



面向21世纪课程教材
Textbook Series for 21st Century

电子技术基础

数字部分 (第四版)

华中理工大学电子学教研室编

主 编 康华光

副主编 邹寿彬

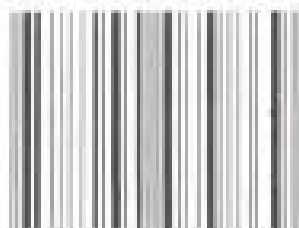


高等教育出版社
HIGHER EDUCATION PRESS

Fundamentals of Electronic Technique(Digital)



ISBN 7-04-007986-0



9 787040 079869 >

定价 27.80 元



面向 21 世纪课程教材
Textbook Series for 21st Century

电子技术基础

数字部分 (第四版)

华中理工大学电子学教研室编

主 编 康华光

副主编 邹寿彬



高等教育出版社
HIGHER EDUCATION PRESS

图书在版编目(CIP)数据

电子技术基础. 数字部分/康华光主编. —4版.
北京:高等教育出版社,2000.7 (2002重印)
面向21世纪课程教材
ISBN 7-04-007986-0

I. 电… II. 康… III. ①电子技术-教材②数字电
路-教材③逻辑电路-教材 IV. TN01

中国版本图书馆CIP数据核字(2000)第17622号

电子技术基础 数字部分 (第四版)
康华光 主编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街55号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 中国科学院印刷厂

开 本 787×960 1/16

版 次 1980年1月第1版

印 张 33

2000年6月第4版

字 数 610 000

印 次 2002年2月第6次印刷

插 页 1

定 价 27.80元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等
质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

第四版序

在电子技术日新月异的形势下,为了培养面向 21 世纪的电子技术人才,本书在第三版的基础上,经过教学改革与实践,对其内容作了较大的修改和更新,使之更符合电子信息时代的要求。在修订过程中,依照 1995 年教育部(原国家教委)颁发的《高等工业学校电子技术基础课程教学基本要求》,提出了如下的总思路:精选内容,推陈出新;讲清基本概念、基本电路的工作原理和基本分析方法。对主要的技术指标,采用工程近似方法进行计算,至于更全面的分析与设计则可借助 CAD 技术来实现。这将有利于读者开拓思路。具体考虑有如下几点:

1. 加强电子系统与信号的概念,为学习模拟电路和数字电路提供了引导性的背景知识。

2. 增加了部分新器件的内容,如砷化镓场效应管(MSFET)、VMOS 功率器件、BiCMOS 门电路、现场可编程逻辑器件(如 CPLD、FPGA)等,以适应新技术发展的需要。

3. 将三端有源器件(BJT、FET)的 6 种电路组态(共射、共集、共基和共源、共漏、共栅)归结为 3 种通用的电路组态,即反相电压放大器、电压跟随器和电流跟随器,这就有利于电子电路的分析与综合,也为学习和使用 BiFET 和 BiCMOS 等一类新型集成电路器件奠定了基础。

4. 根据当前教学上的需要与设备条件的可能性,模拟部分增设了“电子电路的计算机辅助分析与设计”一章,为电子电路仿真与设计作了引导性的介绍,并利用 PSPICE 软件解题;数字部分增设了“数字系统设计基础”一章,将硬件描述语言 ABEL 和 ISP Synario 软件作为数字系统设计的入门性工具,并附有设计实例。

5. 为便于读者深入理解教材内容,加强了例题,其中部分电路具有实用性。同时也重编了具有启发意义的复习思考题和习题,并附有少量的 CAD 例题和习题供各校师生灵活选用。

参加本版模拟部分修订工作的有瞿安连(第 1 章)、康华光(第 2、3、7 章)、陈大钦(第 4、5、8、9 章)、王岩(第 6、8、10 章)、张林(第 11 章及附录)等同志。参加数字部分修订工作的有康华光(第 1、2 章及附录 A、B、C、E、F、G)、邹寿彬(第 3、4、5 章)、杨华(第 6、7 章)、张林、李玲(第 8 章)、彭容修(第 9、10 章)、秦臻(第 11 章)、罗杰(附录 D 和第 8 章的部分内容)。康华光同志为主编,负责全书的策

划、组织和定稿。陈大钦和邹寿彬同志分别为模拟部分和数字部分的副主编,协助主编工作。此外,杨华同志负责重编了模拟部分第2、3、7章的习题和第1章的校订工作;张林和罗杰二位同志协助有关各章的编者,完成了全书的CAD例题和习题的解答工作。

本书由东南大学衣承斌教授主审,参加审阅的,模拟部分为刘京南教授、李桂安副教授;数字部分为皇甫正贤教授、戴义宝副教授。第三版发行期间,承全国各兄弟院校师生给我们以鼓励,寄来了不少宝贵意见和建议,编者在此一并致以衷心的感谢。

编者

1998年7月于武汉华中理工大学

初 版 序

本书是根据高等学校工科基础课电工、无线电类教材编写会议(1977年11月合肥会议)所制订的“电子技术基础”(电力类)教材编写大纲编写的。在编写过程中,我们力图以马列主义、毛泽东思想为指导,运用辩证唯物主义观点和方法来阐明本学科的规律。

“电子技术基础”是电力工程类各专业的一门技术基础课,它是研究各种半导体器件的性能、电路及其应用的学科。从本学科内容大的方面来划分,本书上、中两册属模拟电子技术,下册属数字电子技术;前者主要是讨论线性电路,后者则着重讨论脉冲数字电路。

教材中注意总结我们近年来的教学实践经验,加强了基础理论,如加强了半导体的物理基础和电路的基本分析方法;同时也注意吸取国内外的先进技术,如加强了线性集成电路和数字集成电路(包括中、大规模集成电路)的原理和应用,新增了电子电路的计算机辅助分析等内容。

在内容的安排上,注意贯彻从实际出发,由浅入深、由特殊到一般、从感性上升到理性等原则。通过各种半导体器件及其电路来阐明电子技术中的基本概念、基本原理和基本分析方法。对于基本的和常用的半导体电路(包括脉冲数字电路),除了作定性的分析外,还介绍了工程计算或设计方法。为了加深对课堂知识的理解,列举了若干电路实例,并配有一定数量的例题、思考题和习题。

在使用本教材时,请注意下列几点:

(1) 本课程是在学完普通物理学和电工原理的大部分内容之后开设的,课程之间的相互配合和衔接非常重要。例如,在第一章用能带理论来解释半导体内两种载流子——电子和空穴的导电规律时,应以普通物理学中讲的固体能带理论为基础;又如在分析放大器时,既讨论了稳态分析(频域),也介绍了瞬态分析(时域),在“运算放大器”一章中,又有积分、微分电路以及其他应用,这些内容应以电工原理中的无源线性电路的瞬态分析为基础,只有配合得好,才能取得满意的效果。

(2) 本教材是按课程总学时数约200(包括实验课等环节)而编写的,除了基本内容之外,还编入了部分较深入的内容,这些内容均在标题前注有星号(*)或用小字排印,自成体系。不同专业可按学时多少,由教师灵活选择,也可供读者自学参考。

(3) 课程中各个教学环节的配合十分重要,除了课堂讲授外,还必须通过习题课和实验课等环节加以补充,有些内容可以把这几个环节有机地结合起来。对于实验课,必须予以高度重视,通过实验课,不仅可以验证理论,加深对理论知识的理解,更重要的是,可以学会电子测试技术,使理论紧密结合实践。

参加本书编写工作的有汤之璋(第一章)、陈婉儿(第一、二、九章)、陈大钦(第三、五、十章)、康华光(第四、十一章)、王岩(第六、七、十三章)、林家瑞(第六章)、邹寿彬(第八、十二章)、周劲青(第十一章)和江庚和(第十三章)等同志,最后由康华光同志定稿。在编写过程中,张瑾、朱立群、赵月怀、肖锡湘、杨华、石友惠、汪菊华、罗玉兰以及其他同志参加了许多工作,给予很大支持。

本书由南京工学院李士雄副教授主审,参加主审工作的还有江正战、张志明、衣承斌、陈黎明和丁康源等同志。

在武汉和南京举行的审稿会上,承西安交通大学沈尚贤教授、清华大学童诗白教授、浙江大学邓汉馨副教授、上海交通大学徐俊荣副教授以及重庆大学、山东工学院、沈阳机电学院、合肥工业大学、大连工学院、湖南大学、华南工学院、同济大学、哈尔滨工业大学、天津大学、太原工学院和昆明工学院等兄弟院校的教师代表对初稿进行了认真的审阅,并提出了许多宝贵的意见。

在编写本书第八章(电子电路的计算机辅助分析)的过程中,承中国科学院湖北岩体土力学研究所计算机室协助解题。

对所有为本教材进行审阅并提出宝贵意见以及在编写出版过程中给予热情帮助和支持的同志们,我们在此一并表示衷心的感谢。

由于我们的水平有限,加之时间比较仓促,书中错误和不妥之处,在所难免,殷切希望使用本教材的师生及其他读者,给予批评指正。

编 者

1979年3月

第二版序

本书是在第一版的试用基础上,并按照高等工业学校《电子技术基础教学大纲》(草案)(四年制自动化类和电力类专业试用),总结提高、修改增删而成的。主要做了下列几方面的工作:(1)从本课程的目的和任务出发,在保证打好基础的前提下,精选了内容,例如删去了“电子电路的计算机辅助分析”一章,适当精简了器件内部的物理过程、放大器的频率特性分析、分立元件电路以及设计方面的内容等,在篇幅上有较大的缩减;(2)删繁就简改写了第二、四、六章的大部分内容。同时,将第一版的第九、十章各分为两章,以利于教学;(3)增加了部分新内容,如集成运算放大器的应用电路,中规模数字集成电路等;(4)加强了电路分析方法,如用“虚短”的概念分析集成运算放大器的线性应用电路;在数字电路中,突出了组合逻辑与时序逻辑电路的分析方法;(5)近几年来,由于大规模集成电路的飞速发展,出现了微处理机对各个科学技术领域的渗透,为此,我们充实了“MOS数字集成电路”一章的内容;(6)重新整理并增删了各章所附的思考题和习题。此外,在编排上,把基本内容排大字,选讲内容排小字,自学参考内容既排小字,又带*号。

本版各章基本上由原编者修订,参加的人员有汤之璋、康华光、陈婉儿、王岩、陈大钦、邹寿彬、朱立群等同志,全书由康华光同志定稿。在修订过程中,得到了汤之璋教授的帮助与指导,陈婉儿同志协助校阅了第一至第六章的书稿,肖锡湘、陈晓天、丘小云、石友惠、罗玉兰以及其他同志参加了许多工作。

本书由南京工学院李士雄教授主审,参加审阅工作的还有陈天授、陈黎明、皇甫正贤、郑虎申等同志;在本书第一版的试用期间,承全国有关兄弟院校的师生寄来不少宝贵意见和建议,编者在此深表谢忱。

本版内容虽有所改进,但离教学要求尚有差距,恳请使用本教材的师生和其他读者予以批评指正,以便不断提高。

编者

1982年10月于武汉

第三版序

自本书第一版问世以来,已经历了近十年。在这期间,电子技术领域发生了迅猛而巨大的变化。新技术革命和教学改革不断深入,促使本教材不断改进完善,第三版现在与读者见面了。

新版是在第二版的基础上,经过改革试验、总结提高、修改增删而成的。在修订工作中,依照1987年经国家教委批准的《高等工业学校电子技术课程教学基本要求》,在保证基本教学内容的前提下,为适应电子技术不断发展的新形势和教学上的灵活性以及因材施教的需要,本版适当增加了部分加宽加深的选讲内容,具体考虑如下:

1. 新版在体系上作了较大的调整。在模拟部分中,将“模拟集成电路”一章的位置提前,以致有可能在“反馈放大器”以及后续各章中,均以模拟集成电路为对象进行讨论,这就形成了以模拟集成电路为主干的体系。数字部分则直接以小规模数字集成电路引路,逐步向中大规模集成电路深入,几乎大部分内容都纳入“组合逻辑”和“时序逻辑”两大类电路之中。

2. 在保证基本理论完整性的原则下,删去或精简了一些分立元件电路内容,增强了集成电路的应用,并引入模拟乘法器、开关电容滤波器、压控振荡器、锁相环、直流变换器、门阵列、算术逻辑单元、动态存储器、集成A/D与D/A转换器等新技术内容。

3. 为了开拓学生的知识广度,新增了“调制与解调”一章。

4. 本书数字部分的内容安排与讲述方法,注意到了与“微处理器基础”的密切联系,以利于压缩学时,提高教学效果。

5. 为了贯彻理论联系实际的原则,书中以不同的方式,安排了一定数量的电路实例,并注意阅读电子电路图和查阅电子器件手册的训练。

6. 教材正文与例题、习题紧密配合。例题是正文的补充。某些内容则有意地让读者通过习题来掌握,以调节教学节律,利于理解深化。

7. 在编排上,对于加宽加深的内容,均注有*号,以便于教师选讲和读者自学参考。

本版仍沿用从模拟到数字的体系,若有需要,亦可按从数字到模拟的体系讲授,只需将模拟部分的“半导体二极管和三极管”一章移到数字部分之前讲授即可。

参加新版模拟部分修订工作的有汤之璋(第1章)、康华光(第1、2、6、7章)、王岩(第5、8、11章)和陈大钦(第3、4、8、9、10章及附录A)等同志。参加数字部分修订工作的有康华光(第1、2章)、邹寿彬(第3、4、7章)和赵德宝(第5、6章及附录A)等同志。康华光同志为主编,负责全书的组织和定稿。陈大钦和邹寿彬同志分别为模拟和数字部分的副主编,协助主编工作。在修订过程中,得到了汤之璋教授的支持与帮助。赵德宝、瞿安连、肖锡湘同志协助校订了模拟部分的原稿。陈大钦、瞿安连同志协助校订了数字部分的原稿。丁素芳、罗杰、杨晓安和汪菊华等同志绘制了全书的插图。教研室的其他同志也参加了部分工作。

本书由南京工学院李士雄教授主审,负责组织审稿工作的为衣承斌副教授,参加审阅的,模拟部分为衣承斌、陈黎明、陈天授副教授,李桂安讲师;数字部分为丁康源副教授,郑虎申、严振祥、皇甫正贤讲师。在第二版发行期间,承全国许多师生给我们以鼓励,寄来了不少宝贵意见和建议,编者谨此一并致以谢忱。

本版虽有所改进提高,但离教学改革的要求尚远。敬希读者予以批评指正。

编者

1987年8月于武昌华工园

本书常用符号表(续)

A_0, A_1, A_2, \dots	第 0、1、2、 \dots 位译码器地址输入	N	计数器计数输入端
$A > B, A = B, A < B$	数字比较器 $A > B, A = B, A < B$ 输出	N_i	扇入数
BCD	二-十进制码	N_o	扇出数
$C(CP, CLK)$	进位数, 触发器时钟脉冲输入端	P_D	功耗
C_{ext}	外接电容端	P_{OFF}	空载截止功耗
$CR(CLR)$	清零	P_{ON}	空载导通功耗
CS	片选信号输入	Q	触发器输出
D	D 触发器输入, 数据输入	q	占空比
D_S	移位寄存器串行输入	R	RS 触发器输入
D_{SR}	右移串行输入	R_D	触发器的直接置 0 端
D_{SL}	左移串行输入	S	RS 触发器的输入、和数
E	使能控制端	S_D	触发器的直接置 1 端
$E_1; E_0$	使能输入; 使能输出	T	周期
FF	触发器	T	BJT
G	逻辑门	T_N	N 沟道 MOSFET
G	进位产生变量	T_P	P 沟道 MOSFET
f_{max}	最高工作频率	t	时间
I_{BS}	临界饱和基极电流	t_d	延迟时间
I_{CS}	集电极饱和电流	t_f	下降时间
I_{IL}	低电平输入电流	t_r	上升时间
I_{IH}	高电平输入电流	t_s	存储时间
J, K	JK 触发器输入	t_t	渡越时间
L	逻辑函数	t_{on}	开通时间
LD	预置控制	t_{off}	关闭时间
m	最小项	t_{pd}	平均传输延迟时间
		t_{PHL}	输出由高电平变为低电平时的传输延迟时间
		t_{PLH}	输出由低电平变为高电平时的

	的传输延迟时间	V_{OFF}	关门电平
t_w	脉冲宽度	V_{th}	阈值电压
t_r	恢复时间	$V_{\text{T}+}$	施密特触发特性的正向阈值电 压
V_{CES}	BJT 的饱和压降	$V_{\text{T}-}$	施密特触发特性的负向阈值电 压
V_{CEO}	BJT 的截止压降	$V_{\text{CC}}, V_{\text{DD}}$	电源电压
V_{H}	滞后电压	V_{REF}	参考电压
V_{NH}	高电平噪声容限电压	\times	任意态,无该项
V_{NL}	低电平噪声容限电压	\uparrow	从低电平到高电平的跳变
V_{OH}	输出高电平时的电压	\downarrow	从高电平到低电平的跳变
V_{OL}	输出低电平时的电压	\square	高电平脉冲
V_{SH}	标准高电平	\sqcup	低电平脉冲
V_{SL}	标准低电平		
V_{ON}	开门电平		

内 容 简 介

本书是教育部“高等教育面向 21 世纪教学内容和课程体系改革计划”的研究成果,是面向 21 世纪教材和普通高等教育“九五”国家级重点教材。前版曾获国家优秀教材特等奖及国家级科技进步二等奖。

为适应电子信息时代的新形势和培养面向 21 世纪电子技术人才的迫切需要,在第三版的基础上,经过教学改革与实践,对内容作了较大的修改。精选了常规内容,增加了电子系统与信号的基本知识以及新器件、新技术方面的内容,其中包括运用 CAD 技术对电子电路进行分析与设计的新方法。重编了例题、复习思考题和习题,便于教学。

本书分模拟部分和数字部分出版。模拟部分包括:绪论(电子系统与信号)、半导体二极管及其基本电路、半导体三极管及放大电路基础、场效应管放大电路、功率放大电路、集成运算放大器、反馈放大电路、信号的运算与处理电路、信号产生电路、直流稳压电源和电子电路的计算机辅助分析与设计。数字部分包括:数字逻辑基础、逻辑门电路、组合逻辑电路的分析和设计、常用组合逻辑功能器件、触发器、时序逻辑电路的分析和设计、常用时序逻辑功能器件、存储器和可编程逻辑器件、脉冲波形的产生与变换、数模与模数转换器以及数字系统设计基础。

本书由东南大学衣承斌教授主审。

本书可作为高等学校电气信息类(包括原自动化类、电气类、电子类等)专业“电子技术基础”课程的教材,也可供从事电子技术工作的工程技术人员参考。



面向 21 世纪课程教材



普通高等教育“九五”
国家教委重点教材



主编简介

康华光 湖南衡山人,1925年8月出生。现为华中理工大学教授、博士生导师。长期从事电子技术教学与生物医学工程研究。

康教授1951年毕业于武汉大学电机工程学系并留校任教。1953年院系调整到华中理工大学(原华中工学院)工作至今。现任中国电子学会生物医学电子学分会委员。曾任国家教委高校工科电工课程教学指导委员会副主任兼电子技术课程教学指导小组组长。

由康华光主编的《电子技术基础》(模拟、数字部分)第一、二、三版(高等教育出版社,1979、1982、1988年)曾先后于1988、1992、1996年荣获国家级优秀教材奖、国家级优秀教材特等奖和国家科技进步二等奖。主持研究的“优化电子技术基础课程建设”项目,荣获1989年国家级优秀教学研究成果奖。

在科研方面,主要从事生物医学信息的检测与分析以及细胞电生理研究。建立了国内第一个具有国际先进水平的细胞信使实验室。主持了多项国家级科研课题,开展国内、国际交流与合作,成绩显著。培养了博士、硕士生40余名,发表了近百篇学术论文和专著《膜片钳技术及其应用》。

目 录

1 数字逻辑基础	1
引言	1
1.1 模拟信号与数字信号	1
1.1.1 模拟信号	1
1.1.2 数字信号	3
1.1.2.1 二值数字逻辑和逻辑电平	3
1.1.2.2 数字波形	4
1.1.2.3 模拟量的数字表示	7
1.2 数字电路	9
1.2.1 数字电路的发展与分类	9
1.2.2 数字电路的分析方法与测试技术	10
1.3 数制	11
1.3.1 十进制	11
1.3.2 二进制	12
1.3.3 十、二进制之间的转换	15
1.3.4 十六进制和八进制	17
1.4 二进制码	19
1.5 基本逻辑运算	21
1.6 逻辑函数与逻辑问题的描述	24
小结	25
习题	26
2 逻辑门电路	28
引言	28
2.1 二极管的开关特性	28
2.2 BJT 的开关特性	32
2.2.1 BJT 的开关作用	32
2.2.2 BJT 的开关时间	34
2.3 基本逻辑门电路	36
2.3.1 二极管与门及或门电路	36
2.3.2 非门电路——BJT 反相器	39

2.4	TTL 逻辑门电路	40
2.4.1	基本的 BJT 反相器的动态性能	40
2.4.2	TTL 反相器的基本电路	41
2.4.3	TTL 反相器的传输特性	43
2.4.4	TTL 与非门电路	45
2.4.5	TTL 与非门的技术参数	46
2.4.6	TTL 或非门、集电极开路门和三态门电路	50
2.4.7	改进型 TTL 门电路——抗饱和 TTL 电路	55
2.5	射极耦合逻辑门电路	58
2.6	CMOS 逻辑门电路	61
2.6.1	CMOS 反相器	61
2.6.2	CMOS 门电路	64
2.6.3	BiCMOS 门电路	66
2.6.4	CMOS 传输门	68
2.6.5	CMOS 逻辑门电路的技术参数	69
2.7	NMOS 逻辑门电路	70
2.8	正负逻辑问题	72
2.9	逻辑门电路使用中的几个实际问题	72
2.9.1	各种门电路之间的接口问题	73
2.9.2	门电路带负载时的接口电路	74
2.9.3	抗干扰措施	76
2.10	CAD 例题	77
	小结	82
	习题	83
	CAD 习题	88
3	组合逻辑电路的分析与设计	89
	引言	89
3.1	逻辑代数	89
3.1.1	逻辑代数的基本定律和恒等式	90
3.1.2	逻辑代数的基本规则	91
3.1.3	逻辑函数的代数变换与化简法	92
3.2	逻辑函数的卡诺图化简法	95
3.2.1	最小项的定义及其性质	96
3.2.2	逻辑函数的最小项表达式	97
3.2.3	用卡诺图表示逻辑函数	98
3.2.4	用卡诺图化简逻辑函数	102
3.3	组合逻辑电路的分析	106
3.4	组合逻辑电路的设计	108

3.5 组合逻辑电路中的竞争冒险	113
3.5.1 产生竞争冒险的原因	113
3.5.2 消去竞争冒险的方法	115
*3.6 CAD 例题	116
小结	119
习题	119
*CAD 习题	124
4 常用组合逻辑功能器件	125
引言	125
4.1 编码器	125
4.1.1 编码器的定义与功能	125
4.1.2 集成电路编码器	128
4.2 译码器/数据分配器	133
4.2.1 译码器的定义及功能	133
4.2.2 集成电路译码器	135
4.2.3 数据分配器	141
4.3 数据选择器	143
4.3.1 数据选择器的定义及功能	143
4.3.2 集成电路数据选择器	144
4.4 数值比较器	149
4.4.1 数值比较器的定义及功能	149
4.4.2 集成数值比较器	151
4.5 算术运算电路	156
4.5.1 半加器和全加器	156
4.5.2 多位数加法器	158
4.5.3 减法运算	163
4.5.4 集成算术/逻辑单元举例	166
*4.6 CAD 例题	168
小结	172
习题	173
*CAD 习题	177
5 触发器	179
引言	179
5.1 触发器的电路结构与工作原理	179
5.1.1 基本 RS 触发器	179
5.1.2 同步 RS 触发器	184
5.1.3 主从触发器	186
5.1.4 边沿触发器	188

5.2 触发器的功能	191
5.2.1 RS 触发器	191
5.2.2 JK 触发器	193
5.2.3 T 触发器	197
5.2.4 D 触发器	197
5.3 触发器的脉冲工作特性及主要参数	199
5.3.1 集成触发器的脉冲工作特性	199
5.3.2 集成触发器的主要参数	201
*5.4 CAD 例题	202
小结	205
习题	205
*CAD 习题	211
6 时序逻辑电路的分析和设计	213
引言	213
6.1 时序逻辑电路的基本概念	213
6.1.1 时序逻辑电路的基本结构及特点	213
6.1.2 时序逻辑电路的分类	214
6.1.3 时序逻辑电路功能的描述方法	214
6.2 时序逻辑电路的分析方法	216
6.2.1 分析时序逻辑电路的一般步骤	216
6.2.2 同步时序逻辑电路的分析举例	217
6.2.3 异步时序逻辑电路的分析举例	221
6.3 同步时序逻辑电路的设计方法	223
6.3.1 同步时序逻辑电路设计的一般步骤	223
6.3.2 同步时序逻辑电路设计举例	224
*6.4 CAD 例题	228
小结	231
习题	231
*CAD 习题	238
7 常用时序逻辑功能器件	240
引言	240
7.1 计数器	240
7.1.1 二进制计数器	240
7.1.2 非二进制计数器	246
7.1.3 集成计数器	249
7.2 寄存器和移位寄存器	260
7.2.1 寄存器	260
7.2.2 移位寄存器	261

7.2.3 集成移位寄存器 74194	263
*7.3 CAD 例题	267
小结	269
习题	270
*CAD 习题	273
8 半导体存储器和可编程逻辑器件	275
引言	275
8.1 随机存取存储器(RAM)	275
8.1.1 RAM 的电路结构与工作原理	276
8.1.2 RAM 存储容量的扩展	282
8.1.3 RAM 举例	284
8.2 只读存储器(ROM)	290
8.3 可编程逻辑器件(PLD)	295
8.3.1 PLD 的电路表示法	295
8.3.2 可编程阵列逻辑器件(PAL)简介	298
8.3.3 可编程通用阵列逻辑器件(GAL)	300
8.4 复杂的可编程逻辑器件(CPLD)	308
8.4.1 CPLD 的结构	308
8.4.2 CPLD 的编程	317
*8.5 现场可编程门阵列(FPGA)	323
8.5.1 FPGA 中编程实现逻辑功能的基本原理	323
8.5.2 现场可编程门阵列结构	324
8.5.3 编程实现原理简介	334
小结	338
本章附录	339
习题	340
*CAD 习题	344
9 脉冲波形的产生与变换	347
引言	347
9.1 多谐振荡器	347
9.1.1 门电路组成的多谐振荡器	347
9.1.2 石英晶体振荡器	350
9.2 单稳态触发器	352
9.2.1 门电路组成的微分型单稳态触发器	352
9.2.2 集成单稳态触发器	356
9.2.3 单稳态触发器的应用	361
9.3 施密特触发器	364
9.3.1 门电路组成的施密特触发器	365

9.3.2 集成施密特触发器	367
9.3.3 施密特触发器的应用	369
9.4 555 定时器及其应用	371
9.4.1 555 定时器	372
9.4.2 定时器应用举例	373
*9.5 CAD 例题	378
小结	380
习题	381
*CAD 习题	387
10 数模与模数转换器	388
引言	388
10.1 D/A 转换器	388
10.1.1 倒 T 形电阻网络 D/A 转换器	389
10.1.2 权电流型 D/A 转换器	391
10.1.3 D/A 转换器的输出方式	393
10.1.4 D/A 转换器的主要技术指标	396
10.1.5 集成 D/A 转换器及其应用	398
10.2 A/D 转换器	402
10.2.1 A/D 转换的一般工作过程	402
10.2.2 并行比较型 A/D 转换器	405
10.2.3 逐次比较型 A/D 转换器	407
10.2.4 双积分式 A/D 转换器	410
10.2.5 A/D 转换器的主要技术指标	414
10.2.6 集成 A/D 转换器及其应用	415
*10.3 CAD 例题	418
小结	422
习题	422
*CAD 习题	426
*11 数字系统设计基础	428
引言	428
11.1 数字系统的设计方法	428
11.1.1 数字系统的组成	428
11.1.2 数字系统的设计方法	429
11.1.3 现代数字系统的实现	430
11.2 算法状态机	431
11.2.1 ASM 图符号	431
11.2.2 数字系统的 ASM 图法设计举例	433
11.3 寄存器传输语言	443

11.3.1 寄存器传输语言中的几种操作	444
11.3.2 运用寄存器传输语言设计举例	447
11.4 用可编程逻辑器件实现数字系统	449
11.4.1 交通灯控制系统的实现	450
11.4.2 数字密码锁的实现	452
小结	457
习题	458
*CAD习题	458
附录 A 美国标准信息交换码(ASCII)	460
附录 B 二进制数算术运算	461
附录 C TTL 和 CMOS 逻辑门电路的技术参数	463
附录 D 硬件描述语言 ABEL 及其开发软件 ISP Synario	464
附录 E 电气图用图形符号二进制逻辑单元(GB4728.12-85)简介	477
附录 F 常用逻辑符号对照表	486
附录 G 国产半导体集成电路型号命名法(GB3430-82)	488
参考文献	490
部分习题答案	492
索引(汉英对照)	499
主编简介	504

1 数字逻辑基础

引言 伴随现代电子技术的发展,人们正处于一个信息时代,每天要从周围环境获取大量的信息,例如,电视、广播、印刷媒体等为人们报道世界范围内所发生的各种事件。这些信息通常是通过我们的感觉器官(眼、耳等)进入大脑,并被存储下来,以作进一步的分析。

在电子技术领域里,为了便于存储、分析和传输,常将模拟信号进行编码,即把它转换为数字信号,利用数字逻辑这一强有力的工具来分析和设计复杂的数字电路或数字系统,为信号的存储、分析和传输创造硬件环境。

数字逻辑几乎应用于每一电子设备或电子系统中。计算机、计算器、电视机、音响系统、视频记录设备、光碟、长途电信及卫星系统等,无一不采用数字系统。

本章首先介绍模拟信号与数字信号、数字逻辑的基本概念、数字电路的特点、分析方法及其测试技术,然后讨论数制与码和数字逻辑的基本运算。

1.1 模拟信号与数字信号

1.1.1 模拟信号

从本书模拟部分的讨论已知,模拟信号是时间连续、数值也连续的物理量,它具有无穷多的数值,其数学表达式也较复杂,例如正弦函数、指数函数等。图 1.1.1 表示几种常见的模拟信号。

人们从自然界感知的许多物理量均是属模拟性质的,例如,速度、压力、温度、声音、重量以及位置等等。在工程技术上,为了便于分析,常用传感器将模拟量转换为电流、电压或电阻等电学量。

电流和电压常用图形来表示。图 1.1.2 表示周期性的电压波形,其幅值在 0 V 至 5 V 之间变化,周期为 100 ms。

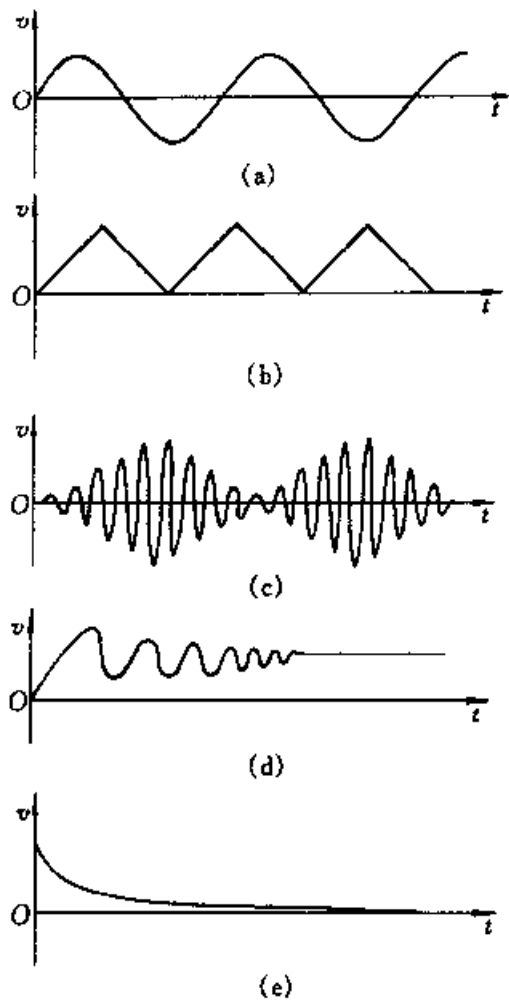


图 1.1.1 几种模拟信号波形

(a) 正弦波 (b) 三角波 (c) 调幅波 (d) 阻尼振荡波 (e) 指数衰减波

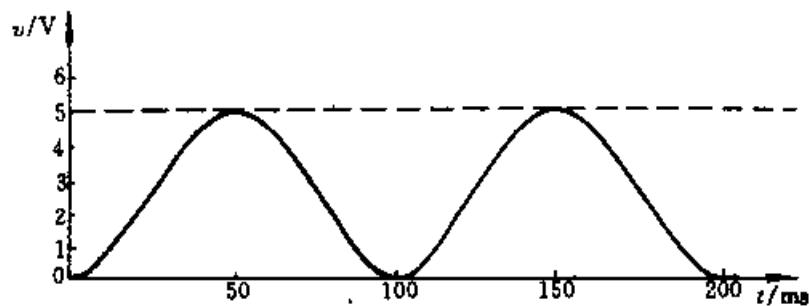


图 1.1.2 周期性电压波形

我们已知,周期性模拟信号的基本参数之一是频率(f),也可用周期(T)表示。频率的单位为赫(Hz),而周期的单位为秒(s),二者之间的关系为 $f = 1/T$ 。对于图 1.1.2 所示的周期性电压波形,其频率 $f = 1/T = 1/100 \text{ ms} = 10 \text{ Hz}$ 。值得指出的是,模拟电压具有无穷多的值。具体地说,以图 1.1.2 为例,在 $0 \sim 5 \text{ V}$ 的范围内,其值的个数为无穷大。

典型的模拟信号包括工频信号、射频信号、视频信号等。我国和欧洲的工频信号的频率为 50 Hz, 美国为 60 Hz。调幅波的射频信号在 530 Hz ~ 1 600 kHz 之间。调频波的射频信号范围为 88 MHz ~ 108 MHz。甚高频(VHF)和超高频(UHF)视频信号在 6 GHz(G 为 10^9)以上。

复习思考题

1.1.1 举出 4 种模拟信号。

1.1.2 以温度为例,试说明怎样把它转换为电信号?

1.1.2 数字信号

电子系统中一般均含有模拟和数字两种构件。模拟电路是系统中必需的组成部分。但为便于存储、分析或传输信号,数字电路更具优越性。在数字电路中,常用二进制数来量化连续变化的模拟信号,而二进制数^①正好是用二值数字逻辑^②中的数字 1 和 0 来表示的,这样就可借助复杂的数字系统(例如计算机)来实现信号的存储、分析和传输。

1.1.2.1 二值数字逻辑和逻辑电平

数字信号在时间上和数值上均是离散的,常用数字 0 和 1 来表示。这里的 0 和 1 不是十进制数中的数字,而是逻辑 0 和逻辑 1,因而称之为二值数字逻辑或简称数字逻辑。

二值数字逻辑的产生,是基于客观世界的许多事物可以用彼此相关又互相对立的两种状态来描述,例如,是与非、真与假、开与关、低与高,如此等等。而且在电路上,可用电子器件的开关特性来实现,由此形成离散信号电压或数字电压。这些数字电压通常用逻辑电平^③来表示。应当注意,逻辑电平不是物理量,而是物理量的相对表示。

表 1.1.1 列出了逻辑电平与电压值的关系。

表 1.1.1

电压/V	二值逻辑	电平
+5	1	H(高电平)
0	0	L(低电平)

* 表中所表示的逻辑体制系正逻辑,即 1 表示高电平,0 表示低电平,负逻辑则相反,详见 2.8 节。

① 二进制数见本章 1.3.2 节。

② 二值数字逻辑是 Binary Digital Logic 的译称。

③ 逻辑电平是 Logic level 的译称。

图 1.1.3 表示用逻辑电平描述的数字波形,其中图 1.1.3a 的逻辑 0 表示 0 V,逻辑 1 表示 5 V。图 1.1.3b 表示 16 位数据的波形。

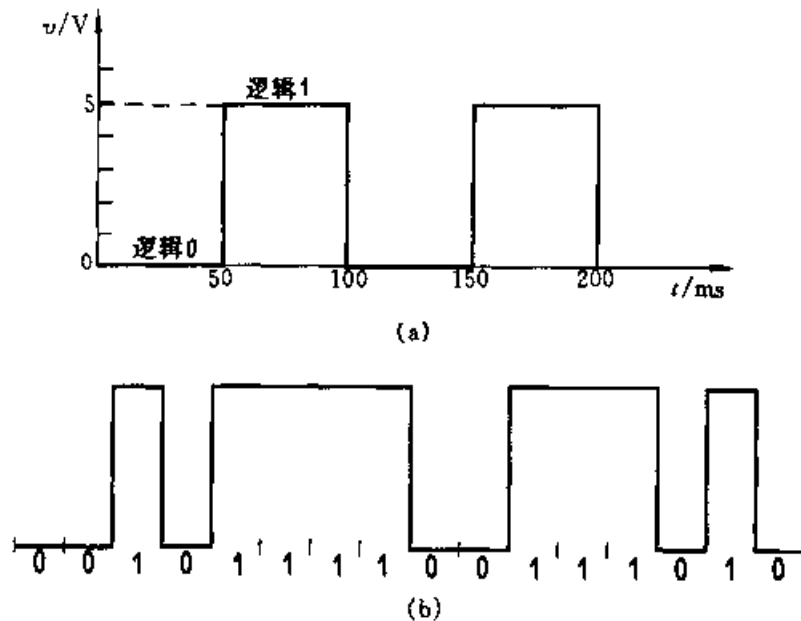


图 1.1.3 数字波形

(a) 用逻辑电平表示数字波形 (b) 16 位数据的图形表示

1.1.2.2 数字波形

如前所述,数字波形是逻辑电平对时间的图形表示。当某波形仅有两个离散值时,通常称之为脉冲波形。此时数字波形与脉冲波形是统一的,只不过所用的表述方式不同而已。前者用逻辑电平表示,而后者则用电压值表示。

与模拟波形的定义相同,数字波形亦有周期性与非周期性之分。图 1.1.4 表示了这两类数字波形。

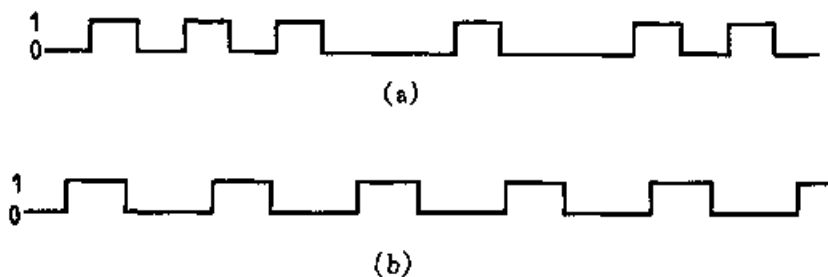


图 1.1.4 数字波形

(a) 非周期性数字波形 (b) 周期性数字波形

周期性数字波形常用周期 T 或频率 f 来描述;而脉冲波形的频率常称为脉冲重复频率 PRR^①。

① PRR 系英文 Pulse Repetition Rate 的缩写。

脉冲波形的脉冲宽度用 t_w 表示,它表示脉冲的作用时间。另一个重要参数是占空比 q ,它表示脉冲宽度 t_w 占整个周期 T 的百分数,常用下式来表示:

$$q(\%) = \frac{t_w}{T} \times 100\% \quad (1.1.1)$$

图 1.1.5 表示两种数字波形及其周期、频率、脉冲宽度和占空比。

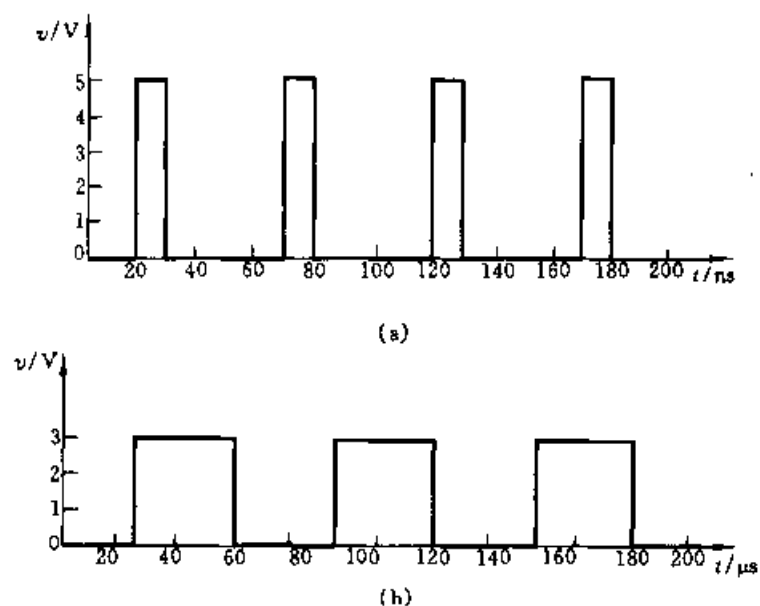


图 1.1.5 周期性数字波形

(a) 幅值 5 V, $t_w = 10$ ns, $T = 50$ ns, $q = 20\%$, $f = \text{PRR} = 20$ MHz

(b) 幅值 3 V, $t_w = 30$ μ s, $T = 60$ μ s, $q = 50\%$, $f = \text{PRR} = 16.67$ kHz

例 1.1.1 设周期性数字波形的高电平持续 6 ms,低电平持续 10 ms,求占空比 q 。

解: 因数字波形的脉冲宽度 $t_w = 6$ ms,周期 $T = 6$ ms + 10 ms = 16 ms,则

$$\text{占空比} = \frac{6 \text{ ms}}{16 \text{ ms}} \times 100\% = 37.5\%$$

在实际的数字系统中,数字波形不能立即上升或下降,而要经历一段时间,因此,有必要定义上升时间 t_r 和下降时间 t_f 。图 1.1.5 所示的数字波形是理想的,认为它的 t_r 和 t_f 均为 0。实际的数字波形是非理想的,它的 t_r 和 t_f 均为有限值,如图 1.1.6 所示。

脉冲波形上升时间的定义是,从脉冲幅值的 10% 到 90% 所经历的时间。脉冲波形的下降时间则相反,即从脉冲幅值的 90% 下降到 10% 所经历的时间。数字信号上升和下降时间的典型值约为几个纳秒(ns),视不同类型的器件和电路而异。至于脉冲宽度则定义为脉冲幅值的 50% 的两个时间点所跨越的时间。上升时间、下降时间和脉冲宽度等 3 个参数如图 1.1.6 所示。

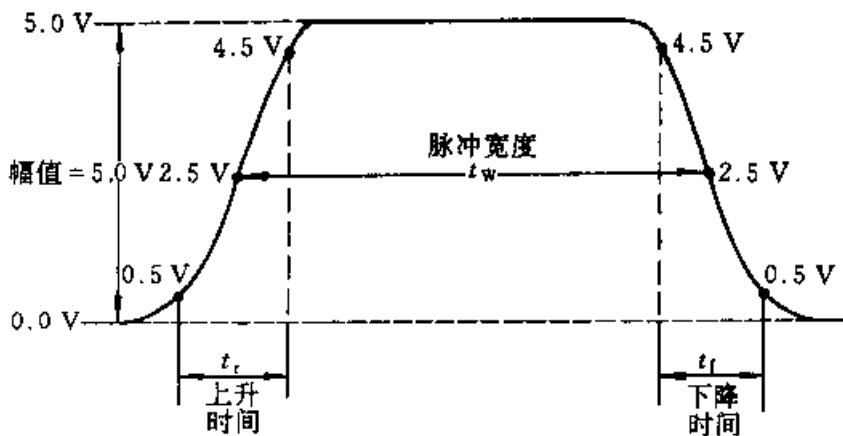


图 1.1.6 非理想脉冲波形

例 1.1.2 试绘出一脉冲波形, 设它的占空比为 50%, 脉冲宽度为 100 ns, 上升时间为 10 ns, 而下降时间为 20 ns。

解: 根据题意, 所绘出的脉冲波形如图 1.1.7 所示。

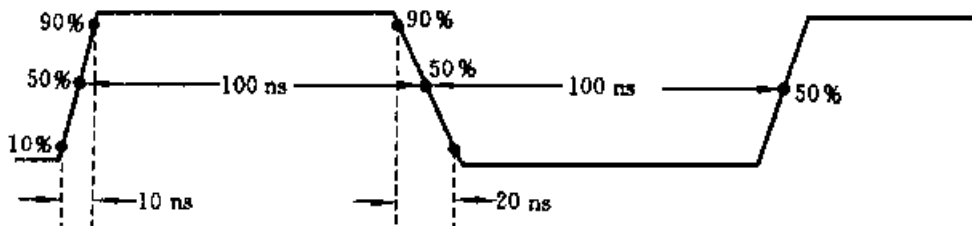


图 1.1.7 例 1.1.2 的波形图

本书所用的数字波形大多数将画成理想波形。实际上, 每一波形均有上升时间和下降时间, 不必在每一波形上表示出来。画波形的目的只是为了知道高、低电平所经历的时间。

图 1.1.8 表示用高、低电平或逻辑 1 和 0 表示的数字波形。图 1.1.8a 为对称方波, 0 和 1 交替地出现。图 1.1.8b 表示一二值位形图^①, 其中 1 或 0 占用的最小时间称为位时间, 意即 1 位数据所占用的时间, 每秒钟所传输数据的位数称为数据率或比特率^②。

例 1.1.3 某通信系统每秒钟传输 1 544 000 位(1.544 兆位)数据, 求每位数据的时间。

解: 按题意, 每位数据的时间为

$$\left[\frac{1.544 \times 10^6}{\text{s}} \right]^{-1} = 647.67 \times 10^{-9} \text{ s} \approx 648 \text{ ns}$$

① 此系 binary bit pattern 的译称。

② 比特率系 bit rate 的译称。

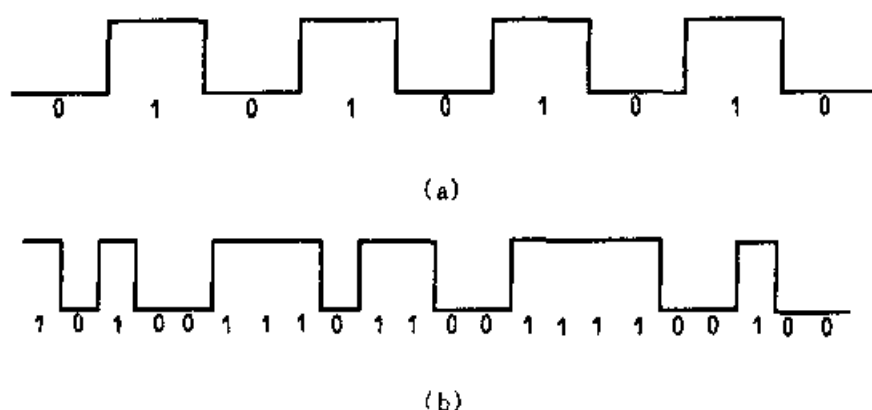


图 1.1.8 用逻辑 1 和 0 表示的二值位形图

(a) 对称方波 (b) 二值数据

表明相互时间关系的多重数字波形图叫做**时序图**^①,时序图中的每一波形常称为时间信号。图 1.1.9 表示一典型的时序图。数字集成电路,包括存储器以及微处理机等均须附有时序图,以便数字系统的分析、应用和设计。图中:CP 为时钟脉冲信号,它是数字系统中的时间参考信号,一般是由石英晶体振荡器来产生^②,波形为对称方波。地址线、片选和数据写入等信号亦示于图中。关于时序图中各个波形的具体作用,在后续各章中将得到深入的理解。

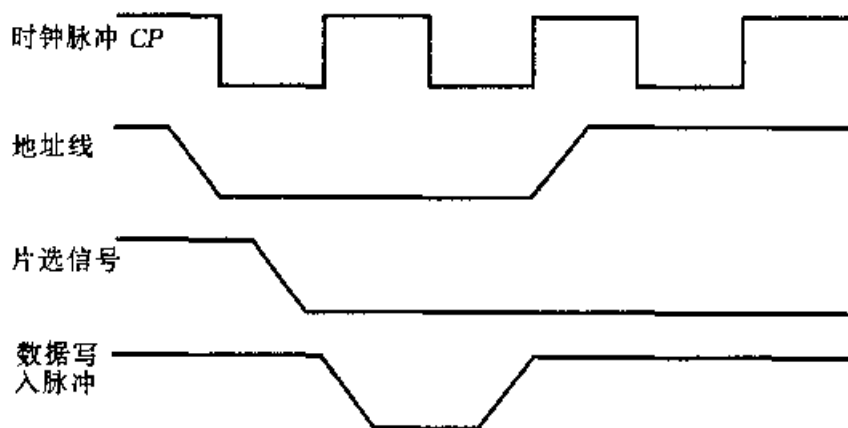


图 1.1.9 数字时序图

1.1.2.3 模拟量的数字表示

前面已经介绍模拟信号和数字信号及其波形。为便于存储、分析和传输,常将模拟信号转换为数字信号。

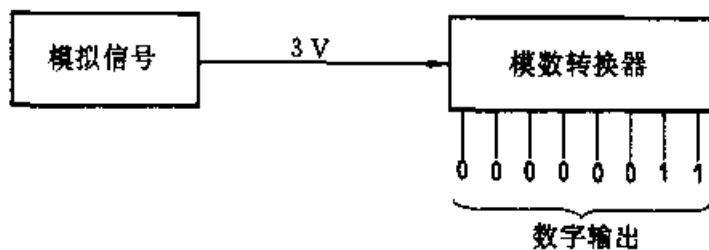
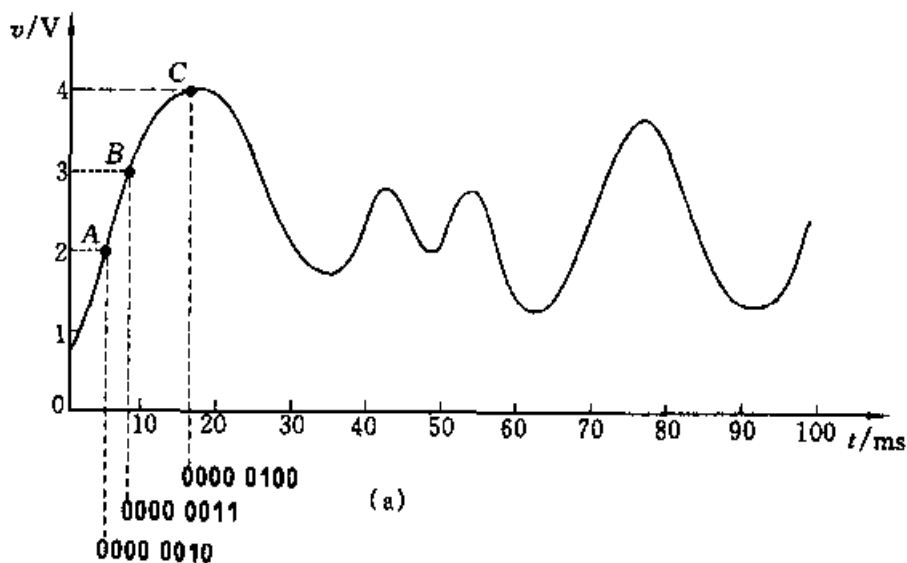
模拟量可以用数字 0、1 的编码来表示,这里的编码所指的是数字 0、1 的字

① 时序图系 Timing diagram 的译称。

② 见第 9 章。

符串,图 1.1.3 和图 1.1.8 所示的数字波形即是这种编码的图形。从下一节的讨论将得知,这种编码就是二进制码。

图 1.1.10a 表示一模拟信号用数字表示的示意图,图中 A、B、C 为模拟信号的 3 个取样点,其值分别用 8 位数字符表示。在 B 点处,模拟电压值为 3 V,相应的数字符为 0000 0011,这个数字符是由模数转换器^① 得来,如图 1.1.10b 所示。其余两个取样点 A 和 C 的电压值分别为 2 V 和 4 V,相应的数字符亦示于图 1.1.10a 中。如果取样点足够多,那么原信号便可较真实地被复制下来。必要时也可用数模转换器还原为模拟信号。



(b)

图 1.1.10 模拟信号的数字表示

(a) 模拟信号波形三个取样点的数字表示 (b) 3 V 模拟电压转换为以 0、1 表示的数字电压

复习思考题

- 1.1.1 什么叫逻辑状态? 并举三例。
- 1.1.2 为什么数字逻辑是二值的?

