

绪 论

电子技术既涵盖模拟电子技术,又包含数字电子技术,是研究电子电路系统的一门科学技术。当前数字电子产品在生活各个领域的应用越来越广泛,如计算机、数字电视机、海量存储设备、教学与科学研究中的仪器仪表等。数字电子产品取代其他电子产品是社会发展的主流趋势,数字电子技术是当今科学发展的主要方向之一。本章首先介绍数字电子技术的发展及应用、模拟电路(信号)和数字电路(信号)的区别、现代数字电路设计方法和传统数字电路设计方法的区别,然后给出数字电子技术基础的学习指导意见。

0.1 数字电子技术的特点

数字电子技术近半个多世纪以来取得了跨越式的发展。数字电子技术应用的典型代表是电子计算机,其从 286、386、486、586 到当前多核计算机是伴随着电子技术的发展而发展的。计算机技术的影响已遍及人类经济生活的各个领域。数字电子技术被广泛地应用于电视、通信、医学诊断、测量、控制以及家庭生活等方面,又鉴于数字信号具有便于存储、处理和传输的特点,促使许多利用模拟电子技术的产品转而运用数字技术,并且出现了许多相关新技术,如数字信号处理技术、数字图像处理技术、嵌入式技术等。

在信息技术的支持下,数字电子技术以极高的速度发展。数字电子技术所关心的已经不再是简单电子电路的设计,而是转向数字电子系统的集成,即将整体包含保护措施、抗干扰措施等数字电子系统集成在一个芯片上。

当前的数字系统设计不仅是硬件的设计,还包含了在计算机上的绘图或软件设计。现代电子技术的发展方向是硬件系统设计技术配合软件系统设计技术。因此,软件设计技术发展是数字系统强有力的支撑,数字电子技术在现代电子技术和信息技术的有机融合下将会迅猛发展。

0.1.1 模拟信号与数字信号

人类在对大自然的探索中,定义了很多物理量,其中包括 7 个基本物理量:质量、长度、时间、温度、电流强度、光强度、物质的量。在目前科学技术处理的水平上,7 个基本物理量中除了“物质的量”,其余 6 个都是连续变化的,因此由基本物理量演变生成的其他物理量也是连续变化的。在信息处理技术方面,数字电路和数字系统是依赖于数字信号的。从物理学角度看,数字信号是一些规则的,但在时间和强度上都不连续的电压或电流信号,而这些信号不是自然信号,更不是规范的物理量。随着数字技术,特别是数字计算技术的迅猛发展,使在数字处理的可靠性、精度等重要指标方面,取得了前所未有的成就。现代人类生活

的方方面面,如数控机床、数字移动电话、个人计算机、石英钟表、计算器等,都离不开数字技术。显然,数字技术与被处理的物理量之间需要转化,这就需要模拟信号和数字信号之间的转换技术。电子电路中的信号分为模拟信号与数字信号两大类。

1. 模拟信号

模拟电路中的半导体器件往往工作在放大状态,即工作在三极管伏安特性曲线的线性放大区。模拟电路灵敏度较高,但容易受到各种外部和内部变化的干扰信号的影响。模拟信号的取值连续且无限,不便于计算机的数字处理和存储,如正弦波、三角波、不规则的连续周期波等。处理模拟信号的电子电路是模拟电路,模拟电路研究各种模拟电子器件及模拟信号的变换、控制、测量和应用等内容。在工程技术上,为了便于处理和分析,通常用传感器将模拟量转换为与之成比例的电压或电流模拟信号,然后再送到电子系统中进一步处理,变成数字信号。随时间变化的电压模拟信号如图 0.1.1(a)所示,随时间变化的电压采样后模拟信号如图 0.1.1(b)所示。

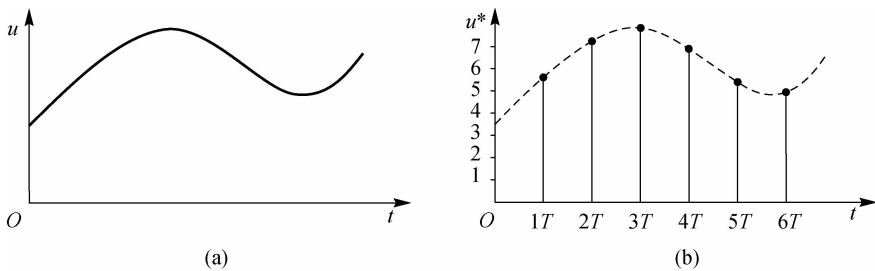


图 0.1.1 模拟信号

(a) 电压模拟信号 (b) 电压采样后模拟信号

2. 数字信号

数字电路中的半导体器件往往工作在开关状态,即工作在三极管伏安特性曲线的非线性饱和区和截止区。电路只有两个状态,而两个状态的二值逻辑便于用逻辑代数进行分析。其数字信号也便于处理和存储,可以扩充二值数字的位数以获得较高的精度。另外,数字器件工作在饱和与截止状态,因而电路受微弱电信号干扰的可能性变小,其稳定性好,可靠性高。最重要的是数字电路便于集成,因此可降低成本,减小体积。随时间变化的模拟信号采样后转换的数字信号如图 0.1.2(a)所示,随时间变化的三位数字信号如图 0.1.2(b)所示。

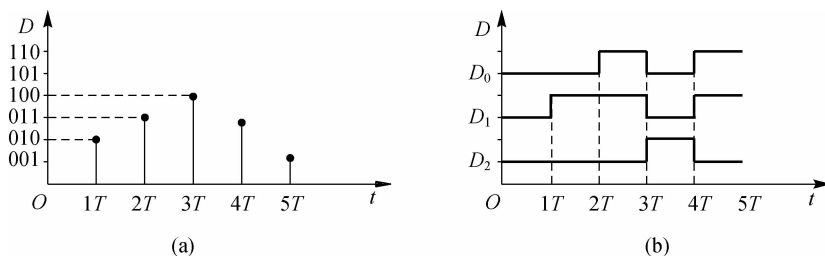


图 0.1.2 数字信号

(a) 模拟信号采样后转换的数字信号 (b) 三位数字信号

0.1.2 现代数字电路设计技术

传统的数字电路理论中,由真值表、卡诺图、布尔方程、状态表和状态图来完整描述逻辑电路的功能。这样的描述方式对于输入变量、状态变量和输出函数个数较少的、规模较小的数字系统可以勉强使用,数字系统设计的质量主要凭借设计者对逻辑设计的熟悉程度和经验,把有关逻辑功能电路拼接成预期的系统,这种设计数字系统的方法,常称为试凑法。试凑法是数字系统设计中最原始、受限制最多、效率和效果欠佳的方法,有很大的局限性,不能适应功能复杂、规模庞大的数字系统的设计要求。

目前,有许多种设计方法适合现在大规模和超大规模数字系统的设计。这些设计方法使得设计各步骤之间相互联系越来越紧密、越来越协调,以求数字系统获得良好的性能和正确的结果。如从上至下设计方法是设计者从整个系统的逻辑功能出发,进行最上层的系统设计,而后按一定的原则将全局系统分成若干子系统,逐级向下,再将每个子系统分为若干功能模块,模块还可以继续向下划分成子模块,直至分成许多最基本模块(甚至单片芯片)实现。从上至下的划分过程中,最重要的是将系统或子系统按图 0.1.3 那样划分成控制电路和若干个受控电路的功能模块。受控电路通常为设计者们所熟悉的各种功能电路,无论是取用现成模块还是自行设计的模块均可按前面讨论的方法设计,无须花费很大的精力。系统设计的主要任务是设计控制电路,而控制电路通常相当于规模不大的时序机,并且控制电路在系统或子系统中只有一个,设计工作不是很复杂。从整体上看,从上至下的设计方法将一个复杂的系统设计工作转化为一个较小规模的时序机和一些受控电路基本模块的设计问题,从而大大简化了设计的难度,缩短了设计周期。从另一角度看,大部分受控电路的基本模块已经经过验证,修改时只需对控制电路进行适当的调整,从而给设计工作带来很大的方便。

图 0.1.3(a)是从上至下设计的一个简单实例。设电路 A 代表整体系统的初始设计描述,经过适当划分,将 A 分解为两个子模块 B 和 C,再将 B 和 C 的子模块中的 D 和 E 用基本单元表示,最后将三层归纳到图 0.1.3(b)所示的逻辑电路图。值得说明的是从上至下的层次化设计的层次数一般不受限制。

数字集成电路技术与其计算机辅助设计(computer aided design,CAD)技术的发展相伴而行,相辅相成,相互促进。经过发展融合,形成电子系统设计自动化(electronic system de-

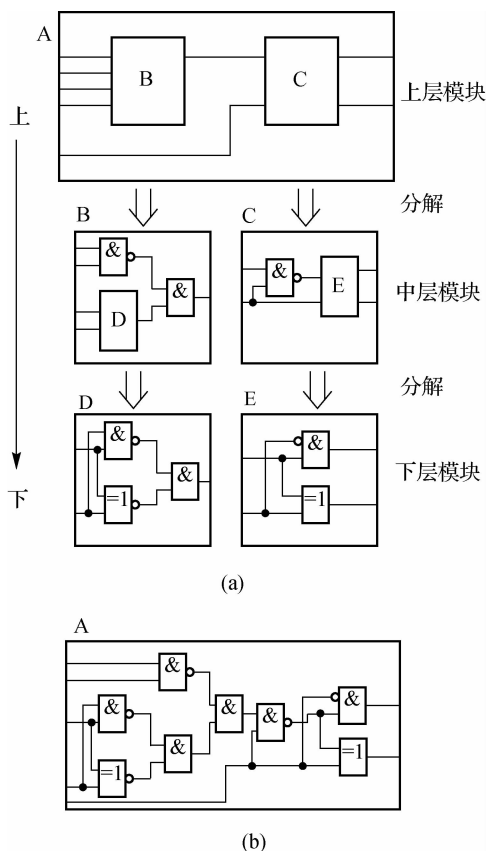


图 0.1.3 从上至下设计方法的一个简单实例
(a)电路 A 的三层分解 (b)电路 A 的逻辑

sign automation, ESDA)。初期的电子设计自动化(EDA)工具多数只能适应某一阶段的工作,不同的阶段,由掌握不同专业知识的技术人员使用相应的 EDA 工具进行设计,前一阶段的成果(输出)是后一阶段工作的依据(输入)。不同阶段的衔接,要靠人工的介入。近期发展起来的 EDA 工具是从上层系统级开始,对电路作功能描述,从上至下地跨越各个层次完成整体设计。

图 0.1.4 描述数字系统从上至下的分层设计流程图。在每一个层次上,大体都有描述、划分、综合和验证等四种类型的工作。描述是设计电路与系统的输入方法,它可以是图形输入或硬件描述语言,如 VHDL(超高速集成电路硬件描述语言)。整个设计流程只有这部分是由设计者完成,因此它是设计者和 EDA 工具的界面。系统验证、功能验证和逻辑验证都属于模拟验证,是对实际数字系统制成模型,输入计算机后,在外部将激励信号施加于此模型,通过观察模型在外部激励信号作用下的反应,判断数字系统是否能实现预期功能。综合则是分析的逆过程,给定电路应实现的功能和实现此电路的约束条件,最后得到一个满足上述要求的设计方案。

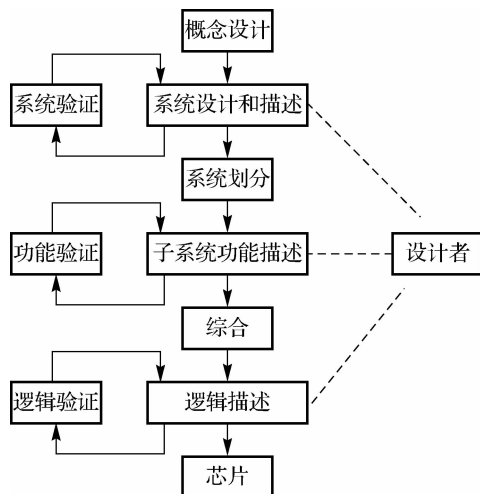


图 0.1.4 现代数字系统分层设计流程图

近几年来,数字系统的划分、综合、验证,甚至到优化都是由电子设计自动化软件平台自动完成。而且 EDA 的软件平台由于各种因素(如改进的算法、不断出现的各种功能模块等)还在不断地推陈出新,正是由于 EDA 的迅速发展,从上至下的设计方法得到广泛的应用。

1. VHDL

1987 年 IEEE 采纳 VHDL 为硬件描述语言标准(IEEE STD-1076)。VHDL 硬件描述语言包括系统行为级、寄存器传输级和逻辑门级多个设计层次,支持结构、数据流和行为 3 种描述形式的混合描述,整个大规模数字集成电路的设计过程都可以用 VHDL 来完成。VHDL 的宽范围描述能力使设计人员将工作重心转移到系统功能的实现与调试,较少的精力用于物理实现。VHDL 可以用简洁的代码描述、实现复杂控制逻辑设计,灵活且方便,设计结果易于保存和重用。VHDL 移植性好,设计不依赖于特定的器件,为众多的 EDA 厂商支持。

2. Verilog HDL

2001 年 IEEE 发布了 Verilog HDL1364-2001 标准。Verilog HDL 和 C 语言在很多语法方面相似,如 Verilog HDL 中也有 if-then-else 结构语句、for 语句、int 变量类型、函数使用等,而且语言风格类似,容易自学。Verilog HDL 既有 VHDL 的设计大规模集成电路的优点,同时兼备 C 语言简单、一目了然的特点。

目前,Verilog HDL 在美国、日本和我国台湾省应用较多,而在欧洲 VHDL 发展较好。这两种语言功能都非常强大,各有特点,均能很好地实现对硬件的描述。一般认为 Verilog HDL 在系统级抽象方面略逊于 VHDL,而在门级、开关级描述方面强于 VHDL。同时这两

种语言部分交融,如果有 C 语言编程基础,Verilog HDL 将会很容易入手。因此对大多数学习者而言,选择 Verilog HDL 还是选择 VHDL,可能更多地依赖于习惯与所处工作环境。本书将给出部分逻辑电路的 VHDL 实例,并通过实例介绍 VHDL 的基本语法。

0.2 数字电子技术基础课程

数字电子技术基础课程是高等院校工科电类专业的一门重要的技术基础课程,其教学目的是使学生掌握数字电子电路的基本概念、基本电路的分析与设计方法。该课程属于专业技术基础课程,和基础课程有着本质的区别,具有较强的工程性和实践性的特点。因此,需要能够通过数据手册读懂常用电子器件的工作原理及其性能参数,掌握用数字电路设计数字系统的分析和设计方法。能够利用常用电子仪器完成数字电路的测试、故障判断和排除。采用电子设计自动化软件平台进行设计、仿真和制作大规模数字系统,提高分析和解决实际问题的能力。

0.2.1 课程内容的分类

本书内容包含数字逻辑代数(数学基础)、数字门电路的内部结构和外部特性、中小规模组合数字电路的分析和设计、触发器、中小规模时序电路的分析和设计、常用中规模时序电路及其应用、存储器和大规模集成电路的分析和设计、脉冲产生与变换和数/模—模/数转换电路。

第 1 章的逻辑代数概论为后续的数字电路的学习奠定数学基础。第 2 章介绍数字电路基本门的内部电路原理及其外部特性。第 3 章是由最简单的数字逻辑门构成无反馈、无记忆的组合电路的分析和设计。第 4 章介绍具有反馈的基本门构成的时序电路的最基本单元触发器,为后续的时序电路的分析和设计铺平道路。第 5 章、第 6 章是具有反馈的中小规模时序集成电路的分析、设计与存储器。第 7 章介绍大规模集成电路的分析和设计方法。第 8 章是数字脉冲信号的产生与变换。第 9 章介绍模拟信号到数字信号和数字信号到模拟信号电路转换的分析和设计方法。数字电子技术知识点方框图如图 0.2.1 所示。

从图 0.2.1 可以看出分析数字电子技术的数学基础是逻辑代数。中小规模数字集成电路包括门电路、组合逻辑电路、触发器和时序逻辑电路。门电路由晶体管构成电路,若干集成基本逻辑门构成无反馈的组合逻辑电路,若干集成基本逻辑门构成有反馈的触发器,组合逻辑电路和触发器又构成时序逻辑电路。模块 D、E、F 构成中小规模集成电路,模块 G 和模块 H 是大规模集成电路,模块 I 和模块 J 是数/模混合电路。模块 K(虚线框)是模拟电路,通过模/数转换电路将输入模拟信号转换成数字信号。

计算机软件专业和部分高职院校不选学模块 C,建议 32~64 学时。其他电类专业建议 48~72 学时。

0.2.2 课程学习方法

鉴于数字电子技术基础课程具有工程性和实践性的特点,因此在学习该课程时应以研究的观点和其他知识点贯穿的方法开展学习,应注重以下几个方面。

(1)奠定良好的数学基础,掌握数字电子技术基础的基本概念、基本理论和基本分析方法。达到举一反三、触类旁通,知其所以然。

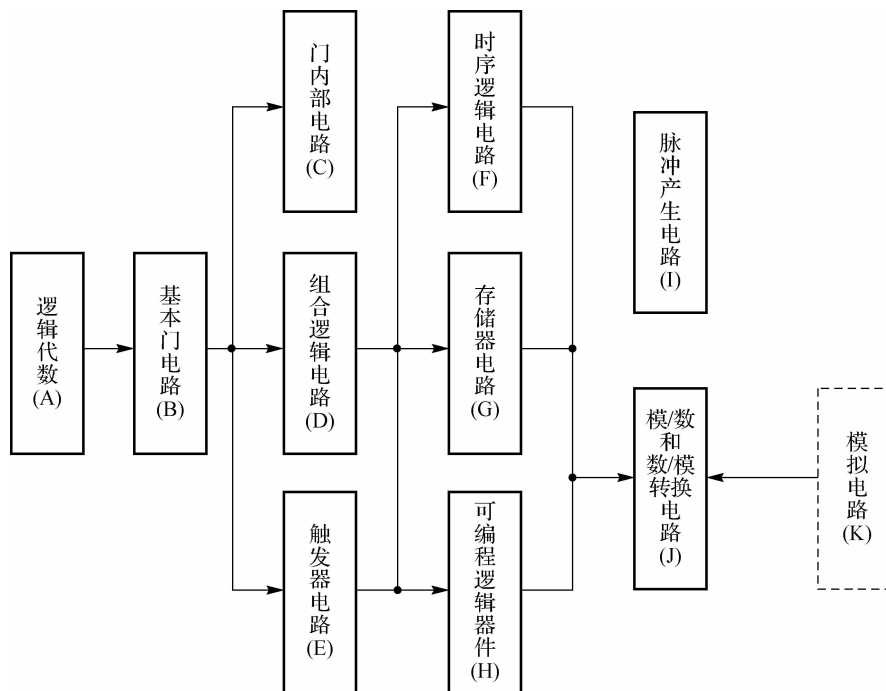


图 0.2.1 数字电子技术知识点方框图

(2)应以工程的理念学习集成电路的外部特性,以工程应用为背景探究数字集成电路的负载能力、抗干扰性能及功率匹配等。

(3)数字电子技术课程具有实践性强的特点,应十分注重实践环节,尤其是具有工程应用性质的综合设计实验。

(4)电子设计自动化在数字电子领域发展已经十分成熟,可编程逻辑器件(PLD 和 FPGA)是当前设计数字系统的主要器件,可以通过电子设计自动化软件平台进行系统设计、仿真和测试,培养现代设计数字系统的能力。

(5)学习数字电子技术基础不应仅仅局限于本课程的知识点,应和相关课程知识点贯穿,应和当今数字技术相联系,更应对目前发展的线性电路理论和非线性电路理论协同研究。

通过多年的数字电子技术课程的研究性教学,希望学习数字电子技术基础课程的学生应考虑如下问题。

(1)简述 1900—2050 年电子技术的发展与展望。

研究内容及要求:在研究性学习中,熟知电子技术的历史发展与现状,并有一全面的综述,结合专业及国际发展趋势与对未来的要求,以某一方面的发展,设想 2050 年时的电子技术发展状况。

(2)以工程概念贯穿半导体物理—数字基础—数字系统—计算机科学。

研究内容及要求:在研究性学习中,从基础理论到实际应用具有全面的认识。以某一技术针对数字电子技术的发展及在本专业中的典型应用、特点、设计手段和方法及未来发展前景等进行系统综述。

(3)简述干扰、驱动、噪声和负载匹配问题对数字系统的影响。

研究内容及要求:以工程概念,定性分析数字系统设计应掌握的工程技术问题,熟练掌握数字系统的外部特性。

(4)基于三极管输入、输出特性曲线,浅谈线性电路理论与非线性电路理论,模拟电路与数字电路的联系和区别。

研究内容及要求:分析三极管输入、输出特性曲线,从曲线的各段研究非线性和线性电路理论,并叙述涉及的领域,联系模拟和数字电路的关系,并叙述涉及的应用产品。

(5)基于从上到下的设计方法,简述集成电路的 EDA 技术,并举例说明从 PLD 或FPGA 芯片上的文字获得的信息。

研究内容及要求:掌握传统设计方法到现代设计方法的区别,了解可编程逻辑器件的设计过程,了解当代集成电路设计公司和可编程逻辑器件的软件设计平台,掌握 PLD 或FPGA 芯片上的文字信息。

(6)高性能模/数与数/模转换电路的外特性研究。

研究内容及要求:采用两种以上方案设计一个转换精度为 32 位、转换速度为 1 ns 的模/数转换电路(不计成本);结合本课程介绍的模/数与数/模转换电路,综述当今电子技术模/数与数/模领域的发展。

第 1 章

逻辑代数概论

逻辑代数是一种用于描述客观事物之间逻辑关系的数学方法,主要研究电路输出和输入间的逻辑关系,即研究输出信号和输入信号之间的关系。此处讨论的信号是数字信号,数字信号是离散的脉冲(矩形波)信号,又称为二值信号。对数字信号进行分析、设计运用的数学工具是逻辑代数,也称为布尔代数。逻辑代数是英国数学家乔治·布尔(G. Boole)首先提出来的,其起源于 19 世纪 50 年代。到 1938 年,美国数学家香农(Shannon)将其发展为更适用于分析开关电路的数学形式。逻辑代数和普通代数一样,有一套完整的运算规则,包括公理、定理和定律。它被广泛地应用于开关电路和数字逻辑电路的变换、分析、化简和设计上,因此也称为开关代数。数字产品(包括计算机)由基本门电路组成,而门电路由半导体器件组成,半导体器件工作在非线性的饱和和截止两种状态。随着数字技术的发展,逻辑代数已成为分析和设计逻辑电路的基本工具和理论基础。

本章是分析和设计数字电路的数学基础,讨论数字逻辑中数的表示方法和常用的几种编码,以及如何使用数字量描述各种信息。从应用的角度介绍逻辑代数的基本概念、基本理论和基本方法,并说明逻辑函数的基本表示方式及其化简,为读者学习和掌握数字电路的分析和设计奠定数学基础。

1.1 数制与数值表示方法

1.1.1 数制

数制就是计数方法,按一定的进位方式计数则为进位计数制。日常生活中遇到的进位计数制很多,其中以十进制数最为普遍,而在计算机或其他数字设备中采用的则是二进制数,为书写方便也采用八进制数或十六进制数。不同进位制数的标记可在数值后用下标或后缀加以区别。十、二、八和十六进制数的后缀分别为 D、B、Q 和 H。八进制数的后缀应为 O,但容易与数字 0 混淆,故用英文字母 Q 表示,对十进制数可省略下标或后缀。

首先从最熟悉的十进制分析归纳数制共同的规律和特点。十进制数制中由 0~9 十个有序数字符号和一个小数点符号“.”组成,并按“逢十进一”规律计数。数码处在不同位置有其不同含义,不同的数位有不同的权值。例如,十进制数 586.2 可表示为如下形式。

$$(586.2)_D = 5 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 + 2 \times 10^{-1}$$

式中等号左边的形式为进位制数的位置计数法,下标 D 表示十进制,也可以用下标 10 表示十进制。右边的形式为该数的按权展开式。式中 10^2 、 10^1 、 10^0 和 10^{-1} 分别表示百位、十位、个位和小数点右的第一位的权值。可见,数位越高,权值越大。对于任意一个十进制数 N ,

其位置表示法和按权展开式为

$$(N)_D = (K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}\cdots K_{-m})_D = \sum_{i=-m}^{n-1} K_i 10^i \quad (1.1.1)$$

式中, n 和 m 为正整数, 分别代表此十进制数的整数和小数部分的位数; K_i 是第 i 位的数字 $0\sim 9$; 10^i 为第 i 位的权值。

根据十进制数的讨论可推出任意进制数 R (基数) 的特征: 有一确定的基数 R , 且逢 R 进一; 有 R 个有序数字和一个小数点符号, 数字从 0 到 $R-1$; 每一个数位均有固定的含义称权 R^i , 不同位数权 R^i 不同。

任意进位制数均可写成按权展开式, 式中每一项为该位数码 K_i 和该位的权 R^i 的乘积。任意 R 进制数的一般表达式为

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}\cdots K_{-m})_R = \sum_{i=-m}^{n-1} K_i R^i \quad (1.1.2)$$

式中, n 和 m 为正整数, 分别代表此 R 进制数的整数和小数部分的位数; K_i 是第 i 位的数字 $0\sim R-1$; R^i 为第 i 位的权值。

用 2、8、16 代替 R 作为基数代入式 (1.1.2) 得到相应进制的表达式。由此可容易地写出各种进制数的按权展开式。

$$(101.1)_B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

$$(716.2)_Q = 7 \times 8^2 + 1 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1}$$

$$(F8C.2)_H = F \times 16^2 + 8 \times 16^1 + C \times 16^0 + 2 \times 16^{-1}$$

表 1.1.1 列出了几种常用数制的对照表。

表 1.1.1 几种常用数制对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

1.1.2 数制之间的转换

二进制数的基数是 2, 只有 0 和 1 两个数码。运用逻辑代数不仅运算简单, 电路也容易实现。不足之处是不便交流, 表示同一数值的数比十进制需要更多的位数, 因此数字电路系统中也常用八进制数和十六进制数。

一个数从一种进制表示形式转换到另一种进制表示形式的过程称为数制转换, 其实质是权值的转换。数制转换的原则是转换前后两个有理数的整数部分和小数部分分别相等。

1. 任意 R 进制数转换为十进制数

利用任意进制按权展开式可以将任意进制数转换成相应的十进制数。

例 1.1.1 将二进制数 101.1 转换成十进制数。

解:将二进制数的每一位乘以该位的权值,然后相加,可得

$$(101.1)_B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 4 + 0 + 1 + 0.5 = (5.5)_D$$

例 1.1.2 将八进制数 716.2 转换成十进制数。

解:将八进制数的每一位乘以该位的权值,然后相加,可得

$$(716.2)_Q = 7 \times 8^2 + 1 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} = 448 + 8 + 6 + 0.25 = (462.25)_D$$

例 1.1.3 将十六进制数 F8C.2 转换成十进制数。

解:将十六进制数的每一位乘以该位的权值,然后相加,可得

$$(F8C.2)_H = F \times 16^2 + 8 \times 16^1 + C \times 16^0 + 2 \times 16^{-1} = 3\,840 + 128 + 12 + 0.125 = (3\,980.125)_D$$

2. 十进制数转换为二进制数

通常采用基数乘除法进行十进制数值转换为二进制数。采用此方法时由于对整数和小数转换方法不同,因此应分别进行转换,然后再将两部分转换结果合并得到完整的二进制数。

(1)整数部分转换。整数部分转换采用除基(2)取余法,即用二进制基数($R=2$)去除十进制数。任意一个十进制整数可写成

$$(N)_D = K_n \times 2^n + K_{n-1} \times 2^{n-1} + \cdots + K_1 \times 2^1 + K_0 \times 2^0 \quad (1.1.3)$$

式中, $K_n, K_{n-1}, \cdots, K_1, K_0$ 是不同位数的二进制数。将式(1.1.3)除以基数 2,得余数 0 或 1,即 K_0 。将式(1.1.3)得到的商减去 K_0 ,再除以基数 2,得 K_1 。以此类推,反复执行上述过程,直到商为 0,所得的余数为二进制数的最高位 K_n 。

例 1.1.4 将十进制数 27 转换为二进制数。

解:根据“除 2 取余”法,可执行如下步骤:

	÷ 2	÷ 2	÷ 2	÷ 2	÷ 2
0	←	1	←	3	←
					6
					13
					27
	↓	↓	↓	↓	↓
余数	1	1	0	1	1
	K_4	K_3	K_2	K_1	K_0

故有 $(27)_D = (11011)_B$ 。

(2)小数部分转换。小数部分转换采用乘基(2)取整法,即用二进制的小数去乘 2。任意一个十进制纯小数可写成

$$(N)_D = K_{-1} \times 2^{-1} + K_{-2} \times 2^{-2} + \cdots + K_{-(m-2)} \times 2^{-(m-2)} + K_{-(m-1)} \times 2^{-(m-1)} \quad (1.1.4)$$

式中, $K_{-1}, K_{-2}, \cdots, K_{-(m-2)}, K_{-(m-1)}$ 是不同位数的二进制小数。将式(1.1.4)乘以基数 2,取出整数 0 或 1,即 K_{-1} 。将式(1.1.4)得到的乘积再乘以基数 2,得 K_{-2} 。以此类推,反复执行上述过程,直到小数部分为 0,或小数部分虽然不为 0,但得到的结果已经满足了精度要求,计算达到最终要求。

例 1.1.5 将十进制数 0.723 转换成误差 ϵ 不大于 2^{-5} 的二进制数。

解:根据“乘 2 取整”法,可执行如下步骤:

$$\begin{array}{ccccccccc}
 & \times 2 & & \times 2 & & \times 2 & & \times 2 & & \times 2 \\
 0.723 & \rightarrow & 0.446 & \rightarrow & 0.892 & \rightarrow & 0.784 & \rightarrow & 0.568 & \rightarrow & 0.136 \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
 \text{整数} & 1 & & 0 & & 1 & & 1 & & 1 \\
 & K_{-1} & & K_{-2} & & K_{-3} & & K_{-4} & & K_{-5}
 \end{array}$$

故有 $(0.723)_D = (0.10111)_B$, 转换误差 $\epsilon < 2^{-5}$ 。

任意包含整数和小数的十进制数可分别对其整数和小数部分按上述方法进行转换,再将两部分结果合并得到对应的二进制数。

利用除基取余法可将十进制整数转换成任意进制的整数,利用乘基取整法可将十进制小数转换成任意进制的小数,然后合并,得到任意进制数,不同之处在于基数是 R 不是 2。不过用这种方法进行转换比较烦琐。

3. 二进制数转换八进制数

八进制基数是 8, 因为 $8=2^3$, 应用 3 位二进制数来表示 1 位八进制数, 采用分组对应的方法将二进制数转换成八进制数。

将二进制数转换成八进制数时, 先从小数点开始向左、向右将二进制的整数和小数每 3 位分为一组, 不足 3 位的分别在整数的最高位前和小数的最低位后补 0, 然后将每 3 位的一组二进制数用等值的八进制数表示, 即可得到所求的八进制数。

例 1.1.6 将二进制数 11010111.0100111 转换成八进制数。

解: 根据将二进制的每 3 位为一组, 可按如下步骤:

$$\begin{array}{ccccccccc}
 011 & 010 & 111. & 010 & 011 & 100 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 3 & 2 & 7 & 2 & 3 & 4
 \end{array}$$

故有 $(11010111.0100111)_B = (327.234)_Q$ 。

反之, 可将八进制数转换成二进制数。

4. 二进制数转换十六进制数

十六进制基数是 16, 因为 $16=2^4$, 应用 4 位二进制数来表示 1 位十六进制数, 采用分组对应的方法将二进制数转换成十六进制数。所不同的是此时每 4 位分为一组, 不足 4 位的分别在整数的最高位前和小数的最低位后补 0。

例 1.1.7 将二进制数 111011.10101 转换成十六进制数。

解: 根据将二进制的每 4 位为一组, 可按如下步骤:

$$\begin{array}{ccccccc}
 0011 & 1011. & 1010 & 1000 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 3 & B & A & 8
 \end{array}$$

故有 $(111011.10101)_B = (3B.A8)_H$ 。

对于任意进制数间的转换可以借助前面介绍的各种方法进行转换, 也可以将二进制和十进制作为中间过渡数制, 间接地转换到目标数制。

1.1.3 数值表示方法

1. 无符号数二进制原码、补码及反码

各种数制都有原码和补码之分。前面介绍的十进制数和二进制数都属于原码。无符号数的补码分为两种：一种称为基数的补码；另一种称为降基数的补码，习惯上称为反码。这里仅讨论二进制数原码、反码及补码表示法。

二进制数 N 的基数的补码又称为 2 的补码，常简称为补码，其定义为

$$[N]_{\text{补}} = 2^n - N \quad (1.1.5)$$

式(1.1.5)中， n 是二进制数 N 整数部分的位数。例如

$$[1010]_{\text{补}} = 2^4 - 1010 = 10000 - 1010 = 0110$$

$$[1010.101]_{\text{补}} = 2^4 - 1010.101 = 10000.000 - 1010.101 = 0101.011$$

二进制数 N 的降基数补码又称为 1 的补码，也就是反码。其定义为

$$[N]_{\text{反}} = (2^n - 2^{-m}) - N \quad (1.1.6)$$

式(1.1.6)中， n 是二进制数 N 整数部分的位数， m 是 N 的小数部分的位数。例如

$$[1010]_{\text{反}} = (2^4 - 2^0) - 1010 = 1111 - 1010 = 0101$$

$$[1010.101]_{\text{反}} = (2^4 - 2^{-3}) - 1010.101 = 1111.111 - 1010.101 = 0101.010$$

从上述例子可以看到，在求二进制数的反码时，可对该数逐位求反得到，这也是二进制数的降基数补码称为反码的原因。

根据定义，二进制整数的补码可由反码在最低有效位加 1 得到。例如

$$N = 10110110$$

$$[N]_{\text{反}} = 01001001$$

$$[N]_{\text{补}} = 01001010$$

无论是补码还是反码，按定义再求补或求反一次，将还原为原码。

2. 二进制正负数的表示法

带符号的十进制数，如 +54、-33.6 等，由符号和绝对值两部分组成。但是数字电路不识别“+”和“-”符号。因此常用二进制数码的最高位来表示正、负号，用 0 表示正，1 表示负，称为符号位，其余各位表示数的绝对值，称为数值位，两部分合起来构成带符号的二进制数。

二进制正负数的表示法有原码、反码和补码三种表示方法。对于正数而言，三种表示法都是一样的，即符号位为 0，随后是二进制数的绝对值，即原码。例如

$$(+43)_D = 00101011$$

这里码长为一字节(即 8 位二进制位)，首位是符号位，后面 7 位是数的绝对值。

二进制负数的原码、反码和补码三种表示方法分别为符号位 1 加原码、符号位 1 加反码、符号位 1 加补码。

表 1.1.2 列出了 4 位二进制带符号数的三种表示方法。由表 1.1.2 可以看出， n 位带符号二进制数可以表示的数值范围如下。

原码： $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。

反码： $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。

补码： $-2^{n-1} \sim +(2^{n-1}-1)$ 。

表 1.1.2 4 位二进制带符号数的原码、反码和补码

十 进 制	二 进 制		
	原 码	反 码	补 码
+8	——	——	——
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1000	1111	——
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	——	——	1000

3. 补码的算术运算

在数字电路中,用原码运算求两个正数 M 和 N 的差值 $M-N$ 时,首先要对减数和被减数进行比较,然后由大数减去小数,最后决定差值的符号。完成这个运算的电路复杂,运算速度也很慢。如果用反码或补码实现减法运算,可把减法运算变成加法运算,即 $M-N$ 变为 $M+(-N)$,即被减数 M 加上减数 N 的反码或补码,这样就把原码减法运算变成了反码或补码加法运算。

1) 反码运算

反码在进行算术运算时不需要判断两数符号位是否相同,两数反码之和等于两数之和的反码,即 $[X_1]_{\text{反}} + [X_2]_{\text{反}} = [X_1 + X_2]_{\text{反}}$,符号位参加运算。当符号位有进位时需循环进位,即把符号位进位加到和的最低位。

例 1.1.8 已知 $X_1 = 0001000$, $X_2 = -0000011$,求 $X_1 + X_2$ 。

解:根据 $[X_1]_{\text{反}} + [X_2]_{\text{反}} = [X_1 + X_2]_{\text{反}}$

$$\begin{array}{r}
 [X_1]_{\text{反}} = 0\ 0001000 \\
 +) \quad [X_2]_{\text{反}} = 1\ 1111100 \\
 \hline
 \begin{array}{c} \boxed{1} 0\ 0000100 \\ +) \quad \quad \quad \rightarrow 1 \end{array} \quad \text{循环进位} \\
 \hline
 [X_1]_{\text{反}} + [X_2]_{\text{反}} = 0\ 0000101
 \end{array}$$

故得 $X_1 + X_2 = +\ 0000101$ 。

2) 补码运算

补码的运算与反码相似,两数补码之和等于两数之和的补码,即 $[X_1]_{\text{补}} + [X_2]_{\text{补}} = [X_1 + X_2]_{\text{补}}$,符号位参加运算。不过不需要循环进位,如有进位,自动丢弃。

例 1.1.9 已知 $X_1 = -0001000$, $X_2 = 0001011$, 求 $X_1 + X_2$ 。

解: 根据 $[X_1]_{\text{补}} + [X_2]_{\text{补}} = [X_1 + X_2]_{\text{补}}$

$$\begin{array}{r} [X_1]_{\text{补}} = 1\ 1111000 \\ +) [X_2]_{\text{补}} = 0\ 0001011 \\ \hline \text{自动丢弃} \leftarrow 0\ 0000011 \end{array}$$

故得 $X_1 + X_2 = +0000011$ 。

由于补码运算无循环进位, 比反码运算简单, 因而应用更为广泛。

值得注意的是, 补码的运算应在其相应位数表示的数值范围内进行, 否则将可能产生错误的计算结果。

1.2 码制与常用的编码

不同的数码不仅可以表示数值的大小, 而且还可以表示特定的信息。用一定位数的数码表示特定信息的过程称为编码。这种具有特定含义的数码称为代码, 因为它已失去了表示数值大小的含义, 仅仅是表示特定信息的代号而已。例如, 运动会中运动员的编号, 楼宇里房间的号码, 这些都是编码。数字系统中常用的是二进制编码, 就是用二进制代码表示有关对象。 n 位二进制代码有 2^n 个状态, 可以表示 2^n 个对象。下面介绍几种常用的二进制代码。

1.2.1 二-十进制码

二-十进制码(binary-coded-decimal, BCD 码), 是用二进制代码来表示人们习惯的十进制数码的编码方法。

要用二进制代码来表示十进制的 0~9 十个数, 至少要用 4 位二进制数。4 位二进制数有 16 种组合, 可从这 16 种组合中选择 10 种组合分别来表示十进制的 0~9 十个数。选择方案有多种, 这就形成了不同的 BCD 码。具有一定规律的常用的 BCD 码如表 1.2.1 所示。

表 1.2.1 常用的 BCD 码

十进制数	8421BCD 码	2421BCD 码	5421BCD 码	余 3 码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 0 0	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 0
位权	8 4 2 1 $b_3 b_2 b_1 b_0$	2 4 2 1 $b_3 b_2 b_1 b_0$	5 4 2 1 $b_3 b_2 b_1 b_0$	无权

1. 8421BCD 码

8421BCD 码的每一位权值是固定的,为恒权码。是使用最广泛的一种 BCD 码,具有自然而简单的特点。它从高位到低位的权值分别为 8、4、2、1,每一位都像二进制数一样,具有标准的 8421 位权,所以这种代码称为有权码。不过在 8421BCD 码中,是在 8421 码中去掉了后 6 种 1010~1111 状态,仅使用了 0000~1001 共 10 个状态。

2. 2421BCD 码

2421BCD 码也是恒权码。从高位到低位的权值分别为 2、4、2、1,每组代码按位权展开求和就是它所代表的十进制。由表 1.2.1 可看出,2421BCD 码具有互补性,该码中的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 这 5 对代码互为反码,即两码对应位的取值相反。

3. 余 3 码

余 3 码是在 8421BCD 码加 3 后得到的,是一种常用的 BCD 码。余 3 码也具有互补性,0 和 9、1 和 8、2 和 7、3 和 6 及 4 和 5 的码组之间互为反码。余 3 码容易求反,有利于简化 BCD 码的减法运算。余 3 码是一种无权码。

注意:BCD 码用 4 位二进制码表示的只是十进制数的一位。如果是多位十进制数,应先将该十进制数的每一位用 BCD 码表示,然后组合起来。

例 1.2.1 将十进制数 83 分别用 8421 码、2421 码和余 3 码表示。

解:由表 1.2.1 可得

$$(83)_D = (1000\ 0011)_{8421\text{BCD码}}$$

$$(83)_D = (1110\ 0011)_{2421\text{BCD码}}$$

$$(83)_D = (1011\ 0110)_{\text{余3码}}$$

1.2.2 格雷码

格雷码(Gray code)是一种常用的无权码,其编码如表 1.2.2 所示。此外,格雷码也有其他方案进行编码,不管采取哪种方案都是按照“相邻性”编码的,即相邻两码之间只有一位数码不同。0 和最大数 $2^n - 1$ 之间也只有一位数码不同,因此它是一种循环码。

表 1.2.2 格雷码

十进制数	G_3	G_2	G_1	G_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

格雷码在传输过程中引起的误差较小。例如,表示 7 的码组是 0100,相邻的 8 是 1100,仅高位码元不同。在许多数字电路中,数是按一定顺序出现的,如果相邻的数对应的码组仅有一位码元不同,那么从一个数过渡到相邻数时,不会瞬间出现别的码组。反之,若相邻数对应码组中有两个以上的码元不同,例如,8421BCD 码的 0111(7)和 1000(8),从一个数过渡到相邻数时,由于在过渡期间会瞬时出现许多别的码组,这样就可能造成逻辑上的差错,格雷码避免了这种瞬间模糊状态,所以它是错误最小化代码,从而获得了广泛的应用。

