7447(共阳极 7 段译码器)、7448(共阴极 7 段译码器)等。如果想用中小规模数字集成电路实现数字系统,则可选用这些器件。



视频讲解

3.6.3 多路选择器与多路分配器

多路选择器功能:对输入的多路数据进行选择,让其中的某一路数据输出。图 3.66 所示为 4 路数据选择器的示意图, $D_3 \sim D_0$ 是 4 路输入, F 为输出, $S_1 S_0$ 是选择控制信号。输出与输入之间的关系如表 3.29 所示。例如,当 $S_1 S_0=00$ 时, D_0 从 F 端输出。

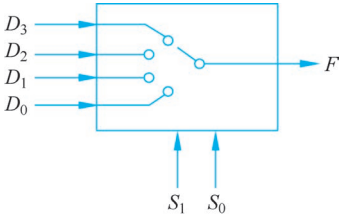


图 3.66 4 路 1 位二进制数选择器示意图

表 3.29 4 路选择器的功能表

输入 S_1	输入 S_0	输出 F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

4 路选择器的输出逻辑表达式为

$$F = D_3 S_1 S_0 + D_2 S_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_0 \bar{S}_1 \bar{S}_0 \quad (3-33)$$

一般来说, N 个选择控制端可以对 2^N 路数据进行选择,其逻辑表达式为

$$F = \sum_{k=0}^{2^N-1} D_k m_k \quad (3-34)$$

根据式(3-34)可知,式(3-33)中的 m_k 为由 $S_1 S_0$ 组成的 4 个最小项,输出量 F 由 4 项组成,每项均为一个数据位 D_k 与 m_k 进行“与”运算。无论 S_1 、 S_0 取何值, m_k 中仅有一个为 1,故对应于 m_k 为 1 的数据位被输出,达到了“4 选 1”的目的。

多路选择器的基本功能是选择数据。但是,在组合逻辑设计中常用来实现各种逻辑函数。下面举例说明。

【例 3-11】 用 4 路选择器分别实现逻辑函数: (1) $F_1 = A \oplus B$; (2) $F_2 = \overline{A+B}$ 。

解: (1) $F_1 = A \oplus B = A\bar{B} + \bar{A}B$, 对照式 (3-33), 将 A, B 分别与 S_0, S_1 对应, 并令 $D_3 D_2 D_1 D_0 = 0110$, 有

$$\begin{aligned} F_1 &= 0 \cdot AB + 1 \cdot A\bar{B} + 1 \cdot \bar{A}B + 0 \cdot \bar{A}\bar{B} \\ &= A\bar{B} + \bar{A}B \end{aligned}$$

电路实现如图 3.67 和图 3.68 所示。

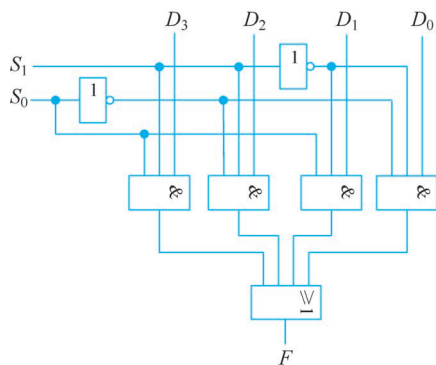


图 3.67 4路1位二进制数选择器逻辑图

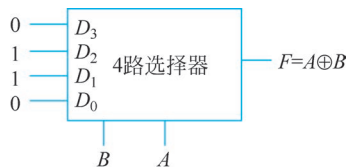


图 3.68 4路选择器实现函数 $F = A \oplus B$

(2) 与(1)类似, 令 $D_3 D_2 D_1 D_0 = 0001$, 有

$$\begin{aligned} F_1 &= 0 \cdot AB + 0 \cdot A\bar{B} + 0 \cdot \bar{A}B + 1 \cdot \bar{A}\bar{B} \\ &= \bar{A}\bar{B} = \overline{A+B} \end{aligned}$$

将图 3.68 中的 $D_3 D_2 D_1 D_0$ 改为 0001, 即可实现函数 $F_2 = \overline{A+B}$ 。

由此可知, 用多路选择器实现逻辑函数的方法是: 将选择信号视为逻辑输入变量, 将多路输入数据视为控制信息, 不同的控制信息将产生不同的逻辑函数。4 路选择器可实现任意 2 变量的逻辑函数。如果对控制信息适当调整, 4 路选择器也可实现任意 3 变量的逻辑函数。现举例说明。

【例 3-12】 用 4 路选择器实现逻辑函数: $F = \bar{A}B + B\bar{C} + A\bar{B}C$ 。

解: 将 B, C 分别视为式 (3-33) 中的 S_1, S_0 , 再对 F 进行变换, 使之具有与式 (3-33) 类似的形式, 即

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

对比式 (3-33), 只要令 $D_3 = \bar{A}, D_2 = 1, D_1 = A, D_0 = 0$, 即可达到目的。电路如图 3.69 所示。

图 3.69 中为了得到反变量, 使用了一个非门。

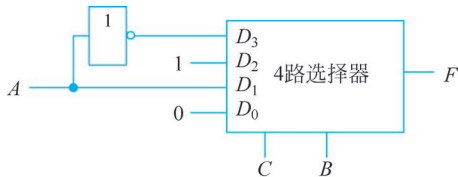


图 3.69 4路选择器实现函数 $F = \bar{A}B + B\bar{C} + A\bar{B}C$

通过以上两例可以看出,多路选择器可以方便地实现逻辑函数。一般地, 2^n 路选择器可以实现具有 n 个变量的逻辑函数,且不需要任何辅助门。并且,通过设置输入数据的值,可以很方便地改变逻辑运算关系。多路选择器已有多种型号和规格的中、小规模集成电路产品供应,如 74153(双 4 路选择器)、74151(8 路选择器)、74150(16 路选择器)等。