^① 见第 10 章。

1.2 数字电路

现代数字电路是用半导体工艺制成的若干数字集成器件构造而成。逻辑门是其基本单元。存储器是用来存储二值数据的数字电路。从整体来看,数字电路可分为**组合逻辑电路**和**时序逻辑电路**两大类。

1.2.1 数字电路的发展与分类

数字电路的结构是以二值数字逻辑为基础的,其中的工作信号是离散的数字信号。电路中的电子器件,如二极管、三极管(BJT、FET)处于开关状态,时而导通,时而截止。

数字电路的发展历史与模拟电路一样,经历了由电子管、半导体分立器件到集成电路。但数字集成电路比模拟集成电路的发展更快。从60年代开始,数字集成器件以双极型工艺制成了小规模逻辑器件,随后发展到中规模;70年代末,微处理器的出现,使数字集成电路的性能产生了质的飞跃。

数字集成器件所用的材料以硅材料为主,在高速电路中,也使用化合物半导体材料,例如砷化镓等。

逻辑门是一种重要的逻辑单元电路。TTL^①逻辑门电路问世较早,其工艺经过不断改进,至今仍为主要的逻辑器件之一。随着MOS工艺特别是CMOS^②工艺的发展,TTL的主导地位有被CMOS器件所取代的趋势。

近年来,可编程逻辑器件(PLD^③)特别是现场可编程门阵列(FPGA^④)的飞速进步,使数字电子技术开创了新局面,不仅规模大,而且将硬件与软件相结合,使器件的功能更加完善,使用也更加灵活。

从集成度来说,数字集成电路可分为小规模、中规模、大规模、超大规模和甚大规模等五类。所谓集成度,是指每一芯片所包含的三极管(BJT或FET)的个数。表1.2.1列出了五类数字集成电路的分类依据。

从表1.2.1可以看到,存储器是基本数字部件之一,集成度也很高。利用存储器可以记忆或存储二值数字1或0。存储的数字信息可以取出来分析或直接利用,例如,打印机可从计算机的存储器里取出信息并打印在纸上。通常数字信息的存储视为将信息写入存储器,而信息恢复则理解为从存储器读出信息。

① TTL系 Transistor-Transistor Logic 的缩写。

② CMOS系 Complementary Metal-Oxide-Semiconductor 的缩写。

③ PLD系 Programmable Logic Device 的缩写。

④ FPGA系 Field Programmable Gate Array 的缩写。

表 1.2.1

分类	三极管的个数	典型集成电路
小规模	最多 10 个	逻辑门电路
中规模	10~100	计数器、加法器
大规模	100~1000	小型存储器、门阵列
超大规模	1000~10 ⁶	大型存储器、微处理器
甚大规模	10 ⁶ 以上	可编程逻辑器件、多功能集成电路

利用组合逻辑电路和时序逻辑电路可以控制、操作和运算数字系统中的信息。有关这方面的细节将在后续各章中详细讨论。

1.2.2 数字电路的分析方法与测试技术

1. 数字电路的分析方法

数字电路的主要研究对象是电路的输出与输入之间的逻辑关系,因而在数字电路中不能采用模拟电路的分析方法,例如,小信号模型分析法。由于数字电路中的器件处于开关状态,因而这里所采用的分析工具是逻辑代数,表达电路的功能主要用功能表、真值表、逻辑表达式及波形图。

随着计算技术的发展,为了分析、仿真与设计数字电路或数字系统,可以采用硬件描述语言,例如 ABEL 语言和 ISP Synario^① 软件借助计算机以实现设计自动化,本书在这方面将作入门性的介绍,这种方法对于设计较复杂的数字系统,优点更为突出。

2. 数字电路的测试技术

数字电路在正确设计和安装后,必须经过严格的测试方可使用。测试时必须备有下列基本仪器设备。

数字电压表 用来测量电路中各点的电压,并观察其测试结果是否与理论分析一致。

电子示波器 常用来观察电路各点的波形。一个复杂的数字系统,在主频率信号源的激励下,有关逻辑关系可从波形图中得到验证。逻辑分析仪是一种专用示波器,例如,它可显示 8 至 32 位的数字波形。

复习思考题

- 1.2.1 数字电路或系统与数字集成器件是什么关系?
- 1.2.2 当前两种主要的逻辑门电路是什么?
- 1.2.3 数字电路从整体上来看可分为几大类?

^① 见附录 D。

1.3 数 制

数字电路经常遇到计数问题。人们在日常生活中,习惯于用十进制数,而在数字系统,例如数字计算机中,多采用二进制数,有时也采用八进制或十六进制数。

1.3.1 十进制

1. 十进制的定义

众所周知,中国的珠算盘是一个十进制计数器。任何一个数都可以用0,1,2,3,4,5,6,7,8,9等十个数码,按一定的规律排列起来表示,其计数规律是“逢十进一”,即 $9+1=10$,这右边的“0”为个位数,左边的“1”为十位数,也就是 $10=1\times 10^1+0\times 10^0$ 。所谓十进制就是以10为基数的计数体制。

这样,每一数码处于不同的位置(数位)时,它所代表的数值是不同的。例如,在4位十进制数中,最右边一位的位权为 10^0 ,而最左边一位的位权为 10^3 ,如图1.3.1所示。这与珠算盘横梁上所标示的个、十、百、千的位权是相同的。

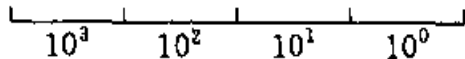


图 1.3.1 十进制数的位权图

例 1.3.1 试用位权来表示十进制数 4567。

解: 将数码与位权相乘,然后相加而得十进制数。

$$\begin{array}{r}
 \begin{array}{cccc}
 4 & 5 & 6 & 7 \\
 | & | & | & | \\
 \hline
 & & & 7 \times 10^0 = 7 \\
 & & & \hline
 & & & 6 \times 10^1 = 6 \times 10 = 60 \\
 & & & \hline
 & & & 5 \times 10^2 = 5 \times 100 = 500 \\
 & & & \hline
 & & & 4 \times 10^3 = 4 \times 1000 = 4000 \\
 & & & \hline
 & & & + \\
 & & & \hline
 & & & 4567
 \end{array}
 \end{array}$$

2. 十进制数的表达式

一般地说,任意十进制数可表示为

$$N_D^{①} = \sum_{i=-\infty}^{\infty} K_i \times 10^i \quad (1.3.1)$$

式中 K_i 为基数“10”的第 i 次幂的系数。

从计数电路的角度来看,采用十进制是不方便的。因为构成计数电路的基本思路是把电路的状态与数码对应起来,而十进制的十个数码,必须由十个不同的而且能严格区分的电路状态与之对应,这样将在技术上带来许多困难,而且也不经济,因此在计数电路中一般不直接采用十进制。

① 下标 D(Decimal) 表示十进制。

1.3.2 二进制

1. 二进制的定义

二进制数与十进制数的区别在于数码的个数和进位的规律不同,十进制数用十个数码,并且“逢十进一”;而二进制数是用两个数码 0 和 1,并且“逢二进一”,即 $1 + 1 = 10$ (读为“壹零”)。必须注意,这里的“10”与十进制数的“10”是完全不同的,它并不代表“拾”。右边的“0”表示 2^0 位数,左边的“1”表示 2^1 位数,也就是 $10 = 1 \times 2^1 + 0 \times 2^0$ 。因此,所谓二进制就是以 2 为基数的计数体制。

当二进制数的数位较多时,可按 2 的乘幂依次表示,如图 1.3.2 所示。

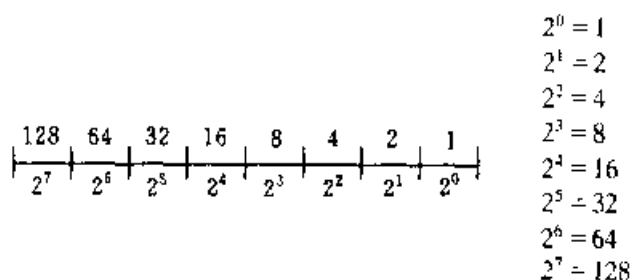


图 1.3.2 二进制数的位权图

2. 二进制数的表达式

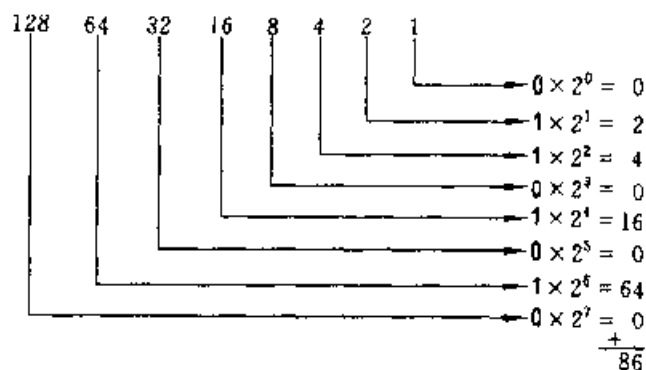
一般地说,二进制数可表示为

$$(N)_B^{\text{①}} = \sum_{i=-\infty}^{\infty} K_i \times 2^i \quad (1.3.2)$$

式中 K_i 为基数“2”的第 i 次幂的系数。这样,可将任意一个二进制数转换为十进制数。

例 1.3.2 试将二进制数 $(01010110)_B$ 转换为十进制数。

解: 将每一位二进制数乘以位权,然后相加便得相应的十进制数。



3. 二进制数的波形表示

在数字电子技术和计算机应用中,二值数据常用数字波形来表示。这样,数

① B 系 Binary 的字头

据比较直观,也便于使用电子示波器进行监视。图 1.3.3 表示一计数器的波形,图中最左列标出了二进制数的位权(2^0 、 2^1 、 2^2 、 2^3)以及最低位(LSB)^①和最高位(MSB)^②,从左至右画出了每位数的波形,最后一行标出了从 0 到 15 的等效十进制数。

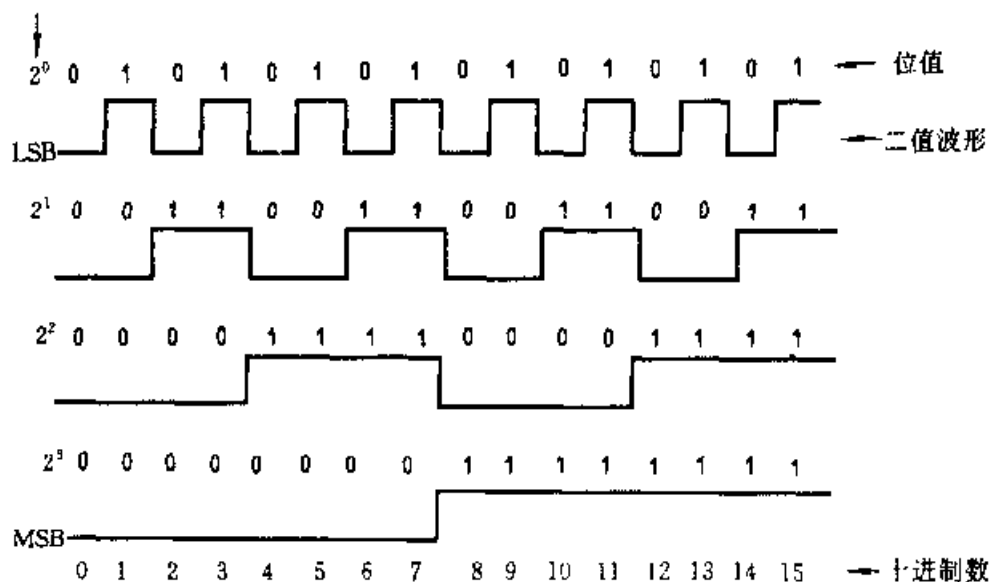


图 1.3.3 用二进制数表示 0~15 波形图

从图 1.3.3 还可看出,每 1 位的波形均为对称方波,其占空比均为 50%,但每一波形的频率逐位减半直至最高位。

4. 二进制数据的传输

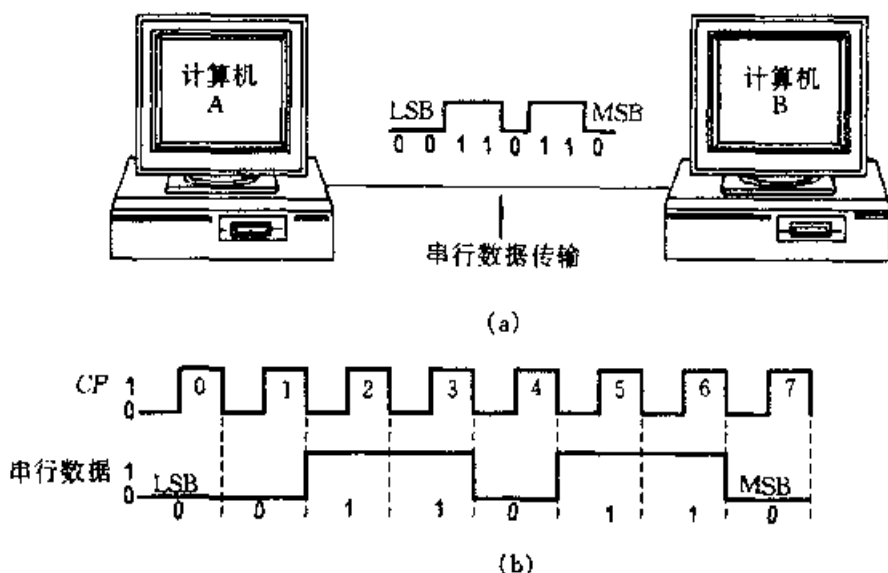
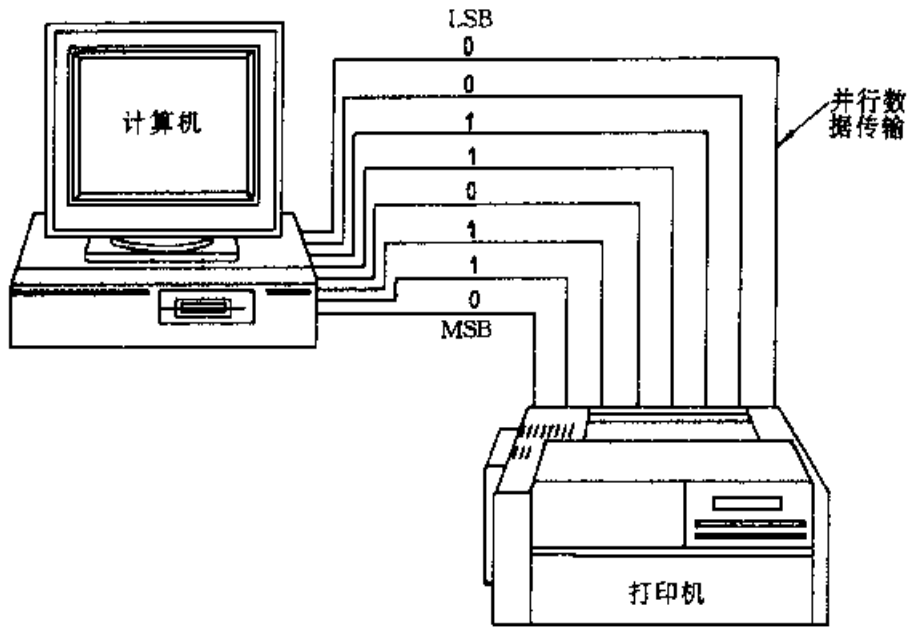


图 1.3.4 二进制数据的串行传输

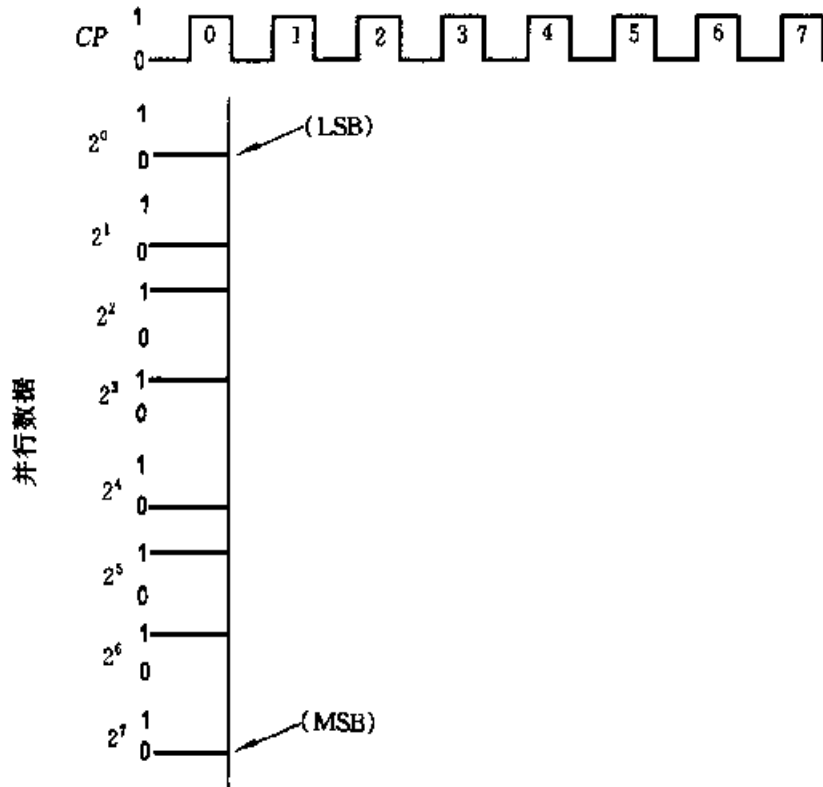
(a) 两台计算机之间的串行通信 (b) 二进制数据的串行表示

① LSB 系 Least Significant Bit 的缩写。

② MSB 系 Most Significant Bit 的缩写。



(a)



(b)

图 1.3.5 并行传输数据的示意图

(a) 计算机与打印机之间的并行通信 (b) 二进制数据的并行表示

二进制数据从一处传输到另一处,可以采用串行的方式或并行的方式。对于串行的方式,所需的设备简单,只需一根导线和一共同接地端即可,每传送 1 位数据需要一个时钟周期。这种传输可以通过电话线或在两台计算机之间进行。

二进制数据作串行传输的示意图如图 1.3.4 所示,图 1.3.4a 表示二进制数据 00110110 在两台计算机之间串行传输。在时钟脉冲 CP 的控制下,数据由最高位 MSB 到最低位 LSB 依次传输(见图 1.3.4b)。注意,每位数据的传输时间即时钟脉冲的周期,而且在时钟脉冲的下降沿完成 1 位数据的传输。

若要求传输速度快,则可采用并行传输的方式。图 1.3.5 表示并行传输数据的示意图。图 1.3.5a 表示一台打印机从一台计算机以并行的方式取用数据,意即将 8 位数据同时传输,所需的时间为一个时钟脉冲的周期,只有串行传输时间的八分之一。但是在设备上的花费要多些,需用 8 条传输线和其他部件。并行传输在数字系统中是一种常用的技术。

由于二进制具有一定的优点,因此它在计算技术中被广泛采用。

(1) 二进制的数字装置简单可靠,所用元件少;二进制只有两个数码 0 和 1,因此,它的每 1 位数都用任何具有两个不同稳定状态的元件来表示,如 BJT 的饱和与截止,继电器接点的闭合和断开,灯泡的亮和不亮等。只要规定其中一种状态表示 1,另一种状态表示 0,就可以表示二进制数。这样,数码的存储、分析和传输,就可以用简单而可靠的方式进行。

(2) 二进制的基本运算规则简单,运算操作方便。

但是,采用二进制也有一些缺点。用二进制表示一个数时,位数多,例如,十进制数 49 表示为二进制时,即为 110001,使用起来不方便也不习惯。因此,在运算时,原始数据多用人们习惯的十进制数,在送入机器时,就必须将十进制原始数据转换成数字系统能接受的二进制数。而在运算结束后,再将二进制数转换为十进制数,表示最终结果。

1.3.3 十-二进制之间的转换

既然同一个数可以用二进制和十进制两种不同形式来表示,那么两者之间就必然有一定的转换关系。对于整数可写成

$$(N)_D = b_n \times 2^n + b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0 \quad (1.3.3)$$

式中 $b_n, b_{n-1}, \cdots, b_1, b_0$ 是二进制数各位数字。将等式两边分别除以 2,得

$$\frac{1}{2}(N)_D = b_n \times 2^{n-1} + b_{n-1} \times 2^{n-2} + \cdots + b_1 \times 2^0 + \frac{b_0}{2} \quad (1.3.4a)$$

由此可知,将十进制数除以 2,其余数为 b_0 。将式(1.3.4a)的商再除以 2,得

$$\frac{1}{2^2}(N)_D = b_n \times 2^{n-2} + b_{n-1} \times 2^{n-3} + \cdots + b_2 \times 2^0 + \frac{b_1}{2} \quad (1.3.4b)$$

其余数为 b_1 。不难推知,将十进制整数每除以一次 2,就可根据余数得到二进制数的 1 位数字。因此,只要连续除以 2 直到商为 0,就可由所有的余数求出二进制数。

例 1.3.3 将十进制数 $(25)_{10}$ 转换为二进制数。

解：根据上述原理，可将 $(25)_{10}$ 按如下的步骤转换为二进制数

$$\begin{array}{r}
 2 \overline{) 25} \cdots \cdots \text{余 } 1 \quad b_0 \\
 \underline{20} \\
 2 \overline{) 12} \cdots \cdots \text{余 } 0 \quad b_1 \\
 \underline{12} \\
 2 \overline{) 6} \cdots \cdots \text{余 } 0 \quad b_2 \\
 \underline{6} \\
 2 \overline{) 3} \cdots \cdots \text{余 } 1 \quad b_3 \\
 \underline{2} \\
 2 \overline{) 1} \cdots \cdots \text{余 } 1 \quad b_4 \\
 \underline{0}
 \end{array}$$

由上得 $(25)_{10} = (11001)_2$ 。

当十进制数较大时，不必逐次除 2，而可利用图 1.3.2 所示的 2 乘幂项与十进制数进行对比，使转换过程得到简化。

例 1.3.4 将 $(133)_{10}$ 转换为二进制数。

解：参看图 1.3.2，对于 133 而言，其最大的乘幂项为 2^7 ($2^7 = 128$)，余下的 5 ($= 133 - 128$) 可等效为 $2^2 + 2^0$ 之和，因而其转换过程可描述如下：

$$\begin{array}{r}
 \begin{array}{r}
 133 \\
 - 128 \\
 \hline
 5
 \end{array}
 \quad \xrightarrow{2^7} \quad
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
 \hline
 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
 \hline
 \end{array} \\
 \begin{array}{r}
 - 4 \\
 \hline
 1
 \end{array}
 \quad \begin{array}{r}
 - 1 \\
 \hline
 0
 \end{array}
 \end{array}$$

注意在转换过程中，首先在 $(133)_{10}$ 中找出含有的 2 的乘幂项，在相应位权上填以 1，余下的空位填以 0，故得 $(133)_{10} = (10000101)_2$ 。

值得指出，多数计算机或数字系统中只处理 4、8、16 或 32 位的二进制数据，因此，数据的位数需配成规格化的位数，如例 1.3.3 中转换结果为 11001，如将它配成 8 位，则相应的高幂项应填以 0，其值不变，即

$$11001 = 00011001$$

对于小数，当它转换为二进制数时，将式(1.3.2)写成

$$(N)_{10} = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-(n-1)} \times 2^{-(n-1)} + b_{-n} \times 2^{-n} \quad (1.3.5)$$

将上式两边分别乘以 2，得

$$2 \times (N)_{10} = b_{-1} \times 2^0 + b_{-2} \times 2^{-1} + \cdots + b_{-(n-1)} \times 2^{-(n-2)} + b_{-n} \times 2^{-(n-1)} \quad (1.3.6)$$

由此可见,将十进制小数乘以2,取其个位数为 b_{-1} 。不难推知,将十进制小数每次除去上次所得积中之个位数连续乘以2,直到满足误差要求进行“四舍五入”为止,就可完成由十进制小数转换成二进制小数。

例 1.3.5 将 $(0.706)_{10}$ 转换为二进制数,要求其误差不大于 2^{-10} 。

解: 按式(1.3.5)所表达的方法,可得 $b_{-1}, b_{-2}, \dots, b_{-10}$:

$$0.706 \times 2 = 1.412 \cdots \cdots 1 \cdots \cdots b_{-1}$$

$$0.412 \times 2 = 0.824 \cdots \cdots 0 \cdots \cdots b_{-2}$$

$$0.824 \times 2 = 1.648 \cdots \cdots 1 \cdots \cdots b_{-3}$$

$$0.648 \times 2 = 1.296 \cdots \cdots 1 \cdots \cdots b_{-4}$$

$$0.296 \times 2 = 0.592 \cdots \cdots 0 \cdots \cdots b_{-5}$$

$$0.592 \times 2 = 1.184 \cdots \cdots 1 \cdots \cdots b_{-6}$$

$$0.184 \times 2 = 0.368 \cdots \cdots 0 \cdots \cdots b_{-7}$$

$$0.368 \times 2 = 0.736 \cdots \cdots 0 \cdots \cdots b_{-8}$$

$$0.736 \times 2 = 1.472 \cdots \cdots 1 \cdots \cdots b_{-9}$$

由于最后的小数小于0.5,根据“四舍五入”的原则, b_{-10} 应为0。所以, $(0.706)_{10} = (0.101101001)_{2}$,其误差 $\epsilon < 2^{-10}$ 。

1.3.4 十六进制和八进制

由于使用二进制数经常是位数很多,不便书写和记忆,因此在数字计算机的资料中常采用十六进制数或八进制数来表示二进制数。上述十进制和二进制的表示法可以推广到十六进制和八进制。

十六进制数采用十六个数码,而且“逢十六进一”。这种数制中有十六个不同的数字:0,1,2,3,4,5,6,7,8,9,A(对应于十进制数中的10),B(11),C(12),D(13),E(14),F(15)。它是以十六为基数的计数体制。

仿照式(1.3.1),十六进制数可表达如下:

$$(N)_{16} = \sum_{i=0}^n K_i \times 16^i$$

式中 K_i 为基数16的第 i 次幂的系数。

例 1.3.6 将十六进制数4E6转换为十进制数

解: $(4E6)_{16} = 4 \times 16^2 + 14 \times 16^1 + 6 \times 16^0 = 1254$

例 1.3.7 将二进制数1001 1100 1011 0100 1000转换为十六进制数。

解: 将二进制数中的每4位与十六进制数对应即得十六进制数:

① 下标H(Hexadecimal)表示十六进制。(59)₁₆又可写成59H。

$$(1001\ 1100\ 1011\ 0100\ 1000)_B = (9CB48)_H$$

例 1.3.8 将十六进制数 $(F156)_H$ 转换为二进制数。

解： 将每位十六进制数用 4 位二进制数代替即得相应的二进制数。

$$(F156)_H = (1111\ 0001\ 0101\ 0110)_B$$

至于十进制数变换为十六进制数,可先将十进制数变换为二进制数,再变换为十六进制数。

同理,对于八进制数,可将 3 位二进制数分为一组,对应于 1 位八进制数。如将例 1.3.7 中的二进制数变换为八进制数,则可写成

$$(10\ 011\ 100\ 101\ 101\ 001\ 000)_B = (2345510)_O^{\text{①}}$$

为便于对照,将十进制、二进制、八进制及十六进制之间的关系列于表 1.3.1 中。

表 1.3.1 几种数制之间的关系对照表

十进制数	二进制数	八进制数	十六进制数
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

① 下标 O(Octal) 表示八进制。为了更清楚起见,某些文献中也用下标 2、8、10 及 16 分别表示二进制、八进制、十进制及十六进制。

复习思考题

- 1.3.1 为什么在计算机或数字系统中通常采用二进制数？
 1.3.2 在二进制数中，其位权的规律如何？
 1.3.3 八进制和十六进制数各用于何种场合？

1.4 二进制码

数字系统中的信息可分为两类，一类是数值，另一类是文字符号（包括控制符）。数值信息的表示方法已如前述。为了表示文字符号信息，往往也采用一定位数的二进制数码表示，这个特定的二进制码称为代码。建立这种代码与十进制数值、字母、符号的一一对应的关系称为编码。

若所需编码的信息有 N 项，则需用的二进制数码的位数 n 应满足如下关系：

$$2^n \geq N$$

下面介绍几种常见的码。

二-十进制码(BCD码^①)在这种编码中，用4位二进制数 $b_3b_2b_1b_0$ 来表示十进制数中的0~9十个数码。若这十个数码与自然二进制数一一对应，则编码关系如表1.4.1中左边一列的0000~1001所示，二进制数码每位的值称为位权， b_0 位的权为 $2^0=1$ ， b_1 位的权为 $2^1=2$ ， b_2 位的权为 $2^2=4$ ， b_3 位的权为 $2^3=8$ 。位权的概念在1.3.1和1.3.2节中已介绍过。例如，二进制码0111所表示的十进制数为 $8 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 7$ ，因此这种BCD码称为8421BCD码，它是一种最基本的BCD码，应用也较普遍。在一般情况下，十进制数与二进制码之间可用下式来表示：

$$(N)_D = W_3b_3 + W_2b_2 + W_1b_1 + W_0b_0 \quad (1.4.1)$$

式中 $W_3 \sim W_0$ 为二进制码中各位的权。

8421BCD码由4位二进制数的0000(0)到1111(15)16种组合中的前10种组成，即0000(0)~1001(9)，其余6种组合是无效的。16种组合中选取10种有效组合方式的不同，可以得到其他二-十进制码，如表1.4.1中的2421码等。余3码是由8421码加3(0011)得来的，不能用式(1.4.1)来表示其编码关系，因而它是一种无权码。

^① BCD码是 Binary-Coded-Decimal 的缩写，即二进制编码的十进制码。

表 1.4.1 几种常见的码

$b_3b_2b_1b_0$ $2^32^22^12^0$	代码对应的十进制数			
	自然二进制码	二—十进制数		
		8421 码	2421 码	余 3 码
0000	0	0	0	
0001	1	1	1	
0010	2	2	2	
0011	3	3	3	0
0100	4	4	4	1
0101	5	5		2
0110	6	6		3
0111	7	7		4
1000	8	8		5
1001	9	9		6
1010	10			7
1011	11		5	8
1100	12		6	9
1101	13		7	
1110	14		8	
1111	15		9	

* 空位表示无效的组舍。

实用上,还有一种常见的无权码叫格雷码^①,其编码如表 1.4.2 所示。这种

表 1.4.2 格 雷 码

b_3	b_2	b_1	b_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

① 格雷系 Gray 的译称。

码的特点是:相邻的两个码组之间仅有一位不同,因而常用于模拟量的转换中,当模拟量发生微小变化而可能引起数字量发生变化时,格雷码仅改变1位,这样与其他码同时改变两位或多位的情况相比更为可靠,即可减少出错的可能性。

其他有关编码的讨论,可参阅文献[8]。

通常,人们可通过键盘上的字母、符号和数值向计算机发送数据和指令,每一个键符可用二进制码来表示,ASCII码^①即是其中的一种,它是用7位二进制码表示的,其编码表见附录A。关于二进制数的算术运算,可参阅附录B。

复习思考题

- 1.4.1 8421BCD码为什么用得较普遍?
- 1.4.2 试检验2421码的编码表。
- 1.4.3 格雷码有什么特点,用于什么场合?

1.5 基本逻辑运算

前已讨论,利用二值数字逻辑中的1(逻辑壹)和0(逻辑零)不仅可以表示二进制数,还可表示许多对立的逻辑状态。在分析和设计数字电路时,所使用的数学工具是逻辑代数(又称布尔代数^②)。逻辑代数是按一定的逻辑规律进行运算的代数,虽然它和普通代数一样也是用字母表示变量,但两种代数中变量的含义是完全不同的,它们之间有着本质的区别,逻辑代数中的变量(逻辑变量)只有两个值,即0和1,而没有中间值。0和1并不表示数量的大小,如前所述而是表示对立的逻辑状态。

在逻辑代数中,有与、或、非3种基本逻辑运算。众所周知,运算是一种函数关系,它可以用语句描述,亦可用逻辑表达式描述,还可用表格或图形来描述。描述逻辑关系的表格为真值表。用规定的图形符号来表示逻辑运算称为逻辑符号。下面分别讨论3种基本的逻辑运算。

1. 与运算

图1.5.1a表示一个简单的与逻辑电路,电压 V 通过开关 A 和 B 向灯泡供电,只有 A 与 B 同时接通时,灯泡才亮。 A 和 B 中只要有一个不接通或二者均不接通时,则灯泡不亮,其真值表如图1.5.1b所示。因此,从这个电路可总结出这样的逻辑关系:“只有当一件事(灯亮)的几个条件(开关 A 与 B 都接通)全部具备之后,这件事(灯亮)才发生”,这种关系称为与逻辑。如果用二值逻辑0和

① ASCII系 American Standard Code for Information Interchange(美国标准信息交换码)的缩写。

② 布尔代数是英国数学家 G. Boole(1815—1864)提出的。

1 来表示,并设开关不通和灯不亮均用 0 表示,而开关接通和灯亮均用 1 表示,则得图 1.5.1c,其中 L 表示灯的状态。若用逻辑表达式来描述,则可写为

$$L = A \cdot B \quad (1.5.1)$$

式中小圆点“ \cdot ”表示 A 、 B 的与运算,也表示逻辑乘。在不致引起混淆的前提下,乘号“ \cdot ”被省略。在某些文献中,也有用符号 \wedge 、 \cap 表示与运算的。用与逻辑门电路实现与运算,其逻辑符号如图 1.5.1d 所示。

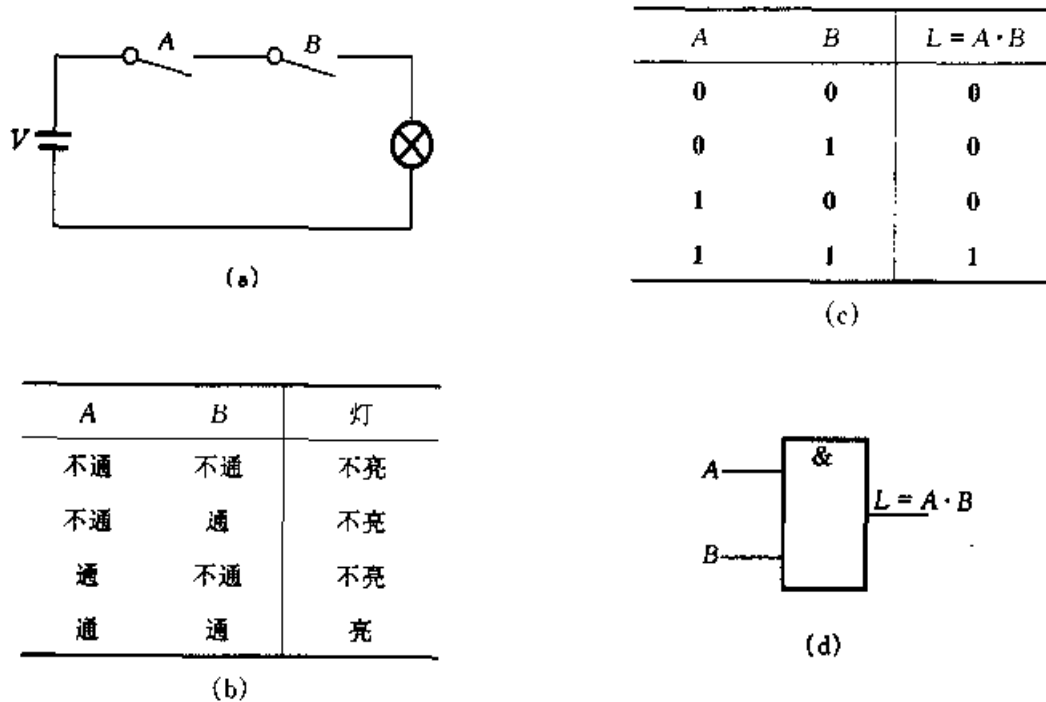


图 1.5.1 与逻辑运算

(a) 电路图 (b) 真值表 (c) 用 0、1 表示的真值表 (d) 与逻辑门电路的符号

2. 或运算

图 1.5.2a 表示一简单的或逻辑电路,电压 V 通过开关 A 或 B 向灯泡供电。只要开关 A 或 B 接通或二者均接通,则灯亮;而当 A 和 B 均不通时,则灯不亮,其真值表如图 1.5.2b 所示。由此可总结出另一种逻辑关系:“当一件事情(灯亮)的几个条件(开关 A 、 B 接通)中只要有一个条件得到满足,这件事(灯亮)就会发生”。这种关系称为或逻辑。或是指 A 接通或 B 接通,即任一个条件具备的意思。仿照前述,用 0、1 表示的或逻辑真值表如图 1.5.2c 所示,若用逻辑表达式来描述,则可写为

$$L = A + B \quad (1.5.2)$$

式中符号“ $+$ ”表示 A 、 B 或运算,也表示逻辑加。在某些文献中,也有用符号 \vee 、 \cup 来表示或运算的,或运算用或逻辑门电路实现,其逻辑符号如图 1.5.2d 所示。

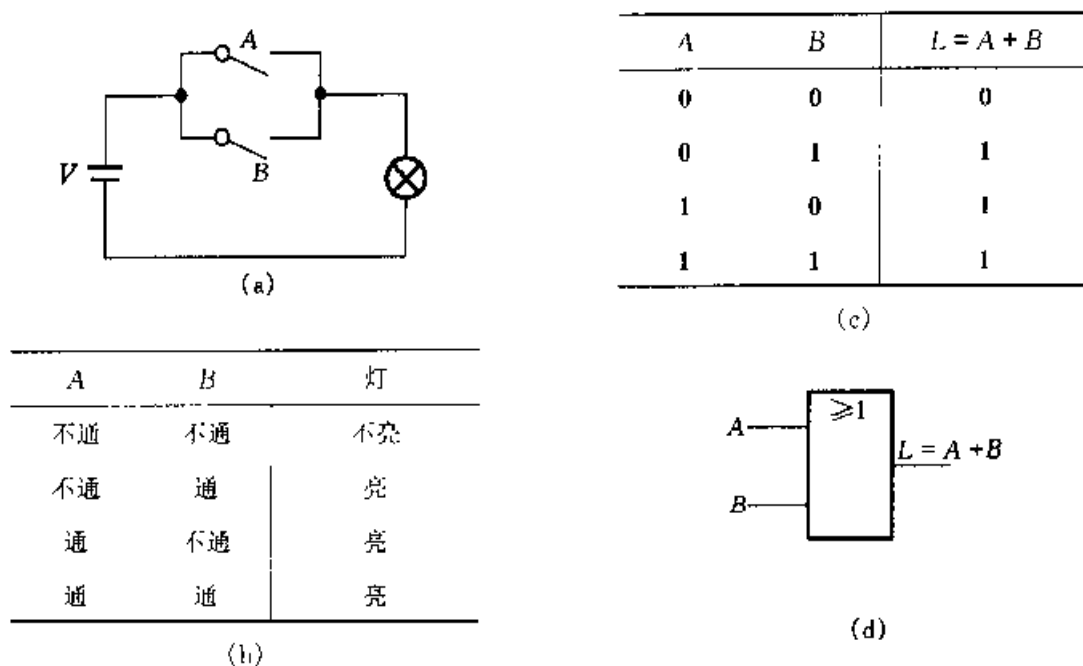


图 1.5.2 或逻辑运算

(a) 电路图 (b) 真值表 (c) 用 0、1 表示的真值表 (d) 或逻辑门电路的符号

3. 非运算

如图 1.5.3a 所示,电压 V 通过一继电器触点向灯泡供电,NC 为继电器 A 的动断(常闭)触点,当 A 不通电时,灯亮;而当 A 通电时,灯不亮。其真值表如图 1.5.3b 所示。由此可总结出第三种逻辑关系,即“一件事情(灯亮)的发生是 以其相反的条件为依据”。这种逻辑关系称为非逻辑。若用 0 和 1 来表示继电器和灯泡的状态,则可得图 1.5.3c。在此图中,读者很容易理解, A 不通电和灯不亮是定义为 0 态,而 A 通电和灯亮是定义为 1 态的。显然 L 与 A 总是处于对立的逻辑状态,若用逻辑表达式来描述,则可写为

$$L = \bar{A} \tag{1.5.3}$$

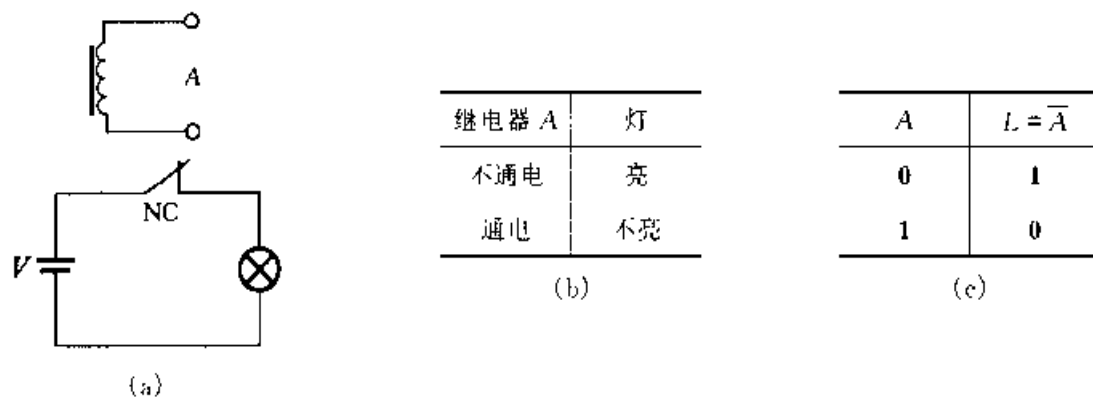


图 1.5.3 非逻辑运算

(a) 电路图 (b) 真值表 (c) 用 0、1 表示的真值表

式中,字母 A 上方的短划“-”表示非运算。在某些文献中,也有用“-”、“ \neg ”或

“.”来表示非运算的。用非逻辑门电路实现非运算,其逻辑符号如图 1.5.4a、b 所示。

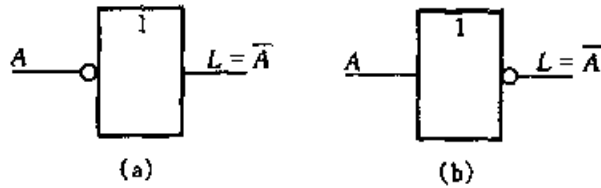


图 1.5.4 非逻辑门电路的符号

(a) 在输入端用小圆圈表示非运算 (b) 在输出端用小圆圈表示非运算

关于逻辑门电路的细节将在第 2 章中讨论。

上述与、或逻辑运算可以推广到多变量的情况:

$$L = A \cdot B \cdot C \cdot \dots \quad (1.5.4)$$

$$L = A + B + C + \dots \quad (1.5.5)$$

其他逻辑函数都可用上述三种基本函数组合而成。表 1.5.1 列出了几种基本的逻辑运算函数式及其相应的逻辑门电路的代表符号,以便于比较和应用。

表 1.5.1 几种常用的逻辑运算

逻辑运算 逻辑门符号		与运算	或运算	非运算	与非运算	或非运算	异或运算
		$L = A \cdot B$	$L = A + B$	$L = \bar{A}$	$L = \overline{A \cdot B}$	$L = \overline{A + B}$	$L = A\bar{B} + \bar{A}B$
逻辑变量 A	B						
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

* 逻辑门将在第 2 章详细讨论。

复习思考题

1.5.1 什么叫与、或、非逻辑? 试列举几种相关的实例,并列写出 3 种逻辑运算的表达式。

1.6 逻辑函数与逻辑问题的描述

前节讨论了与、或、非 3 种基本逻辑运算,即 3 种基本的逻辑函数。本节从工程实际出发,提出逻辑命题,然后用真值表加以描述,从真值表可直接写出逻

辑函数。

一般来说,一个比较复杂的逻辑电路,往往是受多种因素控制的,就是说有多个逻辑变量,输出量与输入变量之间的关系可用逻辑函数来描述。下面看一个简单的实例。

图 1.6.1 是一个控制楼梯照明灯的电路,单刀双掷开关 A 装在楼下, B 装在楼上,这样在楼下开灯后,可在楼上关灯;同样,也可在楼上开灯,而在楼下关灯。因为只有当两个开关都向上扳或向下扳时,灯才亮;而一个向上扳、另一个向下扳时,灯就不亮。

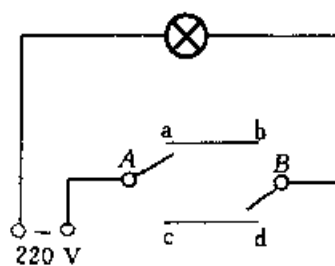


图 1.6.1 逻辑电路举例

上述电路的逻辑关系可用逻辑函数来描述,设 L 表示灯的状态,即 $L=1$ 表示灯亮, $L=0$ 表示灯不亮。用 A 和 B 表示开关 A 和开关 B 位置,用 1 表示开关向上扳,用 0 表示开关向下扳。则 L 与 A 、 B 的关系可用真值表 1.6.1 来表示。

表 1.6.1 图 1.6.1 的真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

由真值表可知,在 A 、 B 状态的 4 种不同组合中,只有第一 ($A=B=0$) 和第四 ($A=B=1$) 两种组合才能使灯亮 ($L=1$)。 A 、 B 之间是与的关系,而两种状态组合之间则是或的关系。不论变量 A 、 B 或输出 L ,凡取 1 值的用原变量表示,取 0 值用反变量表示。故可写出灯亮的逻辑函数

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

它描述了只有开关 A 、 B 都扳上或扳下时灯才亮。这就是从逻辑问题建立逻辑函数的过程。关于更复杂的逻辑问题的描述,将在后续各章中讨论。

小 结

- 由于模拟信息具有连续性,实用上难于存储、分析和传输;应用二值数字逻辑构成的数字电路或数字系统较易克服这些困难。其实质是利用数字 1 和 0 来表示信息。

- 数字系统中常用二进制数来表示数据。所谓二进制是以 2 为基数的计数体制,其中数字 1 和 0 分别表示对立的两个逻辑状态。在集成电路中容易实现。一个 n 位的二进制数可以表示 2^n 个数据。

• 十六进制是二进制的简写,它是以 16 为基数的计数体制,常用于数字电子技术、微处理器、计算机和数据通信中。任意一种格式的数可以在十六进制、二进制和十进制之间相互转换。

• 特殊二进制码常用来表示十进制数,如 8421 码、2421 码、余 3 码、格雷码等。也有用 7 位二进制数来表示符号-数字混合码,如 ASCII 码。

• 数字逻辑是计算机的基础。它不仅可以实现复杂的算术运算,而且可以实现复杂的逻辑运算。逻辑运算中的三种基本运算是与、或、非运算。分析数字电路或数字系统的数学工具是逻辑代数。

习 题

1.1.1 一数字信号的波形如图题 1.1.1 所示,试问该波形所代表的二进制数是什么?

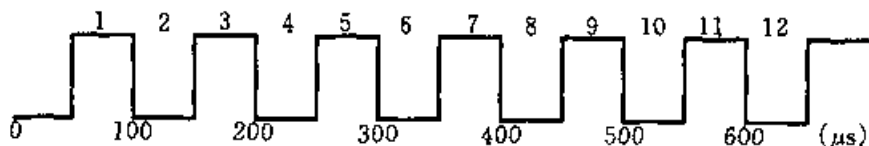


图题 1.1.1

1.1.2 试绘出下列二进制数的数字波形,设逻辑 1 的电压 = 5 V,逻辑 0 的电压 = 0 V:

(1) 001100110011 (2) 0111010 (3) 1111011101

1.1.3 (1) 若某正逻辑波形如图题 1.1.3 所示,试写出相应的逻辑值 1 和 0(与标号 1~12 对应)。



图题 1.1.3

1.1.4 试就下列的逻辑值给出相应的数字波形,设高电平(1)电压为 5 V,低电平(0)电压为 0 V:

(1) 001100110011 (2) 0111010 (3) 1111011101

1.2.1 试按表 1.2.1 所列的数字集成电路的分类依据,指出下列器件属于何种集成度器件:(1) 微处理器;(2) IC 计算器;(3) IC 加法器;(4) 逻辑门;(5) 4 兆位存储器 IC。

1.3.1 将下列十进制数转换为二进制数、八进制数、十六进制数和 8421BCD 码(要求转换误差不大于 2^{-4}):

(1) 43 (2) 127 (3) 254.25 (4) 2.718

1.3.2 将下列数码作为自然二进制数或 8421BCD 码时,分别求出相应的十进制数:

(1) 10010111 (2) 100010010011 (3) 000101001001

1.3.3 将下列每一二进制数转换为十六进制码：

(1) $(101001)_2$ (2) $(11.01101)_2$

1.3.4 将下列十进制数转换为十六进制数：

(1) $(500)_{10}$ (2) $(59)_{10}$ (3) $(.34)_{10}$ (4) $(1002.45)_{10}$

1.3.5 将下列十六制数转换为二进制数：

(1) $(23F.45)_{16}$ (2) $(A040.51)_{16}$

1.3.6 将下列十六进制数转换为十进制数：

(1) $(103.2)_{16}$ (2) $(A45D.0BC)_{16}$

2 逻辑门电路

引言 前章介绍了模拟信号与数字信号、数字电路及其分析方法、数制和码的基本知识。从与、或、非三种基本逻辑运算引出了逻辑变量与逻辑函数的关系。在那里,逻辑符号是以黑匣的方式来表示相应的逻辑门,如与、或、非等基本逻辑门。但是,黑匣法只能建立初步的概念,对于电子设计工作者来说是不够的。为了正确而有效地使用集成逻辑门电路,用户必须对组件内部电路特别是对它的外部特性有所了解。因此,本章将揭开黑匣的奥秘,讲述几种通用的集成逻辑门电路,如BJT-BJT^①逻辑门电路(TTL^②)、射极耦合逻辑门电路(ECL^③)和金属-氧化物-半导体互补对称逻辑门电路(CMOS^④)。

为了掌握上述各种电路的逻辑功能和特性,首先必须熟悉开关器件的开关特性,这是门电路的工作基础。在分析门电路时,着重它们的逻辑功能和外特性,对其内部电路,只作一般介绍。

2.1 二极管的开关特性

用来接通或断开电路的开关器件应具有两种工作状态:一种是接通(要求其阻抗很小,相当于短路),另一种是断开(要求其阻抗很大,相当于开路)。

在数字电路中,二极管和BJT大多数工作在开关状态。它们在脉冲信号的作用下,时而导通,时而截止,相当于开关的“开通”和“关断”。研究它们的开关特性,就是具体分析导通和截止之间的转化问题。当脉冲信号频率很高时,开关状态变化的速度非常快,可达每秒百万次数量级,这就要求器件的导通与截止两种状态的转换要在微秒甚至纳秒数量级的时间内完成。

① BJT是Bipolar Junction Transistor的缩写。

② TTL是Transistor-Transistor Logic的缩写。

③ ECL是Emitter-Coupled Logic的缩写。

④ CMOS是Complementary Metal-Oxide-Semiconductor的缩写。

二极管的开关特性表现在正向导通与反向截止这样两种不同状态之间的转换过程。二极管从反向截止到正向导通与从正向导通到反向截止相比所需的时间很短,一般可以忽略不计,因此下面着重讨论二极管从正向导通到反向截止的转换过程。

1. 二极管从正向导通到截止有一个反向恢复过程

先分析二极管从导通到截止的物理现象。在图 2.1.1a 所示的硅二极管电路中,加入一个如图 2.1.1b 所示的输入电压。在 $0 \sim t_1$ 时间内,输入为 $+V_F$, 二极管导通,电路中有电流流通。设 V_D 为二极管正向压降(硅管为 0.7 V 左右),当 V_F 远大于 V_D 时, V_D 可略去不计,则

$$I_F = \frac{V_F - V_D}{R_L} \approx \frac{V_F}{R_L}$$

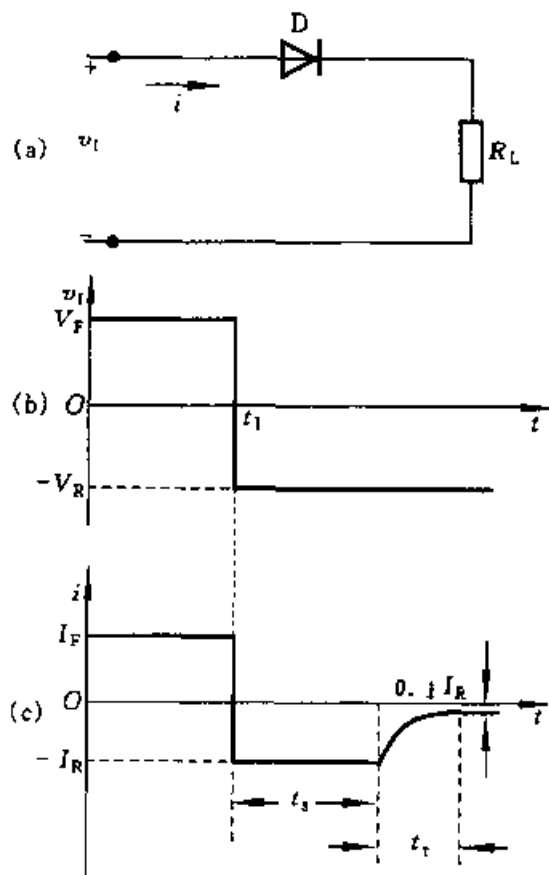


图 2.1.1 二极管的开关特性

(a) 电路 (b) 输入电压 (c) 二极管电流

在 t_1 时, v_i 突然从 $+V_F$ 变为 $-V_R$ 。在理想情况下,二极管将立刻转为截止,电路中应只有很小的反向电流。但实际情况是,二极管并不立刻截止,而是先由正向的 I_F 变到一个很大的反向电流 $I_R = V_R/R_L$,这个电流维持一段时间 t_s 后才开始逐渐下降,再经过 t_r 后,下降到一个很小的数值 $0.1 I_R$,这时二极管

才进入反向截止状态,如图 2.1.1c 所示。

通常把二极管从正向导通转为反向截止所经过的转换过程称为反向恢复过程。其中 t_s 称为存储时间, t_t 称为渡越时间, $t_{re} = t_s + t_t$ 称为反向恢复时间。

由于反向恢复时间的存在,使二极管的开关速度受到限制。

2. 产生反向恢复过程的原因——电荷存储效应

产生上述现象的原因是由于二极管外加正向电压 V_F 时,载流子不断扩散而存储的结果。当外加正向电压时,P 区空穴向 N 区扩散,N 区电子向 P 区扩散,这样,不仅使势垒区(耗尽区)变窄,而且使载流子有相当数量的存储,在 P 区内存储了电子,而在 N 区内存储了空穴,它们都是非平衡少数载流子,如图 2.1.2a 所示。例如,空穴由 P 区扩散到 N 区后,并不是立即与 N 区中的电子复合而消失,而是在一定的路程 L_p 内(这段路程 L_p 通常称为扩散长度),一方面继续扩散,一方面与电子复合消失,这样就会在 L_p 范围内存储一定数量的空穴,并建立起一定空穴浓度分布,靠近结边缘的浓度最大,离结越远,浓度越小。正向电流越大,存储的空穴数目越多,浓度分布的梯度也越大。电子扩散到 P 区的情况也类似,图 2.1.2b 表示二极管中存储电荷的分布。

我们把正向导通时,非平衡少数载流子积累的现象叫做电荷存储效应。

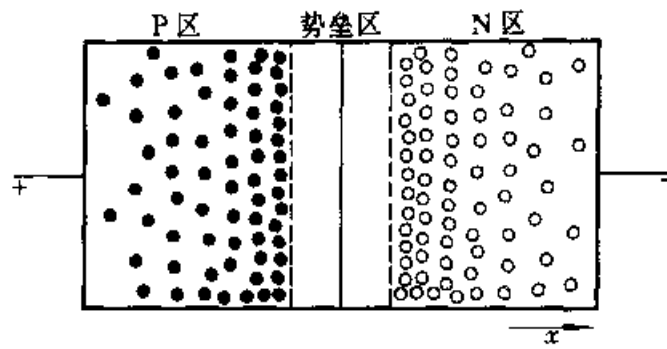
当输入电压突然由 $+V_F$ 变为 $-V_R$ 时,P 区存储的电子和 N 区存储的空穴不会马上消失,但它们将通过下列两个途径逐渐减少:①在反向电场作用下,P 区电子被拉回 N 区,N 区空穴被拉回 P 区,形成反向漂移电流 I_R ,如图 2.1.3 所示;②与多数载流子复合。

在这些存储电荷消失之前,PN 结仍处于正向偏置,即势垒区仍然很窄,PN 结的电阻仍很小,与 R_L 相比可以忽略,所以此时反向电流 $I_R = (V_R + V_D)/R_L$ 。 V_D 表示 PN 结两端的正向压降,一般 $V_R \gg V_D$,即 $I_R \approx V_R/R_L$ 。在这段期间, I_R 基本上保持不变,主要由 V_R 和 R_L 所决定。

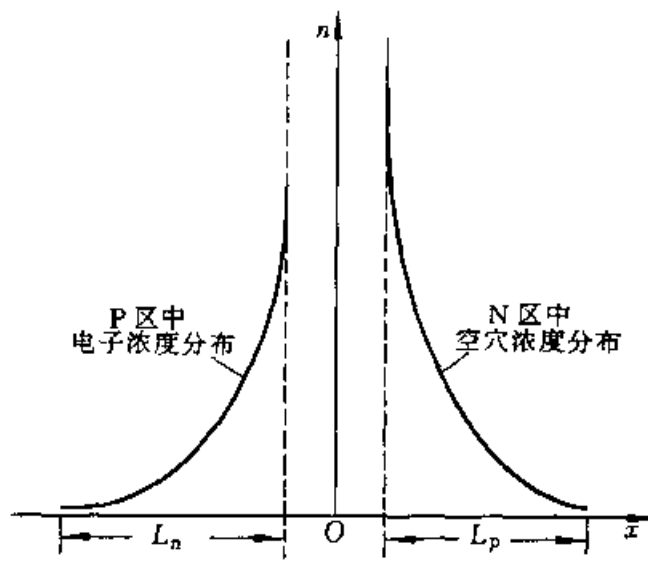
经过时间 t_s 后,P 区和 N 区所存储的电荷已显著减小,势垒区逐渐变宽,反向电流 I_R 逐渐减小到正常反向饱和电流的数值,经过时间 t_t ,二极管转为截止。

由上可知,二极管在开关转换过程中出现的反向恢复过程,实质上是由于电荷存储效应所引起的,反向恢复时间就是存储电荷消失所需要的时间^①。

① 二极管的正向导通和反向截止过程所经历的时间也可以借助结电容的充放电过程来解释。反向恢复时间就是结电容放电所需的时间。实际上,由于正向充电电流远大于反向放电电流,因此,反向恢复时间远大于正向导通时间。



(a)



(b)

图 2.1.2 二极管中存储电荷的分布

(a) 示意图 (b) 浓度分布曲线

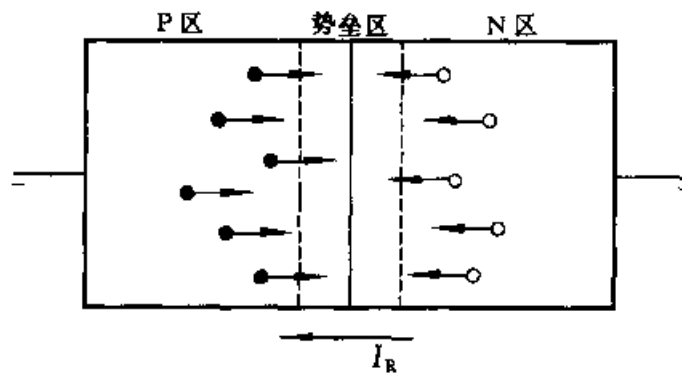


图 2.1.3 存储电荷形成 I_R

手册中给出了各种二极管在一定条件下测出的反向恢复时间,一般开关管的反向恢复时间在纳秒(ns)数量级。

3. 二极管的开通时间

二极管从截止转为正向导通所需的时间称为开通时间,这个时间同反向恢复时间相比是很短的。这是由于PN结在正向偏压作用下,势垒区迅速变窄,有利于少数载流子的扩散,正向电阻很小,因而它在导通过程中及导通以后,其正向压降都很小,比输入电压 V_F 小得多,故电路中的正向电流 $I_F \approx V_F/R_L$,它由外电路的参数决定,而几乎与二极管无关。因此,只要电路在 $t=0$ 时加入 $+V_F$ 的电压,回路的电流几乎是立即达到 V_F/R_L 。这就是说,二极管的开通时间是很短的,它对开关速度的影响很小,以致可以忽略不计。

复习思考题

2.1.1 影响二极管开关速度的主要因素是什么?