1.2.3 字符编码

在计算机应用中,为了实现人机通信,必须对所用到的十进制数码、字母和专用符号等用特定的二进制代码来表示,对这些字符的编码称为字符代码。目前最常用的字符代码是 ASCII(American Standard Code for Information Interchange)码。

ASCII 码是美国标准信息代码,为 7 位二进制字符代码,可组成 $2^7 = 128$ 种状态来表示 128 个字符,其中 96 个为图形字符(大小写英文字母各 26 个,数字字符 10 个,专用符号 34 个等),控制字符 32 个。部分字符的 ASCII 码如表 1.2.3 所示。

表 1.2.3 部分字符的 ASCII 码

字 符	ASCII 码	字 符	ASCII 码	字 符	ASCII 码
空	0100000	4	0110100	K	1001011
.	0101110	5	0110101	L	1001100
(	0101000	6	0110110	M	1001101
+	0101011	7	0110111	N	1001110
\$	0100100	8	0111000	O	1001111
*	0101010	9	0111001	P	1010000
)	0101001	A	1000001	Q	1010001
—	0101101	B	1000010	R	1010010
/	0101111	C	1000011	S	1010011
,	0101100	D	1000100	T	1010100
^	0100111	E	1000101	U	1010101
=	0111101	F	1000110	V	1010110
0	0110000	G	1000111	W	1010111
1	0110001	H	1001000	X	1011000
2	0110010	I	1001001	Y	1011001
3	0110011	J	1001010	Z	1011010

1.3.1 基本逻辑运算

逻辑代数中有与、或、非三种最基本的逻辑运算，它们可以由相应的逻辑电路实现。

1. 与逻辑

与逻辑在网络中查找资料可以体现，如查找包含“数字”和“逻辑”关键字的文章。用与的方法输入这两个关键字后，计算机仅将同时包含这两个关键字的文章搜索出来。又如在集体中选拔三好（工作好、学习好、科研好）学生，同时满足所有项，才可获得三好学生的奖励。

与逻辑的概念可用图 1.3.1(a)所示的灯开关控制电路来说明。只有当开关 A 与 B 全闭合时，灯 F 才会亮，只要 A、B 中有一个开关不闭合，灯 F 就不亮。可见，与逻辑的含意是：只有当决定一件事情的所有条件全部具备时，这件事情才会发生。

如图 1.3.1(b)所示，列出两个开关所有可能的组合，并列出对应的灯的状态。图 1.3.1(c)为其真值表。

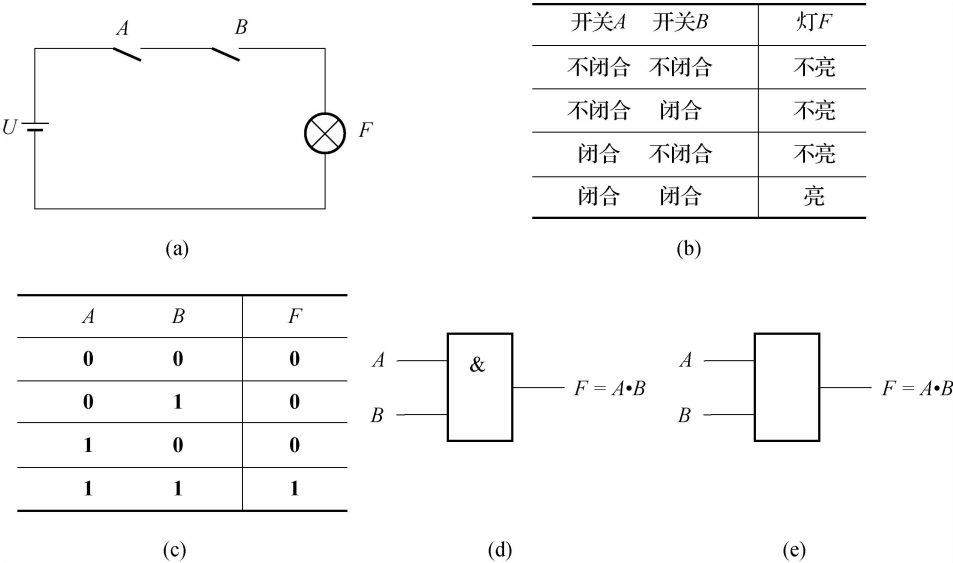


图 1.3.1 与逻辑运算
(a)电路图 (b)逻辑关系表 (c)逻辑真值表 (d)国标符号 (e)曾用符号

逻辑变量间的与逻辑运算又称逻辑乘，可用逻辑表达式表示为

$$F=A \cdot B$$

式中，“·”是与逻辑运算符，在不至于引起混淆的情况下，与运算符“·”可以省略。

在逻辑电路中，把能实现与运算的基本单元也称作与门，其国际标准逻辑符号如图 1.3.1(d)所示，曾用逻辑符号如图 1.3.1(e)所示。与运算可以推广到多变量，即

$$F=A \cdot B \cdot C \cdot \cdots$$

2. 或逻辑

或逻辑在网络中查找资料可以体现，如查找包含“数字”和“逻辑”关键字的文章。用或的方法输入这两个关键字后，计算机将包含“数字”、“逻辑”和同时包含这两个关键字的文章

都搜索出来。又如在集体中选拔单项好(工作好、学习好或科研好)的学生,只要满足其中一项,就可获得单项好的奖励。

或逻辑的概念可用图 1.3.2(a)所示的灯开关控制电路来说明。只要开关 A 、 B 中有一个或一个以上的开关闭合,灯 F 就亮。只有当开关 A 和 B 全断开时,灯 F 才不亮。可见,只要决定一件事情的所有条件中有一个或一个以上的条件具备,这件事情就会发生,即或逻辑。可以用列表的方式表示上述逻辑关系,如图 1.3.2(b)所示,左边列出两个开关所有可能的组合,右边列出对应的灯的状态。如果对逻辑变量 A 、 B 、 F 作如下定义: A 、 B 开关闭合时用 1 表示,开关断开时用 0 表示; F 为 1 时表示灯点亮, F 为 0 时表示灯不亮。将逻辑变量的取值代入图 1.3.2(b),可得到图 1.3.2(c)所示的真值表。

或逻辑又称为逻辑加,可用逻辑表达式表示为

$$F=A+B$$

式中,“+”为或逻辑运算符。

在逻辑电路中,把能实现或运算的基本单元称作或门,其国际标准逻辑符号如图 1.3.2(d)所示,曾用逻辑符号如图 1.3.2(e)所示。或逻辑也可以推广到多变量

$$F=A+B+C+\cdots$$

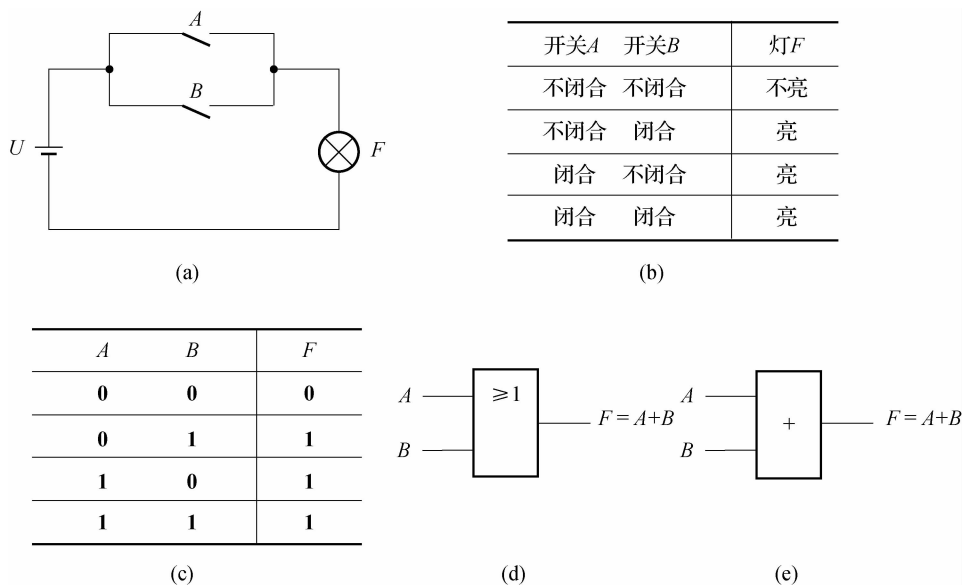


图 1.3.2 或逻辑运算

(a) 电路图 (b) 逻辑关系表 (c) 逻辑真值表 (d) 国标符号 (e) 曾用符号

3. 非逻辑

非逻辑的概念可用图 1.3.3(a)所示的灯开关控制电路来说明。当开关 A 闭合时,灯不亮;而当 A 不闭合时,灯亮。可见,逻辑非的含意是:当决定某一事件的条件满足时,事件不发生;反之事件发生。

图 1.3.3(b)和(c)分别表示非逻辑的逻辑关系表和真值表。

逻辑变量间的非逻辑运算,可用逻辑表达式表示为

$$F=\bar{A}$$

式中，“ $\bar{}$ ”为非逻辑上画线运算符，若 A 称为原变量，则 $\bar{A}$ 为其反变量，读作“ A 非”。

在逻辑电路中，把能实现非运算的基本单元称作非门，其国际标准逻辑符号如图 1.3.3(d) 所示，曾用逻辑符号如图 1.3.3(e) 所示，图中的小圆圈表示取反，非门又称为反相器。

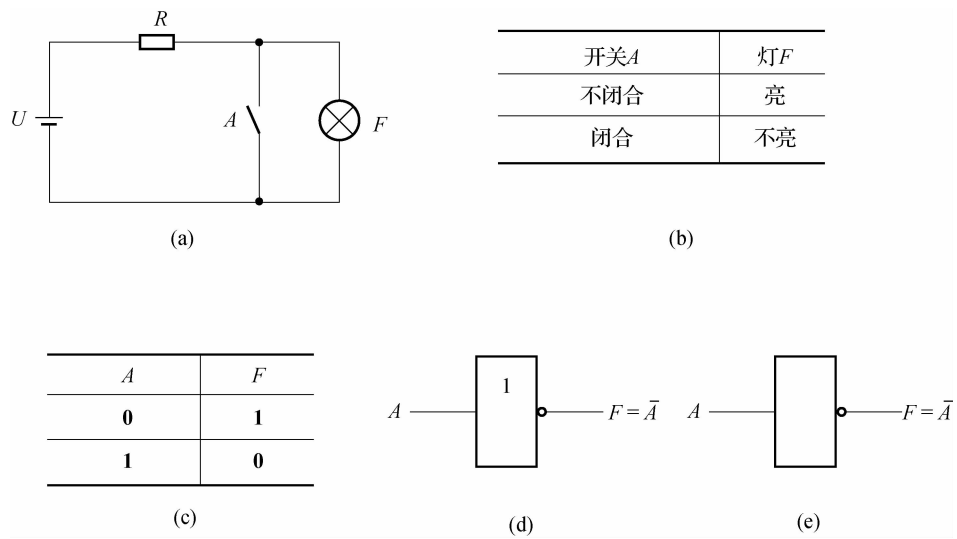


图 1.3.3 非逻辑运算

(a) 电路图 (b) 逻辑关系表 (c) 逻辑真值表 (d) 国标符号 (e) 曾用符号

1.3.2 复合逻辑运算

实际中可以用与、或、非这三种基本的逻辑运算组合来实现复杂得多的逻辑问题，最常见的复合逻辑运算有与非、或非、与或非、异或和同或等。

1. 与非逻辑

将与逻辑运算和非逻辑运算组合成与非逻辑，其逻辑表达式为 $F = \overline{AB}$ ，运算的顺序是先执行与运算，再执行非运算。实现与非运算的国际标准逻辑符号和曾用符号如图 1.3.4(a) 所示，图中的小圆圈表示非运算。

2. 或非逻辑

将或逻辑运算和非逻辑运算组合成或非逻辑，其逻辑表达式为 $F = \overline{A+B}$ ，运算的顺序是先执行或运算，再执行非运算。实现或非运算的国际标准逻辑符号和曾用符号如图 1.3.4(b) 所示。

3. 与或非逻辑

依照与、或、非运算顺序进行组合成与或非逻辑，其逻辑表达式为 $F = \overline{AB+CD}$ ，实现与或非运算的国际标准逻辑符号和曾用符号如图 1.3.4(c) 所示。

根据上述逻辑运算的含义，读者不难列出相应的逻辑真值表。

4. 异或逻辑

异或可以是多个逻辑变量，若是二变量的异或逻辑运算，当两个输入变量取值相同时，输出逻辑值为 0；当两个变量取值不同时，输出逻辑值为 1。其逻辑表达式为 $F = A \oplus B$ ，二变量异或运算的国际标准逻辑符号和曾用符号如图 1.3.4(d) 所示，逻辑真值表如图 1.3.5(a) 所示。

5. 同或逻辑

同或运算与异或运算相反,当两个变量取值相同时,输出逻辑值为 1;当两个变量取不同时,输出逻辑值为 0。其逻辑表达式为 $F=A\odot B$,同或相应逻辑门的国际标准逻辑符号和曾用符号如图 1.3.4(e)所示,逻辑真值表如图 1.3.5(b)所示。由同或运算和异或运算的逻辑含义及它们的真值表可以看出,同或和异或互为非运算,即 $A\odot B=\overline{A\oplus B}$ 。

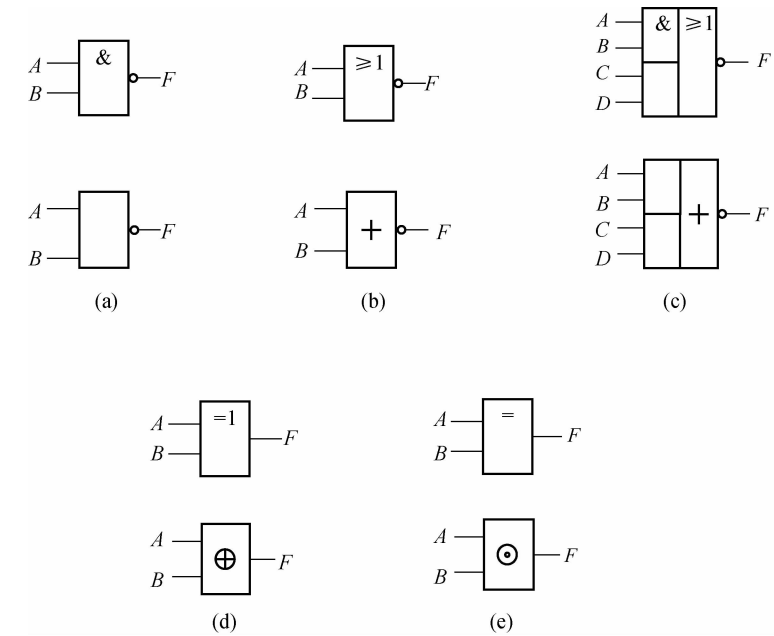


图 1.3.4 几个常用复合逻辑运算符号
(a)与非国标和曾用符号 (b)或非国标和曾用符号 (c)与或非国标和曾用符号
(d)异或国标和曾用符号 (e)同或国标和曾用符号

<i>A</i>	<i>B</i>	<i>F</i>
0	0	0
0	1	1
1	0	1
1	1	0

(a)

<i>A</i>	<i>B</i>	<i>F</i>
0	0	1
0	1	0
1	0	0
1	1	1

(b)

图 1.3.5 异或和同或真值表
(a)异或逻辑 (b)同或逻辑

1.3.3 逻辑函数及其表示方法

应用有限个与、或、非等逻辑运算符将若干个逻辑变量 A 、 B 、 C 等连接起来,所得的输入和输出的逻辑关系表达式称为逻辑函数,如

$$F(A,B)=A+B$$

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

通常 A 、 B 、 C 称为输入逻辑变量, F 称为输出逻辑函数, 当前者取值确定之后, 可以得到输出函数的逻辑值。

1. 逻辑函数的表示方法

逻辑函数可以用逻辑真值表、逻辑函数表达式、逻辑图、波形图、卡诺图和 VHDL 语言等方法表示。它们之间可以相互转换, 例如, 逻辑图可以转换为逻辑函数表达式, 逻辑表达式可以用真值表表示, 真值表可以用波形图替换等。下面用一个例子介绍这几种方法逻辑函数的建立及表现形式。

1) 真值表

以“三人表决”逻辑为例, 按“少数服从多数”的原则确立结果。将三个人的意见分别设置为逻辑变量 A 、 B 、 C , 并规定只能有同意或不同意两种意见。将表决结果设置为逻辑函数 F , 也只有“通过”与“不通过”两种情况。

对于逻辑变量 A 、 B 、 C , 设同意为逻辑 1, 不同意为逻辑 0。对于逻辑函数的结果 F , 设通过为逻辑 1, 没通过为逻辑 0。

根据“少数服从多数”的原则, 将输入变量不同取值组合与函数值间的对应关系列成表格, 得到函数的真值表如表 1.3.1 所示。

表 1.3.1 “三人表决”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

真值表的特点一目了然, 输入变量按 2^n 个状态取值并确定后, 就可在真值表中查出相应的函数值。把一个实际的逻辑问题抽象成一个逻辑函数时, 使用真值表是最方便的。在设计逻辑电路时, 通常先根据设计要求列出真值表。真值表的不足之处是当变量比较多时, 所要表达的状态按 2 的指数增加, 表格较大, 过于烦琐。

2) 逻辑函数表达式

在表 1.3.1 中依次找出函数值 F 等于 1 对应的变量组合, 变量 A 、 B 、 C 为 1 的写成原变量, 为 0 的写成反变量, 把 F 等于 1 对应的状态 A 、 B 、 C 中各个变量(包括反变量)相与, 即完成与项。然后, 把这些与项相或, 就得到相应的函数表达式。用此方法由表 1.3.1 写出“三人表决”函数的逻辑表达式为

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

反之, 由表达式也可以转换成真值表。首先画出真值表的表格, 将变量及变量的所有取值组态按照二进制递增的次序列入表格左边, 然后按照表达式, 依次对变量的各种取值组合进行运算, 求出相应的函数值, 填入表格右边对应的位置, 即可得到对应逻辑函数的真值表。

3) 逻辑图

逻辑函数表达式的运算关系用前面介绍的逻辑符号表现出来称为函数的逻辑图,逻辑图与数字电路器件有对应关系,便于构成实际数字电路。化简后的“三人表决”函数式对应的逻辑图如图 1.3.6(a)所示。

4) 波形图

反映输入和输出波形变化的图形称为波形图,又称时序图。图 1.3.6(b)是给定 A、B、C 波形后所画出的输出函数 F 的 6 种状态波形图,其余两种状态读者完成。

波形图能清晰、直观地反映出变量间的时间关系和函数值随时间变化的规律。它同实际电路中的电压波形相对应,常用于数字电路的分析和调试。

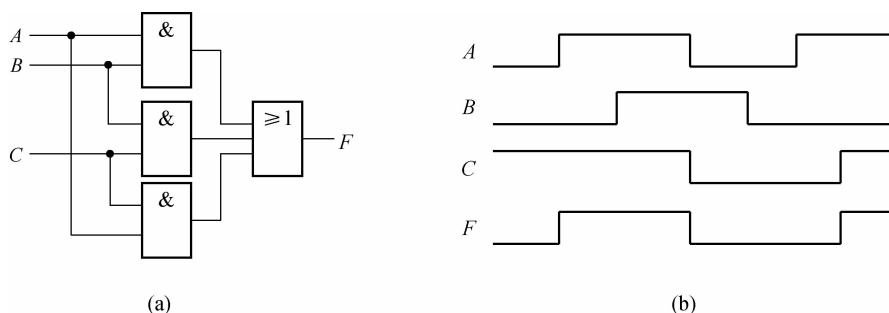


图 1.3.6 “三人表决”逻辑图与波形图

(a) 逻辑图 (b) 波形图

5) 卡诺图和 VHDL

卡诺图和 VHDL 同样可以表现“三人表决”逻辑,将在以后的内容中介绍。

2. 逻辑函数的标准形式

表示一个逻辑函数有不同的逻辑表达式,即使同一逻辑函数也会有不同的表达形式。最常见的标准式有最小项之和、最大项之积。

1) 最小项及其性质

一个 n 变量的逻辑函数中,包含全部 n 个变量的乘积项称为最小项,其中每个变量只能以原变量或反变量的形式出现一次。最小项有时也称为标准乘积项或者乘积项。

n 个变量的函数有 2^n 个最小项,可记作 $m_i, i=0 \sim (2^n-1)$,称为最小项的编号。编号的方法是:把使最小项为 1 的那一组变量取值组合当成二进制数,与这个二进制数对应的十进制数就是该最小项的编号。例如,当 $n=3$ 时,有 $2^3=8$ 个最小项。最小项 $\overline{A}B\overline{C}$ 为 1 对应的变量取值为 010,十进制数为 2,因此最小项 $\overline{A}B\overline{C}$ 的编号为 m_2 。其余最小项的编号以此类推。

为了说明最小项的性质,列出三变量全部最小项的真值表如表 1.3.2 所示。可以观察最小项具有以下几个性质。

(1) 对于任意一个最小项,只有一组变量取值使它的值为 1,而其余各组变量取值均使它的值为 0。

(2) 同一函数的任意两个不同的最小项的乘积为 0。

表 1.3.2 三变量全部最小项的真值表

变 量			m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
A	B	C	$\overline{A}\overline{B}\overline{C}$	$\overline{A}B\overline{C}$	$A\overline{B}\overline{C}$	ABC	$\overline{A}\overline{B}C$	$\overline{A}BC$	$A\overline{B}C$	ABC
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

(3)全部最小项的和为 1。

2)最大项及其性质

一个 n 变量的逻辑函数中,包含全部 n 个变量的和项称为最大项,其中每个变量只能以原变量或反变量的形式出现一次。最大项有时也称为标准和项或和项。

n 个变量的函数有 2^n 个最大项,可记作 $M_i, i=0 \sim (2^n-1)$,称为最大项的编号。最大项编号的方法是:最大项为 0 的变量取值组合当成二进制数,则该二进制数所对应的十进制数就是该最大项的编号。例如,当 $n=3$ 时,有 $2^3=8$ 个最大项。最大项 $(\overline{A}+B+\overline{C})$ 为 0 对应的变量取值为 101,十进制数为 5,因此最大项 $(\overline{A}+B+\overline{C})$ 的编号为 M_5 。其余最大项的编号以此类推。最大项也具有几条类似最小项的性质。

(1)对于任意一个最大项,只有一组变量取值使它的值为 0,而其余各种变量取值均使它的值为 1。

(2)变量相同的任意两个不同的最大项的和为 1。

(3)全部最大项之积为 0。

3)最小项和最大项之间的关系

表 1.3.3 列出了三变量函数的全部最小项和最大项,以及它们的编号。

表 1.3.3 最小项和最大项关系

A	B	C	十进制数	最小项 m_i	最大项 M_i
0	0	0	0	$\overline{A}\overline{B}\overline{C}$ m_0	$A+B+C$ M_0
0	0	1	1	$\overline{A}\overline{B}C$ m_1	$A+B+\overline{C}$ M_1
0	1	0	2	$\overline{A}B\overline{C}$ m_2	$A+\overline{B}+C$ M_2
0	1	1	3	$\overline{A}BC$ m_3	$A+\overline{B}+\overline{C}$ M_3
1	0	0	4	$A\overline{B}\overline{C}$ m_4	$\overline{A}+B+C$ M_4
1	0	1	5	$A\overline{B}C$ m_5	$\overline{A}+B+\overline{C}$ M_5
1	1	0	6	$AB\overline{C}$ m_6	$\overline{A}+\overline{B}+C$ M_6
1	1	1	7	ABC m_7	$\overline{A}+\overline{B}+\overline{C}$ M_7

(1)相同编号的最小项和最大项存在互补关系,即

$$m_i = \overline{M_i}; M_i = \overline{m_i}$$

(2)由若干个最小项之和表示的函数 F ,其反函数 $\overline{F}$ 可用与这些最小项相对应的最大项之积表示。如

$$\begin{aligned} F &= m_1 + m_3 + m_5 + m_7 \\ \overline{F} &= \overline{m_1 + m_3 + m_5 + m_7} \\ &= \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_7} \\ &= M_1 \cdot M_3 \cdot M_5 \cdot M_7 \end{aligned}$$

4)标准形式

逻辑函数的标准积之和表达式、标准之和之积表达式和真值表一样具有唯一性。

若函数的积之和(与或)表达式中的每一个乘积项均为最小项,则这种表达式称为标准积之和表达式,也称最小项表达式。例如

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

上式中各乘积项均是最小项,因此为最小项表达式。该最小项表达式也可写成

$$F(A, B, C, D) = m_0 + m_1 + m_5 + m_8 = \sum m(0, 1, 5, 8)$$

常见的逻辑函数表达形式为非标准表达式,如果将非标准表达式变换成标准积之和表达式可遵循下面的步骤:首先将函数转换成积之和表达式,对非最小项的积项,利用互补性质($A + \overline{A} = 1$)乘以该乘积项缺少的变量,既可使积项逻辑值不变,又可在该积项增补变量,多次利用互补性质,直到每个积项均为最小项为止。一个函数的最小项表达式与函数真值表一一对应,具有唯一性质。

例 1.3.1 求函数 $F = \overline{A}\overline{B}\overline{C}D + \overline{A}CD + AC$ 的标准积之和表达式。

$$\begin{aligned} \text{解: } F &= \overline{A}\overline{B}\overline{C}D + \overline{A}(B + \overline{B})CD + A(B + \overline{B})C \\ &= \overline{A}\overline{B}\overline{C}D + \overline{A}BCD + \overline{A}\overline{B}CD + ABC(D + \overline{D}) + \overline{A}BC(D + \overline{D}) \\ &= \overline{A}\overline{B}\overline{C}D + \overline{A}BCD + \overline{A}\overline{B}CD + ABCD + ABC\overline{D} + \overline{A}BCD + \overline{A}\overline{B}CD \\ &= m_3 + m_7 + m_9 + m_{10} + m_{11} + m_{14} + m_{15} \\ &= \sum m(3, 7, 9, 10, 11, 14, 15) \end{aligned}$$

同一个函数的两种标准形式所含最小项和最大项的编号互不重复,而又互补。不难证明,如果一个函数的最小项表达式为 $F = \sum m_i$,最大项表达式为 $F = \prod M_k$,其中 m_i 是编号为 i 的全部最小项,则 M_k 是不包含在 i 内的所有其他编号的最大项。两种标准表达式之间可以互相转换。

3. 逻辑函数公式、规则和常用表达形式

1)基本逻辑函数和异或运算公式

逻辑函数的形式转换与化简和普通函数一样可以通过其公式和规则进行,逻辑函数的基本公式见表 1.3.4,主要包括 9 个定律,即交换律、结合律、分配律、互补律、0-1 律、还原律、重叠律、吸收律和反演律。另外,还包括由基本公式衍生出的异或运算的基本公式。其中有的定律与普通代数相似,有的定律与普通代数完全不同,使用时切勿混淆。

基本公式中的互补律、0-1 律、还原律等简单公式可根据与、或、非三种基本逻辑运算法则推导出来。较为复杂的公式可用其他更简单的公式来证明。还可以用真值表来证明,即

检验等式两边函数的真值表是否一致。

表 1.3.4 逻辑代数的基本公式和异或运算公式

名 称	公 式 1	公 式 2
0-1 律	$A \cdot 1 = A$ $A \cdot 0 = 0$	$A + 0 = A$ $A + 1 = 1$
互补律	$A\bar{A} = 0$	$A + \bar{A} = 1$
重叠律	$AA = A$	$A + A = A$
交换律	$AB = BA$	$A + B = B + A$
结合律	$A(BC) = (AB)C$	$A + (B + C) = (A + B) + C$
分配律	$A(B + C) = AB + AC$	$A + B \cdot C = (A + B) \cdot (A + C)$
反演律	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$
吸收律	$A(A + B) = A$ $A(\bar{A} + B) = AB$ $(A + B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C)$	$A + AB = A$ $A + \bar{A}B = A + B$ $AB + \bar{A}C + BC = AB + \bar{A}C$
还原律	$\bar{\bar{A}} = A$	
交换律	$A \oplus B = B \oplus A$	
结合律	$(A \oplus B) \oplus C = A \oplus (B \oplus C)$	
分配律	$A(B \oplus C) = AB \oplus AC$	
变量常量运算	$A \oplus 1 = \bar{A}; A \oplus 0 = A; A \oplus A = 0; A \oplus \bar{A} = 1$	
因果互换关系	若 $A \oplus B = C$, 则 $A \oplus C = B, B \oplus C = A$	
多变量运算	在多变量异或运算中, 如果为 1 的变量个数为奇数, 则结果为 1, 否则结果为 0, 与为 0 的变量个数无关	

例 1.3.2 证明吸收律 $A + \bar{A}B = A + B$ 。

证: 利用互补律增加与项 $B + \bar{B}$, 再利用分配律和重叠律进行变换, 有

$$\begin{aligned} A + \bar{A}B &= A(B + \bar{B}) + \bar{A}B = AB + A\bar{B} + \bar{A}B = AB + AB + A\bar{B} + \bar{A}B \\ &= A(B + \bar{B}) + B(A + \bar{A}) = A + B \end{aligned}$$

例 1.3.3 证明反演律 $\overline{AB} = \bar{A} + \bar{B}$ 和 $\overline{A + B} = \bar{A}\bar{B}$ 。

证: 用真值表 1.3.5, 分别列出两公式等号两边函数的逻辑值可以证明反演律。由于等式两列所列出的逻辑值是相等的, 所以两式成立。

表 1.3.5 真值表证明反演律

A	B	$\overline{AB}$	$\overline{A+B}$	$\overline{A+B}$	$\overline{AB}$
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0
		证明 $\overline{AB}=\overline{A+B}$		证明 $\overline{A+B}=\overline{AB}$	

反演律又称摩根定律,是非常重要的公式,它经常用于逻辑函数的变换,以下是它的两个变形公式,也是常用的。

$$AB=\overline{\overline{A+B}}; \quad A+B=\overline{\overline{AB}}$$

2)三个基本运算规则

(1)代入规则。在任何含有变量 B 的逻辑等式中,等式两边所有出现 B 的地方用另一个逻辑函数 BC 代替,则逻辑等式依然成立。

例 1.3.4 用反演律 $\overline{AB}=\overline{A+B}$ 证明 $\overline{ABC}=\overline{A+BC}=\overline{A+B+C}$ 成立。

证:利用代入规则可以方便地扩展公式,在反演律 $\overline{AB}=\overline{A+B}$ 中用 BC 代替等式中的 B ,则新的等式仍成立

$$\overline{ABC}=\overline{A+BC}=\overline{A+B+C}$$

由此例可知,利用代入规则,反演律可以推广到 n 个变量,即

$$\begin{aligned} \overline{A_1 \cdot A_2 \cdot \dots \cdot A_n} &= \overline{A_1 + A_2 + \dots + A_n} \\ \overline{A_1 + A_2 + \dots + A_n} &= \overline{A_1} \cdot \overline{A_2} \cdot \dots \cdot \overline{A_n} \end{aligned}$$

(2)反演规则。对于任意一个逻辑函数式 F ,若将式中的运算符“ $\cdot$ ”换成“ $+$ ”,“ $+$ ”换成“ $\cdot$ ”;把常量“0”换成“1”,“1”换成“0”;原变量换成反变量,反变量换成原变量,则得到的新函数式是原函数式 F 的反函数 $\overline{F}$ 。 $\overline{F}$ 为 F 的反函数,这个规则称为反演规则。

例 1.3.5 利用反演规则,求 $F=A\overline{B}+(\overline{A+C})\overline{B}+\overline{A}B\overline{C}$ 的反函数 $\overline{F}$ 。

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

式中 $\overline{A+C}\overline{B}$ 不考虑非号时,以 $(A+C)B$ 为一函数,按反演规则变换。

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

式中 $\overline{A+C}\overline{B}$ 考虑非号时,以 $(A+C)B$ 为一变量,去掉非号就可以了。

反演规则实际上是反演律及代入规则的推广应用。在运用反演规则时需注意运算的顺序,必须保持原函数的运算顺序,必要时加入括号。

(3)对偶规则。对于任意一个逻辑函数式 F ,若将式中的运算符“ $\cdot$ ”换成“ $+$ ”,“ $+$ ”换成“ $\cdot$ ”;常量“0”换成“1”,“1”换成“0”,得到的新函数式是原函数式 F 的对偶式 F' ,也称对偶函数。应用对偶函数时同样需注意保持原式中的运算顺序不变,并且只变换运算符和常量,变量不需要变换。

例如, $F = \overline{A+B} \cdot \overline{BC} + AB$ 则 $F' = \overline{AB+B+C} \cdot (A+B)$

需要注意的是,一般情况下 $\overline{F} \neq F'$ 。对偶规则的基本性质是:如果两个函数式相等,则它们相应的对偶式也相等。

3) 逻辑函数常用表达形式

(1) 五种常用表达式。常见的逻辑表达式主要有五种形式,即“与—或”、“或—与”、“与非—与非”、“或非—或非”和“与—或非”表达式。

$$\begin{aligned} F &= AB + \overline{AC} && \text{“与—或”式} \\ &= (A+C)(\overline{A}+B) && \text{“或—与”式} \\ &= \overline{AB} \cdot \overline{AC} && \text{“与非—与非”式} \\ &= \overline{A+C} + \overline{A+B} && \text{“或非—或非”式} \\ &= \overline{A} \cdot \overline{C} + A \cdot \overline{B} && \text{“与—或非”式} \end{aligned}$$

上式中“与—或”表达式和“或—与”表达式是逻辑函数的基本表示形式。其表达式便于与其他形式相互转换,并且易于在后面介绍的卡诺图上表示,逻辑函数的公式均以这两种基本形式出现。

(2) 表达式之间的转换。以逻辑函数 $F = AB + \overline{AC}$ 为例说明几种表达式之间是如何转换的。

例 1.3.6 将“与—或”式转换为“或—与”式。

解:利用互补律和吸收律在“与—或”式 $F = AB + \overline{AC}$ 中添加 $A\overline{A}$ 和 BC 两项,再用分配律,有

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

例 1.3.7 将“与—或”式转换为“与非—与非”式。

解:利用还原律对“与—或”式 $F = AB + \overline{AC}$ 两次取反,再利用反演律变换,有

$$F = AB + \overline{AC} = \overline{\overline{AB} + \overline{\overline{AC}}} = \overline{\overline{AB} \cdot \overline{AC}}$$

例 1.3.8 将“或—与”式转换为“或非—或非”式。

解:利用还原律对“或—与”式 $F = (A+C)(\overline{A}+B)$ 两次取反,再利用反演律变换,有

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

例 1.3.9 将“或—与”式转换为“与—或非”式。

解:利用还原律对“或—与”式 $F = (A+C)(\overline{A}+B)$ 两次取反,再两次利用反演律变换,有

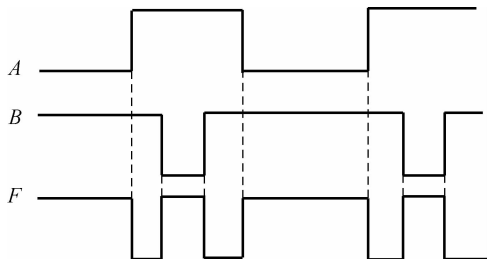
$$F = (A+C)(\overline{A}+B) = \overline{\overline{(A+C)(\overline{A}+B)}} = \overline{\overline{A+C} + \overline{\overline{A}+B}} = \overline{\overline{AC} + AB}$$

可见一个逻辑函数可以根据设计者的需要构成上述任何一种形式,不同形式由于其简化程度和构成项不同,其表达形式也不尽相同。

思考题

题 1.3.1 对于如思考题图 1.1 所示的波形, A 、 B 为输入, F 为输出, 其反映的逻辑关系是_____。

- (A) 与非关系 (B) 异或关系 (C) 同或关系 (D) 或关系
(E) 无法判断



思考题图 1.1

题 1.3.2 信号 A 和 0 异或相当于_____门, 信号 A 和 1 异或相当于_____门。

题 1.3.3 连续异或 $(1 \oplus 1 \oplus 1 \oplus 1 \oplus \dots)$ 1985 个 1 的结果是_____, 单数个 1 连续异或、双数个 1 连续异或的结果分别是_____、_____。

题 1.3.4 已知逻辑函数 $F = A(B + DC)$, 下列肯定可以使 $F = 1$ 的是_____。

- (A) $A=0, BC=0, D=0$ (B) $A=0, BD=0, C=0$
(C) $AB=1, C=0, D=0$ (D) $AC=1, B=0$

题 1.3.5 指出下列各式中哪些是四变量 A 、 B 、 C 、 D 的最小项和最大项。在最小项后的括号里填 m , 在最大项后的括号里填 M , 其他填 $\times$ 。

- (1) $A + \bar{B} + \bar{D}$ () (2) $\bar{A}\bar{B}\bar{C}\bar{D}$ ()
(3) ABD () (4) $AB(C + \bar{D})$ ()
(5) $\bar{A} + B + C + D$ () (6) $A + B + C\bar{D}$ ()

题 1.3.6 最小项 $ABCD$ 的逻辑相邻项是_____。

- (A) $AB\bar{C}\bar{D}$ (B) $ABC\bar{D}$ (C) $A\bar{B}CD$ (D) $\bar{A}B\bar{C}D$

题 1.3.7 某一逻辑函数真值确定后, 下面描述该函数功能的方法中, 具有唯一性的是_____。

- (A) 逻辑函数的最简与或式 (B) 逻辑函数的最小项之和表达式
(C) 逻辑函数的最简或与式 (D) 逻辑函数的最大项之和表达式

题 1.3.8 利用反演规则, 求出 $F = A \oplus B \oplus C \oplus \dots$ 函数的逻辑表达式为_____。

- (A) 0 (B) 1
(C) 不唯一 (D) 同或

1.4 逻辑函数的化简

1.4.1 化简概念

同一个逻辑函数可以写成与非、或非、与或非等不同形式的逻辑表达式,即使是同种形式其繁简程度也不尽相同。简洁的逻辑表达式不仅逻辑关系明显,而且实现该逻辑函数所需要的器件较少,具有电路简单、成本低、电路可靠性高等特点。因此,在采用小规模集成电路设计电路时,经常需要通过一定的手段对逻辑函数进行化简。

化简的目的是使逻辑电路所用的逻辑门数量少、每个门的输入端个数少、逻辑电路的级数少,并且要保证电路可靠地工作。常用的逻辑表达式是“与—或”式,化简“与—或”式也比较方便。常用的“与—或”表达式的“最简”的标准有以下两条。

(1)与项最少,即表达式中“+”号最少,可以使电路实现时所用的逻辑门的个数最少。

(2)每个与项中的变量数最少,即表达式中“ $\cdot$ ”号最少,可以使电路所用逻辑门的输入端个数最少。

由于逻辑函数表达式的类型多种多样,因此,“最简”的标准也有所不同。如最简的“与—或”表达式转换成其他的表达形式就可能不是最简的。

1.4.2 代数法化简逻辑函数

代数法化简是直接利用逻辑代数的基本公式和基本规则化简逻辑函数,化简的过程就是不断地用等式变换的方法消去函数式中多余的乘积项和多余的因子,达到函数最简。由于“与—或”式的逻辑形式与人们逻辑思维相吻合,下面以“与—或”式的形式举例说明逻辑函数的化简。

1. 并项法

运用互补律 $AB + \overline{A}B = B$,可以消去一个变量 A ,将两项合并为一项。

例 1.4.1 试用并项法化简逻辑函数

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

解: $F = ABC\overline{C} + ABC + A\overline{B} = AB + A\overline{B} = A$

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

2. 吸收法

运用吸收律 $A + AB = A$ 消去多余的与项。

例 1.4.2 试用吸收法化简逻辑函数

$$F = A\overline{B} + A\overline{B}C + A\overline{B}DE \text{ 和 } F = \overline{A}B + \overline{A}D + \overline{B}E.$$

解: $F = A\overline{B} + A\overline{B}C + A\overline{B}DE = A\overline{B} + A\overline{B}(C + DE) = A\overline{B}$

$$F = \overline{A}B + \overline{A}D + \overline{B}E = \overline{A} + \overline{B} + \overline{A}D + \overline{B}E = \overline{A} + \overline{B}$$

3. 消元法

运用消因律将 $A + \overline{A}B = A + B$ 中消去多余因子 $\overline{A}$ 。

例 1.4.3 试用消元法化简逻辑函数

$$F=AB+\overline{A}C+\overline{B}C \text{ 和 } F=\overline{A}+AB+\overline{B}E。$$

$$\text{解: } F=AB+\overline{A}C+\overline{B}C=AB+(\overline{A}+\overline{B})C=AB+\overline{A}\overline{B}C=AB+C$$

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

4. 配项法

在逻辑表达式中先通过乘以 $A+\overline{A}$ 或加上 $A\overline{A}$, 或重复写入某一项, 再与其他项合并, 以获得更简单的化简结果。

例 1.4.4 试用配项法化简逻辑函数

$$F=AB+\overline{A}C+BCD \text{ 和 } F=AB\overline{C}+\overline{A}BC\cdot\overline{A}B。$$

$$\text{解: } F=AB+\overline{A}C+BCD=AB+\overline{A}C+BCD(A+\overline{A})$$

$$=AB+\overline{A}C+ABCD+\overline{A}BCD=AB+\overline{A}C$$

$$F=AB\overline{C}+\overline{A}BC\cdot\overline{A}B=AB\overline{C}+\overline{A}BC\cdot\overline{A}B+AB\cdot\overline{A}B=AB(\overline{C}+\overline{A}B)+\overline{A}BC\cdot\overline{A}B$$

$$=AB\cdot\overline{A}BC+\overline{A}BC\cdot\overline{A}B=\overline{A}BC(AB+\overline{A}B)=\overline{A}BC$$