多路选择器内部的逻辑门数一般比专门实现同样逻辑功能的电路所需的门多。当逻辑变量较多时,所需多路选择器的路数将急剧增加,尽管可采用多片多路选择器级联、并联、加辅助逻辑等措施来实现所需的逻辑功能,但这将导致电路复杂化,功耗增加。因此,在实际中应根据具体情况而定。

与多路选择器相反,多路分配器的功能是:将输入的一位数据,有选择性地从多个输出端中的某一个输出。图 3.70 所示是 4 路分配器的功能示意图,功能表如表 3.30 所示。

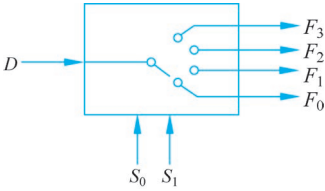


图 3.70 多路分配器功能示意图

表 3.30 4 路分配器的功能表

输入		输出			
S_1	S_0	F_3	F_2	F_1	F_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 3.30 可知,4 路分配器的输出表达式为

$$F_i = Dm_i \quad (i = 0, 1, 2, 3) \tag{3-35}$$

式(3-35)中的 m_i 为控制变量 S_1S_0 对应的四个最小项。由式(3-35),可以画出 4 路分配器的逻辑电路图,如图 3.71 所示。

多路分配器的基本功能是用于数据分路传送。利用多路分配器也可以实现逻辑函数。

【例 3-13】 用 4 路分配器实现函数 $F = A \odot B$ 。

解: 先将 F 转换为标准形式,见式(3-36)。

$$F = AB + \overline{A}\overline{B} = m_3 + m_0 \tag{3-36}$$

对比式(3-35),令 $D=1$,并增加一个辅助“或”门,即可达到目的,电路如图 3.72 所示。

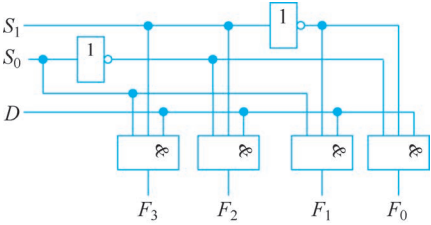


图 3.71 4 路分配器的逻辑电路图

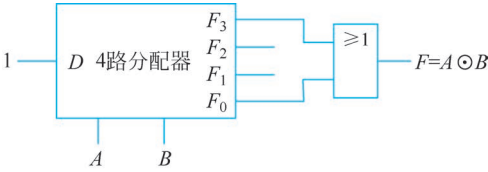


图 3.72 用 4 路分配器实现 $F = A \odot B$

多路分配器已有多种型号、规格的中、小规模集成电路产品供应,如 74139(双 4 路分配器)、74138(8 路分配器)等。

【延伸思考】

在组合逻辑电路中,每个基本逻辑门电路仅能实现一个简单的逻辑功能,但根据逻辑代数的基本定理和运算法则将其组合连接,便可实现任意复杂的组合逻辑功能。由此,我们可

延伸思考个人与集体的关系。个人是组成集体的重要单元,而集体是个人力量的凝聚,具有更强大的功能。集体主义强调国家利益、社会整体利益与个人利益的辩证统一。马克思指出,只有在集体中,个人才能获得全面发展其才能的手段。国家强大,国民才有坚强后盾;国民奋斗,国家才能继续向前。

回望历史,中华民族屡经挫折而不屈,屡遭坎坷而不衰,成为世界历史上唯一文明不曾中断的伟大民族,其重要原因也在于,“天下兴亡,匹夫有责”乃至舍生取义的集体主义精神已深深融入我们的民族意识中。其迸发出的精神力量,一次又一次地挺起这个古老民族的脊梁,铸就了这个泱泱大国的尊严。这种精神,体现为战火纷飞年代的“我以我血荐轩辕”,体现为和平建设时期的“敢教日月换新天”,体现为复兴路上的“只争朝夕,不负韶华”,也同样体现为面对灾害时的“众志成城、一方有难八方支援”。

新中国成立后,全新的社会主义制度重新定义个人、集体与国家的关系,也让集体主义产生了新的飞跃。在社会主义制度下,国家、集体、个人三者之间的根本利益是一致的。国家强大,国民才有坚强后盾;国民奋斗,国家才能继续向前。尤其这些年,海外撤侨时“你身后是强大的祖国”的底气,抢险救灾时“把人民生命放在第一位”的要求,小康路上“一个都不能少”的承诺……一次次亲历的事件,让人们切实感受到这些看似不可能实现的目标、完成的壮举,在国家政策的正确引领下,将无数个人的力量汇聚成更为强大的集体力量,均得以实现。

由此可见,个人命运与集体命运、国家民族命运的血脉相连。新时代的我们,应以国家、集体目标为导向,在现有社会的运行机制下,充分发挥个人在集体中的作用,在提升个人综合实力的同时,增强集体凝聚力、竞争力和综合性创新能力,以更好地实现个人价值。