2.2 BJT 的开关特性

2.2.1 BJT 的开关作用

BJT的开关作用对应于有触点开关的“断开”和“闭合”。图2.2.1a所示电

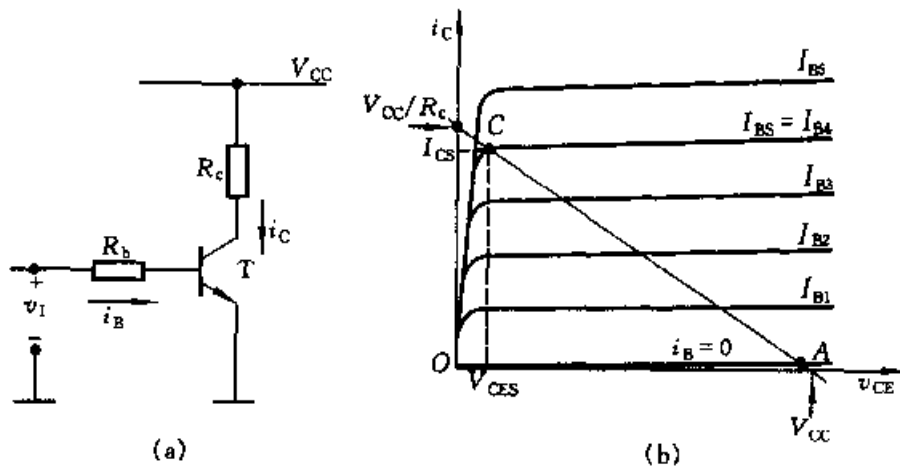


图 2.2.1 BJT 的开关工作状态

(a) 电路 (b) 工作状态图解

路用来说明 BJT 开关作用,图中 BJT 为 NPN 型硅管。当输入电压 $v_i = -V_{BE}$ 时,BJT 的发射结和集电结均为反向偏置 ($v_{BE} < 0, v_{BC} < 0$),只有很小的反向漏电流 I_{EBO} 和 I_{CBO} 分别流过两个结,故 $i_B \approx 0, i_C \approx 0, V_{CE} \approx V_{CC}$,对应于图 2.1.2b 中的 A 点。这时集电极回路中的 c、e 极之间近似于开路,相当于开关断开一

样。BJT的这种工作状态称为**截止**^①。

当 $v_1 = +V_{B2}$ 时,调节 R_b ,使 $i_B = V_{CC}/\beta R_c$,则 BJT 工作在图 2.2.1b 中的 C 点,集电极电流 i_C 已接近于最大值 V_{CC}/R_c ,由于 i_C 受到 R_c 的限制,它已不可能像放大区那样随着 i_B 的增加而成比例地增加了(当然,由于输出特性在上升部分不是完全重合的,故 i_B 增加时, i_C 会有微小的增加),即认为集电极电流已达到**饱和**,对应的基极电流称为**基极临界饱和电流** $I_{BS}(V_{CC}/\beta R_c)$,而集电极电流称为**集电极饱和电流** $I_{CS}(V_{CC}/R_c)$ 。此后,如果再增加基极电流,则饱和程度加深,但集电极电流基本上保持在 I_{CS} 不再增加,集电极电压

$$v_{CE} = V_{CC} - I_{CS}R_c = V_{CES} \approx 0.2 \sim 0.3 \text{ V}$$

这个电压称为 BJT 的**饱和压降**,它也基本上不随 i_B 增加而改变。由于 V_{CES} 很小,集电极回路中的 c、e 极之间近似于短路,相当于开关闭合一样。BJT 的这种工作状态称为**饱和**。由于 BJT 饱和后管压降均为 0.3 V,而发射结偏压为 0.7 V,因此饱和后集电结为正向

偏置,亦即 BJT 饱和时集电结和发射结均处于正向偏置,这是判断 BJT 工作在饱和状态的重要依据。图 2.2.2 示出了 NPN 型 BJT 饱和时各电极电压的典型数据,以供参考。

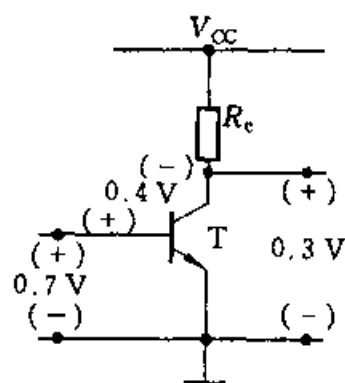


图 2.2.2 NPN 型硅 BJT 饱和时各电极电压的典型数据

表 2.2.1 NPN 型 BJT 截止、放大、饱和工作状态的特点

工作状态	截止	放大	饱和	
条件	$i_B \approx 0$	$0 < i_B < \frac{I_{CS}}{\beta}$	$i_B > \frac{I_{CS}}{\beta}$	
工作特点	偏置情况	发射结和集电结均为反偏	发射结正偏,集电结反偏	发射结和集电结均为正偏
	集电极电流	$i_C \approx 0$	$i_C \approx \beta i_B$	$i_C = I_{CS} \approx \frac{V_{CC}}{R_c}$ 且不随 i_B 增加而增加
	管压降	$V_{CE0} \approx V_{CC}$	$V_{CE} = V_{CC} - i_C R_c$	$V_{CES} \approx 0.2 \sim 0.3 \text{ V}$
	c、e 间等效内阻	很大,约为数百千欧,相当于开关断开	可变	很小,约为数百欧,相当于开关闭合

^① 对于 NPN 型硅管来说,为使其工作于截止状态,发射结并不一定要求为反向偏置,当 $v_{BE} < 0.5 \text{ V}$ 时,即已进入截止区。

由此可见,BJT 相当于一个由基极电流所控制的无触点开关,BJT 截止时相当于开关“断开”,而饱和时相当于开关“闭合”。NPN 型 BJT 截止、放大、饱和三种工作状态的特点列于表 2.2.1 中,以便比较。

2.2.2 BJT 的开关时间

BJT 的开关过程和二极管一样,也是内部电荷“建立”和“消散”的过程。因此,BJT 饱和与截止两种状态的相互转换也是需要一定的时间才能完成的。

如在图 2.2.1a 所示电路的输入端加入一个如图 2.2.3a 所示的理想方波,其幅度在 $-V_{BI}$ 和 $+V_{B2}$ 之间变化,则输出电流 i_C 的波形^① 如图 2.2.3b 所示。可见 i_C 的波形已不是和输入波形一样的理想方波,起始部分和平顶部分都延迟了一段时间,上升和下降沿都变得缓慢了。为了对 BJT 开关的瞬态过程进行定量描述,通常引入以下几个参数来表征:

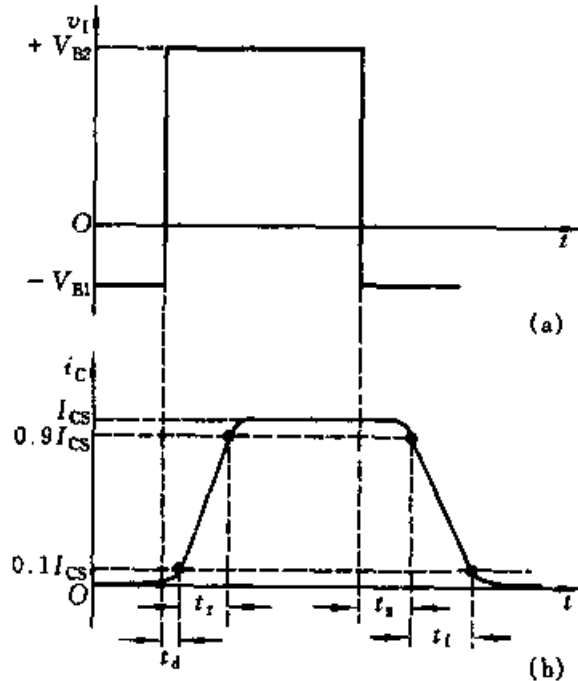


图 2.2.3 开关电路的波形

(a) 输入电压波形 (b) 输出电流波形

延迟时间 t_d ——从 $+V_{B2}$ 加入到集电极电流 i_C 上升到 $0.1I_{CS}$ 所需的时间;

上升时间 t_r —— i_C 从 $0.1I_{CS}$ 上升到 $0.9I_{CS}$ 所需的时间;

存储时间 t_s ——从输入信号降到 $-V_{B1}$ 到 i_C 降到 $0.9I_{CS}$ 所需的时间;

下降时间 t_f —— i_C 从 $0.9I_{CS}$ 降到 $0.1I_{CS}$ 所需的时间。

^① i_C 的波形与输出电压 v_{CE} 的波形形状相同,相位相反,可通过测量 v_{CE} 的波形求得 i_C 的波形。

以上4个参数称为BJT的开关时间参数,如图2.2.3b所示。它们都是以集电极电流 i_C 的变化为基准的。通常把 $t_{on} = t_d + t_r$ 称为**开通时间**,它反映了BJT从截止到饱和所需的时间,而把 $t_{off} = t_s + t_f$ 称为**关闭时间**,它反映了BJT从饱和到截止所需的时间。开通时间和关闭时间总称为BJT的开关时间,它随管子类型不同而有很大差别,一般在几十至几百纳秒的范围,可以从器件手册中查到。

BJT的开关时间限制了BJT开关运用的速度。开关时间越短,开关速度越高。因此,要设法减小开关时间。下面分别讨论造成开关时间的原因,以便能正确合理使用器件。

1. 延迟时间 t_d

由于BJT截止时($v_i = -V_{BE}$ 时),发射结和集电结都是反向偏置,故势垒区较宽,势垒区中有较多的空间电荷。当输入电压由 $-V_{BE}$ 跳变到 $+V_{BE}$ 时,便有正向基极电流 I_{B1} 流入基区, I_{B1} 的作用是抵消势垒区中的电荷,使势垒区变窄,致使发射区的电子逐渐注入到基区,并扩散到集电结而被集电极所收集,形成集电极电流 i_C 。这就是产生 t_d 的原因。正向基极电流越大, t_d 就越短。

2. 上升时间 t_r

经过延迟时间后,发射区不断向基区注入电子,形成集电极电流 i_C 。但是,开始时注入的电子较少,基区中电子浓度很小,经过一定的时间后,基区中才建立起相应于 $0.9I_{CS}$ 所需的电子浓度梯度。

3. 存储时间 t_s

经过上升时间以后,集电极电流继续增加到 I_{CS} ,这时由于进入了饱和,集电结转为正向偏置,收集电子的能力减弱,造成超量的电子电荷在基区存储,同时集电区靠近结势垒区的边界处也积累起一定的空穴电荷。

当输入电压跳变到 $-V_{BE}$ 时,上述存储电荷不能立即消散,使 i_C 不能立即下降而要维持一段时间 t_s ,它决定于存储电荷的多少,饱和越深,存储电荷越多, t_s 越大。

实际上,当输入电压跳变到 $-V_{BE}$ 后,上述存储电荷便在反向电压作用下漂移形成反向基极电流 I_{B2} ,这相当于从基区和集电区把这些存储电荷“吸出”一样,加速了转换过程。所以反向基极电流的数值越大, t_s 也就越短。

4. 下降时间 t_f

可以认为, t_f 是基区中对应于 $0.9I_{CS}$ 的存储电荷消散所需要的时间,也就是BJT从临界饱和和经过放大区到截止区的时间。

综上所述可见,开通时间 $t_{on}(=t_d+t_r)$ 就是建立基区电荷时间。关闭时间 $t_{off}(=t_s+t_f)$ 就是存储电荷消散的时间。^①

可用改进管子内部构造和外电路的方法来提高 BJT 的开关速度。对管子内部,可减小基区宽度和发射结、集电结的面积,也可以在基区掺金形成复合中心,这些措施都可以使开关时间缩短。对外电路来说,可适当选择正向基极电流 I_{B1} 和反向基极电流 I_{B2} 以及临界饱和电流 I_{CS} 等。手册给出的开关时间都是在一定的条件下测出的,如 NPN 外延平面型硅管 3DK8B 的开关时间: $t_d \leq 10 \text{ ns}$, $t_r \leq 80 \text{ ns}$, $t_f \leq 100 \text{ ns}$ (测试条件为: $I_{CS} = 100 \text{ mA}$, $I_B = 10 \text{ mA}$, $R_L = 50 \Omega$)。

复习思考题

2.2.1 数字电路中的 BJT 工作在何种工作状态? 它与放大电路中的 BJT 有何不同?

2.2.2 影响 BJT 开关速度的有哪些因素? 其中最主要的因素是什么?

2.3 基本逻辑门电路

基本逻辑运算有与、或、非运算,这在第 1 章中已作过简要的介绍。相应的基本逻辑门有与、或、非门。本节介绍简单的二极管门电路和 BJT 反相器(非门),作为逻辑门电路的基础。

2.3.1 二极管与门及或门电路

1. 与门电路

用电子电路来实现逻辑运算时,它的输入、输出量均为电压(以 V 为单位)或电平(用 1 或 0 表示)。输入量作为条件,输出量作为结果,输入与输出量之间能满足与逻辑关系的电路,称为与门电路^②。

图 2.3.1a 表示由半导体二极管组成的与门电路,图 2.3.1b 为它的代表符号。图中 A、B、C 为输入端,L 为输出端。输入信号为 +5 V 或 0 V。此电路按输入信号的不同可有下列两种情况:

(1) 若输入端中有任意一个例如 V_A 为 0 V,另两个为 +5 V,在这种情况下, D_1 导通,使 L 点电压 V_L 被钳制在 0 V。此时, D_2 、 D_3 受反向电压作用而截止,所以 $V_L \approx 0 \text{ V}$ 。

^① BJT 的开关时间也可利用发射结和集电结的结电容储能效应进行分析。

^② 与门电路的英文名称为 AND gate。

由此可见,与门几个输入端中,只有加低电压输入的二极管才导通,并把 L 钳制在低电压(接近 0 V),而加高电压输入的二极管都截止。

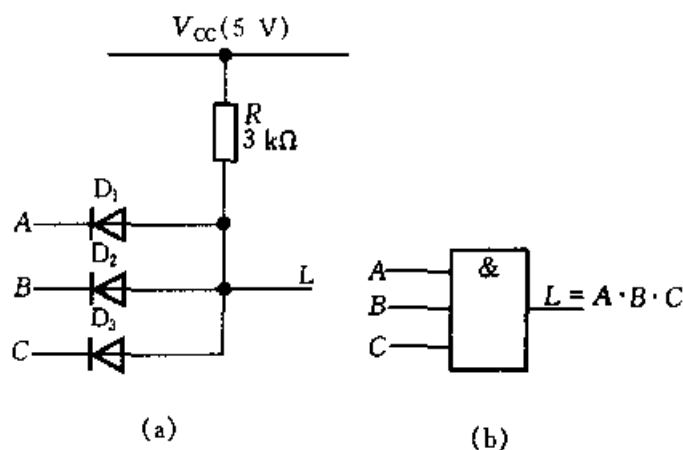


图 2.3.1 二极管与门

(a) 电路 (b) 逻辑符号

(2) 输入端 A 、 B 、 C 都处于高电压 $+5\text{ V}$, 这时, D_1 、 D_2 、 D_3 都截止, 所以输出端 L 点电压 V_L 与 $+V_{CC}$ 相等, 即 $V_L = +5\text{ V}$ 。

把上述分析结果归纳起来列入表 2.3.1 中, 可见图 2.3.1 的电路满足与逻辑的要求: 只有所有输入端都是高电压时, 输出才是高电压, 否则输出就是低电压, 所以它是一种与门。

按照第 1 章介绍过的规定, $+5\text{ V}$ 为高电平并用逻辑 1 表示; 0 V 为低电平, 用逻辑 0 表示, 于是表 2.3.1 可表示为表 2.3.2 的形式, 这是三变量的真值表。

表 2.3.1 与门输入与输出电压的关系

输 入			输 出
V_A/V	V_B/V	V_C/V	V_L/V
0	0	0	0
0	0	+5	0
0	+5	0	0
0	+5	+5	0
+5	0	0	0
+5	0	+5	0
+5	+5	0	0
+5	+5	+5	+5

表 2.3.2 与逻辑真值表

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

从表 2.3.2 可明显地看到, L 与 A 、 B 、 C 之间的关系是, 只有当 A 、 B 、 C 都是 1 时, L 才为 1, 否则 L 为 0, 其逻辑表达式为

$$L = A \cdot B \cdot C \quad (2.3.1)$$

2. 或门电路

图 2.3.2a 表示由二极管组成的或门^① 电路, 因其输出、输入量之间能满足或逻辑的关系而得名。图 2.3.2b 是它的逻辑符号。

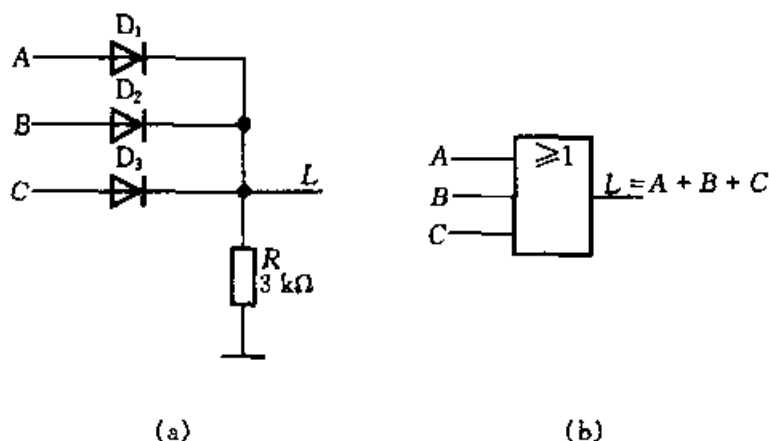


图 2.3.2 二极管或门

(a) 电路 (b) 逻辑符号

对这种电路也可分为两种情况进行分析:

- (1) 输入端 A 、 B 、 C 都处于 0 V 时, D_1 、 D_2 、 D_3 都处于截止状态, $V_L = 0$ V。
- (2) 若输入端中有任何一个例如 V_A 为 +5 V, 而另两个为 0 V 时, D_1 导通, 使 V_L 处于高电压, D_2 、 D_3 受反向电压作用而截止, 这时 $V_L \approx +5$ V。

如用二值数字逻辑中的 1 和 0 分别表示高、低电平, 则上述逻辑关系可列成真值表, 如表 2.3.3 所示。

表 2.3.3 说明, A 、 B 、 C 中只要有一个为 1, L 就是 1, 这就是或逻辑关系, 可

① 或门的英文名称为 OR gate。

写成下列逻辑表达式

$$L = A + B + C \quad (2.3.2)$$

表 2.3.3 或逻辑真值表

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

2.3.2 非门电路^① ——BJT 反相器

图 2.3.3a 表示一基本反相器电路,它的开关性能已在 2.2 节作过详细的讨论。图 2.3.3b 表示反相器的传输特性,图中标出了 BJT 的三个工作区域。

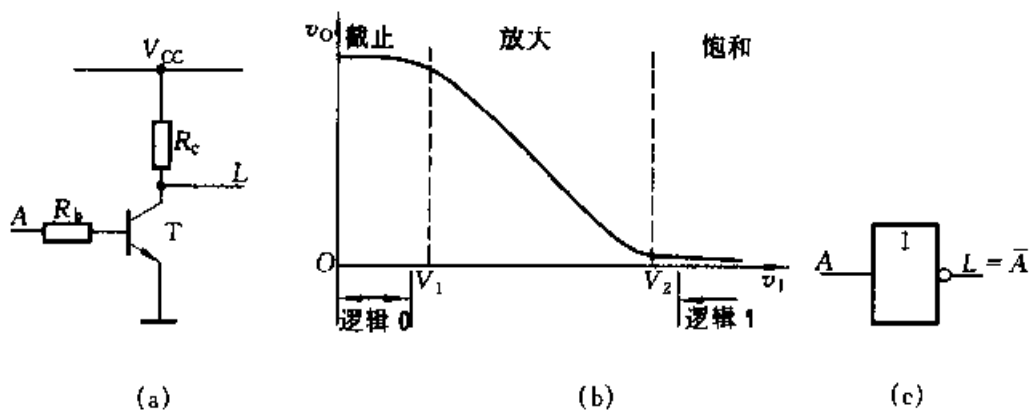


图 2.3.3 基本 BJT 反相器

(a) 电路 (b) 传输特性 (c) 代表符号

对于饱和型反相器来说,输入信号必须满足下列条件:

$$\text{逻辑 } 0: v_i < V_1$$

$$\text{逻辑 } 1: v_i > V_2$$

由传输特性可见,当输入为逻辑0时,BJT 将截止,输出电压将接近于 V_{CC} ,即逻辑1。另一方面,当输入为逻辑1时,BJT 将饱和导通,输出电压约为 $0.2 \sim 0.3 \text{ V}$,即为逻辑0。可见反相器的输出与输入量之间的逻辑关系是非逻辑关系,其真值表如表 2.3.4 所示。图 2.3.3c 为非门的代表符号。

^① 非门电路的英文名称为 NOT gate。

表 2.3.4 非逻辑真值表

输入 A	输出 L
0	1
1	0

以上所讨论的是基本的与、或、非门,利用它们可以实现与、或、非三种逻辑运算。但是由于它们的输出电阻比较大,带负载的能力差,开关性能也不理想,因而下面将讨论性能更好的逻辑门电路。

复习思考题

2.3.1 什么叫与逻辑和或逻辑?如何用二极管来实现与门和或门?

2.3.2 什么叫非逻辑?为什么可以利用 BJT 来构成非门?模拟电路中的反相电压放大器与数字电路中的非门(反相器)在电路结构和工作模式方面有何不同?

2.3.3 利用二极管和 BJT 可以构成数字逻辑运算中所需的与、或、非三种门电路,它们有什么缺点?应如何改进。

2.4 TTL 逻辑门电路

TTL 逻辑门电路由若干 BJT 和电阻组成。这种门电路于本世纪 60 年代即已问世,随后经过电路结构和工艺方面的改进,至今仍广泛应用于各种数字电路或系统中。TTL 电路的基本环节是带电阻负载的 BJT 反相器(非门),为了改善它的开关速度和其他性能,往往还需增加其他若干元器件。下面首先来讨论一下基本的 BJT 反相器的开关速度不高的原因,再讨论改进的 TTL 反相器和 TTL 逻辑门电路。

2.4.1 基本的 BJT 反相器的动态性能

在 2.2.2 节中,对 BJT 的开关特性已作了简要的分析,BJT 开关速度受到限制的原因主要是由于 BJT 基区内存储电荷的影响,电荷的存入和消散需要一定的时间。

当图 2.3.3 所示的基本反相器考虑负载电容 C_L 的影响时,其电路如图 2.4.1 所示。 C_L 中包含门电路之间的接线电容以及门电路的输入电容。当反相器输出电压 v_o 由低向高过渡时,电路由 V_{CC} 通过 R_c 对 C_L 充电。反之,当 v_o 由高向低过渡时, C_L 又将通过 BJT 放电。这样, C_L 的充、放电过程均需经历一定的时间,这必然会增加输出电压 v_o 波形的上升时间和下降时间。特别是 C_L 充电回路的时间常数 $R_c C_L$ 较大时, v_o 上升较慢,即增加了上升时间。

基于器件内部和负载电容的影响,导致基本BJT反相器的开关速度不高。寻求更为实用的TTL电路结构,是下面所要讨论的问题。

2.4.2 TTL反相器的基本电路

由前面的分析已知,带电阻负载的BJT反相器,其动态性能不理想。在保持逻辑功能不变的前提下,可以另外增加若干元器件以改善其动态性能,如减少由于BJT基区电荷存储效应和负载电容所引起的时延。这需改变反相器输入电路和输出电路的结构,以形成TTL反相器的基本电路。

图2.4.2表示TTL反相器的基本电路,它是针对图2.4.1所示电路存在的问题而提出的改进电路。该电路由三部分组成,即由BJT T_1 组成电路的输入级, T_3 、 T_4 和二极管 D 组成输出级,以及由 T_2 组成的中间级作为输出级的驱动电路,将 T_2 的单端输入信号 v_{i2} 转换为互补的双端输出信号 v_{i3} 和 v_{i4} ,以驱动 T_3 和 T_4 。

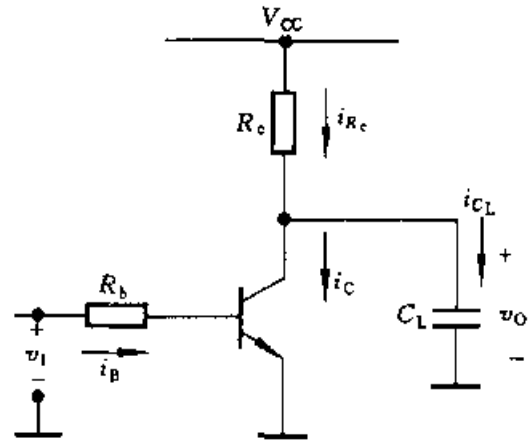


图 2.4.1 带负载电容 C_L 的BJT反相器

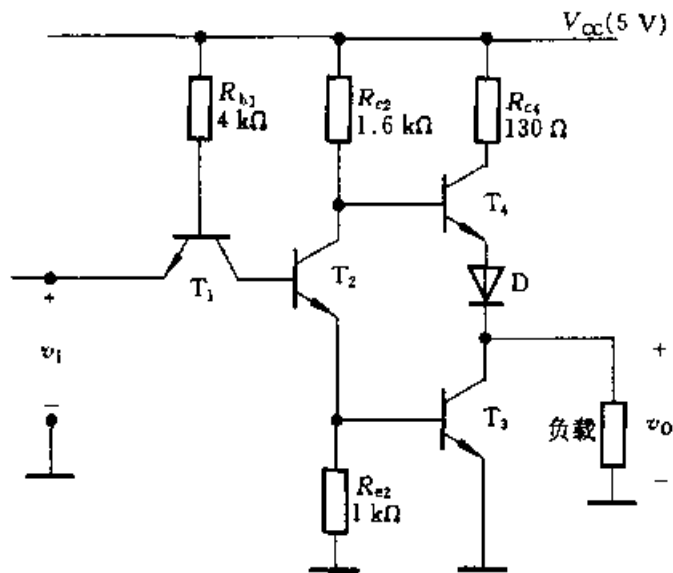


图 2.4.2 TTL反相器的基本电路

下面首先分析TTL反相器的工作原理,然后对输入级和输出级的性能作重点分析。

1. TTL反相器的工作原理

这里主要分析TTL反相器的逻辑关系,并估算电路中有关各点的电压,以

得到简单的定量概念。

(1) 当输入为高电平, 如 $v_i = 3.6 \text{ V}$ 时, 电源 V_{CC} 通过 R_{B1} 和 T_1 的集电结向 T_2 、 T_3 提供基极电流, 使 T_2 、 T_3 饱和, 输出为低电平, 如 $v_o = 0.2 \text{ V}$ 。此时

$$V_{B1} = V_{BC1} + V_{BE2} + V_{BE3} = (0.7 + 0.7 + 0.7) \text{ V} = 2.1 \text{ V} \quad (2.4.1)$$

显然, 这时 T_1 的发射结处于反向偏置, 而集电结处于正向偏置。所以 T_1 处于发射结和集电结倒置使用的放大状态。由于 T_2 和 T_3 饱和, 输出 $V_{C3} = 0.2 \text{ V}$, 同时可估算出 V_{C2} 的值:

$$V_{C2} = V_{CES2} + V_{BE3} = (0.2 + 0.7) \text{ V} - 0.9 \text{ V} \quad (2.4.2)$$

此时, $V_{B4} = V_{C2} = 0.9 \text{ V}$ 。作用于 T_4 的发射结和二极管 D 的串联支路的电压为 $V_{C2} - V_o = (0.9 - 0.2) \text{ V} = 0.7 \text{ V}$, 显然, T_4 和 D 均截止, 实现了反相器的逻辑关系: 输入为高电平时, 输出为低电平。

(2) 当输入为低电平且电压为 0.2 V 时, T_1 的发射结导通, 其基极电压等于输入低电压加上发射结正向压降, 即

$$V_{B1} = (0.2 + 0.7) \text{ V} = 0.9 \text{ V} \quad (2.4.3)$$

此时 V_{B1} 作用于 T_1 的集电结和 T_2 、 T_3 的发射结上, 所以 T_2 、 T_3 都截止, 输出为高电平。

由于 T_2 截止, V_{CC} 通过 R_{C2} 向 T_4 提供基极电流, 致使 T_4 和 D 导通, 其电流流入负载。输出电压为

$$v_o \approx V_{CC} - V_{BE4} - V_D = (5 - 0.7 - 0.7) \text{ V} = 3.6 \text{ V} \quad (2.4.4)$$

同样也实现了反相器的逻辑关系: 输入为低电平时, 输出为高电平。

2. 采用输入级以提高工作速度

当 TTL 反相器输入电压由高 (3.6 V) 变低 (0.2 V) 的瞬间, $V_{B1} = (0.2 + 0.7) \text{ V} = 0.9 \text{ V}$ 。但由于 T_2 、 T_3 原来是饱和的, 它们的基区存储电荷还来不及消散, 在此瞬间, T_2 、 T_3 的发射结仍处于正向偏置, T_1 的集电极电压为

$$V_{C1} = V_{BE2} + V_{BE3} = (0.7 + 0.7) \text{ V} = 1.4 \text{ V} \quad (2.4.5)$$

此时, T_1 的集电结为反向偏置 [集电结电压 = $V_{B1} - V_{C1} = (0.9 - 1.4) \text{ V} = -0.5 \text{ V}$], 因输入为低电平 (0.2 V) 时, T_1 的发射结为正向偏置, 于是 T_1 工作在放大区。这时产生基极电流 i_{B1} , 其射极电流 $\beta_1 i_{B1}$ 流入低电平的输入端。集电极电流 $i_{C1} \approx \beta_1 i_{B1}$ 的方向是从 T_2 的基极流向 T_1 的集电极, 它很快地从 T_2 的基区抽走多余的存储电荷, 使 T_2 迅速地脱离饱和而进入截止状态。 T_2 的迅速截止导致 T_4 立刻导通, 相当于 T_3 的负载是个很小的电阻, 使 T_3 的集电极电流加大, 多余的存储电荷迅速从集电极消散而达到截止, 从而加速了状态转换。

3. 采用推拉式输出级以提高开关速度和带负载能力

图 2.4.2 采用了由 T_3 、 T_4 组成推拉式输出级^①。其中 T_4 组成电压跟随器,而 T_3 为共射极电路,作为 T_4 的射极负载。这种输出级的优点是,既能提高开关速度,又能提高带负载能力。根据所接负载的不同,输出级的工作情况可归纳如下:

(1) 当输出为低电平时, T_3 处于深度饱和状态,反相器的输出电阻就是 T_3 的饱和电阻,这时可驱动较大的电流负载。而且由于 T_4 截止,所以负载电流就是 T_3 的集电极电流,也就是说, T_3 的集电极电流可以全部用来驱动负载。

(2) 输出为高电平时, T_3 截止, T_4 组成的电压跟随器的输出电阻很小,所以输出高电平稳定,带负载能力也较强。

(3) 输出端接有负载电容 C_L 时,当输出由低电平跳变到高电平的瞬间, T_2 和 T_3 由饱和转为截止,由于 T_3 的基极电流是经 T_2 放大的电流,所以 T_2 比 T_3 更早脱离饱和,于是 T_2 的集电极电压 v_{C2} 比 T_3 的集电极电压 v_{C3} 上升更快。同时由于电容 C_L 两端的电压不能突变,使 c_2 和 c_3 之间的电位差增加,因而使 T_4 在此瞬间基极电流很大, T_4 集电极与发射极之间呈现低电阻,故电源 V_{CC} 经 R_{C4} 和 T_4 的饱和电阻对电容 C_L 迅速充电,其时间常数很小,使输出波形上升沿陡直。而当输出电压由高变低后,输出管 T_3 深度饱和,也呈现很低的电阻,已充电的 C_L 通过它很快放电,迅速达到低电平,因而使输出电压波形的上升沿和下降沿都很好。

2.4.3 TTL 反相器的传输特性

现在来分析图 2.4.2 所示 TTL 反相器的传输特性。这里所指的传输特性是 $v_O = f(v_I)$ 的关系曲线,它应该是一条光滑的曲线,但为了简明起见,下面用折线作近似的描述,并估算各个转折点的 v_I 、 v_O 值,如图 2.4.3 所示。

由图可见,传输特性由 4 条线段 AB、BC、CD 和 DE 所组成。现分别介绍如下:

AB 段:此时输入电压 v_I 很低, T_1 的发射结为正向偏置。在稳态情况下, T_1 饱和致使 T_2 和 T_3 截止,同时 T_4 导通。根据上一节的分析,输出高电平时, $v_O = 3.6 \text{ V}$ 。

当 v_I 增加直至 B 点, T_1 的发射结仍维持正向偏置并处于饱和状态。但 $v_{BE} = v_{C1}$ 增大导致 T_2 的发射结正向偏置。当 T_1 仍维持在饱和状态时, v_{BE} 的值可表示为

$$v_{BE} = v_I + V_{CES} \quad (2.4.6)$$

为求得 B 点所对应的 v_I ,可以考虑 v_{BE} 刚好使 T_2 的发射结正向偏置并开始导

^① 又称为带有源上拉(active pull-up)电路或图腾柱(Totem pole)输出级。

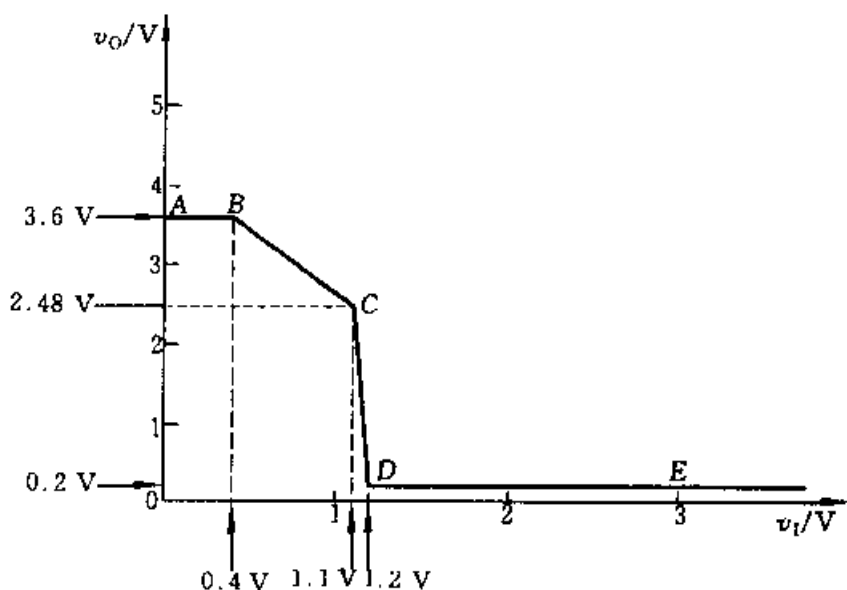


图 2.4.3 基本的 TTL 反相器的传输特性

电。此时 v_{BE} 应等于 T_2 发射结的正向电压 $V_F \approx 0.6$ V。但 $i_{E2} \approx 0$, 在忽略 $v_{R_{c2}}$ 的情况下, 于是由式(2.4.6)得

$$v_1(B) = V_F - V_{CES} = 0.6 \text{ V} - 0.2 \text{ V} \approx 0.4 \text{ V} \quad (2.4.7)$$

BC 段: 当 v_1 的值大于 B 点的值时, 由 T_1 的集电极供给 T_2 的基极电流, 但 T_1 仍保持为饱和状态, 这就需要使 T_1 的发射结和集电结均为正向偏置。

在 BC 段内, T_2 对 v_1 的增量作线性放大, 其电压增益可表示为

$$\frac{\Delta v_{C2}}{\Delta v_{B2}} \approx -\frac{R_{c2}}{R_{e2}} \text{①}$$

电压增量 Δv_{C2} 通过 T_4 的电压跟随作用而引至输出端, 形成输出电压的增量 $-(R_{c2}/R_{e2})\Delta v_{B2}$, 考虑到式(2.4.6)的关系, 在一定范围内, 有 $\Delta v_{B2} = \Delta v_1$, 所以传输特性 BC 段的斜率为 $dv_o/dv_1 = -R_{c2}/R_{e2} = -1.6$ 。必须注意到在 BC 段内, R_{c2} 上所产生的电压降还不足以使 T_3 的发射结正向偏置, T_3 仍维持截止状态。

一旦 R_{c2} 上的电压 $v_{R_{c2}}$ 达到一定的值, 能使 T_3 的发射结正偏, 并有 $v_{BE3} = V_F = 0.7$ V 时, 则有

$$v_{BE3} = i_{E2} R_{c2} = V_F$$

或

$$i_{E2} = \frac{V_F}{R_{c2}} = \frac{0.7 \text{ V}}{1 \text{ k}\Omega} = 0.7 \text{ mA}$$

式中 $V_F = 0.7$ V 表示 T_3 已导通。由于 $i_{C2} \approx i_{E2} = 0.7$ mA, C 点处的输出电压

① 见本书模拟部分第 7 章, 例 7.4.2。

变为

$$\begin{aligned} v_o(C) &= V_{CC} - i_{C2} R_{L2} - 2V_F \\ &= 5 \text{ V} - (0.7 \text{ mA})(1.6 \text{ k}\Omega) - 2 \times (0.7 \text{ V}) \approx 2.48 \text{ V} \end{aligned} \quad (2.4.8)$$

由于在 C 点 T_4 和 T_3 均已导通,因而在式(2.4.8)中设 $V_F \approx 0.7 \text{ V}$ 。根据线段 BC 的斜率为 -1.6 ,对应于 C 点的 v_i 值可由下述关系求得:

$$\frac{\Delta v_o}{\Delta v_i} = \frac{v_o(C) - v_o(B)}{v_i(C) - v_i(B)} = -1.6$$

由此得

$$\begin{aligned} v_i(C) &= \frac{v_o(C) - v_o(B)}{-1.6} + v_i(B) \\ &= \frac{2.48 \text{ V} - 3.6 \text{ V}}{-1.6} + 0.4 \text{ V} \approx 1.1 \text{ V} \end{aligned} \quad (2.4.9)$$

CD 段:当 v_i 的值继续增加并超越 C 点,使 T_3 饱和导通,输出电压迅速下降至 $v_o \approx 0.2 \text{ V}$ 。D 点处的 $v_i(D)$ 值,可以根据 T_2 、 T_3 两发射结电压 $V_F \approx 0.7 \text{ V}$ 来估算。因此有

$$v_i(D) = v_{BE3} + v_{BE2} - v_{CES1} = (0.7 + 0.7 - 0.2) \text{ V} = 1.2 \text{ V} \quad (2.4.10)$$

DE 段:当 v_i 的值从 D 点再继续增加时, T_1 将进入倒置放大状态,保持 $v_o = 0.2 \text{ V}$ 。至此,得到了 TTL 反相器的 ABCDE 折线型传输特性。

2.4.4 TTL 与非门电路

图 2.4.2 所示的基本 TTL 反相器不难改变成为多输入端的与非门。它的主要特点是在电路的输入端采用了多发射极的 BJT,如图 2.4.4 所示。器件中的每一个发射极能各自独立地形成正向偏置的发射结,并可促使 BJT 进入放大或饱和区。两个或多个发射极可以并联地构成一大面积的组合发射极。

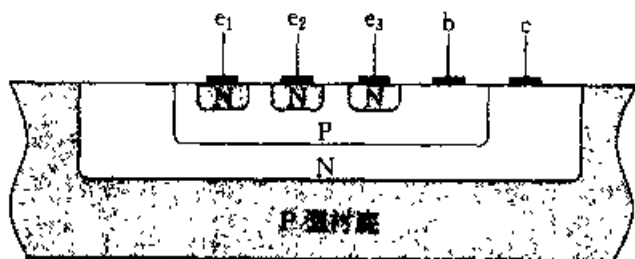


图 2.4.4 NPN 型多发射极 BJT 的结构示意图

图 2.4.5a 说明采用多发射极 BJT 用作 3 输入端 TTL 与非门的输入器件。当任一输入端为低电平时, T_1 的发射结将正向偏置而导通, T_2 将截止。结果将导致输出为高电平。只有当全部输入端为高电平时, T_1 将转入倒置放大状态,

T_2 和 T_3 均饱和,输出为低电平。

图 2.4.5b 为 3 输入端 TTL 与非门的代表符号。

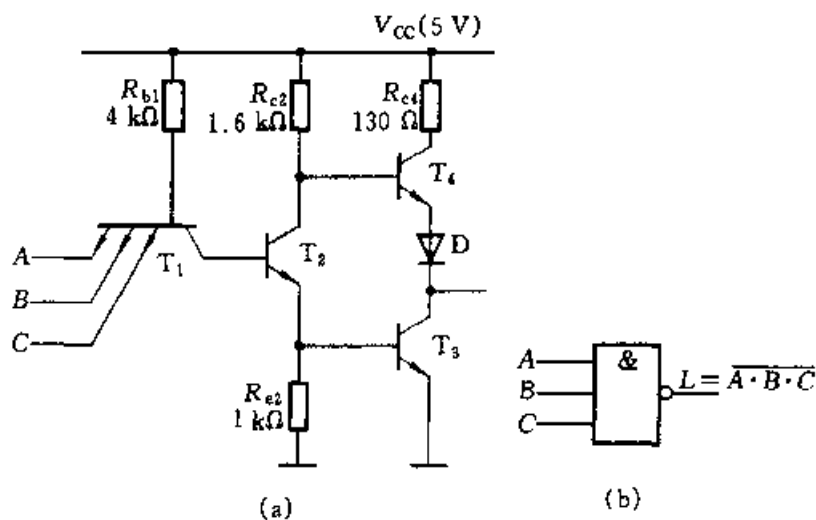


图 2.4.5 具有多发射极 BJT 的 3 输入端与非门电路

(a) 电路图 (b) 代表符号

2.4.5 TTL 与非门的技术参数

制造 TTL 门电路的厂家,通常都要为用户提供各种逻辑器件的数据手册,手册中要给出门电路输入 $v-i$ 特性,输出 $v-i$ 特性,整个电路的电压传输特性 v_1-v_0 ,输入和输出的高、低电压,噪声容限,传输延迟时间,功耗等。除输入和输出 $v-i$ 特性外,其他各项技术参数分别介绍如下:

1. 传输特性

各种类型的 TTL 门电路,其传输特性大同小异。前已讨论,TTL 反相器的传输特性如图 2.4.3 所示。那是一条从理论分析所得的折线 ABCDE。实用上,如有需要,可用实验方法来求得,如用示波器扫描来获得,也可用 PSPICE 程序仿真得出。其他 TTL 门电路的传输特性具有类似的特点。

2. 输入和输出的高、低电压

前已讨论,数字电路中的高、低电压常用高、低电平来描述,并规定在正逻辑体制中,用逻辑 1 和 0 分别表示高、低电平。作为门电路的技术参数常用高、低电压表示,以 V 为单位进行量化,有利于具体应用。例如,从图 2.4.3 所示的基本的 TTL 反相器的传输特性上可以求得输入和输出的高、低电压的数值如下:

$$\begin{aligned}
 \text{输出高电压} \quad V_{OH} &\approx V_{O(A)} = 3.6 \text{ V} \\
 \text{输出低电压} \quad V_{OL} &= V_{CES} = 0.2 \text{ V} \\
 \text{输入低电压} \quad V_{IL} &= V_1(B) = 0.4 \text{ V} \\
 \text{输入高电压} \quad V_{IH} &= V_1(D) = 1.2 \text{ V}
 \end{aligned} \tag{2.4.11}$$

由于不同类型的 TTL 器件,其 $v_i - v_o$ 特性各不相同,因而其输入和输出高、低电压也各异。例如,STTL 与非门^① 的传输特性上,由高电平转向低电平的瞬间非常急剧,以致 BC 和 CD 两线段合而为一,消去了 C 点,有关高低电压值亦有变化[2]。

3. 噪声容限

噪声容限表示门电路的抗干扰能力。二值数字逻辑电路的优点在于它的输入信号允许一定的容差。如将图 2.4.3 所示的反相器的传输特性中的输入、输出高、低电压绘制成图 2.4.6 所示,则由此图可知,高电平(逻辑1)所对应的电压范围($V_{IH} \sim V_{OH}$)和低电平(逻辑0)所对应的电压范围($V_{OL} \sim V_{IL}$)分别称之为高、低电平的噪声容限,用符号 V_{NH} 和 V_{NL} 表示:

$$V_{NH} = V_{OH} - V_{IH} \quad (2.4.12)$$

$$V_{NL} = V_{IL} - V_{OL} \quad (2.4.13)$$

式(2.4.11)所列的高、低电平电压是一种较理想的情况。对于 TTL 门电路(如 74 系列)来说,高、低电平所对应的标准电压值为: $V_{OL} = 0.4 \text{ V}$, $V_{OH} = 2.4 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $V_{IH} = 2 \text{ V}$ 。这些极限参数是在直流电源电压允许的波动范围内和最大的扇出数(下面将介绍)为 10 的情况下所规定的。当然,上述参数是比较保守的,亦即留有一定的余地。

根据标准高、低电平的电压值,可求得 TTL 门电路的高、低电平的噪声容限如下:

$$\text{高电平噪声容限 } V_{NH} = V_{OH} - V_{IH} = 2.4 \text{ V} - 2 \text{ V} = 0.4 \text{ V}$$

$$\text{低电平噪声容限 } V_{NL} = V_{IL} - V_{OL} = 0.8 \text{ V} - 0.4 \text{ V} = 0.4 \text{ V}$$

4. 扇入与扇出数

(1) TTL 门电路的扇入数取决于它的输入端的个数,例如一个 3 输入端的与非门,其扇入数 $N_i = 3$ 。

(2) 扇出数的情况则稍复杂,现以 TTL 与非门带同类门作为负载时来讨论。这时可有两种情况,一种情况是负载电流从外电路流入与非门,称为灌电流负载;另一种是负载电流从与非门流向外电路,称为拉电流负载。灌与拉形象地表明了负载的性质。下面分别予以介绍:

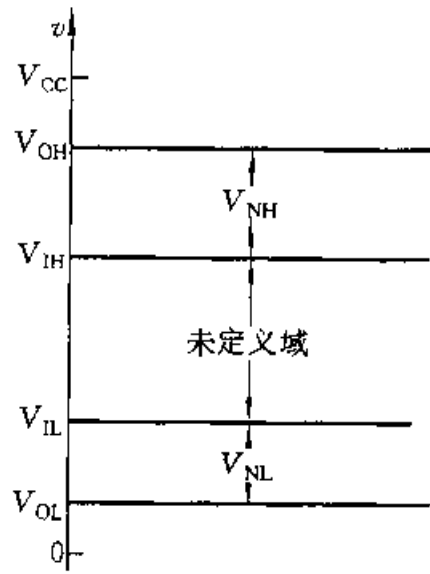


图 2.4.6 噪声容限的图解

^① 见 2.4.7 节。

① 灌电流工作情况

图 2.4.7a 表示 TTL 与非门的灌电流负载的情况。图中左边为驱动门,右边为负载门,当驱动门的输出端为逻辑 0 (低电压 V_{OL}) 时,负载门由电源 V_{CC} 通过 R_{b1} 、 T_1 的发射结和输入端有电流 I_{IL} 灌入驱动门的 T_3 的集电极,这就是灌电流负载的由来。不难理解,当负载门的个数增加时,总的灌电流 I_{IL} 将增加,同时也将引起输出低电压 V_{OL} 的升高。前已述及, TTL 门电路的标准输出低电压 $V_{OL} = 0.4 \text{ V}$,这就限制了负载门的个数。在输出为低电平的情况下,所能驱动的同类门的个数由下式决定:

$$N_{OL} = \frac{I_{OL}(\text{驱动门})}{I_{IL}(\text{负载门})} \quad (2.4.14)$$

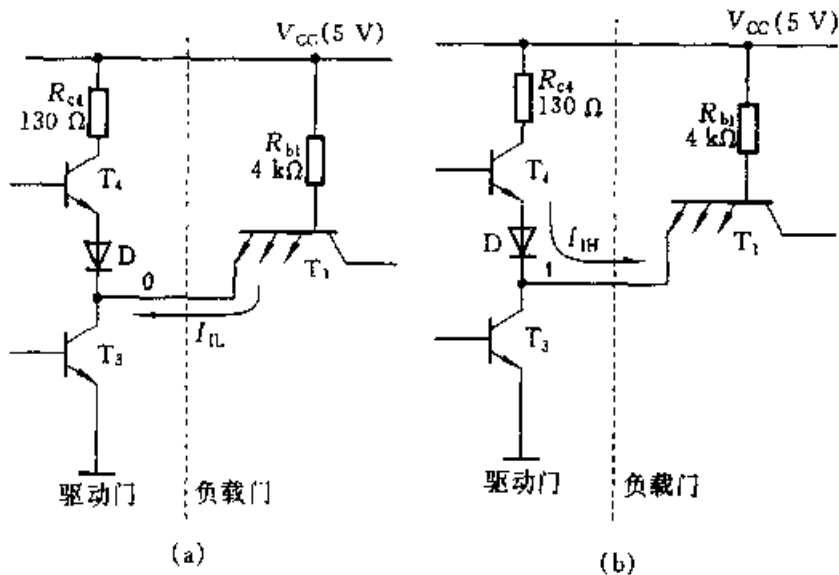


图 2.4.7 与非门的带负载能力

(a) 灌电流负载 (b) 拉电流负载

② 拉电流工作情况

当驱动门的输出为高电平时,将有电流 I_{IH} 从驱动门拉出而流至负载门。当负载门的个数增多时,必将引起输出高电压的降低,但不得低于标准高电压的低限值 $V_{IH} = 2 \text{ V}$ 。这样,输出为高电平时的扇出数可表示如下:

$$N_{OH} = \frac{I_{OH}(\text{驱动门})}{I_{IH}(\text{负载门})} \quad (2.4.15)$$

通常基本的 TTL 门电路^①,其扇出数约为 10,而性能更好的门电路的扇出数最高可达 30~50。

① 基本的 TTL 门电路所指的是国外的典型的 TTL 电路,常用 Standard TTL 标称。

一般 TTL 器件的数据手册中,并不给出扇出数,而须用计算或用实验的方法求得,并注意在设计时留有余地,以保证数字电路或系统能正常地运行。

通常,输出低电平电流 I_{OL} 大于输出高电平电流 I_{OH} , $N_{OL} \neq N_{OH}$, 因而,在实际的工程设计中,常取二者中的最小值。