在化简“或一与”逻辑函数时也可像“与一或”逻辑表达式那样化简, 化简过程同样依据基本公式与基本规则。或者利用对偶规则将“或一与”式变为“与一或”式进行化简, 再将简化后的“与一或”式用对偶式还原为“或一与”式。

通过以上讨论可见, 利用代数法化简逻辑函数对函数变量数目无限制, 方法灵活、技巧性强, 并且无一定步骤可遵循。能否得到满意的结果, 主要取决于设计者对公式的熟练掌握程度、综合应用能力和实践经验。代数法化简是其他化简方法的理论依据, 美国工程师卡诺 (Karnaugh) 发明的图解法比代数法更为简便、直观, 具有较强的规律性。

1.4.3 图解法化简逻辑函数

图解法又称卡诺图法, 比较容易掌握, 不需要记忆大量的公式, 不存在采用何种简化路径的问题。只要按照规定的步骤与方法, 最终均可得到最简逻辑表达式。一般运用于五变量以下的逻辑函数简化。

1. 卡诺图

如果两个最小项中只有一个变量不同, 则称这两个最小项为逻辑相邻, 简称相邻项。

若将 n 变量的全部 2^n 个最小项用一个小方格表示, 并使具有逻辑相邻性的最小项在几何位置上也相邻地排列起来, 则所得的图形称为 n 变量卡诺图。二变量至五变量卡诺图如图 1.4.1(a)~1.4.1(d) 所示。

卡诺图实际上是真值表的另一种表现形式, 一个逻辑函数的真值表有多少行, 卡诺图就有多少个小方格。所不同的是真值表是一维的, 其中最小项是按照二进制加法规律排列的。而卡诺图是二维的, 它将自变量分为两组, 一组水平排列, 另一组垂直排列, 组合表示的最小项是按照相邻性规律排列的。

仔细观察可以发现, 卡诺图具有以下特点。

(1) 对于含有 n 个逻辑变量的函数, 其卡诺图内含有 2^n 个小方格, 分别对应 2^n 个最小项。

(2) 卡诺图中行、列两组变量取值按循环码规律排列, 以保证几何位置上相邻的小方格其对应的最小项为逻辑相邻项。

(3) 卡诺图中相邻的情况有两种, 一种是小方格在几何位置上邻接; 另一种是小方格以

上下或左右的中心线为轴对称。如图 1.4.1(c)中 m_1 和 m_3 是邻接;图 1.4.1(c)中 m_1 和 m_9 ,图 1.4.1(d)中 m_9 和 m_{13} 则为对称。

需要说明的是变量在行、列中的分组不同,卡诺图的表现形式也不同。在进行分组时,最好按变量字符的顺序划分。如四变量最好以 AB 、 CD 分组,不宜分为 AC 、 BD 。

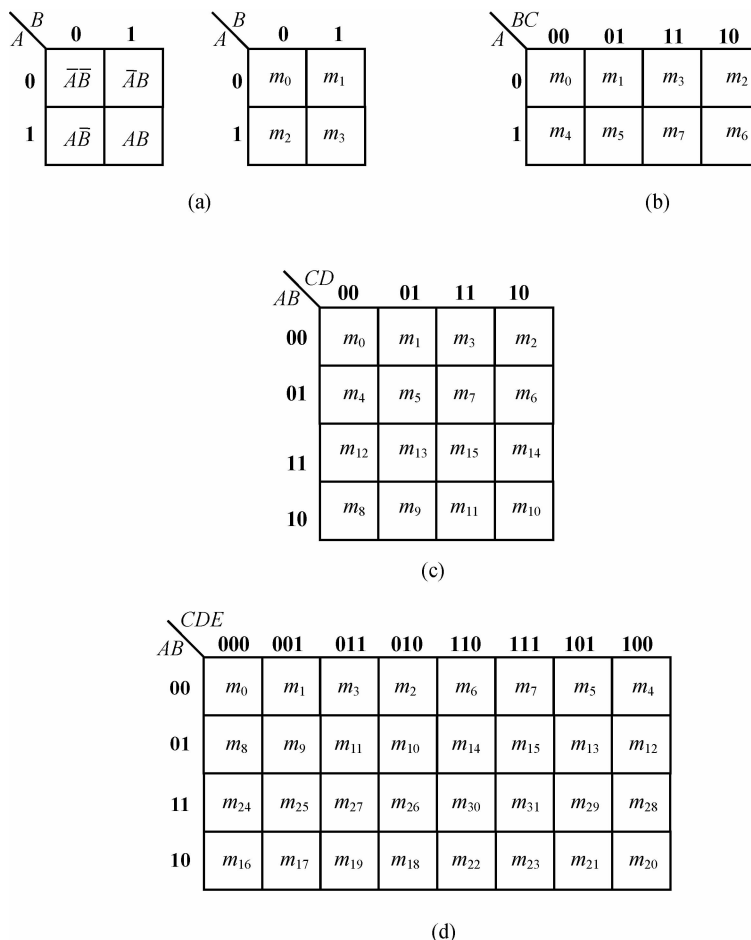


图 1.4.1 二变量至五变量卡诺图

(a)二变量卡诺图 (b)三变量卡诺图 (c)四变量卡诺图 (d)五变量卡诺图

2. 用卡诺图表示逻辑函数

1) 从真值表到卡诺图

由于任意一个 n 变量的逻辑函数都可以表示为最小项之和的形式,而 n 变量的卡诺图包含了 n 变量的所有最小项,所以可以用 n 变量卡诺图表示任意一个 n 变量的逻辑函数。方法是在逻辑函数 F 的真值表中,将各行的取值填入卡诺图中对应的小方格,可得到逻辑函数 F 的卡诺图。

例 1.4.5 某逻辑函数的真值表如表 1.4.1 所示,用卡诺图表示该逻辑函数。

解: 该函数为三变量函数,先画出三变量卡诺图,然后根据表 1.4.1 将 8 个最小项的取值 0 或 1 填入卡诺图中对应的 8 个小方格中,如图 1.4.2 所示。

表 1.4.1 真值表

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

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

图 1.4.2 例 1.4.5 的卡诺图

2) 从逻辑表达式到卡诺图

(1) 如果逻辑表达式为最小项表达式, 把函数式中出现的最小项对应到卡诺图的小方格。这些小方格中填入 1, 其他的小方格中填入 0。

例 1.4.6 用卡诺图表示逻辑函数 $F = \overline{A}\overline{B}\overline{C} + \overline{A}BC + AB\overline{C} + ABC$ 。

解: (1) 该函数为三变量函数, 且为最小项表达式, 写成简化形式 $F = m_0 + m_3 + m_6 + m_7$, 然后画出三变量卡诺图(见图 1.4.3), 将卡诺图中 m_0 、 m_3 、 m_6 、 m_7 对应的小方格填 1, 其他小方格填 0。

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

图 1.4.3 例 1.4.6 的卡诺图

(2) 如果逻辑表达式是“与—或”表达式, 而且不是最小项表达式, 可将其先化成最小项表达式, 再填入卡诺图。也可根据表达式直接填入, 即分别找出每一个与项所包含的所有小方格, 在这些小方格中全部填入 1。

(3) 如果逻辑表达式不是“与—或”表达式, 可先将其化成“与—或”表达式再填入卡诺图。

3. 用卡诺图化简逻辑函数

1) 卡诺图化简逻辑函数的依据

如果两个相邻最小项出现在同一个逻辑函数中, 可以合并为一项, 同时消去互为反变量的那个变量。如

$$ABC + A\overline{B}C = AC(B + \overline{B}) = AC$$

由此可见, 利用相邻项的合并可以进行逻辑函数化简。

卡诺图中任意两个相邻的小方格取值为 1 时, 它们代表的最小项为相邻项, 可以合并为 1 项, 消去取值不同的那个变量。不难证明, 4 个、8 个以至 2^n 个最小项相邻时也可以合并为 1 项, 消去 2 个、3 个以至 n 个变量。

卡诺图中合并最小项的规律有以下几点。

(1) 2 个相邻的最小项合并(用一个包围圈表示), 消去 1 个取值不同的变量而合并为 1 项, 如图 1.4.4 所示。

(2) 4个相邻的最小项合并(用一个包围圈表示),消去2个取值不同的变量而合并为1项,如图1.4.5所示。

(3) 8个相邻的最小项合并(用一个包围圈表示),消去3个取值不同的变量而合并为1项,如图1.4.6所示。

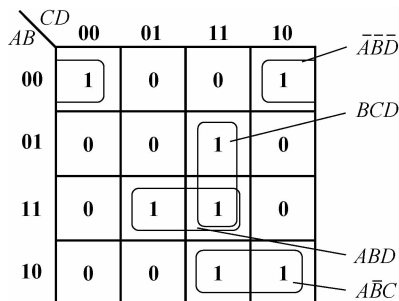


图 1.4.4 2个相邻的最小项合并

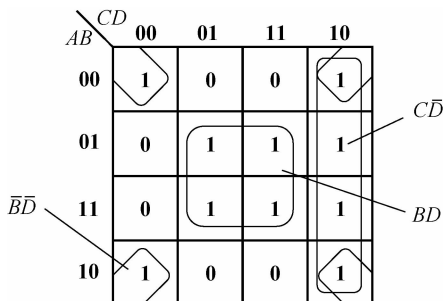


图 1.4.5 4个相邻的最小项合并

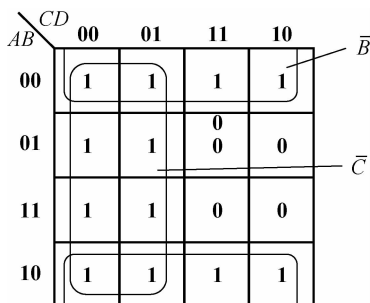


图 1.4.6 8个相邻的最小项合并

2) 卡诺图合并最小项的原则

用卡诺图化简逻辑函数时,在卡诺图中找到相邻的最小项,并画上圈。为了保证将逻辑函数化到最简,画圈时必须遵循以下原则。

(1) 圈要尽可能大,使得消去的变量数最多。但每个圈内只能含有 2^n ($n=0,1,2,3,\dots$) 个相邻项。

(2) 圈的个数尽量少,使得化简后的逻辑函数的与项最少。

(3) 卡诺图中所有取值为1的方格均要被圈过,不能漏掉取值为1的最小项。

(4) 取值为1的方格可以被重复圈在不同的包围圈中,但在新画的包围圈中至少要含有1个未被圈过的1方格,否则该包围圈是多余的,通常称作冗余项。

3) 卡诺图将逻辑函数化简为最简“与—或”表达式

将逻辑函数化为最简“与—或”表达式的步骤如下。

(1) 由真值表或函数表达式画出逻辑函数的卡诺图。

(2) 合并相邻的最小项,即根据前述原则画圈。

(3) 写出化简后的表达式。每一个圈写一个最简与项,写法是:取值为1的变量用原变量表示,取值为0的变量用反变量表示,将这些变量相与。然后将所有与项进行逻辑或,即得最简“与—或”表达式。

例 1.4.7 用卡诺图化简逻辑函数。

$$F(A,B,C,D) = \sum m(0,2,3,4,6,7,10,11,13,14,15)$$

解:由表达式画出卡诺图,如图 1.4.7 所示。画圈合并最小项,得化简的“与一或”表达式

$$F = C + \overline{A}\overline{D} + ABD$$

例 1.4.8 用卡诺图化简逻辑函数 $F = \overline{B}C + B\overline{C} + \overline{A}C + A\overline{C}$ 。

解:由逻辑函数画出卡诺图,如图 1.4.8 所示。画包围圈合并最小项,如图 1.4.8(a)、图 1.4.8(b)所示,得到简化程度相同的两个“与一或”表达式。

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

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

通过这个例子可以说明,一个逻辑函数的化简结果不是唯一的。

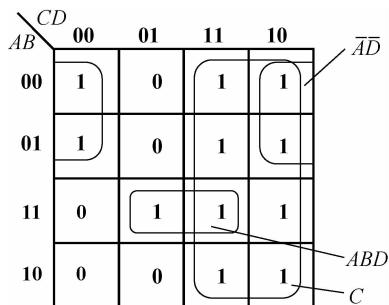
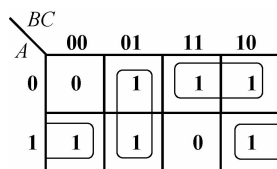
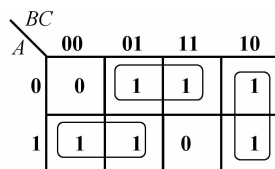


图 1.4.7 例 1.4.7 卡诺图



(a)



(b)

图 1.4.8 例 1.4.8 卡诺图

(a)解法 1 (b)解法 2

4)卡诺图将逻辑函数化为最简“或一与”表达式

可以根据前述的化简方法得到 $\overline{F}$ 的最简“与一或”表达式,然后根据逻辑函数的反演规则,得到 F 的最简“或一与”表达式。

也可以根据函数的卡诺图,直接写出函数的最简“或一与”表达式:在卡诺图上对所有的 0 画圈,画圈的原则与前述画 1 的原则相同。每一个圈写一个最简或项,取值为 0 的变量用原变量表示,取值为 1 的变量用反变量表示,将这些变量相或。然后将所有或项进行逻辑与得到最简“或一与”表达式。

例 1.4.9 用卡诺图将逻辑函数 F 化为最简“或一与”表达式。

$$F(A,B,C,D) = \sum m(6,7,8,9,12,13)$$

解:由表达式画出卡诺图,如图 1.4.9 所示。

对 0 画圈,直接写出最简“或一与”表达式

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

4. 具有无关项的逻辑函数的化简

1) 无关项

约束项或任意项的最小项统称为无关项,常用符号“ $\times$ ”或“ Φ ”表示其逻辑值。前面讨论的逻辑函数,对于自变量的所有取值都有一个确定的函数值与之对应,不是 0 就是 1。在许多实际问题中,经常会遇到这种情况,实际问题抽象为逻辑函数后,输入逻辑变量的某些取值组合

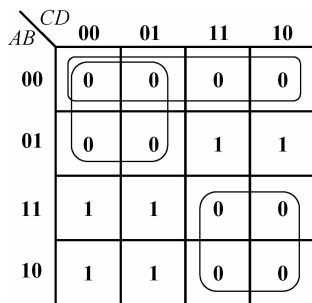


图 1.4.9 例 1.4.9 卡诺图

禁止出现,或者一些取值组合出现时,输出逻辑值可以是任意的,不影响电路的正确逻辑。

例如,有三个逻辑变量 A 、 B 、 C ,分别表示计算器的加法、减法、乘法三种操作。因为计算器是按顺序逐条执行命令的,每次只能执行一种操作,所以不允许两个以上操作同时进行,即不允许两个以上变量同时为 1。也就是说, ABC 的取值只可能是 000、001、010、100 中的某一种,而不可能是 011、101、110、111 中的任何一种。这些不会出现的变量取值组合所对应的最小项为约束项。

有时也会遇到另一种情况,在输入变量的某些组合情况下,函数值是 1 还是 0 皆可,不影响电路的逻辑功能,这些变量取值组合所对应的最小项为任意项。

例 1.4.10 有一标注三个水位的储水箱,如图 1.4.10 所示。当水位高于 A ,则 $A=1$,否则 $A=0$;当水位高于 B ,则 $B=1$,否则 $B=0$;当水位高于 C ,则 $C=1$,否则 $C=0$ 。试列出可能的各种组态 ABC 的值,写出约束项和任意项。

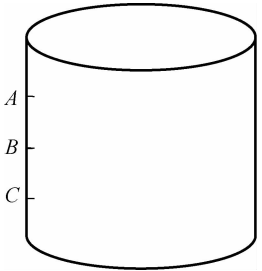


图 1.4.10 例 1.4.10 图

解: ABC 共有 8 种组态,只有 4 种可能组态,即当水位低于 C 点时, ABC 的值是 000,当水位高于 A 点时, ABC 的值是 111,在 B 和 C 之间时, ABC 的值是 001,在 A 和 B 之间时, ABC 的值是 011。

再讨论 110、101、010 和 100,它们分别是以下几种情况。

- (1)水位高于 A 和 B ,而低于 C ,不可能出现,属于约束项。
- (2)水位高于 A 和 C ,而低于 B ,不可能出现,属于约束项。
- (3)水位高于 B ,而低于 C ,不可能出现,属于约束项。
- (4)水位高于 A ,而低于 B 和 C ,不可能出现,属于约束项。

共有 4 个约束项,没有任意项。

又如十字路口的红绿两色交通信号灯,规定红灯停,绿灯行,而红绿灯全亮是被禁止的,两灯全不亮则表示可行可停,依情况而定。如果红、绿灯分别用 A 、 B 表示,且灯亮为 1,灯灭为 0。车用 F 表示,车行为 1,车停为 0。不难列出该函数的真值表如表 1.4.2 所示。

表 1.4.2 交通信号真值表

红灯 A	绿灯 B	车 F
0	0	×
0	1	1
1	0	0
1	1	×

其中, A 、 B 全为 1,为约束项; A 、 B 全为 0 是任意项,函数值用“ $\times$ ”表示。带有无关项的逻辑函数的最小项表达式可以写为

$$F = \sum m(\quad) + \sum d(\quad); \text{或者 } F = \sum m(\quad) + \sum \phi(\quad)$$

上面的交通信号函数可写成 $F = \sum m(1) + \sum d(0,3)$ 。

2) 利用无关项化简逻辑函数

化简具有无关项的逻辑函数时,要充分利用无关项可以当 0 也可以当 1 的特点,尽量扩大卡诺圈,使逻辑函数最简。在考虑无关项时,哪些无关项当作 1,哪些无关项当作 0,要以

尽量扩大卡诺圈、减少圈的个数,使逻辑函数最简为原则。

例 1.4.11 化简 $F(A,B,C,D) = \sum m(3,6,9,11,13) + \sum d(1,2,5,7,8,15)$ 。

解:画出 4 变量卡诺图,将最小项 1 和无关项“×”填入卡诺图如图 1.4.11 所示。合并最小项。与 1 方格圈在一起的无关项被当作 1,没有圈的无关项作为 0。

写出逻辑函数的最简“与-或”表达式

$$F = D + \bar{A}C$$

卡诺图化简法的优点是简单、直观,有一定的化简步骤可循,不易出错,且容易化到最简。但是在逻辑变量超过 5 个时,就失去了简单、直观的优点,不宜采用。

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

图 1.4.11 例 1.4.11 卡诺图

思考题

题 1.4.1 卡诺图中的逻辑相邻或对称相邻具有_____特征,其数值不同只是在位上差_____位。

- (A) 余 3 码 2 (B) 8421 码 3
(C) 循环码 2 (D) 格雷码 1

题 1.4.2 在思考题图 1.2 所示的卡诺图中,化简后的逻辑函数是_____。

- (A) $AB + BC + CA$ (B) $\bar{A}\bar{B} + \bar{B}\bar{C} + \bar{C}\bar{A}$ (C) $A\bar{B} + B\bar{C} + C\bar{A}$ (D) $\bar{A}\bar{B} + \bar{B}\bar{C} + \bar{C}\bar{A}$

题 1.4.3 任意项和约束项有微小的区别,区别在于任意项值_____,约束项值_____。约束项和任意项统称为_____。

题 1.4.4 有三个逻辑变量 A 、 B 、 C ,它们分别表示一台电动机的正转、反转和停止的命令, $A=1$ 表示正转, $B=1$ 表示反转, $C=1$ 表示停止。电动机任何时候只能执行一个命令,请写出描述上述情况的约束项逻辑表达式。

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

思考题图 1.2

1.5 硬件描述语言 HDL 基础

20 世纪末,集成电路硬件描述语言(Hardware Description Language, HDL)有很多,主要有 VHDL、Verilog HDL、AHDL、ABEL 等,还有众多软件公司研制开发的具有自己特色的电路硬件描述语言和软件平台。这些硬件描述语言必然有很大的差异,一旦选用某种语言作为输入工具,就被束缚在这个设计环境之中,不能在众多的软件工具中选择一个最佳组合作为自己的最优设计环境。直到 20 世纪 80 年代初,美国国防部提出了 VHSIC(very high speed integrated circuit)计划,其目标之一是为下一代集成电路的生产,实现阶段性的工艺极限以及完成 10 万门级以上的设计而研制一种新的描述方法。1981 年提出了一种新的 HDL,称之为 VHSIC hardware description language,简称为 VHDL。VHDL 语言能成功地描述超大规模的电路系统,1993 年由 IEEE 确认 VHDL 为集成电路硬件描述语言的标准。

准。此后 VHDL 在电子领域得到了广泛的接受,并逐步取代了原有非标准的硬件描述语言,作为可相互交流的设计环境。

Verilog HDL 语言是在 C 语言基础上发展起来的另一种硬件描述语言,充分保留了 C 语言简洁、高效的编程风格。可用于从算法级、门级到开关级的多种抽象设计层次的数字电路系统的建模。Verilog HDL 语言于 1995 年同样被确认为集成电路硬件描述语言的标准。

在电子工程领域中,VHDL 语言和 Verilog HDL 语言相比,后者比前者简单,直观,容易学习。前者比后者严谨,并有很强的大规模行为级描述能力和系统级描述能力。本教材选用 VHDL 语言。

VHDL 的主要优点如下。

1)强大的系统描述能力

具有多层次的硬件描述功能,既可以进行门级电路描述,又可以进行系统级电路描述。既可以采用行为级描述,又可采用结构级描述。也可对上述描述综合成混合描述,经过层层细化求精,最终成为可直接付诸生产的电路级或版图参数描述。在 VHDL 的环境下可进行全过程设计,系统地创建高层次的数字电路系统模型。

2)可持续的再利用模块

支持大规模电路系统设计的分解和已有设计的再利用。可以将大规模电路系统按层次分解,采用结构化的开发手段,从上至下的设计方法,建立不同层次、不同种类的模块存档保存,也可以将保存的模块调出再次使用。设计人员之间可交流与共享设计成果,减少了硬件设计时间,降低了开发成本。

3)较长的生命设计周期

VHDL 的硬件描述与工艺技术无关,不会因工艺变化而使描述过时。而且与工艺技术有关的参数可通过 VHDL 提供的属性加以修正,工艺改变时,只需修改相应程序中的属性参数即可。

4)良好的语言可读性

VHDL 程序可以被计算机接受,也容易被读者理解。用 VHDL 书写的源文件,既是程序又是文档,既是技术人员之间交换信息的文件,又可作为合同签约者之间的文件。

5)方便的设计移植能力

VHDL 已成为 IEEE 承认的一个工业标准,事实上已成为通用硬件描述语言。同一设计描述可被不同的 EDA 软件平台支持,很方便地将设计描述进行移植。

1.5.1 VHDL 的基本组成

VHDL 程序描述的是设计单元,可以把设计的任意复杂电路视作一个数字单元或芯片,还可以是一个门电路。下面通过介绍一段 VHDL 的程序来了解 VHDL 的基本组成。

例 1.5.1 请用 VHDL 程序设计一个“三人表决”电路,即“少数服从多数”电路。

解:按前面介绍的“少数服从多数”的原则确立结果,将三个人的意见分别设置为输入变量 A、B、C,规定有同意或不同意两种意见如表决结果的 F 为输出逻辑函数,实现此电路的 VHDL 程序如下:

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;
```

```

ENTITY agree IS
    PORT(A,B,C :IN std_logic;
          F :OUT std_logic);
END agree;
ARCHITECTURE agree_arc OF agree IS
BEGIN
    F<=(A AND B)OR(B AND C)OR(C AND A);
END agree_arc;

```

VHDL 程序中不区分字母的大小写,本教材中 VHDL 中的关键字用大写,其他大小写不加以区别。

1. 程序包——参数部分

每个程序中的程序包有 IEEE 标准的标准程序包或设计者自身设计的程序包,而且程序包的调用数量不限。程序包是设计中的子程序和公用数据类型的集合,是构成设计工具的工具箱,工具箱中最基本的工具是数据类型包,调用此标准程序包的 VHDL 语言一般格式为:

LIBRARY 库名;

下面紧接的语句是 USE,用来调用库中的子库,其一般格式为:

USE 子库名;

这两句设置在 VHDL 程序的最前面,表示以后在程序中要用到的库和子库。

在例 1.5.1 程序中的第一行和第二行

LIBRARY ieee;

USE ieee.std_logic_1164.all;

就是表示以后在程序(实体或结构体)中要用到数据类型包中的数据类型。

VHDL 可以把任意复杂的电路系统视作一个模块,图 1.5.1 给出了 VHDL 程序模块示意图,程序包见图中调用程序包方框。

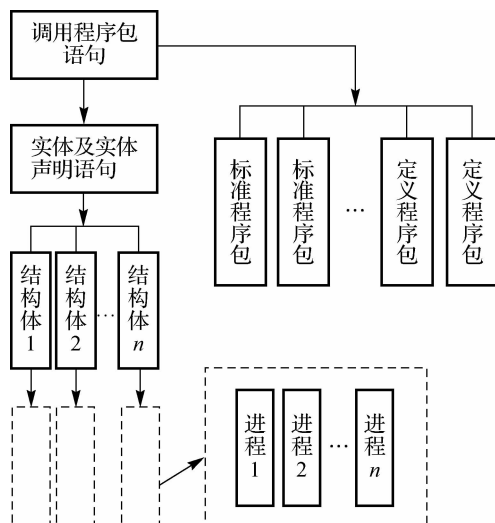


图 1.5.1 程序模块示意图

2. 设计实体——接口部分

设计实体在 VHDL 中是一个最基本的部分,类似于集成电路模块及管脚。一个 VHDL 描述的电路模块中仅有一个设计实体,它提供该设计模块的公共信息。VHDL 设计的电路系统可分层次,所以设计的模块实体既可以是顶层实体,又可以是最底层实体。

设计实体中的一部分是外部可见特性,如设计模块的名称、端口引脚信息等,还有一部分是不可见的,不再赘述。其一般表示格式为

ENTITY 实体名 IS

类属表;

PORT(端口表);

说明语句;

END 实体名;

在例 1.5.1 程序中的第三行至第六行

ENTITY agree IS

PORT(A,B,C :IN std_logic;

F :OUT std_logic);

END agree;

描述了表决电路实体,如图 1.5.2 所示。设计实体在图 1.5.1 VHDL 程序模块示意图中的位置见实体方框。

实体部分的大写单词 ENTITY、IS、PORT、IN、OUT 和 END 为关键字。在 ENTITY…END 之间表示实体内容,ENTITY 后的字符串 agree 表示实体的名称,即电路的符号名。端口(引脚)信息关键字 PORT 中的语句有四个端口,分别是输入(IN)模式 A、B 和 C,输出(OUT)模式 F,并描述了信号的流向。端口信息除了输入输出之外,还可以是双向、缓冲器等。std_logic 表示信号取值的数据类型为标准逻辑位,数据类型除了标准逻辑位之外,还可以是实数、整数、无符号数、物理以及以上数据类型组成的记录和数组集合等,信号类型也可以由设计者定义。

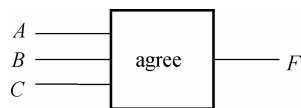


图 1.5.2 表决电路实体

3. 结构体——描述部分

结构体(ARCHITECTURE)用来描述实体硬件的互联关系、数据的传输和变换以及动态行为,动态行为包括并行行为和顺序行为。一个实体可以对应多个结构体,每个结构体可以代表硬件的某一方面特性,如行为特性、结构特性。而每一特性的描述,又由其层次、实现方法不同形成多个结构体。每一个结构体在实体之后,结构体在图 1.5.1 VHDL 程序模块示意图中的位置在实体之后。结构体的一般表示格式为:

ARCHITECTURE 结构体名 OF 实体名 IS

说明语句;

BEGIN

描述语句;

END 结构体名;

当表决电路的符号和外部端口 A、B、C 和 F 确定之后,就可以编写结构体中的内容确定实体内部电路。与实体 agree 相对应的结构体中的逻辑电路图如图 1.5.3 所示,相对应的

VHDL 结构体描述是例 1.5.1 程序中的第七行至第十行。

```

ARCHITECTURE agree_arc OF agree IS
BEGIN
    F <= (A AND B) OR (B AND C) OR (C AND A);
END agree_arc;

```

在关键字 ARCHITECTURE 标明的语句中描述实体的内部结构,结构体的名字为 agree_arc,且正在描述的结构体是实体 agree 的结构体。实体和结构体之间的关系是多种多样的,上例在结构体中的描述是行为描述,两两输入变量先与再或到输出 F 。此行描述读作 F 得到右边表达式值。在结构体中还有多种描述方法,如图 1.5.1 示意图中虚线框内的进程行为语句等将在后面更详细地讨论。

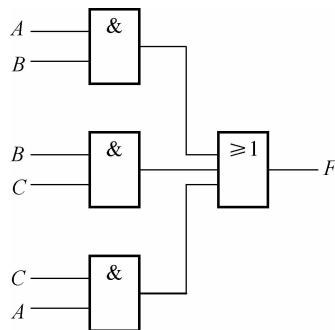


图 1.5.3 表决电路

1.5.2 VHDL 数据类型和属性

VHDL 和其他软件一样,都有保持数据和转换数据的功能。VHDL 硬件描述语言除了保持常量、变量功能外,还有保持、传递信号的功能。VHDL 用来保持数据的信号、变量和常量称为目标。目标的一般表示格式为:

目标种类 目标名 1,目标名 2,...:数据类型:=表达式或初始值。

1. 目标种类

1) 常量(CONSTANT)

常量的一般表示格式为:CONSTANT 常量名 1,常量名 2,...:数据类型:=表达式

对某些特定目标赋予数值,一次仅被分配一个值,不能再次改变。可以同时说明一个或多个常量,如 CONSTANT A1,A2:integer 和 CONSTANT A:integer=3。

2) 变量(VARIABLE)

变量的一般表示格式为:VARIABLE 变量名 1,变量名 2,...:数据类型:=表达式

用于对暂时数据的局部存储,变量可以赋予一系列的值。赋予变量一个新的数值后,它立即接受当前的数值。变量只在进程和子程序内部定义。如 VARIABLE A1,A2:integer 和 VARIABLE A:integer=3。

3) 信号(SIGNAL)

信号的一般表示格式为:SIGNAL 信号名 1,信号名 2,...:数据类型:=表达式

信号表示把元件端口连接在一起的互连线。和变量一样可以赋予一系列的值,但不同的是赋予信号的数值要到未来的某个时刻,才接受当前的数值。如 SIGNAL A1,A2:integer 和 SIGNAL A:integer=3。

在一行中定义多个目标名时,用“,”分开。目标的一般表示格式中的表达式是为了规定目标的初始值,这也可缺省。

2. 数据类型

VHDL 是一种非常严格的数据类型化语言,规定每个信号、常量、变量或表达式有确定的数据类型,一般来说,在表达式中分配数值给目标时的数据类型不可以被混用。每个目标和表达式的类型可在仿真之前静态地确定。

为了规定目标的特征,VHDL 含有很宽范围的数据类型。VHDL 除了有基本的数据类型之外,设计者还可以建立新的数据类型,类型说明部分规定类型名和类型范围,它的一般形式是:

TYPE <类型名> IS <类型范围> ;

类型范围包容了整数类型、实数类型、物理类型、枚举类型到文件类型的任何值。VHDL 有大量与类型兼容的有关规则:赋予目标的表达式数据类型必须和目标数据类型相同;许多预先确定的运算符的操作数必须具有相同的类型;一个实体的类型和它连接到的形式的类型相同。

VHDL 中可用数据类型共有五类:标量类型、复合类型、子类型、文件类型和寻址类型。

1) 标量类型

标量数据类型是基本的数据类型,它包括整数、实数、物理和枚举类型。一个信号或变量的数值有可能经常更换,但这个信号或变量一次只能被一种类型说明。

(1) 整数类型。整数类型与算术类型相似,它可以定义一个连续性的整数范围用来预定算术函数等,还可用作次序上升和下降的循环迭代。

(2) 实数类型。实数类型可以代替或模仿数学上实数的目标,它可用来表示带小数点的数或带小数点的浮点数,它的最大范围是 $-1.0E+38 \sim +1.0E+38$ 。

(3) 物理类型。物理类型可用来表示距离、电流和时间等物理量,物理类型要提供一个基本单位,然后在这个基本单位上定义多个或零个次级单位,每个次级单位都是基本单位的整数倍。

(4) 枚举类型。整数类型、实数类型和物理类型比较容易理解,较难理解的只是带有一些抽象意义的枚举类型数据。枚举类型在形式上是定义括弧内的字符串文字表,一个字符串文字在枚举类型定义中只能出现一次,但允许同样一个字符串文字出现在不同的枚举类型的字符串文字表中,枚举类型的字符串文字表中的文字是由设计者定义的,这些值可以是单个字母,也可以是一个字符串,如 Tea,Lunch,a 等。

下面给出枚举数据类型编程实例,以便加深理解。

例 1.5.2 请用 VHDL 设计一个交通指示灯流程电路。

解: PACKAGE traffic_pkg IS

TYPE tra IS(green,red,yellow);

END traffic_pkg;

USE work.traffic_pkg.all;

ENTITY traffic IS

PORT(previous_tra:IN tra;

next_tra :OUT tra);

END traffic;

ARCHITECTURE traffic_arc OF traffic IS

BEGIN

WITH previous_tra SELECT

next_tra <=green WHEN red,

yellow WHEN green,

```

        red WHEN yellow;
END traffic_arc;

```

自定义程序包中,定义了枚举类型数据为 tra,枚举类型表中有 3 个字符串,代表着枚举信号 red(红灯)、green(绿灯)和 yellow(黄灯)。在实体中定义的端口信号是两个,一个是输入枚举信号 previous_tra,另一个是输出枚举信号 next_tra。在结构体中,判断输入是枚举数据的哪一个,然后将另一个枚举型数据送到输出,如前面亮的是红灯,接下来,间隔一定的时间后是绿灯,最后是黄灯,然后进入循环状态。

2) 复合类型

复合类型由数组类型和记录类型组成,它们的元素是标量类型的元素。数组类型和记录类型对建立较复杂系统和抽象数据类型的建模是非常有用的,用记录和数组的巧妙组合能做出更容易理解的模块。

数组类型由相同的标量元素组成,即同构复合类型。数组可以是一维、二维或多维,可以是限定性也可以是非限定性,限定性数组的定义指定了数组的上下界,而非限定性数组则不指定上下界,本章只涉及限定性数组。定义一维数组可以用以下两行语句。

```

TYPE word IS ARRAY(15 DOWNT0 0)OF BIT;
CONSTANT ROM :BIT_VECTOR(0 TO 15);

```

下面的第一行是 1 个二维数组类型的语句,第二到第四行是二维限定性数组的定义,表示行的范围是 1~10,列的范围是 1~40。

```

TYPE r_ma IS ARRAY( 1 TO 10,1 TO 40)OF std_logic;
TYPE column IS RANGE 1 TO 40;
TYPE row IS RANGE 1 TO 10;
TYPE matrix IS ARRAY(row,column)OF std_logic;

```

记录型类型可以把各物理、整数、实数、枚举归成一个单一的目标组,记录的每个元素由它的字段名访问,记录元素包括任何类型的元素,它的元素类型可以不同也可以相同,所以记录类型是异构的。

3) 子类型

子类型的设置可以定义物理、整数、实数、枚举、数组和记录型类型的子集,是为了对需要赋值的语句加以约束,建立有所限制范围子类型。如果需要的基本类型范围较大,就可以选择其中一部分作为子类型,子类型的范围要小于它属于的数据类型范围。子类型还可以再定义子类型。

下面列举一个八位多路选择器 VHDL 程序,首先定义了整数范围,再对整数范围限定了一个子集,使其仅使用整数类型中的 8 个离散数据。

例 1.5.3 请用子类型语句设计一个从 8 路信号选择 1 路送到输出的 VHDL 程序。

解: PACKAGE mux_pkg IS

```

SUBTYPE eightval IS INTEGER RANGE 0 TO 7;
END mux_pkg;
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE work.mux_pkg.all;

```

```

ENTITY mux8 IS
    PORT(I0,I1,I2,I3,I4,I5,I6,I7:IN std_logic;
         sel :IN eightval;
         Q :OUT std_logic);
END mux8;
ARCHITECTURE mux8_arc OF mux8 IS
BEGIN
    WITH sel SELECT
        Q<= I0 WHEN 0,
            I1 WHEN 1,
            I2 WHEN 2,
            I3 WHEN 3,
            I4 WHEN 4,
            I5 WHEN 5,
            I6 WHEN 6,
            I7 WHEN 7;
    END mux8_arc;

```

mux_pkg 程序包描述了一个 eightval 的子类型,它对整数类型加了一个限制,允许子类型的数值范围在 0~7。实体 mux8 的输入端口之一 sel 的数据类型则用子类型 eightval 规定,它的值域是 0~7,它和定义的子类型的范围相对应,即完全包容子类型的集合。

4) 文件类型

文件数据类型要有一个文件类型说明语句,用它来指定文件类型名和基本类型,还有与其相匹配文件对象说明语句,指定目标的名字以及所属数据类型和文件所在的数据通道。它的一般形式为:

```

TYPE 数据类型名 IS FILE OF INTEGER;
FILE 文件名 :数据类型名 IS IN “数据通道”

```

用数据类型名指出文件类型 INTEGER。文件名是由设计者定义,“数据通道”指出数据文件存放在硬盘的哪个子目录下的哪个文件。文件数据类型只能在子程序或进程中操作。

5) 寻址类型

在此不再讨论寻址类型,请读者参考有关文献。

3. 属性

VHDL 中的属性使 VHDL 程序更加简明扼要、容易理解,VHDL 的属性在时序电路设计程序中几乎处处可见,如值类属性的左边界、右边界、上下边界以及值类属性的长度,用于返回数组的边界或长度。

VHDL 用来检测信号上升沿和下降沿以及前一次发生的事件为函数信号属性,它反映一个信号是否正好有值的变化或事件的发生,如 clk'EVENT,这个属性为“EVENT”,对检查信号(时钟)边沿触发是有效的。它可用来检查一个刚刚发生变化的信号,即推断出在信号上发生了一个跳变。下面举一个例子,说明函数信号属性的用法。