例2.4.1 试计算基本的 TTL 与非门(图 2.4.5)7410 带同类门时的扇出数。

解: (1) 从 TTL 数据手册(或附录 C)上查得 7410 的参数如下:

$$I_{OL} = 16 \text{ mA}, I_{IL} = -1.6 \text{ mA}$$

$$I_{OH} = 0.4 \text{ mA}, I_{IH} = 0.04 \text{ mA}$$

数据前的负号表示电流的流向,对于灌电流取负号,计算时只取绝对值。

(2) 根据式(2.4.14)可计算低电平输出时的扇出数

$$N_{OL} = \frac{16 \text{ mA}}{1.6 \text{ mA}} = 10$$

(3) 根据式(2.4.15)可计算高电平输出时的扇出数

$$N_{OH} = \frac{0.4 \text{ mA}}{0.04 \text{ mA}} = 10$$

可见这时 $N_{OL} = N_{OH}$ 。如前所述,若 $N_{OL} \neq N_{OH}$, 则取较小的作为电路的扇出数。

5. 传输延迟时间

传输延迟时间是表征门电路开关速度的参数,它意味着门电路在输入脉冲波形的作用下,其输出波形相对于输入波形延迟了多长的时间。为便于说明,假设在门电路的输入端加入一脉冲波形,其幅度为 $0 \sim V_{CC}$ (单位为 V), 相应的输

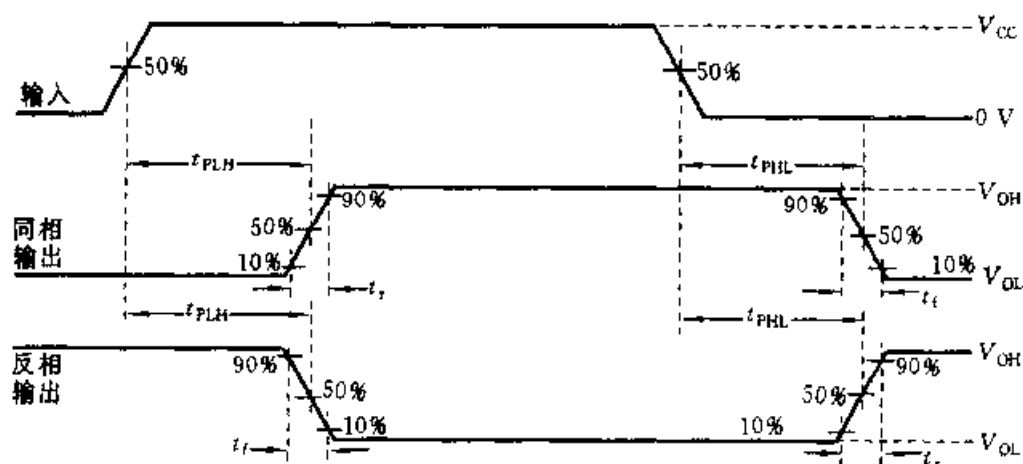


图 2.4.8 TTL 门电路传输延迟波形图

出波形如图 2.4.8 所示。通常门电路输出由低电平转换到高电平或者由高电平转换到低电平所经历的时间分别用 t_{PLH} 和 t_{PHL} 表示。有时也采用平均传输延迟

时间这一参数,即 $t_{pd} = (t_{PLH} + t_{PHL})/2$ 。例如,与非门 7402 的 $t_{PLH} = 22$ ns, $t_{PHL} = 15$ ns, $t_{pd} = (22 + 15) \text{ ns}/2 = 18.5$ ns。在图 2.4.8 中还标出了上升时间 t_r 和下降时间 t_f ,与图 2.2.3 所定义的一致。

6. 功耗

功耗是门电路重要参数之一。功耗有静态和动态之分。所谓静态功耗指的是当电路没有状态转换时的功耗,即与非门空载时电源总电流 I_{CC} 与电源电压 V_{CC} 的乘积。当输出为低电平时的功耗称为空载导通功耗 P_{ON} ;当输出为高电平时的功耗称为截止功耗 P_{OFF} , P_{ON} 总比 P_{OFF} 大。至于动态功耗,只发生在状态转换的瞬间,或者电路中有电容性负载时,例如 TTL 门电路约有 5 pF 的输入电容,由于电容的充、放电过程,将增加电路的损耗。对于 TTL 门电路来说,静态功耗是主要的。

7. 延时-功耗积

理想的数字电路或系统,要求它既具有高速度,同时功耗又低。在工程实践中,要实现这种理想情况是较难的。高速数字电路往往需要付出较大的功耗为代价。一种综合性的指标叫做延时-功耗积,用符号 DP 表示,单位为焦耳,即

$$DP = t_{pd} P_D \quad (2.4.16)$$

式中 $t_{pd} = (t_{PLH} + t_{PHL})/2$, P_D 为门电路的功耗,一个逻辑门器件的 DP 的值愈小,表明它的特性愈接近于理想情况。

8. TTL 集成门电路的封装

图 2.4.9 表示 14 脚 TTL 集成门电路。图 a 为封装图,图 b 为含有四个 2 输入端与非门的引脚分布图。

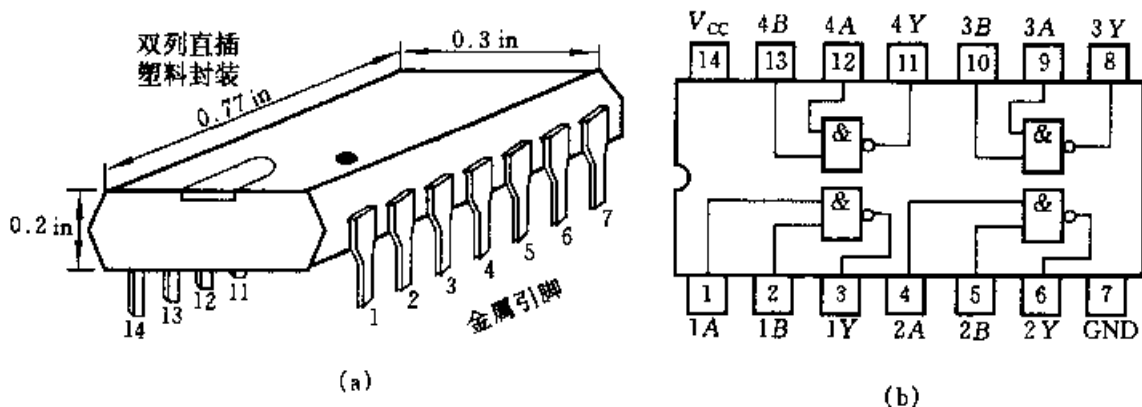


图 2.4.9 14 脚数字集成门电路

(a) 外封装图 (b) 四个 2 输入端与非门的引脚分布图

2.4.6 TTL 或非门、集电极开路门和三态门电路

上两节讨论的是基本的 TTL 与非门及其技术参数。实际上,对图 2.4.5 的

电路结构稍作改进,便可得到其他逻辑功能的门电路,如或非门、集电极开路门和三态输出门电路等,下面分别加以讨论。

1. TTL 或非门

图 2.4.10a 表示 TTL 或非门的逻辑电路,图 b 是它的代表符号。由图可见,或非逻辑功能是对 TTL 与非门(图 2.4.5)的结构改进而来,即用两个 BJT T_{2A} 和 T_{2B} 代替 T_2 。若两输入端为低电平,则 T_{2A} 和 T_{2B} 均将截止, $i_{B3} = 0$, 输出为高电平。若 A 、 B 两输入端中有一个为高电平,则 T_{2A} 或 T_{2B} 将饱和,导致 $i_{B3} > 0$, i_{B3} 便使 T_3 饱和,输出为低电平。这就实现了或非功能。即 $L = \overline{A + B} = \overline{A} \cdot \overline{B}$ 。这个式子表明,图 2.4.10a 就正逻辑而言是或非门,但就负逻辑而言则是与非门^①。

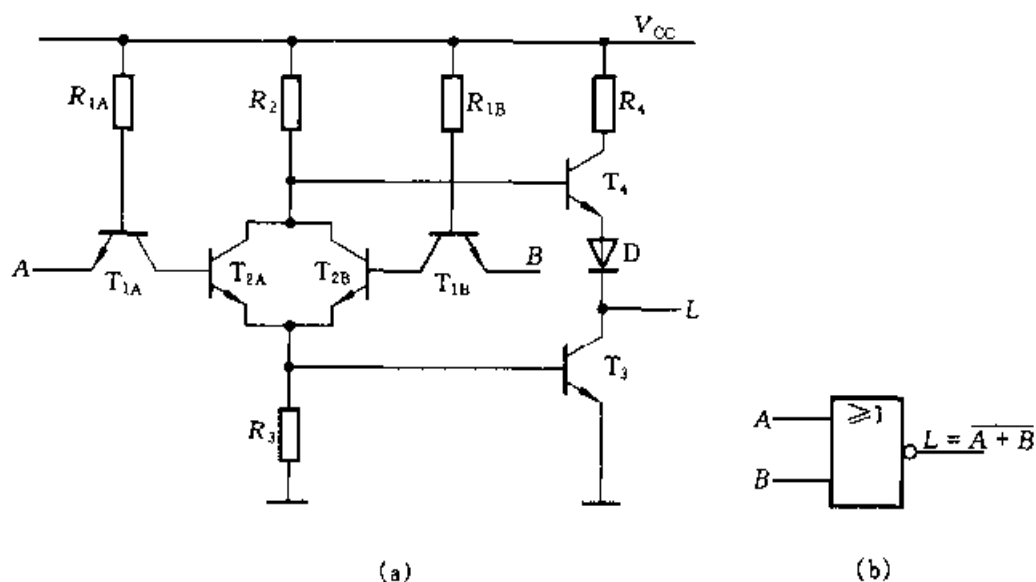


图 2.4.10 TTL 或非门
(a) 电路图 (b) 代表符号

2. 集电极开路门

在工程实践中,往往需要将两个门的输出端并联以实现与逻辑的功能称为线与。现在来考察一种情况,如将两只 TTL 门电路 G_1 和 G_2 的输出端联接在一起,并设 G_1 的输出处于高电平,而另一门 G_2 的输出为低电平。这样,从 G_1 的 T_4 到 G_2 的 T_3 将形成一低阻通路(参阅图 2.4.5),从而产生很大的电流,有可能导致器件损毁,无法形成有用的线与逻辑关系。这一问题可以采用集电极开路(OC^②)门来解决。所谓集电极开路是指 TTL 与非门电路的推拉式输出级中,删去电压跟随器,如图 2.4.11a 所示。为了实现线与的逻辑功能,可将多个

① 关于正负逻辑问题见本章 2.8 节。

② OC 系 Open Collector 的缩写。

门电路输出管 T_3 的集电极至电源 V_{CC} 之间,加一公共的上拉电阻 R_p ^①,如图 2.4.11b 所示。为了简明起见,图中以两个 OC 门并联为例。图 c 为其代表符号,其中图标“ \triangle ”表示集电极开路之意。上拉电阻 R_p 的值可以这样来计算,主要考虑 OC 门必须驱动一定的拉电流或灌电流负载。有关这两类负载的概念前已讨论,这里仍然适用,所不同的是驱动门是由多个 TTL 门的输出端直接并联而成。当 OC 门中的一个 TTL 门的输出为低电平,其他为高电平时,灌电流将由一个输出 BJT(如 T_1 或 T_2) 承担,这是一种极限情况,此时上拉电阻 R_p 具有限制电流的作用。为保证 I_{OL} 不超过额定值 $I_{OL(max)}$,必须合理选用 R_p 的值。例如 $V_{CC} = 5\text{ V}$, $R_p = 1\text{ k}\Omega$, 则 $I_{OL} = 5\text{ mA}$ 。

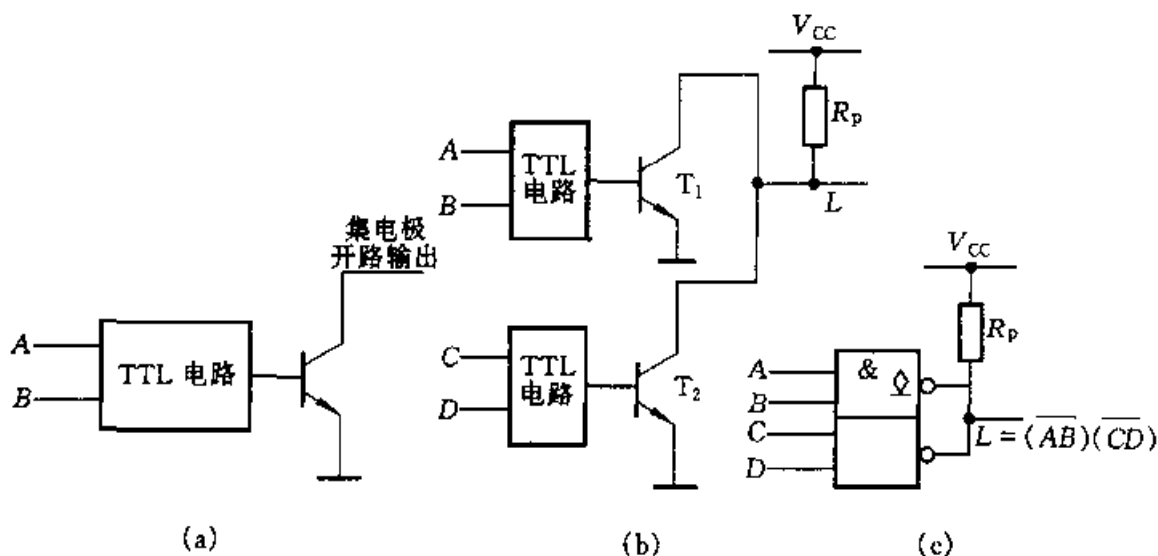


图 2.4.11 集电极开路(OC)门

(a) OC 门的输出级 (b) 由 OC 门构成的线与逻辑原理图 (c) 代表符号

另一方面,由于门电路的输出、输入电容和接线电容存在, R_p 的大小必将影响 OC 门的开关速度, R_p 的值愈大,负载电容的充电时间常数亦愈大,因而开关速度愈慢。 R_p 的最小值 $R_{p(min)}$ 可按下式来确定:

$$R_{p(min)} = \frac{V_{CC} - V_{OL(max)}}{I_{OL(max)} - I_{IL(total)}} \quad (2.4.17)$$

式中: V_{CC} —— 直流电源电压;

$V_{OL(max)}$ —— 驱动器件 V_{OL} 最大值;

$I_{OL(max)}$ —— 驱动器件 I_{OL} 最大值;

$I_{IL(total)}$ —— 接到上拉电阻下端的全部灌电流负载的 I_{IL} 总值。

R_p 的最大值 $R_{p(max)}$ 可按下式来确定:

① 上拉电阻系 pull-up resistor 的译称。

$$R_{p(max)} = \frac{V_{CC} - V_{IH(min)}}{I_{IH(max)}} \quad (2.4.18)$$

式中: V_{CC} ——直流电源电压;

$V_{IH(min)}$ ——负载器件 V_{IH} 最小值;

$I_{IH(total)}$ ——接到上拉电阻下端的全部拉电流负载的 I_{IH} 总值。

应当注意,在式(2.4.18)中,因驱动门的输出 BJT 均截止,可以认为没有 I_{OH} 流经 $R_p(max)$ 。

实际上, R_p 的值选在 $R_{p(min)}$ 和 $R_{p(max)}$ 之间,并且选用靠近 $R_{p(min)}$ 的标准值。

例 2.4.2 设 TTL 与非门 74LS01(OC) 驱动 8 个 74LS04(反相器), 试确定一合适大小的上拉电阻 R_p , 设 $V_{CC} = 5\text{ V}$ 。

解: (1) 从器件手册(或附录 C) 查出式(2.4.17) 和式(2.4.18) 中参数: $V_{CC} = 5\text{ V}$, $V_{OL(max)} = 0.5\text{ V}$, $I_{OL(max)} = 8\text{ mA}$, $I_{IL} = 400\text{ }\mu\text{A}$, $V_{IH(min)} = 2\text{ V}$, $I_{IH} = 20\text{ }\mu\text{A}$ 。

① 按式(2.4.17) 有 $I_{IL(total)} = 400\text{ }\mu\text{A} \times 8 = 3.2\text{ mA}$

得 $R_{p(min)} = \frac{5\text{ V} - 0.5\text{ V}}{8\text{ mA} - 3.2\text{ mA}} = 937\text{ }\Omega$

② 按式(2.4.18) 计算 $R_{p(max)}$, 其中 $V_{CC} = 5\text{ V}$, $I_{IH(total)} = 20\text{ }\mu\text{A} \times 8 = 0.16\text{ mA}$ 。所以

$$R_{p(max)} = \frac{5\text{ V} - 2\text{ V}}{0.16\text{ mA}} = 18.75\text{ k}\Omega$$

根据上述计算, R_p 的值可在 $937\text{ }\Omega$ 至 $18.75\text{ k}\Omega$ 之间选择。为使电路有较快的开关速度, 可选用一标准值为 $1\text{ k}\Omega$ 的电阻器为宜。

集电极开路门除了可以实现多门的线与逻辑关系外, 还可用于直接驱动较大电流的负载。例如在图 2.4.12 中, 使用集电开路反相器 7406 驱动指示灯 (12 V, 20mA)。当输入为高电平时, 输出为低电平, 此时灯亮, 否则灯灭。该电路也可用于驱动发光二极管。

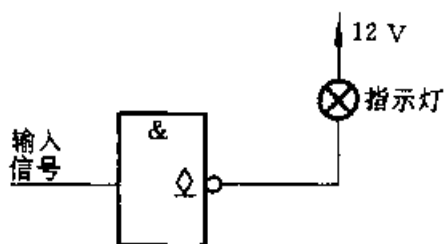


图 2.4.12 用集电极开路反相器驱动一指示灯

3. 三态与非门(TSL^①)

利用 OC 门虽然可以实现线与的功能, 但外接电阻 R_p 的选择要受到一定的限制而不能取得太小, 因此影响了工作速度。同时它省去了有源负载, 使得带负载能力下降。为保持推拉式输出级的优点, 还能作线与联接, 人们又开发了一种三态与非门, 它的输出除了具有一般与非门的

① TSL 是英文 Tristate Logic 的缩写。

两种状态,即输出电阻较小的高、低电平状态外,还具有高输出电阻的第三状态,称为**高阻态**,又称为**禁止态**。

一个简单的 TSL 门的电路如图 2.4.13a 所示,图 b 是它的代表符号。其中 CS 为片选信号输入端^①,A、B 为数据输入端。

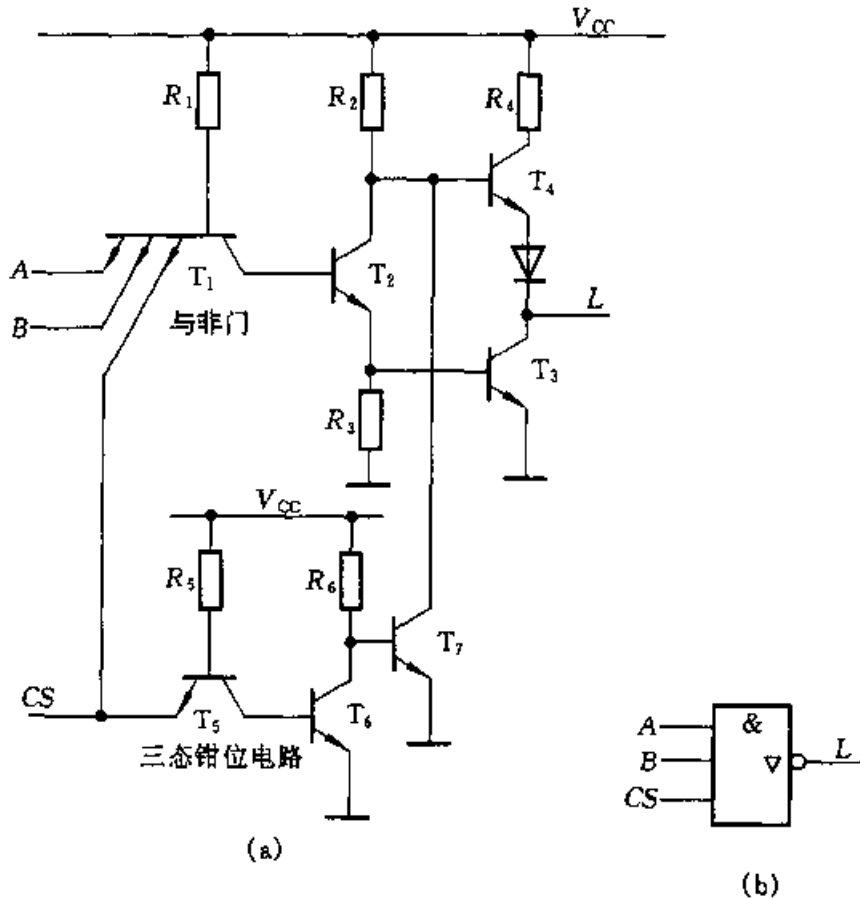


图 2.4.13 三态与非门电路

(a) 电路图 (b) 代表符号

当 $CS = 1$ 时, TSL 门电路中的 T_5 处于倒置放大状态, T_6 饱和, T_7 截止, 即其集电极相当于开路。此时输出状态将完全取决于数据输入端 A 、 B 的状态, 电路输出与输入的逻辑关系与一般与非门相同。这种状态称为 TSL 的工作状态。但当 $CS = 0$ 时, T_7 导通, 使 T_4 的基极钳制于低电平。同时由于低电平的信号送到 T_1 的输入端, 迫使 T_2 和 T_3 截止。这样 T_3 和 T_4 均截止, 门的输出端 L 出现开路, 既不是低电平, 又不是高电平, 这就是第三工作状态。这样, 当 CS 为高电平时, TSL 门的输出信号送到总线, 而当 CS 为低电平时, 门的输出与数据总线断开, 此时数据总线的状态由其他门电路的输出所决定。

为了实现门电路的线与逻辑关系, 本节采取了若干技术措施来解决, 如

^① CS 是 Chip Select 的缩写。

TTL或非门、集电极开路门和三态门,它们在实际的数字电路或系统设计中均将得到应用。

表 2.4.1 TSL 门的真值表

CS	数据输入端		输出端 L
	A	B	
1	0	0	1
	0	1	1
	1	0	1
	1	1	0
0	×	×	高阻

2.4.7 改进型 TTL 门电路——抗饱和 TTL 电路

抗饱和 TTL 电路是目前传输速度较高的一类 TTL 电路。这种电路由于采用肖特基势垒二极管 SBD^①钳位方法来达到抗饱和的效果,一般称为 SBD TTL 电路(简称 STTL 电路),其传输速度远比图 2.4.5 所示的基本 TTL 电路为高。

肖特基势垒二极管是一种利用金属和半导体相接触在界面形成势垒的二极管。利用金属铝和 N 型硅半导体相接触形成的势垒二极管的工作特点如下:

(1) 它和 PN 结一样,同样具有单向导电性,这种铝-硅势垒二极管(Al-SiSBD)导通电流的方向是从铝到硅。

(2) Al-SiSBD 的导通阈值电压较低,约为 0.4~0.5 V,比普通硅 PN 结约低 0.2 V。

(3) 势垒二极管的导电机构是多数载流子,因而电荷存储效应很小。

采用肖特基二极管为什么能提高 TTL 电路的速度呢?大家知道,BJT 工作在饱和时,发射结和集电结都处在正向偏置,集电结正向偏置电压越大,则表明饱和程度越深。

为了限制 BJT 的饱和深度,在 BJT 的基极和集电极并联上一个导通阈值电压较低的肖特基二极管,如图 2.4.14a 所示。通常把它们看成一个器件,并用图 2.4.14b 的符号表示。当 BJT 集电结的正向偏压刚要达到 SBD 的导通阈值电压时,这个二极管首先导通,使集电结正向偏压钳制在 0.4 V 左右,如果流向基极的电流增大,企图使集电结正向偏压加大时,则一部分电流就会通过肖特基二极管直接流向集电极,而不会使 BJT 基极电流过大,因此,肖特基二极管起了抵抗 BJT 过饱和的作用,因而这种电路就称为抗饱和电路,它能使电路的开关时

① SBD 是 Schottky-Barrier-Diode 的缩写。

间大为缩短。

图 2.4.15 为肖特基 TTL (STTL) 与非门的典型电路。与图 2.4.5 所示的基本 TTL 与非门电路相比,作了若干改进。在基本的 TTL 电路中, T_1 、 T_2 和 T_3 工作在深度饱和区,管内电荷存储效应对电路的开关速度影响很大。现在除 T_4 外,其余的 BJT 均采用 SBD 钳位,以达到明显的抗饱和效果。其次,基本电路中的所有电阻值这里几乎都减半。这两项改进导致门电路的开关时间大为缩短。由于电阻值的减小也必然会引起门电路功耗的增加。

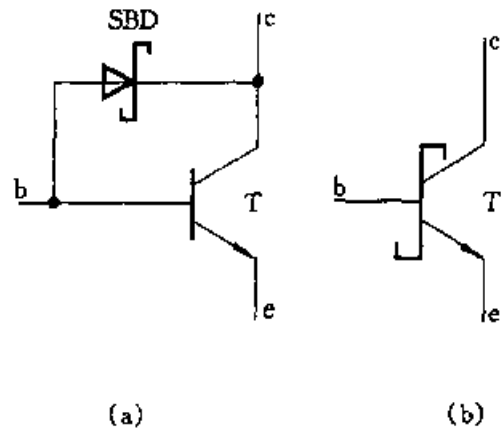


图 2.4.14 带有肖特基二极管钳位的 BJT
(a) 电路连接方式 (b) 代表符号

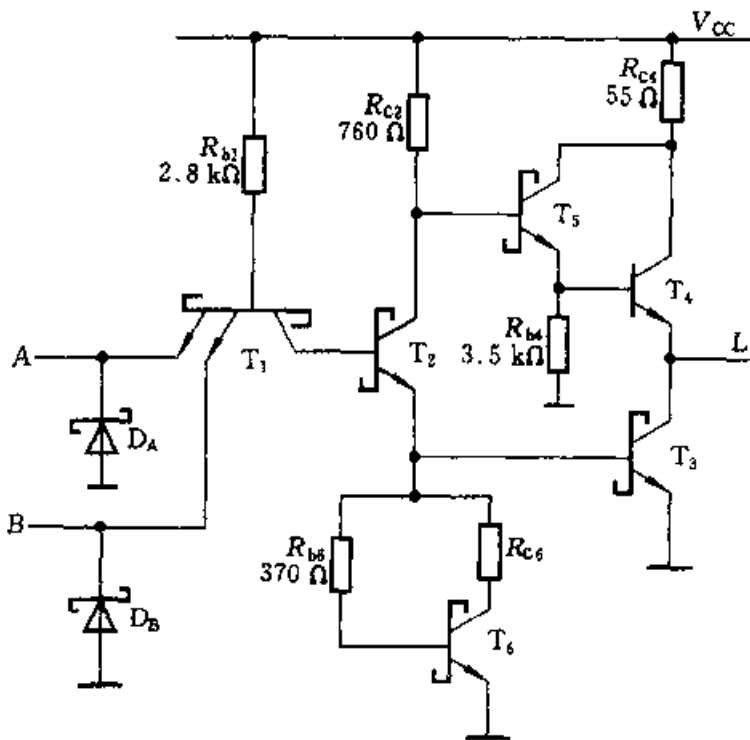


图 2.4.15 肖特基 TTL 与非门的典型电路

STTL 门电路还有以下三点对基本 TTL 电路(图 2.4.5)的性能作了改进:

(1) 二极管 D 被由 T_4 和 T_5 所组成的复合管所代替,当输出由低电平向高电平过渡时,由于复合管电路的电流增益很大,输出电阻很小,从而减小了电路对负载电容的充电时间。

(2) 电路输入端所加的 SBD D_A 和 D_B ,用来减小由门电路之间的连线而引起的杂散信号。

(3) 基本电路(图 2.4.5)中的 R_{e2} ($1\text{ k}\Omega$) 为由 T_6 与 R_{c6} 、 R_{b6} 的组合电路所代替。这个组合电路是有源非线性电阻: 当其两端的电压(发射极 e_2 对地)较低时, 呈现很大的电阻, 而当其两端的电压达到 0.7 V 左右时, 则呈现很小的电阻。这样, 当与非门的全部输入端由低电平转向高电平时, 有源电阻开始不导通, 使 T_3 很快达到饱和; 反之, 当电路的全部输入端(或其中之一)由高电平转向低电平时, T_2 和 T_3 将截止, 由于 T_3 饱和时, $V_{BE3} \approx 0.7\text{ V}$, 在转换开始的瞬间, 有源电阻的阻值很小, T_3 基区存储的电荷通过此低阻回路很快消散。由于这个缘故, 有源非线性电路称为有源下拉电路^① 它与有源上拉电路是对应的。意即将 V_{BE3} 从 0.7 V 很快拉到 0 V , 从而使输出电压很快升高, 即提高了开关速度。

基于上述特点, STTL 与非门具有较为理想的传输特性。与图 2.4.3 的 TTL 反相器的传输特性相比, C 点不再存在了, 由 B 点直接下降到 D 点, 即传输特性变化非常陡峭[2]。

除典型的肖特基型(STTL)外, 尚有低功耗肖特基型(LSTTL)、先进的肖特基型(ASTTL)、先进的低功耗型(ALSTTL)等, 它们的技术参数各有特点, 是在 TTL 工艺的发展过程中逐步形成的。

以上所讨论的为基本 TTL 反相器、与非门和 STTL 与非门。表 2.4.2 列出了各类 TTL 门电路系列的传输延迟时间 t_{pd} 和功耗 P_D , 以资比较。

表 2.4.2 TTL 门电路的各种系列的性能比较

参数 \ 类型	基本的 TTL (74 系列)	肖特基 TTL (74S 系列)	低功耗肖特基 TTL(74LS 系列)	先进的肖特基 TTL(74AS 系列)	先进的低功耗肖特基 TTL(74ALS 系列)
t_{pd}/ns	10	3	9	1.5	4
P_D/mW	10	20	2	20	1
DP/pJ	100	60	18	30	4

稍详细的技术参数见附录 C

复习思考题

2.4.1 一般的 BJT 反相器的动态性能存在什么问题? 而 TTL 反相器的输入级和推挽式输出级各有什么特点, 它们是怎样提高电路的开关速度的?

2.4.2 TTL 反相器的传输特性说明什么问题? 从特性上可以得到 V_{OH} 、 V_{OL} 、 V_{IL} 和 V_{IH} 等 4 个参数, 这些参数代表什么意义?

2.4.3 TTL 与非门和 TTL 反相器在电路结构上和功能上有何不同?

^① 有源下拉电路系 active pull-down circuit 的译称, 它与有源上拉电路是对应的。

2.4.4 TTL 门电路中为了实现线与非的逻辑功能可以采用或非门、集电极开路门和三态门,试说明其原理,它们各用于何种场合?

2.4.5 抗饱和 TTL 电路为什么可以提高开关速度?

* 2.5 射极耦合逻辑门电路

由于 TTL 门中 BJT 工作在饱和状态,开关速度受到了限制。只有改变电路的工作方式,从饱和型变为非饱和型,才能从根本上提高速度。ECL 门就是一种非饱和型高速数字集成电路,它的平均传输延迟时间可在 2 ns 以下,是目前双极型电路中速度最高的。

1. ECL 门的基本结构

图 2.5.1 是最简单的 ECL 门,硅 BJT T_1 、 T_2 、 T_3 组成发射极耦合电路。 T_3 的基极接一个固定的参考电压 V_{REF} ,输入信号接到 T_1 、 T_2 的基极。输出信号由 T_1 、 T_2 或 T_3 的集电极取得。

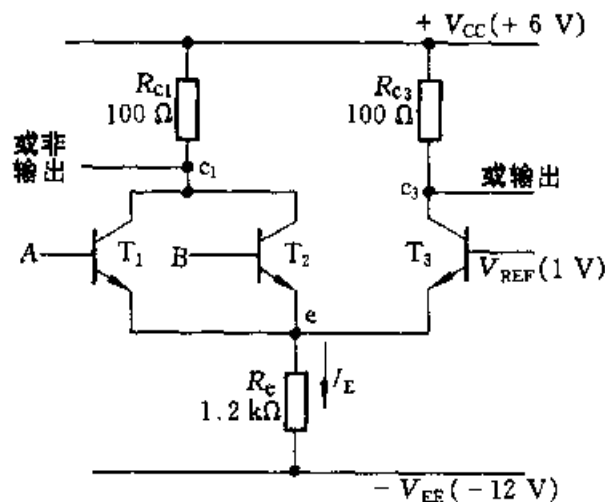


图 2.5.1 ECL 门基本电路

(1) 当输入端 A、B 都接低电平 0 (设 $V_A = V_B = 0.5 \text{ V}$) 时

由于 $V_{REF} = 1 \text{ V}$, 因此 T_3 优先导通, 这就使发射极 e 的电位 $V_E = V_{REF} - V_{BE3} = 0.3 \text{ V}$ 。对于 T_1 、 T_2 来说, 由于 $V_E = 0.3 \text{ V}$, 而 $V_A = V_B = 0.5 \text{ V}$, 虽然基极电位比发射极电位高 0.2 V, 但因为是硅管, 仍可保证截止。这时流过 R_c 的电流将全部由 T_3 提供, 且有 $I_E = [V_E - (-V_{EE})] / R_c = (0.3 \text{ V} + 12 \text{ V}) / 1.2 \text{ k}\Omega \approx 10 \text{ mA}$ 。这样

$$V_{c3} = V_{CC} - I_E R_{c3} = 6 \text{ V} - 10 \text{ mA} \times 0.1 \text{ k}\Omega = 5 \text{ V}$$

而 $V_{c1} = V_{CC} = 6 \text{ V}$

由此可见, 当输入为 0 时, T_1 、 T_2 截止, 输出端 c_1 为高电平 1 (+6 V); T_3 导

通,输出端 c_3 为低电平0(+5 V)。而且由于 $V_{B3} = V_{REF} = 1$ V,而 $V_{C3} = 5$ V,所以 T_3 处于放大状态而未达到饱和。

(2) 当输入端 A、B 中有一个接高电平1(设 A 接高电平, $V_A = 1.5$ V)时

由于 $V_A > V_{REF}$,所以 T_1 优先导通,这就使 $V_E = 1.5$ V - 0.7 V = 0.8 V,对 T_3 来说,这时基极电位比发射极电位仅高 0.2 V,可以保证 T_3 截止。流过 R_c 的电流由 T_1 提供,且有 $I_E = (0.8$ V + 12 V)/1.2 k $\Omega = 10.6$ mA,而

$$V_{C1} = V_{CC} - I_E R_{c1} = 6$$
 V - 10.6 mA \times 0.1 k $\Omega \approx 5$ V

$$V_{C3} = V_{CC} = 6$$
 V

此时 T_1 处于放大状态。由于 T_1 和 T_2 的发射极和集电极是分别连在一起的,所以只要 A、B 中有一个接高电平,都会使 c_1 为低电平0(+5 V),而 c_3 为高电平1(+6 V)。

由上分析可得 $c_1 = \overline{A+B}$ 或非输出

$$c_3 = A+B \quad \text{或输出}$$

即 ECL 门的基本逻辑功能是同时具备或非/或输出,称之为互补逻辑输出。

同时,不论是哪个 BJT 导通,所形成的发射极电流 I_E 都是很接近的。这个电流受输入信号控制,分别流入 T_1 或 T_2 或 T_3 ,就像一个开关在控制,所以 ECL 电路又称为电流开关型电路(CML)^①。

以上所述的是具有 A、B 两个输入端的或非电路,只要增加相同类型的 BJT 与 T_1 并联,就能增加门电路的输入端数。

图 2.5.1 所示电路中存在的问题是,输出端的高、低电平与输入端的高、低电平电压不一致(尽管摆幅相等)。实用上,可采用加电压跟随器等移动电平值的措施来解决。

2. ECL 门的实际电路

图 2.5.2 是 ECL 门的电路实例。由于集成电路的特点,本电路只用一种负电源 $-V_{EE} = -5.2$ V,而 $V_{CC} = 0$ V。图中 $T_1 \sim T_4$ 组成多端输入,并与 T_5 组成射极耦合电路。 T_6 组成一个简单的电压跟随器,它为 T_5 提供一个参考电压 V_{REF} 。为了补偿温度漂移,在 T_6 的基极回路接入了两个二极管。

图中 T_7 和 T_8 组成电压跟随器,起电平移动作用, V_{C4} 和 V_{C3} 通过电压跟随器后,使输出变为标准的 ECL 电平。其典型值是:高低电平的电压分别为 -0.9 V 和 -1.75 V。同时由于有了这两个电压跟随器作为输出级,也有效地提高了 ECL 门的带负载能力。

^① CML 是 Current Mode Logic 的缩写。

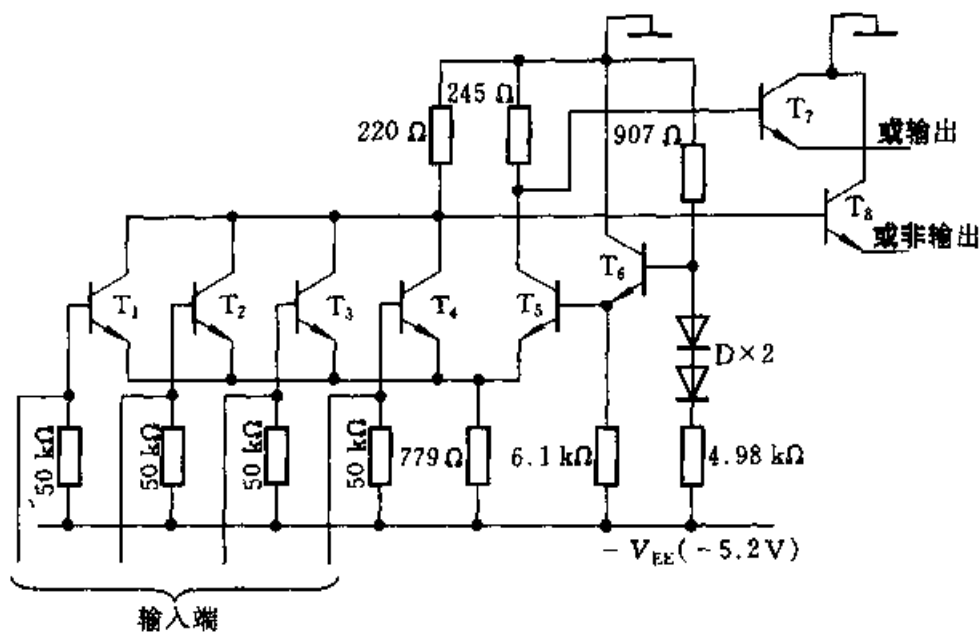


图 2.5.2 实际的 ECL 门电路

3. ECL 门的工作特点

(1) BJT 工作在截止区或放大区, 集电极电位总高于基极电位, 这就避免了 BJT 因工作在饱和状态而产生的存储电荷问题。

(2) 逻辑电平的电压摆幅小, 输入电压变化 $\Delta V_i \approx 1 \text{ V}$ ($-1.85 \sim -0.81 \text{ V}$), 集电极输出电压变化 $\Delta V_o \approx 0.85 \text{ V}$ ($-1.75 \sim -0.9 \text{ V}$), 高低电平的电压差值已经小到只能区分 BJT 的导通和截止两种状态。集电极输出电压的变化小, 这不仅有利于电路的转换, 而且可采用很小的集电极电阻 R_c 。因此, ECL 门的负载电阻总是在几百欧的数量级, 使输出回路的时间常数比一般饱和型电路小, 有利于提高开关速度。

ECL 门的速度快, 常用于高速系统中。它的主要缺点是制造工艺要求高, 功耗大, 抗干扰能力较弱。而且由于输出电压为负值, 若与其他门电路接口, 需用专门的电平移动电路。

双极型逻辑门电路除了 TTL 和 ECL 之外, 尚有集成注入逻辑门电路 (IIL^① 或 I^2L) 和高阈值逻辑门电路 (HTL^②)。IIL 电路由于它的电路简单, 易于在硅片上实现高集成度的器件, 因而在大规模和超大规模集成电路中得到应用, 但不制成单个集成门电路。由于它的高、低电平电压差值很小, 抗干扰能力较差, 因而这种门电路的推广受到限制。至于 HTL 电路, 虽然它有较强的抗干扰能力, 但它的功耗大, 开关速度也不高, 已不生产, 为下面要讨论的 CMOS 电路所取代。

① IIL 系 Integrated-Injection Logic 的缩写。

② HTL 系 High-Threshold Logic 的缩写。

复习思考题

2.5.1 ECL 门电路为什么具有很高的开关速度? 该类电路有什么特点?

2.6 CMOS 逻辑门电路

CMOS 逻辑门电路是在 TTL 电路问世之后,所开发出的第一种广泛应用的数字集成器件,从发展趋势来看,由于制造工艺的改进,CMOS 电路的性能有可能超越 TTL 而成为占主导地位的逻辑器件。CMOS 电路的工作速度可与 TTL 相比较,而它的功耗和抗干扰能力则远优于 TTL。此外,几乎所有的超大规模存储器件,以及 PLD 器件都采用 CMOS 工艺制造,且费用较低。

早期生产的 CMOS 门电路为 4000 系列,随后发展为 4000 B 系列。当前与 TTL 兼容的 CMOS 器件如 74HCT 系列等可与 TTL 器件交换使用。下面首先讨论 CMOS 反相器,然后介绍其他 CMOS 逻辑门电路。

2.6.1 CMOS 反相器

由本书模拟部分已知,MOSFET 有 P 沟道和 N 沟道两种,每种中又有耗尽型和增强型两类。由 N 沟道和 P 沟道两种 MOSFET 组成的电路称为互补 MOS 或 CMOS 电路。

图 2.6.1a 表示 CMOS 反相器电路,由两只增强型 MOSFET 组成,其中一个为 N 沟道结构,另一个为 P 沟道结构。图 2.6.1b 为在本书中使用的简化电路。为了电路能正常工作,要求电源电压 V_{DD} 大于两个管子的开启电压的绝对值之和,即 $V_{DD} > (|V_{IN}| + |V_{IP}|)$ 。

1. 工作原理

首先考虑两种极限情况:当 v_i 处于逻辑 0 时,相应的电压近似为 0 V;而当 v_i 处于逻辑 1 时,相应的电压近似为 V_{DD} 。假设在两种情况下,N 沟道管 T_N 为工作管,P 沟道管 T_P 为负载管。但是,由于电路是互补对称的,这种假设可以是任意的,相反的情况亦将导致相同的结果。

图 2.6.2 分析了当 $v_i = V_{DD}$ 时的工作情况。参看图 2.6.2b。在 T_N 的输出特性 $i_D - v_{DS}(v_{GSN} = V_{DD})$ (注意 $v_{DSN} = v_O$) 上,叠加一条负载线,它是负载管 T_P 在 $v_{SGP} = 0$ V 时的输出特性 $i_D - v_{SD}$ 。由于 $v_{SGP} < V_T$ ($V_{TN} = |V_{TP}| = V_T$),负载曲线几乎是一条与横轴重合的水平线。两条曲线的交点即工作点。显然,这时的输出电压 $v_{OL} \approx 0$ V (典型值 < 10 mV),而通过两管的电流接近于零。这就是说,电路的功耗很小(微瓦数量级)。

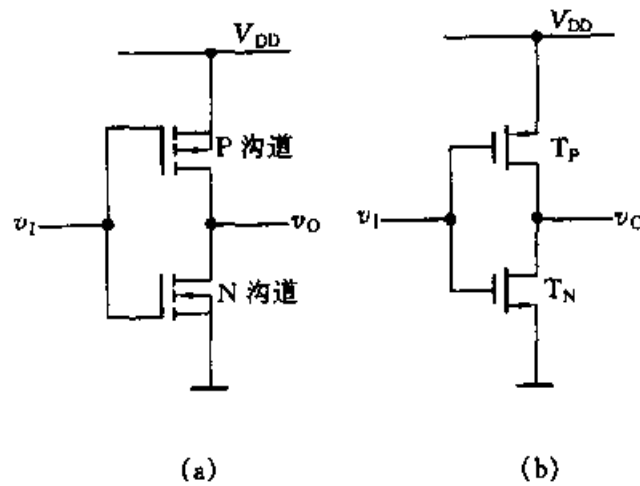


图 2.6.1 CMOS 反相器
(a) 电路 (b) 本书中简化电路

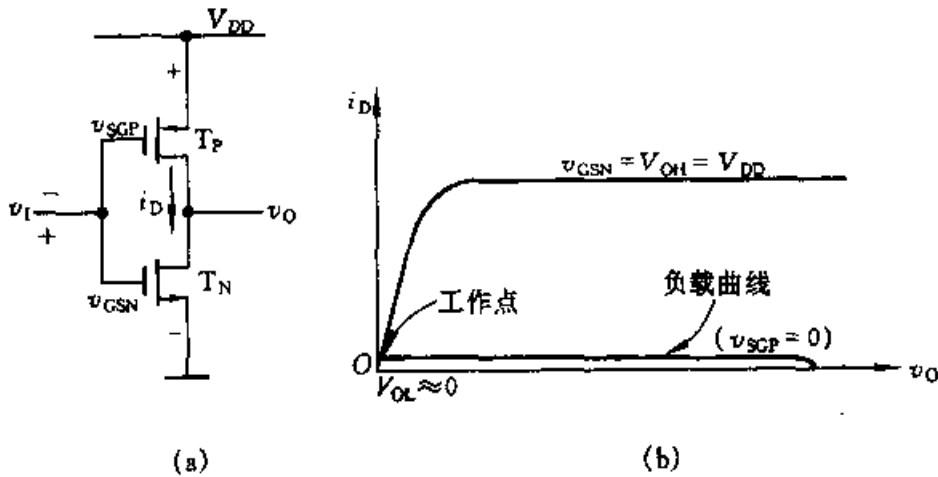


图 2.6.2 CMOS 反相器在输入为高电平时的图解分析
(a) 电路 (b) 图解

图 2.6.3 分析了另一种极限情况,此时对应于 $v_i = 0 \text{ V}$,其工作状态示于图 2.6.3b 中。此时工作管 T_N 在 $v_{GSN} = 0$ 的情况下运用,其输出特性 $i_D - v_{DS}$ 几乎与横轴重合,负载曲线是负载管 T_P 在 $v_{SGP} = V_{DD}$ 时的输出特性 $i_D - v_{DS}$ 。由图可知,工作点决定了 $v_o = V_{OH} \approx V_{DD}$;通过两器件的电流接近零值。可见上述两种极限情况下的功耗都很低。

由此可知,基本 CMOS 反相器近似于一理想的逻辑单元,其输出电压接近于零或 $+V_{DD}$,而功耗几乎为零。

2. 传输特性

CMOS 反相器的传输特性可仿照前述两种极限情况的图解步骤来求得,改变 v_i 的值,可得出相应的 v_o 值。图 2.6.4 表示 CMOS 反相器的典型传输特

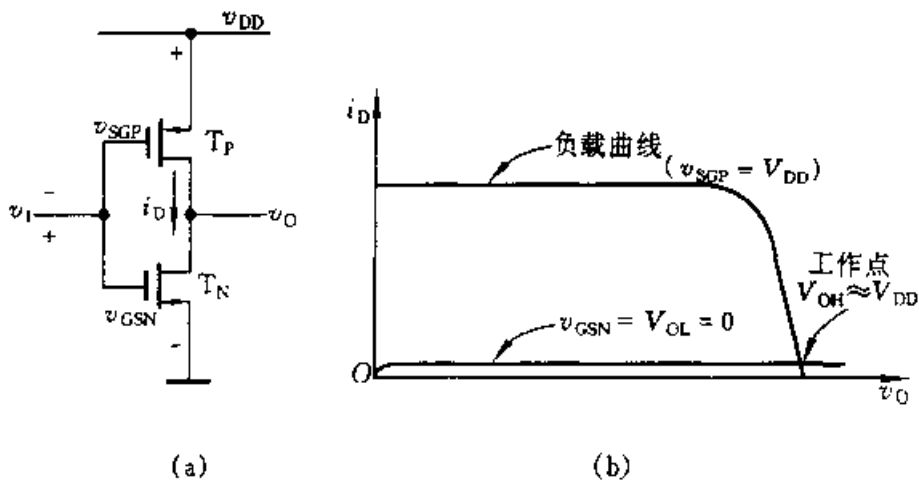


图 2.6.3 CMOS反相器在输入为低电平时的图解分析

(a) 电路 (b) 图解

性。图中 $V_{DD} = 10\text{ V}$, $V_{TN} = |V_{TP}| = V_T = 2\text{ V}$ 。由于 $V_{DD} > (V_{TN} + |V_{TP}|)$, 因此, 当 $V_{DD} - |V_{TP}| > v_I > V_{TN}$ 时, T_N 和 T_P 两管同时导通。考虑到电路是互补对称的, 一器件可将另一器件视为它的漏极负载。还应注意到, 器件在放大区(饱和区)呈现恒流特性, 两器件之一可当作高阻值的负载。因此, 在过渡区域, 传输特性变化比较急剧。两管在 $v_I = V_{DD}/2$ 处转换状态。

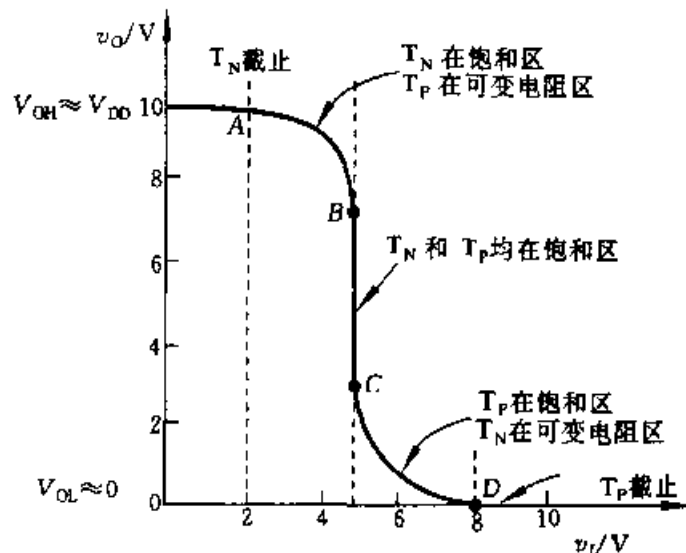


图 2.6.4 CMOS反相器的传输特性

3. 工作速度

CMOS反相器在电容负载情况下, 它的开通时间与关闭时间是相等的, 这是因为电路具有互补对称的性质。图 2.6.5 表示当 $v_I = 0\text{ V}$ 时, T_N 截止, T_P 导通, 由 V_{DD} 通过 T_P 向负载电容 C_L 充电的情况。由于 CMOS 反相器中, 两管的

g_m 值均设计得较大,其导通电阻较小,充电回路的时间常数较小。类似地,亦可分析电容 C_L 的放电过程。CMOS 反相器的平均传输延迟时间约为 10 ns。

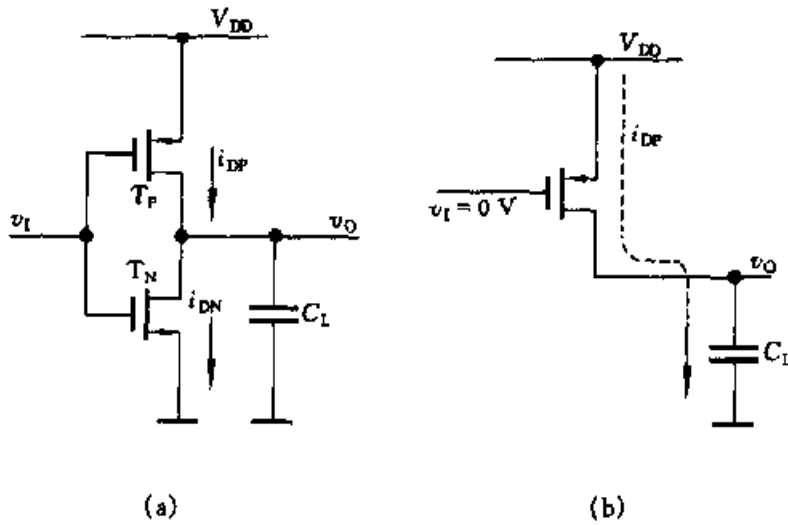


图 2.6.5 CMOS 反相器在电容负载下的工作情况
(a) 电路 (b) 负载电容充电

2.6.2 CMOS 门电路

1. 与非门电路

图 2.6.6 是 2 输入端 CMOS 与非门电路,其中包括两个串联的 N 沟道增强型 MOS 管和两个并联的 P 沟道增强型 MOS 管。每个输入端连到一个 N 沟道和一个 P 沟道 MOS 管的栅极。当输入端 A、B 中只要有一个为低电平时,就会使与它相连的 NMOS 管截止,与它相连的 PMOS 管导通,输出为高电平;仅当 A、B 全为高电平时,才会使两个串联的 NMOS 管都导通,使两个并联的 PMOS 管都截止,输出为低电平。