```

IF clk='1' AND clkEVENT THEN

```

$q \leq d;$
END IF;

上述语句中用到了函数信号属性 $\text{clk}'\text{EVENT}$, 说明如果时钟信号 clk 为高电平且事件刚刚发生, 即时钟上升沿, 此时 q 得到 d 的信号。

4. 运算操作符

为了构成 VHDL 的程序表达式, VHDL 定义了 21 个运算操作符, 逻辑运算符、关系运算符和连接运算符。表 1.5.1 按运算操作符的优先级顺序由低到高列出, 在编写程序时, 应用括号确定。

表 1.5.1 VHDL 运算操作符

优 先 级	运算操作符类型	操 作 符	功 能
低  高	逻辑运算符	AND	逻辑与
		OR	逻辑或
		NAND	逻辑与非
		NOR	逻辑或非
		XOR	逻辑异或
	关系运算符	=	等于
		/=	不等于
		<	小于
		>	大于
		<=	小于等于
		>=	大于等于
	算数运算符	+, -	加、减
	连接运算符	&	连接
	关系运算符	+, -	正、负
		*	乘
		/	除
		MOD	求模
		REN	求余
		**	指数
		ABS	取绝对值
	逻辑运算符	NOT	取反

思考题

题 1.5.1 VHDL 语言的数据类型有 _____、_____、_____、_____。

题 1.5.2 一个 VHDL 是否应有一个结构体? 结构体的作用是什么? 一个 VHDL 可以有多个结构体吗?

本章小结

本章主要介绍数字电路系统分析和设计过程中需要的数学基础,主要包括常用的数制和码制、逻辑代数的基本运算公式和定律、逻辑函数的表示形式及化简方法和 VHDL 基础等。

数字系统中用二进制进行逻辑运算,并表示数据。在二进制位数较多时,也使用十六进制或八进制计数。采用的各种计数制之间可以相互转换。BCD 码对应日常生活中的十进制,是常用的编码,其中 8421BCD 码使用最广泛。另外,格雷码由于可靠性高,也是一种常用码。

分析数字电路与系统的数学工具是逻辑代数。逻辑代数中的三种基本运算是与、或、非运算。一个逻辑问题可用逻辑函数来描述,逻辑函数有真值表、逻辑表达式、卡诺图、逻辑图、波形图和 VHDL 程序等几种常用的表达方法,它们各具特点并可以相互转换。

为了使逻辑电路简单、成本低、可靠性高,逻辑函数化简成为获得最简逻辑电路的方法。化简的方法主要有公式法和卡诺图法。这两种基本的化简方法不适于较多变量的复杂逻辑函数。逻辑函数输入变量较多时,可采用计算机辅助化简,目前已有多种实用的逻辑化简软件平台,如 MAXPLUS II 等。

VHDL 是一种数据类型非常严谨的电路设计语言,包含五种数据类型。数据或信号的传输应严格遵守数据类型一致。VHDL 具有很强的电路设计优势,可以设计芯片级、系统级和版图级的电路。VHDL 不仅可以进行逻辑综合,还可以进行逻辑测试和仿真。用 VHDL 设计数字系统是当今数字电子电路分析和综合的发展趋势。

习 题

习题 1.1 写出下列二进制数的原码、补码和反码。

(1) $(+1010)_B$ 的原码为 _____; 补码为 _____; 反码为 _____。

(2) $(-1100)_B$ 的原码为 _____; 补码为 _____; 反码为 _____。

习题 1.2 十进制数与 BCD 码间的转换。

(1) $(6)_D = (\quad)_{8421} = (\quad)_{\text{余3码}}$ 。

(2) $(0110)_{8421} = (\quad)_{\text{余3码}} = (\quad)_{8421\text{BCD}}$ 。

习题 1.3 用补码运算。

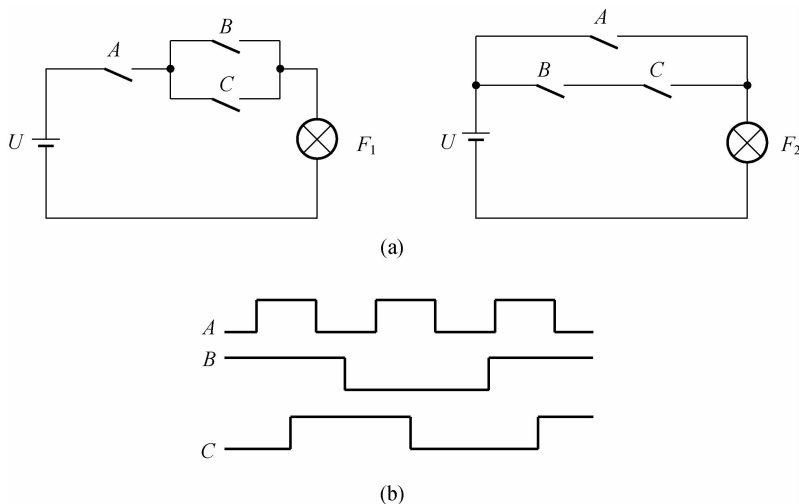
(1) $34 + 21 = \underline{\hspace{2cm}}$ 。 (2) $35 - 16 = \underline{\hspace{2cm}}$ 。

习题 1.4 将给定的进制数转换成相应的进制数。

(1) $(101011.010)_B = (\quad)_Q = (\quad)_D = (\quad)_H$ 。

(2) $(25.678)_D = (\quad)_B = (\quad)_Q = (\quad)_H$ 。

习题 1.5 写出题图 1.1 所示开关电路中 F 和 A 、 B 、 C 之间逻辑关系的真值表、逻辑函数和逻辑电路图。若已知 A 、 B 、 C 变化波形,画出 F_1 、 F_2 的波形。



题图 1.1 习题 1.5 电路图
(a)逻辑电路图 (b)A、B、C变化波形

习题 1.6 用逻辑代数的基本公式和常用公式证明下列各等式。

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

$$(2) AB+AC+BC = \overline{(A+B)(A+C)(B+C)}$$

习题 1.7 试画出用与非门和反相器实现下列函数的逻辑图。

$$(1) F = AB + BC + A\bar{C}$$

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

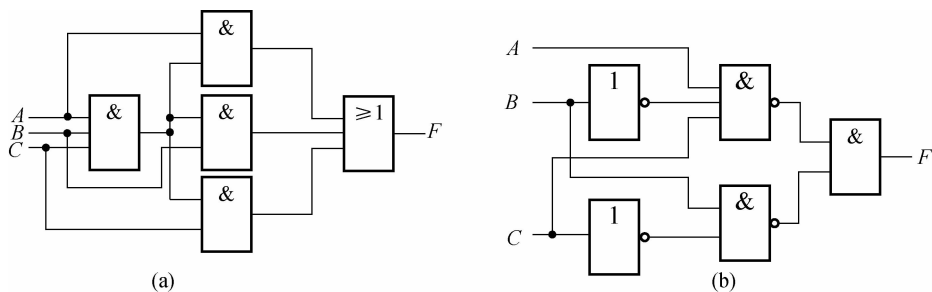
习题 1.8 试画出用或非门和反相器实现下列函数的逻辑图。

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

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

习题 1.9 已知函数 $F(A, B, C, D) = \sum m(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14)$, 试用最少数目的与非门实现, 要求电路的输入仅为原变量。

习题 1.10 写出题图 1.2(a)、(b)中各逻辑图的逻辑函数式, 并化为最简与或式。



题图 1.2 习题 1.10 电路图

习题 1.11 用代数法将下列逻辑函数化简为最简与或式。

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

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

习题 1.12 求下列函数的反函数并化为最简与或形式。

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

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

习题 1.13 证明下列逻辑恒等式(方法不限)。

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

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

习题 1.14 用卡诺图化简下列函数,分别写出其最简与或式。

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

$$(2) F(A, B, C, D) = \sum m(1, 7, 9, 10, 11, 12, 13, 15)$$

习题 1.15 用卡诺图化简下列无关项的函数,分别写出其最简与或式和或与式。

$$(1) F(A, B, C, D) = \sum m(5, 6, 9, 10) + \sum d(0, 1, 2, 13, 14, 15)$$

习题 1.16 试用卡诺图对已知函数做逻辑运算。

$$\text{已知} \begin{cases} F = A\overline{C}D + \overline{A}B\overline{D} + BCD + \overline{A}CD \\ G = \overline{\overline{A}\overline{C}\overline{D} + BC + A\overline{C}\overline{D}} \end{cases}$$

试求: (1) $F \cdot G$ 。

(2) $F + G$ 。

(3) $F \oplus G$ 。

习题 1.17 多输出函数卡诺图化简。3 个函数为同一电路的三个输出端,试用最少数目的与非门实现其电路。

$$F_1 = \sum m(3, 4, 6, 7, 12, 14, 15)$$

$$F_2 = \sum m(0, 2, 3, 4, 7, 8, 9, 12)$$

$$F_3 = \sum m(0, 2, 3, 7, 9, 11)$$

习题 1.18 请写出一个“三人表决电路”的与或逻辑表达式, A、B、C 中 A 有一票决定权。

研究论文

简述 1900—2050 年电子技术的发展与展望。

研究内容及要求: 在研究性学习中, 了解电子技术的历史发展与现状, 并作一全面的综述。结合专业及国内外发展趋势与对未来要求, 以某一方面的发展, 设想 2050 年时的电子技术发展状况。

集成电路是将电路中的半导体器件、电阻、电容及连线制作在一起,如图 2.0.1 和图 2.0.2 所示。图 2.0.1 是由 4 个 MOS 管组成的一个 2 输入的与非门,图 2.0.2 是一块塑料或陶器上集成的 4 个 2 输入的与非门。用塑料或陶瓷将电路封装在一个壳体内,将输入、输出端连接到外部的引脚,就构成了集成电路,即内部为逻辑门电路的集成电路为集成逻辑门。图 2.0.3 是将图 2.0.2 中的电路(CMOS 四 2 输入与非门)用塑料封装在一起,这种封装形式称双列直插式封装,简称 DIP(dual in-line package)封装。

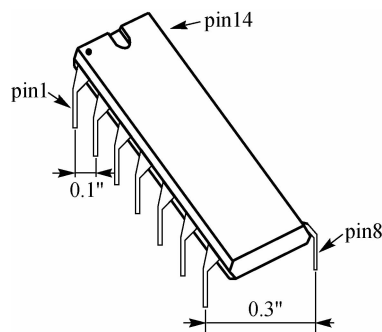
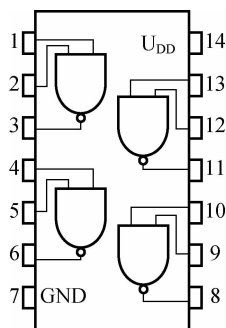
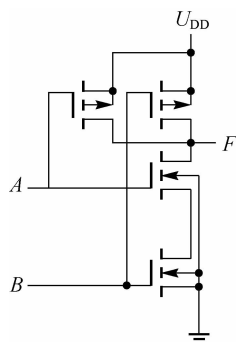


图 2.0.1 CMOS 2 输入与非门

图 2.0.2 4 个 2 输入与非门

图 2.0.3 DIP 封装外形图

第 1 章介绍的与、或、非基本逻辑运算和与非、或非、与或非、异或、同或等复合逻辑运算都可以用集成逻辑门实现。由金属氧化物半导体(MOS)N 型、P 型晶体管组成的集成电路称为单极型集成电路。由双极型晶体管[晶体管-晶体管逻辑(transistor-transistor-logic, TTL)、发射极耦合逻辑(ECL)、集成注入逻辑(I²L)]等组成的集成电路称为双极型集成电路。用这些集成逻辑门可以组成各种数字电路或数字系统。

为正确使用各种数字电路集成芯片,需要理解基本逻辑门的组成与工作原理。为更好地工程应用数字集成电路芯片,需要掌握其电气特性和技术参数,如输入/输出电压、扇出能力、功耗、速度、抗干扰和噪声容限等。

对本章内容的理解与熟练掌握可以使读者合理选择、正确使用集成电路设计数字电路与系统,将为后续章节内容的学习奠定良好的基础。

2.1 半导体晶体管的开关特性

2.1.1 逻辑电平

在逻辑电路中,常用高、低电压,即高、低电平表示数字逻辑运算中的“逻辑 1”或“逻辑 0”。一定范围电压的输入信号,其代表的逻辑电平不会改变,即高电平或低电平代表一定的电压范围。高电平或低电平不是一个固定的数值,允许一定范围的变化,也就是说一定范围的低电压 U_{IL} 表示低电平,一定范围的高电压 U_{IH} 表示高电平。如典型的双极型 TTL 门电路中,2.4~3.6 V 输入电压范围内的电压为高电平,0~0.8 V 输入电压范围内的电压为低电平,其高电平的最低电压和低电平的最高电压差值是 1.6 V。TTL 门电路输入高电平和低电平的示意图如图 2.1.1 所示。

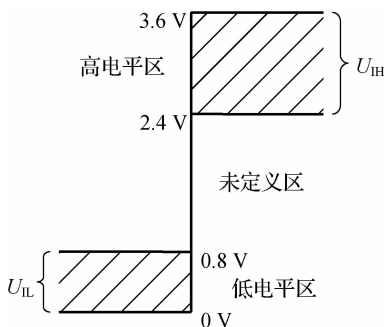


图 2.1.1 TTL 门电路输入高电平和低电平示意图

不同的晶体管器件、不同的逻辑电路组成,其集成电路的输入/输出的高电平或低电平对应的电压范围不同,其差值也不同。如同样是双极型晶体管组成的 ECL 电路,电路中的晶体三极管工作在非饱和区和截止区,其高电平的最低电压和低电平的最高电压差值是 0.8 V。又如单极型、超低电压的 74AUC 系列,其高电平的最低电压是 1.2 V,低电平的最高电压是 0.63 V,差值是 0.57 V。特别要说明的是某些单极型器件组成的数字电路的高低电平电压的差值可以达到 5~30 V。

数字电路中用高电平表示逻辑“1”,用低电平表示逻辑“0”,称之为正逻辑;反之,用高电平表示逻辑“0”,用低电平表示逻辑“1”,称之为负逻辑。

正逻辑和负逻辑是两种不同的表现形式,一个数字逻辑系统中,可以用正逻辑表示,也可用负逻辑表示,不会导致逻辑的错误,但混用时容易出现混乱的情况。本书不作特别说明,均使用正逻辑。

2.1.2 三极管(TTL)开关特性

图 2.1.2(a)是典型的由 NPN 型三极管组成的共射极电路,图 2.1.2(b)是该三极管的输出特性曲线。输出特性曲线中有三个工作区域,即放大区、截止区和饱和区。在模拟电子技术的学习中,主要是使电路工作在放大的线性区。而在数字电子技术学习中,双极型 NPN 三极管是作为一个开关元件,大多数情况下工作在饱和(导通)状态和截止(断开)状态的非线性区。

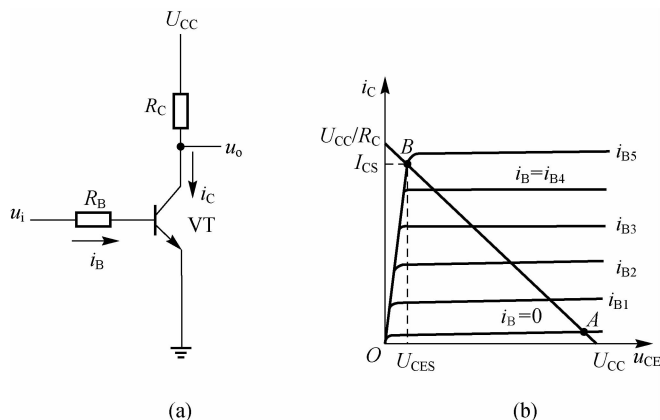


图 2.1.2 双极型晶体管开关电路

(a) 电路 (b) 输出特性曲线

1. 截止状态

图 2.1.2(a) 中三极管的发射结导通电压为 $0.5 \sim 0.7 \text{ V}$ 。当输入电压等于低电平, 即 $u_i = U_{IL} \approx 0 \text{ V}$ 时, 发射结没有正向偏置。 $u_B - u_C < 0$, 集电结反向偏置。流过 PN 结的电流很小, 使得 $i_B \approx 0$, $i_C \approx 0$, $u_o = u_{CE} \approx U_{CC}$ 。这时, 输出电压在 U_{CC} 附近, 基极和集电极的电流都近似等于 0, 对应图 2.1.2(b) 中的 A 点附近。在这种情况下, 集电极回路和基极回路几乎是开路状态, 相当于开关断开, 此种情况三极管处在截止状态。

2. 饱和状态

当输入电压等于高电平时, 即 $u_i = U_{IH} \approx 3.6 \text{ V}$, 三极管工作在图 2.1.2(b) 中的临界值 B 点, 三极管的 i_B 称为临界饱和基极电流 I_{BS} , i_C 称为临界饱和集电极电流 I_{CS} , 集射间的电压称为临界饱和集电极电压 U_{CES} , $U_{CES} \approx 0.3 \sim 0.6 \text{ V}$ 。 i_B 上升到 B 点以后, i_B 再增加, 集电极电流也不会按比例增加, 达到了饱和。若再增加, 则饱和程度更深, 进入深度饱和, 而集电极电流维持在 I_{CS} 附近, U_{CES} 的值保持不变。集电极回路和基极回路几乎是短路状态, 相当于开关闭合, 三极管处在饱和状态。

3. 浅饱和状态

当三极管处于饱和状态, 尤其是深度饱和状态时, 聚集在基区的电荷很浓。在从饱和区回到截止区时, 必须用很多的时间消耗掉基区储存的电荷, 即从深度饱和状态到截止状态的时间长, 电路的转换速度因此降低。为了提高电路的转换速度, 使三极管从饱和到截止尽快释放电荷, 就不能让三极管处于深度饱和状态, 而是仅仅处于刚刚饱和的临界饱和状态, 这就是浅饱和状态。

如图 2.1.3(a) 所示的电路中, 三极管集电结并联一个肖特基二极管 (SBD), 肖特基二极管的正向压降较小, $U_D = 0.4 \text{ V}$, 容易导通。当三极管处于饱和状态, 并趋于深度饱和状态时, 肖特基二极管分流了三极管的一部分电流, 使其工作在浅饱和状态。肖特基三极管图形符号如图 2.1.3(b) 所示。

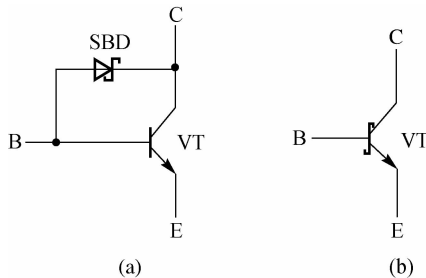


图 2.1.3 肖特基三极管

(a) 电路结构 (b) 图形符号

4. 开关时间

三极管的饱和状态和截止状态的相互转换是基区电荷的聚集和消散。而聚集和消散电荷需要一定的时间,集电极电流 i_C 的变化滞后于输入电压 u_i 的变化,如图 2.1.4(a)和图 2.1.4(b)所示。

图 2.1.4(a)波形图是三极管输入端加上输入高低电平的脉冲信号,图 2.1.4(b)和图 2.1.4(c)所示波形分别表示三极管电路输出电流 i_C 波形和输出电压 u_o 波形。图 2.1.4(b)中的 t_d 为延迟时间、 t_r 为上升时间、 t_s 为存储时间、 t_f 为下降时间。开通时间 t_{on} 为 $t_d + t_r$,关闭时间 t_{off} 为 $t_s + t_f$ 。当输入从低电平到高电平时,即经过开通时间 t_{on} ,发射区向基区扩散电子,产生基极电流 i_B ,同时通过基区流向集电区产生集电极电流 i_C 。当 i_C 不断增加,直到最大值 $I_{C(sat)}$,三极管进入饱和状态,此时基区储存的电荷越来越多。当输入从高电平到低电平时,即经过关闭时间 t_{off} ,基区聚集的大量电荷开始消散。如果处在浅饱和状态,聚集的电荷很快消失,三极管很快通过放大区进入截止状态。如果处在深度饱和状态,随着聚集电荷的消散,先进入浅饱和状态,再进入截止状态。显然深度饱和状态转换到截止状态比浅饱和转换到截止状态关闭时间 t_{off} 要长。所以要提高三极管的开关速度,应降低三极管的饱和深度,加速基区聚集的电荷消散。

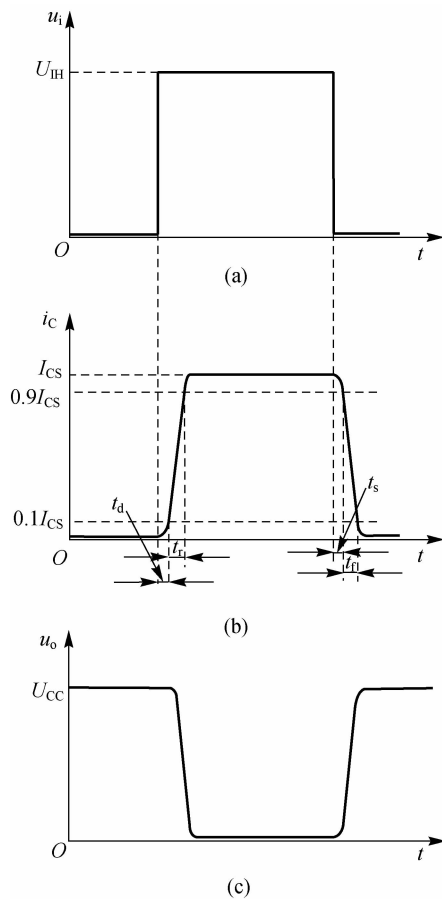


图 2.1.4 三极管开关时间
(a)输入电压波形 (b)输出电流波形
(c)输出电压波形

2.1.3 MOS 管开关特性

MOSFET(metal-oxide-semiconductor field-effect transistor)是金属氧化物场效应晶体管。参与导电的载流子只有一种(电子或空穴),故称为单极型器件,由单极型器件组成的基本门电路称为单极型逻辑门电路。MOS 集成电路主要包括 NMOS 电路、PMOS 电路以及互补 MOS 电路,即 CMOS 电路,本教材只介绍 CMOS 基本逻辑门电路。

图 2.1.5(a)和图 2.1.5(b)是 NMOS 管和 PMOS 管的逻辑符号。其中 G 为栅极,D 为漏极,S 为源极,B 为衬底,一般情况下衬底和源极接在一起。图 2.1.5(c)是由增强型的 NMOS 管组成的共源极电路,VT₂ 是用以代替漏极的电阻 R_d ,称为有源负载。输入在栅源两端控制漏源的电流,不像双极型晶体管那样由基极电流控制,所以是电压控制器件。图 2.1.5(d)是 MOS 管的输出特性曲线,可以看出栅源电压 $u_{GS} = u_i$ 是一簇曲线,在负载线上 i_D 随着 u_i 的增加而增加。MOS 管有三个工作区,即截止区、可变电阻区和恒流区。

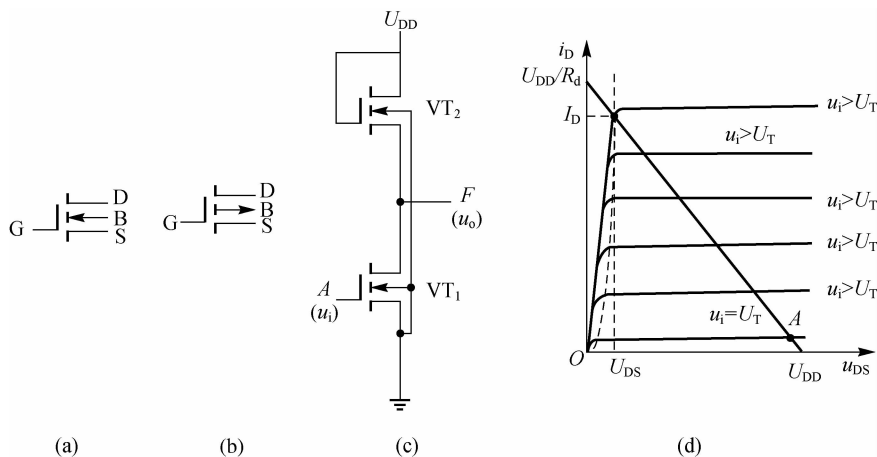


图 2.1.5 双极型晶体管开关电路

(a)NMOS 管 (b)PMOS 管 (c)共源极电路 (d)MOS 管输出特性曲线

1. 截止区

图 2.1.5(c)中 NMOS 管(VT_1)的栅源间输入电压 u_i 小于开启电压 U_T 时,如图 2.1.5(d)中 $u_i = U_T$ 下方时,MOS 管截止,此时漏极 D 和源极 S 间没有形成电流, $i_D \approx 0$,呈高阻状态,输出电阻很高。

2. 可变电阻区

当加在栅源两端电压 u_i 大于开启电压 U_T 时,并且 $u_{DS} > 0$, $u_{GS} > u_{DS}$,则 MOS 管工作在图 2.1.5(d)的第一象限靠近纵轴的区域,即虚曲线左侧。此时 D-S 间有导通沟道,有电流通过。

3. 开关时间

在图 2.1.5(c)中 NMOS 管(VT_1)的栅源间加上理想脉冲信号。当加上高电压 $u_i = U_{DD}$ 时,如图 2.1.6 所示,此时 D-S 间有导通沟道,NMOS 管相当于开关闭合。当加低电压 $u_i = 0$ V 时,此时 D-S 间无导通沟道,相当于 D-S 间存在极大电阻,类似开关断开。由于 NMOS 管极与极之间存在寄生电容,使 NMOS 管在导通与闭合两种状态间转换时,尤其在频率较高的情况下,很容易受到电容充电和放电的影响。输出电压 u_o 的上升和下降时间变得缓慢了,输出电压 u_o 的变化滞后于输入电压 u_i 的变化,可观察图 2.1.6(b)所示的输入由低电平到高电平 50%至输出由高电平到低电平 50%的延迟时间为 t_{PHL} ,输入由高电平到低电平 50%至输出由低电平到高电平 50%的延迟时间为 t_{PLH} ,输出电压 u_o 已经不再是理想的矩形波形。

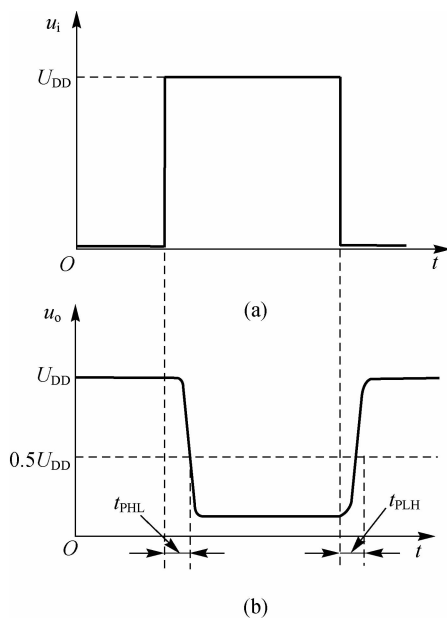


图 2.1.6 MOS 管开关时间

(a)输入电压波形 (b)输出电压波形

思考题

题 2.1.1 肖特基三极管是普通三极管集电结并联一个_____,其正向压降较小容易导通。当三极管趋于深度饱和状态时,肖特基二极管_____了三极管的一部分电流,使三极管工作在浅饱和状态。

题 2.1.2 三极管的饱和状态到截止状态的转换,是_____电荷的消散。而消散电荷需要一定的时间,这样引起输出波形的_____。

题 2.1.3 由 NMOS 增强型管制成的有源负载_____。

- (A) 栅极和电源连接在一起 (B) 是两端元件
(C) 栅极和源极连接在一起 (D) 是三端元件

题 2.1.4 如果晶体管基极—射极输入是理想的矩形波,且能使晶体管导通与截止。在集电极—射极输出的波形是非理想的矩形波。(判断对错题)

2.2 TTL 基本逻辑门电路

2.2.1 标准 TTL 非门

图 2.1.2(a)实现的是输出电压相对输入电压反相的功能,即非逻辑门,其工作速度受到两方面的影响,一是非门内部晶体管工作在饱和状态时对开关速度的影响,二是非门输出端接容性负载时对工作速度的影响。

晶体管工作在饱和状态时基区存储大量的载流子,当晶体管由饱和转向截止时,存储的载流子来不及消散,晶体管不能迅速脱离饱和状态,产生了延时,因此影响非门的开关速度。

当非门输出端接容性负载时,在输出由低电平转向高电平的瞬间,需要对负载电容充电,充电时间常数的大小,将影响非门从低电平变化到高电平需要的时间,产生上升的坡度,不但影响非门的开关速度,也使输出波形不理想,如图 2.1.4(c)所示。

针对上述问题,应加速饱和和管存储电荷的扩散速度,减小对负载电容充电的时间常数。以下将介绍的集成电路逻辑门是实际应用电路,能较好地解决上述两个问题。

1. TTL 非门电路结构

图 2.2.1 为标准 TTL 非门电路图。

该电路由三部分组成:输入级、中间级和输出级。输入级由晶体管 VT_1 、二极管 VD_1 和基极电阻 R_1 组成。输入端的二极管 VD_1 是钳位二极管,一方面可以抑制输入端可能出现的负向干扰脉冲,另一方面防止输入电压为负时 VT_1 的发射极电流过大, VD_1 起到保护 VT_1 的作用。中间级由 VT_2 、 R_2 和 R_3 组成,它是一个反相器,从 VT_2 的集电极 C_2 和发射极 E_2 上可以分别获得两个相位相反的电压信号,供给输出级使用。输出级

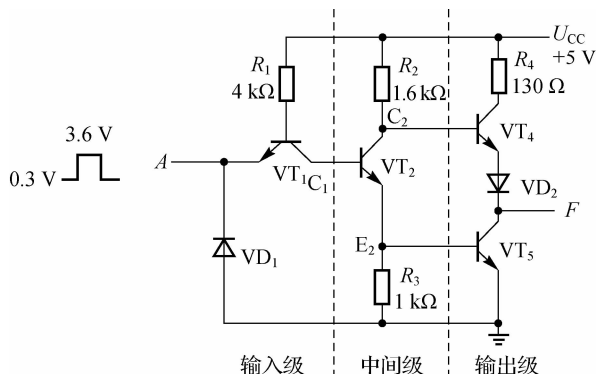


图 2.2.1 TTL 非门电路

由 VD_2 、 VT_4 、 VT_5 和电阻 R_4 组成。由于 VT_4 的基极信号为 C_2 ， VT_5 的基极信号为 E_2 ，而 C_2 和 E_2 信号相位相差 180° ，所以 VD_2 、 VT_4 导通时 VT_5 截止； VT_5 导通时 VD_2 、 VT_4 截止。这种电路称为推挽式电路，也称图腾柱电路，推挽式输出电路结构具有较强的带负载能力。

2. TTL 非门工作原理

根据非逻辑运算关系，可以将输入端 A 的输入信号分为两种情况，一种情况是 A 接高电平，另一种情况是 A 接低电平。现按此两种情况分别讨论 TTL 非门的工作原理。

1) 输入端接低电平

设 A 端接低电平 (0.3 V)， VT_1 管对应 A 端的发射结正偏导通，则 $U_{B1} = U_A + U_{BE1} = (0.3 + 0.7)\text{ V} = 1\text{ V}$ 。由于 $U_{B1} = 1\text{ V}$ ，无法使 VT_1 的集电结和 VT_2 、 VT_5 的发射结都导通，所以 VT_2 、 VT_5 截止。此时 VT_2 的集电极电位 $U_{C2} \approx U_{CC} = 5\text{ V}$ ，电源 U_{CC} 经 R_2 向 VT_4 提供基极电流，合理选择 R_2 及 R_4 ，使 VT_4 处于导通状态，门电路输出为高电平， $U_{OH} = U_{CC} - U_{BE4} - U_{VD2} \approx (5 - 0.7 - 0.7)\text{ V} = 3.6\text{ V}$ 。

2) 输入端接高电平

当输入端 A 接高电平 (3.6 V) 时，如果不考虑 VT_2 和 VT_5 的存在，则 VT_1 管基极 $U_{B1} = (3.6 + 0.7)\text{ V} = 4.3\text{ V}$ 。由于 VT_2 和 VT_5 的存在， VT_2 和 VT_5 的发射结正偏导通， VT_1 管基极最多不超过 2.1 V ，即 $U_{B1} = U_{BC1} + U_{BE2} + U_{BE5} = (0.7 \times 3)\text{ V} = 2.1\text{ V}$ ，所以 VT_1 发射结反偏截止，这时 VT_1 的集电结正偏导通， VT_1 的基极电流 I_{B1} 流向 VT_1 的集电结 C_1 并注入 VT_2 的基极。

$$I_{B1} = \frac{U_{CC} - U_{B1}}{R_1} \approx \left(\frac{5 - 2.1}{4} \right) \text{ mA} \approx 0.73\text{ mA}$$

此时 VT_1 处于倒置工作状态，集电极当发射极用，发射极当集电极用。倒置工作状态时电流放大系数 $\beta_{\text{反}}$ 很小 ($\beta_{\text{反}} < 0.02$)， $I_{B2} = I_{C1} = (1 + \beta_{\text{反}}) I_{B1} \approx I_{B1}$ ，由于 I_{B1} 较大足以使 VT_2 饱和，这时 VT_2 的集电极电位为

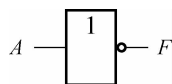
$$U_{C2} = U_{CES2} + U_{BE5} \approx (0.3 + 0.7)\text{ V} = 1\text{ V}$$

该电压加至 VT_4 基极，由于 VD_2 的存在，不足以使 VT_4 导通，因此 VT_4 截止。对于 VT_5 管，其基极电流 $I_{B5} \approx I_{E2}$ ，且 $I_{C5} \approx I_{E4} \approx 0$ ，所以 VT_5 处于饱和状态，门电路输出端 F 为低电平， $U_{OL} = U_{CES5} \approx 0.3\text{ V}$ 。

综上所述，当输入端接高电平时，输出为低电平；当输入端接低电平时，输出为高电平。由此可见，电路的输出和输入之间满足非逻辑关系

$$F = \overline{A}$$

图 2.2.2 非门的逻辑符号



电路逻辑符号如图 2.2.2 所示。

上述两种输入条件下 TTL 非门各管工作状态如表 2.2.1 所示。

表 2.2.1 TTL 非门各三极管工作状态

输 入	VT_1	VT_2	VT_4	VT_5	输 出
高电平	倒置	饱和	截止	深饱和	低电平
低电平	深饱和	截止	导通	截止	高电平

3. TTL 非门工作速度

(1)在输出由低电平转向高电平变化的瞬间, VT_4 和 VT_5 同时导通, 加速 VT_5 管脱离饱和状态。由于 VT_2 迅速截止, U_{C2} 迅速上升, 导致 VT_4 的迅速导通, 在 VT_5 尚未完全脱离饱和和区时, 将出现 VT_4 和 VT_5 瞬间同时导通的情况, 在 VT_5 的集电极产生一个几十毫安的集电极尖峰电流, 这个瞬时大电流破坏了 VT_5 的饱和条件, 加速了 VT_5 中存储电荷的消散速度, 使 VT_5 迅速脱离饱和状态, 进入截止状态, 进一步提高了非门的开关速度。

(2)推拉电路结构降低了非门的输出电阻, 减小对负载电容的充电时间。当非门输出低电平时, VT_5 处于深度饱和状态, 输出电阻很低; 当非门输出高电平时, VT_4 导通, 组成射极跟随器, 输出电阻也很低。因此无论哪种状态输出, 非门的输出电阻都很低, 减小了对负载电容的充电时间, 使容性负载对工作速度的影响小。

4. TTL 非门电气特性及主要参数

电气特性是指集成电路在外部表现出来的各种特性。掌握集成电路的外特性及其主要参数是用户在工程应用时, 正确设计使用、维护电路的重要依据。一般情况下, 集成电路生产厂家会提供各种逻辑电路的数据手册。下面将介绍手册中通常提供的特性曲线及其主要参数。

1) 电压传输特性

非门的电压传输特性是指非门输出电压 u_o 与输入电压 u_i 之间的关系, 即 $u_o = f(u_i)$ 的函数关系。非门电压传输特性的测试电路及特性关系分别如图 2.2.3(a) 和 2.2.3(b) 所示。电压传输特性关系大致分为四段: AB 段、BC 段、CD 段和 DE 段。

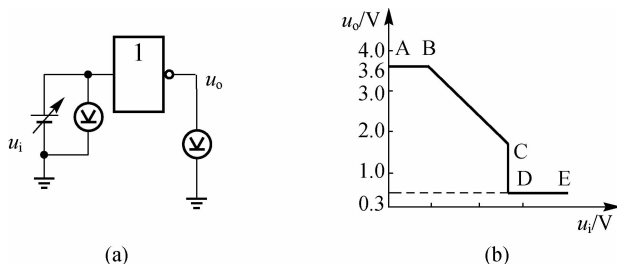


图 2.2.3 TTL 与非门电压传输特性

(a) 非门电压传输特性测试电路 (b) 非门电压传输特性

AB 段: 当 $u_i < 0.6$ V 时, VT_1 的基极电压 $U_{B1} < (0.6 + 0.7)$ V = 1.3 V, 不足以使 VT_2 导通, VT_2 和 VT_5 截止, VT_4 导通, 输出高电平 $u_o = 3.6$ V, 故 AB 段为截止段。

BC 段: 当输入 u_i 在 0.6 V 和 1.3 V 之间的范围变化时, VT_2 导通而 VT_5 截止。由于 VT_2 工作在放大区, 所以 U_{C2} 随 U_{B2} 升高而下降, 经 VT_4 射极跟随器使 u_o 下降, 故称 BC 段为线性段。

CD 段: 随着输入电压略微升高, 输出电压急剧下降, 这时 VT_5 开始导通, VT_2 尚未饱和, VT_2 、 VT_4 、 VT_5 均处于放大状态, 故 u_i 稍有提高, 就会使 u_o 迅速下降, 称 CD 段为线性转折段。

DE 段: 当 u_i 继续升高, 随着 u_i 增加 VT_1 进入倒置工作状态, $U_{B1} = 2.1$ V, 此时 VT_2 、 VT_5 饱和, VT_4 截止, 输出低电平 0.3 V, 且 u_o 基本上不随 u_i 的增大而变化, 称 DE 段为饱和段, 非门此时导通。

根据电压传输特性分析,当 $U_{CC}=5\text{ V}$ 时,TTL 非门有如下几个重要参数。

2) 阈值电压、输出高低电平、开门电平、关门电平和噪声容限

(1) 阈值电压 U_{TH} 。电压传输特性 CD 段中点所对应的输入电压称为阈值电压。阈值电压也称门檻电压, $U_{TH}=1.3\sim 1.4\text{ V}$ 。近似分析时,可以认为 $u_i > U_{TH}$ 时非门导通,输出低电平; $u_i < U_{TH}$ 时非门截止,输出高电平。

输出高电平 U_{OH} 和输出低电平 U_{OL} : U_{OH} 是电压传输特性曲线 AB 段所对应的输出电压, $U_{OH}=3.6\text{ V}$ 。 U_{OL} 是电压传输特性曲线 DE 段所对应的输出电压, $U_{OL}=0.3\text{ V}$ 。

TTL 器件一般要求 $U_{OH} \geq 3\text{ V}$, $U_{OL} < 0.4\text{ V}$ 。

(2) 开门电平 U_{ON} 和关门电平 U_{OFF} 。开门电平 U_{ON} 是指输出电平 $U_{OL}=0.3\text{ V}$ 时,允许输入高电平的最小值。在图 2.2.4 中的特性曲线上, U_{ON} 是指输出电压为 0.3 V 时所对应的输入高电平的最小值。只有当 $u_i \geq U_{ON}$ 时,输出才为低电平。 U_{ON} 典型值为 1.4 V ,一般产品要求 $U_{ON} \leq 1.8\text{ V}$ 。

关门电平 U_{OFF} 指的是在保证输出电压为额定高电平 U_{OH} 的 90% (即 2.7 V) 时,允许输入低电平的最大值。只有当 $u_i < U_{OFF}$ 时,输出才为高电平。一般产品要求 $U_{OFF} \geq 0.8\text{ V}$ 。

TTL 非门的额定高电平 $U_{OH}=3\text{ V}$,额定低电平 $U_{OL}=0.3\text{ V}$,这是由于器件的离散性引入的参数。

(3) 噪声容限 U_{NL} 和 U_{NH} 。实际应用中,由于外界磁场、相邻信号之间的相互耦合和电源波动等原因,可能使输入电压 u_i 偏离规定值。为保证电路可靠工作,对于干扰电压的限制,称为噪声容限,噪声容限也表示门电路的抗干扰能力,用 U_N 表示。噪声容限有低电平噪声容限 U_{NL} 和高电平噪声容限 U_{NH} 两个指标。

低电平噪声容限是指在保证输出高电平的前提下,允许叠加在输入低电平 U_{IL} 上的最大正向干扰电压,用 U_{NL} 表示

$$U_{NL}=U_{OFF}-U_{OL}$$

高电平噪声容限是指在保证输出低电平的前提下,允许叠加在输入高电平上的最大负向干扰电压,用 U_{NH} 表示

$$U_{NH}=U_{OH}-U_{ON}$$

若 $U_{OFF}=0.8\text{ V}$, $U_{OL}=0.3\text{ V}$,则 $U_{NL}=0.5\text{ V}$ 。若 $U_{ON}=1.8\text{ V}$, $U_{OH}=3.6\text{ V}$,则 $U_{NH}=1.8\text{ V}$ 。图 2.2.4 表明 U_{ON} 、 U_{OFF} 与 U_{NL} 、 U_{NH} 之间的关系。

3) 输入特性

输入特性指的是输入电流与输入电压之间的关系曲线,即 $i_i=f(u_i)$ 的函数关系,关系到前级电路连接的重要特性,如图 2.2.5 所示。设输入电流 i_i 流入 VT_1 发射极时方向为正,反之为负。

(1) 输入短路电流 I_{IS} 。输入短路电流 I_{IS} 指的是当 $u_i=0\text{ V}$ 时由输入端流出的电流。此时 A 信号撤掉,并接地,流出 VT_1 发射极的电流称为输入短路电流 I_{IS} ,如图 2.2.5(b) 所示。

输入短路电流 I_{IS} 的计算公式为

$$I_{IS}=-\frac{U_{CC}-U_{BE1}}{R_1}=-\frac{5-0.7}{4}\text{ mA}\approx -1.1\text{ mA}$$

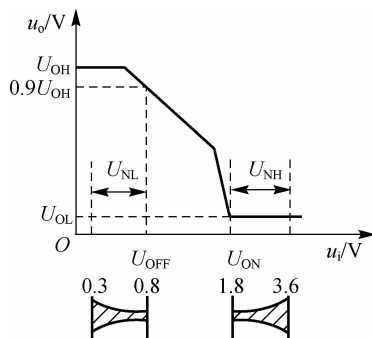


图 2.2.4 U_{ON} 、 U_{OFF} 与 U_{NL} 、 U_{NH} 之间的关系

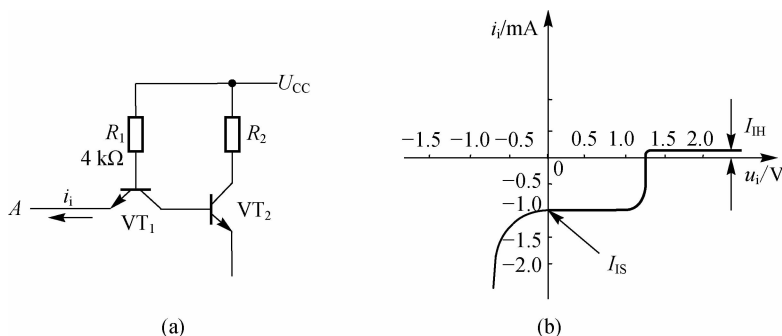


图 2.2.5 TTL 非门输入电路和特性曲线

(a)输入电路 (b)特性曲线

(2)低电平输入电流 I_{IL} 。低电平输入电流 I_{IL} 指的是当 $u_i = 0.3 \text{ V}$ 时,由输入端流出的电流。此时 A 信号接低电平,流出 VT_1 发射极的电流称为低电平输入电流 I_{IL} ,低电平输入电流 I_{IL} 的计算公式为

$$I_{IL} = -\frac{U_{CC} - U_{BE1} - 0.3}{R_1} \approx -1.0 \text{ mA}$$

显然 I_{IL} 的数值比 I_{IS} 的数值略大一些,近似分析时,可以用 I_{IS} 来代替 I_{IL} 。

(3)输入漏电流 I_{IH} (或称高电平输入电流)。输入漏电流 I_{IH} 指的是输入端接高电平 $U_{IH} = 3.6 \text{ V}$ 时, VT_1 管倒置,相当于集射极互换,此时电流从输入端流向本级电路。由于三极管工作在倒置状态,电流放大倍数很小,因此高电平输入电流也很小,约 $10 \mu\text{A}$ 。标准 TTL 门电路每个输入端的 $I_{IH} \leq 50 \mu\text{A}$ 。

4)输入负载特性

实际工程应用时常会遇到输入端经过一个电阻接地的情况,如图 2.2.6(a)所示。当输入电流流经 R_i 时,必然会在 R_i 上产生压降,从而形成输入电压 u_i 。 u_i 在一定范围内会随着 R_i 的增加而升高,形成 $u_i = f(R_i)$ 变化曲线,称为输入负载特性曲线,如图 2.2.6(b)所示。

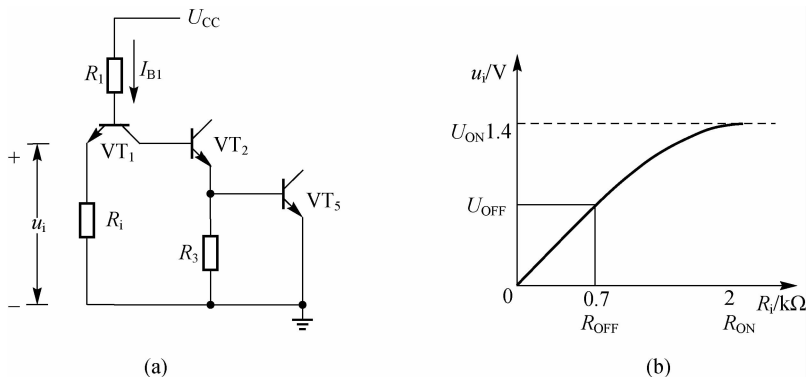


图 2.2.6 TTL 非门输入负载

(a)TTL 非门输入负载电路 (b)TTL 非门输入负载特性

从图 2.2.6(b)可见,当 R_i 较小时 ($0 \sim 0.7 \text{ V}$), u_i 与 R_i 成正比。由于 u_i 很小时, VT_2 、 VT_5 截止,忽略 VT_2 的反向基极电流,近似认为

$$u_i = \frac{R_i}{R_1 + R_i} (U_{CC} - U_{BE1})$$

为了保证电路稳定地输出高电平,必须使 $u_i \leq U_{OFF}$,即

$$u_i = \frac{R_i}{R_1 + R_i} (U_{CC} - U_{BE1}) \leq U_{OFF}$$

故

$$R_i \leq \frac{U_{OFF}}{U_{CC} - U_{BE1} - U_{OFF}} R_1$$

求出的电阻为关门电阻 R_{OFF} ,若 $U_{OFF} = 0.8 \text{ V}$, $R_1 = 4 \text{ k}\Omega$,则 $R_{OFF} \leq 0.9 \text{ k}\Omega$ 。若要使非门稳定在截止状态,输出高电平,必须取 $R_i < R_{OFF}$ 。

当 R_i 增大时, u_i 进一步增加,当 u_i 增大到 1.4 V 时, VT_2 、 VT_5 导通, U_{bi} 被钳位在 2.1 V ,致使 U_i 不会超过 1.4 V ,如图 2.2.6(b)所示。为保证电路稳定地输出低电平,应该有 $u_i \geq U_{ON}$ 。即

$$u_i = \frac{R_i}{R_1 + R_i} (U_{CC} - U_{BE1}) \geq U_{ON}$$

故

$$R_i \geq \frac{U_{ON}}{U_{CC} - U_{BE1} - U_{ON}} R_1$$

此时求得的输入电阻 R_i 称为开门电阻 R_{ON} 。若 $U_{ON} = 1.4 \text{ V}$, $R_1 = 4 \text{ k}\Omega$,则 $R_{ON} \approx 2 \text{ k}\Omega$ 。若要保证非门可靠导通,输出低电平,应选择 $R_i \geq R_{ON}$ 。

5) 扇入系数 N_i 和扇出系数 N_O

扇入系数 N_i 指的是输入端的个数,如一个二输入端的与非门,其扇入系数 $N_i = 2$ 。

扇出系数 N_O 衡量门电路带负载能力的大小。以同类门电路作负载时, N_O 表示可驱动同类门的个数。 N_O 分为两种情况,一种称为灌电流负载 N_{OL} ,另一种称为拉电流负载 N_{OH} ,如图 2.2.7 所示。

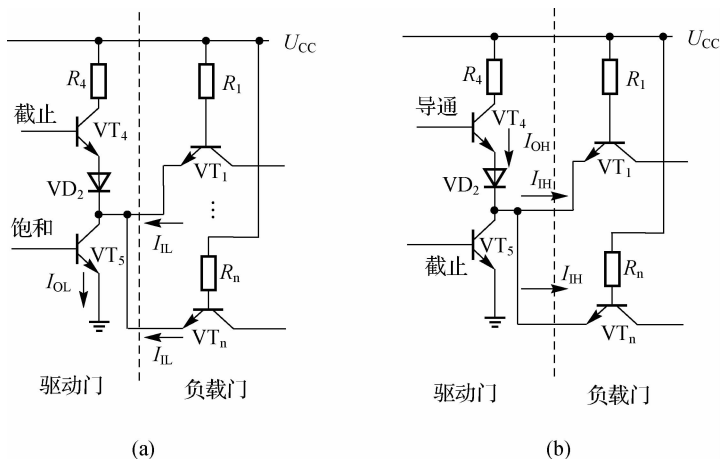


图 2.2.7 非门的带负载能力

(a) 外接灌电流负载 (b) 外接拉电流负载

当驱动门输出低电平,外接负载电流从外电路流入驱动门电路时,称为灌电流负载。当外接负载门的数量增多时,流入 VT_5 集电极电流增加,总灌电流将增加,同时引起输出低电平 U_{OL} 的略升。输出低电平 U_{OL} 只要不超过正常逻辑功能的允许值,就可以带数个同类门。

灌电流负载 N_{OL} 的计算公式为

$$N_{OL} = I_{OLmax} / I_{IL}$$

其中 I_{OLmax} 为驱动门的最大允许灌电流, I_{IL} 是一个负载门灌入本级的电流。 N_{OL} 限制了接入负载门的个数, N_{OL} 越大, 说明门的灌电流负载能力越强。

当驱动门输出高电平时, 负载电流从驱动门输出, 流向外接负载门, 称为拉电流负载, 如图 2.2.7(b) 所示。当外接负载门的数量增多时, 被拉出的电流增加, 总的拉电流将增加, 同时引起输出高电平 U_{OH} 的下降。输出高电平 U_{OH} 只要不超过正常逻辑功能的允许值, 就可以带数个同类门。拉电流负载 N_{OH} 的计算公式为

$$N_{OH} = I_{OHmax} / I_{IH}$$

其中 I_{OHmax} 为驱动门的最大允许拉电流, I_{IH} 是负载门高电平输入电流。

通常 N_{OL} 与 N_{OH} 不相等, 取较小的作为门电路的扇出系数 N_O 。在 TTL 集成器件的数据手册中, 一般不给出扇出系数, 因为有的时候需要外接不同的门电路, 就需要进行计算求得, 在实际工程应用时应当注意留有一定的裕量。

6) 功耗

功耗是门电路的重要参数之一, 功耗有空载导通功耗 P_{ON} 和空载截止功耗 P_{OFF} 两个参数。

空载截止功耗 P_{OFF} 指的是输出端开路、输入端接地或接低电平、图 2.2.1 中 VT_5 截止时的电源电流 I_{CCH} 与电源电压 U_{CC} 的乘积。标准 TTL 芯片有 $I_{CCH} \leq 5 \text{ mA}$, 所以 $P_{OFF} \leq 25 \text{ mW}$ 。

空载导通功耗 P_{ON} 指的是输出端开路、输入端悬空或接高电平、 VT_5 导通时的电源电流 I_{CCL} 与电源电压 U_{CC} 的乘积。标准 TTL 芯片有 $I_{CCL} \leq 10 \text{ mA}$, 所以 $P_{ON} \leq 50 \text{ mW}$ 。

显然, 可以看出 $P_{OFF} \leq P_{ON}$ 。

7) 平均传输延迟时间

在 TTL 电路中, 由于三极管的结电容和输入、输出端的寄生电容使输出波形发生了畸变和延迟, 并由于基极电荷的消散, 使较为理想的输入矩形波的电压信号通过门电路变为比输入信号滞后、波形上升沿和下降沿不再陡峭的输出电压。称输出波形滞后输入波形的时间为平均传输延迟时间, 它是表示门电路开关速度的参数。TTL 传输延迟时间如图 2.2.8 所示, 从输入波形上升沿的 50% 幅值处到输出波形下降沿 50% 幅值处所需时间, 称为导通延迟时间 t_{PHL} ; 从输入波形下降沿 50% 幅值处到输出波形上升沿 50% 幅值处所需时间, 称为截止延迟时间 t_{PLH} 。通常 $t_{PLH} > t_{PHL}$ 。两者的平均值称为平均传输延迟时间 t_{pd} , 即

$$t_{pd} = \frac{t_{PLH} + t_{PHL}}{2}$$

t_{pd} 越小, 电路的开关速度越高。因此, 平均传输延迟时间反映了集成电路的工作速度。