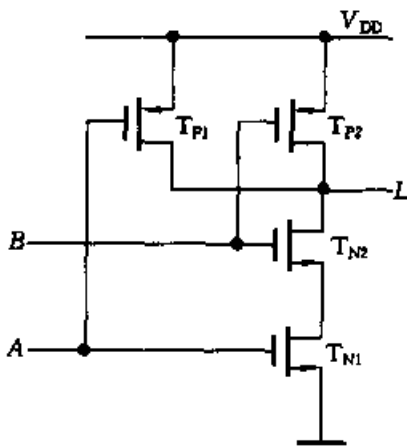


图 2.6.6 CMOS 与非门

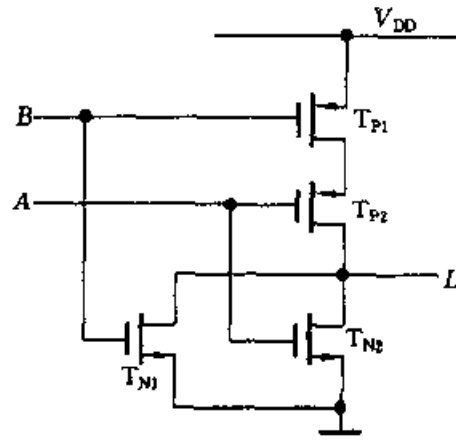


图 2.6.7 CMOS 或非门

因此,这种电路具有与非的逻辑功能,即

$$L = \overline{A \cdot B}$$

n 个输入端的与非门必须有 n 个 NMOS 管串联和 n 个 PMOS 管并联。

2. 或非门电路

图 2.6.7 是 2 输入端 CMOS 或非门电路。其中包括两个并联的 N 沟道增强型 MOS 管和两个串联的 P 沟道增强型 MOS 管。

当输入端 A 、 B 中只要有一个为高电平时,就会使与它相连的 NMOS 管导通,与它相连的 PMOS 管截止,输出为低电平;仅当 A 、 B 全为低电平时,两个并联 NMOS 管都截止,两个串联的 PMOS 管都导通,输出为高电平。

因此,这种电路具有或非的逻辑功能,其逻辑表达式为

$$L = \overline{A + B}$$

显然, n 个输入端的或非门必须有 n 个 NMOS 管并联和 n 个 PMOS 管并联。

比较 CMOS 与非门(图 2.6.6)和或非门(图 2.6.7)可知,与非门的工作管是彼此串联的,其输出电压随管子个数的增加而增加;或非门则相反,工作管彼此并联,对输出电压不致有明显的影晌。因而或非门用得较多。

3. 异或门电路

CMOS 异或门电路如图 2.6.8 所示。它由一级或非门和一级与或非门组成。或非门的输出 $X = \overline{A + B}$ 。而与或非门的输出 L 即为输入 A 、 B 的异或

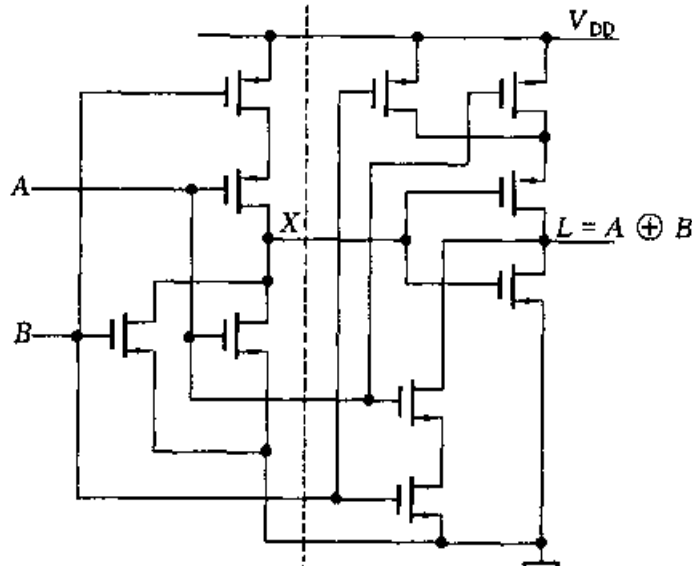


图 2.6.8 异或门电路

$$L = \overline{A \cdot B + X} = \overline{A \cdot B + \overline{A + B}} = \overline{A \cdot B + \overline{A} \cdot \overline{B}} = \overline{A \odot B} = A \overline{B} + \overline{A} B = A \oplus B$$

如在异或门的后面增加一级反相器就构成异或非门,由于具有 $\overline{L} = A \cdot B + \overline{A} \cdot \overline{B}$ 的功能,因而称为同或门。异或门和同或门的逻辑符号如图 2.6.9 所示。

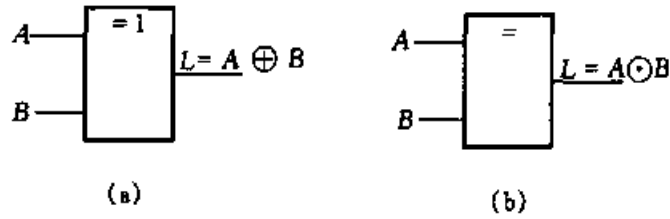


图 2.6.9 异或门和同或门的逻辑符号

(a) 异或门 (b) 同或门

CMOS 逻辑门电路中,除上述几种外,还有三态门、缓冲器等类型,此处从略。

2.6.3 BiCMOS 门电路

双极型 CMOS 或 BiCMOS 的特点在于,利用了双极型器件的速度快和 MOSFET 的功耗低两方面的优势,因而这种逻辑门电路受到用户的重视。

1. BiCMOS 反相器

图 2.6.10 表示基本的 BiCMOS 反相器电路,为了清楚起见, MOSFET 用符号 M 表示, BJT 用 T 表示。 T_1 和 T_2 构成推拉式输出级。而 M_p 、 M_N 、 M_1 、 M_2 所组成的输入级与基本的 CMOS 反相器很相似。输入信号 v_i 同时作用于 M_p 和 M_N 的栅极。当 v_i 为高电压时, M_N 导通而 M_p 截止;而当 v_i 为低电压时,情况则相反, M_p 导通, M_N 截止。当输出端接有同类 BiCMOS 门电路时,输出级能提供足够大的电流为电容性负载充电。同理,已充电的电容负载也能迅速地通过 T_2 放电。

上述电路中, T_1 和 T_2 的基区存储电荷亦可通过 M_1 和 M_2 释放,以加快电路的开关速度。当 v_i 为高电压时 M_1 导通, T_1 基区的存储电荷迅速消散。这种作用与 TTL 门电路的输入级中 T_1 类似。同理,当 v_i 为低电压时,电源电压 V_{DD} 通过 M_p 以激励 M_2 ,使 M_2 导通,显然, T_2 基区的存储电荷通过 M_2 而消散。可见,门电路的开关速度可得到改善。

2. BiCMOS 门电路

根据前述的 CMOS 门电路的结构和工作原理,同样可以用 BiCMOS 技术实现或非门和与非门。如果要实现或非逻辑关系,输入信号用来驱动并联的 N 沟道 MOSFET,而 P 沟道 MOSFET 则彼此串联。这一思路可用图 2.6.11 所示的 2 输入端或非门来说明。若两输入端 A 和 B 均为低电平时,则两个 MOSFET M_{pA} 和 M_{pB} 均导通, T_1 导通而 M_{nA} 和 M_{nB} 均截止,输出 L 为高电平。与此同时, M_2 通过 M_{pA} 和 M_{pB} 被 V_{DD} 所激励,从而为 T_2 的基区存储电荷提供一条释放通路。

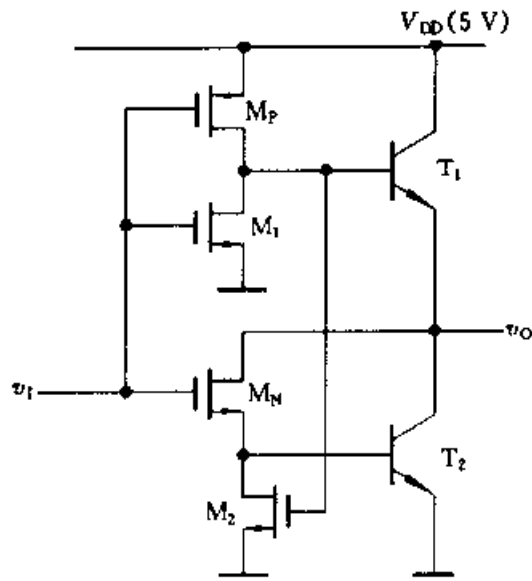


图 2.6.10 基本的 BiCMOS 反相器电路

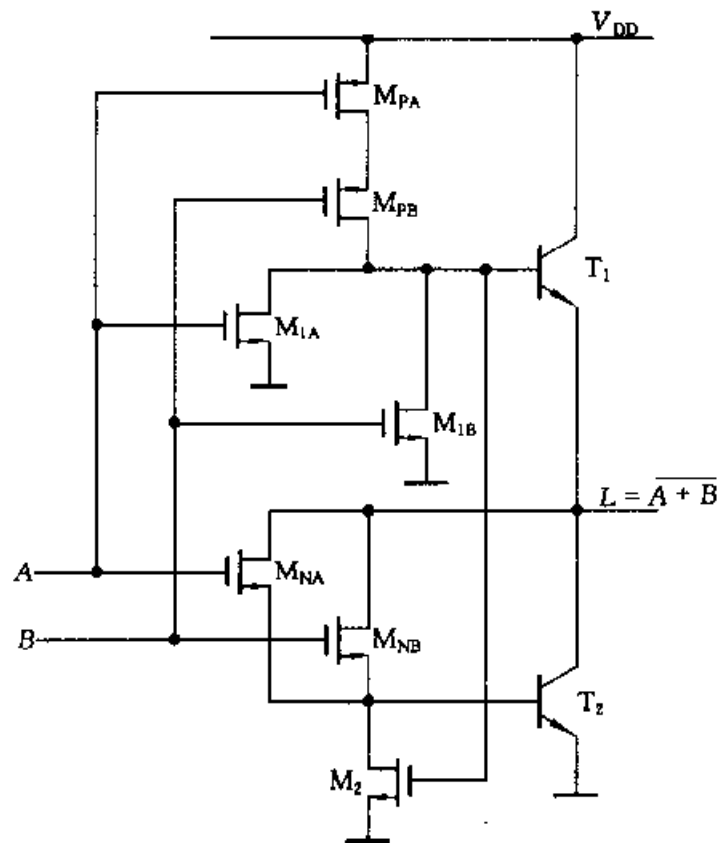


图 2.6.11 2 输入端或非门电路

另一方面,当两输入端 A 和 B 中之一为高电平时,则 M_{PA} 和 M_{PB} 的通路被断开,并且 M_{NA} 或 M_{NB} 导通,将使输出端为低电平。同时, M_{IA} 或 M_{IB} 为 T_1 的基极存储电荷提供一条释放道路。由此可知,在图 2.6.11 的逻辑电路中,只要有

一个输入端接高电平,输出即为低电平。

同理,可以构成与非门电路,见习题 2.6.1。

2.6.4 CMOS 传输门

MOSFET 的输出特性在原点附近呈线性对称关系,因而它们常用作模拟开关。模拟开关广泛地用于取样-保持电路、斩波电路、模数和数模转换电路等。下面着重介绍 CMOS 传输门。

所谓传输门(TG)^①就是一种传输模拟信号的模拟开关。CMOS 传输门由一个 P 沟道和一个 N 沟道增强型 MOSFET 并联而成,如图 2.6.12a 所示。图 2.6.12b 是它的代表符号。 T_P 和 T_N 是结构对称的器件,它们的漏极和源极是可互换的。设它们的开启电压 $|V_T| = 2\text{ V}$,且输入模拟信号的变化范围为 -5 V 到 $+5\text{ V}$ 。为使衬底与漏源极之间的 PN 结任何时刻都不致正偏,故 T_P 的衬底接 $+5\text{ V}$ 电压,而 T_N 的衬底接 -5 V 电压。两管的栅极由互补的信号电压 ($+5\text{ V}$ 和 -5 V) 来控制,分别用 C 和 \bar{C} 表示。

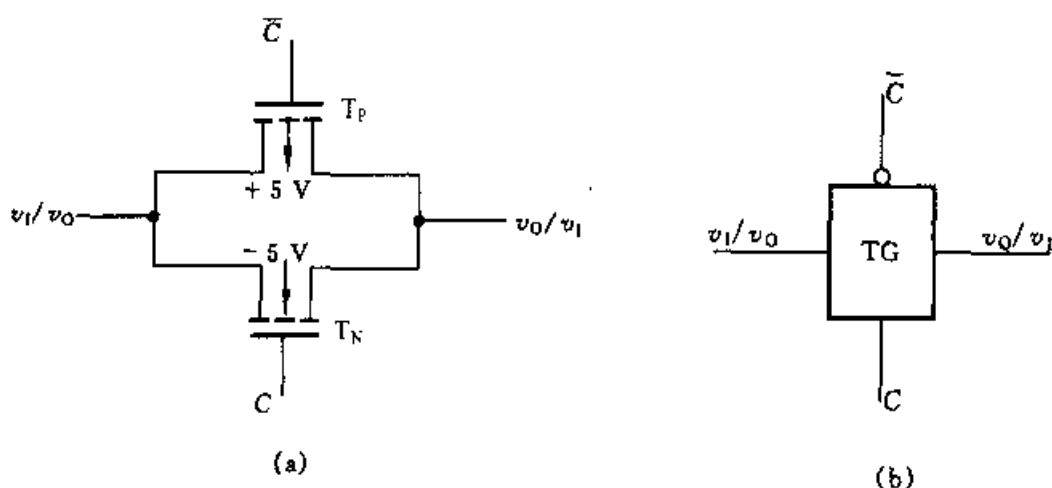


图 2.6.12 CMOS 传输门

(a) 电路 (b) 代表符号

传输门的工作情况如下:当 C 端接低电压 -5 V 时, T_N 的栅压即为 -5 V , v_1 取 -5 V 到 $+5\text{ V}$ 范围内的任意值时, T_N 均不导通。同时, T_P 的栅压为 $+5\text{ V}$, T_P 亦不导通。可见,当 C 端接低电压时,开关是断开的。

为使开关接通,可将 C 端接高电压 $+5\text{ V}$ 。此时 T_N 的栅压为 $+5\text{ V}$, v_1 在 -5 V 到 $+3\text{ V}$ 的范围内, T_N 导通。同时, T_P 的栅压为 -5 V , v_1 在 -3 V 到 $+5\text{ V}$ 的范围内, T_P 将导通。

^① TG 是 Transmission Gate 的缩写。

由上分析可知,当 $v_i < -3\text{ V}$ 时,仅有 T_N 导通,而当 $v_i > +3\text{ V}$ 时,仅有 T_P 导通。当 v_i 在 -3 V 到 $+3\text{ V}$ 的范围内, T_N 和 T_P 两管均导通。进一步分析还可看到,一管导通的程度愈深,另一管的导通程度则相应地减小。换句话说,当一管的导通电阻减小,则另一管的导通电阻就增加。由于两管系并联运行,可近似地认为开关的导通电阻近似为一常数。这是 CMOS 传输出门的优点。

在正常工作时,模拟开关的导通电阻值约为数百欧,当它与输入阻抗为兆欧级的运放串接时,可以忽略不计。

CMOS 传输门除了作为传输模拟信号的开关之外,也可作为各种逻辑电路的基本单元电路,见参考文献[9]、[11]。

2.6.5 CMOS 逻辑门电路的技术参数

CMOS 逻辑集成器件从本世纪 60 年代末发展至今,由于制造工艺的不断完善,它的技术参数从总体上来说已经达到或者接近 TTL 器件的水平,其中某些参数优于 TTL 器件。例如,CMOS 器件的功耗低、扇出数大,噪声容限亦大,这些均是由于 CMOS 器件的固有特性所决定的。但也应注意到,这里讲的功耗低是指静态功耗(微瓦量级)而言。实际上,因为它的输入电容约为 10 pF ,当工作频率较高时,其动态功耗随频率的增加而增加,其值可达 LST-TL 量级。

最早的 CMOS 器件为 4000 系列(1967 年)。随后出现的有 74C 系列,引脚编号与 TTL 兼容,但因输入和输出电压值仍不兼容,因而这种系列未能得到推广。由于 CMOS 工艺的重大突破,一种新型的高速 CMOS 器件系列诞生,这就是 74HC 系列。这类器件不仅功耗低,而且在相同电源电压 V_{CC} 的条件下,其噪声容限约为 TTL 电路的两倍。它的平均传输延迟时间每门可小到 $6\sim 10\text{ ns}$ 。这种速度与基本的 TTL 或 LSTTL 门电路相当。此外,尚有与 TTL 兼容的新系列,即 74HCT 系列和 74BCT(BiCMOS)系列等。表 2.6.1 列出了上述 CMOS 器件的主要参数,以便于与 TTL 器件系列相比较。

表 2.6.1 CMOS 门电路各系列的性能比较

系列	基本的 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 BiCMOS(74BCT 系列)
$t_{pd}/\text{ns}(C_L = 15\text{ pF})$	75	10	13	2.9
P_{10}/mW	0.002	1.55	1.002	0.0003~7.5
DP/pJ	0.15	15.5	13.026	0.00087~22

* 稍详细的参数见附录 C 或 CMOS 数据手册。

复习思考题

2.6.1 CMOS 逻辑门电路与 TTL 电路相比有哪些优点？为什么说，从发展的观点来看，CMOS 器件的应用有取代 TTL 电路的趋势？

2.6.2 什么叫 BiCMOS 逻辑门电路？试说明 BiCMOS 门电路的优缺点。

2.7 NMOS 逻辑门电路

NMOS 逻辑门电路是全部由 N 沟道 MOSFET 构成。由于这种器件具有较小的几何尺寸，适合于制造大规模集成电路。此外，由于 NMOS 集成电路的结构简单，易于使用 CAD 技术进行设计。与 CMOS 电路类似，NMOS 电路中不使用难于制造的电阻。NMOS 反相器是整个 NMOS 逻辑门电路的基本构件，它的工作管常用增强型器件，而负载管可以是增强型也可以是耗尽型。现以增强型器件作为负载管的 NMOS 反相器为例来说明它的工作原理。

图 2.7.1a 表示 NMOS 反相器的原理电路，其中 T_1 为工作管， T_2 为负载管，二者均属增强型器件。若 T_1 和 T_2 在同一工艺过程中制成，它们必将具有相同的开启电压 V_T 。图 2.7.1b 是它在本书中的简化电路。由图 2.7.1 可见，负载管 T_2 的栅极与漏极同接电源 V_{DD} ，因而 T_2 总是工作在它的恒流区，处于导通状态。

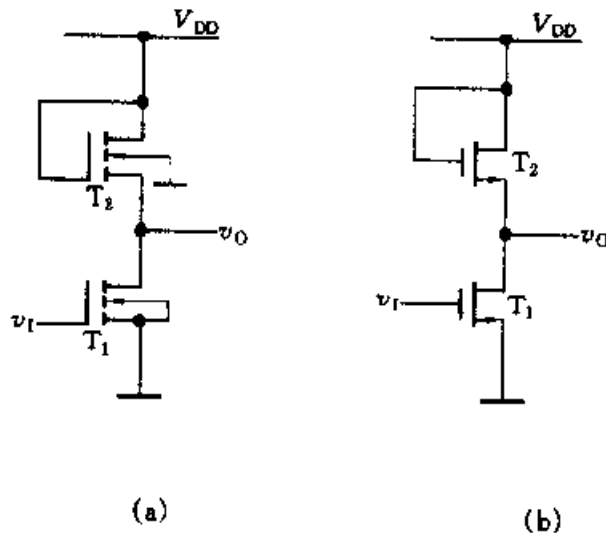


图 2.7.1 NMOS 反相器
(a) 实际电路 (b) 本书中简化电路

当输入 v_1 为高电压(超过管子的开启电压 V_T)时， T_1 导通，输出 v_0 为低电压。输出低电压的值，由 T_1 、 T_2 两管导通时所呈现的电阻值之比决定。通常 T_1

的跨导 g_{m1} 远大于 T_2 管的跨导 g_{m2} , 以保证输出低电压值在 $+1\text{ V}$ 左右^①。当输入电压 v_1 为低电压(低于管子的开启电压 V_T)时, T_1 截止, 输出 v_O 为高电压。由于 T_2 管总是处于导通状态, 因此输出高电压值约为 $(V_{DD} - V_T)$ 。通常 g_{m1} 在 $100 \sim 200\ \mu\text{S}$ 之间, 而 g_{m2} 约为 $5 \sim 15\ \mu\text{S}$ 。 T_1 导通时的等效电阻 $R_{\text{eq}1}$ 约为 $3 \sim 10\ \text{k}\Omega$, 而 T_2 的 $R_{\text{eq}2}$ 约在 $100 \sim 200\ \text{k}\Omega$ 之间。负载管导通电阻是随工作电流而变化的非线性电阻。

在 NMOS 反相器的基础上, 可以制成 NMOS 门电路。图 2.7.2 为 NMOS 或非门电路。只要输入 A 、 B 中任一个为高电平, 与它对应的 MOSFET 导通时, 输出为低电平; 仅当 A 、 B 全为低电平时, 所有工作管都截止时, 输出才为高电平。可见电路具有或非功能, 即

$$L = \overline{A + B}$$

或非门的工作管都是并联的, 增加管子的个数, 输出低电平基本稳定, 在整体电路设计中较为方便, 因而 NMOS 门电路是以或非门为基础的。这种门电路不像 TTL 或 CMOS 电路作成小规模单个芯片, 主要用于大规模集成电路。

以上讨论和分析了各种逻辑门电路的结构、工作原理和性能, 为便于比较, 现用它们的主要技术参数传输延迟时间 t_{pd} 和功耗 P_D 综合描述各种逻辑门电路的性能, 如图 2.7.3 所示。

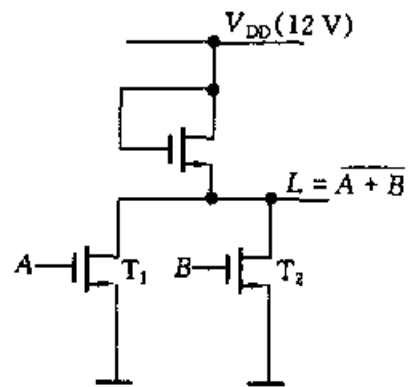


图 2.7.2 NMOS 或非门电路

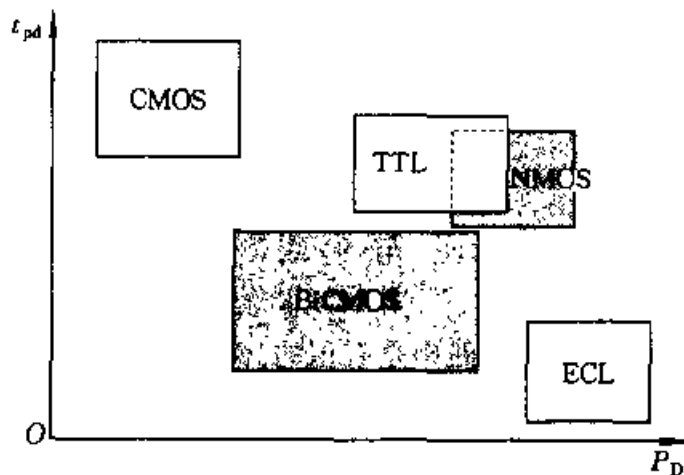


图 2.7.3 各种门电路的延迟时间与功耗的关系图

^① MOSFET 的跨导 g_m 的值取决于管子的几何结构。

2.8 正负逻辑问题

1. 正负逻辑的规定

在逻辑电路中,输入和输出一般都用电平来表示。若用 H 和 L 分别表示高、低电平,则门电路的功能可用表 2.8.1 所示的电平表来描述。但是,这个门体现了什么逻辑关系尚不清楚,因为还未确切说明电平与逻辑状态之间的隶属关系。这种关系可由人们任意地加以规定。如令 $H=1, L=0$,则称之为**正逻辑体制**,于是很容易由表 2.8.1 导出表 2.8.2。显然,后者表示一正逻辑与非门的真值表。与此相反,若令 $H=0, L=1$,则称之为**负逻辑体制**。据此,由本例可得出**负逻辑或非门**的真值表,如表 2.8.3 所示。

表 2.8.1 电平表示例

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

表 2.8.2 正与非门真值表

A	B	L
0	0	1
0	1	1
1	0	1
1	1	0

表 2.8.3 负或非门真值表

A	B	L
1	1	0
1	0	0
0	1	0
0	0	1

对于同一电路,可以采用正逻辑,也可以采用负逻辑。正逻辑和负逻辑两种体制不牵涉到逻辑电路本身的结构问题,但根据所选正负逻辑的不同,即使同一电路也具有不同的逻辑功能。本书如无特殊说明,一律采用正逻辑,即规定高电平为逻辑1,低电平为逻辑0。

2. 正负逻辑的等效变换

一般用正逻辑函数描述电路,在过渡到负逻辑时,只需按下列方式互换各种运算:

与非 \leftrightarrow 或非

与 \leftrightarrow 或

非 \leftrightarrow 非

正负逻辑的概念在工程实践中是有用的,例如,在 2.4.6 节中所讨论的正逻辑 TTL 或非门等效于负逻辑与非门,不过此时的逻辑值(1,0)是按负逻辑规定的。

2.9 逻辑门电路使用中的几个实际问题

以上讨论了几种逻辑门电路特别是重点地讨论了 TTL 和 CMOS 两种电路。在具体的应用中可以根据要求来选用何种器件。器件的主要技术参数有传

输延迟时间、功耗、噪声容限,带负载能力等,据此可以正确地选用一种器件或两种器件混用。下面对几个实际问题,如不同门电路之间的接口技术,门电路与负载之间的匹配等进行讨论。

2.9.1 各种门电路之间的接口问题

在数字电路或系统的设计中,往往由于工作速度或者功耗指标的要求,需要采用多种逻辑器件混合使用,例如,TTL 和 CMOS 两种器件都要使用。由前面几节的讨论已知,每种器件的电压和电流参数各不相同,因而需要采用接口电路,一般需要考虑下面三个条件:

- 驱动器件必须能对负载器件提供灌电流最大值[见式(2.4.14)]。
- 驱动器件必须对负载器件提供足够大的拉电流[见式(2.4.15)]。
- 驱动器件的输出电压必须处在负载器件所要求的输入电压范围,包括高、低电压值。

其中条件 1 和 2,属于门电路的扇出数问题,已在 2.4.5 节作过详细的分析。条件 3 属于电压兼容性的问题。其余如噪声容限、输入和输出电容以及开关速度等参数在某些设计中也必须予以考虑。

下面分别就 CMOS 门驱动 TTL 门或者相反的两种情况的接口问题进行分析。

1. CMOS 门驱动 TTL 门

在这种情况下,只要两者的电压参数兼容,不需另加接口电路,仅按电流大小计算出扇出数即可。

图 2.9.1 表示 CMOS 门驱动 TTL 门的简单电路。当 CMOS 门的输出为高电平时,它为 TTL 负载提供拉电流,反之则提供灌电流。

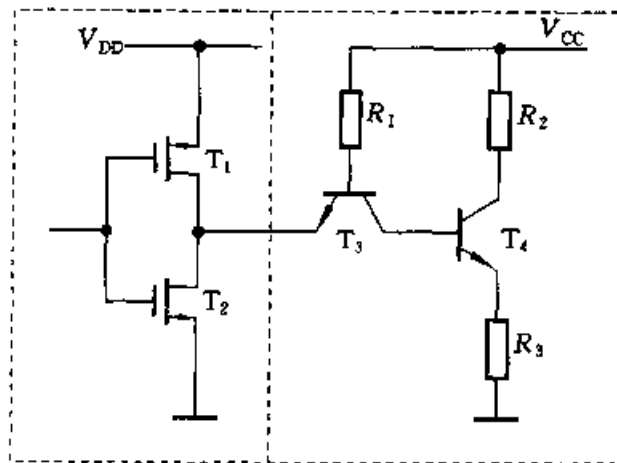


图 2.9.1 CMOS 门驱动 TTL 门

例 2.9.1 74HC00 与非门电路用来驱动一个基本的 TTL 反相器和六个 74LS 门电路。试验算此时的 CMOS 门电路是否过载？

解：(1) 由附录 C 查得接口参数如下：一个基本的 TTL 门电路， $I_{IL} = 1.6 \text{ mA}$ ，六个 74LS 门的输入电流 $I_{IL} = 6 \times 0.4 \text{ mA} = 2.4 \text{ mA}$ 。总的输入电流 $I_{IL(\text{total})} = 1.6 \text{ mA} + 2.4 \text{ mA} = 4 \text{ mA}$ 。

(2) 因 74HC00 门电路的 $I_{OL} = I_{IL} = 4 \text{ mA}$ ，所驱动的 TTL 门电路未过载。

2. TTL 门驱动 CMOS 门

此时 TTL 为驱动器件，CMOS 为负载器件。由附录 C 可知，当 TTL 输入为低电平时，它的输出电压参数与 CMOS HC 的输入电压参数是不兼容的。例如，LSTTL 的 $V_{OH(\text{min})}$ 为 2.7 V，而 HC CMOS 的 $V_{IH(\text{min})}$ 为 3.5 V。为了克服这一矛盾，常采用如图 2.9.2 所示的接口措施。由图可知，用上拉电阻 R_p 接到 V_{DD} 可将 TTL 的输出高电平电压升到约 5 V，上拉电阻的值取决于负载器件的数目以及 TTL 和 CMOS 的电流参数。此时 R_p 仍可利用式(2.4.14~2.4.15)作具体的计算。

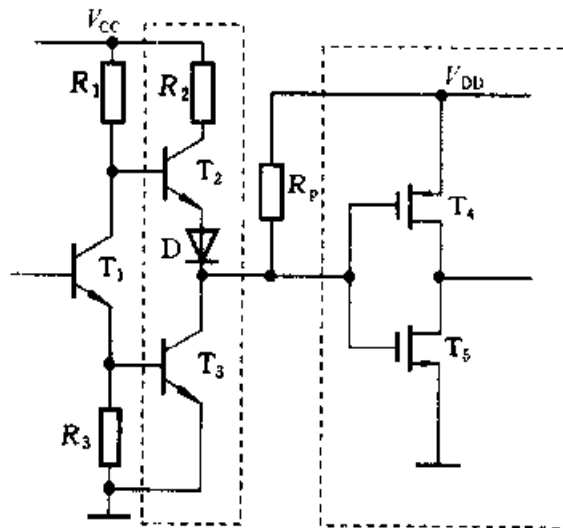


图 2.9.2 TTL 门驱动 CMOS 门

当 TTL 驱动 CMOS-HCT 时，由于电压参数兼容，不需另加接口电路。基于这一情况，在数字电路设计中，也常用 CMOS-HCT 当作接口器件，以去除上拉电阻。

2.9.2 门电路带负载时的接口电路

1. 用门电路直接驱动显示器件

在数字电路中，往往需要用发光二极管来显示信息的传输，如简单的逻辑器件的状态，七段数码显示，图形符号显示等。在每种情况下均需接口电路将数字

信息转换为模拟信息显示。

图 2.9.3 表示 CMOS 反相器 74HC04 驱动一发光二极管 LED, 电路中串接了一限流电阻 R 以保护 LED。限流电阻的大小可分别按下面两种情况来计算。

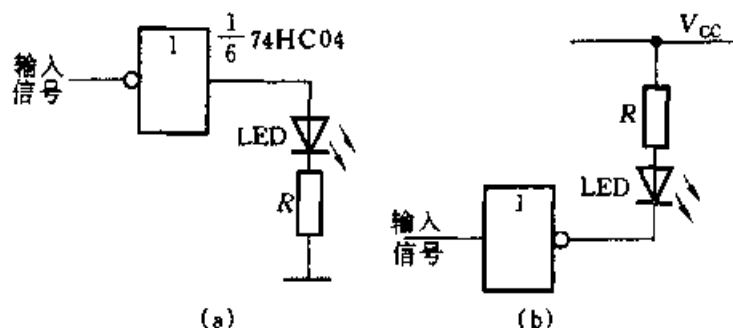


图 2.9.3 CMOS 74HC04 驱动 LED 的电路

对于图 2.9.3a, 当门电路的输入为低电平时, 输出为高电平, 于是

$$R = \frac{V_{OH} - V_F}{I_D} \quad (2.9.1)$$

反之, 当 LED 接入电路的情况如图 2.9.3b 所示时, 门电路的输入信号应为高电平, 输出为低电平, 故有

$$R = \frac{V_{CC} - V_F - V_{OL}}{I_D} \quad (2.9.2)$$

以上两式中, I_D ——LED 的电流, V_F ——LED 的正向压降, V_{OH} 和 V_{OL} 为门电路的输出高、低电平电压, 常取典型值。

例 2.9.2 试用 CMOS 反相器 74HC04 作为接口电路, 使门电路的输入为高电平时, LED 导通。设 $I_D = 10 \text{ mA}$, $V_F = 2.2 \text{ V}$, $V_{CC} = 5 \text{ V}$, $V_{OL} = 0.1 \text{ V}$ (由附录 C 查得)。

解: 根据题意, 可用式 (2.9.2) 来计算限流电阻值。

$$R = \frac{(5 - 2.2 - 0.1) \text{ V}}{10 \text{ mA}} = 270 \Omega$$

相应的电路如图 2.9.4 所示。

2. 机电性负载接口

在工程实践中, 往往会遇到用各种数字电路以控制机电性系统的功能, 如控制电动机的位置和转速, 继电器的接通与断开, 流体系统中的阀门的开通和关闭, 自动生产线中的机械手多参数控制等。

下面以继电器的接口电路为例来说明。

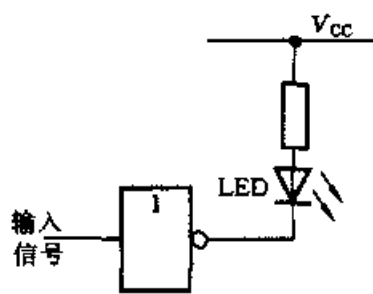


图 2.9.4 例 2.9.2 的电路

在继电器的应用中, 继电器本身有额定的电压和电流参数。一般情况下, 需

用运算放大器以提升到必须的数—模电压和电流接口值。

对于小型继电器,可以将两个反相器并联作为驱动电路,如图 2.9.5 所示。

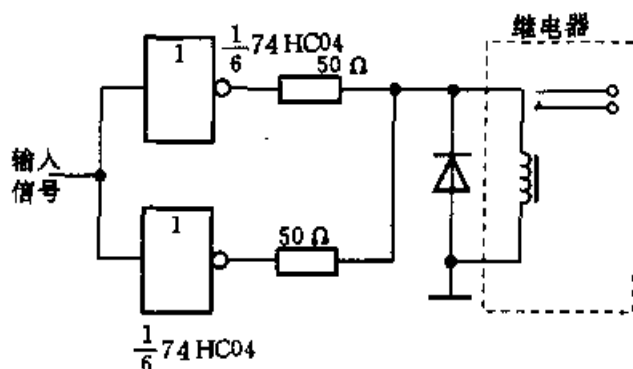


图 2.9.5 继电器驱动电路

2.9.3 抗干扰措施

在利用逻辑门电路(TTL 或 CMOS)作具体的设计时,还应当注意下列几个实际问题:

1. 多余输入端的处理措施

集成逻辑门电路在使用时,一般不让多余的输入端悬空,以防止干扰信号引入。对多余输入端的处理以不改变电路工作状态及稳定可靠为原则。

对于 TTL 与非门,一般可将多余的输入端通过上拉电阻(1~3 kΩ)接电源正端,也可利用一反相器将其输入端接地,其输出高电位可接多余的输入端。

对于 CMOS 电路,多余输入端可根据需要使之接地(或非门)或直接接 V_{DD} (与非门)。

2. 去耦合滤波器

数字电路或系统往往是由多片逻辑门电路构成,它们是由一公共的直流电源供电。这种电源是非理想的,一般是由整流稳压电路供电,具有一定的内阻抗。当数字电路运行时,产生较大的脉冲电流或尖峰电流,当它们流经公共的内阻抗时,必将产生相互的影响,甚至使逻辑功能发生错乱。一种常用的处理方法是采用去耦合滤波器,通常是用 10~100 μF 的大电容器与直流电源并联以滤除不需要的频率成分。除此以外,对于每一集成芯片还加接 0.1 μF 的电容器以滤除开关噪声。

3. 接地和安装工艺

正确的接地技术对于降低电路噪声是很重要的。这方面可将电源地与信号地分开,先将信号地汇集在一点,然后将二者用最短的导线连在一起,以避免含有多种脉冲波形(含尖峰电流)的大电流引到某数字器件的输入端而导致系统正常的逻辑功能失效。此外,当系统中兼有模拟和数字两种器件时,同样需将二者

的地分开,然后再选用一个合适共同点接地,以去除二者之间的影响。必要时,也可设计模拟和数字两块电路板,各备直流电源,然后将二者恰当的地连接在一起。在印刷电路板的设计或安装中,要注意连线尽可能短,以减少接线电容而导致寄生反馈有可能引起寄生振荡。有关这方面技术问题的详细介绍,可参阅有关文献。集成数字电路的数据手册,也提供某些典型电路应用设计,亦是有益的参考资料。

此外,CMOS 器件在使用和储藏过程中要注意静电感应导致损伤的问题。静电屏蔽是常用的防护措施。

复习思考题

2.9.1 当 TTL 门电路驱动 CMOS 门电路时,是否需要加接口电路? 为什么?

2.9.2 为什么说电压、电流参数为门电路之间主要的接口参数?

* 2.10 CAD 例题

例 CE2.10.1 应用 PSPICE 程序对图 2.4.2 所示的基本 TTL 反相器进行仿真,设负载电阻为 $40\text{ k}\Omega$ 。此电阻值与该门电路带同类门时所提供的电流(拉电流)相当。求电压传输特性 $v_1 - v_o$ 。

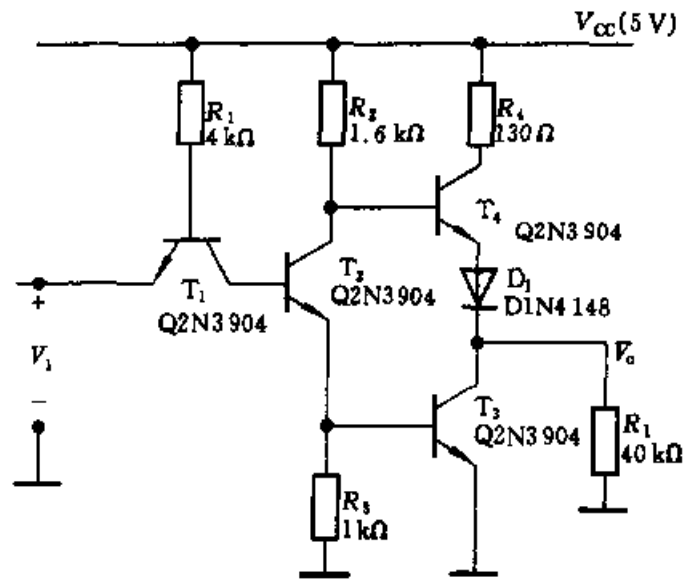
解: 将图 2.4.2 的电路重绘于图 CE2.10.1a 中,其中 BJT 模型 Q2N3904 只定义 $\text{BF}=100$ 和 $\text{BR}=0.1$ (即正向 β 和反向 β 值),其余参数均使用默认值。负载电阻 $\text{RL}=40\text{ k}\Omega$ 接在电路的输出端。设置 DC 扫描分析,仿真后传输特性如图 CE2.10.1b 所示。

例 CE2.10.2 应用 PSPICE 程序对图 2.6.5 所示的带电容负载 C_L 的 CMOS 反相器进行仿真。设参数和元件值为: $C_L=2\text{ pF}$,电源电压 $V_{\text{DD}}=5\text{ V}$,当两增强型 MOSFET M_1 和 M_2 对称,其电导参数 $|K|=0.2\text{ mA/V}^2$ 及开启电压 $|V_T|=1\text{ V}$ 。试证明其传输延迟时间 $t_{\text{PHL}}=t_{\text{PHL}}=1.6\text{ ns}$ 。

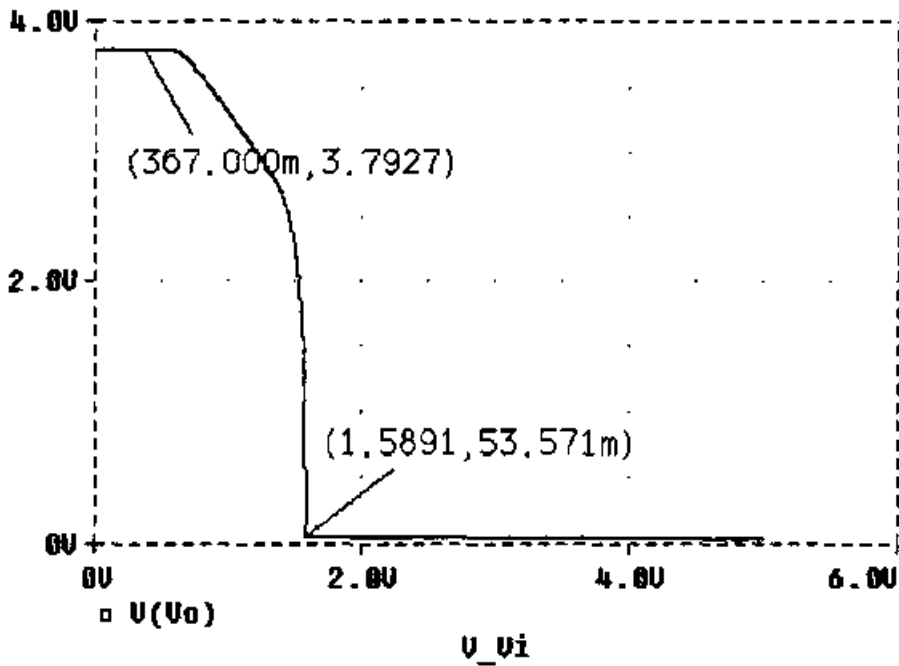
解: 已知场效应管模型中的开启电压 $|V_T| \cdot |V_{\text{TR}}|$,跨导系数 $\text{KP}=2|K|$,且选择沟道长度调制系数 $\text{LAMBDA}=0.01/\text{V}$ 。(参阅本书模拟部分附录)

将 M_1 和 M_2 的参数定义为题目中要求的参数。选用幅度为 5 V 、上升时间和下降时间均为 1 ns 的脉冲信号源 PWL,设置瞬态分析,仿真后的波形如图 CE2.10.2b 所示。由图中看出传输延迟时间约为 1.6 ns 。

例 CE2.10.3 十字路口的路况如图题 2.9.11 所示(见后),现重绘于图 CE2.10.3a 中。通道 A 为主干道,当通道 A 没有车辆行驶,而通道 B_1 或 B_2 有车辆停留等待时,则 B_1 或 B_2 的车辆可以行驶。当通道 A 有车时,无论 B_1



(a)



(b)

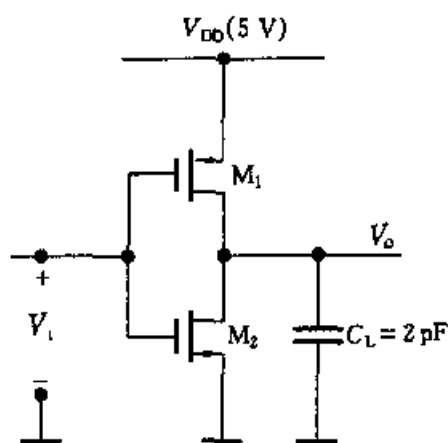
图 CE2.10.1

(a)基本 TTL 反相器电路 (b) 传输特性曲线

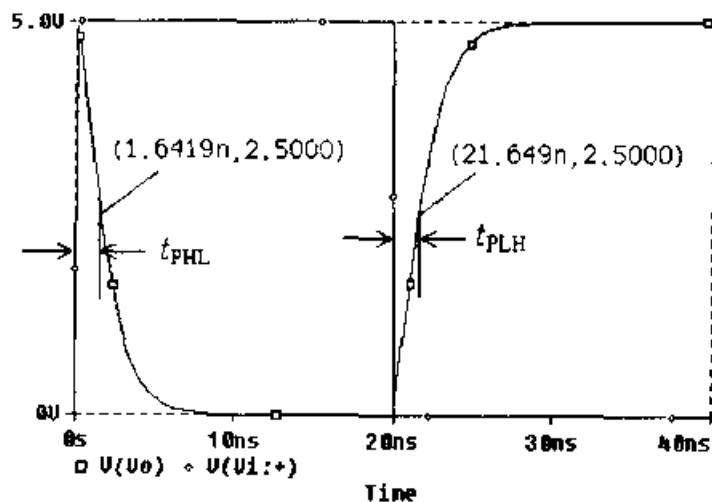
或 B_2 的情况如何,通道 A 允许通行。试用逻辑门电路和 ABEL 语言设计该交通灯控制电路,然后用 ISP Synario 软件进行输入和逻辑功能仿真,并给出仿真波形。

解: (1) 逻辑设计。

假设输入信号用 1 表示有车,用 0 表示无车;



(a)



(b)

图 CE2.10.2

(a) 带电容负载的 CMOS 反相器 (b) 传输延迟时间的仿真结果

输出信号 $LA = 1$, 表示 A 车道绿灯亮, 可以行驶; $LA = 0$, 表示 A 车道红灯亮, 禁止行驶。同理, $LB = 1$, 表示 B 车道绿灯亮, 可以行驶; $LB = 0$, 表示 B 车道红灯亮, 禁止行驶。

由题意可以列出真值表, 如表例 CE2.10.3 所示。

表例 CE2.10.3

A	B ₁	B ₂	LA	LB
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

* 表中 A 含 A₁ 和 A₂。

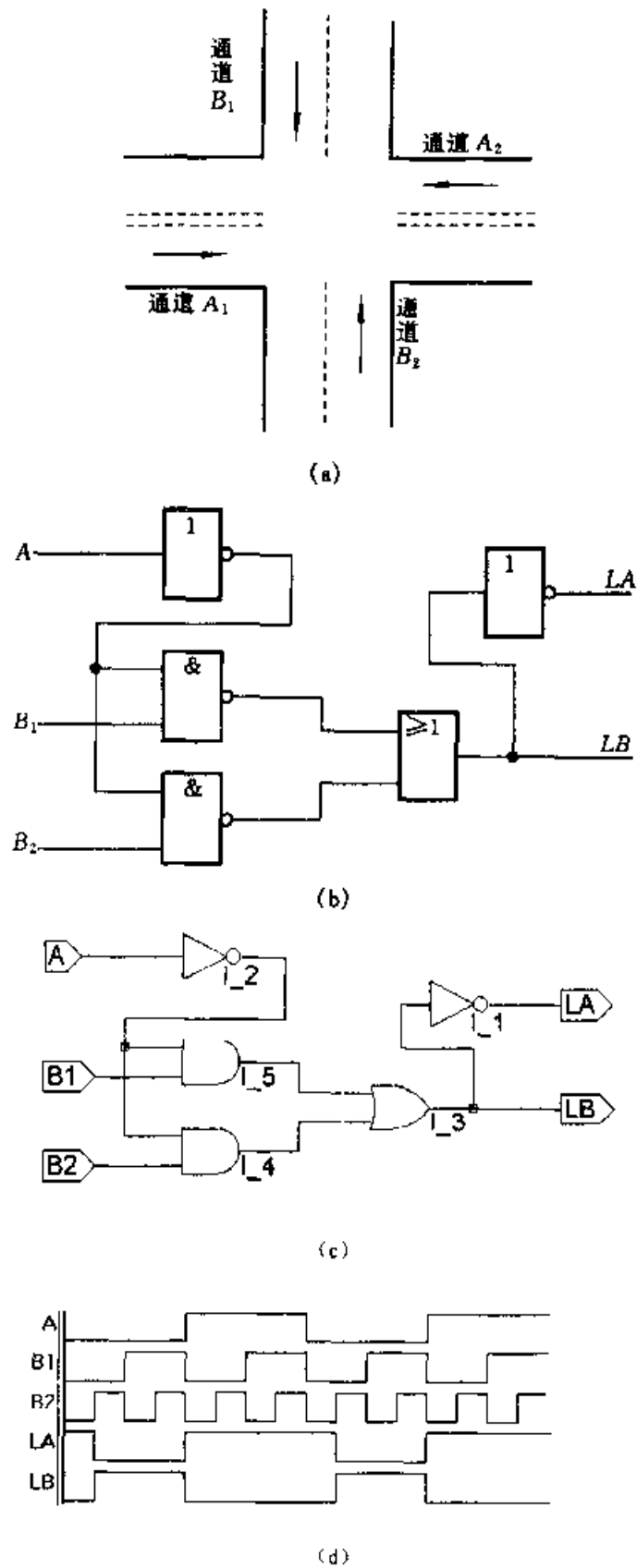


图 CE2.10.3 交通灯控制电路

(a) 路况图 (b) 逻辑电路图 (c)、(d)用 ISP Synano 软件控制打印的逻辑电路图和逻辑功能仿真波形图


```

3 -> X;
4 -> X;
5 -> X;
6 -> X;
7 -> X;|
end           "模块结束

```

(4) 进行逻辑功能仿真。

对测试向量文件进行编译和逻辑功能仿真,得到如图 CE2.10.3d 所示的仿真波形图。

小 结

• 在双极型逻辑门电路中,不论哪一种逻辑门电路,其中的关键器件是二极管和 BJT。影响它们的开关速度的主要因素是器件内部的电荷存储和消散的时间。

• 利用二极管和 BJT 可构成简单的逻辑与、或、非门电路。

• 逻辑非门(反相器)电路的主要技术参数为扇出数、噪声容限、传输延迟时间及功耗和功耗-延迟时间积等。

• TTL 逻辑门电路是当前应用较广泛的门电路之一,电路的基础是 NPN 型 BJT 反相器。TTL 反相器的输入级由 BJT 构成,它可能工作在饱和、截止、放大和倒置放大等 4 种模式,取决于反相器的输入和输出的状态。TTL 反相器输出级常用推拉式结构实现,其特点是输出阻抗低,带负载能力强。无论输入级还是输出级均有利于提高开关速度。如将 TTL 反相器的输入 BJT 改为多发射极结构便可构成与非门电路。

• 利用肖特基二极管可以构成抗饱和 TTL 电路,同时在驱动级引入有源下拉电路可以提高工作速度。

• 在 TTL 逻辑门电路中,为了实现线与的逻辑功能,可以采用或非门、集电极开路门和三态门来实现。

• ECL 逻辑门电路是以差分放大电路为基础的,因为它不工作到 BJT 的饱和区,因而它的开关速度是众多逻辑门电路中最高的。其缺点是功耗较大,噪声容限低。

• CMOS 逻辑门电路是由互补的增强型 N 沟道和 P 沟道 MOSFET 构成,它是目前应用较广泛的另一种逻辑门电路。与 TTL 门电路相比,它的优点是功耗低,扇出数大(指带同类门负载),噪声容限亦大,开关速度与 TTL 接近,有取代 TTL 之趋势。BiCMOS 是取 MOS 和 ECL 两者的优势,其开关速度较高,功

耗亦低。

- NMOS 逻辑门电路结构简单,易于集成化,因而在大规模集成电路中应用较多。

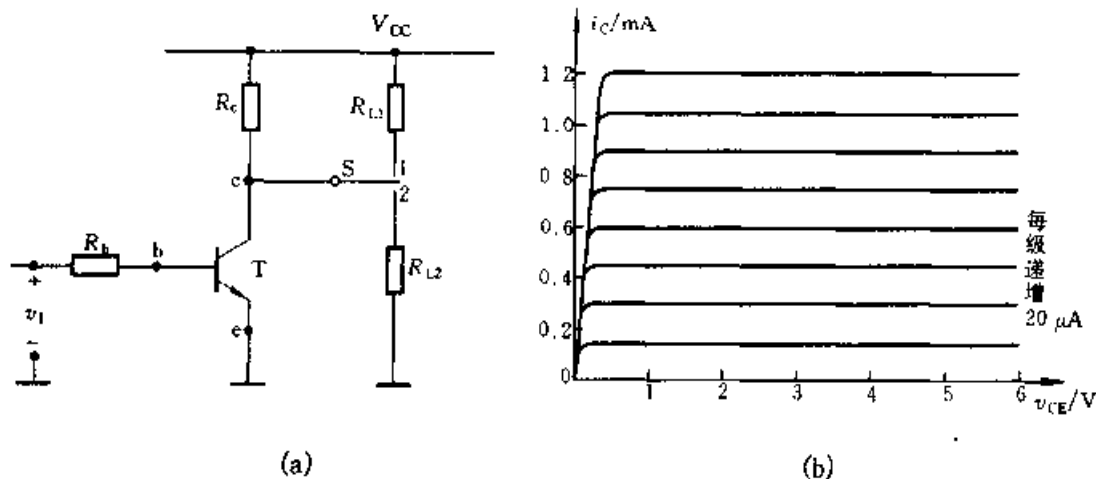
- 采用 NMOS、CMOS、TTL、ECL 和 BiCMOS 工艺可以制成多种系列的与非门和或非门等。

- 在逻辑体制中有正、负逻辑的规定,本书主要采用正逻辑。同样一个逻辑门电路,利用正、负逻辑等效变换原则,可以达到灵活运用目的。

- 在逻辑门电路的实际应用中,有可能遇到不同类型门电路之间,门电路与负载之间的接口技术问题以及抗干扰工艺问题。正确分析与解决这些问题,是数字电路设计工作者应当掌握的基本功。

习 题^①

2.2.1 图题 2.2.1a 表示一 BJT 反相器电路,图 b 为 BJT 的输出 $v-i$ 特性,试求解下列问题:(1) 设图 a 中的参数为: $V_{CC} = 5\text{ V}$, $R_c = 4.3\text{ k}\Omega$, $R_b = 30\text{ k}\Omega$, $v_i = 5\text{ V}$,用图解法求 Q 点 (I_B , I_C , V_{CE});(2) 若将开关 S 置于位置 1,并设 $R_{L1} = 10\text{ k}\Omega$,问此时 V_{CE} = ? 并说明 R_{L1} 的大小或 $R'_c = R_c \parallel R_{L1}$ 大小的变化,对 V_{CE} 值有何影响,从而检查 BJT 的饱和深度;(3) 若电路其他参数不变,将 S 置于位置 2,并设 $R_{L2} = 15\text{ k}\Omega$,问此时的 V_{CE} = ?