平均传输延迟时间不仅仅取决于电路的结构和制作工艺, 还取决于电路的分布参数, 往往是通过实验的方法得到, 不依赖于计算。一般 TTL 非门的 $t_{pd} = 10 \sim 40 \text{ ns}$ 。

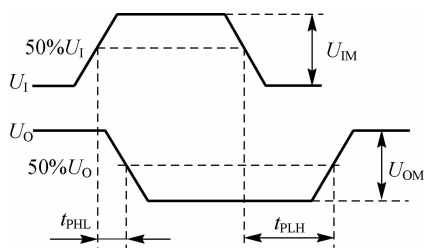


图 2.2.8 非门电路的传输延迟时间

2.2.2 基本和复合 TTL 逻辑门

在集成电路中,除了非门以外还有其他由基本门电路复合的门电路,如与非门、或非门、异或门和同或门等几种常见的门电路。这里只介绍这些电路,其他逻辑门的详细内容请读者阅读 TTL 器件手册。

1. TTL 与非门电路

图 2.2.9 为标准 TTL 与非门电路图。与图 2.2.1 比较,可以看出结构相似,仅输入级 VT_1 管由一个发射极变为两个发射极,显然输入信号由单一 A 增加到 A 和 B 。

根据与非逻辑运算关系,可以将输入端 A 、 B 的输入信号分为两种情况,一是 A 、 B 全接高电平,另一种情况是 A 、 B 中至少有一个低电平。现按此两种情况分别讨论 TTL 与非门的工作原理。

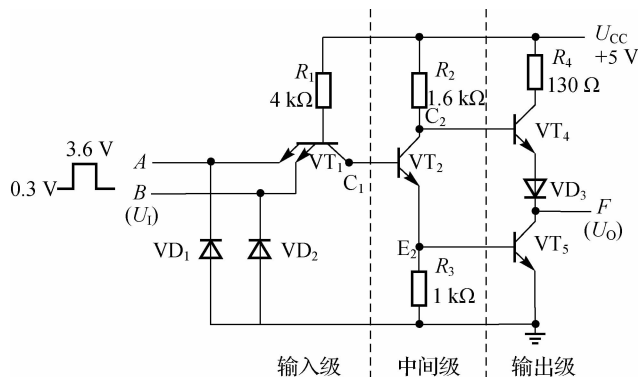


图 2.2.9 TTL 与非门电路

1) 输入端全部接高电平

当输入端 A 、 B 全部接高电平(3.6 V)时,分析可参考非门的高电平输入的情况。 VT_5 处于深度饱和状态,门电路输出端 F 为低电平 $U_{OL} = U_{CES5} \approx 0.3 \text{ V}$ 。

2) 输入端至少有一个接低电平

当输入端 A 、 B 至少有一个接低电平(0.3 V)时,分析可参考非门的低电平输入的情况。当 A 或 B 有一个输入低电平时, VT_1 导通,基极电压为 1 V,不足以使 VT_5 导通, VT_5 截止。输出端 F 为高电平 $U_{OH} = U_{CC} - U_{BE4} - U_{VD3} \approx (5 - 0.7 - 0.7) \text{ V} = 3.6 \text{ V}$ 。

综上所述,当输入端全部接高电平时,输出为低电平;当输入端至少有一个接低电平时,输出为高电平。由此可见,电路的输出和输入之间满足与非逻辑关系

$$F = \overline{AB}$$

电路逻辑符号如图 2.2.10 所示。

晶体管 VT_1 采用多发射极,可以加速 VT_2 管脱离饱和状态。当输入端全为高电平时, VT_2 管饱和,基区存储了大量的载流子,当 VT_1 的输入端突然变为低电平时, VT_1 管的基极电流 I_{b1} 流向低电平输入端,在瞬间产生 VT_1 正向基极电流的同时,也产生很大的集电极电流 I_{C1} , I_{C1} 恰好为 VT_2 和 VT_5 提供了很大的反向基极电流,使 VT_2 基区的存储电荷迅速消散,因而使 VT_2 迅速脱离饱和状态,进入截止状态,相对于图 2.2.1 所示的非

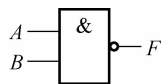


图 2.2.10 与非门的逻辑符号

门电路提高了开关速度。

发射极的数量还可以增加,但不能无限地增加,有关知识请参考其他文献。

2. TTL 或非门电路

图 2.2.11 是或非门标准电路。图中 R_1 、 VT_1 、 VT_2 构成的电路和 R_{11} 、 VT_{11} 、 VT_{12} 构成的电路完全相同, VT_2 和 VT_{12} 对应的集电极与发射极并联。

当 $A=1$ 时, VT_2 、 VT_4 同时导通, VT_3 截止, 输出 $F=0$; 当 $B=1$ 时, VT_{12} 、 VT_4 同时导通, VT_3 截止, 输出 $F=0$ 。只有当 $A=B=0$ 时, VT_2 和 VT_{12} 同时截止, VT_4 截止, VT_3 、 VD_3 导通, 输出 $F=1$, 所以电路实现或非逻辑功能, 其表达式为 $F=\overline{A+B}$ 。

通过上述分析可知, 或非门的或运算是通过 VT_2 和 VT_{12} 对应的集电极与发射极并联实现的。

3. TTL 与或非门电路

图 2.2.12 是 74 系列与或非门标准电路。不难看出, 该电路只是将图 2.2.11 或非门电路中的每个输入端改用多发射极三极管。由于三极管多发射极之间, 即 A 、 B 之间或 C 、 D 之间实现与逻辑运算, 因此图 2.2.12 可以实现与或非逻辑运算。其表达式为 $F=\overline{AB+CD}$ 。

4. TTL 异或门电路

图 2.2.13 是 TTL 异或门电路。当 $A=B=0$ 时, VT_2 、 VT_3 导通, VT_4 、 VT_5 截止, VT_7 、 VT_9 导通, VT_8 、 VD_3 截止, 因此 $F=0$; 当 $A=B=1$ 时, VT_1 、 VT_2 、 VT_3 倒置, VT_6 、 VT_9 导通, VT_8 、 VD_3 截止, 因此 $F=0$ 。

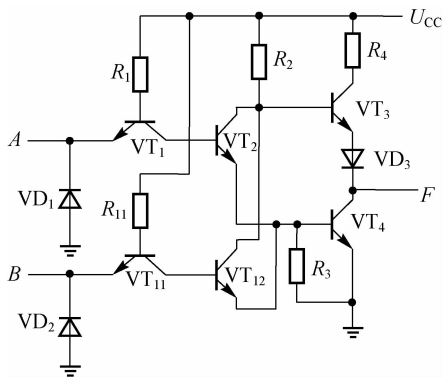


图 2.2.11 TTL 或非门电路

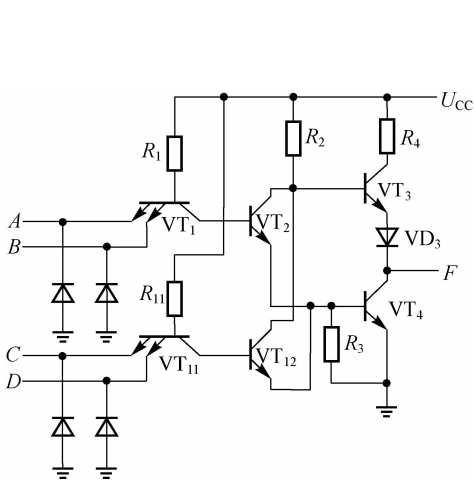


图 2.2.12 TTL 与或非门电路

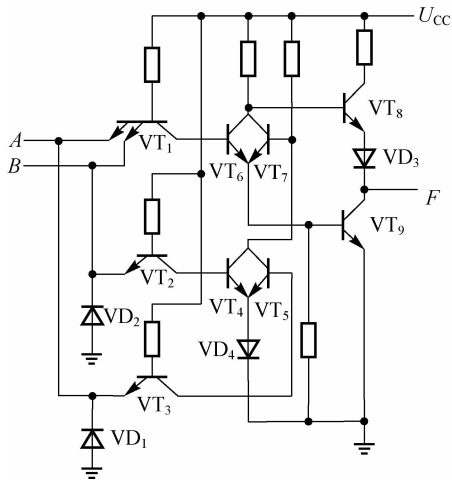


图 2.2.13 TTL 异或门电路

当 $A=0, B=1$ 或 $A=1, B=0$ 时, VT_1 导通, VT_6 截止; VT_4 、 VT_5 必有一个导通, VT_7 截止。由于 VT_6 、 VT_7 同时截止, 因此 VT_9 截止, VT_8 、 VD_3 导通, 所以 $F=1$ 。

因此, 输出 F 与输入 A 、 B 之间实现异或逻辑, 表达式为 $F=A\bar{B}+\bar{A}B=A\oplus B$ 。

为方便实现各种逻辑功能的电路和增加驱动能力, 实现工程上的应用, 下面介绍集电极

开路门电路,这些都是在 TTL 与非门基础上构成的,其他产品可以查阅 TTL 器件的数据手册。

5. TTL 集电极开路与非门(OC 门)

在实际工程应用中,往往需要将两个以上的输出端连接在一起使用,称此种方法为“线与”。但前面介绍的普通 TTL 门电路由于采用推拉式输出级,无论输出高电平或输出低电平,输出电阻都很低,“线与”之后会产生很大的输出电流,甚至烧毁晶体管。所以不允许将几个门电路的输出端直接并联,实现与逻辑。

观察图 2.2.14 电路表示两个 TTL 与非门输出端直接并联的情况。虚线两侧逻辑门 1 (图 2.2.9 电路)的输出电路和逻辑门 2 的输出电路直接连接在一起。假定门 1 输出高电平(VT_4 导通),门 2 输出低电平(VT_5 导通),此时在 U_{CC} 与地之间形成一个低阻通路,产生一个很大的电流,电流流向是 $U_{CC} \rightarrow$ 门 1 的 $R_5 \rightarrow VT_4 \rightarrow$ 门 2 的 VT_5 ,该电流将会超出正常的工作电流,不仅抬高门 2 的输出低电平,而且会因功耗过大损坏器件。因此普通 TTL 电路绝不允许将输出端直接并联使用,集电极开路与非门(简称 OC 门)可以解决这一问题。

1) 电路结构与工作原理

图 2.2.15(a)为典型 OC 门电路图,图 2.2.15(b)为逻辑符号,符号中的图标“ $\diamond$ ”表示集电极开路。由图可知,去掉普通 TTL 门中 VT_3 、 VT_4 管, VT_5 管的集电极开路,则构成集电极开路与非门。

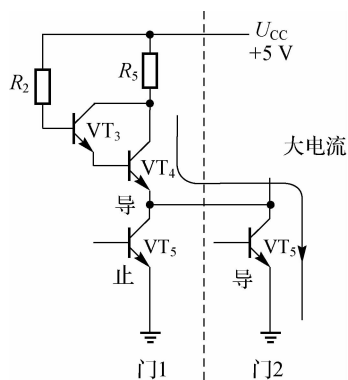


图 2.2.14 普通与非门电路输出端并联

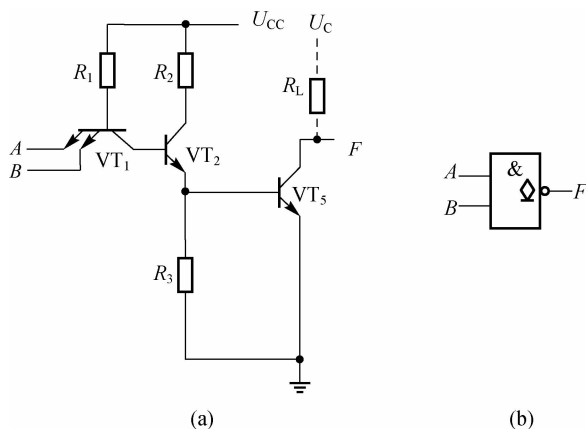


图 2.2.15 集电极开路与非门

(a) 电路图 (b) 逻辑符号

使用 OC 门时,需要外接一个电源 U_C ($U_C = 5 \sim 30 V$) 及电阻 R_L ,当输入端全为高电平时, VT_2 、 VT_5 导通,输出 F 为低电平;输入端有一个为低电平时, VT_2 、 VT_5 截止,输出 F 为高电平,接近电源电压 U_C ,因此 OC 门同样可以实现与非逻辑功能。只要电阻 R_L 阻值和电源电压 U_C 的数值选择得当,就能够做到既保证输出高、低电平的要求,输出三极管的负载电流又不过大。

为了实现“线与”逻辑功能,将 OC 门输出端直接并联,外接公共负载电阻 R_L 和电源 U_C ,如图 2.2.16(a)所示。 F_1 与 A、B 之间同样实现与非逻辑, F_2 与 C、D 之间实现与非逻辑,将 F_1 、 F_2 输出端直接连在一起,即“线与”,只要 F_1 、 F_2 有一个是低电平, F 就是低电平;

只有 F_1 、 F_2 同时为高电平, F 才是高电平, 因此 F 与 F_1 、 F_2 之间实现线与逻辑, 表达式为

$$F = F_1 \cdot F_2 = \overline{AB} \cdot \overline{CD} = \overline{AB + CD}$$

由上述表达式可知, 将两个 OC 结构的与非门线连接可以实现与或非的逻辑功能, 如图 2.2.16(b) 所示。

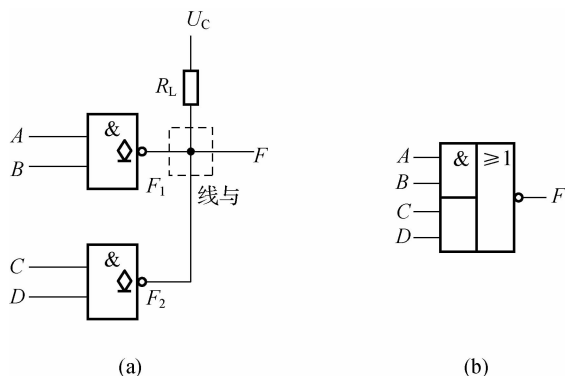


图 2.2.16 线与连接及其等效逻辑

(a) 线与逻辑图 (b) 等效逻辑

2) 外接负载 R_L

(1) 最大负载电阻 R_{Lmax} 。外接负载 R_L 是如何选取的呢? 观察图 2.2.17 为 n 个 OC 门输出端并联使用, 驱动 m 个普通 TTL 与非门的情况, R_L 可按下述原则选取。

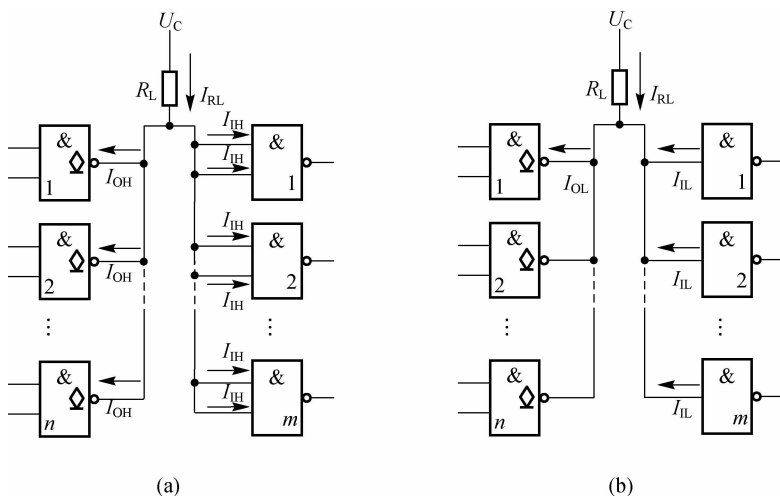


图 2.2.17 负载电阻 R_L 的计算

(a) 负载电阻 R_L 最大值的计算 (b) 负载电阻 R_L 最小值的计算

图 2.2.17(a) 为 n 个 OC 门“线与”, 且输出高电平, 驱动 m 个普通门。为了保证输出电压大于输出高电压的最小值 U_{OHmin} , R_L 不能选得过大。因此 R_L 最大值的计算为

$$R_{Lmax} = \frac{U_C - U_{OHmin}}{nI_{OH} + mI_{iH}}$$

式中, U_C 为外接电源电压; U_{OHmin} 为 OC 门额定高电平最小电压; I_{OH} 为 OC 门截止时的漏电

流; I_{IH} 为负载门输入端为高电平时的输入漏电流。

(2) 最小负载电阻 R_{Lmin} 。如图 2.2.17(b) 为只有一个 OC 门导通的情况, 此时灌电流最大, 为保证输出电流不超过最大允许电流 I_{OLmax} , R_L 不能选得过小。因此 R_L 最小值的计算为

$$R_{Lmin} = \frac{U_C - U_{OLmax}}{I_{OL} - mI_{IL}}$$

式中, U_{OLmax} 为 OC 门额定低电平最大电压; I_{OL} 为 OC 门最大允许灌电流; I_{IL} 为负载门输入短路电流。

综合以上两种情况, R_L 选取应满足表达式

$$R_{Lmin} < R_L < R_{Lmax}$$

设 $n=4, m=3, I_{OH}=50 \mu A, I_{IH}=40 \mu A, U_{OHmin}=2.4 V, U_{OLmax}=0.4 V, I_{IL}=1.4 mA, I_{OL}=16 mA, U_C=5 V$ 时, 由上述公式可求得 $R_{Lmax}=8.1 k\Omega, R_{Lmin}=390 \Omega$, 因此 R_L 取值范围为

$$390 \Omega < R_L < 8.1 k\Omega$$

上述方法仅仅确定了 R_L 的取值范围, R_L 的具体取值还要考虑电阻的标称值、电路的功耗和开关速度等因素。 R_L 越小, 电路的开关速度越快, 功耗越大。如果电路开关速度没有特殊要求, 可以将 R_L 取大些, 以降低电路的功耗。

3) OC 门的应用

集电极开路门除了可以实现多门的线与逻辑功能外, 还可以实现电平变换、驱动大电流负载等功能。在数字电子设备中, 常会碰到用门电路直接驱动较大电流负载的情况, 如驱动感性器件。利用 OC 门可以实现具有较大电流负载的驱动, 只要合理选择 U_C , 使驱动电流小于 OC 门中 T_5 所能承受的最大值即可。图 2.2.18 为驱动感性器件的电路连接, 其中图 2.2.18(a) 为驱动干簧继电器, 图 2.2.18(b) 为驱动脉冲变压器。

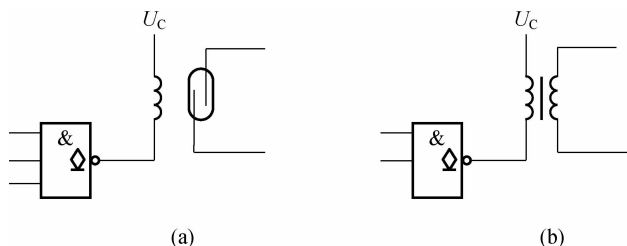


图 2.2.18 OC 门驱动感性负载

(a) 驱动干簧继电器 (b) 驱动脉冲变压器

6. TTL 三态输出逻辑门(TSL 门)

三态输出逻辑门除具有一般逻辑门输出的二值逻辑(高电平和低电平状态)外, 还具有第三种状态输出(高阻)。从前面定义的高电平和低电平的电压范围来看, 在三极管的工作区域分析不出第三种状态在什么电压范围。可以从另一个角度分析, 它不在一个电压范围定义之中, 而是无电压或者无电压范围, 可以将电路所处的这种状态称为高阻状态, 又称禁止态或失效态。三态门被广泛用于计算机的数据总线中。

1) TTL 三态门的工作原理

三态门是在一般与非门的基础上附加控制电路构成的。图 2.2.19(a) 是一个三态输出的与非门电路, 它是由两个门电路加上一个二极管 VD_2 组成的。图中虚线左半部分(VT_1 、 VT_2 、 VT_3 、 VT_4) 构成一个非门电路, 它是三态门的控制电路部分, 其输入端 $\bar{E}$ 称为使能端。

虚线右半部分为六管 TTL 与非门。

当 $\bar{E}$ 为高电平“1”时, VT_4 饱和导通, C 端输出低电平 0.3 V, 使 VT_6 、 VT_7 、 VT_{10} 截止。同时由于 VD_2 导通, 使 VT_6 的集电极 $U_{C6} = U_{D2} + U_{CE4} = (0.7 + 0.3) \text{ V} = 1 \text{ V}$, 不足以使 VT_9 导通, 此时三态门的输出管 VT_9 、 VT_{10} 均处于截止状态, 三态门的输出 F 端呈现开路, 输出既不是高电平, 也不是低电平, 这就是三态门的高阻状态, 记为 Z 。

当 $\bar{E}$ 为低电平“0”时, VT_4 截止, C 端相当悬空, 类似多发射极管 VT_5 是三输入与非门电路。当 A 和 B 都为高电平时, 使 VT_6 、 VT_{10} 饱和导通, VT_6 的集电极电压为 1 V, 使 VT_8 、 VT_9 截止, 输出低电平。当 A 和 B 至少有一个低电平时, 使 VT_6 、 VT_{10} 截止, VT_{10} 不通。由于 $U_{CC} \rightarrow R_6 \rightarrow VD_2$ 不能形成电流通路, 使 VD_2 截止, VT_8 、 VT_9 导通, 输出高电平。此时虚线右侧的电路执行正常的与非功能 $F = \overline{AB}$ 。由于输出端有三种可能的状态: 高电平、低电平、高阻, 故将这种门电路称为三态逻辑门。低电平使能的三态门逻辑符号如图 2.2.19(b) 所示。

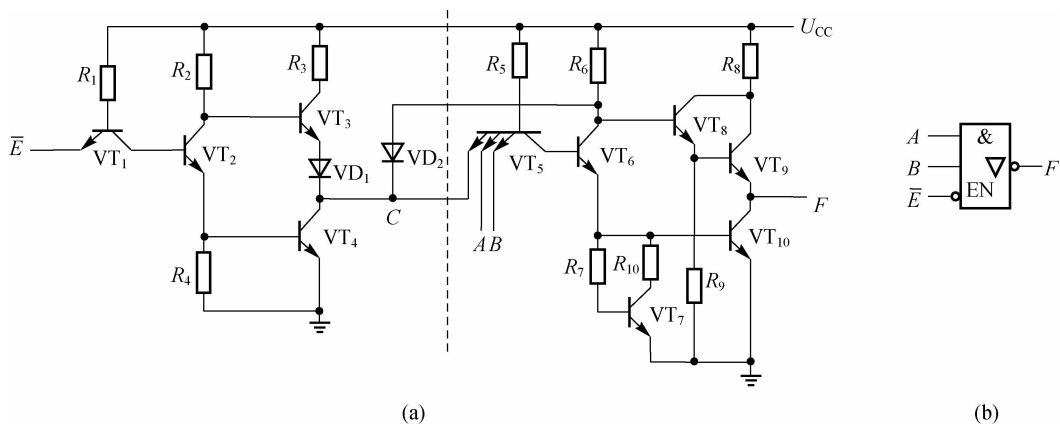


图 2.2.19 低电平使能 TTL 三态逻辑门电路与逻辑符号

(a) 低电平使能三态逻辑门电路 (b) 低电平使能逻辑符号

三态门的使能端有两种控制形式, 可以是低电平使能, 也可以是高电平使能。图 2.2.20(b) 所示逻辑符号是高电平使能。

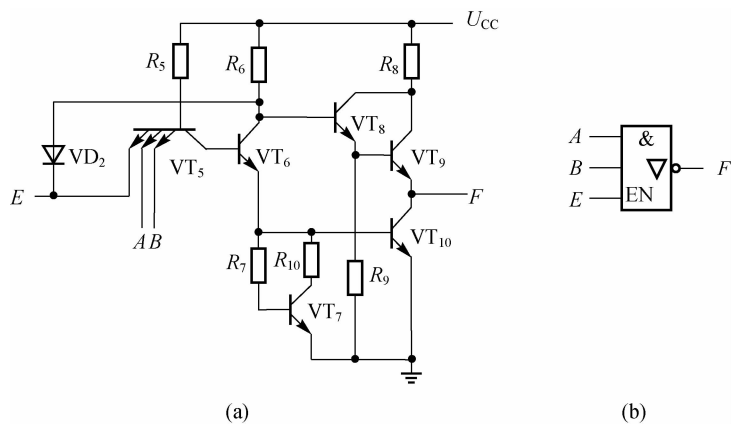


图 2.2.20 高电平使能 TTL 三态逻辑门电路与逻辑符号

(a) 高电平使能三态逻辑门电路 (b) 高电平使能逻辑符号

图 2.2.20(a)所示电路去掉了图 2.2.19(a)的控制电路(非门)。其工作原理的分析同低电平使能的三态门电路,不同的是图 2.2.20(a)电路为高电平使能三态门电路。

在工程应用中,前者的驱动能力大于后者。

低电平使能三态门电路和高电平使能三态门电路的逻辑表达式如下。

$$\text{低电平使能} \begin{cases} F = \overline{AB} | \overline{E}=0 \\ F = Z | \overline{E}=1 \end{cases}$$

$$\text{高电平使能} \begin{cases} F = \overline{AB} | E=1 \\ F = Z | E=0 \end{cases}$$

2) TTL 三态门的應用

TTL 三态门在实际的数字电路或复杂的数字系统中得到广泛的应用,它可以实现输出端的“线与”连接,广泛用于数据总线和双向数据传输。

(1)实现总线结构。在一些复杂的数字系统,特别是微型计算机系统中,为了减少各个单元电路之间连线的数目,希望能在同一数据通道上分时传输若干路数据,以减少数据传输线的数目。

图 2.2.21(a)为三态总线结构的连接方式,只要在工作时,保证任何时刻只有一个控制端有效,即只有一个门处于数据传输状态,其他门处于禁止状态,就可以将各路数据轮流送到公共的数据传输通道上(通常称为总线),各路数据之间互不干扰。

例 2.2.1 图 2.2.21(b)是不同时刻让不同的三态门只能一个三态门工作的分时输入波形。当 B_1 、 B_2 、 B_n 为高电平“1”时,试绘出总线(BUS)输出波形。

解:在 t_1 时刻,第 1 号三态门使能,将 A_1 信号送到总线。 t_2 时刻,第 2 号三态门使能,将 A_2 信号送到总线。最后第 n 号三态门使能,将 A_n 信号送到总线。请仔细观察图 2.2.21(b)中的 BUS 波形对应的其他波形。

(2)实现双向数据传输。利用三态门的逻辑功能可以实现数据的双向传输,如图 2.2.22 所示。当 $E=0$ 时,门 1 工作,门 2 禁止,数据从总线 A 传到总线 B;当 $E=1$ 时,门 1 禁止,门 2 工作,数据从总线 B 传到总线 A。

集成电路出现初期,各种逻辑门电路在速度、功耗和抗干扰能力等方面都存在着局限性,如 74 系列器件。为了满足实际应用中

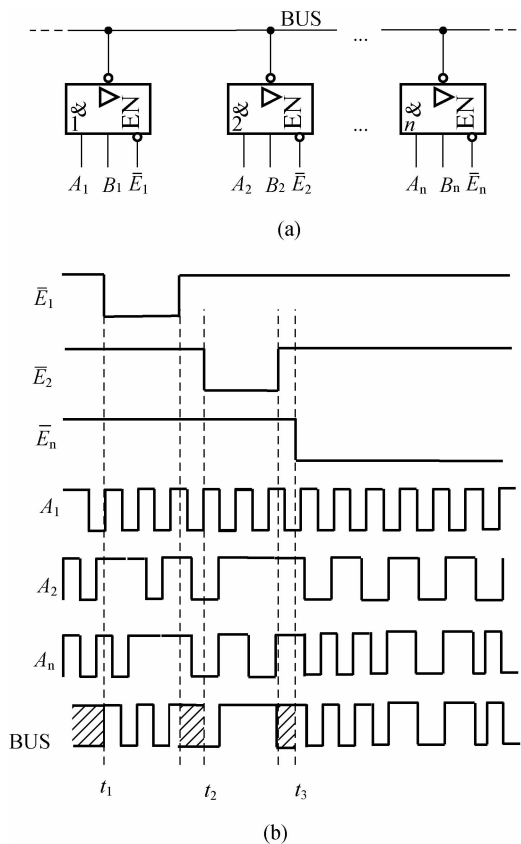


图 2.2.21 三态门实现总线结构与输入/输出波形图
(a)三态门实现总线结构 (b)输入/输出波形图

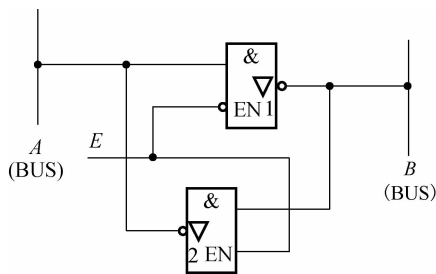


图 2.2.22 三态门实现双向数据传输

种要求,又相继出现了改进型 TTL 门电路对上述局限性进行了改进。如 74H 系列、74S 系列、74LS 系列等器件,下面介绍这些改进型 TTL 门电路。

7. TTL 快速系列(74H 系列)

在图 2.2.9 中,输出级采用了达林顿结构,将原电路输出级的 VT_4 和 VD_3 用复合管 VT_3 和 VT_4 代替,进一步减小门电路输出高电平时的输出电阻,提高对容性负载的充电速度。另外降低电路中所有电阻的阻值,缩短电路中各个节点电位的上升时间和下降时间,提高三极管的开关速度,电路如图 2.2.23 所示。

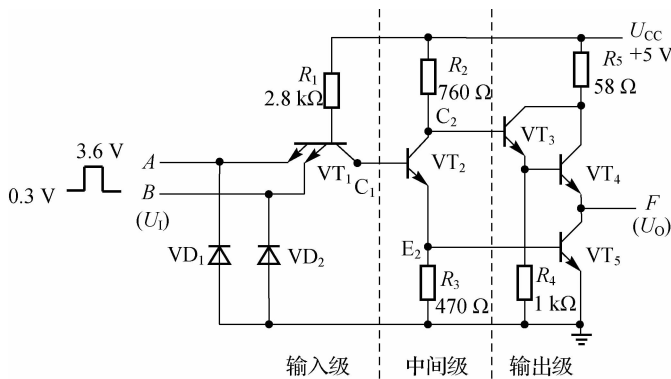


图 2.2.23 74H 系列与非门电路

快速 74H 系列门电路的传输时间比 74 系列减小了一半,但是由于前者电流的增大,使电路的功耗变大了。

8. TTL 肖特基系列(74S 系列)

与图 2.2.9 比较,一是增加了一个由晶体管 VT_6 、电阻 R_6 和 R_3 构成的有源泄放电路,用该电路代替 VT_2 射极电阻 R_3 。采用有源泄放电路的目的主要有两点,即提高工作速度和提高抗干扰能力。二是将标准 TTL 门电路中所有可能工作在饱和区的晶体管都改用图 2.1.3 所示的抗饱和三极管,即肖特基三极管代替。肖特基系列 TTL 与非门电路如图 2.2.24 所示。

当门电路由截止状态变为导通状态时,因为 VT_6 基极通过 R_3 接至 VT_2 发射极,而 VT_5 基极直接与 VT_2 发射极相连,所以 VT_5 比 VT_6 先导通, I_{c2} 全部流入 VT_5 基极,使 VT_5 迅速饱和,减少了电路的导通时间。当电路由导通状态变为截止状态时,由于 VT_6 基极没有存储电荷的泄放回路,而 VT_6 和 R_6 为 VT_5 的基区提供了一个存储电荷的泄放回路,即为 VT_5 基区提供了一个较大的反向驱动电流,使 VT_5 迅速脱离饱和状态,因此 VT_6 比 VT_5 截止滞后,缩短了电路截止时间,有效地提高了电路的工作速度。

在标准 TTL 电路中, VT_2 的 U_{BE} 达到 0.7 V 以上, VT_2 先于 VT_5 导通,出现电压传输特性曲线的线性区。而在改进型 TTL 电路中, VT_2 的 U_{BE} 必须达到 1.4 V 以上,才会使 VT_2 、 VT_5 同时导通,因此电压传输特性曲线线性区变窄,曲线变陡,低电平噪声容限 U_{NL} 提高了 0.7 V 左右,提高了电路的抗干扰能力。图 2.2.25 为 74S 系列与非门的电压传输特性曲线。由图可见,此时的开门电平 U_{ON} 、关门电平 U_{OFF} 和阈值电压 U_{TH} 基本重合,工程上认为是同一个值,即 $U_{ON}=U_{OFF}=U_{TH}=1.4$ V。

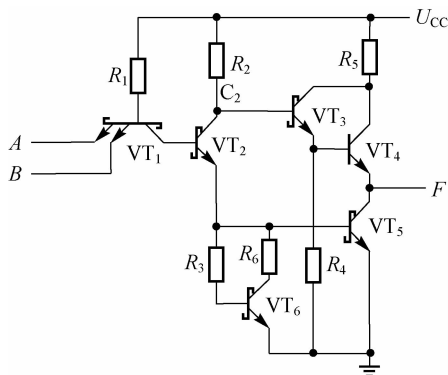


图 2.2.24 74S 系列与非门电路

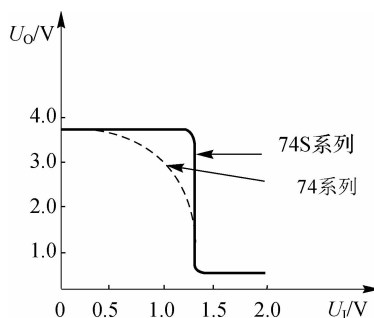


图 2.2.25 74S 系列与非门的电压传输特性

9. TTL 低功耗肖特基系列(74LS 系列)

74LS 系列与非门电路(见图 2.2.26)与图 2.2.9 与非门电路比较,电路中采用多种措施,以达到缩短传输延迟时间、降低功耗的目的。

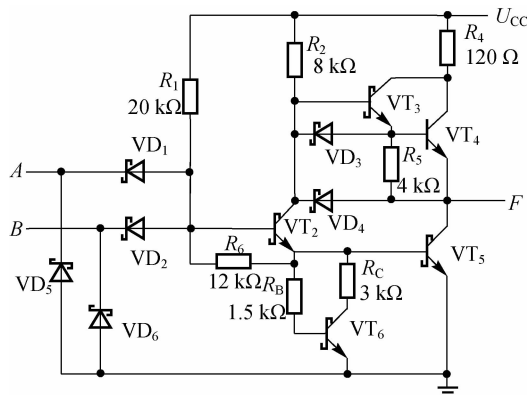


图 2.2.26 74LS 系列与非门电路

(1)为了缩短传输延迟时间,电路中使用肖特基管和有源泄放电路,另外,还将输入级的多发射极管改用 SBD 代替,由于 SBD 没有电荷存储效应,因此有利于提高电路的工作速度。电路中还接入了 VD_3 和 VD_4 两个 SBD,当电路的输出端由高电平变为低电平时, VD_4 经 VT_2 的集电极和 VT_5 的基极提供了一条通路,一是为了加快负载电容的放电速度,二是为了加速 VT_5 的导通过程。另外, VD_3 经 VT_2 的集电极为 VT_4 的基极提供了一条放电通路,加快了 VT_4 的截止过程。

(2)为降低功耗,提高了电路中各电阻的阻值,将电阻 R_5 原来接地的一端改接到输出端,以减小 VT_3 导通时电阻 R_5 上的功耗。

74LS 系列门电路的功耗仅为 74 系列的五分之一,传输延迟时间与 74 系列相当,具有较好的综合性能。

10. TTL 集成电路产品

TTL 集成电路产品很多,市场上常见的国际标准系列产品有 54/74 系列、54H/74H 系列、54S/74S 系列、54LS/74LS 系列、54AS/74AS 系列、54ALS/74ALS 系列和 54F/74F 系

列等。关于 TTL 系列产品的特点如表 2.2.2 所示。在各系列产品中,包含各种 TTL 逻辑门电路,如与非门、或非门、反相器、异或门和同或门等,还包含后面几章要学习的中规模集成电路触发器、计数器、寄存器等其他数字电路。

表 2.2.2 TTL 系列产品及其特点

系 列	名 称	特 点
54/74 系列	TTL 通用标准系列	TTL 最早产品,中速器件,目前仍使用
54H/74H 系列	TTL 快速系列	74 系列改进型,速度较 74 系列高,功耗大
54S/74S 系列	TTL 肖特基系列	采用肖特基晶体管和有源泄放回路,速度高,品种较 74LS 系列少
54LS/74LS 系列	TTL 低功耗肖特基系列	目前主要应用的产品,品种齐全,价格低廉
54AS/74AS 系列	TTL 先进的肖特基系列	74S 系列的改进产品,速度和功耗得到改进
54ALS/74ALS 系列	TTL 先进的低功耗肖特基系列	74LS 系列的改进产品,速度和功耗有较大改进,但品种少,价格略高
54F/74F 系列	TTL 高速系列	与 74ALS 及 74AS 产品相当,属高速型产品,品种较少

各种系列的 TTL 门电路平均传输延迟时间和平均功耗/每门比较如表 2.2.3 所示。

表 2.2.3 不同系列 TTL 门电路主要参数

参 数	CI74 74/54	CI74H 74H/54H	CI74S 74S/54S	CI74LS 74LS/54LS	74AS/ 54AS	74ALS/ 54ALS
平均传输延迟时间 t_{pd}/ns	10	6	3	9.5	1.5	4
平均功耗/每门 P/mW	10	22	19	2	20	1

不同系列的 TTL 器件,只要器件型号的功能编号一样,则表明它们的逻辑功能、引脚排列和外形尺寸完全相同。如 7400、74H00、74S00、74LS00 等,它们的逻辑功能相同,都是两输入端四与非门,引脚排列和外形尺寸也完全相同,所不同的只是电气特性,如传输延迟时间、功耗等。

TTL 集成电路的命名分为以下几个部分,规则如下:

例如:SN 74 LS 00 DIP

生产公司的名称:如 HD 表示日本 HITACHI 公司;SN 表示美国 Texas 公司。

适用范畴:74 表示民品和工业品,工作温度范围为 $0^{\circ}C \sim 70^{\circ}C$;54 表示军品,工作温度范围为 $-55^{\circ}C \sim 125^{\circ}C$ 。

产品的系列:如 H、S、LS、AS、ALS 和 F 等,缺省表示是标准系列。

集成电路逻辑功能编号:如 00 表示的是两输入四与非门。

产品封装形式:如 DIP 为双列直插式。

思考题

题 2.2.1 两输入异或电路的一个输入端接地,另一个输入端接信号,输出与输入的关系式为_____。

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

题 2.2.2 两输入异或电路的一个输入高电平,另一个输入端接信号,输出与输入的关系式为_____。

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

题 2.2.3 某逻辑门的输出相对于输入的平均延迟时间为多少? 已知 $t_{\text{PLH}}=5\text{ ns}$, $t_{\text{PHL}}=3\text{ ns}$ 。

题 2.2.4 典型 TTL 非门中的 VT_1 什么时候处于倒置状态?

- (A) 输入接高电平 (B) 输入接低电平 (C) 前级输出高电平

题 2.2.5 典型 TTL 非门中的电阻增加,功耗_____,速度_____。

题 2.2.6 TTL 三态门输出有三种状态,即_____,_____和_____。

题 2.2.7 普通 TTL 与非门输出不能“线与”的原因是什么? OC 门“线与”在输出的要求是什么?

题 2.2.8 三态门输出能“线与”吗? 如果能“线与”,条件是什么?

题 2.2.9 74LS 系列和 74 系列相比性能有哪些改进?

2.3 其他类型门电路

TTL 逻辑门电路中晶体管工作在截止区和饱和区,即使是速度较快的改进型的肖特基逻辑门电路也是工作在截止区和浅饱和区,电路的工作速度受到了限制。如果将电路中的晶体管工作状态由饱和改为非饱和,使晶体管不进入饱和区,也是提高电路的工作速度的方法之一。发射极耦合逻辑(简称 ECL)门电路可以实现这样的非饱和和高速数字集成电路的功能。

2.3.1 发射极耦合逻辑(ECL)门电路

发射极耦合逻辑电路是一种非饱和型高速数字集成电路,其平均传输延迟时间在 2 ns 以下。ECL 是目前唯一能提供亚毫微秒开关时间的实用电路,主要应用于每秒运算百万次以上的大型高速计算机、数字通信系统、高精度测量设备以及频率合成器等方面。

1. ECL 门电路工作原理

图 2.3.1 是典型 ECL 或/或非门逻辑电路,由于电路中 VT_3 管的输入信号是通过发射极电阻 R_5 耦合过来的,故发射极耦合逻辑电路名称由此而得。

ECL 或/或非门电路由三部分构成:偏置电路级、差分输入级和射极跟随输出级。正常工作时, VT_3 管基极提供基准电压 $U_{\text{BB}}=-1.29\text{ V}$,输入信号的高低电平分别为 -0.8 V 和 -1.6 V 。

1) 偏置电路

偏置电路由 VT_4 、 VD_1 、 VD_2 和电阻 R_6 、 R_7 、 R_8 组成。为使电路在输入高、低电平时具有相同的噪声容限, VT_3 管基极的基准电压 U_{BB} 选输入高低电平的平均值,通过合理设计电

路中电阻 R_6 、 R_7 、 R_8 的参数,使 U_{BB} 为 -1.29 V 。在 VT_4 基极回路中的二极管 VD_1 、 VD_2 用来对 VT_4 的发射结进行温度补偿,由于 VT_4 、 VD_1 、 VD_2 具有相同的温度特性,来补偿由温度引起 U_{BE4} 的变化。

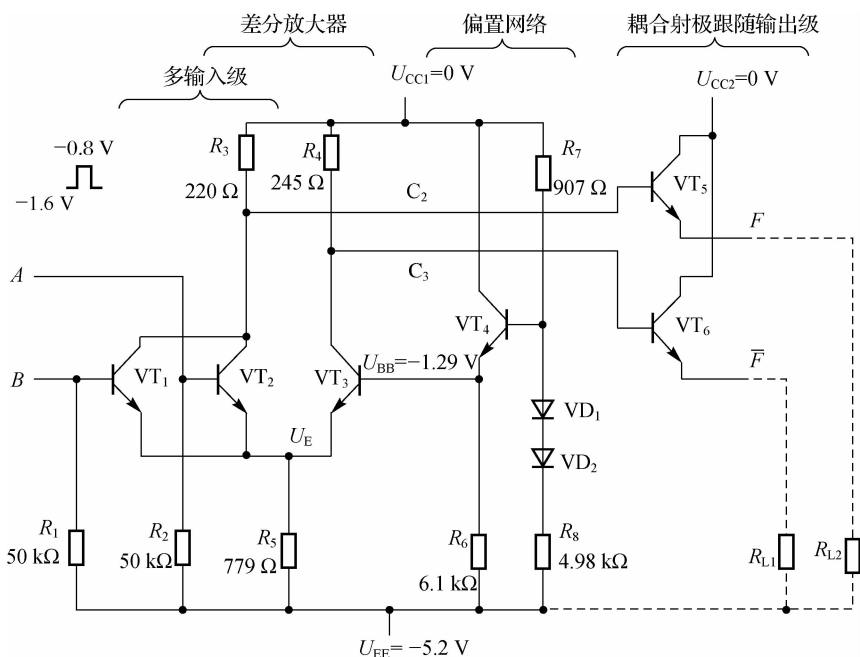


图 2.3.1 ECL 或/或非门电路

2) 输入级

VT_1 、 VT_2 两个输入管组成两个输入端, VT_3 加有固定偏压 $U_{BB} = -1.29\text{ V}$, 组成差分输入。

当输入 A 和 B 全为低电平 -1.6 V 时, 由于 U_{BB} 电平等于 -1.29 V , VT_3 的电位高于 VT_1 和 VT_2 的电位, 所以 VT_3 管先导通, 使发射极电位 $U_E = -1.29\text{ V} - 0.7\text{ V} = -1.99\text{ V}$ 。这时输入管 VT_1 和 VT_2 发射结压降只有 0.39 V , 故各管截止。流过 VT_1 和 VT_2 的电流几乎为 0 , 其集电极电阻 R_3 上的压降很小可以被忽略。若输出电阻 R_{L1} 、 R_{L2} 连接到 -5.2 V , $F = -U_{BE5} = -0.7\text{ V}$, 输出为高电平。

流过 R_5 的电流由导通的 VT_3 管提供, 忽略 VT_6 基极电流, $i_{R5} = (U_E - U_{EE})/R_5$, 可以求出 VT_3 集电极电压为 $i_{R5}R_4 = -1\text{ V}$, $\bar{F} = -1.7\text{ V}$, 输出为低电平。

当输入端有一个高电平时, 设 A 为高电平, 则 VT_2 管基极电压为 -0.8 V , 高于基准电压 U_{BB} , 因此 VT_2 先于 VT_3 导通, 使发射极耦合电压 U_E 为 $-0.8\text{ V} - 0.7\text{ V} = -1.5\text{ V}$ 。此时 VT_3 管发射结上的压降为 $-1.29\text{ V} + 1.5\text{ V} = 0.21\text{ V}$, 因此 VT_3 管截止。忽略 R_4 上的压降, 使 $\bar{F} = -0.7\text{ V}$, 输出为高电平。同样可分析 $F = -U_{BE5} = -1.7\text{ V}$, 输出为低电平。由于 VT_1 、 VT_2 两管输入回路并联, 输入 B 为高电平, A 为低电平, 分析方法类似。

综上所述, 输出 F 和 $\bar{F}$ 与 A 、 B 之间的逻辑关系为或非逻辑和或逻辑关系, 即

$$F = \overline{A+B} \quad \bar{F} = A+B$$

电路中, R_1 和 R_2 为下拉电阻, 其作用是给未使用的输入管提供低电平, 保证其可靠截止, 防止干扰。

3) 输出级

射极输出器 VT_5 和 VT_6 组成输出级, 下拉电阻 R_{L1} 、 R_{L2} 与发射极之间是开路的。射极输出器有两个作用: 一是实现前后级隔离, 增加驱动能力; 二是实现电平转换, 将输出管 VT_5 和 VT_6 基极电位的高、低电平转换成 -0.7 V 和 -1.7 V 。

电路驱动负载时, 如果负载较小, 可以将输出端 F 、 $\bar{F}$ 分别与 R_{L1} 、 R_{L2} 相连, 以获得规定的输出电平。负载较大时, 输出电平已能满足要求, 输出端与 R_{L1} 、 R_{L2} 断开连接。这样做既可以方便使用, 又能降低功耗。

另外经分析计算, 观察到无论 VT_1 、 VT_2 、 VT_3 哪个晶体管导通, 导通管的集电极电压总是高于基极电压, 集电结反偏, 晶体管处于非饱和状态, 即放大状态。

由此可见, 该 ECL 电路具有或/或非逻辑功能, 逻辑符号如图 2.3.2 所示。

2. ECL 逻辑门电路的工作特点

与 TTL 门电路相比, ECL 门电路有其突出的特点。

1) 开关速度高

目前 ECL 电路是工作速度最快的数字集成电路之一。

(1) ECL 电路中的三极管工作在放大区或截止区, 可以消除由于三极管饱和带来的基极消散电荷时间。

(2) ECL 电路中的电阻取值小, 高、低电压的差值比 TTL 小得多, 可以缩短电路的上升时间和下降时间。

(3) ECL 电路的输出方式采用射极输出器, 其输出电阻小, 使负载电容充电的时间常数减小, 因此开关速度高。

目前 ECL 传输延迟时间已做到 0.1 ns 以内。

2) 逻辑功能强

ECL 门电路具有或/或非互补输出端, 且采用射极开路形式, 允许多个输出端直接并联, 实现输出变量的线或逻辑功能。图 2.3.3 利用两个 ECL 门实现线或的电路连接。

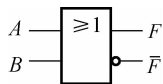


图 2.3.2 ECL 或/或非门逻辑符号

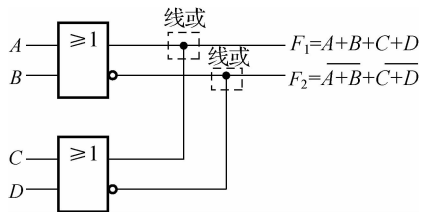


图 2.3.3 ECL 门电路的线或连接

3) 负载能力强

输入级的射极输入电阻 R_5 具有很强的电流负反馈作用, 所以输入阻抗高, 输入电流小。ECL 门电路采用射极输出, 电路输出阻抗小, 输出电流大。因此电路的扇出系数大, N_O 可达 100 以上。但是扇出系数增加时, 负载电容也会增加, 将使电路失去高速的优越性, 因此实际应用时扇出系数一般不超过 10。

虽然 ECL 电路具有很多优点, 但使用 ECL 门电路时, 还应注意 ECL 电路的一些不足。

1) 功耗大

ECL 门电路功耗为输入级、基准电源和输出级三部分之和。为了提高工作速度, 电路中

电阻值较小,而且三极管又工作在非饱和区,所以 ECL 电路的功耗比 TTL 大,每门平均功耗达 40 mW,可见 ECL 的高速是以增加功耗为代价的。

2)抗干扰能力差

ECL 电路的逻辑摆幅约 1 V,直流噪声容限 U_N 约 300 mV,因此抗干扰能力差。

3)输出电平稳定性差

由于 ECL 电路中的三极管导通时工作在放大区,电路的输出电平与 VT_5 、 VT_6 的发射结压降有关,所以输出电平的变化与温度变化和电路参数的变化直接相关。

2.3.2 集成注入逻辑(I²L)门电路

集成注入逻辑电路简称 I²L,该电路结构简单、功耗低,适合于大规模集成电路的制造,可满足高密度大规模集成电路制造的需要。

1. I²L 非门电路的工作原理

图 2.3.4(a)所示是 I²L 非门电路结构。电路由一个 PNP 管 VT_1 和一个 NPN 型多集电极管 VT_2 组成。 VT_1 构成恒流源电路, VT_2 构成反相器, VT_2 各集电极之间相互隔离。通过 R_E 和 R_C 连接到电源 U_E 。

电源 U_E 通过 R_E 向 VT_1 发射极流入电流,经 VT_1 的集电极流向 A 或流到 VT_2 的基极,流到 VT_2 的基极称注入 VT_2 。故称注入逻辑电路。由于 PNP 管 VT_1 的发射极接固定电压 U_E ,基极接地,所以 VT_1 工作在恒流状态,恒流用 I_O 表示。

输入信号端 A 接到 VT_2 管的基极,其为低电平(0.1 V)时, I_O 从输入端 A 流出, VT_2 截止, C_1 、 C_2 和 C_3 输出高电平。当输入 A 为高电平(0.7 V)或开路时, I_O 流入 VT_2 的基极,使 VT_2 饱和导通,输出 C_1 、 C_2 和 C_3 为低电平,约 0.1 V。由此可见,电路的任何一个输出与输入之间都是非逻辑关系。简化逻辑符号的画法如图 2.3.4(b)所示,还可以进一步将射极和地简化掉。

VT_1 管也做成多集电极,以使用一个多集电极 PNP 管驱动多个 NPN 管,在实际的集成电路设计中,实现大规模集成电路的集成。

2. 其他 I²L 门电路

1)I²L 与非门

图 2.3.5 所示的 I²L 电路是与非门电路,与图 2.3.4(a)进行比较,将注入极电源 U_E 作为信号输入 B 使用,输入信号 A 连接不变,则图 2.3.5 电路实现与非逻辑运算。

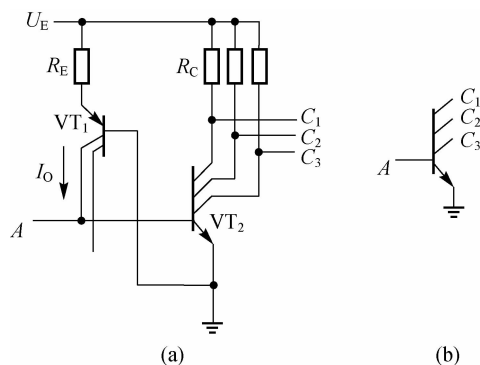


图 2.3.4 I²L 非门电路
(a)电路原理图 (b)简易画法

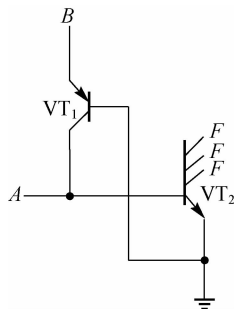


图 2.3.5 I²L 与非门电路

分析图 2.3.5 所示电路,当 A 、 B 均为高电平时,分析和 I^2L 非门相同, VT_1 、 VT_2 导通,输出低电平 0.1 V。当输入 B 高, A 低时的分析和非逻辑相同,输出高电平。当输入 B 为低电平时, VT_1 截止,无法向 VT_2 的基极提供驱动电流,故 VT_2 也截止,使 F 输出高电平。因此 F 与 A 、 B 之间为与非逻辑关系。

2) I^2L 与或非门

将两个 I^2L 与非门输出线与,就可以实现与或非逻辑,如图 2.3.6 所示。逻辑关系表达式为 $F = \overline{AB} \cdot \overline{CD} = \overline{AB + CD}$ 。

3) I^2L 异或门

如图 2.3.7 所示电路用了 6 个 I^2L 基本单元,两两非门输出组成“线与”,经过逐步分析,可实现异或逻辑。逻辑关系表达式为 $F = \overline{AB} \cdot \overline{\overline{AB}} = \overline{\overline{AB} \cdot \overline{AB}} = \overline{AB + \overline{AB}}$ 。

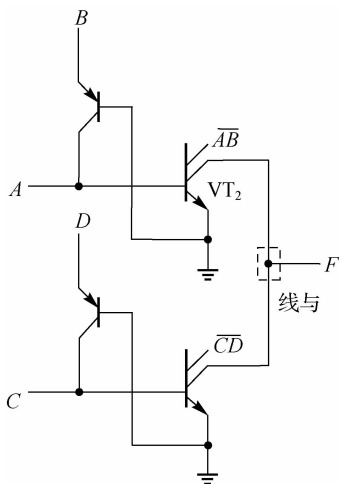


图 2.3.6 I^2L 与或非门

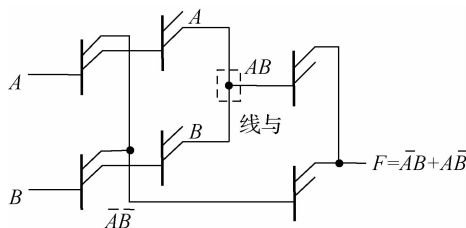


图 2.3.7 I^2L 异或门

从 I^2L 的各种门电路组成结构可以看到, I^2L 电路结构有效地压缩了门的个数与级数。

3. I^2L 门电路的主要特点

集成注入逻辑(I^2L)电路具有如下特点。

(1) I^2L 电路能在低电压 0.8 V (VT_2 的发射结导通)、微电流 (1 nA/单元) 情况下工作,是目前功耗最低的集成电路。

(2) I^2L 门电路结构紧凑、简单,电路中只包含 NPN 和 PNP 管。各逻辑单元之间不需要隔离,电路结构集成度高,简化了工艺,节省了芯片面积。 I^2L 的基本逻辑单元面积仅为 TTL 的十分之一。

(3) I^2L 门电路的功耗与速度二者之积(品质因数) $M = Pt_{pd}$ 是目前最好的。

(4) I^2L 只需四次光刻,两次扩散,比常规的 TTL 要少两次光刻两次扩散,而工艺简单,成品率高。在工艺上与 TTL、ECL 和 MOS 等电路集成在同一芯片上,提供了方便的兼容性。

尽管 I^2L 电路在上述方面具有很多优点,但也有不足的一面。

I^2L 属于饱和型电路,使电路的开关速度较低, I^2L 反相器的传输时间 t_{pd} 一般为 20~30 ns。

其次 I^2L 的高、低电平的差值仅 700 mV 左右,噪声容限 U_N 不大于 250 mV,比 ECL 还

低,抗干扰能力较差。

另外,由于各管子输入特性的离散性,多个 I^2L 芯片一起使用时,基极电流分配会出现不均的现象,使多片连接性能差,严重时电路无法正常工作。

思考题

题 2.3.1 在 ECL 电路中的输入差分三极管,经原理分析后,不管哪个管导通,其集电极电压值相对于基极电压值_____。

- (A)大 (B)小 (C)相等 (D)反偏

题 2.3.2 ECL 门电路和 TTL 门电路相比的主要优点是_____。

- (A)速度快 (B)便宜 (C)低功耗 (D)工艺简单

题 2.3.3 ECL 门电路的电压摆率为_____。

- (A)0.8 V (B)3.3 V (C)5 V (D)30 V

题 2.3.4 I^2L 两输入与非门中输入 A 为高电平时,流经 VT_1 的电流为 0, VT_2 _____。

题 2.3.5 I^2L 的主要性能可选以下哪些?

- (A)抗干扰能力强 (B)功耗低 (C)结构简单 (D)6 次光刻,4 次扩散

2.4 CMOS 逻辑门电路

TTL、ECL 和 I^2L 等集成逻辑门是由双极型晶体管组成,电流控制三极管的导通与闭合,属于电流控制器件。晶体管中的空穴和电子两种载流子参与导电,故称为双极型器件。

本节将介绍的 CMOS 逻辑门电路只有一种载流子(电子或者空穴)参与导电,故称为单极型器件。MOS 集成电路主要包括 NMOS 电路、PMOS 电路以及互补 MOS 电路,即 CMOS 电路。MOS 集成电路和双极型 TTL 器件相比,各有特色,但前者优势大于后者。前者具有功耗低、工艺简单、抗干扰能力强、集成度高等特点。

目前,NMOS 门电路与 PMOS 门电路逐渐被 CMOS 门电路所代替,CMOS 具有更快的速度,更低的功耗,所以 CMOS 集成技术已经成为数字电路的主流技术。故教材中不再介绍 NMOS 电路和 PMOS 门电路。

2.4.1 CMOS 基本单元

1. CMOS 反相器

1) CMOS 反相器结构

CMOS 反相器电路如图 2.4.1 所示。它是由 PMOS 和 NMOS 两管组成的互补型 MOS 电路。其中 VT_1 为 NMOS 管, VT_2 为 PMOS 管,两管电气特性完全对称。两管的栅极相连作为反相器的输入端,漏极相连作为反相器的输出端, VT_2 源极 S_2 接 U_{DD} , VT_1 源极 S_1 接地。为使衬底与漏源之间的 PN 结始终处于反偏, NMOS 管的衬底总是接到电路的最低电位, PMOS 管的衬底总是接到电路的最高电位。NMOS 管的栅源开启电压 $U_{VT1} > 0$, PMOS 管的栅源开启电压 $U_{VT2} < 0$ 。为了使 CMOS 电路能正常工作,要求电源电压 $U_{DD} > (U_{VT1} + |U_{VT2}|)$, U_{DD} 取值范围较大,可为 3~18 V。

CMOS 管的栅极与衬底沟道之间的一层 SiO_2 绝缘层(栅氧化层)很薄,厚度约为

$0.1\ \mu\text{m}$ 。栅氧化层的击穿电压值为 $100\sim 200\ \text{V}$,电阻却高达 $10^{12}\ \Omega$,因此只要有少量的电荷,就可能在栅极氧化层感应出强电场,造成栅氧化层永久击穿。如果在 CMOS 输入端加上保护电路,栅氧化层便可得到保护,不被击穿。图 2.4.2 所示电路是具有保护电路的 CMOS 反相器。图中 VD_1 、 VD_2 为双极型二极管,正向导通压降约为 $1\ \text{V}$,反向击穿电压约为 $30\ \text{V}$ 。输入电阻 R 一般为 $1\sim 3\ \text{k}\Omega$ 。 $\text{VD}_1\cdots\text{VD}'_1$ 和 R 为分布式二极管,其结构是制作电阻 R 时 P 型区和 N 型衬底间自然形成的。 C_1 为 PMOS 管栅极等效电容, C_2 为 NMOS 管栅极等效电容。