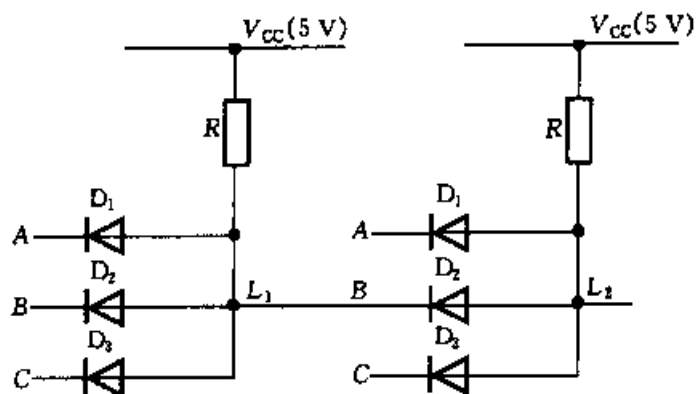


图题 2.2.1

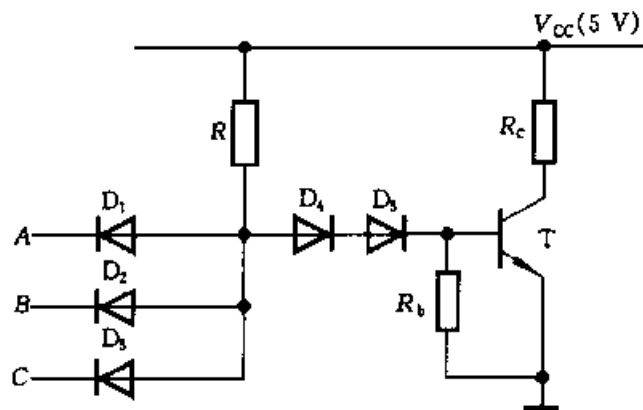
2.2.2 在图题 2.2.1a 所示的电路中:(1) 开关 S 悬空,当输入端接入 $0\text{ V} \sim 5\text{ V}$ 的方波脉冲信号,试近似绘出 v_o 的波形;(2) 当 S 分别置于位置 1 和 2 时, v_o 的波形的幅值有何变化?

2.3.1 图题 2.3.1 表示二极管与门带同类与门负载的情况,若门的级数愈多,则每级的门的输出电压 v_o 将作怎样的变化趋势? 如何解决此缺点? *

① 习题中所用集成门电路的技术参数见附录 C。



图题 2.3.1



图题 2.3.2

2.3.2 图题 2.3.2 表示一由二极管与门和 BJT 反相器相结合的二极管 BJT 与非门电路(称为 DTL 电路),以解决图题 2.3.1 所示电路存在的问题,试分析该电路为什么能实现与非逻辑关系? 二极管 D_4 、 D_5 起什么作用,可否省略 D_5 ?

2.4.1 为什么说 TTL 与非门的输入端在以下 4 种接法下都属于逻辑 0:(1)输入端接地;(2)输入端接低于 0.8 V 的电源;(3)输入端接同类与非门的输出低电压 0.2 V;(4)输入端通过 $500\ \Omega$ 的电阻接地。

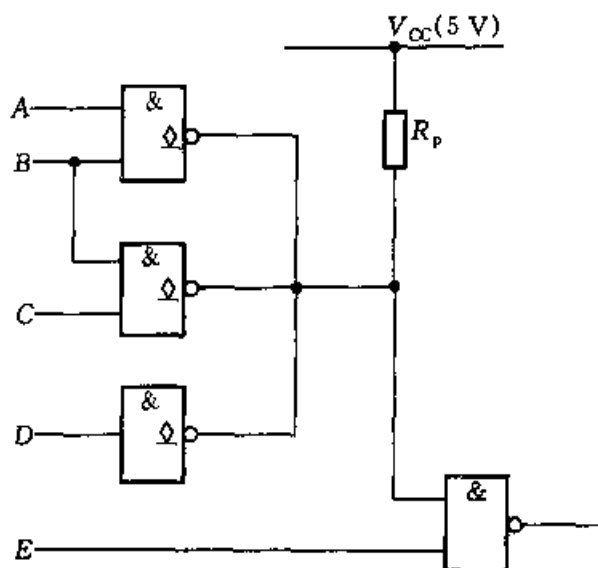
2.4.2 为什么说 TTL 与非门的输入端在以下 4 种接法下,都属于逻辑 1:(1)输入端悬空;(2)输入端接高于 2 V 的电源;(3)输入端接同类与非门的输出高电压 3.6 V;(4)输入端接 $10\ \text{k}\Omega$ 的电阻到地。

2.4.3 求下列情况下的扇出数:(1)LSTTL 门驱动同类门;(2) LSTTL 门驱动基本的 TTL 门。

2.4.4 设有一 74LS 00 反相器驱动两个 7404 反相器和 4 个 74LS 00 反相器。(1)问驱动门是否超载?(2)若超载,试提出一改进方案;若未超载,问还可增加几个 74LS 00 门?

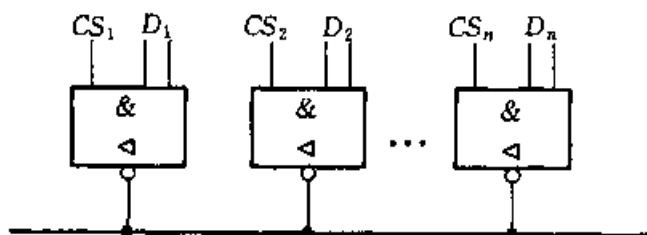
2.4.5 求图题 2.4.5 所示电路的输出逻辑表达式。

2.4.6 试用 3 个集电极开路 TTL 与非门 7401 和一个 TTL 反相器 74LS04 实现图题 2.4.5 的电路,设 $V_{CC} = 5\ \text{V}$,求 $R_{p(\min)}$ 和 $R_{p(\max)}$ 。



图题 2.4.5

2.4.7 图题 2.4.7 表示三态门作总线传输的示意图,图中 n 个三态门的输出接到数据传输总线, D_1, D_2, \dots, D_n 为数据输入端, CS_1, CS_2, \dots, CS_n 为片选信号输入端。试问 CS 信号如何进行控制,以便数据 D_1, D_2, \dots, D_n 通过该总线进行正常传输。



图题 2.4.7

2.6.1 图题 2.6.1 表示 2 输入端 BiCMOS 与非门电路,试分析该电路是怎样实现与非逻辑关系的,即 $L = \overline{A \cdot B}$?

2.6.2 某厂生产的双互补对和反相器(4007)引出端如图题 2.6.2 所示,试分别连接:(1) 3 反相器;(2) 3 输入端或非门;(3) 3 输入端与非门;(4) 或非门 [$L = \overline{C(A+B)}$];(5) 传输门。

2.6.3 某 CMOS 器件的电路如图题 2.6.3 所示,试写出其逻辑表达式,说明它是什么逻辑电路?

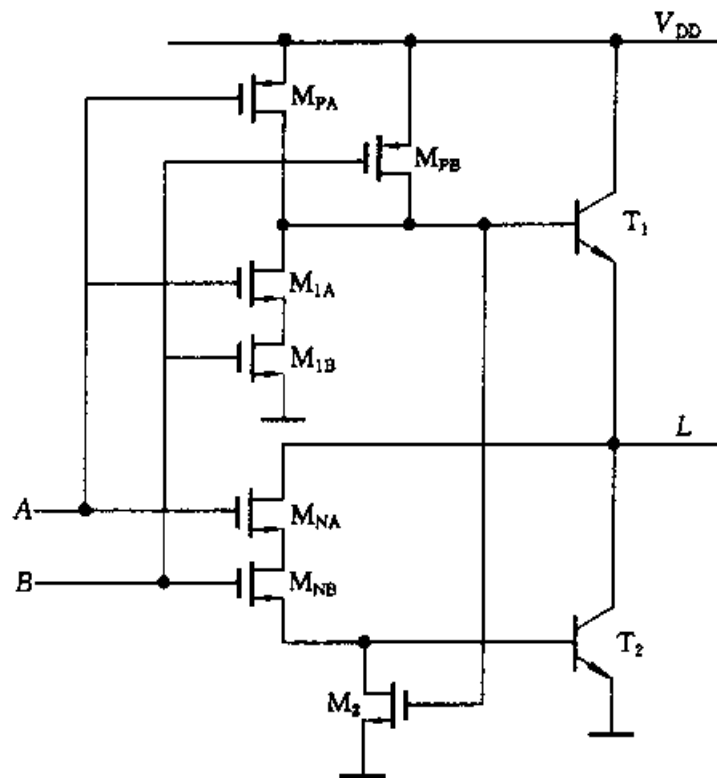
2.6.4 试利用 CMOS 传输门设计一个 CMOS 三态输出两输入或非门,并列出其真值表。

2.6.5 从附录 C 所列出的 CMOS 和 TTL 逻辑门电路的技术参数,比较两类电路的优缺点。

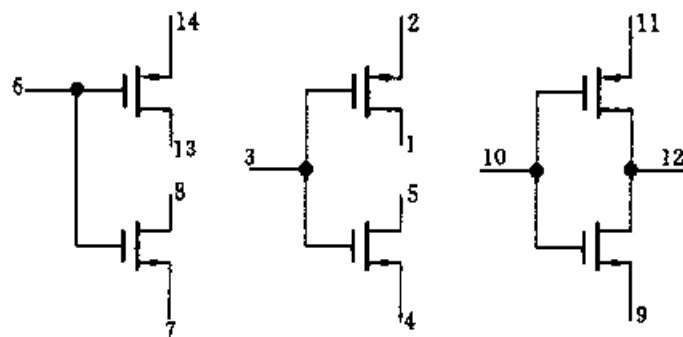
2.7.1 写出图题 2.7.1 所示电路的逻辑表达式。

2.7.2 写出图题 2.7.2 所示电路的逻辑表达式。

2.9.1 复习并解释下列名词:驱动、负载、拉电流、灌电流、扇出。



图题 2.6.1



图题 2.6.2

2.9.2 当 CMOS 和 TTL 两种门电路相互连接时,有哪 8 个电压和电流参数? 试列出这些参数并对每一参数进行解释。

2.9.3 当用 74TTL 电路去驱动 74HC 电路时,试简述其设计思路,是否需要接口电路? 计算其扇出数,并对接口电路就开关速度和功耗两方面作出评价。

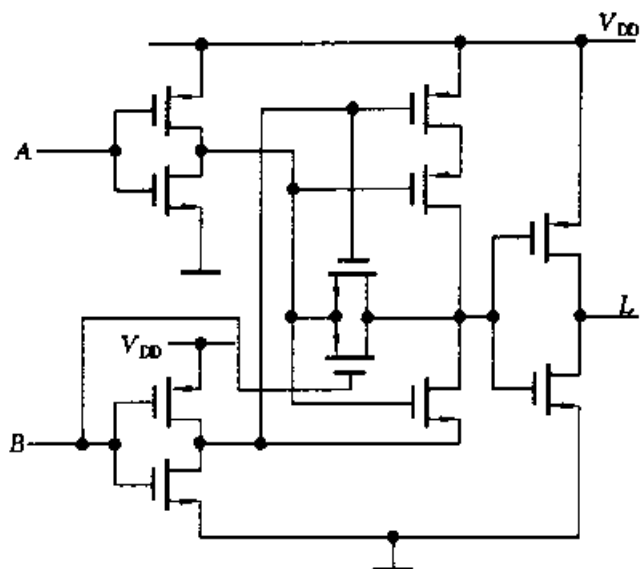
2.9.4 当用 LSTTL 去驱动 HC CMOS 时,重复题 2.9.3。

2.9.5 当用 ALSTTL 去驱动 HC CMOS 时,重复题 2.9.3。

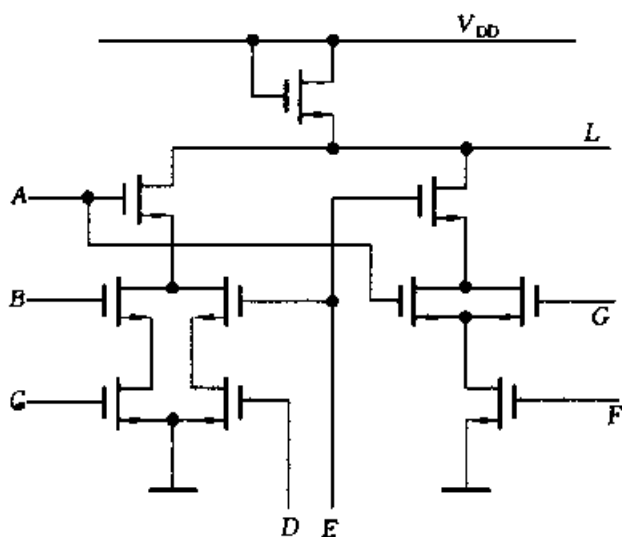
2.9.6 当用 HC CMOS 去驱动基本的 74TTL 门电路时,试简述其设计思路,指出是否需要加接口电路? 并就开关速度和功耗两方面对接口电路地进行评价。

2.9.7 当用 HC CMOS 驱动 LSTTL 时,重复 2.9.6。

2.9.8 当用 HC CMOS 驱动 ALSTTL 时,重复 2.9.6。



图题 2.6.3



图题 2.7.1

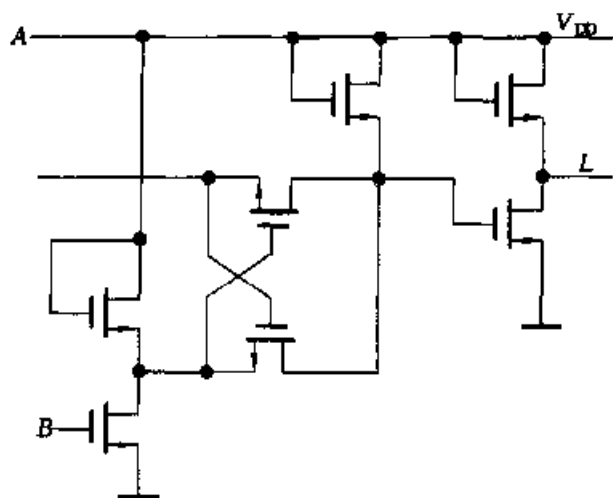
2.9.9 复习一下 TTL 门的输出电路。若 TTL 的输出级超载时,电路会出现什么现象?用什么仪器进行判断?

2.9.10 设计一发光二极管(LED)驱动电路,设 LED 的参数为 $V_F = 2.5 \text{ V}$, $I_D = 4.5 \text{ mA}$;若 $V_{CC} = 5 \text{ V}$,当 LED 发亮时,电路的输出为低电平,选用集成门电路的型号,并画出电路图。

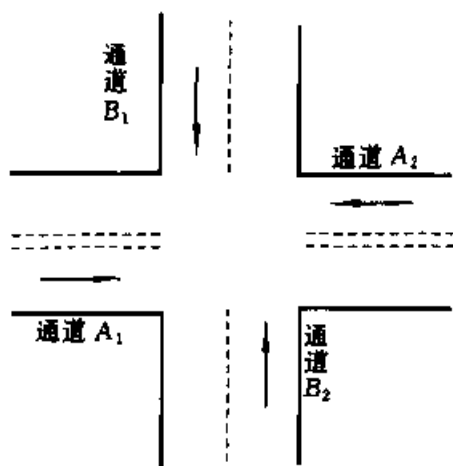
2.9.11 十字路口的路况如图题 2.9.11 所示。通道 A(含 A_1 和 A_2)为主干道,当通道 A 没有车辆行驶,而通道 B_1 或 B_2 有车辆停留等待时,则该处的车辆可以行驶;当通道 A 有车时,无论通道 B 的情况如何,通道 A 允许通行。试用逻辑门电路设计交通灯控制电路。

2.9.12 路况仍如图题 2.9.11 所示。交通管理规则如下:(1)若通道 B(含 B_1 和 B_2)没有两辆车等待,则通道 A 允许通行;(2)若通道 A 处有两辆车停留待发时,则无论通道 B 处

的情况如何,通道 A 都允许通行。试用逻辑门电路设计交通灯控制电路。



图题 2.7.2



图题 2.9.11

* CAD 习题

C2.10.1 应用 PSPICE 程序分析图 2.4.2 所示 TTL 反相器的各节点和支路中的电压、电流值。设 v_l 为低电位(0.2 V)。

C2.10.2 应用 PSPICE 程序分析图 2.4.2 所示 TTL 反相器的各节点和支路中的电压、电流值。设 v_1 为高电位(3.7 V)。

C2.10.3 应用 PSPICE 程序确定带电容负载 $C_L = 5\text{pF}$ 、 $V_{DD} = 5\text{V}$ 时的传输延迟时间 t_{PHL} 和 t_{PLH} 。设 MOSFET 的参数与例 CE2.10.2 相同。

C2.10.4 十字路口的路况仍如图题 2.9.11 所示,交通管理规则如下:(1)通道 A(含 A_1 和 A_2)为主干道,若通道 B(含 B_1 和 B_2)处没有停留两辆车等待,则通道 A 允许通行;(2)若通道 A 有两辆车停留待发时,无论 B 处的情况如何,则通道 A 允许通行。试用逻辑门电路和 ABEL 语言设计该交通灯控制电路,然后用 ISP Synario 软件进行输入和逻辑功能仿真,并给出仿真波形。

3 组合逻辑电路的分析与设计

引言 数字系统中常用的各种数字部件,就其结构和工作原理而言可分为两大类,即组合逻辑电路和时序逻辑电路。前一章介绍了构成数字系统的基本单元——逻辑门电路。本章将首先介绍分析和设计数字电路的工具——逻辑代数和卡诺图,然后运用这些基本知识分析和设计组合逻辑电路。

在任何时刻,输出状态只决定于同一时刻各输入状态的组合,而与先前状态无关的逻辑电路称为组合逻辑电路。图 3.0.1 是组合逻辑电路的一般框图,它可用如下的逻辑函数来描述,即

$$L_i = f(A_1, A_2, \dots, A_n) \quad (i = 1, 2, \dots, m) \quad (3.0.1)$$

式中 A_1, A_2, \dots, A_n 为输入变量。

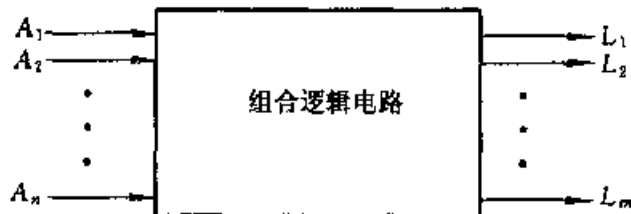


图 3.0.1 组合逻辑电路的一般框图

组合逻辑电路具有如下特点:

- (1) 输出、输入之间没有反馈延迟通路;
- (2) 电路中不含记忆单元。

3.1 逻辑代数

逻辑代数亦称为布尔代数,其基本思想是英国数学家布尔于 1854 年提出的。1938 年,香农把逻辑代数用于开关和继电器网络的分析、化简,率先将逻辑代数用于解决实际问题。经过几十年的发展,逻辑代数已成为分析和设计逻辑电路不可缺少的数学工具。

逻辑代数提供了一种方法,即使用二值函数进行逻辑运算,这样,一些用语言描述显得十分复杂的逻辑命题,使用数学语言后,就变成了简单的代数式。逻辑电路中的一个逻辑命题,不仅包含肯定和否定两重含义,而且包含条件与结果许多种可能的组合。比如,一个3输入端的与非门存在着输入与输出状态的八种可能的组合。用语言描述既啰嗦又不清晰,用真值表则一目了然,而用代数式 $L = \overline{ABC}$ 表达就更为简明。

逻辑代数有一系列的定律和规则,用它们对数学表达式进行处理,可以完成对电路的化简、变换、分析和设计。

3.1.1 逻辑代数的基本定律和恒等式

常用的逻辑代数定律和恒等式如表 3.1.1 所示。

表 3.1.1 逻辑代数定律和恒等式

基本定律			
	加	乘	非
	$A + 0 = A$	$A \cdot 0 = 0$	
	$A + 1 = 1$	$A \cdot 1 = A$	
	$A + A = A$	$A \cdot A = A$	$\overline{\overline{A}} = A$
	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$	
结合律	$(A + B) + C = A + (B + C)$	$(AB)C = A(BC)$	
交换律	$A + B = B + A$	$AB = BA$	
分配律	$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$	
反演律(摩根定律 ^①)	$\overline{A \cdot B \cdot C \cdots} = \overline{A} + \overline{B} + \overline{C} + \cdots$	$\overline{A + B + C + \cdots} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdots$	
吸收律	$A + AB = A$		
	$A(A + B) = A$		
	$A + \overline{A}B = A + B$		
	$(A + B)(A + C) = A + BC$		
其他常用恒等式			
	$AB + \overline{A}C + BC = AB + \overline{A}C$		
	$AB + \overline{A}C + BCD = AB + \overline{A}C$		

表 3.1.1 中的基本定律是根据逻辑加、乘、非三种基本运算法则,推导出的逻辑运算的一些基本定律。

① 摩根定律系 De Morgan's 的译称。

表 3.1.1 所列的定律,最有效的证明方法是检验等式左边的函数与右边函数的真值表是否吻合。

例如,要证明 $A + A = A$ 时,令 $A = 1$,则 $A + A = 1 + 1 = 1 = A$;再令 $A = 0$,则 $A + A = 0 + 0 = 0 = A$;除此之外,别无其他可能,可见 $A + A = A$ 。

恒等式可以用其他更基本的定律加以证明,我们来证明其中的第一条。

$$\begin{aligned} AB + \overline{AC} + BC &= AB + \overline{AC} + (A + \overline{A})BC \\ &= AB + \overline{AC} + ABC + \overline{A}BC \\ &= AB(1 + C) + \overline{AC}(1 + B) \\ &= AB + \overline{AC} \end{aligned}$$

在以上所有定律中,反演律具有特殊重要的意义。反演律又称为摩根定律,它经常用于求一个函数的非函数或者对逻辑函数进行变换。该定律可以用真值表加以证明。为了证明 $\overline{A + B} = \overline{A} \cdot \overline{B}$, $\overline{AB} = \overline{A} + \overline{B}$,按 A 、 B 取值情况列出真值表如表 3.1.2 所示。从表中可以直接得出结果。

表 3.1.2 摩根定律的证明

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

本节所列出的基本公式反映了逻辑关系,而不是数量之间的关系,在运算中不能简单套用初等代数的运算规则。如初等代数中的移项规则就不能用,这是因为逻辑代数中没有减法和除法的缘故。这一点在使用时必须注意。

3.1.2 逻辑代数的基本规则

1. 代入规则

在任何一个逻辑等式中,如果将等式两边出现的某变量 A ,都用一个函数代替,则等式依然成立,这个规则称为代入规则。

例如,在 $B(A + C) = BA + BC$ 中,将所有出现 A 的地方都代以函数 $A + D$,则等式仍成立,即得

$$B[(A + D) + C] = B(A + D) + BC = BA + BD + BC$$

代入规则可以扩展所有基本定律的应用范围。例如前面用真值表证明了用二变量表示的摩根定律 $\overline{AB} = \overline{A} + \overline{B}$,若用 $L = CD$ 代替等式中的 A ,则 $(\overline{CD})B = \overline{CD} + \overline{B} = \overline{C} + \overline{D} + \overline{B}$,以此类推,摩根定律对任意多个变量都成立。

2. 反演规则

根据摩根定律,求一个逻辑函数 L 的非函数 \bar{L} 时,可以将 L 中的与(\cdot)换成或($+$),或($+$)换成与(\cdot);再将原变量换为非变量(如 A 换成 \bar{A}),非变量换为原变量;并将1换成0,0换成1;那么所得的逻辑函数式就是 \bar{L} 。这个规则称为反演规则。

利用反演规则,可以较容易地求出一个函数的非函数。例如,要求 $L = \bar{A}\bar{B} + CD + 0$ 的非函数 \bar{L} 时,按照上述法则,得

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

又如 $L = A + \overline{BC} + \overline{D + E}$, 则 $\bar{L} = \bar{A} \cdot \overline{(\bar{B} + \bar{C})} \cdot \overline{\overline{DE}}$ 。

运用反演规则时必须注意两点:

(1) 保持原来的运算优先顺序,即如果在原函数表达式中, AB 之间先运算,再和其他变量进行运算,那么非函数的表达式中,仍然是 AB 之间先运算。

(2) 对于反变量以外的非号应保留不变。

3. 对偶规则

L 是一个逻辑表达式,如把 L 中的与(\cdot)换成或($+$),或($+$)换成与(\cdot);1换成0,0换成1,那么就得到一个新的逻辑函数式,这就是 L 的对偶式,记作 L' 。例如, $L = (A + \bar{B})(A + C)$, 则 $L' = A \cdot \bar{B} + AC$ 。变换时仍需注意保持原式中先与后或的顺序。

所谓对偶规则,是指当某个逻辑恒等式成立时,则其对偶式也成立。

利用对偶规则,可从已知公式中得到更多的运算公式,例如,吸收律 $A + \bar{A}B = A + B$ 成立,则它的对偶式 $A(\bar{A} + B) = AB$ 也是成立的。

3.1.3 逻辑函数的代数变换与化简法

在第1章,曾经通过列写真值表,得到了楼梯照明灯控制的逻辑表达式,它是一个同或函数 $L = AB + \bar{A}\bar{B}$ 。那么,对应唯一的真值表,逻辑函数表达式和实现它的逻辑电路是不是唯一的呢?下面就讨论这个问题。

1. 逻辑函数的变换

先看两个例子。

例 3.1.1 函数 $L = \overline{A \cdot AB} + \overline{B \cdot AB}$ 对应的逻辑图如图 3.1.1 所示。利用逻辑代数的基本定律对上述表达式进行变换。

$$\begin{aligned} \text{解: } L &= \overline{A \cdot AB} + \overline{B \cdot AB} \\ &= \overline{AB(A + B)} \\ &= \overline{\overline{\overline{AB} \cdot \overline{AB}}} \\ &= AB + \bar{A}\bar{B} \end{aligned}$$

结果表明,图 3.1.1 所示电路也是一个同或门。

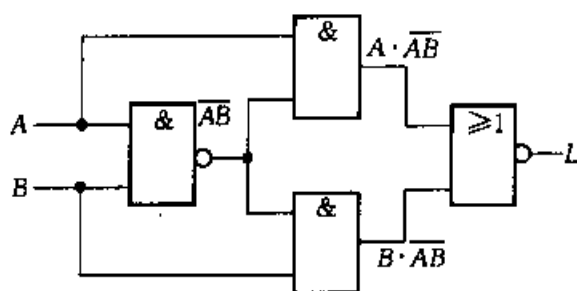


图 3.1.1 同或门逻辑电路之一

例 3.1.2 求同或函数的非函数。

$$\begin{aligned} \text{解: } L &= \overline{AB + \overline{A}\overline{B}} = \overline{AB} \cdot \overline{\overline{A}\overline{B}} = (\overline{A} + \overline{B})(A + B) \\ &= \overline{A}B + A\overline{B} \end{aligned}$$

这个函数称为异或函数,它表示当两个输入变量取值相异(一个为0,另一个为1)时,输出函数值为1。在 MOS 门电路中,我们已接触过异或门,上面的推导更明确地告诉我们,异或门和同或门互为非函数。所以在异或门电路的输出端再加一级反相器,也能得到同或门,如图 3.1.2 所示。

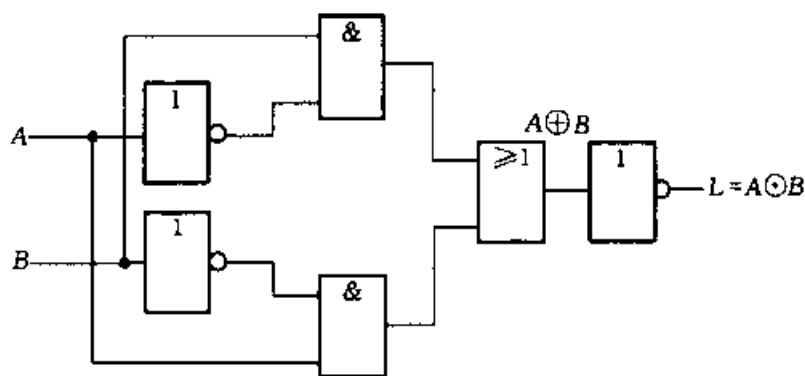


图 3.1.2 同或门逻辑电路之二

对应同或函数唯一的真值表,已列举出三种不同形式的逻辑表达式和三个逻辑电路,事实上还可以列举许多。由此可以得出结论:一个特定的逻辑问题,对应的真值表是唯一的,但实现它的电路多种多样。这给设计电路带来了方便,当我们手头缺少某种逻辑门的器件时,可以通过函数表达式的变换,避免使用这种器件而改用其他器件。这种情形在实际工作中常会遇到。

2. 逻辑函数的化简

根据逻辑表达式,可以画出相应的逻辑图。但是直接根据某种逻辑要求而归纳出来的逻辑表达式及其对应的逻辑图,往往并不是最简的形式,这就需要对逻辑表达式进行化简。

一个逻辑函数可以有多种不同的逻辑表达式,如与-或表达式、或-与表达

式、与非-与非表达式、或非-或非表达式以及与-或-非表达式等。例如：

$$\begin{aligned}
 L &= AC + \overline{CD} && \text{与-或} \\
 &= (A + \overline{C})(C + D) && \text{或-与} \\
 &= \overline{\overline{AC}} \cdot \overline{\overline{CD}} && \text{与非-与非} \\
 &= \overline{(A + \overline{C}) + (C + D)} && \text{或非-或非} \\
 &= \overline{\overline{AC} + \overline{CD}} && \text{与-或-非}
 \end{aligned}$$

以上五个式子是同一函数不同形式的最简表达式。以下将着重讨论与或表达式的化简,因为与或表达式易于从真值表直接写出,且只需运用一次摩根定律就可以从最简与或表达式变换为与非-与非表达式,从而可以用与非门电路来实现。

最简与或表达式有以下两个特点:

- ① 与项(即乘积项)的个数最少。
- ② 每个乘积项中变量的个数最少。

代数法化简逻辑函数是运用逻辑代数的基本定律和恒等式进行化简,常用下列方法:

① 并项法

利用 $A + \overline{A} = 1$ 的公式,将两项合并成一项,并消去一个变量。如

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

② 吸收法

利用 $A + AB = A$ 的公式,消去多余的项。如

$$L = \overline{A}B + \overline{A}BCD(E + F) = \overline{A}B$$

③ 消去法

利用 $A + \overline{A}B = A + B$,消去多余的因子,如

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

④ 配项法

先利用公式 $A + \overline{A} = 1$,增加必要的乘积项,再用并项或吸收的办法使项数

减少。如

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

使用配项的方法要有一定的经验,否则越配越繁。通常对逻辑表达式进行化简,要综合使用上述技巧。以下再举几例。

例 3.1.3 化简 $L = AD + A\overline{D} + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF$

$$\begin{aligned}
 \text{解: } L &= A + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF \quad (\text{利用 } A + \overline{A} = 1) \\
 &= A + \overline{A}C + BD + \overline{B}EF \quad (\text{利用 } A + AB = A) \\
 &= A + C + BD + \overline{B}EF \quad (\text{利用 } A + \overline{A}B = A + B)
 \end{aligned}$$

例 3.1.4 化简 $L = AB + A\overline{C} + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(F + G)$

$$\begin{aligned}
 \text{解: } L &= A(B + \overline{C}) + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(F + G) \quad (\text{分配律}) \\
 &= A\overline{B}C + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(F + G) \quad (\text{摩根定律}) \\
 &= A + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(F + G) \\
 &\quad (\text{利用 } A + \overline{A}B = A + B) \\
 &= A + \overline{B}C(D + \overline{D}) + \overline{C}B + \overline{B}D + \overline{D}B(C + \overline{C}) \\
 &\quad (\text{利用 } A + \overline{A} = 1 \text{ 配项及 } A + AB = A) \\
 &= A + \overline{B}CD + \overline{B}C\overline{D} + \overline{C}B + \overline{B}D + \overline{D}BC + \overline{D}B\overline{C} \quad (\text{分配律}) \\
 &= A + (\overline{B}CD + \overline{B}D) + (\overline{B}C\overline{D} + \overline{D}BC) + (\overline{C}B + \overline{D}B\overline{C}) \quad (\text{结合律}) \\
 &= A + \overline{B}D + C\overline{D} + B\overline{C} \quad (\text{利用 } A + AB = A \text{ 及 } A + \overline{A} = 1)
 \end{aligned}$$

复习思考题

- 3.1.1 什么是布尔代数?
- 3.1.2 (a) $A + 1 = ?$ (b) $A \cdot 0 = ?$ (c) $A \cdot A = ?$ (d) $A + A = ?$
- 3.1.3 写出二变量的摩根定律表达式。
- 3.1.4 是“与-或”表达式还是“或-与”表达式与真值表一致?
- 3.1.5 最简与或表达式的标准是什么?

3.2 逻辑函数的卡诺图^①化简法

由上面讨论可见,利用代数法可使逻辑函数变成较简单的形式。但使用这

^① 卡诺图是美国工程师 Karnaugh 在 20 世纪 50 年代提出的。

种方法要求熟练掌握逻辑代数的基本定律,而且需要一些技巧,特别是经代数化简后得到的逻辑表达式是否是最简式较难掌握,这就给使用代数法产生一定困难。本节介绍的卡诺图法可以比较简便地得到最简的逻辑表达式。

3.2.1 最小项的定义及其性质

1. 最小项的意义

n 个变量 X_1, X_2, \dots, X_n 的最小项是 n 个因子的乘积,每个变量都以它的原变量或非变量的形式在乘积项中出现,且仅出现一次。

举例来说,设 A, B, C 是 3 个逻辑变量,由这 3 个变量可以构成许多乘积项,如 $\overline{A}\overline{B}\overline{C}, \overline{A}\overline{B}C, \overline{A}B, A\overline{B}C\overline{A}, A(B+C)$ 等。其中 $\overline{A}\overline{B}\overline{C}, \overline{A}\overline{B}C$ 是最小项,而 $\overline{A}B, A\overline{B}C\overline{A}, A(B+C)$ 则不是最小项。

2. 最小项的性质

为了分析最小项的性质,以下列出 3 个变量的所有最小项的真值表,如表 3.2.1 所示。

表 3.2.1 3 变量最小项真值表

A	B	C	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	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

观察表 3.2.1 可以看出,最小项具有下列性质:

(1) 对于任意一个最小项,只有一组变量取值使得它的值为 1,而在变量取其他各组值时,这个最小项的值都是 0。

(2) 不同的最小项,使它的值为 1 的那一组变量取值也不同。

(3) 对于变量的任一组取值,任意两个最小项的乘积为 0。

(4) 对于变量的任一组取值,全体最小项之和为 1。

3. 最小项的编号

最小项通常用 m_i 表示,下标 i 即最小项编号,用十进制数表示。以 $\overline{A}BC$ 为例,因为它和 011 相对应,所以就称 $\overline{A}BC$ 是和变量取值 011 相对应的最小项,而 011 相当于十进制中的 3,所以把 $\overline{A}BC$ 记为 m_3 。按此原则,3 个变量的最小项

代表符号如表 3.2.2 所示。

表 3.2.2 3 变量最小项编号

最小项	变量取值			表示符号
	A	B	C	
$\overline{A}\overline{B}\overline{C}$	0	0	0	m_0
$\overline{A}\overline{B}C$	0	0	1	m_1
$\overline{A}B\overline{C}$	0	1	0	m_2
$\overline{A}BC$	0	1	1	m_3
$A\overline{B}\overline{C}$	1	0	0	m_4
$A\overline{B}C$	1	0	1	m_5
$AB\overline{C}$	1	1	0	m_6
ABC	1	1	1	m_7

3.2.2 逻辑函数的最小项表达式

利用逻辑代数的基本公式,可以把任一个逻辑函数化成一种典型的表达式,这种典型的表达式是一组最小项之和,称为**最小项表达式**。下面举例说明把逻辑表达式展开为最小项表达式的方法。例如,要将 $L(A, B, C) = AB + \overline{A}C$ 化成最小项表达式,这时可利用 $A + \overline{A} = 1$ 的基本运算关系,将逻辑函数中的每一项都化成包含所有变量 A、B、C 的项,即

$$\begin{aligned} L(A, B, C) &= AB + \overline{A}C = AB(C + \overline{C}) + \overline{A}C(B + \overline{B}) \\ &= ABC + AB\overline{C} + \overline{A}CB + \overline{A}C\overline{B} \end{aligned}$$

此式是由四个最小项构成的,它是一组最小项之和,因此是一个最小项表达式。

对照表 3.2.2,上式中各最小项可分别表示为 m_7 、 m_6 、 m_3 、 m_1 ,所以可写为

$$L(A, B, C) = m_1 + m_3 + m_6 + m_7$$

为了简化,常用最小项下标编号来代表最小项,故上式又可写为

$$L(A, B, C) = \sum m(1, 3, 6, 7)$$

又如,要将 $L(A, B, C) = \overline{(AB + \overline{A}\overline{B} + \overline{C})AB}$ 化成最小项表达式,可经下列几步:

(1) 多次利用摩根定律去掉非号,直至最后得到一个只在单个变量上有非号的表达式,即

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

(2) 利用分配律除去括号,直至得到一个与或表达式

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

(3) 在所得式子中,有一项 AB 不是最小项(缺少变量 C),则用 $(C + \overline{C})$ 乘此项,可得

$$\begin{aligned} L(A, B, C) &= \overline{A}BC + A\overline{B}C + AB(C + \overline{C}) \\ &= \overline{A}BC + A\overline{B}C + ABC + AB\overline{C} \\ &= m_3 + m_5 + m_6 + m_7 = \sum m(3, 5, 6, 7) \end{aligned}$$

由此可见,任一个逻辑函数都可化成为唯一的最小项表达式。

3.2.3 用卡诺图表示逻辑函数

1. 卡诺图的引出

一个逻辑函数的卡诺图就是将此函数的最小项表达式中的各最小项相应地填入一个特定的方格图内,此方格图称为卡诺图。因此,卡诺图是逻辑函数的一种图形表示。

下面从讨论一变量卡诺图开始,逐步过渡到多变量卡诺图。

大家知道, n 个变量的逻辑函数有 2^n 个最小项,因此一个变量的逻辑函数有两个最小项。设变量为 D ,则逻辑函数的最小项表达式为

$$L(D) = \overline{D} + D$$

其中 \overline{D} 和 D 是两个最小项,分别记为 m_0 和 m_1 ,即 $m_0 = \overline{D}$, $m_1 = D$ 。这两个最小项可用两个相邻的方格来表示,如图 3.2.1a 所示。方格上的 \overline{D} 和 D 分别表示原变量和非变量。为了简明起见,非变量 \overline{D} 可以不标出,只标出原变量 D ,即可得图 3.2.1b。图 3.2.1c 是进一步的简化画法,其中 m_0 、 m_1 只用其下标编号来表示。

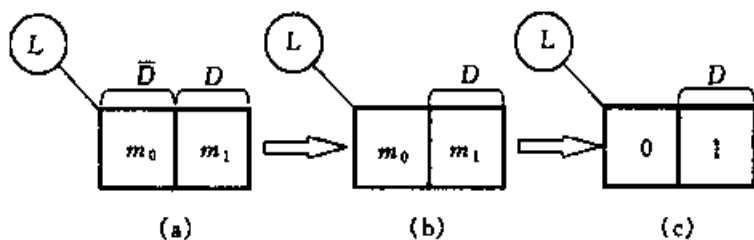


图 3.2.1 1 变量卡诺图

如果逻辑函数的变量增为两个,设为 C 、 D ,则 2 变量逻辑函数的最小项为 $2^2 = 4$ 项,函数的最小项表达式为 $L(C, D) = \overline{C}\overline{D} + \overline{C}D + C\overline{D} + CD$,即 $m_0 = \overline{C}\overline{D}$, $m_1 = \overline{C}D$, $m_2 = C\overline{D}$, $m_3 = CD$ 。由于有 4 个最小项,可用 4 个相邻的方格来表示。这 4 个方格可以由折叠了的 1 变量卡诺图展开来获得,如由图 3.2.2a 按箭头方向展开成图 3.2.2b。在图 3.2.2b 中,变量 D 标在图的底下,

标的规律符合展开的规律(参看图 3.2.1c),即中间两格底下为 D ,两边的两格底下为 \bar{D} (图中未标出 \bar{D})。因为变量 C 的标法必须区别于 D ,这样就有两种可能的标法,可以标在展开前两方格的顶上,也可标在展开后新的两个方格的顶上,图 b 采用后一种标法,以保持左边的第一格仍为 m_0 项,即维持展开前两方格最小项序号不改变。由图 3.2.2b 可看到一个规律:新的方格内最小项的编号比对应的原方格增加了 $2^{n-1} = 2^{2-1} = 2$ 。按照这个规律折叠时(图 3.2.2a),方格 1 后面为方格 3,方格 0 后面为方格 2,展开后即得图 3.2.2b 所示的 2 变量卡诺图。

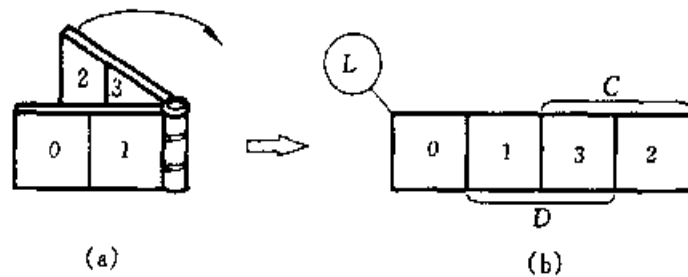


图 3.2.2 2 变量卡诺图

综上所述,可归纳“折叠展开”的法则如下:

- ① 新增加的方格按展开方向应标以新变量。
- ② 新的方格内最小项编号应为展开前对应方格编号加 2^{n-1} 。

按照同样的方法,可从折叠的 2 变量卡诺图展开获得 3 变量卡诺图。3 变量逻辑函数 $L(B, C, D)$ 应有 8 个最小项,可用 8 个相邻的方格来表示,这 8 个方格可由图 3.2.3a 展开成图 3.2.3b 来获得。新增加的 4 个方格按展开方向应标以新增加的变量 B (以区别于原来的变量 C, D)。而且,新增加的方格内最小项的编号为展开前对应方格编号加 $2^{n-1} = 2^{3-1} = 4$,这样即可获得 3 变量卡诺图,如图 3.2.3b 所示。在图中,可根据某一方格所处的位置,列出该方格代表的最小项,例如,2 号方格处于变量为 \bar{B}, C, \bar{D} 的区域,则 $m_2 = \bar{B}C\bar{D}$,余类推。

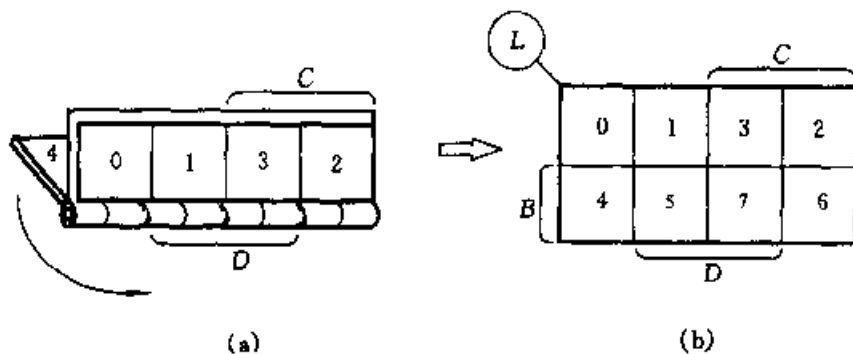


图 3.2.3 3 变量卡诺图

同理,可得4变量卡诺图,如图3.2.4所示。

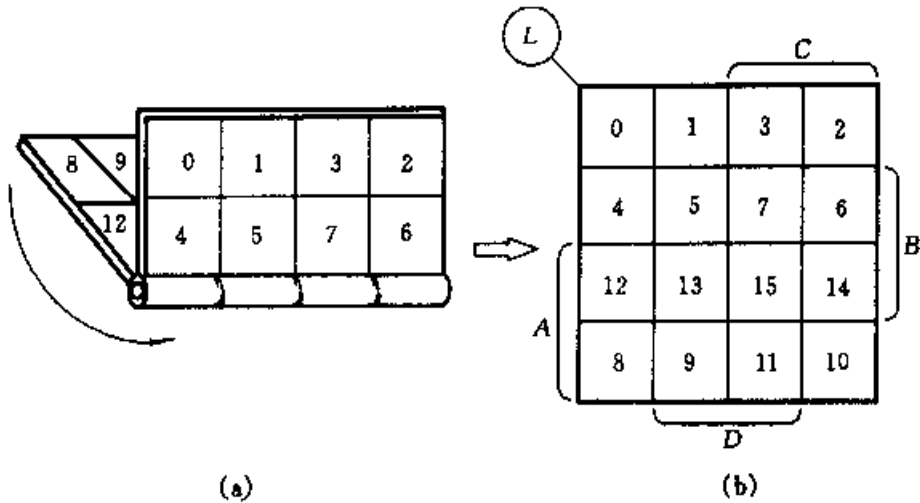


图 3.2.4 4 变量卡诺图

在使用时,只要熟悉了卡诺图上各变量的取值情况(即方格外各变量 A 、 B 、 C 、 D 等取值的区域),就可直接填入对应的最小项。

2. 卡诺图的特点

上面所得各种变量的卡诺图,其共同特点是可以直接观察相邻项。也就是说,各小方格对应于各变量不同的组合,而且上下左右在几何上相邻的方格内只有一个因子有差别,这个重要特点成为卡诺图化简逻辑函数的主要依据。现以4变量卡诺图为例来说明,为清楚起见,把各最小项填入对应方格内,如图3.2.5所示。可见,图中各行或各列上下左右相邻的方格内只有一个因子不同,例如, m_4 对应于 $\overline{A}B\overline{C}\overline{D}$, m_5 对应于 $\overline{A}BC\overline{D}$, 它们的差别仅在 D 和 \overline{D} , m_5 和 m_{13} 只差 A 和 \overline{A} , 余类推。要特别指出的是,卡诺图水平方向同一行里,最左和最右端的方格也是符合上述相邻规律的,例如, m_4 和 m_6 的差别仅在 C 和 \overline{C} 。同样,垂直方向同一列里最上端和最下端两个方格也是相邻的,这是因为都只有一个因子有差别。这个特点说明卡诺图呈现循环邻接的特性。

以上各卡诺图变量的排列形式(即卡诺图方格外 A 、 B 、 C 、 D 等所表示的变量)是为了获得循环邻接的特性。实际上,在满足循环邻接的前提下,卡诺图还有其他形式的画法,上面所列的只是其中的一种。

3. 卡诺图的简化表示法

图3.2.5所示的卡诺图可以简化成如图3.2.6所示。在后图中,用0、1分别表示非变量和原变量,变量 A 、 B 、 C 、 D 的每组取值,与方格内的最小项编号一一对应,例如,0000对应于 $\overline{A}\overline{B}\overline{C}\overline{D}$, 1111对应于 $ABCD$, 余类推。这样,只要标出方格外纵、横两向的二元常量,即可由二进制码推出相应的最小项的十进制编号。

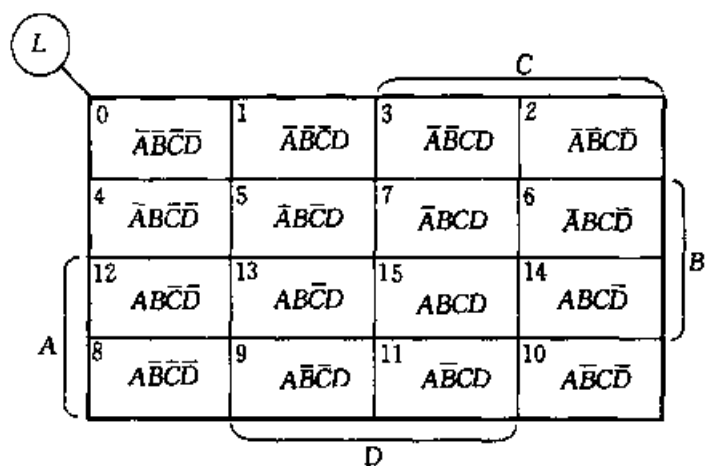


图 3.2.5 填入最小项的卡诺图

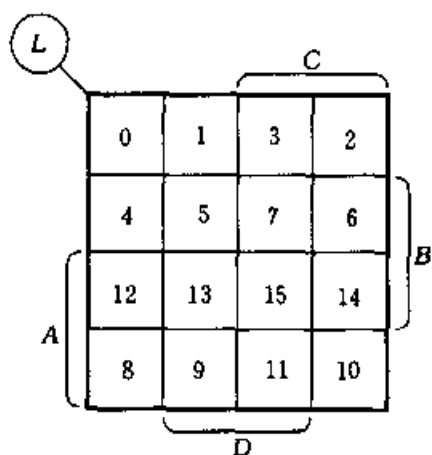
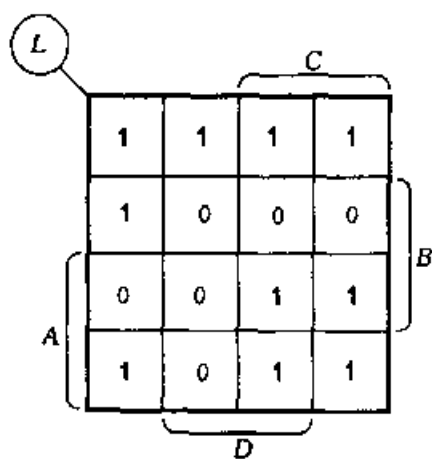


图 3.2.6 图 3.2.5 的简化表示法

图 3.2.7 $L(A, B, C, D)$ 的卡诺图

4. 已知逻辑函数画卡诺图

上面得到各种变量卡诺图的一般形式。根据逻辑函数的最小项表达式,就可以得到相应的卡诺图。例如,要画出逻辑函数 $L(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 8, 10, 11, 14, 15)$ 的卡诺图时,可根据图 3.2.6 所示 4 变量卡诺图的简化形式,对上列逻辑函数最小项表达式中的各项,在卡诺图相应方格内填入 1,其余填入 0,即可得图 3.2.7 所示的 $L(A, B, C, D)$ 的卡诺图。

例 3.2.1 画出

$$L(A, B, C, D) = (\bar{A} + \bar{B} + \bar{C} + \bar{D})(\bar{A} + \bar{B} + C + \bar{D})(\bar{A} + B + \bar{C} + D) \\ (A + \bar{B} + \bar{C} + D)(A + B + C + D)$$

的卡诺图。

解: (1) 由摩根定律,上式化成

$$\bar{L} = ABCD + AB\bar{C}D + A\bar{B}C\bar{D} + \bar{A}BC\bar{D} + \bar{A}\bar{B}C\bar{D} \\ = \sum m(15, 13, 10, 6, 0)$$

(2) 因上式中最小项之和为 \bar{L} , 故对 \bar{L} 中的各最小项, 在卡诺图相应方格内应填入 0, 其余填入 1, 即得图 3.2.8 所示的卡诺图。

3.2.4 用卡诺图化简逻辑函数

1. 化简的依据

我们知道, 卡诺图具有循环邻接的特性, 若图中两个相邻的方格均为 1, 则这两个相邻最小项的和将消去一个变量, 如图 3.2.6 所示 4 变量卡诺图中的方格 5 和方格 7, 它们的逻辑加是 $\bar{A}B\bar{C}D + \bar{A}BCD = \bar{A}BD(\bar{C} + C) = \bar{A}BD$,

消去了变量 C , 即消去了相邻方格中不相同的那个因子。若卡诺图中 4 个相邻的方格为 1, 则这 4 个相邻的最小项的和将消去两个变量, 如上述 4 变量卡诺图中的方格 2、3、7、6, 它们的逻辑加是

$$\begin{aligned}\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} &= \bar{A}\bar{B}\bar{C}(D + \bar{D}) + \bar{A}BC(D + \bar{D}) \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}BC = \bar{A}\bar{C}\end{aligned}$$

消去了变量 B 和 D , 即消去相邻 4 个方格中不相同的那两个因子, 这样反复应用 $A + \bar{A} = 1$ 的关系, 就可使逻辑表达式得到简化。这就是利用卡诺图法化简逻辑函数的基本原理。

2. 化简的步骤

用卡诺图化简逻辑函数的步骤如下:

- (1) 将逻辑函数写成最小项表达式。
- (2) 按最小项表达式填卡诺图, 凡式中包含了的最小项, 其对应方格填 1, 其余方格填 0。

(3) 合并最小项, 即将相邻的 1 方格圈成一组(包围圈), 每一组含 2^n 个方格, 对应每个包围圈写成一个新的乘积项。

(4) 将所有包围圈对应的乘积项相加。

有时也可以由真值表直接填卡诺图, 以上的(1)、(2)两步就合为一步。

画包围圈时应遵循以下原则:

- (1) 包围圈内的方格数必定是 2^n 个, n 等于 0、1、2、3、...
 - (2) 相邻方格包括上下底相邻, 左右边相邻和四角相邻。
 - (3) 同一方格可以被不同的包围圈重复包围, 但新增包围圈中一定要有新的方格, 否则该包围圈为多余。
 - (4) 包围圈内的方格数要尽可能多, 包围圈的数目要尽可能少。
- 化简后, 一个包围圈对应一个与项(乘积项), 包围圈越大, 所得乘积项中的

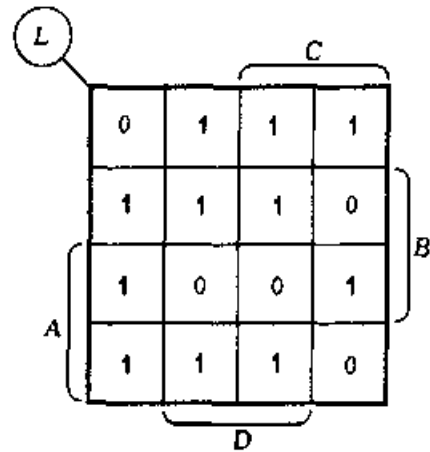


图 3.2.8 例 3.2.1 的卡诺图

变量越少。实际上,如果做到了使每个包围圈尽可能大,结果包围圈个数也就会少,使得消失的乘积项个数也越多,就可以获得最简的逻辑函数表达式。下面通过举例来熟悉用卡诺图化简逻辑函数的方法。

例 3.2.2 一个逻辑电路的输入是 4 个逻辑变量 A 、 B 、 C 、 D , 它的真值表如表 3.2.3 所示,用卡诺图法求化简的与-或表达式及与非-与非表达式。

表 3.2.3 例 3.2.2 的真值表

A	B	C	D	L	A	B	C	D	L
0	0	0	0	1	1	0	0	0	1
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	1
0	0	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	0	0	1
0	1	0	1	1	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	1

解: (1) 由真值表画出卡诺图,如图 3.2.9 所示。

(2) 画包围圈合并最小项,得简化的与-或表达式。

$$L = \overline{C}\overline{D} + A\overline{B}\overline{D} + \overline{A}B\overline{C} + ABCD$$

(3) 求与非-与非表达式。二次求非

$$L = \overline{\overline{\overline{C}\overline{D} + A\overline{B}\overline{D} + \overline{A}B\overline{C} + ABCD}}$$

然后利用摩根定律得

$$L = \overline{\overline{\overline{C}\overline{D}} \cdot \overline{A\overline{B}\overline{D}} \cdot \overline{\overline{A}B\overline{C}} \cdot \overline{ABCD}}$$

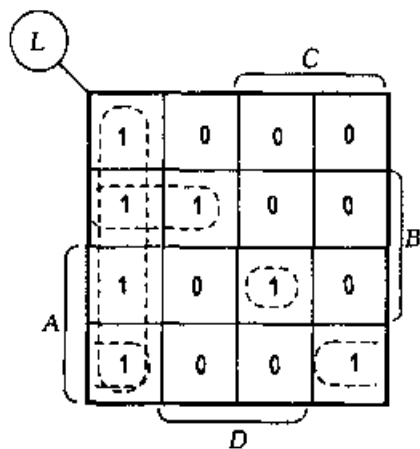


图 3.2.9 例 3.2.2 的卡诺图

利用卡诺图表示逻辑函数式时,如果卡诺图中各小方格被 1 占去了大部分,虽然可用包围 1 的方法进行化简,但由于要重复利用 1 项,往往显得零乱而易出错。这时采用包围 0 的方法化简更为简单。即求出非函数 \overline{L} , 再对 \overline{L} 求非,其结果相同,下面举例说明。

例 3.2.3 化简下列逻辑函数

$$L(A, B, C, D) = \sum m(0 \sim 3, 5 \sim 11, 13 \sim 15)$$

解: (1) 由 L 画出卡诺图,如图 3.2.10a 所示。

(2) 用包围 1 的方法化简,如图 3.2.10b 所示,得

$$L = \overline{B} + C + D$$

(3) 用包围 0 的方法化简,如图 3.2.10c 所示,得

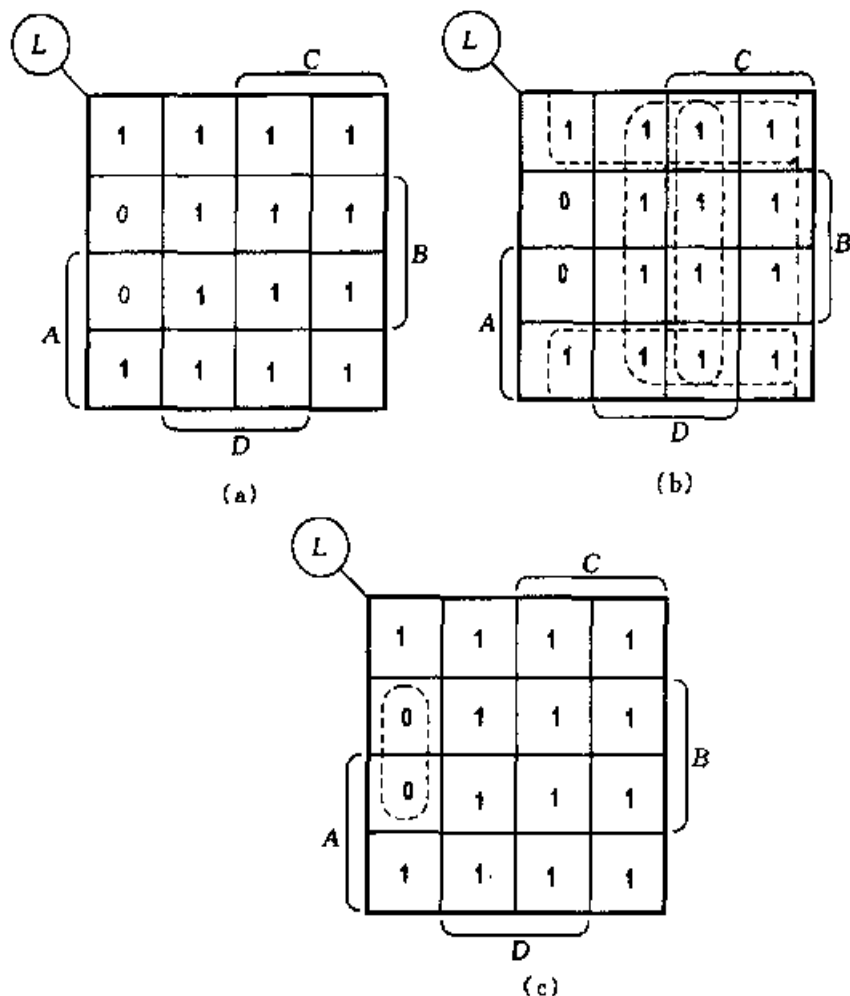


图 3.2.10 例 3.2.3 的卡诺图

$$\bar{L} = B\bar{C}\bar{D}$$

对 \bar{L} 求非, $L = \bar{B} + C + D$ 。

两种方法结果相同。

实际中经常会遇到这样的问题,在真值表内对应于变量的某些取值下,函数的值可以是任意的,或者这些变量的取值根本不会出现,这些变量取值所对应的最小项称为无关项或任意项。

无关项的意义在于,它的值可以取0或取1,具体取什么值,可以根据使函数尽量得到简化而定。下面再举一个利用无关项化简的例子。

例 3.2.4 要求设计一个逻辑电路,能够判断 1 位十进制数是奇数还是偶数,当十进制数为奇数时,电路输出为1,当十进制数为偶数时,电路输出为0。

解: 第一步,列写真值表。用 8421BCD 码表示十进制数,4 位码即为输入变量,当对应的十进制数为奇数时,函数值为1,反之为0,得到表 3.2.4 所示的真值表。

注意,8421BCD 码只有 10 个,表中 4 位二进制码的后 6 种组合是无关的,设

这4位码来自十进制计数器(将在第6章介绍)的输出端,则无关状态根本不会出现,这后6种变量组合就是无关项,它们对应的函数值可以任意假设,为0为1都可以,通常以 \times 表示。

第二步,将真值表的内容填入4变量卡诺图,如图3.2.11所示。

第三步,画包围圈,此时应利用无关项,显然,将 m_{13} 、 m_{15} 、 m_{11} 对应的方格视为1,可以得到最大的包围圈,由此可写出 $L = D$ 。

若不利用无关项, $L = \overline{AD} + \overline{B} \overline{CD}$,结果复杂得多。

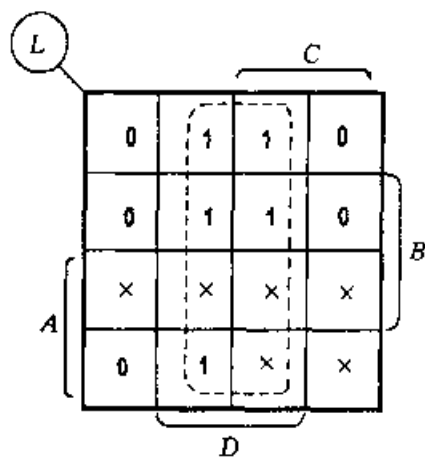


图 3.2.11 例 3.2.4 的卡诺图

表 3.2.4 例 3.2.4 的真值表

对应十进制数	输入变量				输出 L
	A	B	C	D	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
无关项	1	0	1	0	\times
	1	0	1	1	\times
	1	1	0	0	\times
	1	1	0	1	\times
	1	1	1	0	\times
	1	1	1	1	\times

复习思考题

- 3.2.1 4变量和5变量的卡诺图各有几个方格?
- 3.2.2 什么是最小项?
- 3.2.3 什么是无关项?

3.2.4 试列出在逻辑电路设计中使用卡诺图的理由。

3.3 组合逻辑电路的分析

分析组合逻辑电路的目的是为了确定已知电路的逻辑功能,其步骤大致如下:

- (1) 由逻辑图写出各输出端的逻辑表达式;
- (2) 化简和变换各逻辑表达式;
- (3) 列出真值表;
- (4) 根据真值表和逻辑表达式对逻辑电路进行分析,最后确定其功能。

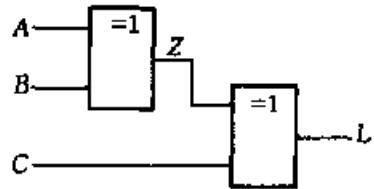


图 3.3.1 例 3.3.1 的逻辑图

下面举例来说明组合逻辑电路的分析方法。

例 3.3.1 已知逻辑电路如图 3.3.1 所示,分析该电路的功能。

解: 第一步,根据逻辑图可写出输出函数的逻辑表达式为

$$L = A \oplus B \oplus C = (A \oplus B) \oplus C$$

第二步,列写真值表。将输入变量 A 、 B 、 C 8 种可能的组合一一列出,为方便起见,表中增加中间变量 $A \oplus B$ 。根据每一组变量取值的情况和上述表达式,分别确定 $A \oplus B$ 的值和 L 值,填入表中,如表 3.3.1 所示。

表 3.3.1 例 3.3.1 的真值表

A	B	C	$A \oplus B$	$L = (A \oplus B \oplus C)$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

第三步,分析真值表后可知,当 A 、 B 、 C 3 个输入变量中取值有奇数个 1 时, L 为 1, 否则 L 为 0。可见该电路可用于检查 3 位二进制码的奇偶性,由于它在输入二进制码含有奇数个 1 时,输出有效信号,因此称为奇校验电路。

例 3.3.2 一个双输入端、双输出端的组合逻辑电路如图 3.3.2 所示,分析该电路的功能。

解: 第一步,由逻辑图写出逻辑表达式,并进行化简和变换。

$$\begin{aligned}
 Z_1 &= \overline{AB} \\
 Z_2 &= \overline{A \cdot \overline{AB}} \\
 Z_3 &= \overline{B \cdot \overline{AB}} \\
 S &= \overline{Z_2 \cdot Z_3} = \overline{Z_2} + \overline{Z_3} \\
 &= A \cdot \overline{AB} + B \cdot \overline{AB} \\
 &= A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) \\
 &= A\overline{B} + \overline{A}B = A \oplus B \\
 C &= \overline{Z_1} = AB
 \end{aligned}$$

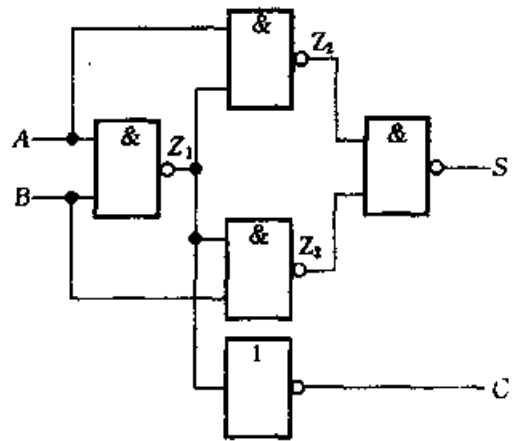


图 3.3.2 例 3.3.2 图

第二步,列写真值表,如表 3.3.2 所示。

表 3.3.2 例 3.3.2 的真值表

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

第三步,分析真值表可知,A、B 都是0时,S 为0,C 也为0;当 A、B 有 1 个为1时,S 为1,C 为0;当 A、B 都是1时,S 为0,C 为1。这符合两个 1 位二进制数相加的原则,即 A、B 为两个加数,S 是它们的和,C 是向高位的进位。这种电路可用于实现两个 1 位二进制数的相加,实际上它是运算器中的基本单元电路,称为半加器。

对于比较简单的组合逻辑电路,有时也可用画波形图的方法进行分析。为了避免出错,通常是根据输入波形,逐级画出输出波形,最后根据逻辑图的输出端与输入端波形之间的关系确定功能。用画波形图的分析法对例 3.3.1 和例 3.3.2 的分析结果分别如图 3.3.3 和图 3.3.4 所示。

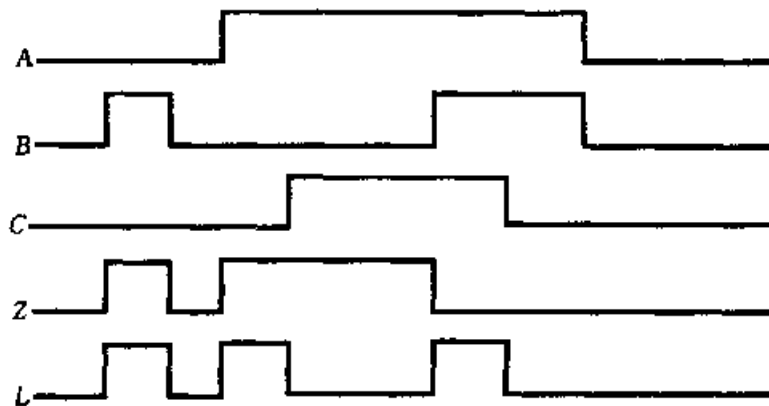


图 3.3.3 例 3.3.1 的波形分析图

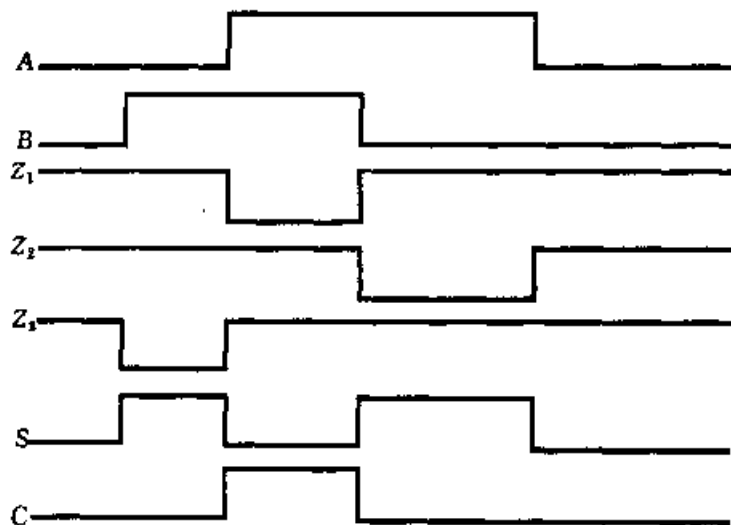


图 3.3.4 例 3.3.2 的波形分析图

复习思考题

- 3.3.1 什么是组合逻辑电路?
- 3.3.2 列出分析组合逻辑电路的步骤。
- 3.3.3 组合逻辑电路都能用波形分析法进行分析吗?

3.4 组合逻辑电路的设计

组合逻辑电路的设计与分析过程相反,其步骤大致如下:

- (1) 根据对电路逻辑功能的要求,列出真值表;
- (2) 由真值表写出逻辑表达式;
- (3) 简化和变换逻辑表达式,从而画出逻辑图。

组合逻辑电路的设计,通常以电路简单,所用器件最少为目标。在前面所介绍的用代数法和卡诺图法来化简逻辑函数,就是为了获得最简的形式,以便能用最少的门电路来组成逻辑电路。但是,由于在设计中普遍采用中、小规模集成电路(一片包括数个门至数十个门)产品,因此应根据具体情况,尽可能减少所用的器件数目和种类,这样可以使组装好的电路结构紧凑,达到工作可靠而且经济的目的。

下面举例说明设计组合逻辑电路的方法和步骤。

例 3.4.1 试用 2 输入与非门和反相器设计一个 3 输入(I_0 、 I_1 、 I_2)、3 输出(L_0 、 L_1 、 L_2)的信号排队电路。它的功能是:当输入 I_0 为 1 时,无论 I_1 和 I_2 为 1 还是 0,输出 L_0 为 1, L_1 和 L_2 为 0;当 I_0 为 0 且 I_1 为 1,无论 I_2 为 1 还是 0,输出

L_1 为1,其余两个输出为0;当 I_2 为1且 I_0 和 I_1 均为0时,输出 L_2 为1,其余两个输出为0。如 I_0 、 I_1 、 I_2 均为0,则 L_0 、 L_1 、 L_2 也均为0。

解:(1) 根据题意列出真值表,如表 3.4.1 所示。

表 3.4.1 例 3.4.1 的真值表

输 入			输 出		
I_0	I_1	I_2	L_0	L_1	L_2
0	0	0	0	0	0
1	×	×	1	0	0
0	1	×	0	1	0
0	0	1	0	0	1

注:×表示可取任意值,即既可取0也可取1。

(2) 根据真值表写出各输出逻辑表达式:

$$L_0 = I_0$$

$$L_1 = \bar{I}_0 I_1$$

$$L_2 = \bar{I}_0 \bar{I}_1 I_2$$

(3) 根据要求将上式变换为与非形式:

$$L_0 = I_0$$

$$L_1 = \overline{\overline{\bar{I}_0 I_1}}$$

$$L_2 = \overline{\overline{\overline{I_0 \bar{I}_1 \cdot I_2}}}$$

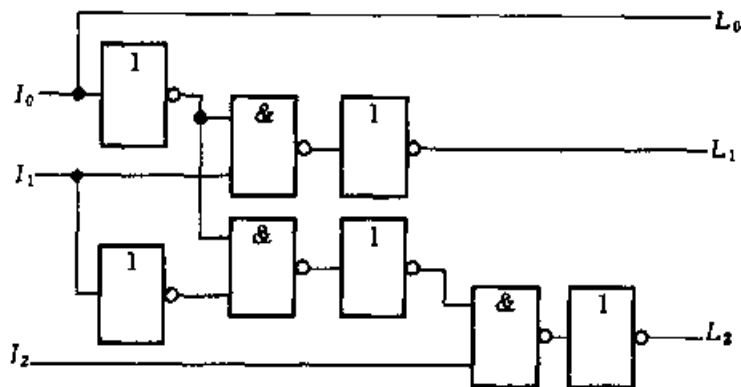


图 3.4.1 例 3.4.1 逻辑图

由此可画出逻辑图,如图 3.4.1 所示。该逻辑电路可用一片内含四个 2 输入端的与非门和另一片内含六个反相器的集成电路组成;也可用两片内含四个 2 输入端与非门的集成电路组成。原逻辑表达式虽然是最简形式,但它需一片反相器和一片 3 输入端的与门才能实现,器件数和种类都不能节省。由此可见,最简的逻辑表达式用一定规格的集成器件实现时,其电路结构不一定是简单

和最经济的。设计逻辑电路时应以集成器件为基本单元,而不应以单个门为单元,这是工程设计与理论分析的不同之处。

该电路可选取四双输入端与非门 74LS00 和六反相器 74LS04 构成,也可以用两片 74LS00。

例 3.4.2 试设计一可逆的 4 位码变换器。在控制信号 $C=1$ 时,它将 8421 码转换为格雷码; $C=0$ 时,它将格雷码转换为 8421 码。

解: (1) 列出真值表,如表 3.4.2 所示。当 $C=1$ 时,输入 $X_3 X_2 X_1 X_0$ 作为 8421 码,对应的输出 $g_3 g_2 g_1 g_0$ 为格雷码;当 $C=0$ 时,输入则作为格雷码,对应的输出 $b_3 b_2 b_1 b_0$ 为 8421 码。此时, $X_3 X_2 X_1 X_0$ 作为格雷码的排列顺序体现在它与 $b_3 b_2 b_1 b_0$ 的一一对应关系上。

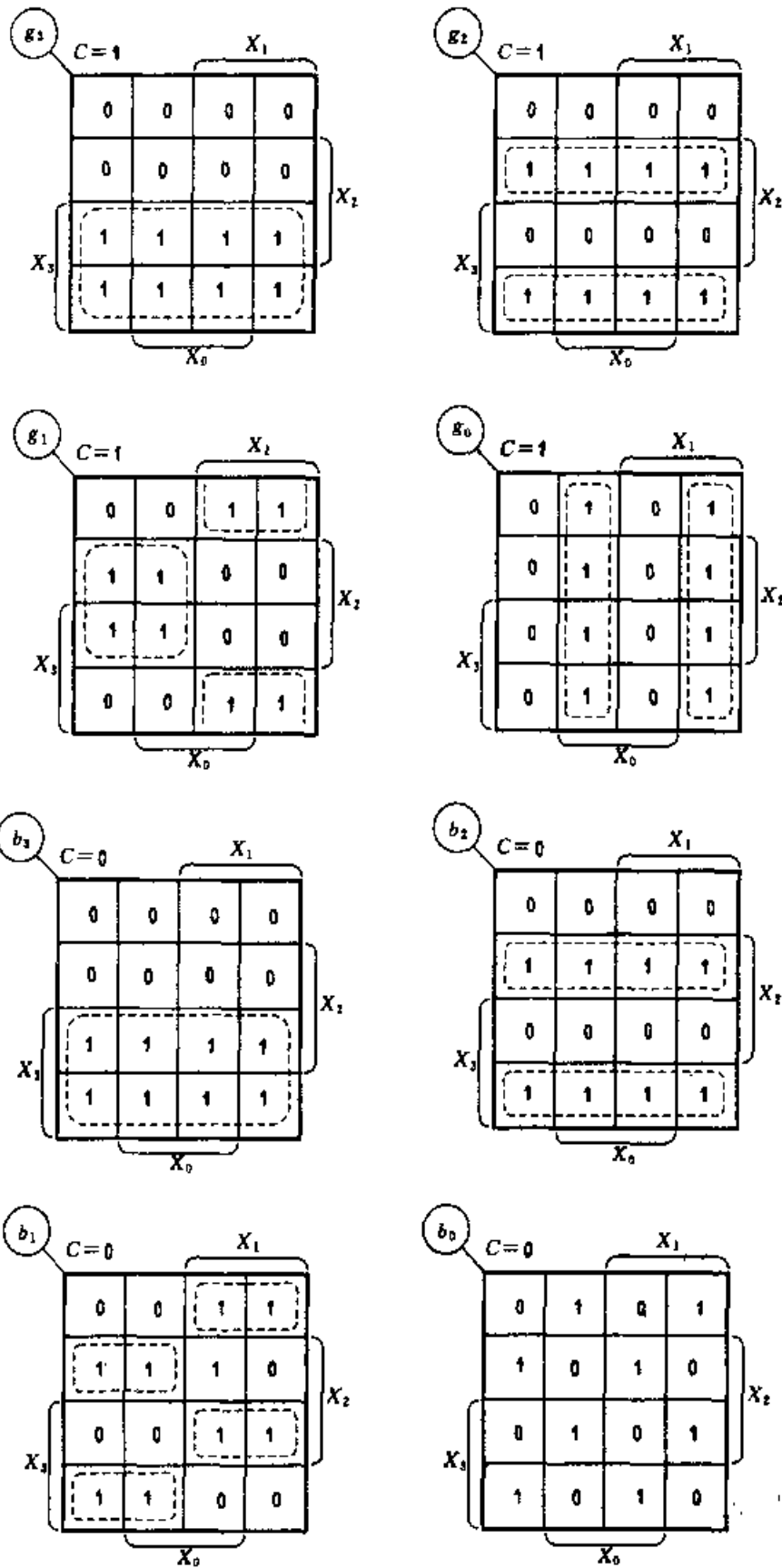
(2) 分别画出 $C=1$ 和 $C=0$ 时各输出函数的卡诺图,如图 3.4.2a 所示。

表 3.4.2 例 3.4.2 中码变换关系真值表

输 入				输 出 ($Y_i = g_i + b_i$)							
X_3	X_2	X_1	X_0	g_3	g_2	g_1	g_0	b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0	1	1	1
0	1	0	1	0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1	0	1	0	0
0	1	1	1	0	1	0	0	0	1	0	1
1	0	0	0	1	1	0	0	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	0	0
1	0	1	1	1	1	1	0	1	1	0	1
1	1	0	0	1	0	1	0	1	0	0	0
1	1	0	1	1	0	1	1	1	0	0	1
1	1	1	0	1	0	0	1	1	0	1	1
1	1	1	1	1	0	0	0	1	0	1	0

(3) 由卡诺图可求得各输出逻辑表达式。若同时考虑 C 变量,当 $C=1$ 时,有

$$\left. \begin{aligned} g_3 &= X_3 C \\ g_2 &= (X_3 \bar{X}_2 + \bar{X}_3 X_2) C = (X_3 \oplus X_2) C \\ g_1 &= (X_2 \bar{X}_1 + \bar{X}_2 X_1) C = (X_2 \oplus X_1) C \\ g_0 &= (X_1 \bar{X}_0 + \bar{X}_1 X_0) C = (X_1 \oplus X_0) C \end{aligned} \right\} (3.4.1)$$



(a)

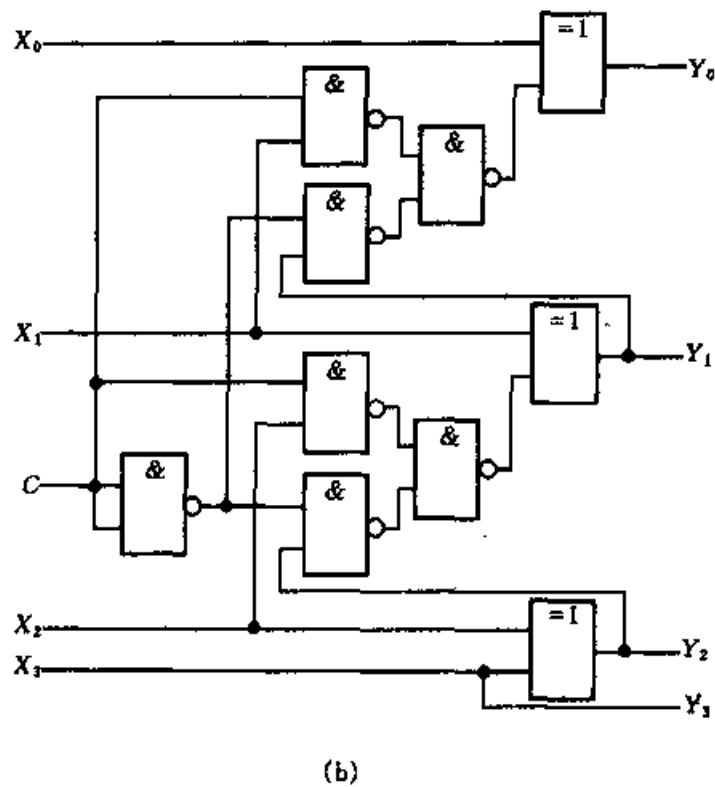


图 3.4.2 例 3.4.2 的卡诺图和逻辑图

(a) 输出函数卡诺图 (b) 逻辑图

当 $C=0$ 时, 有

$$\begin{aligned}
 b_3 &= X_3 \bar{C} \\
 b_2 &= (X_3 \bar{X}_2 + \bar{X}_3 X_2) \bar{C} = (X_3 \oplus X_2) \bar{C} \\
 b_1 &= (X_3 \bar{X}_2 \bar{X}_1 + \bar{X}_3 X_2 \bar{X}_1 + X_3 X_2 X_1 + \bar{X}_3 \bar{X}_2 X_1) \bar{C} \\
 &= [(X_3 \bar{X}_2 + \bar{X}_3 X_2) \bar{X}_1 + (X_3 X_2 + \bar{X}_3 \bar{X}_2) X_1] \bar{C} \\
 &= [(X_3 \oplus X_2) \bar{X}_1 + (\overline{X_3 \oplus X_2}) X_1] \bar{C} \\
 &= (X_3 \oplus X_2 \oplus X_1) \bar{C} \\
 b_0 &= (X_3 \oplus X_2 \oplus X_1 \oplus X_0) \bar{C}
 \end{aligned} \tag{3.4.2}$$

由式(3.4.1)和式(3.4.2)可得总的输出逻辑表达式

$$\begin{aligned}
 Y_3 &= g_3 + b_3 = X_3 C + X_3 \bar{C} = X_3 \\
 Y_2 &= g_2 + b_2 = (X_3 \oplus X_2) C + (X_3 \oplus X_2) \bar{C} = X_3 \oplus X_2 \\
 Y_1 &= g_1 + b_1 = (X_2 \oplus X_1) C + (X_3 \oplus X_2 \oplus X_1) \bar{C} \\
 &= (X_2 \oplus X_1) C + (Y_2 \oplus X_1) \bar{C} \\
 \text{展开且重新组合, 得} \\
 Y_1 &= X_1 \oplus (CX_2 + \bar{C}Y_2) \\
 &= X_1 \oplus (\overline{CX_2 \cdot \bar{C}Y_2}) \\
 Y_0 &= g_0 + b_0 = X_0 \oplus (\overline{CX_1 \cdot \bar{C}Y_1})
 \end{aligned} \tag{3.4.3}$$

由式(3.4.3)可画出如图 3.4.2b 所示的逻辑图。

从以上逻辑表达式和逻辑图可以看出,用异或门代替与门和或门能使逻辑电路比较简单。在化简和变换逻辑表达式时,应尽可能使某些输出作为另一些输出的条件,例如,利用 Y_2 作为 Y_1 的一个输入, Y_1 又作为 Y_0 的一个输入,这样可以使电路更简单。在化简时,若注意综合考虑,使式中的相同项尽可能多,则可以使电路得到简化。此外,我们还将与或换成与非形式,从而减少了门电路的种类。该逻辑电路可由两片各含四个 2 输入端的与非门(7400)和一片内含四个异或门(7486)的集成电路组成。

复习思考题

3.4.1 列出设计组合逻辑电路的步骤。

3.4.2 为什么说在组合逻辑电路设计中正确列出真值表是最关键的一步?

3.5 组合逻辑电路中的竞争冒险

前面分析组合逻辑电路时,都没有考虑门电路的延迟时间对电路产生的影响。实际上,从信号输入到稳定输出需要一定的时间。由于从输入到输出的过程中,不同通路上门的级数不同,或者门电路平均延迟时间的差异,使信号从输入经不同通路传输到输出级的时间不同。由于这个原因,可能会使逻辑电路产生错误输出。通常把这种现象称为竞争冒险。

3.5.1 产生竞争冒险的原因

首先来分析图 3.5.1 所示电路的工作情况,以建立竞争冒险的概念。在图 3.5.1a 中,与门 G_2 的输入是 A 和 \bar{A} 两个互补信号。由于 G_1 的延迟, \bar{A} 的下

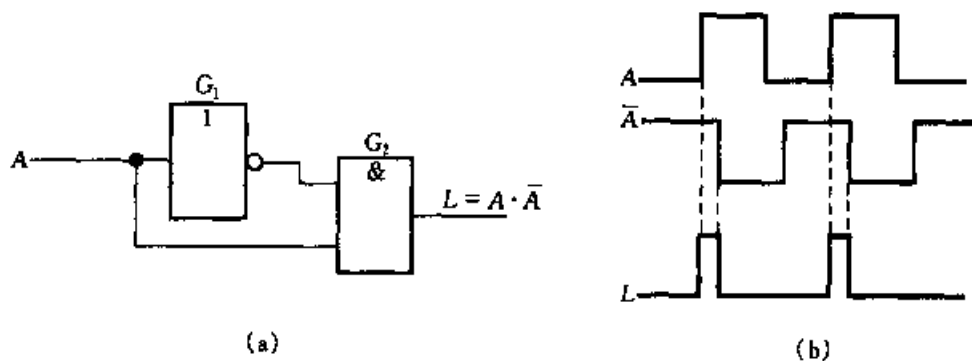


图 3.5.1 产生正跳变脉冲的竞争冒险

(a) 逻辑电路 (b) 工作波形

降沿要滞后于 A 的上升沿,因此在很短的时间间隔内, G_2 的两个输入端都会出现高电平,致使它的输出出现一个高电平窄脉冲(它是按逻辑设计要求不应出现的干扰脉冲),如图 3.5.1b 所示。与门 G_2 的 2 个输入信号分别由 G_1 和 A 端两个路径在不同的时刻到达的现象,通常称为竞争,由此而产生输出干扰脉冲的现象称为冒险。

下面进一步分析组合逻辑电路产生竞争冒险的原因。设有一个逻辑电路如图 3.5.2a 所示,其工作波形如图 3.5.2b 所示。它的输出逻辑表达式为 $L = AC + B\bar{C}$ 。由此式可知,当 A 和 B 都为 1 时, $L = 1$,与 C 的状态无关。但是,由图 3.5.2b 可以看出,在 C 由 1 变 0 时, \bar{C} 由 0 变 1 有一延迟时间,在这个时间间隔内, G_2 和 G_3 的输出 AC 和 $B\bar{C}$ 同时为 0,而使输出出现一负跳变的窄脉冲,即冒险现象。这是产生竞争冒险的原因之一,其他原因这里不作详述。

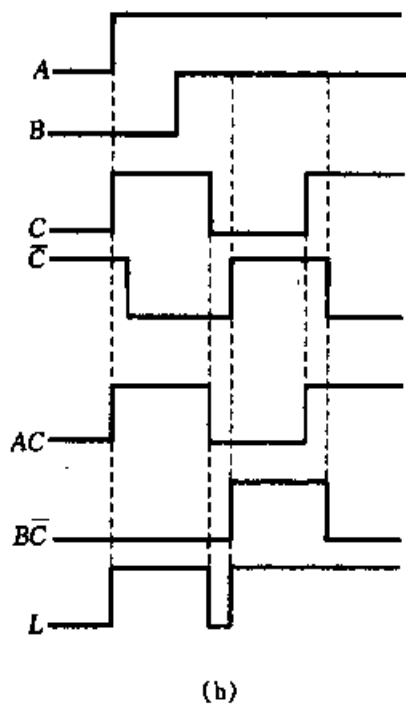
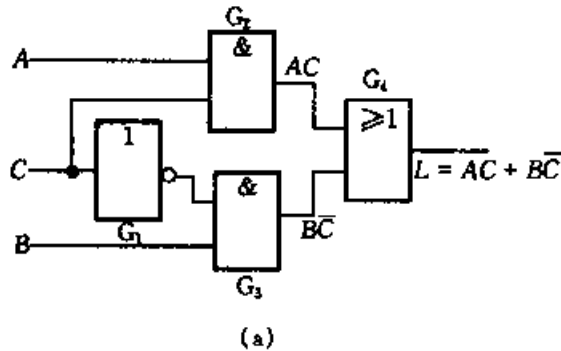


图 3.5.2 产生负跳变脉冲的竞争冒险

(a) 逻辑电路 (b) 工作波形

由以上分析可知,当电路中存在由反相器产生的互补信号,且在互补信号的状态发生变化时可能出现冒险现象。

3.5.2 消去竞争冒险的方法

针对上述原因,可以采取以下措施消去竞争冒险现象。

1. 发现并消掉互补变量

例如,函数式 $F = (A + B)(\bar{A} + C)$, 在 $B = C = 0$ 时, $F = A\bar{A}$ 。若直接根据这个逻辑表达式组成逻辑电路,则可能出现竞争冒险。可以将该式变换为 $F = AC + \bar{A}B + BC$, 这里已将 $A\bar{A}$ 消掉。根据这个表达式组成逻辑电路就不会出现竞争冒险。

2. 增加乘积项

对于图 3.5.2a 所示的逻辑电路,可以根据 3.2 节中所述的常用恒等式,在其输出逻辑表达式中增加乘积项 AB ^①。这时, $L = AC + B\bar{C} + AB$, 对应的逻辑电路如图 3.5.3 所示。由图 3.5.2b 可以看出,出现负跳变窄脉冲处,正是 A 和 B 均为 1 时。显然,对于图 3.5.3 所示电路,当 $A = B = 1$ 时, G_5 输出为 1, G_4 输出亦为 1,这就消除了 C 跳变时对输出状态的影响,从而消去了竞争冒险。

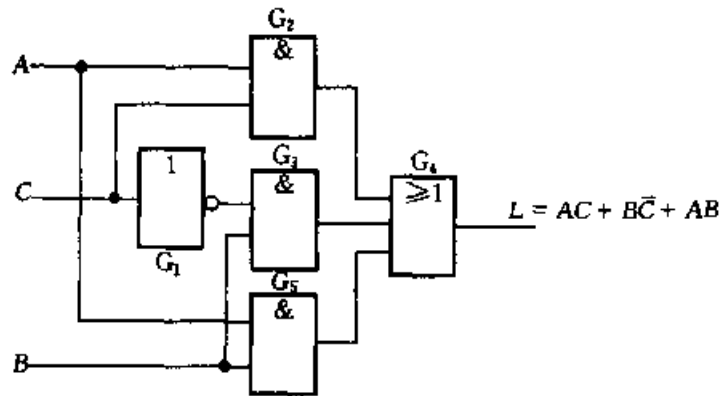


图 3.5.3 增加了乘积项 AB 的逻辑电路

3. 输出端并联电容器

如果逻辑电路在较慢速度下工作,为了消去竞争冒险,可以在输出端并联电容器,其容量为 $4 \sim 20$ pF 之间。如图 3.5.4a 所示,即在图 3.5.2a 所示电路的输出端并联电容 C 。由于或门 G_4 存在一输出电阻 R_o ,致使输出波形上升沿和下降沿变化比较缓慢。因此对于很窄的负跳变脉冲起到平波的作用,如图 3.5.4b 所示。显然,这时在输出端不会出现逻辑错误。

① 这类乘积项称为蕴含项。

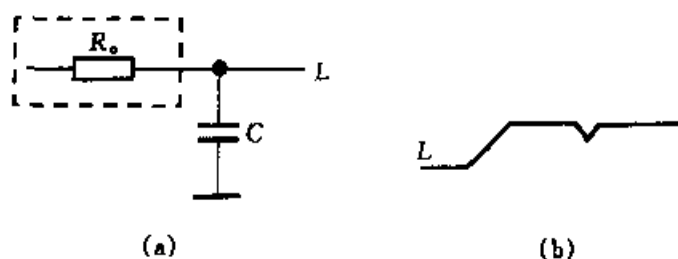


图 3.5.4 并联电容器消去竞争冒险

(a) 电路 (b) 输出波形

以上介绍了产生竞争冒险的原理和克服竞争冒险的方法。要能很好地解决这一问题,还必须在实践中积累和总结经验。

复习思考题

3.5.1 什么是组合逻辑电路中的竞争冒险?

3.5.2 列出三种消去组合逻辑电路竞争冒险的方法。

* 3.6 CAD 例题

例 CE3.6.1 试用 ISP Synario 软件对图 CE3.6.1(即图题 3.3.5)所示逻辑电路进行分析,并给出逻辑功能仿真波形图。然后说明电路的功能。

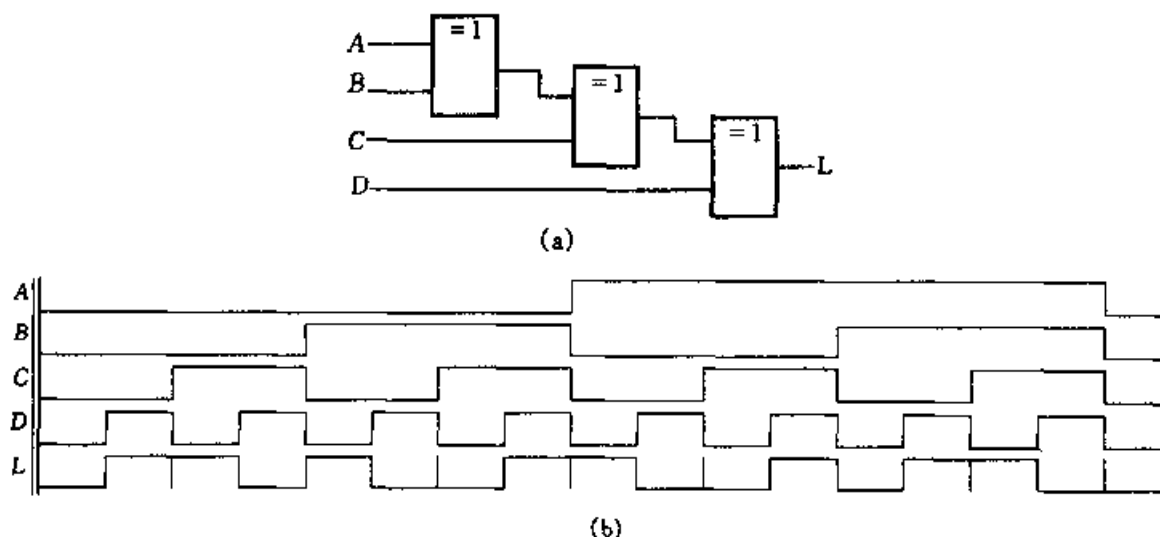


图 CE3.6.1 逻辑电路图及其仿真波形图

(a) 电路图 (b) 仿真波形图

解: (1) 输入设计文件。

打开 ISP Synario 软件的原理图编辑器,输入图 CE3.6.1a 所示逻辑电路,并保存文件。然后对原理图文件进行编译和逻辑简化。

(2) 编写并输入测试向量文件。

根据图 CE3.6.1a 所示逻辑电路,编写的测试向量文件如下:

```

module      CE3_6_1                "模块开始
            A,B,C,D                pin;    "定义输入信号
            L                       pin istype 'com'; "定义输出信号
            X = .X.;                "定义特殊常数
test-vectors                                "测试向量段开始
            ([A,B,C,D]->L)
            ^ B0000 -> X;
            ^ B0001 -> X;
            ^ B0010 -> X;
            ^ B0011 -> X;
            ^ B0100 -> X;
            ^ B0101 -> X;
            ^ B0110 -> X;
            ^ B0111 -> X;
            ^ B1000 -> X;
            ^ B1001 -> X;
            ^ B1010 -> X;
            ^ B1011 -> X;
            ^ B1100 -> X;
            ^ B1101 -> X;
            ^ B1110 -> X;
            ^ B1111 -> X;
            ^ B0000 -> X;
end

```

(3) 进行逻辑功能仿真。对测试向量文件进行编译和逻辑功能仿真,得到如图 CE3.6.1b 所示的仿真波形图。

图中 L 是输出波形,其中的小竖线是由于输入信号 A 、 B 、 C 、 D 在其变化的瞬间不能同时到达最后一个异或门所致,即竞争冒险。

(4) 根据仿真波形可知,该电路是一个 4 位的奇数校验器。即当 4 位数 A 、 B 、 C 、 D 中有奇数个 1 时输出 L 为 0,否则输出 L 为 1。

例 CE3.6.2 试用 ABEL 语言设计一个 4 输入、4 输出逻辑电路。当控制信号 $C=0$ 时,输出状态与输入状态相反; $C=1$ 时,输出状态与输入状态相同。

解: (1) 逻辑设计

设输入变量为 E, F, G, H , 输出变量为 $L1, L2, L3, L4$ 。

根据题意, 当 $C=0$ 时, $L1, L2, L3, L4$ 的状态分别与 E, F, G, H 的状态相反; 当 $C=1$ 时, $L1, L2, L3, L4$ 的状态分别与 E, F, G, H 的状态相同。于是可以列出表示输入、输出逻辑关系的真值表。为节省篇幅, 下面仅列出 $L1$ 与 E 的真值表, 如表例 3.6.2 所示。

表例 CE3.6.2

C	E	$L1$	C	E	$L1$
0	0	1	1	0	0
0	1	0	1	1	1

根据真值表可以列出逻辑方程如下: $L1 = E \cdot C + \bar{E} \cdot \bar{C} = E \odot C$

同理可得: $L2 = F \cdot C + \bar{F} \cdot \bar{C} = F \odot C$

$$L3 = G \cdot C + \bar{G} \cdot \bar{C} = G \odot C$$

$$L4 = H \cdot C + \bar{H} \cdot \bar{C} = H \odot C$$

(2) 输入设计文件

使用 ISP Synario 的文本编辑器, 输入表示以上逻辑关系的 ABEL 程序, 即将逻辑表达式中“同或”符号 \odot 用 ABEL 语言中“同或”符号 $!\$$ 表示。

完整的 ABEL 语言程序(包括部分测试向量)如下:

```

module CE3_6_2                                "模块开始
  C,E,F,G,H      pin;                        "定义输入信号
  L1,L2,L3,L4    pin istype'com';          "定义输出信号
  X = .X.;      "定义特殊常数
equations      "逻辑方程段开始
  L1 = E!$C; "L1 = (E & C) # (!E & !C)
  L2 = F!$C;
  L3 = G!$C;
  L4 = H!$C;
test_vectors      "测试向量段开始
  ([C,E] -> L1)
  [0,0] -> X;
  [0,1] -> X;
  [1,0] -> X;
  [1,1] -> X;
test_vectors      "测试向量段开始

```

$([C, F] \rightarrow L2)$

$[0, 0] \rightarrow X;$

$[0, 1] \rightarrow X;$

$[1, 0] \rightarrow X;$

$[1, 1] \rightarrow X;$

end

(3) 编译 ABEL 源文件, 并进行逻辑功能仿真

对 ABEL 源文件进行编译、逻辑简化, 然后对测试向量进行编译和功能仿真, 得到如图 CE3.6.2 所示的仿真波形。鉴于 L3、L4 的逻辑功能和仿真波形与 L1、L2 类似, 为节省篇幅, 图中只给出了 L1、L2 的波形。

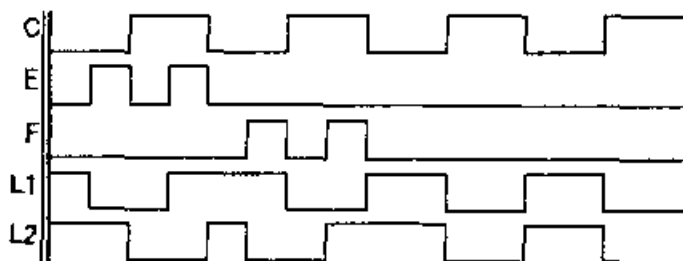


图 CE3.6.2 L1、L2 的逻辑功能仿真波形图

小 结

• 逻辑代数是分析和设计逻辑电路的工具。一个逻辑问题可用逻辑函数来描述。逻辑函数可用真值表、逻辑表达式、卡诺图和逻辑图表达, 这 4 种表达方方式各具特点, 可根据需要选用。

• 组合逻辑电路的输出状态只决定于同一时刻的输入状态, 它可由逻辑门电路以及可编程器件(PLD)、现场可编程门阵列(FPGA)、只读存储器(ROM)等(将在第 8 章讨论)组成。

• 分析组合逻辑电路的目的是确定已知电路的逻辑功能, 其步骤大致是: 写出各输出端的逻辑表达式→化简和变换逻辑表达式→列出真值表→确定功能。

• 应用逻辑门电路设计组合逻辑电路的步骤大致是: 列出真值表→写出逻辑表达式(或填写卡诺图)→逻辑化简和变换→画出逻辑图。

习 题

3.1.1 用真值表证明下列恒等式:

(a) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

$$(b) (A+B)(A+C) = A+BC$$

$$(c) \overline{A \oplus B} = \overline{A} \overline{B} + AB$$

3.1.2 用逻辑代数定律证明下列等式:

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

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

$$(c) A + A\overline{B}\overline{C} + \overline{A}CD + (\overline{C} + \overline{D})E = A + CD + E$$

3.1.3 用代数法化简下列各式:

$$(a) AB(BC + A)$$

$$(b) (A+B)(A\overline{B})$$

$$(c) \overline{ABC}(B + \overline{C})$$

$$(d) A + ABC + A\overline{BC} + CB + C\overline{B}$$

$$(e) \overline{AB + A\overline{B} + AB + A\overline{B}}$$

$$(f) \overline{(A+B)} + \overline{(A+B)} + \overline{(AB)}(\overline{AB})$$

$$(g) (A+B+\overline{C})(A+B+C)$$

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

$$(i) \overline{AB} + (A + \overline{B})$$

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

$$(k) ABC\overline{D} + ABD + BC\overline{D} + ABCD + B\overline{C}$$

$$(l) \overline{AC + \overline{ABC} + \overline{BC} + ABC}$$

$$(m) \overline{AB + ABC + A(B + A\overline{B})}$$

3.1.4 将下列各式转换成与-或形式:

$$(a) \overline{A \oplus B \oplus C \oplus D}$$

$$(b) \overline{A+B+C+D} + \overline{C+D+A+D}$$

$$(c) \overline{AC \cdot BD} \quad \overline{BC \cdot AB}$$

3.1.5 将题 3.1.4 中(a)和(b)转换成“与-非”形式。

3.1.6 逻辑函数表达式为 $L = \overline{ABC\overline{D}}$, 画出实现该式的逻辑电路, 限使用 2 输入与非门和反相器

3.1.7 利用与非门实现下列函数:

$$(a) L = AB + AC$$

$$(b) L = \overline{D}(A + C)$$

$$(c) L = \overline{(A+B)(C+D)}$$

3.1.8 利用或非门实现下列函数:

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

$$(b) L = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$

3.2.1 将下列函数展开为最小项表达式:

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

$$(b) L = \overline{A}\overline{B} + ABD(B + \overline{CD})$$

3.2.2 用卡诺图法化简下列各式:

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

(b) $A\overline{B}CD + AB\overline{C}D + A\overline{B} + A\overline{D} + A\overline{B}C$

(c) $(\overline{A}\overline{B} + B\overline{D})\overline{C} + BD(\overline{A}\overline{C}) + \overline{D}(\overline{A} + \overline{B})$

(d) $A\overline{B}CD + D(\overline{B}\overline{C}D) + (A + C)B\overline{D} + \overline{A}(\overline{B} + C)$

(e) $L(A, B, C, D) = \sum m(3, 4, 5, 6, 9, 10, 12, 13, 14, 15)$

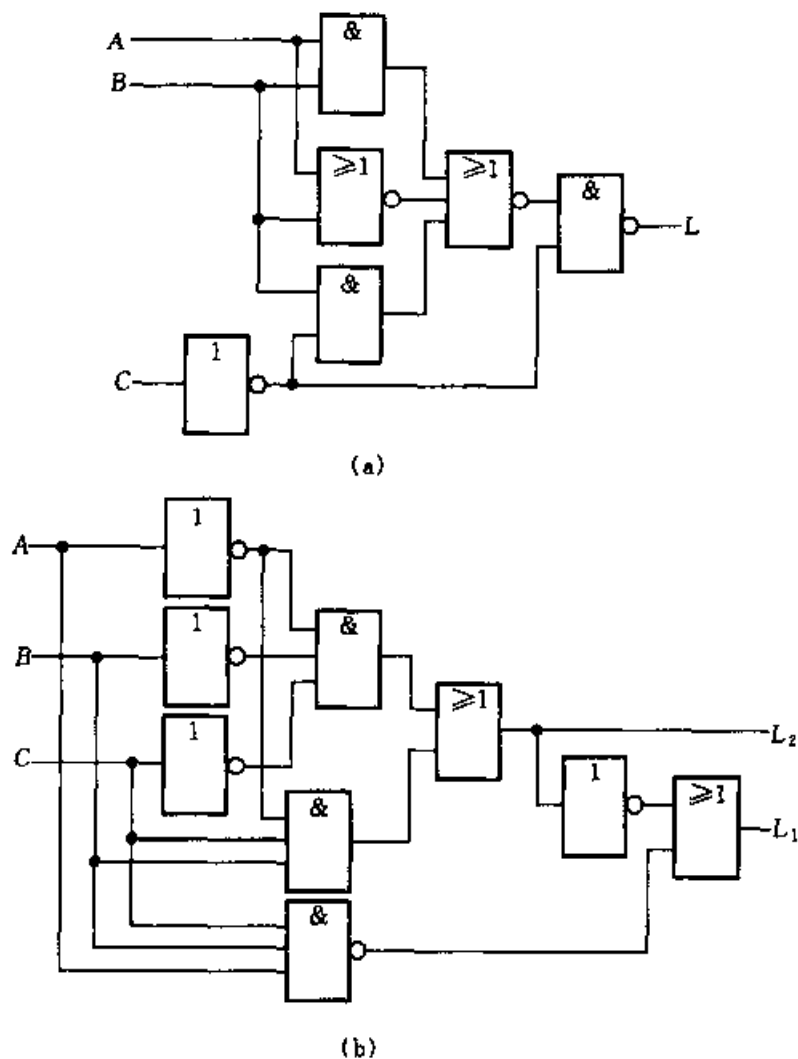
(f) $L(A, B, C, D) = \sum m(0, 1, 2, 5, 6, 7, 8, 9, 13, 14)$

(g) $L(A, B, C, D) = \sum m(0, 2, 4, 6, 9, 13) + \sum d(1, 3, 5, 7, 11, 15)$

(h) $L(A, B, C, D) = \sum m(0, 13, 14, 15) + \sum d(1, 2, 3, 9, 10, 11)$

3.2.3 已知逻辑函数 $L = A\overline{B} + B\overline{C} + C\overline{A}$, 试用真值表、卡诺图和逻辑图(与非-与非)表示。