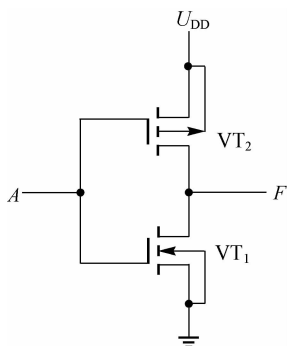


图 2.4.1 CMOS 反相器

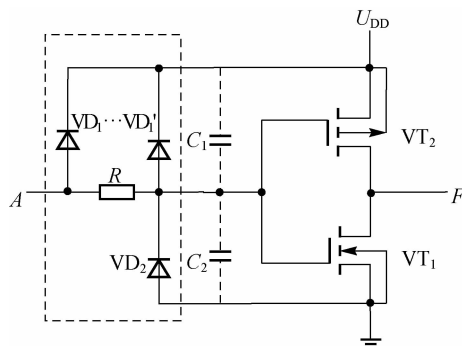


图 2.4.2 带输入保护电路的 CMOS 反相器

2) CMOS 反相器工作原理

在图 2.4.1 中,当输入为高电平($U_{\text{IH}}=U_{\text{DD}}$)时, $U_{\text{GS1}}>U_{\text{VT1}}$, VT_1 导通, VT_2 截止,输出为低电平($U_{\text{OL}}\approx 0\ \text{V}$)。同理,当输入为低电平($U_{\text{IL}}=0\ \text{V}$)时, $U_{\text{GS1}}<U_{\text{VT1}}$, VT_1 截止,内阻很大。而 $|U_{\text{GS2}}|>|U_{\text{VT2}}|$, VT_2 导通,内阻很小。电路中电流近似为零(忽略 VT_1 的截止漏电流),输出为高电平($U_{\text{OH}}\approx U_{\text{DD}}$)。可见电路实现非逻辑功能,即

$$F=\bar{A}$$

对于 CMOS 反相器,不论输入高电平还是低电平, VT_1 和 VT_2 总是一个工作在导通状态,一个工作在截止状态,即两管工作在互补状态。因此,该电路称为互补型 MOS 电路。

因为 CMOS 反相器工作时两管中总有一个是导通的,所以电路输出阻抗很小,有效地减小了对负载电容的充放电时间,因此 CMOS 门电路工作速度快,甚至可以同 TTL 门电路媲美。又因为 CMOS 电路工作时,总有一管截止,因此电源静态电流非常小,电路静态功耗极低,一般在纳瓦数量级。此外由于 CMOS 门输入阻抗高,所以 CMOS 电路级联时扇出系数很大。

在图 2.4.2 保护电路中,输入电压在 $0\sim U_{\text{DD}}$ 时,保护电路均不导通,起不到作用。此时流过二极管的电流为 0,按图 2.4.1 电路原理正常逻辑工作。若 MOS 管受到静电干扰或正负尖峰脉冲,二极管起到了钳位保护作用,MOS 管不会损坏。当输入信号电压大于电源电压和二极管正向导通电压($u_i>U_{\text{DD}}+U_{\text{D}}$)时,输入保护二极管 VD_1 导通,流过二极管 VD_1 的电流迅速增加。当输入信号电压小于二极管正向导通电压($u_i<-U_{\text{D}}$)时,输入保护二极管 VD_2 导通,流过二极管 VD_2 的电流迅速增加。电流随输入信号电压 u_i 的增大而增大。如果电流继续增大,首先二极管被击穿,如果时间较短,二极管可以恢复正常工作。如果时间较长,电压过大,二极管还是可以被损坏,进而击穿 MOS 管栅极。

电阻 R 和 MOS 管的栅极电容组成积分电路,延迟了时间和减缓了幅度。但 R 过大将

损坏输入信号的质量,所以 R 制作时要小一些。

3) CMOS 反相器传输特性

在图 2.4.1 所示的 CMOS 反相器电路中,设 $U_{DD} > (U_{VT1} + |U_{VT2}|)$, VT_1 和 VT_2 的参数完全对称,即 $U_{VT1} = |U_{VT2}|$, VT_1 和 VT_2 的导通内阻和截止内阻相同,则电压传输特性曲线如图 2.4.3 所示。电压传输特性曲线大致分为三个部分:AB 段、BC 段和 CD 段。

反相器工作在特性曲线的 AB 段时,由于 $u_i = U_{GS1} < U_{VT1}$, $|U_{GS2}| > |U_{VT2}|$,故 VT_1 截止, VT_2 导通。输出高电平 ($U_{OH} \approx U_{DD}$)。

在特性曲线的 CD 段,由于 $u_i = U_{GS1} > U_{VT1}$,故 VT_1 导通。又由于 $u_i > U_{DD} - |U_{VT2}|$,使 $|U_{GS2}| < |U_{VT2}|$,故 VT_2 截止。输出低电平 ($U_{OL} \approx 0$ V)。

在特性曲线的 BC 段,由于输入电压的工作区间为 $U_{VT1} < u_i < U_{DD} - |U_{VT2}|$,此时, $U_{GS1} > U_{VT1}$, $|U_{GS2}| > |U_{VT2}|$,故 VT_1 和 VT_2 同时导通。在 VT_1 和 VT_2 管参数完全对称,导通内阻相同的情况下,如 $u_i = \frac{1}{2}U_{DD}$,则 $u_o = \frac{1}{2}U_{DD}$,工作在电压传输特性转折区的中点。由传输特性曲线可知,CMOS 反相器的阈值电压 $U_{TH} \approx \frac{1}{2}U_{DD}$,具有较大的噪声容限。另外转折区的变化率很大,所以 CMOS 反相器更接近于理想开关特性。

4) CMOS 反相器噪声容限

随着电源电压 U_{DD} 的增加,噪声容限也相应地变大。在 U_{DD} 固定的情况下,低电平噪声容限 U_{NL} 和高电平噪声容限 U_{NH} 始终相等。国产 4000 系列 CMOS 电路的测试结果表明, $U_{NL} = U_{NH} \geq 30\%U_{DD}$ 。

为了提高 CMOS 反相器的噪声容限,可以通过适当提高电源电压 U_{DD} 的方法实现,这在 TTL 门电路中是不可能的。

5) CMOS 反相器传输延迟时间

尽管 CMOS 反相器在工作过程中不会像 TTL 电路那样发生载流子的聚集和消散,但是由于集成电路内部电阻、容性负载的影响等多种原因,CMOS 反相器仍然会产生传输延迟。

由于 CMOS 反相器的输出电阻比 TTL 门电路的输出电阻大,所以容性负载对传输延迟时间会产生更大的影响。

由于 CMOS 反相器的互补对称性,当反相器接容性负载时,它的导通延迟时间 t_{PHL} 和截止延迟时间 t_{PLH} 是相等的。CMOS 反相器的平均传输延迟时间约为 10 ns。

2. CMOS 传输门(TG)

CMOS 传输门与 CMOS 反相器一样,也是构成各种逻辑电路的一种基本单元电路。CMOS 传输门电路图和逻辑符号如图 2.4.4(a)、图 2.4.4(b)所示,它由 PMOS 管和 NMOS 管连接而成。

图 2.4.4(a)中 VT_1 是 NMOS 管, VT_2 是 PMOS 管, NMOS 管的栅源开启电压为 U_{VT1} , PMOS 管的栅源开启电压为 U_{VT2} , 设

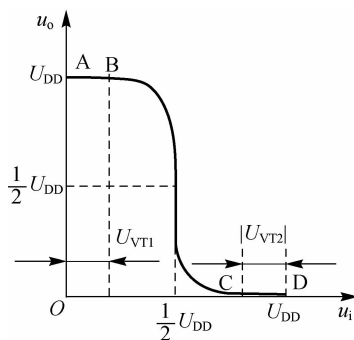


图 2.4.3 CMOS 反相器电压传输特性

$$U_{DD} > (U_{VT1} + |U_{VT2}|)$$

VT_1 和 VT_2 的参数对称。 C 、 $\bar{C}$ 是一对互补的电压控制信号, C_L 为负载电容。因为 MOS 管源极和漏极的对称性, 所以 CMOS 传输门的输出与输入端可以互换。一般输入电压变化范围为 $0 \sim U_{DD}$, 控制电压为 0 或 U_{DD} 。

当控制端电压 $C = 0 \text{ V}$, $\bar{C} = U_{DD}$, 输入信号 u_i 在 $0 \sim U_{DD}$ 范围内变化时, VT_1 和 VT_2 均截止, 输入与输出之间呈高阻, 输入与输出之间相当于开关断开, C_L 上的电平保持不变, 这种状态称为传输门截止, 可以实现信息的保存。

当控制端电压 $C = U_{DD}$, $\bar{C} = 0 \text{ V}$ 时, u_i 在 $0 \sim (U_{DD} - U_{VT1})$ 范围变化时 VT_1 导通; u_i 在 $|U_{VT2}| \sim U_{DD}$ 范围变化时 VT_2 导通, 即 u_i 在 $0 \sim U_{DD}$ 范围变化时, VT_1 和 VT_2 至少有一个导通, 使输入与输出之间呈低阻, 相当于开关接通, 使输入电压 u_i 的变化传到输出端, 即 $u_o = u_i$, 这种状态称为传输门导通, 可以实现信息的传输。

由于 MOS 管的对称性, 其源极和漏极可以互换, 输入和输出端可以互换使用, 因此 CMOS 传输门是双向器件。传输门的导通电阻为几百欧姆, 当它与输入阻抗为兆欧级电路连接时, 可以忽略不计。传输门的截止电阻达 $50 \text{ M}\Omega$ 以上, 每个门的平均延迟时间为几十至一二百纳秒, 已接近理想开关特性。

利用 CMOS 传输门和 CMOS 反相器可以组成模拟开关, 如图 2.4.5 所示。反相器的输入和输出提供传输门两个互补控制信号 (C 和 $\bar{C}$)。当控制端 $C = 1$ 时, 模拟开关接通; $C = 0$ 时, 模拟开关断开。它可以传输幅值在 U_{CH} (控制电压高电平) 和 U_{CL} (控制电压低电平) 之间任意大小的模拟电压。

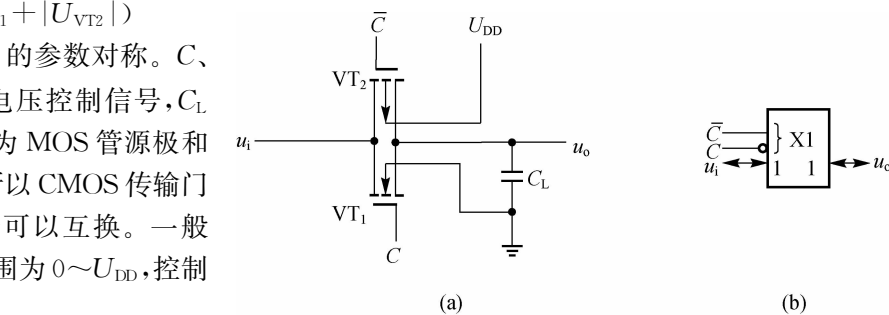


图 2.4.4 CMOS 传输门

(a) 电路图 (b) 逻辑符号

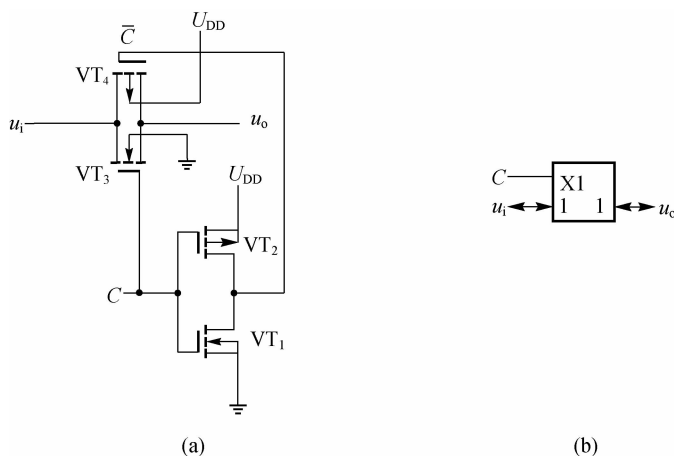


图 2.4.5 CMOS 模拟开关

(a) 电路图 (b) 逻辑符号

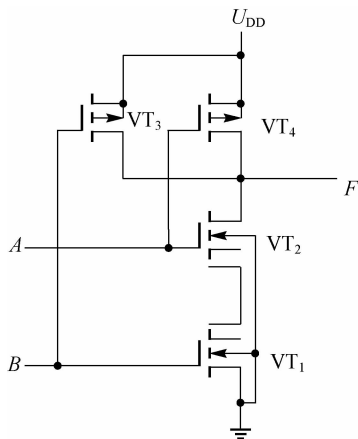
模拟开关是一种电子开关,控制模拟电压信号的传输,模拟开关的接通与断开由数字信号控制,广泛应用于数字系统。

2.4.2 其他 CMOS 逻辑门电路

在 CMOS 门电路的数据手册中,除了反相器外,还有与 TTL 门相类似的多种门电路,如与非门、或非门、异或门、同或门和三态门等。

1. CMOS 与非门和或非门电路

(1)图 2.4.6 为两输入 CMOS 与非门逻辑原理电路,负载部分和驱动部分分别由两个并联的 PMOS 管 VT_3 、 VT_4 和两个串联的 NMOS 管 VT_1 、 VT_2 组成。当输入端 A 和 B 为高电平时, VT_1 和 VT_2 导通而 VT_3 和 VT_4 截止,输出低电平。当输入端 A 和 B 有一个或一个以上为低电平时,两个 NMOS 管至少有一个管截止,两个 PMOS 管至少有一个导通,输出高电平,所以该电路实现与非逻辑,即 $F = \overline{AB}$ 。



A	B	VT_1	VT_2	VT_3	VT_4	F
0	0	断	断	通	通	1
0	1	断	通	断	通	1
1	0	断	断	通	断	1
1	1	通	通	断	断	0

图 2.4.6 CMOS 与非门

(2)图 2.4.7 为两输入 CMOS 或非门逻辑原理电路。两个串联的 PMOS 管 VT_3 、 VT_4 构成负载部分,而两个并联的 NMOS 管 VT_1 、 VT_2 构成驱动部分。当输入 A 和 B 都为低电平时, VT_1 和 VT_2 截止, VT_3 和 VT_4 导通,输出高电平。当输入 A 和 B 有一个或一个以上为高电平时,两个 PMOS 管至少有一个截止,两个 NMOS 管至少有一个导通,输出低电平。所以该电路实现或非逻辑,即 $F = \overline{A+B}$ 。

(3)图 2.4.6 与非门逻辑原理电路中,输入端的扇入系数直接影响输出电压值(高、低电平),扇入系数为 n ,就是 n 个串联的驱动管,输出低电平的电压值就是 n 个 NMOS 管导通时 U_{DS} 压降之和。若 n 个输入全是低电平时,就是 n 个负载管并联,使负载电阻变小,输出高电平电压值变高。为了保持和前后级电路的电平电压的一致性,在每个输入和输出端增加一个 CMOS 反相缓冲器。

但增加缓冲器后,电路的逻辑输出也发生了变化。图 2.4.7 或非门电路加上反相缓冲器后为图 2.4.8,图 2.4.6 与非门电路加上反相缓冲器后为图 2.4.9。还可以用正、负等价互换原则分析,将与非逻辑换成或非逻辑,或非逻辑换成与非逻辑。图 2.4.8(a)中非虚线部分是或非逻辑,分析电路逻辑为

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

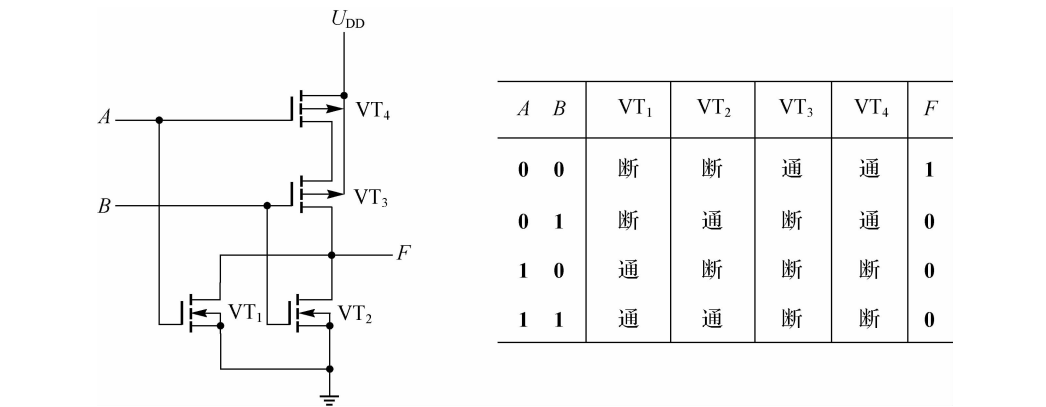


图 2.4.7 CMOS 或非门

逻辑电路如图 2.4.8(b)所示,等效逻辑如图 2.4.8(c)所示。增加了缓冲器逻辑门之后,整体电路的输出高、低电平不受扇入系数的影响,和前后级高、低电平电压值保持一致。

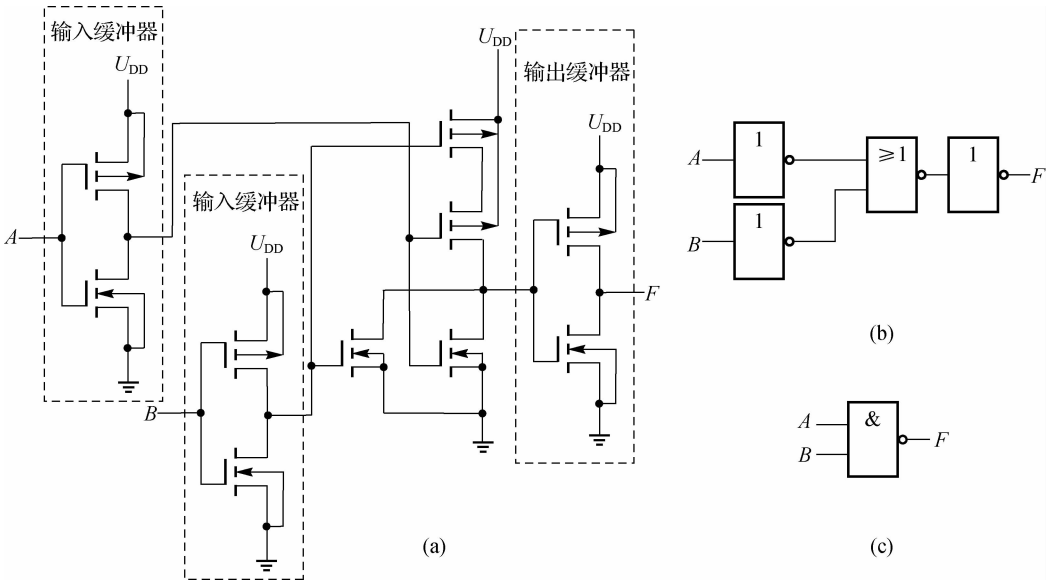


图 2.4.8 带输入/输出缓冲器 CMOS 与非门

同理得到图 2.4.9 或非门逻辑电路图,分析其电路逻辑为

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

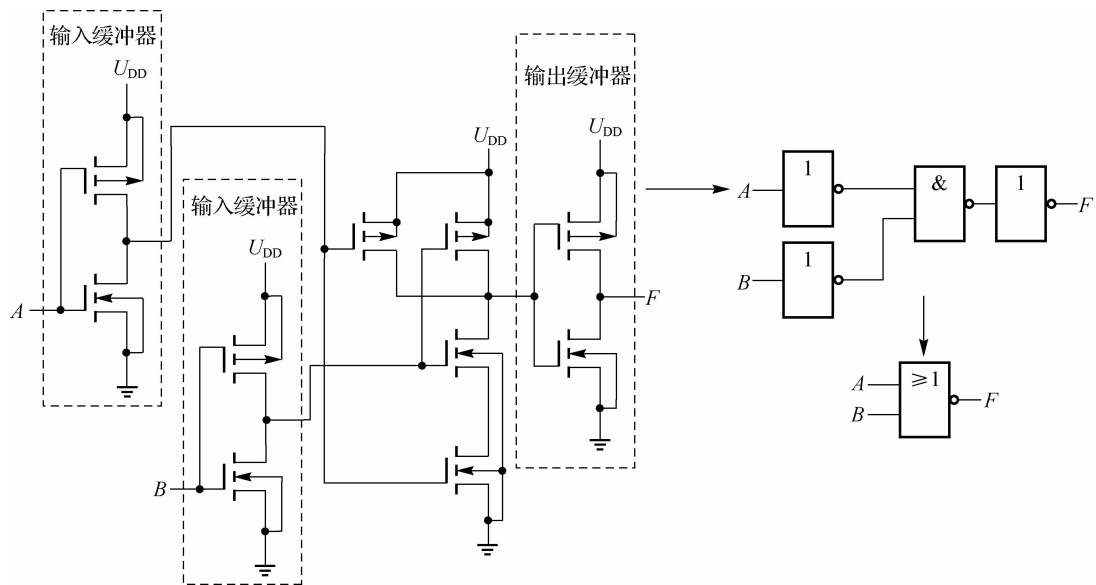


图 2.4.9 带输入/输出缓冲器 CMOS 或非门

2. CMOS 同或门

CMOS 同或门如图 2.4.10 所示,由三个 CMOS 反相器和一个 CMOS 传输门组成。传输门的互补控制信号是 A 和 $\bar{A}$ 。

1)输入端 A 和 B 信号相同

当 $A=B=0$ 时,反相器 2 由于受到 A 和 $\bar{A}$ 的控制,使反相器 2 的两只 MOS 管都截止断开,反相器的输出是高阻。传输门工作,则 $C=B=0, F=\bar{C}=1$ 。

当 $A=B=1$ 时,传输门 TG 断开,反相器 2 和普通反相器一致,可导通,正常工作, $C=\bar{B}=0, F=\bar{C}=1$,如图 2.4.10(a)所示。

2)输入端 A 和 B 信号相异

当 $A=0, B=1$ 时,传输门 TG 导通,反相器 2 输出处于高阻态, $C=B=1, F=\bar{C}=0$ 。

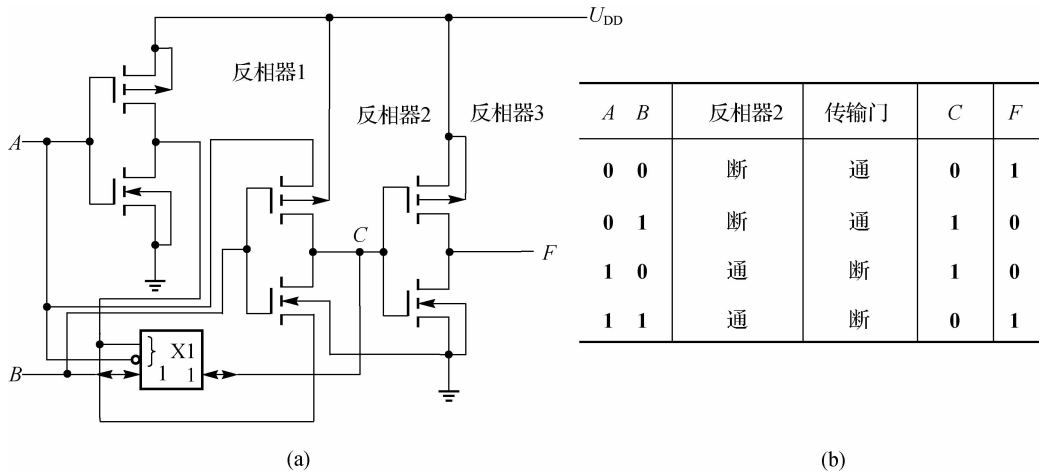


图 2.4.10 CMOS 同或门

(a)CMOS 同或门电路 (b)CMOS 同或门真值表

当 $A=1, B=0$ 时, 传输门 TG 断开, 反相器 2 和普通反相器一致, 可导通, 正常工作, 输出 $C=\overline{B}=1, F=\overline{C}=0$ 。

CMOS 同或真值表如图 2.4.10(b) 所示。

综上所述, 该电路实现同或逻辑, 即 $F=A \odot B = \overline{A}\overline{B} + AB$ 。

3. CMOS 2 选 1 数据选择器

CMOS 2 选 1 数据选择电路如图 2.4.11 所示, 由一个 CMOS 反相器和两个 CMOS 传输门组成。

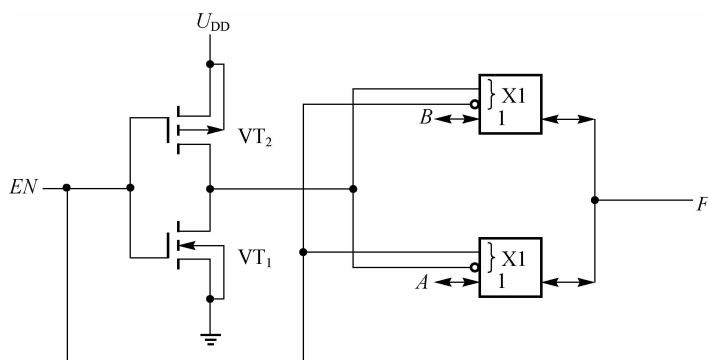


图 2.4.11 2 选 1 数据选择电路

A 和 B 为输入信号, EN 和 $\overline{EN}$ 是传输门的互补控制信号。

当 EN 为“1”时, F 选择 A 作为输出信号, 使 $F=A$; 当 EN 为“0”时, F 选择 B 作为输出信号 $F=B$, 具有这样功能的电路称为数据选择器。也有 4 选 1 选择器和 8 选 1 选择器等。

4. CMOS 漏极开路门电路 (OD 门)

两个普通 CMOS 与非门输出端直接并联如图 2.2.14 连接一样。电源、导通的 PMOS、导通的 NMOS 和地之间形成一个低阻通路, 产生一个很大的电流, 就像普通 TTL 电路并联一样。由于回路电流将会超出正常的工作电流, 会因功耗过大损坏器件。并且抬高输出电平, 无法确定高电平还是低电平。

为了实现输出并联(线与), 将 CMOS 上方的 PMOS 去掉, 或者将 NMOS 上的增强型有源负载或电阻去掉, 如图 2.4.12 虚线内所示, 称此电路为漏极开路输出 (OD 门)。

除了具有输出并联(线与)功能外, 还可以驱动大电流负载和输出电平转换。OD 门在使用时, 外串接一个上拉电阻。上拉电阻性能分析和计算可根据 OC 门分析计算方法得到。

5. CMOS 三态门

与 TTL 三态门一样, CMOS 三态门在基本逻辑门电路基础上, 增加控制电路构成, 如图 2.4.13 和图 2.4.14 所示。

当图 2.4.13 中的使能信号 $\overline{EN}$ 为低电平时, VT_3 导

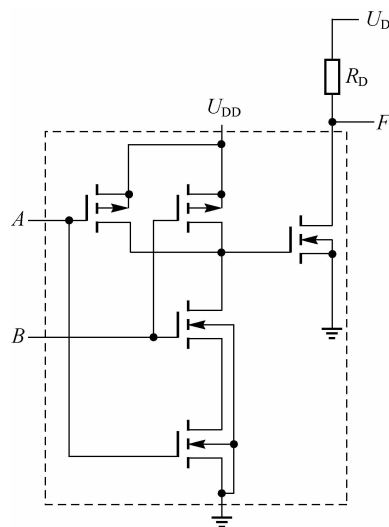


图 2.4.12 OD 输出与门

通,使 VT_2 源极为高电平。信号 A 经或非门反相,再经 CMOS 非门反相,使 $F=A$ 。使能信号 $\overline{EN}$ 为高电平时, VT_3 截止,使 VT_2 源极无电位,使 VT_2 截止, VT_1 栅极为低电平, VT_1 截止,输出 F 处于高阻状态 Z 。

再分析图 2.4.14 中的使能信号 $\overline{EN}$ 为低电平时,为信号 A 连接的 CMOS 门提供了高电平和低电平,使此 CMOS 处于普通工作状态。当 A 为高电平时,输出 F 为低电平, A 为低电平时,输出 F 为高电平,即 $F=\overline{A}$ 。当使能信号 $\overline{EN}$ 为高电平时,信号 A 连接的 CMOS 门中的 PMOS 没有提供高电平,也没有为 NMOS 源极提供低电平,使输出 F 处于高阻状态 Z 。

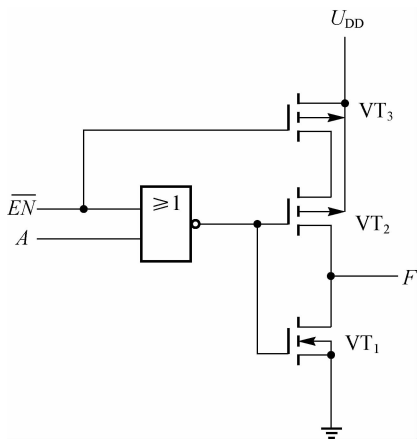


图 2.4.13 CMOS 三态门 1

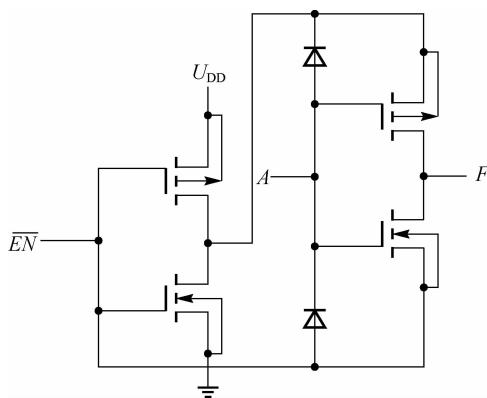


图 2.4.14 CMOS 三态门 2

6. CMOS 逻辑门特点

1) 噪声容限高

CMOS 基本逻辑门阈值电压是电源电压的 45%~50%，比 TTL 阈值电压大约高 1 V。因此 CMOS 门电路噪声容限大于同类门的噪声容限。

2) 功耗低

CMOS 功耗分为静态功耗和动态功耗。

(1) 静态功耗。当输出处于低电平或高电平不变时,输出缓冲器一个 MOS 管导通,另一个 MOS 管截止。由于截止时等效电阻很大,同时流过两个管子的电流很小,此时功耗称为静态功耗。

(2) 动态功耗。在 CMOS 管输出高低电平转化时,会出现瞬时 NMOS 和 PMOS 同时导通,产生了动态功耗。动态功耗随工作频率的增加而增加。

CMOS 静态功耗小,但动态功耗类似于 TTL 电路。

3) 抗辐射能力强

CMOS 管是单极型器件,一种载流子工作,而且是多数载流子受电压控制导电器件。射线辐射对多数载流子浓度影响不大,因此 CMOS 电路特别适用于射线辐射强的航天、卫星和核试验的环境下工作,它比双极型 TTL 两种载流子工作的电流控制器件的抗辐射能力强。

4) 集成度高、温度稳定性好

CMOS 集成电路由于功耗低,内部发热量小,所以集成度可大大提高。又由于其电路本

身的互补对称结构,当环境温度变化时其参数有互补作用,因而其温度稳定性比 TTL 好。

5) 电源利用率高

输出高电平 U_{OH} 近似等于电源电压,低电平 U_{OL} 近似为零,所以逻辑摆幅约等于电源电压,使电源电压得到了充分利用。比 TTL 电源利用率高。

6) 电路的扇出系数大

由于 CMOS 电路的互补特性,其高、低电平输出时, MOS 管导通内阻比单沟道 MOS 管小得多,输出驱动电流比较大。在频率不太高的情况下, CMOS 电路的扇出能力几乎不受限制。但是 CMOS 管存在着输入电容(每管 5 pF 左右),电路的负载主要是容性负载,扇出系数增加会使工作速度下降,因此 CMOS 电路的扇出系数主要决定于工作速度,一般可以大于 50,比 TTL 扇出系数要大。

7) 电源取值范围宽

CMOS 电路电源一般可取 10 V,由实验可知,电源在很大范围变化时电路仍能保持正确的逻辑关系。不同系列产品可提供不同的电源电压,有的系列的电压范围是 1.8~18 V,有的是 7~15 V,比 TTL 电源电压取值范围宽。

8) 易受静态干扰

由于 CMOS 电路输入阻抗高,容易受静电感应出现击穿。虽然在电路内部设置保护电路,长时间超过允许值范围,也会被击穿。所以在使用和存放时还应注意静电屏蔽,焊接时电烙铁应接地良好。

思考题

题 2.4.1 CMOS 非门的保护电路作用是当输入电压为高低电平范围内时,电路正常工作。当瞬时受到电磁高电压冲击时,保护二极管起到_____作用。

- (A) 分流 (B) 分压 (C) 积分 (D) 微分

题 2.4.2 TTL 电路时间延迟主要是由载流子的聚集和消散引起的, CMOS 反相器产生传输延迟的主要原因是由于集成电路_____。

- (A) 反相器 (B) 内部电阻和容性负载
(C) 电阻 (D) 电源

题 2.4.3 CMOS 逻辑门电路电压摆率可为_____。

- (A) 0.8 V (B) 3.3 V (C) 5 V (D) 30 V

题 2.4.4 CMOS 逻辑电路的基本单元是_____和_____。

题 2.4.5 CMOS 与非门输入/输出端加 CMOS 非门的原因是什么? 其逻辑输出有什么变化?

题 2.4.6 CMOS 非门是由一个 NMOS 和一个 PMOS 组成,其栅极相连作为输入,漏极相连作为输出, NMOS 源极需接_____电平, PMOS 源极接_____电平。

- (A) 高 低 (B) 高 高 (C) 低 高 (D) 低 低

题 2.4.7 能够“线与”的逻辑门电路是_____。

- (A) OD 门 (B) OC 门
(C) 三态门 (D) 注入逻辑门

题 2.4.8 OD 门除了能“线与”之外,它的作用还有_____。

- (A)驱动大负载 (B)电平移位 (C)线或 (D)存储

题 2.4.9 CMOS 逻辑门有静态功耗和动态功耗能之分,判断对错。

- (A)动态功耗是当 NMOS 和 PMOS 都导通的时候
(B)动态功耗是输出高电平的时候
(C)动态功耗是输出低电平的时候
(D)静态功耗是输出电平不变的时候

题 2.4.10 双极型 TTL 两种载流子工作的电流控制器件在抗辐射能力方面比单极型 CMOS 一种载流子工作的电压控制器件_____。因为射线辐射对_____浓度影响不大。

- (A)多子 (B)强 (C)弱 (D)少子

2.5 改进型 CMOS 门电路

2.4 节介绍的 CMOS 电路是 CMOS 器件中的 4000 系列,电路虽然有功耗低、抗干扰能力强等独特的优点和完整的系列产品,但在开关速度和驱动能力方面有其弱点。为提高上述性能,在 CMOS 4000 系列基础上进行了改进,改进的电路有高速 CMOS 和 Bi-CMOS 电路,Bi-CMOS 集合了 CMOS 和 TTL 的优点,是 CMOS 集成电路最重要的突破,发展也相当迅速,受到用户的普遍欢迎。

2.5.1 高速 CMOS 门电路

CMOS 4000 系列用金属栅工艺制造,在 MOS 管各极之间存在着较大的寄生电容,图 2.5.1 中标出了影响电路开关速度的主要寄生电容的位置,这些寄生电容的存在,使 CMOS 4000 系列在高频应用时受到了限制,降低了 MOS 管的开关速度。

寄生电容越小,速度就可以提高。因此降低极与极间电容值,就是制作高速 CMOS 器件的主要方法。高速 CMOS 系列与 4000 系列器件的电路原理图完全相同,仍然采用互补对称型的电路结构。

为了减小寄生电容,高速 CMOS 电路从工艺上作了改进,首先采用硅栅工艺制造,另外尽可能地减小沟道的长度,缩小 MOS 管的尺寸,综合工艺上的各种改进措施,使高速 CMOS 的寄生电容减小,高速 CMOS 的开关速度达到 4000 系列的 8~10 倍。

高速 CMOS 系列与 4000 系列寄生电容的比较如表 2.5.1 所示。

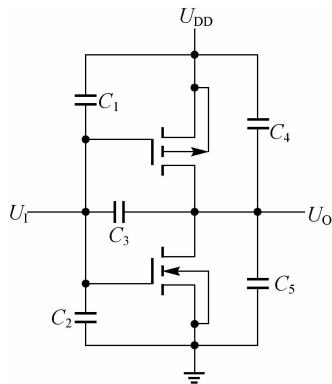


图 2.5.1 CMOS 反相器的寄生电容

表 2.5.1 高速 CMOS 系列与 4000 系列寄生电容

寄生电容	4000 系列	高速 CMOS 系列
C_1	0.25 pF	0.1 pF
C_2	0.12 pF	0.05 pF
C_3	0.37 pF	0.15 pF
C_4	0.66 pF	0.31 pF
C_5	0.54 pF	0.22 pF

高速 CMOS 系列芯片采用标准 TTL 的型号,命名为 54HC/74HC 系列,只要芯片后面的序号与 54LS/74LS 相同,则表明两种器件的逻辑功能相同,外形尺寸和引脚排列也相同。

高速 CMOS 系列具有以下特点。

- (1)与 4000 系列一样,具有从简单门电路到大规模集成电路的全系列产品。
- (2)器件功能、器件引脚与 TTL 74 系列相同。
- (3)高速 CMOS 门的典型传输延迟为 8~11.5 ns,与 TTL 基本相同,比 CMOS 4000 系列提高一个数量级。
- (4)相邻输入端之间电流耦合小,有助于在交通或重工业噪声环境中使用。

2.5.2 双极型 CMOS 门电路(Bipolar CMOS)

双极型 CMOS 电路实现逻辑功能采用 CMOS 器件,驱动输出级采用 TTL 射极跟随输出电路。由于同时采用 TTL 和 CMOS 电路,因此 Bi-CMOS 电路不仅具有 CMOS 电路的低功耗,而且具有 TTL 输出电阻低、负载能力强、传输延迟时间短等特点。

下面介绍 Bi-CMOS 非门和与非门电路。

1. Bi-CMOS 非门

Bi-CMOS 非门电路组成如图 2.5.2 所示。其中 VT_1 是驱动管, VT_3 和 VT_4 是双极型 TTL 输出管, VT_2 是输出管 VT_4 基极的下拉负载管,形成有源负载电路。 C_L 为负载电容。

当输入高电平时,虚线框内非门输出低电平, VT_2 和 VT_3 截止。 VT_1 和 VT_4 导通,输出低电平。当输入低电平时,虚线框内非门输出高电平, VT_2 导通, VT_1 和 VT_4 截止, VT_3 射极跟随输出高电平。输出与输入之间实现非逻辑。

由于 VT_3 和 VT_4 的导通内阻很小,对负载电容的充放电速度很快,故电路的传输延迟时间很短,可以达到 1 ns 以下。

2. Bi-CMOS 与非门

Bi-CMOS 与非门电路组成如图 2.5.3 所示。和与非门电路相比,输出方式是一样的,只是 MOS 逻辑功能器件的连接不同。

当输入端 A、B 中有低电平时,如 B 为低电平,虚线框内与非门输出高电平。则 VT_1 导通, VT_4 截止。 VT_2 截止,因此使 VT_5 导通,射极跟随输出为高电平。当输入端 A、B 全为高电平时,虚线框内与非门输出低电平,则有 VT_5 、 VT_1 截止, VT_2 和 VT_3 导通,使得 VT_4

基极获得足够的电压使 VT_4 导通, 输出低电平。输出与输入之间实现与非逻辑。

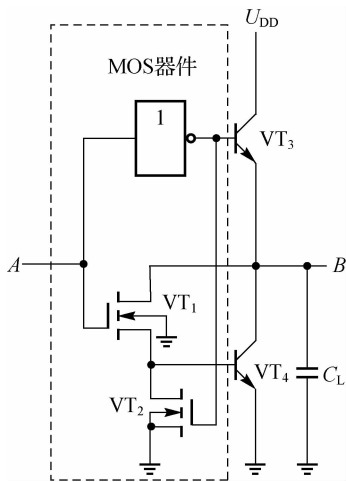


图 2.5.2 Bi-CMOS 非门电路

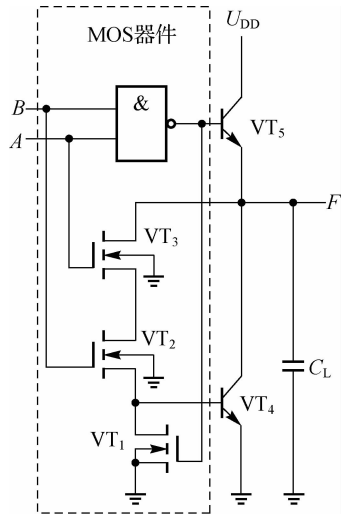


图 2.5.3 Bi-CMOS 与非门电路

2.5.3 CMOS 门电路主要参数

初期的 CMOS 集成电路为 4000 系列, 随后又有 74C 系列, 编号和引脚与 TTL 电路兼容, 但是输入电压与输出电压不兼容。CMOS 高速系列 74HC 系列产品不仅功耗低, 而且每个门的传输时间小(6~10 ns), 有的甚至更小, 与 74LS 系列相当。相同电源电压的情况下, 其噪声容限约为 TTL 电路的 2 倍。此外, 还有 74HCT 系列和 74BCT(Bi-CMOS) 系列等, 都已经做到与 TTL 电路兼容。

CMOS 集成电路各系列的主要技术参数如表 2.5.2 所示, 可以与 TTL 集成电路参数进行比较。

表 2.5.2 CMOS 集成电路各系列的主要技术参数

参 数	基本 CMOS (4000/4000B 系列)	高速 CMOS (74HC 系列)	与 TTL 兼容的 高速 CMOS (74HCT 系列)	与 TTL 兼容的 高速 Bi-CMOS (74BCT 系列)
t_{pd}/ns ($C_L=15\text{ pF}$)	75	10	13	2.9
P_D/mW	0.002	1.55	1.000 2	0.000 3~7.5

思考题

题 2.5.1 减小极与极间的_____可以提高 4000 系列的速度,就是采用硅栅工艺制造,尽可能地减小沟道的_____。

- (A)寄生电容 (B)长度 (C)宽度 (D)电阻

题 2.5.2 高速 CMOS 在电路结构和工作原理方面与 CMOS4000 系列_____。

- (A)不同 (B)相同 (C)差别很大 (D)略有不同

题 2.5.3 双极型 CMOS 电路实现逻辑功能采用_____电路,驱动输出级采用_____输出电路。

- (A)CMOS (B)NMOS (C)TTL (D)混合

题 2.5.4 双极型 CMOS 电路的传输延迟时间短是由于负载电容的充放电速度_____,原因是 TTL 输出管导通内阻很_____。

- (A)快 (B)大 (C)慢 (D)小

2.6 工程应用的技术问题

工程中设计数字电路与系统时,可根据系统设计的需要,考虑集成电路的电气性能。从开关速度、功耗、电压匹配、电流匹配、噪声容限、输入、输出电容等方面选择某一种合适的器件,或者从几种系列中选择性能最佳的逻辑器件,再将它们混合组装在一起。下面对若干实际应用问题进行研究。

2.6.1 不同系列电路的接口问题

数字系统中,往往对系统的速度和功耗有较高的要求,同一系统中会混合使用不同系列的逻辑器件,常常碰到不同系列逻辑器件的接口问题,如同时使用 CMOS 和 TTL 电路。在接口电路的设计过程中,由于各种集成电路的电压和电流等参数的不同,电压、电流参数成为电路之间主要的接口参数,不同系列电路的电压、电流参数满足下面四个不等式即可。

电压匹配:驱动电路(前级电路)的输出电压应在负载电路(后级电路)所要求的输入电压范围内,前者输出为高电平时的电压最小值大于等于后者输入为高电平的电压最小值,前者输出为低电平时的电压最大值小于等于后者输入为低电平的电压最大值,即

$$U_{OH(\min)} \geq U_{IH(\min)}, U_{OL(\max)} \leq U_{IL(\max)}$$

电流匹配:驱动电路能对负载电路提供足够大的灌电流和拉电流,以便能驱动后级电路。驱动电路与负载电路电流之间的驱动应满足

$$I_{OH(\max)} \geq n I_{IH(\max)}, I_{OL(\max)} \geq m I_{IL(\max)}$$

n 和 m 是负载拉电流和灌电流的个数。

为了方便计算参数与设计电路,表 2.6.1 列出了 TTL 系列和 CMOS 系列的输入、输出特性参数。

表 2.6.1 TTL 和 CMOS 门电路的输入、输出特性参数

电路种类 参数名称	TTL 74 系列	TTL 74LS 系列	CMOS 4000 系列	高速 CMOS 74HC 系列	高速 CMOS 74HCT 系列
$U_{IH(min)}/V$	2	2	3.5	3.5	2
$U_{IL(max)}/V$	0.8	0.8	1.5	1	0.8
$I_{IH(max)}/\mu A$	40	20	0.1	0.1	0.1
$I_{IL(max)}/mA$	-1.6	-0.4	-0.1×10^{-3}	-0.1×10^{-3}	-0.1×10^{-3}
$U_{OH(min)}/V$	2.4	2.7	4.6	4.4	4.4
$U_{OL(max)}/V$	0.4	0.5	0.05	0.1	0.1
$I_{OH(max)}/mA$	-0.4	-0.4	-0.51	-4	-4
$I_{OL(max)}/mA$	16	8	0.51	4	4

1. TTL 系列驱动 CMOS 系列

TTL 为驱动电路,CMOS 为负载电路。TTL 电路输出电流较大,又由于 CMOS 输入电阻很高,则输入电流很小。几乎从前级吸取微量电流就可正常工作,所以在 TTL 系列驱动 CMOS 系列上,仅考虑电压匹配就可以。

根据电源电压把接口电路分为电源电压相同的接口和电源电压不同的接口。

1) 电源电压相同

前者采用 74LS 系列,后者采用 74HC 系列,电源电压同为 5 V,由表 2.6.1 可知,两种电路直接连接时,TTL 电路输出高电平只有 2.7 V,而 CMOS 电路的输入高电平要求高于 3.5 V,不满足电压匹配条件。

解决方案之一:在 TTL 电路输出端与电源之间接一上拉电阻 R_P ,使 TTL 电路的输出高电平升高到电源电压,以实现与 74HC 电路的电压匹配,如图 2.6.1(a)所示。

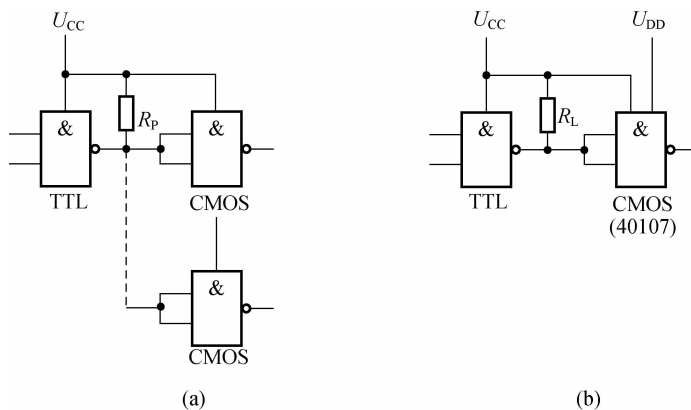


图 2.6.1 TTL 驱动 CMOS 门电路

(a) 电源电压相同 (b) 电源电压不同

若 TTL 采用 74LS 系列,CMOS 采用 74HCT 系列,由于两者电压参数匹配,不需要另外加任何接口电路,因此,在数字系统的设计中,常选用 74HCT 系列器件,减少上拉电阻的

设计。

解决方案之二:先用 TTL 电路驱动 74HCT 电路,再由 74HCT 电路驱动 74HC 电路,这是一种 TTL 门电路驱动 74HC 门电路的最佳选择。

2) 电源电压不同

如果 CMOS 电源 U_{DD} 高于 TTL 电路电源 U_{CC} ,也有两种解决方案。

解决方案之一:选用具有电平偏移功能的 CMOS 逻辑门,如 40107,该电路有两个电源输入端 U_{CC} 和 U_{DD} ,当 $U_{CC}=5\text{ V}$ 、 $U_{DD}=10\text{ V}$ 时,其输入接收 TTL 电平 $1.5\text{ V}/3.5\text{ V}$,输出 CMOS 电平 $9\text{ V}/1\text{ V}$,满足作为负载的 CMOS 电路对输入电压的要求。TTL 门与 CMOS 门 CC74HC109 的电路连接方式如图 2.6.1(b)所示。

解决方案之二:采用 TTL 的 OC 门作为 CMOS 的驱动电路,因 OC 门 T_5 管的耐压远超过 U_{DD} ,所以将 OC 门 T_5 管的外接电阻 R_L 直接与 CMOS 电源 U_{DD} 连接,电路也能可靠工作。

2. CMOS 系列驱动 TTL 系列

采用 4000 系列 CMOS 电路驱动 74 系列 TTL 电路,由表 2.6.1 提供的参数可知,满足电压匹配要求。主要看 CMOS4000 系列是否能提供 TTL74 系列负载所需要的灌电流和拉电流。查表 2.6.1,从观察到的参数可知,CMOS 电路的电流驱动能力不满足 TTL 电路的要求,主要是在 CMOS 电路输出低电平时能承受的灌电流较小,仅为 0.51 mA ,而 74 系列 TTL 门的输入短路电流较大,需要 1.6 mA 。在这种情况下,用 CMOS 门驱动 TTL 门,将不能保证 CMOS 规定的输出低电平。可以选择多种方案来解决这个问题。

(1)采用驱动门并联的方式,提高驱动门的带负载能力。

(2)采用增加一级 CMOS 驱动器的方法,如选择同相驱动器 CC4010,电路如图 2.6.2(a)所示。

(3)采用漏极开路的 CMOS 驱动器 CC40107,可驱动 10 个 74 系列负载门,如图 2.6.2(b)所示。

(4)采用分立元件驱动电路作为 4000 系列 CMOS 电路和 74 系列电路的跨接电路,使电流放大,从而驱动 TTL 负载门,其电路如图 2.6.2(c)所示。

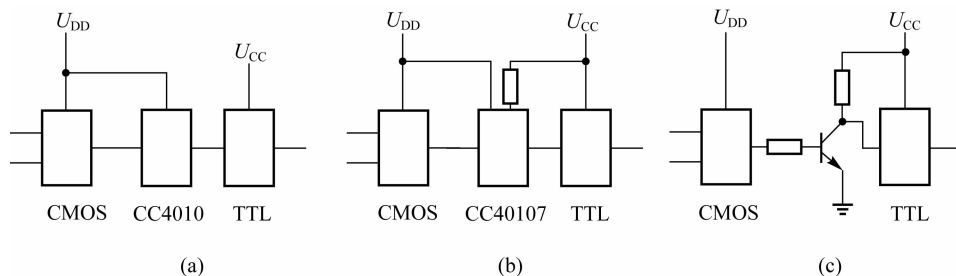


图 2.6.2 CMOS 门驱动 TTL 门

(a)采用同相驱动器 (b)采用漏极开路驱动器 (c)采用分立元件驱动电路

由表 2.6.1 提供的参数可知,用 4000 系列 CMOS 电路驱动一个 74LS 系列 TTL 电路时,直接连接即可。但是如果 74LS 系列负载电路数量增加,仍然需要采用图 2.6.2 提供的方法,提高 4000 系列的驱动能力。

由表 2.6.1 还可以了解到,用 74HC 系列或 74HCT 系列 CMOS 电路驱动 TTL 电路,

无论负载门是 74 系列还是 74LS 系列,都可以直接相连,根据表 2.6.1 提供的参数可以计算出驱动电路的个数。

2.6.2 具有负载的接口电路

在工程应用中,常会遇到数字电路控制电动机的方向、继电器的接通与断开、显示器件的亮与灭等情况,上述情况都需要设计合理的接口电路。无论用数字电路驱动哪种负载,只要查询数字电路的数据手册,了解门电路的驱动能力,了解负载的额定电压和额定电流,根据上述参数设计接口电路即可。

1. 驱动大电流负载

电气系统有的器件所需的工作电压和工作电流比较大,即使微型继电器的驱动电流也会在 10 mA 以上。这些电气元件要正常工作,需要提供较大驱动电流和电压,就要放大电流或电压以提高带负载能力。

负载电流较小:如微型继电器,可以将两个反相器并联作为驱动电路。

负载电流较大:需要在集成电路或系统的输出端与负载之间接入一个功率驱动器件,可直接用于驱动机电系统。如图 2.6.2(c)中的输出 TTL 电路换成电气元件,图中的单级三极管为驱动电路,可以换成达林顿晶体管。并将集电极外加电阻和电源(采用集电极开路输出结构),使输出高电平近似等于外加电压,通过调节外加电压达到负载对高电平电压的要求。

2. 驱动 LED 负载

设 LED 的工作电流为 I_D 、LED 的正向压降为 U_D 。限流电阻 R 可以按以下公式选择。

对于图 2.6.3(a),门电路输出高电平时,限流电阻的选择应满足如下关系式

$$R = \frac{U_{OH} - U_D}{I_D}$$

对于图 2.6.3(b),门电路输出低电平时,限流电阻的选择应满足如下关系式

$$R = \frac{U_{CC} - U_D - U_{OL}}{I_D}$$

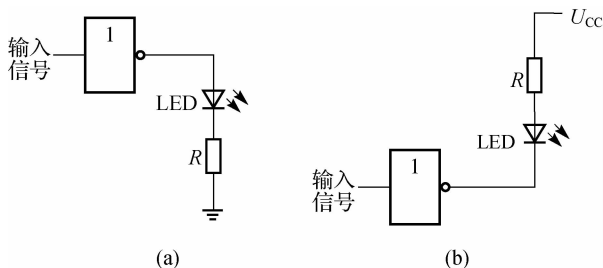


图 2.6.3 门电路驱动发光二极管电路

2.6.3 数字电路的抗干扰方法

数字系统经常会遇到电磁耦合、静电电压、尖峰电流等干扰,使电路系统的正常逻辑发生混乱,运行不正常。以下介绍几种消除干扰的方法。

1. 增加输入保护电路

MOS管栅极金属板和导电沟道之间有一层很薄的绝缘二氧化硅,很容易被静电高压击穿,损坏 MOS 管。为了防止过压或静电干扰,增加正、反两方向的保护二极管电路,如图 2.4.2 所示,使 MOS 管钳位在允许的电压范围之内,同时防止冲击电流。

2. 增加功率输出缓冲电路

如前所述,多扇入的 CMOS 逻辑门电路会抬高输出低电平,容易受到各种噪声的干扰,使逻辑电平出现错误。为了降低输出低电平,增加功率输出缓冲电路,如图 2.4.9 所示。功率缓冲电路的扇出能力远比普通门高,电路结构与普通缓冲电路类似,仅在输出级加大了功率驱动。

3. 增加去耦合滤波电容

数字电路是由多片逻辑门电路构成,由具有一定的内阻统一直流电源供电。当数字电路在高、低状态之间交替变换时,产生较大的脉冲电流或尖峰电流,相互影响,甚至使逻辑功能发生错误。将 $10\sim 100\ \mu\text{F}$ 去耦合滤波电容接在直流电源与地之间,滤除尖峰电流。并在每一个集成芯片的电源与地之间接一个 $0.01\sim 0.1\ \mu\text{F}$ 的电容器滤除开关噪声。

4. 电源或地与不用输入端串接电阻

干扰信号进入输入端,将影响电路的逻辑,甚至损坏集成电路。采取与门或者与非门的多余输入端与正电源串接 $1\sim 3\ \text{k}\Omega$ 电阻,对 CMOS 电路可以直接接入电源。或门或者或非门的多余输入端接地。

一般多余输入端不和使用信号并接,因为高速数字电路输入端并接会增加输入端等效容性负载,使信号的传输速度下降。特别是 CMOS 电路的多余输入端绝对不能悬空。由于它的输入电阻很大,容易受到静电或电磁电荷的影响而破坏电路的正常工作状态。

5. 接地技术

当系统中同时有模拟和数字两种器件时,需将二者的地连在一起,然后再选用一个合适共同点接地,或者用光电转换电路隔离,具有模拟地和数字地,以消除二者之间的影响。也可以制作模拟和数字两块电路板,提供本身直流电源,然后将二者之间的地恰当地连接在一起。

6. 安装技术

在印制电路板的设计或安装中,先将信号集中在一起,再将信号用尽可能短的连线连在电源、地上,以减少接线电容产生寄生反馈而引起的寄生振荡。

2.6.4 数字电路使用注意事项

数字电路的规范使用与保护,可增加数字系统的可靠性,延长数字系统的寿命。

(1)具有推拉输出结构的 TTL 门电路的输出端不能直接并联使用,输出端也不允许直接接电源 U_{CC} 或直接接地,电源与输出端可接一个上拉负载电阻 R 。

(2)多种电源供电时,先接入低电压,再加上较高的电压,然后再输入信号。关断电源之前,应先去掉输入信号。若信号源与电路板使用两组电源供电,开机时应先接通电路板电源,再接通信号源,关机时先断开信号源后断开电路电源。

(3)门与门之间的导线应尽可能短,过长会造成信号的畸变。当多路并行传输信号时,信号线间要有一定的距离或中间加地线进行屏蔽。否则,相邻平行线可能由于电磁耦合而

互相干扰。

(4)CMOS 器件在使用、储藏、运输、高温老化过程中,需屏蔽于接触良好的金属屏蔽盒内或用金属铝箔纸包装,防止静电感应击穿栅极,烧毁场效应晶体管。

(5)所有测试仪器,外壳必须有良好的接地。进行电路故障处理时,应将 CMOS 电路的输入端与前级输出端脱开。也可用 $50\sim 100\text{ k}\Omega$ 的电阻将输入端与地或电源相连。

思考题

题 2.6.1 增加功率输出缓冲电路是为了_____带负载能力,降低输出_____。

(A)增加 (B)降低 (C)低电平电压 (D)高电平电压

题 2.6.2 增加去耦合滤波电容为了消除_____,每一个芯片的电源与地之间接一个 $0.01\sim 0.1\text{ }\mu\text{F}$ 的电容器滤除_____。

(A)尖峰电流 (B)开关噪声 (C)直流 (D)正弦波

题 2.6.3 TTL 电路驱动 CMOS 电路,仅考虑_____。

(A)电压匹配 (B)电流匹配 (C)功率匹配 (D)频率匹配

题 2.6.4 TTL 电路驱动 CMOS 电路,TTL 输出_____满足 CMOS 电路要求。CMOS 电路驱动 TTL 电路,CMOS 输出_____满足 TTL 电路的要求。

(A)电压 (B)电流
(C)输出高电平的最小值 (D)高电平电压

2.7 VHDL 的并行和结构行为

在 VHDL 中,描述电路逻辑的程序称为行为描述,行为描述有并行行为描述、进程行为描述和顺序行为描述。它们可以相互联系成为混合描述体,进程行为描述语句存在于结构体中,本身之间是并行行为,进程行为内部的语句是顺序行为。本节介绍结构体中的并行描述和结构描述。

2.7.1 并行描述

结构体的内部语句的执行或发生不按规定语句的次序,执行的次序仅由对语句中的敏感信号发生的事件决定,产生其并发性。一般语言中,每个赋值语句按源文件规定的顺序执行。结构体中最常见并行信号赋值语句的一般格式如下。

<对象><=<表达式>

表达式中至少有一个敏感信号,每当敏感信号逻辑值发生变化时,这个信号赋值语句就执行,如果执行的结果是一个不同于当前信号值的新值,就为该对象的信号赋值语句做一个事件处理。

并行行为语句的两个以上的并行赋值语句在字面上的顺序并不表明它们的执行顺序,如下面的两个结构体在功能上是等价的。

例 2.7.1 分析下面程序的动态行为。

```
ENTITY exe IS
    PORT(a,b :IN BIT; x,y :OUT BIT);
END exe;
```

```

ARCHITECTURE exe_arc1 OF exe IS
BEGIN
    x<=a AND b;
    y<=NOT x OR b;
END exe_arc1;
ARCHITECTURE exe_arc2 OF exe IS
BEGIN
    y<=NOT x OR b;
    x<=a AND b;
END exe_arc2;

```

}1
}2

分析:语言表面结构体“1”中的 x 得到信号 a 和信号 b 相与的值,然后再执行 y 得到信号 x 取非再和信号 b 相或的值。实际上是同时执行的,如当 b 逻辑值改变后,同时执行情况“1”中的两条赋值语句。如同为两个电路同时加上信号一样。因此结构体“1”和结构体“2”逻辑功能是等价的。

下面以两路数据选择器为例,讨论一个较为复杂的并行信号赋值语句,以便更详细地说明并行概念,两输入数据选择器的 VHDL 程序如下。

例 2.7.2 分析下面两输入数据选择器程序的动态行为。

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY mux2 IS
    PORT(I0,I1,A :IN std_logic;
          Q :OUT std_logic);
END mux2;
ARCHITECTURE mux2_arc OF mux2 IS
    SIGNAL sel :INTEGER;
BEGIN
    Q<=I0 AFTER 10 ns WHEN sel=0 ELSE
        I1 AFTER 10 ns;
    sel<=0 WHEN A='0' ELSE
        1;
END mux2_arc;

```

分析:此程序从“Q<=”到“10 ns;”为第一句,其中有一信号为 sel 需要判别,第二句是从“sel<=”到“1;”计算 sel 的值,这句用的是选择信号赋值语句,根据满足条件给出信号 sel 的值,即根据信号 A 值,把 0 到 1 的值分配给 sel。

从程序的先后顺序看,两输入数据选择器的程序好像不能工作,因为要用到的选择信号 sel 的值是以后计算出来的,事实上得出这种印象是来自计算 sel 在使用 sel 之后。但因为 VHDL 语言的并行行为的特点,此程序是能够正常工作的。第一句和第二句是并行的。而第二句对信号 A 是敏感的,每当 A 的值变化时,将执行第二句来更新信号 sel,第一句的程序对信号 sel 敏感,每当信号 sel 的值发生变化时,就执行第一句程序。

2.7.2 结构描述

对一个电路结构进行描述,称为结构描述。结构描述主要是具体叙述它由哪些子元件组成以及各个子元件之间的互联关系。VHDL 中,实体主要描述三个方面,即元件、端口与信号。

元件:门、芯片或者电路板。

端口:元件与外界的连接点,数据通过端口进入或流出元件。

信号:信号在端口处将元件连接起来,作为硬件连线的一种抽象描述,既能保持变化的数据,又可看作两个元件之间数据传输的通路。

结构描述的基本单元则是“调用元件语句”。下面举例说明“调用元件语句”的用法。

例 2.7.3 用 VHDL 结构描述设计同或电路,同或电路由与、或、非基本门组成。

解:首先用 VHDL 设计与、或、非基本门电路。

与门:

```
ENTITY aa IS
    PORT(a1,a2 :IN BIT;
          a3 :OUT BIT);
END aa;
ARCHITECTURE aa_arc OF aa IS
BEGIN
    a3<=a1 AND a2 AFTER 10 ns;
END aa_arc;
```

或门:

```
ENTITY bb IS
    PORT(b1,b2 :IN BIT;
          b3 :OUT BIT);
END bb;
ARCHITECTURE bb_arc OF bb IS
BEGIN
    b3<=b1 OR b2;
END bb_arc;
```

非门:

```
ENTITY cc IS
    PORT(c1 :IN BIT;
          c2 :OUT BIT);
END cc;
ARCHITECTURE cc_arc OF cc IS
BEGIN
    c2<=NOT c1;
END cc_arc;
```

在 VHDL 程序设计结构描述时先讨论以下问题。

(1)从外部看一个实体说明中的元件端口是它和外界的连接点,这些连接点实际是集成芯片的外部引脚。连接点在关键字 PORT 后面加以说明,每个端口说明包括端口名、数据流向以及数据类型。定义的目标是默认、规定的信号 SIGNAL,如 a1,a2,b1,b2,c1,c2。

(2)VHDL 为设计者提供的端口流向有 IN、OUT,分别表示数据的流入、流出。结构描述中应记住端口的信号名称、信号流向的顺序。

(3)结构体中的元件说明语句一般形式如下。

```
COMPONENT<元件名>
PORT(<端口 1>...<端口 n> :<端口类型>;
      <端口 1>...<端口 n> :<端口类型>);
END COMPONENT;
```

元件说明语句是以“COMPONENT”开头和以“END COMPONENT;”结尾,COMPONENT 后是调用的元件名,PORT 后面是被调用元件的端口信息。元件调用语句的作用是调用先设计好的模块存档子元件。元件说明语句 COMPONENT 是在结构体的说明区部分,说明了元件名、输入引脚、输出引脚。虽然不调用具体子元件的功能,实际上已将所用子元件的模块调出,且按功能连接好。

(4)元件调用语句的一般形式如下。

```
<标号标识符> :<元件符号>PORT MAP(<关联表>);
```

标号标识符是对被调用元件重起的元件名。而元件符号就是被调用的子元件模块名,即和元件说明语句中的 COMPONENT 语句说明的元件名相对应。

元件调用语句首先要使调用语句的元件符号和元件说明语句的名一致,然后产生映射 MAP。语句 MAP 后面的括号中是引脚的关联关系,即实体的端口或者是结构体说明区中定义的信号,映射到调用元件语句中的关联表中。

VHDL 结构描述的关键是定义中间信号(准备连接元件的连接线),定义 4 个中间信号 temp_c1、temp_c2、temp_c3 和 temp_c4,将两个与门、两个非门和一个或门的端口连接起来形成 VHDL 对同或门的结构描述。图 2.7.1(b)所示虚线框有各元件之间的连线命名。temp_c1、temp_c2 将两个非门的输出连到其中一个与门的输入端,与门输出端用中间信号 temp_c3 表示。然后用中间信号 temp_c3 和 temp_c4 作为或门的输入端。用 5 个元件调用语句定义这 5 个连接元件的连接关系。每一个调用元件语句都为调进来的元件(如与门、非门和或门)取了个名字,如 U0、U1 等。

下面是同或门的 VHDL 程序描述。

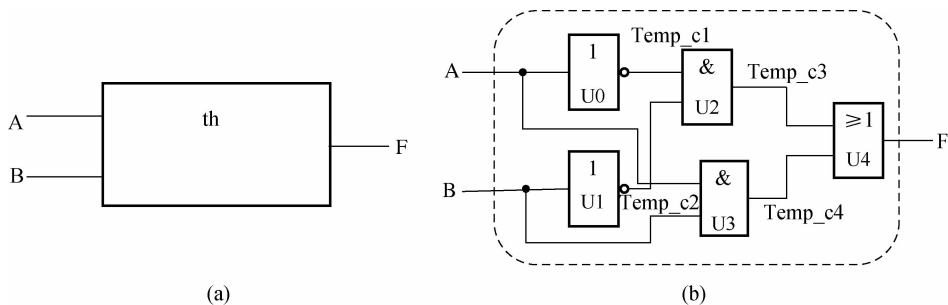


图 2.7.1 同或门符号和同或门结构电路

(a)同或门 (b)同或门结构电路

```

ENTITY th IS
    PORT(A,B :IN BIT;
          F :OUT BIT);
END th;
ARCHITECTURE th_arc OF th IS
    SIGNAL temp_c1,temp_c2,temp_c3,temp_c4 :BIT;
    COMPONENT aa
        PORT(a1,a2 :IN BIT;
              a3 :OUT BIT);
    END COMPONENT;
    COMPONENT bb
        PORT(b1,b2 :IN BIT;
              b3 :OUT BIT);
    END COMPONENT;
    COMPONENT bb
        PORT(c1 :IN BIT;
              c3 :OUT BIT);
    END COMPONENT;
BEGIN
    U0 :cc PORT MAP(A,temp_c1);
    U1 :cc PORT MAP(B,temp_c2);
    U2 :aa PORT MAP(temp_c1,temp_c2,temp_c3);
    U3 :aa PORT MAP(A,B,temp_c4);
    U4 :bb PORT MAP(temp_c3,temp_c4,F);
END fullsub_arc;

```

同或门结构体的语句部分使用了 5 个调用语句 U0、U1、U2、U3 和 U4,其中 U0 和 U1 调用了非门,PORT MAP 是将非门的端口映射到同或门的局部元件的端口,A 和 B 对应非门的输入,temp_c1 和 temp_c2 对应非门的输出,在同或门中重新为局部调用元件引脚命名。U2、U3 和 U4 调用方法和 U0 和 U1 相同,利用中间信号将 U0、U1、U2、U3 和 U4 正确连接组成同或门 th,如图 2.7.1(b)所示。从以上可以看出 VHDL 的结构描述类似于绘制逻辑电路原理图,容易理解和掌握。

思考题

题 2.7.1 结构描述说明元件语句主要作用是_____,调用元件语句的主要作用是_____。

题 2.7.2 判断对错。

- (1)并行行为的语句执行的顺序按设计者安排的执行。 ()
- (2)并行行为的语句执行的顺序不按设计者安排的执行。 ()
- (3)结构描述的语句是并行性的。 ()
- (4)结构描述语句不是并行性的。 ()

本章小结

本章从集成逻辑门 CMOS 内部电路组成结构、工作原理及其外部电气特性出发,介绍了双极型 TTL 门电路、单极型门电路。

重点放在了 TTL 和 CMOS 集成电路的外部特性上,外部特性的主要内容是电气特性及其主要参数。

TTL 电路输入级采用多发射极晶体管,输出级采用推拉式结构,所以工作速度快,驱动能力强,是目前使用最广泛的集成逻辑门之一。CMOS 集成电路具有功耗低、扇出系数大、电源电压范围宽、抗干扰能力强、集成度高等一系列特点,使之在整个数字集成电路中占据主导地位的趋势日益明显。

实际工程应用中,经常考虑开关速度、功耗、电压匹配、电流匹配、噪声容限、输入、输出负载等问题,还有门电路与负载之间的接口电路的连接问题。正确分析和解决这些问题,是数字电路设计工作者应当掌握的基本能力。

集成逻辑门电路的种类很多,可以从制造工艺、输出方式、逻辑功能和集成度等多个方面对其进行分类。

1. 制造工艺

双极型、单极型、Bi-CMOS 型。

(1)双极型分为 TTL 电路、ECL 电路、 I^2L 电路和 HTL 电路。其中 TTL 电路包含 54/74 系列、54H/74H 系列、54LS/74LS 系列、54ALS/74ALS 系列等。

(2)单极型分为 NMOS 电路、PMOS 电路和 CMOS 电路。其中 CMOS 电路包含 4000 系列、54HC/74HC 系列、54HCT/74HCT 系列等。

2. 集成度

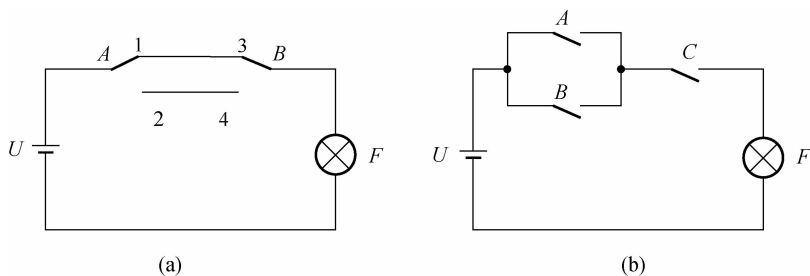
小规模集成电路 SSI、中规模集成电路 MSI、大规模集成电路 LSI 和超大规模集成电路 VLSI。

习 题

习题 2.1 CMOS 门电路与 TTL 门电路相比最大的优点是什么?

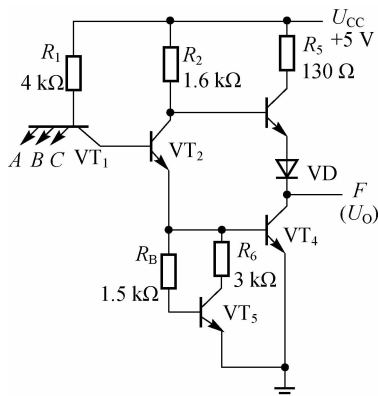
习题 2.2 从抗干扰能力来评价 TTL、ECL 和 CMOS 集成电路。

习题 2.3 灯控制电路如题图 2.1(a)、(b)所示。试写出电路的功能表、真值表和逻辑表达式。



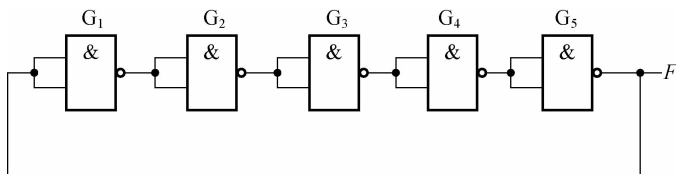
题图 2.1 习题 2.3 图

习题 2.4 试分析题图 2.2 所示 TTL 电路的工作原理和逻辑功能,并说明 V_D 、 R_B 、 R_6 和 VT_5 的作用,计算 U_{OL} 、 U_{OH} 、 U_{IL} 、 U_N 。



题图 2.2 习题 2.4 图

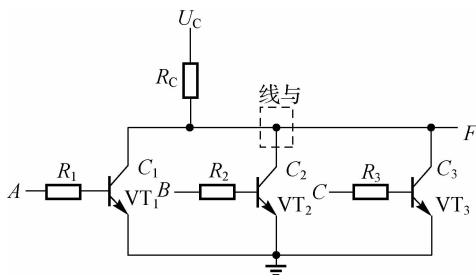
习题 2.5 题图 2.3 所示电路由 TTL 与非门组成。设 $G_1 \sim G_4$ 门的平均传输延迟时间同为 40 ns,现测得输出端 F 的振荡频率为 2.5 MHz,试求 G_5 的平均传输延迟时间 t_{pd5} 。



题图 2.3 习题 2.5 图

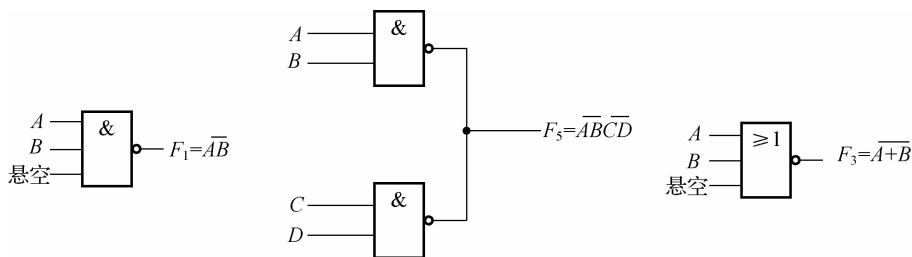
习题 2.6 有两个 TTL 与非门芯片,测得它们的关门电平分别为 $U_{OFFA} = 1.1 \text{ V}$, $U_{OFFB} = 0.9 \text{ V}$;开门电平分别为 $U_{ONA} = 1.3 \text{ V}$, $U_{ONB} = 1.7 \text{ V}$ 。它们输出的高电平和低电平相同,试判断哪一个门的抗干扰能力大。

习题 2.7 题图 2.4 所示电路中的 R_C 和 U_C 是外加的电阻和电压,分析其工作原理,试写出 F 与 C_1 、 C_2 和 C_3 之间的电平关系表、真值表和逻辑式,并画出等效的逻辑图。



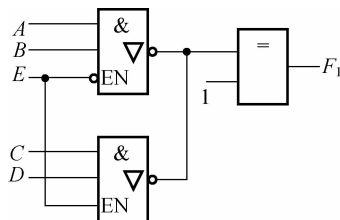
题图 2.4 习题 2.7 图

习题 2.8 若要实现题图 2.5 中各 TTL 门电路输出端所示的逻辑功能,各电路的连接是否正确? 如果不正确,试说明理由。



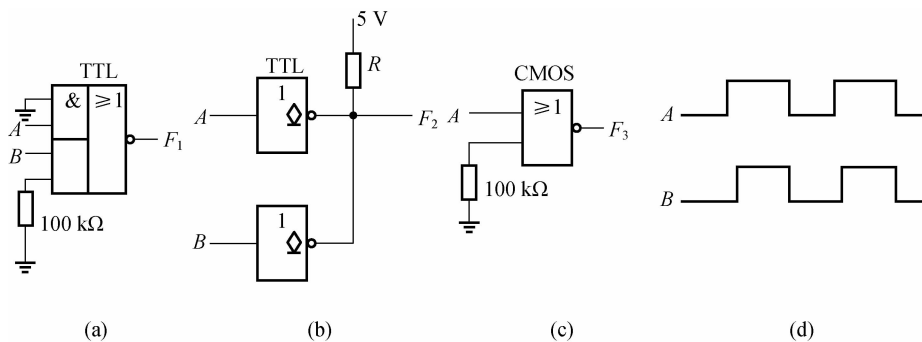
题图 2.5 习题 2.8 图

习题 2.9 分析如题图 2.6 所示的逻辑电路, 写出输入信号与输出信号之间的逻辑表达式。



题图 2.6 习题 2.9 图

习题 2.10 电路如题图 2.7(a)~(c)所示, 已知输入信号 A、B 的波形如题图 2.7(d)所示, 画出各电路的输出波形。



题图 2.7 习题 2.10 图

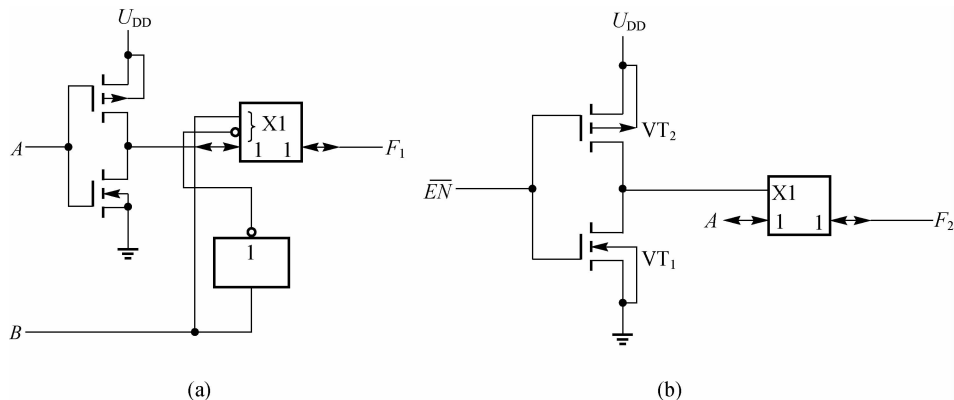
习题 2.11 用最少数量的扇入系数为 2 的 ECL 或/或非门和 I^2L 基本单元电路实现下列操作的逻辑图。

$$\text{ECL: } F_1 = A + B + C + D + E + F \quad I^2L: F_2 = AB + CD$$

习题 2.12 CMOS 电路如题图 2.8 所示。

(1) 分析电路逻辑功能, 分别写出 F_1 、 F_2 的逻辑表达式。

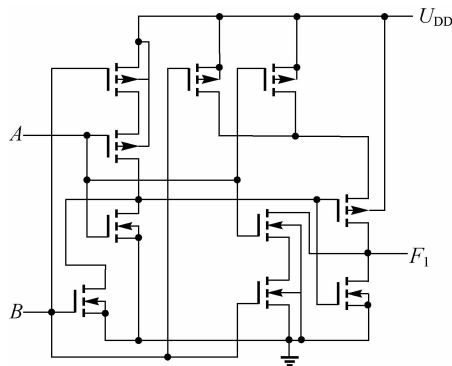
(2) 试说明两种电路的相同之处和不同之处。



题图 2.8 习题 2.12 图

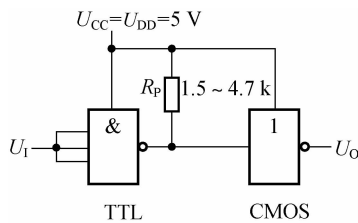
习题 2.13 试画出实现方程 $F=AB+C$ 的 CMOS 电路图,要求输出低电平最低。

习题 2.14 题图 2.9 所示为 CMOS 门电路,试写出电路图的逻辑式。



题图 2.9 习题 2.14 图

习题 2.15 TTL-CMOS 接口电路如题图 2.10 所示,试从电压匹配的角度分析 R_P 的作用。



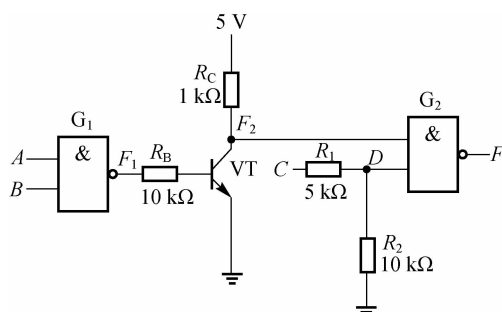
题图 2.10 习题 2.15 图

习题 2.16 电路如题图 2.11 所示,已知电路中的门均为 TTL 门,其参数 $U_{ON}=1.8\text{ V}$, $U_{OFF}=0.8\text{ V}$, $R_{ON}=2\text{ k}\Omega$, $R_{OFF}=0.8\text{ k}\Omega$, $I_{IH}\approx 0$, $I_{IL}=1.4\text{ mA}$, $U_{OH}=3.6\text{ V}$, $U_{OL}=0.3\text{ V}$, $I_{OHmax}=400\text{ }\mu\text{A}$;电路中的三极管参数 $\beta=60$, $I_{CM}=30\text{ mA}$,管子饱和时 $U_{BE}=0.7\text{ V}$, $U_{CES}=$

0.3 V, 输入信号的高低电平分别为 3 V 和 0 V。

(1) 试判断在 A、B、C 不同取值情况下, 三极管的工作状态。

(2) 试分析该电路能否实现 $Y = \overline{(A+B)} \cdot C$ 的逻辑功能。

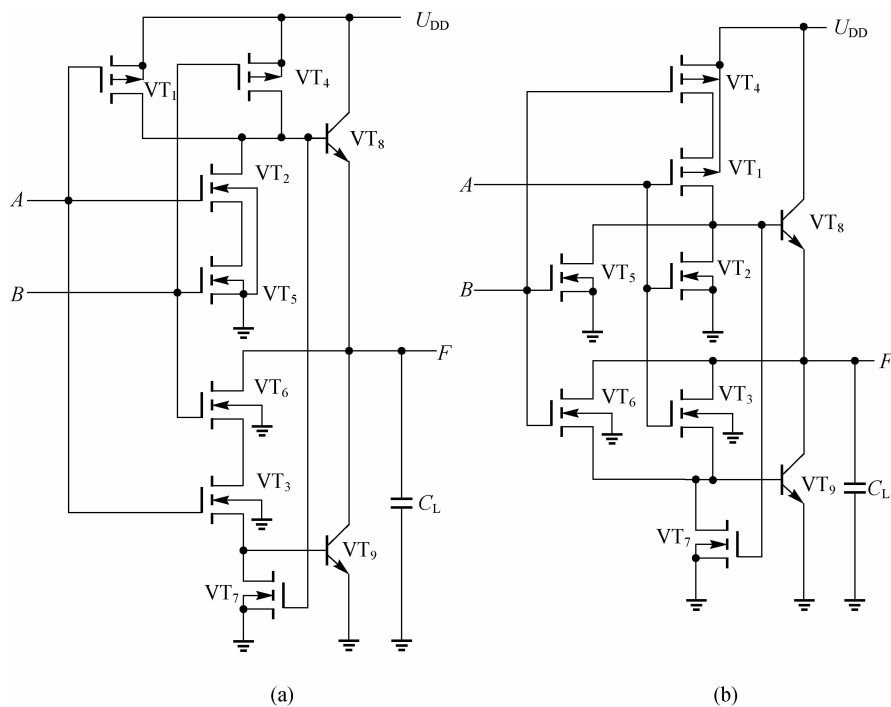


题图 2.11 习题 2.16 图

习题 2.17 Bi-CMOS 电路如题图 2.12 所示。

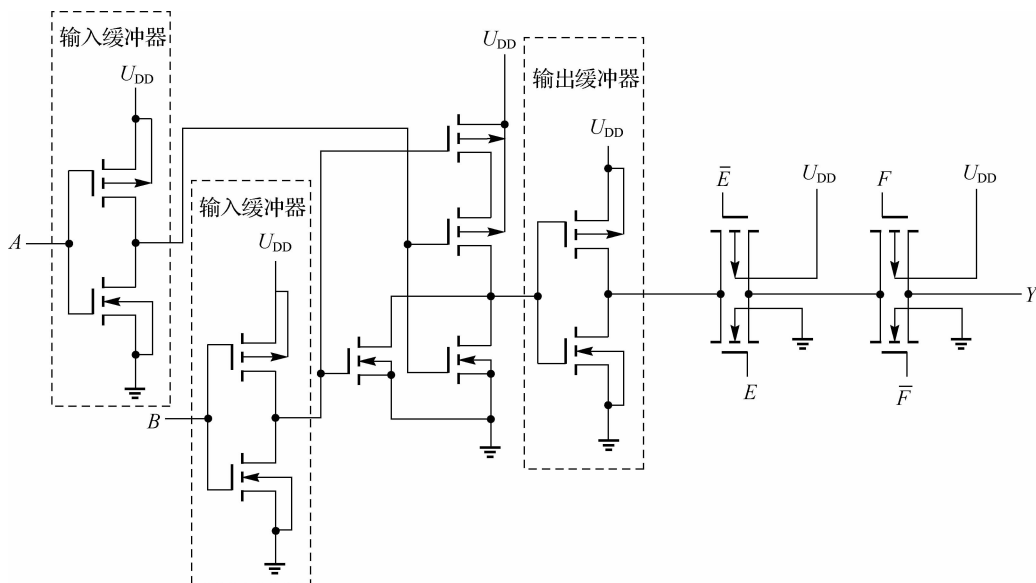
(1) 分析电路逻辑功能, 分别写出 F_1 、 F_2 的逻辑表达式。

(2) 试说明两种电路的相同之处和不同之处。



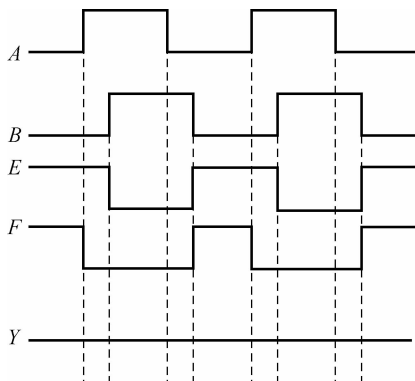
题图 2.12 习题 2.17 图

习题 2.18 分析题图 2.13 所示 CMOS 门电路原理,写出其逻辑表达式。



题图 2.13 习题 2.18 图

习题 2.19 根据习题 2.18 电路,并给出 A、B、E 和 F 波形,如题图 2.14 所示。绘出输出波形 Y。



题图 2.14 习题 2.19 图

习题 2.20 用 VHDL 的结构描述设计一个全减器。

研究论文

基于三极管输入、输出特性曲线,浅谈线性电路理论与非线性电路理论,模拟电路与数字电路联系与特点。

研究内容及要求:分析三极管输入、输出特性曲线,从曲线的各段研究非线性和线性电路理论,并叙述涉及的领域。阐述模拟和数字电路的关系,并列举涉及的几项应用产品。

第 2 章介绍了集成逻辑门电路是由单极型晶体管 CMOS 等器件或双极型晶体管 TTL 等器件组成的。而由集成逻辑门可以构成数字组合电路和时序电路,进一步组成数字系统。本章将以第 1 章逻辑代数为数学工具,用集成逻辑门构成小规模和中规模的组合逻辑电路,讨论数字电路中的组合电路分析和设计的基本方法,并介绍组合逻辑电路设计中的竞争和冒险现象以及消除的方法。

当前数字电路与系统的设计方法和手段有两种。经典的方法是利用通用集成电路器件构成预定功能的逻辑电路与系统。进入 21 世纪后,构建数字电路与系统的手段不断改进,较为复杂的数字系统可以通过电子设计自动化平台的设计工具,在可编程逻辑器件上制作集成电路芯片,称此种设计方法为现代设计方法。

设计方法的变化,使从事数字电路与系统设计工作者从繁重的工作中解脱出来,但两种方法的设计原理是一致的。针对逻辑电路的两种实现方法,本章将重点讨论若干常用中规模集成电路模块的工作原理及其应用方法,利用 VHDL 对典型集成电路模块的功能加以描述。本章最后将讨论可编程逻辑器件的基本单元、基本原理和特点,为进一步掌握分析和设计数字电路与系统奠定基础。

3.1 组合逻辑电路的分析与设计

如图 3.1.1 所示的数字电路中,根据逻辑功能数字电路可分为上下两个组成部分:上面虚线框内是组合逻辑电路(简称组合电路),下面框内是时序逻辑电路(简称时序电路)。在绘制逻辑电路时,习惯上信号从左边进,在右边出,也有的电路信号从下面进,上面出。

组合电路的信号从左边进去之后,信号向右发送,一直到输出。任何信号都不会返回到输入端。从时序电路框图可以看出,输出的信号可以返回到输入。因此,组合电路的特点是任一时刻电路的输出状态仅取决于该时刻各输入信号状态的组合,而与输入信号作用前电路的原输出状态无关。

图 3.1.1 虚线框中是多输入、多输出的组合逻辑电路。 $X_1, X_2, \dots, X_n$ 是电路的输入信号, $F_1, F_2, \dots, F_m$ 是输出信号,每一个输出信号是全部或者部分输入信号的函数,输出信号与输入信号的逻辑关系可用逻辑表达式

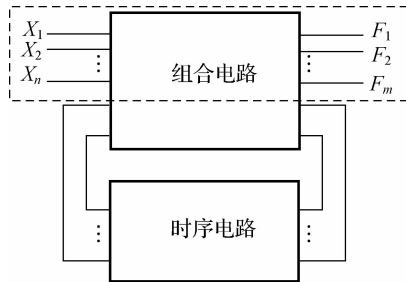


图 3.1.1 数字电路方框图

表示为

$$F_i = f_i(X_1, X_2, \dots, X_n), i = 1, 2, \dots, m$$

综上所述,组合逻辑电路在电路结构上的特点如下。

- (1)由集成逻辑门组成。
- (2)只有输入到输出的通路,没有从输出到输入的反馈回路。
- (3)电路不包含时序电路(记忆、存储)元件。

3.1.1 组合逻辑电路的分析

分析逻辑电路时,首先确定电路的性质,观察输入信号是否单方向从输入流经电路到输出,不存在反馈信号。如果具备这样的性质,就是组合逻辑电路。组合逻辑电路的分析是根据给定的逻辑电路图找出其输出信号与输入信号之间的逻辑关系,从而确定逻辑功能。组合逻辑电路的分析框图如图 3.1.2 所示,分析步骤如下。

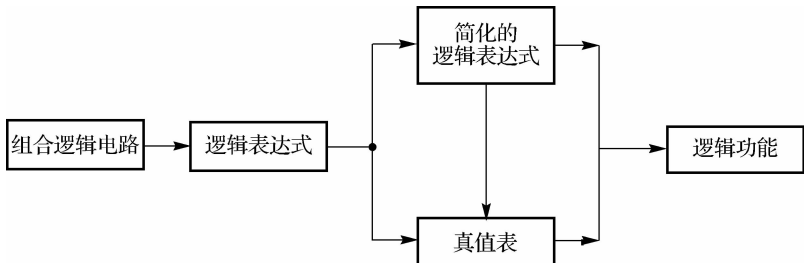


图 3.1.2 组合逻辑电路分析步骤

(1)根据给定逻辑电路图,从电路的输入到输出逐级写出各输出端的逻辑函数,最后得到描述输出与输入信号关系的逻辑函数。

(2)利用公式法或卡诺图法,简化或变换输出逻辑表达式,使逻辑关系更加清晰、可读。如果能够直接判断电路功能,可省去步骤(3)。

(3)根据逻辑函数表达式或最简逻辑函数表达式,列出真值表。

(4)根据真值表,确定电路的逻辑功能。

例 3.1.1 分析多输入、单输出电路的逻辑功能,电路如图 3.1.3 所示。

解:输入到输出不存在反馈信号,信号是从输入单方向到输出,确定是组合逻辑电路。

(1)由逻辑电路图逐级写出逻辑表达式。首先借助第一级的输出信号 F_1 、 F_2 和 F_3 ,写出其逻辑表达式。然后代入到输出逻辑函数 F 中。

$$F_1 = \overline{AB} \quad F_2 = \overline{BC} \quad F_3 = \overline{AC}$$

$$F = \overline{F_1 \cdot F_2 \cdot F_3}$$

(2)化简与变换。将上面得到的逻辑表达式进行化简或变换成熟悉的与或两级表达式,以方便列出真值表或直接表述电路逻辑功能。

$$F = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{CA}} = AB + BC + CA$$

(3)由表达式列出真值表。步骤(2)中,经过化简与变换的表达式有 3 个乘积项之和,所以很容易列出真值表,如表 3.1.1 所示。

表 3.1.1 例 3.1.1 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

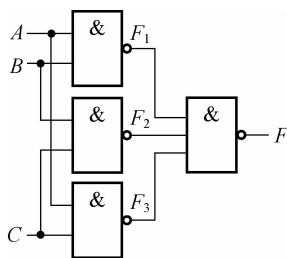


图 3.1.3 例 3.1.1 电路图

(4)分析逻辑功能。由真值表可知,当 A、B、C 三个输入信号中有两个为“1”时,电路输出为“1”,可以称这个组合逻辑电路为“表决电路”或“少数服从多数电路”。

例 3.1.2 分析多输入、多输出电路的逻辑功能,电路如图 3.1.4 所示。

解:(1)图 3.1.4 的电路比图 3.1.3 电路复杂一些,有 4 个输入信号 A_1 、 A_0 、 B_1 和 B_0 , 3 个输出信号 F_1 、 F_2 和 F_3 。对于多输出信号的组合逻辑电路,分析方法也是相同的。由逻辑电路图逐步写出逻辑表达式。

$$F_1 = A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0 + A_0 \bar{B}_1 \bar{B}_0$$

$$F_2 = \bar{A}_1 B_1 + \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_0 B_1 B_0$$

$$F_3 = \bar{A}_1 \bar{B}_1 \bar{A}_0 \bar{B}_0 + A_1 A_0 B_1 B_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 \bar{A}_0 B_1 \bar{B}_0$$

(2)对于较复杂的逻辑表达式,可直接根据逻辑表达式列出真值表,如表 3.1.2 所示。

表 3.1.2 例 3.1.2 真值表

A_1	A_0	B_1	B_0	F_3	F_2	F_1
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	1	0	0

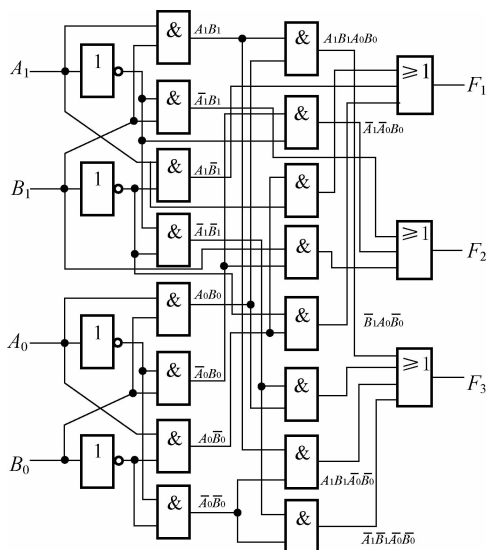


图 3.1.4 例 3.1.2 电路图

(3)由真值表可以看出,输入信号 $A_1 A_0$ 、 $B_1 B_0$ 是两位二进制数进行比较,输出信号之一 F_1 是当 $A_1 A_0$ 大于 $B_1 B_0$ 时为高电平, F_2 是当 $A_1 A_0$ 小于 $B_1 B_0$ 时为高电平, F_3 是当 $A_1 A_0$ 等于 $B_1 B_0$ 时为高电平。因此,图 3.1.4 所示电路是两位二进制数比较电路。

从以上两例题可以看出,虽然掌握了组合逻辑电路的分析步骤,但阐述组合逻辑电路功能更主要是依据对输入/输出信号和逻辑电路的熟练程度和经验。

在实际工程工作中,经常会遇到对已设计好的电路进行评估,如研究电路有确定输出时,其输入的必备条件,以及电路的故障诊断等大量的数字电路分析问题,对于数字电路故障诊断等问题请参考文献 3。

3.1.2 组合逻辑电路的设计

组合逻辑电路设计习惯上称为组合逻辑电路综合,是组合电路分析的逆过程。根据给定欲实现组合电路功能的文字描述,设计出实现相应功能的最简单或者最合理的组合逻辑电路。

组合逻辑电路的设计方法有多种,可采用小规模、中规模和可编程逻辑器件实现。虽然采用的器件和设计方法不相同,但基本逻辑问题的描述和考虑的实际工程问题是具有共性的。本章先介绍用小规模集成电路实现给定功能的逻辑电路的设计方法,再介绍用中规模和可编程逻辑器件实现组合逻辑电路。组合逻辑电路设计的一般步骤如下。

1. 确定输入/输出变量

从具有一定因果关系的文字描述事件中,确定输入信号(变量)和输出信号(变量)。分析事件的因果关系是把引起事件的原因设置为输入变量,把事件产生的结果设置为输出变量。确定逻辑输入/输出的哪种状态用逻辑 0 表示,哪种状态用逻辑 1 表示。

2. 列真值表和写逻辑表达式

在真值表中,列出输入变量的所有(全)组态,根据步骤 1 的因果关系写出对应输入的输出变量,依据真值表写出逻辑表达式。

3. 选择器件类型

根据问题的具体要求、器件的功能和资源情况,采用小规模集成电路(SSI)、中规模集成组合逻辑器件(MSI)或可编程器件(PLD)实现所需要的逻辑电路。

4. 逻辑函数化简或变换

用小规模集成电路进行设计时,其步骤如下。

- (1)将逻辑表达式化为最简。
- (2)变换为与器件类型相适应的表现形式,如与非门、扇入系数等。
- (3)同时考虑逻辑电路的实际工程问题,即传递延时、驱动能力等。

5. 画出逻辑电路图

根据步骤 4 设计出逻辑电路图。

下面用小规模集成逻辑门举例说明设计组合逻辑电路的方法和步骤。

例 3.1.3 请设计一表决电路。共有 4 人参加某学生集体的三好学生投票,多数人投赞成票可以通过,其中班主任投赞成票也通过,即班主任具有一票通过权。

解:(1)确定输入/输出变量。确定输入 A 、 B 、 C 和 D 为投票人,且 A 为班主任。输出 F 为结果“通过”成立。投票人投同意票为逻辑 1,不同意为逻辑 0。输出通过为逻辑 1,不通过为逻辑 0。

(2)列真值表和写逻辑表达式。在真值表 3.1.3 中,列出输入变量 A 、 B 、 C 和 D 的所有(全)组态,根据题意列出输出变量,依据真值表写出逻辑表达式为 $F=A+\overline{A}BCD$ 。