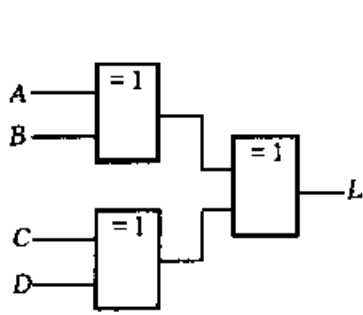
3.3.1 写出如图题 3.3.1 所示电路对应的真值表。



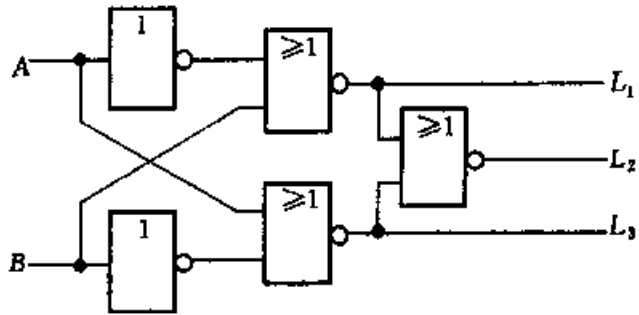
图题 3.3.1

3.3.2 试分析图题 3.3.2 所示逻辑电路的功能。

3.3.3 逻辑电路如图题 3.3.3 所示, 试分析其逻辑功能。

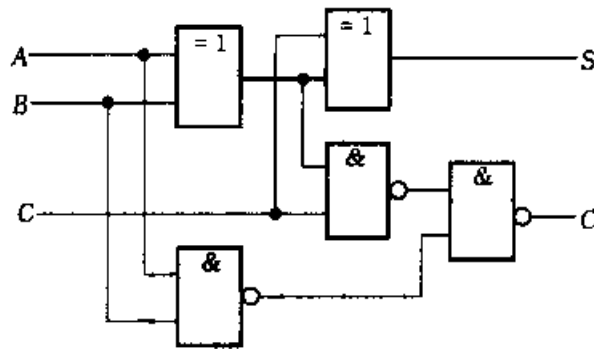


图题 3.3.2



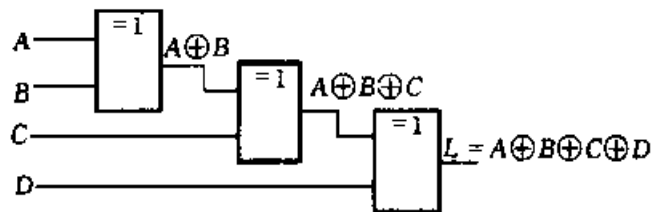
图题 3.3.3

3.3.4 试分析图题 3.3.4 所示逻辑电路的功能。



图题 3.3.4

3.3.5 分析图题 3.3.5 所示逻辑电路的功能。



图题 3.3.5

3.3.6 分析图题 3.3.6 所示逻辑电路的功能。

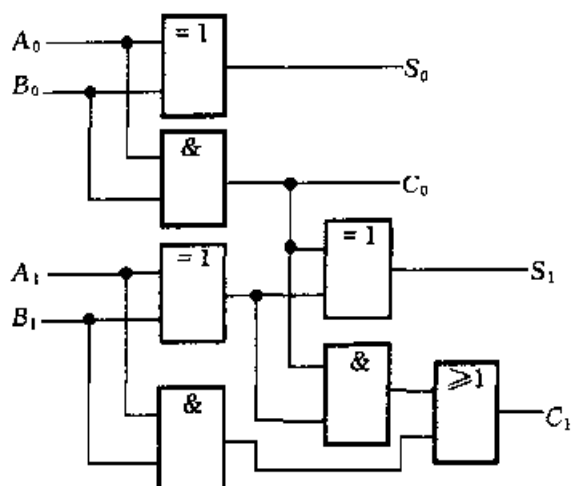
3.3.7 设有 4 种组合逻辑电路,它们的输入波形(A、B、C、D)如图题 3.3.7 所示,其对应的输出波形分别为 W、X、Y、Z,试分别写出它们简化的逻辑表达式。

3.4.1 逻辑功能由最小项表达式 $L = \sum m(1, 3, 4, 7, 13, 14, 15)$ 表示。

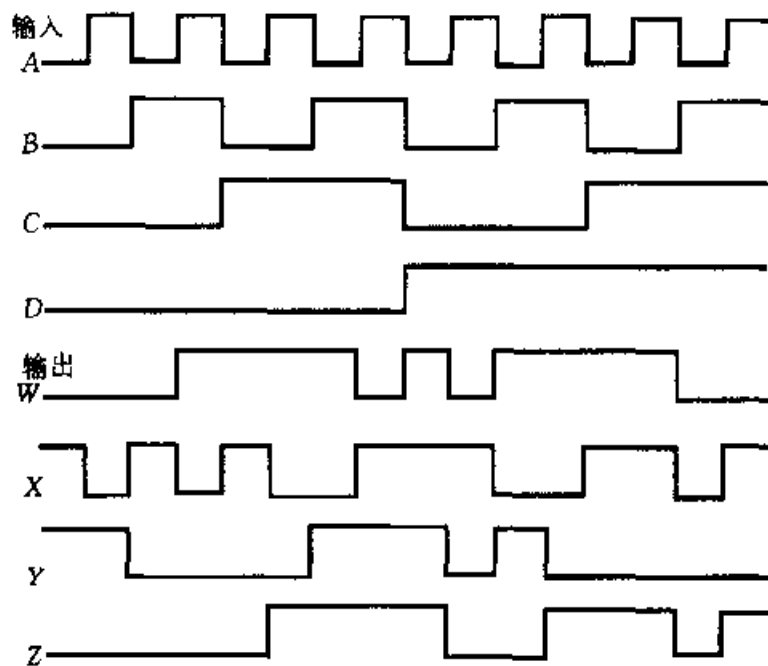
- (a) 用与或非门 7454 构成;
- (b) 用最简便的方法,采用与非门构成;
- (c) 用最简便的方法,采用或非门构成。

3.4.2 逻辑功能由最小项表达式 $L = \sum m(2, 4, 8, 9, 10, 12, 14)$ 表示。

- (a) 用最简便的方式,用与或非门构成;
- (b) 用最简便的方式,用与非门构成;



图题 3.3.6



图题 3.3.7

(c) 采用最简便的方式,用或非门构成。

3.4.3 试用 2 输入与非门和反相器设计一个 4 位的奇偶校验器,即当 4 位数中有奇数个 1 时输出为 0, 否则输出为 1。

3.4.4 试设计一个 4 输入、4 输出逻辑电路。当控制信号 $C=0$ 时,输出状态与输入状态相反; $C=1$ 时,输出状态与输入状态相同。

3.4.5 设计一个电路,能实现表题 3.4.5 所示的逻辑功能,请选用合适的 SSI 门电路时,尽可能做到种类少,数目少。

表题 3.4.5 的真值表

A	B	C	L_1	L_2
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

3.4.6 设计 2 位二进制数相加的逻辑电路。提示：

$$\begin{array}{r}
 A_1 \quad A_0 \\
 + \quad B_1 \quad B_0 \\
 \hline
 C_1 \quad S_1 \quad S_0
 \end{array}$$

A_1 、 A_0 和 B_1 、 B_0 分别为被加数和加数， S_1 、 S_0 为相加的和， C_1 为进位位。

3.4.7 某雷达站有 3 部雷达 A、B、C，其中 A 和 B 功率消耗相等，C 的功率是 A 的两倍。这些雷达由两台发电机 X 和 Y 供电，发电机 X 的最大输出功率等于雷达 A 的功率消耗，发电机 Y 的最大输出功率是 X 的 3 倍。要求设计一个逻辑电路，能够根据各雷达的启动和关闭信号，以最节约电能的方式启、停发电机。

* CAD 习题

C3.6.1 试用 ISP Synario 软件对图题 3.3.3 所示逻辑电路进行分析，并给出逻辑功能仿真波形图。然后说明电路的逻辑功能。

C3.6.2 试用 ISP Synario 软件对图题 3.3.4 所示逻辑电路进行分析，并给出逻辑功能仿真波形图。然后说明电路的功能。

C3.6.3 试用 ISP Synario 软件对图题 3.3.6 所示逻辑电路进行分析，并给出逻辑功能仿真波形图。然后说明电路的逻辑功能。

4 常用组合逻辑功能器件

引言 随着微电子技术的不断发展,单片集成器件所具有的逻辑功能越来越复杂,种类也越来越多。

从本章开始,要接触中、大规模集成组合逻辑电路。将介绍编码器和译码器、数据选择器和数据分配器、数值比较器、算术/逻辑运算单元。对于这些常用的集成组合逻辑电路,着重分析它们的功能及基本的应用方法。在学习这些内容的过程中,要以前面所学习的真值表、逻辑代数、卡诺图等作为工具,进行逻辑电路的分析与设计。

4.1 编 码 器

4.1.1 编码器的定义与功能

1. 4 线 - 2 线编码器

在数字系统里,常常需要将某一信息(输入)变换为某一特定的代码(输出)。把二进制码按一定的规律编排,例如 8421 码、格雷码等,使每组代码具有一特定的含义(代表某个数或控制信号)称为编码。具有编码功能的逻辑电路称为编码器。编码器有若干个输入,在某一时刻只有一个输入信号被转换为二进制码。例如 8 线 - 3 线编码器和 10 线 - 4 线编码器分别有 8 输入、3 位二进制码输出和 10 输入、4 位二进制码输出。下面分析 4 输入、2 位二进制码输出的编码器的工作原理。

4 线 - 2 线编码器的功能表如表 4.1.1 所示。

表 4.1.1 4 线 - 2 线编码器功能表

输 入				输 出	
I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

表 4.1.1 所示的编码器为高电平输入有效,因而可由功能表得到如下逻辑表达式:

$$Y_1 = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

$$Y_0 = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

根据逻辑表达式画出逻辑图如图 4.1.1 所示。该逻辑电路可以实现如表 4.1.1 所示的功能,即当 $I_0 \sim I_3$ 中在某一个输入为 1,输出 $Y_1 Y_0$ 即为相对应的代码,例如当 I_1 为 1 时, $Y_1 Y_0$ 为 01。这里还有一个问题请读者注意。当 I_0 为 1, $I_1 \sim I_3$ 都为 0 和 $I_0 \sim I_3$ 均为 0 时 $Y_1 Y_0$ 都是 00,而这两种情况在实际中是必须加以区分的,这个问题留待后面加以解决。当然,编码器也可以设计为低电平有效。

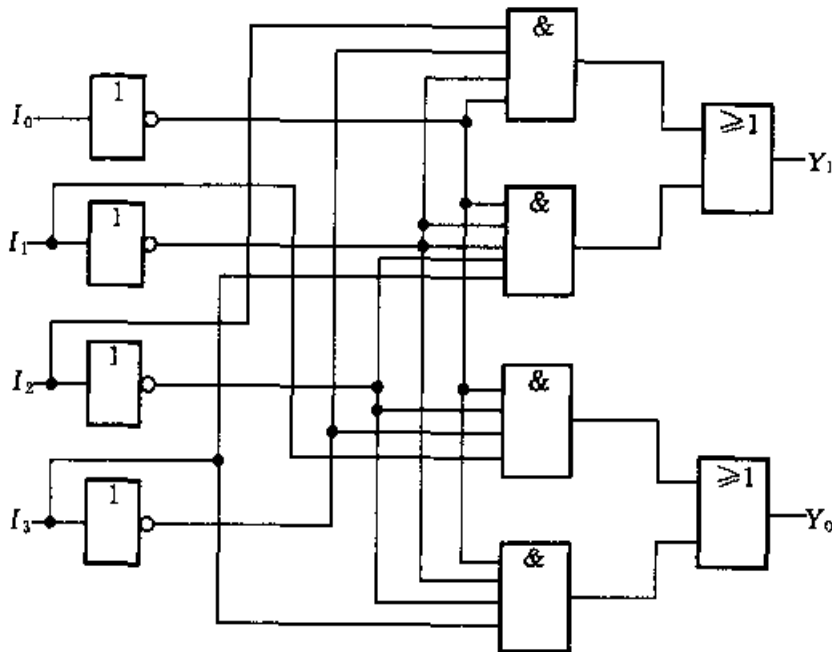


图 4.1.1 4 线 - 2 线编码器逻辑图

2. 键盘输入 8421BCD 码编码器

计算机的键盘输入逻辑电路就是由编码器组成。图 4.1.2 是用十个按键和门电路组成的 8421 码编码器,其功能如表 4.1.2 所示,其中 $S_0 \sim S_9$ 代表十个按键,即对应十进制数 0~9 的输入键,它们对应的输出代码正好是 8421BCD 码,同时也把它们作为逻辑变量, $ABCD$ 为输出代码 (A 为最高位), GS 为控制使能标志。

对功能表和逻辑电路进行分析,都可得知:① 该编码器为输入低电平有效;② 在按下 $S_0 \sim S_9$ 中任意一个键时,即输入信号中有一个为有效电平时 $GS = 1$,代表有信号输入,而只有 $S_0 \sim S_9$ 均为高电平时 $GS = 0$,代表无信号输入,此时的输出代码 0000 为无效代码。由此解决了前面提出的如何区分两种情况下输出都是全 0 的问题。

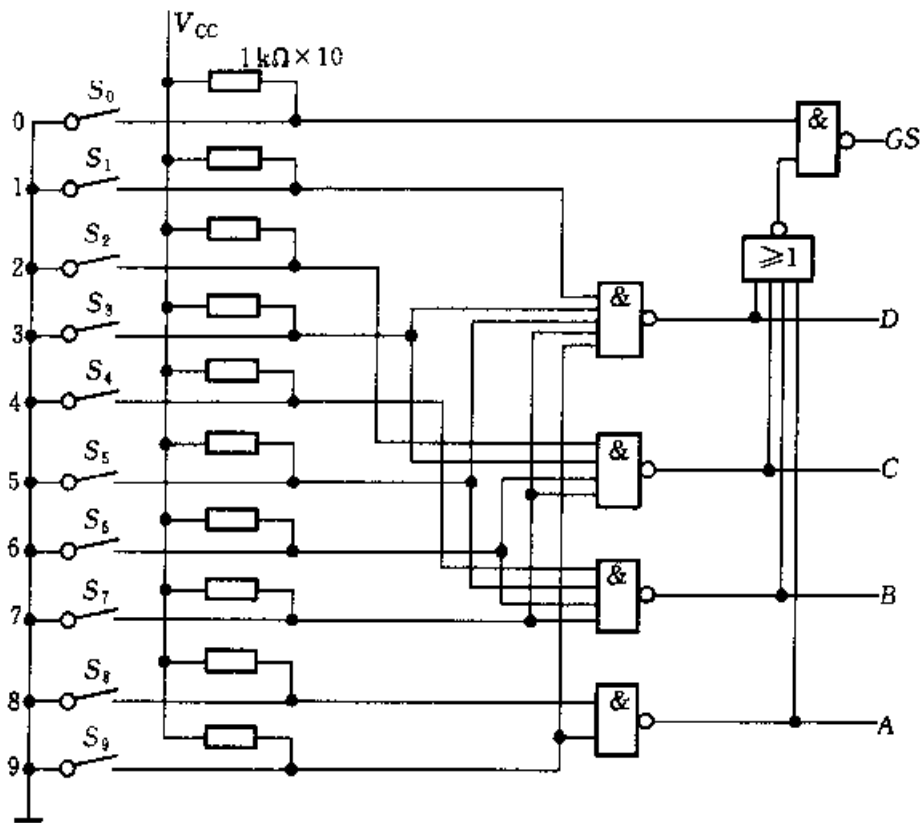


图 4.1.2 用十个按键和门电路组成的 8421BCD 码编码器

表 4.1.2 十个按键 8421BCD 码编码器功能表

输 入										输 出				
S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0	A	B	C	D	GS
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
1	1	1	1	1	1	1	0	1	1	0	0	1	0	1
1	1	1	1	1	1	0	1	1	1	0	0	1	1	1
1	1	1	1	1	0	1	1	1	1	0	1	0	0	1
1	1	1	1	0	1	1	1	1	1	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	0	0	1	1

3. 优先编码器

上述机械式按键编码电路虽然比较简单,但当同时按下两个或更多个键时,其输出将是混乱的。在数字系统中,特别是在计算机系统中,常常要控制几个工作对象,例如微型计算机主机要控制打印机、磁盘驱动器、输入键盘等。当某个部件需要实行操作时,必须先送一个信号给主机(称为服务请求),经主机识别后再发出允许操作信号(服务响应),并按事先编好的程序工作。这里会有几个部件同时发出服务请求的可能,而在同一时刻只能给其中1个部件发出允许操作信号。因此,必须根据轻重缓急,规定好这些控制对象允许操作的先后次序,即优先级别。识别这类请求信号的优先级别并进行编码的逻辑部件称为优先编码器。4线-2线优先编码器的功能表如表4.1.3所示。

表 4.1.3 4线-2线优先编码器功能表

输 入				输 出	
I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
×	1	0	0	0	1
×	×	1	0	1	0
×	×	×	1	1	1

分析表4.1.3中 $I_0 \sim I_3$ 的优先级别。例如,对于 I_0 ,只有当 I_1, I_2, I_3 均为0,即均无有效电平输入,且 I_0 为1时,输出为00。对于 I_3 ,无论其他3个输入是否为有效电平输入,输出均为11。由此可知 I_3 的优先级别高于 I_0 的优先级别,且这4个输入的优先级别的高低次序依次为 I_3, I_2, I_1, I_0 。

由表4.1.3可以得出该优先编码器的逻辑表达式为

$$Y_1 = I_2 \bar{I}_3 + I_3$$

$$Y_0 = I_1 \bar{I}_2 \bar{I}_3 + I_3$$

由于这里包括了无关项,逻辑表达式比前面介绍的非优先编码器简单些。

4.1.2 集成电路编码器

这里介绍两种常用的集成电路优先编码器74147和74148,它们都有TTL和CMOS(74HC147、74HC148)的定型产品。74147和74HC147,74148和74HC148在逻辑功能上没有区别,只是电性能参数不同。以下只分析逻辑功能及介绍它们的应用方法。

1. 8线-3线优先编码器74148

优先编码器74148的逻辑图和引脚图分别如图4.1.3a和4.1.3b所示,其功能如表4.1.4所示。

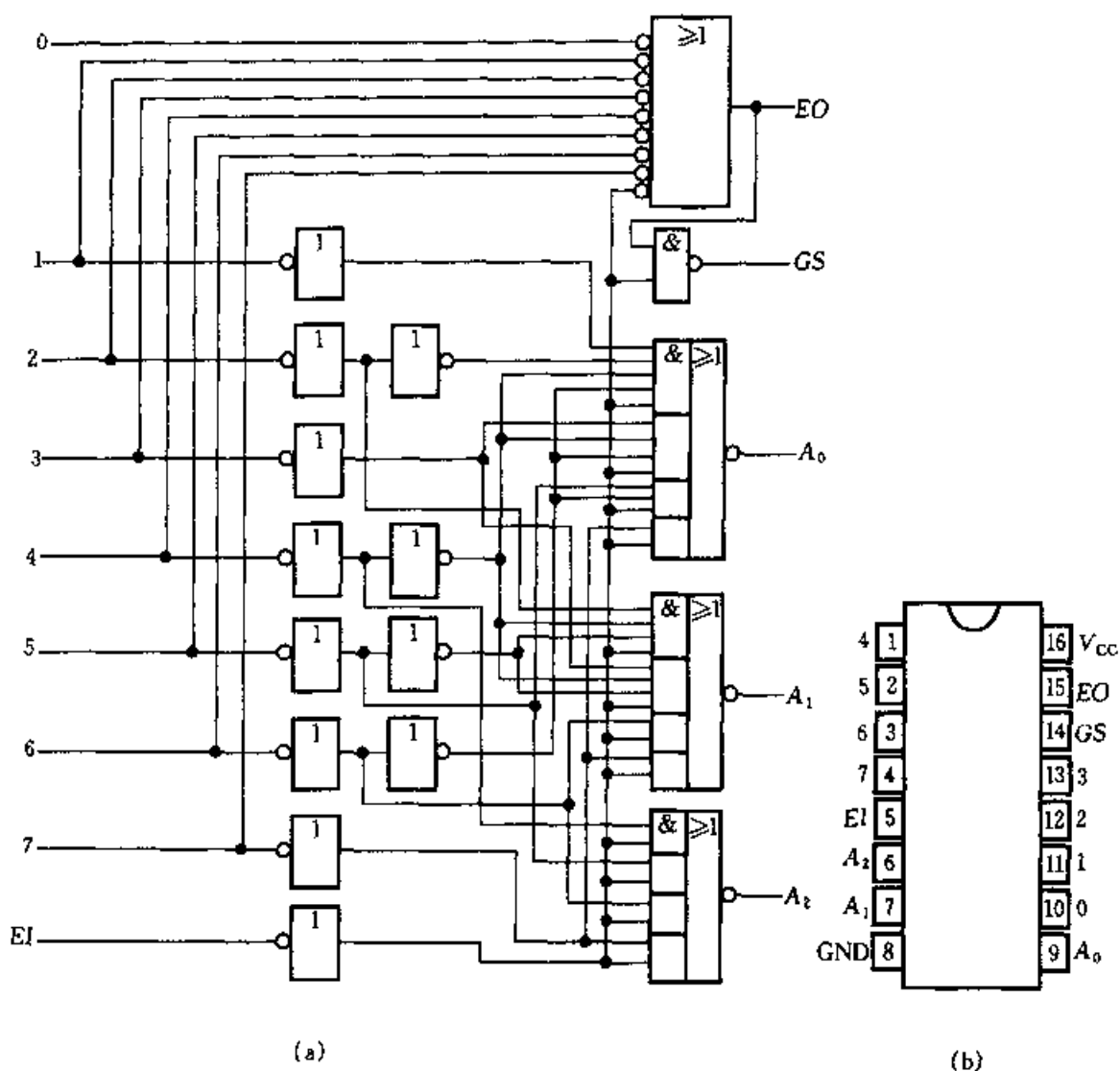


图 4.1.3 优先编码器 74148 的逻辑图和引脚图

(a) 逻辑图 (b) 引脚图

该编码器有 8 个信号输入端, 3 个二进制码输出端。此外, 电路还设置了输入使能端 EI , 输出使能端 EO 和优先编码工作状态标志 GS 。

当 $EI = 0$ ^① 时, 编码器工作; 而当 $EI = 1$ 时, 则不论 8 个输入端为何种状态, 3 个输出端均为高电平, 且优先标志端和输出使能端均为高电平, 编码器处于非工作状态。这种情况被称为输入低电平有效, 输出也有低电平有效的情况。当 EI 为 0, 且至少有一个输入端有编码请求信号(逻辑 0) 时, 优先编码工作状态标志 GS 为 0, 表明编码器处于工作状态, 否则为 1。由功能表可知, 在 8 个输入端均无低电平输入信号和只有输入 0 端(优先级别最低位) 有低电平输入时, $A_2 A_1 A_0$ 均为 111, 出现了输入条件不同而输出代码相同的情况, 这可由 GS 的

① 功能表中用高、低电平表示, 叙述时用逻辑状态表示, 采用正逻辑体制。

状态加以区别,当 $GS = 1$ 时,表示 8 个输入端均无低电平输入,此时 $A_2 A_1 A_0 = 111$ 为非编码输出; $GS = 0$ 时, $A_2 A_1 A_0 = 111$ 表示响应输入 0 端为低电平时的输出代码(编码输出)。 EO 只有在 EI 为 0,且所有输入端都为 1 时,输出为 0,它可与另一片同样器件的 EI 连接,以便组成更多输入端的优先编码器。

表 4.1.4 优先编码器 74148 功能表^{*}

EI ^①	输 入								输 出				
	0	1	2	3	4	5	6	7	A_2	A_1	A_0	GS	EO
H	×	×	×	×	×	×	×	×	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	×	×	×	×	×	×	×	L	L	L	L	L	H
L	×	×	×	×	×	×	L	H	L	L	H	L	H
L	×	×	×	×	×	L	H	H	L	H	L	L	H
L	×	×	×	×	L	H	H	H	L	H	H	L	H
L	×	×	L	H	H	H	H	H	H	L	L	L	H
L	×	L	H	H	H	H	H	H	H	L	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

* H——高电平 L——低电平 ×——任意电平

从功能表不难看出,输入优先级别的次序依次为 7,6,⋯,0。输入有效信号为低电平,当某一输入端有低电平输入,且比它优先级别高的输入端无低电平输入时,输出端才输出相对应的输入端的代码。例如,输入 5 为 0,且优先级别比它高的输入 6 和输入 7 均为 1 时,输出代码为 010,这就是优先编码器的工作原理。

为了能够列出逻辑表达式,我们设输入 0,1,⋯,7 对应的变量为 I_0, I_1, \dots, I_7 ,根据功能表,写出各输出端的逻辑表达式。

$$\overline{EO} = \overline{EI} I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7$$

$$EO = \overline{\overline{EI} I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7} \quad (4.1.1)$$

$$GS = EI + \overline{EI} I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7 \quad (4.1.2)$$

$$= EI + \overline{EO}$$

$$= \overline{\overline{EI} \cdot EO}$$

以 A_2 为例推导输出代码的逻辑表达式。

① 在相关教材或文献中,通常采用在逻辑变量上加非号和不加非号的方式分别表示低电平有效和高电平有效。当这类加非号的逻辑变量出现在逻辑表达式中时,由于它并不是逻辑非,因此就会引起与同一表达式中的其他非号的严重混淆,实际上导致无法对这种加非号的逻辑变量列写逻辑表达式,就难以完全做到功能表—逻辑表达式—逻辑图一致。例如,若此处用 \overline{EI} ,则后面的式(4.1.1)就无法得到。因此,本书不采用对变量符号加非号的方式表示低电平有效,但仍采用符号图框外加小圆圈的方式表示低电平有效。但有例外,如同一引脚线有两种功能操作,如 R/\overline{W} , R 表示读出操作,而 \overline{W} 代表写入操作,此时符号框外加小圆圈。

$$A_2 = EI + \overline{EI}(I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7 + \overline{I_0} I_1 I_2 I_3 I_4 I_5 I_6 I_7 + \overline{I_1} I_2 I_3 I_4 I_5 I_6 I_7 + \overline{I_2} I_3 I_4 I_5 I_6 I_7 + \overline{I_3} I_4 I_5 I_6 I_7)$$

利用 $A + \overline{AB} = A + B$ 和 $A + \overline{A} = 1$ 的关系,化简得

$$A_2 = EI + I_4 I_5 I_6 I_7$$

经过变换得

$$A_2 = \overline{EI\overline{I_4}} + \overline{EI\overline{I_5}} + \overline{EI\overline{I_6}} + \overline{EI\overline{I_7}} \quad (4.1.3)$$

同理可得输出 A_1 和 A_0 的逻辑表达式如下:

$$A_1 = \overline{EI\overline{I_2} I_4 I_5} + \overline{EI\overline{I_3} I_4 I_5} + \overline{EI\overline{I_6}} + \overline{EI\overline{I_7}}$$

$$A_0 = \overline{EI\overline{I_1} I_2 I_4 I_6} + \overline{EI\overline{I_3} I_4 I_6} + \overline{EI\overline{I_5} I_6} + \overline{EI\overline{I_7}}$$

由上式分析可知,式(4.1.1~4.1.3)与图 4.1.3 所示逻辑电路一致。

下面通过举例来了解 74148 的应用。

例 4.1.1 用两片 74148 组成 16 位输入、4 位二进制码输出的优先编码器,逻辑图如图 4.1.4 所示,试分析其工作原理。

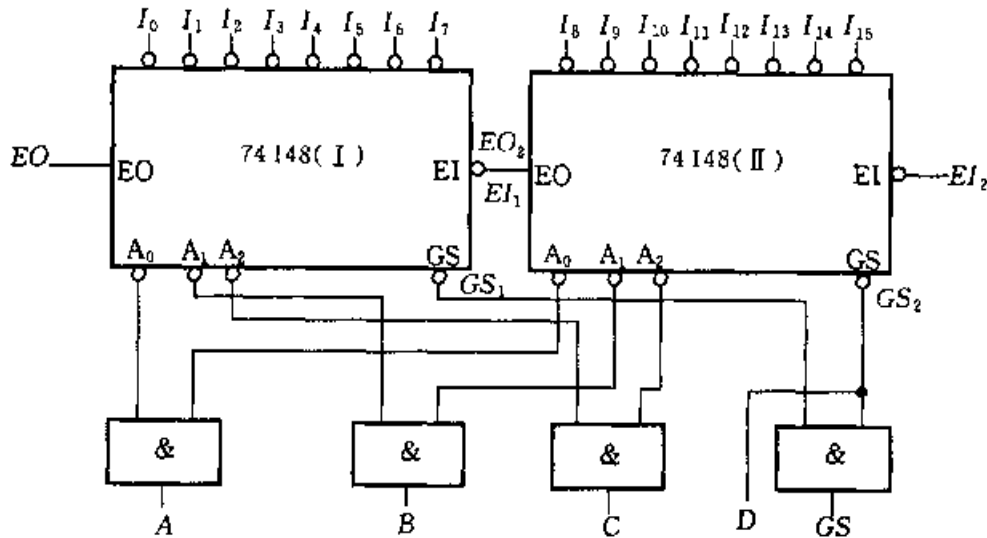


图 4.1.4 例 4.1.1 的逻辑图

解: 根据表 4.1.4 对逻辑图进行分析,可以得出:

(1) 当 $EI_2 = 1$ 时, $EO_2 = 1$, 从而使 $EI_1 = 1$, 这时 74148(I)(II) 均禁止编码, 它们的输出端 $A_2 A_1 A_0$ 都是 111。由电路图可知, $GS = GS_1 \cdot GS_2 = 1$, 表示此时整个电路的代码输出端 $DCBA = 1111$ 是非编码输出。

(2) 当 $EI_2 = 0$ 时, 高位片(II)允许编码, 但若 $I_{15} \sim I_8$ 都是高电平, 即均无编码请求, 则 $EO_2 = 0$, 从而 $EI_1 = 0$, 允许低位片(I)编码。这时高位片的 $A_2 A_1 A_0 = 111$, 使门 C、B、A 都打开, C、B、A 取决于低位片的 $A_2 A_1 A_0$, 而 $D = GS_2$, 总是等于 1, 所以输出代码在 1111~1000 之间变化。如果 I_0 单独有

效,输出为 **1111**;如果 I_7 及任意其他输入同时有效,则输出为 **1000**,低位片以 I_7 的优先级别最高。

(3) 当 $EI_2 = 0$ 且 $I_{15} \sim I_8$ 中有编码请求(至少一个为低电平)时, $EO_2 = 1$, 从而 $EI_1 = 1$,高位片编码,低位片禁止编码。显然,高位片的编码级别优先于低位片。此时 $D = GS_2 = 0$, C 、 B 、 A 取决于高位片的 $A_2A_1A_0$, 输出代码在 **0111** ~ **0000** 之间变化,同理可知,高位片中 I_{15} 的优先级别最高。

整个电路实现了 16 位输入的优先编码,其中 I_{15} 具有最高的优先级别,优先级别从 I_{15} 至 I_0 依次递减。

例 4.1.2 优先编码器 74147 的功能如表 4.1.5(实线部分)所示,试用 74147 和适当的门构成输出为 8421BCD 码并具有编码输出标志的编码器。

表 4.1.5 例 4.1.2 表

输 入									输 出					
1	2	3	4	5	6	7	8	9	D	C	B	A	GS	
H	H	H	H	H	H	H	H	H	H	H	H	H	H	0
×	×	×	×	×	×	×	×	L	L	H	H	L	L	1
×	×	×	×	×	×	×	L	H	L	H	H	H	L	1
×	×	×	×	×	×	L	H	H	H	L	L	L	L	1
×	×	×	×	×	L	H	H	H	H	L	L	H	L	1
×	×	×	L	H	H	H	H	H	H	L	H	L	L	1
×	×	L	H	H	H	H	H	H	H	H	L	L	L	1
×	L	H	H	H	H	H	H	H	H	H	L	H	L	1
L	H	H	H	H	H	H	H	H	H	H	H	L	L	1

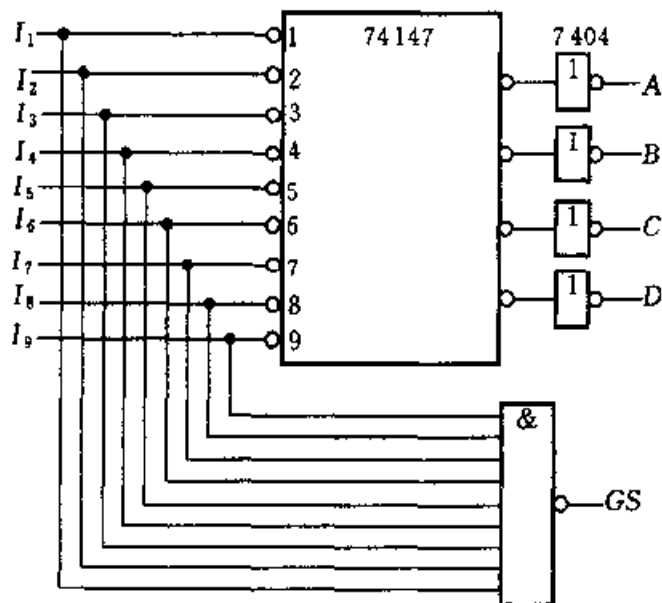


图 4.1.5 例 4.1.2 的逻辑图

解: 由表 4.1.5 可知,输出 $DCBA$ 是 8421BCD 码的反码,因此只要在

74147 的输出端增加反相器就可以获得题中所要求的输出码。在输入端均为高电平时 GS 为 0, 而有低电平信号输入时 GS 为 1, 可由与非门实现此功能, 题中所要求的编码器的逻辑电路如图 4.1.5 所示。

复习思考题

- 4.1.1 什么是编码? 什么是优先编码?
- 4.1.2 说明图 4.1.2 中输出信号 GS 的作用。
- 4.1.3 说明 74148 的输入信号 EI 和输出信号 GS 、 EO 的作用。

4.2 译码器/数据分配器

4.2.1 译码器的定义及功能

译码是编码的逆过程, 它的功能是将具有特定含义的二进制码进行辨别, 并转换成控制信号, 具有译码功能的逻辑电路称为译码器。

译码器可分为两种类型, 一种是将一系列代码转换成与之一一对应的有效信号。这种译码器可称为唯一地址译码器, 它常用于计算机中对存储器单元地址的译码, 即将每一个地址代码转换成一个有效信号, 从而选中对应的单元。另一种是将一种代码转换成另一种代码, 所以也称为代码变换器, 以下先介绍二进制唯一地址译码器。

图 4.2.1 表示二进制译码器的一般原理图, 它具有 n 个输入端, 2^n 个输出端和一个使能输入端。在使能输入端为有效电平时, 对应每一组输入代码, 只有其中一个输出端为有效电平, 其余输出端则为非有效电平。

下面首先分析由门电路组成的译码电路, 以便熟悉译码器的工作原理和电路结构。

2 输入变量的二进制译码器逻辑图如图 4.2.2 所示。由于 2 输入变量 A 、 B 共有 4 种不同状态组合, 因而可译出 4 个输出信号 $Y_0 \sim Y_3$, 故图 4.2.2 为 2 线输入, 4 线输出译码器, 简称 2 线 - 4 线译码器。

由图可写出各输出端的逻辑表达式

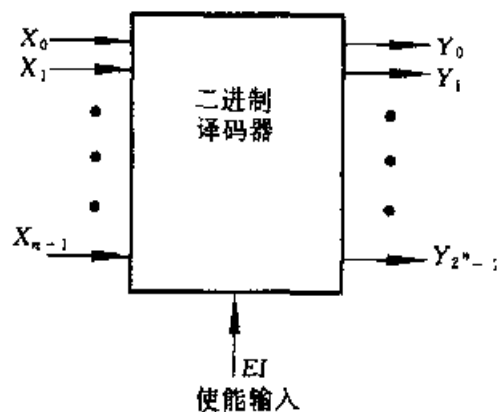


图 4.2.1 二进制译码器一般原理图

$$\begin{aligned}
 Y_0 &= \overline{\overline{EI}AB} \\
 Y_1 &= \overline{E\overline{I}AB} \\
 Y_2 &= \overline{EI\overline{A}B} \\
 Y_3 &= \overline{EIA\overline{B}}
 \end{aligned}
 \tag{4.2.1}$$

根据式(4.2.1)可列出功能表,如表 4.2.1 所示。由表可知,对于正逻辑,当 EI 为 1 时,无论 A 、 B 为何种状态,输出全为 1,译码器处于非工作状态。而当 EI 为 0 时,对应于 A 、 B 的某种状态组合,其中只有一个输出量为 0,其余各输出量均为 1。比如, $AB = 00$ 时,输出 Y_0 为 0, $Y_1 \sim Y_3$ 均为 1。由此可见,译码器是通过输出端的逻辑电平以识别不同的代码。

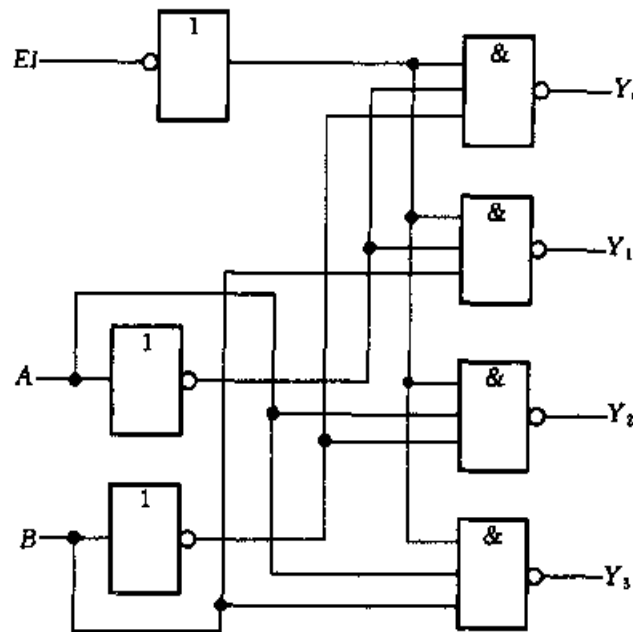


图 4.2.2 2线-4线译码器逻辑图

表 4.2.1 2线-4线译码器功能表

输入			输出			
EI	A	B	Y_0	Y_1	Y_2	Y_3
H	×	×	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

4.2.2 集成电路译码器

1. 74138 集成译码器

图 4.2.3a 为常用的集成译码器 74138 的逻辑图,其引脚如图 4.2.3b 所示,它的功能表如表 4.2.2 所示。由图可知,该译码器有 3 个输入 A 、 B 、 C ,它们共有 8 种状态的组合,即可译出 8 个输出信号 $Y_0 \sim Y_7$,故该译码器称为 3 线-8 线译码器。与图 4.2.2 比较,该译码器的主要特点是,设置了 G_1 、 G_{2A} 和 G_{2B} 3 个使能输入端。由功能表可知,对于正逻辑,当 G_1 为 1,且 G_{2A} 和 G_{2B} 均为 0 时,译码器处于工作状态。

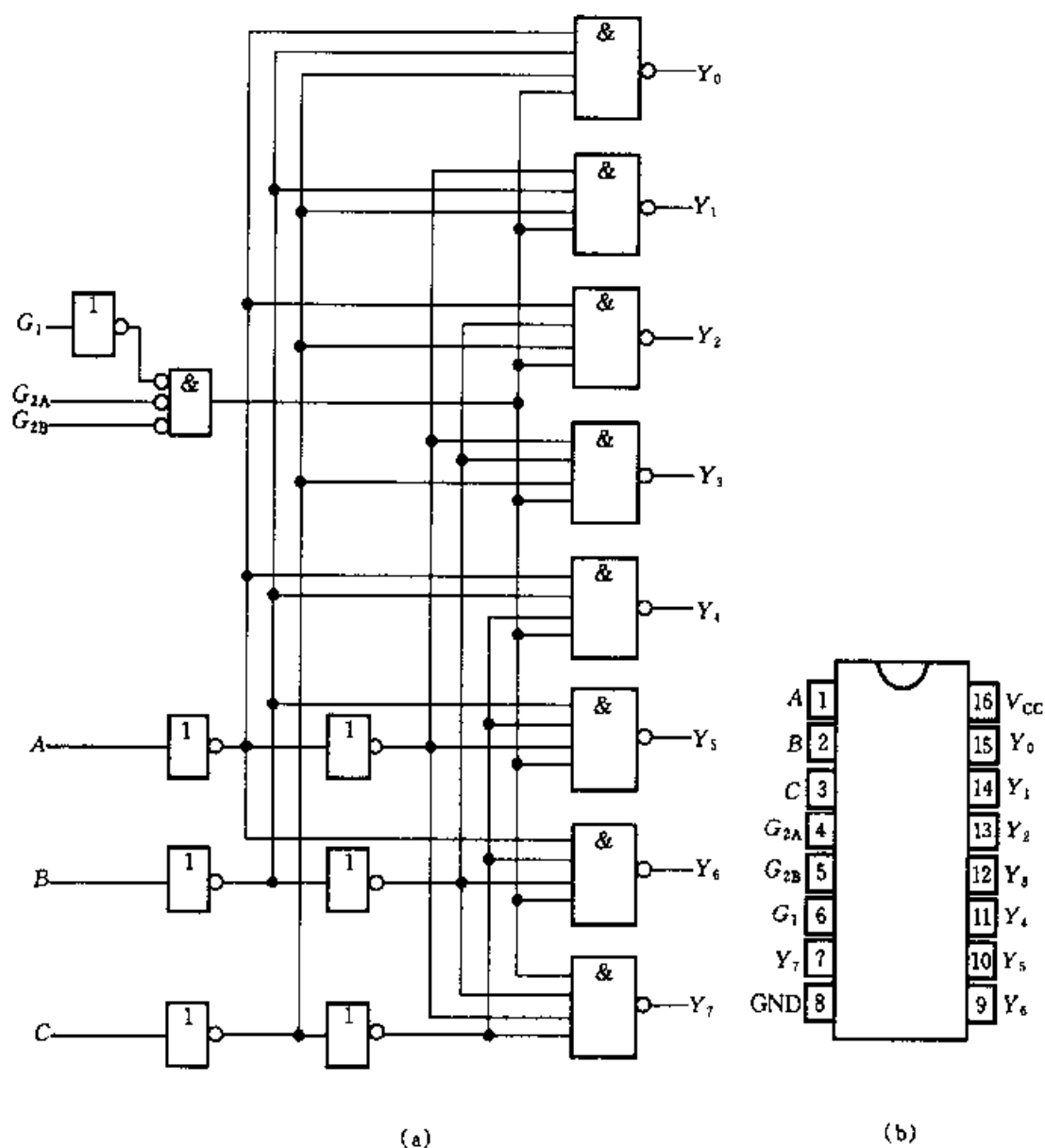


图 4.2.3 74138 集成译码器逻辑图和引脚图

(a) 逻辑图 (b) 引脚图

表 4.2.2 74138 集成译码器功能表

输 入						输 出							
G_1	G_{2A}	G_{2B}	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×	H	×	×	×	×	H	H	H	H	H	H	H	H
×	×	H	×	×	×	H	H	H	H	H	H	H	H
L	×	×	×	×	×	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

由功能表可得

$$\overline{Y_0} = G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}} \cdot \overline{C} \cdot \overline{B} \cdot \overline{A}$$

$$Y_0 = \overline{G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}} \cdot \overline{C} \cdot \overline{B} \cdot \overline{A}}$$

其他各输出端的逻辑表达式请读者自行推导。不难证明,由功能表推导出的各输出端的逻辑表达式与逻辑图是一致的。

显然,一个3线-8线译码器能产生3变量函数的全部最小项,利用这一点能够方便地实现3变量逻辑函数。

例 4.2.1 用一个3线-8线译码器实现函数 $F = \overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ$ 。

解: 第一步,将3个使能端按允许译码的条件进行处理,即 G_1 接 +5V, G_{2A} 和 G_{2B} 接地,于是得到各输出端的逻辑表达式为 $Y_0 = \overline{C}\overline{B}\overline{A}$, $Y_1 = \overline{C}\overline{B}A$, $Y_2 = \overline{C}B\overline{A}$, $Y_3 = \overline{C}BA$, $Y_4 = C\overline{B}\overline{A}$, $Y_5 = C\overline{B}A$, $Y_6 = CB\overline{A}$, $Y_7 = CBA$ 。

第二步,将输入变量 X 、 Y 、 Z 分别接到 C 、 B 、 A 端,并利用摩根定律进行变换,可得到

$$\begin{aligned} F &= \overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ \\ &= \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{X}\overline{Y}\overline{Z}}}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{X}Y\overline{Z}}}}}}}}}} \\ &= \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{C}\overline{B}\overline{A}}}}}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{C}\overline{B}A}}}}}}}}}} \\ &= \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{Y_0}}}}}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{Y_2}}}}}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{Y_4}}}}}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{Y_7}}}}}}}}}} \\ &= \overline{Y_0 \cdot Y_2 \cdot Y_4 \cdot Y_7} \end{aligned}$$

可见,3线-8线译码器再加一个与非门,即可实现题目所指定的组合逻辑

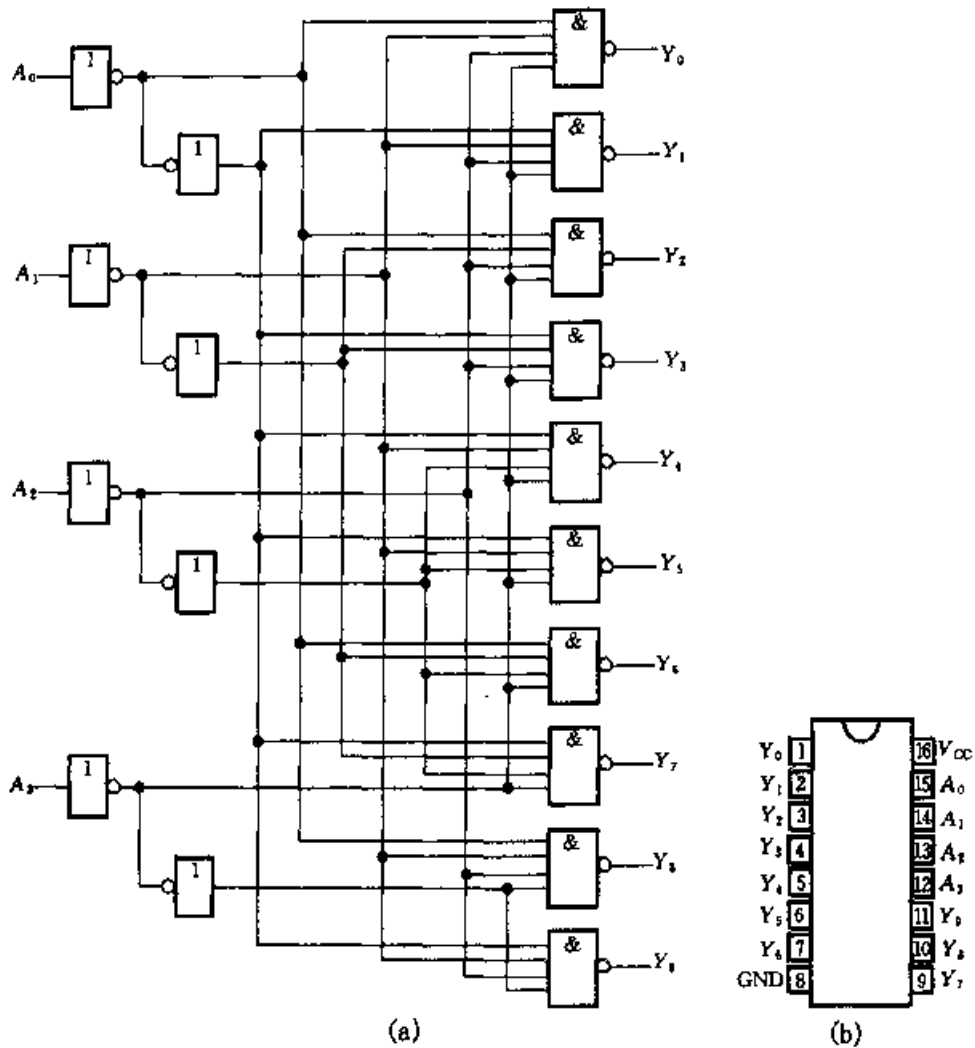


图 4.2.5 7442 二-十进制译码器的逻辑图

(a) 逻辑图 (b) 引脚图

对于 Y_0 输出从逻辑图和功能表都可以得出 $Y_0 = \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0}$, 当 $A_3 A_2 A_1 A_0 = 0000$ 时, 输出 $Y_0 = 0$, 它对应于十进制数 0, 其余输出依此类推。

3. 七段显示译码器

在数字测量仪表和各种数字系统中, 都需要将数字量直观地显示出来, 一方面供人们直接读取测量和运算的结果; 另一方面用于监视数字系统的工作情况。因此, 数字显示电路是许多数字设备不可缺少的部分。数字显示电路通常由译码器、驱动器和显示器等部分组成, 如图 4.2.6 所示。下面对显示器和译码驱

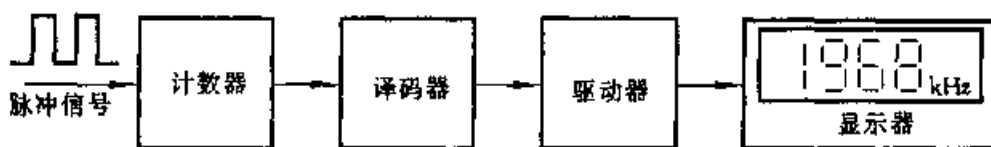


图 4.2.6 数字显示电路组成方块图

动器分别进行介绍。

数码显示器是用来显示数字、文字或符号的器件,现在已有多种不同类型的产品,广泛应用于各种数字设备中,目前数码显示器件正朝着小型、低功耗、平面化方向发展。

数码的显示方式一般有三种:第一种是字形重叠式,它是将不同字符的电极重叠起来,要显示某字符,只须使相应的电极发亮即可,如辉光放电管、边光显示管等。第二种是分段式,数码是由分布在同一平面上若干段发光的笔划组成,如荧光数码管等。第三种是点阵式,它由一些按一定规律排列的可发光的点阵所组成,利用光点的不同组合便可显示不同的数码,如场致发光记分牌。

数字显示方式目前以分段式应用最普遍,图 4.2.7 表示七段式数字显示器利用不同发光段方式组合,显示 0~15 等阿拉伯数字。在实际应用中,10~15 并不采用,而是用 2 位数字显示器进行显示。

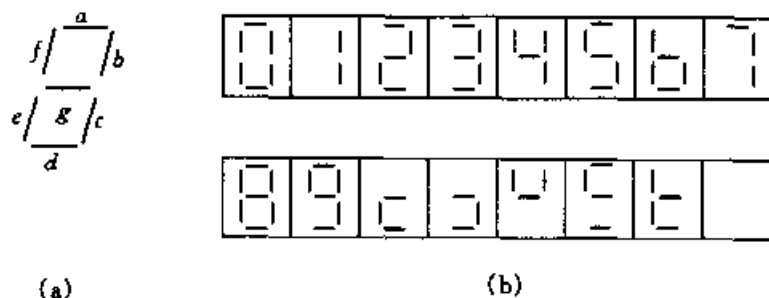


图 4.2.7 七段数字显示器发光段组合图

(a) 分段布置图 (b) 段组合图

按发光物质不同,数码显示器可分为下列几类:(1) 半导体显示器,亦称发光二极管显示器;(2) 荧光数字显示器,如荧光数码管、场致发光数字板等;(3) 液体数字显示器,如液晶显示器、电泳显示器等;(4) 气体放电显示器,如辉光数码管、等离子体显示板等。

如前所述,分段式数码管是利用不同发光段组合的方式显示不同数码的。因此,为了使数码管能将数码所代表的数显示出来,必须将数码经译码器译出,然后经驱动器点亮对应的段。例如,对于 8421 码的 0011 状态,对应的十进制数为 3,则译码驱动器应使 a 、 b 、 c 、 d 、 g 各段点亮。即对应于某一组数码,译码器应有确定的几个输出端有信号输出,这是分段式数码管电路的主要特点。下面介绍常用的 7448 七段显示译码器。

7448 七段显示译码器输出高电平有效,用以驱动共阴极显示器。该集成显示译码器设有多个辅助控制端,以增强器件的功能。7448 的功能表如表 4.2.4 所示,它有 3 个辅助控制端 LT 、 RBI 、 BI/RBO ,现分别简要说明如下:

① 灭灯输入 BI/RBO

BI/RBO 是特殊控制端,有时作为输入,有时作为输出。当 BI/RBO 作输入使用且 $BI=0$ 时,无论其他输入端是什么电平,所有各段输出 $a\sim g$ 均为 0,所以字形熄灭。

② 试灯输入 LT

当 $LT=0$ 时, BI/RBO 是输出端,且 $RBO=1$,此时无论其他输入端是什么状态,所有各段输出 $a\sim g$ 均为 1,显示字形 8 。该输入端常用于检查 7448 本身及显示器的好坏。

表 4.2.4 7448 功能表

十进制 或功能	输入						BI/RBO	输出							字 形
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	H	H	H	H	H	H	L	0
1	H	×	L	L	L	H	H	L	H	H	L	L	L	L	1
2	H	×	L	L	H	L	H	H	H	L	H	H	L	H	2
3	H	×	L	L	H	H	H	H	H	H	H	L	L	H	3
4	H	×	L	H	L	L	H	L	H	H	L	L	H	H	4
5	H	×	L	H	L	H	H	H	L	H	H	L	H	H	5
6	H	×	L	H	H	L	H	L	L	H	H	H	H	H	6
7	H	×	L	H	H	H	H	H	H	H	L	L	L	L	7
8	H	×	H	L	L	L	H	H	H	H	H	H	H	H	8
9	H	×	H	L	L	H	H	H	H	H	H	L	H	H	9
10	H	×	H	L	H	L	H	L	L	L	H	H	L	H	a
11	H	×	H	L	H	H	H	L	L	H	H	L	L	H	b
12	H	×	H	H	L	L	H	L	H	L	L	L	H	H	c
13	H	×	H	H	L	H	H	H	L	L	H	L	H	H	d
14	H	×	H	H	H	L	H	L	L	L	H	H	H	H	e
15	H	×	H	H	H	H	H	L	L	L	L	L	L	L	f
消隐	×	×	×	×	×	×	L	L	L	L	L	L	L	L	
脉冲消隐	H	L	L	L	L	L	L	L	L	L	L	L	L	L	
灯测试	L	×	×	×	×	×	H	H	H	H	H	H	H	H	8

③ 动态灭零输入 RBI

当 $LT=1$, $RBI=0$ 且输入代码 $DCBA=0000$ 时,各段输出 $a\sim g$ 均为低电平,与 BCD 码相应的字形 0 熄灭,故称“灭零”。利用 $LT=1$ 与 $RBI=0$ 可以实现某一位 0 的“消隐”。此时 BI/RBO 是输出端,且 $RBO=0$ 。

④ 动态灭零输出 RBO

BI/RBO 作为输出使用时,受控于 LT 和 RBI 。当 $LT=1$ 且 $RBI=0$,输入代码 $DCBA=0000$ 时, $RBO=0$; 若 $LT=0$ 或者 $LT=1$ 且 $RBI=1$, 则 RBO

= 1。该端主要用于显示多位数字时,多个译码器之间的连接。

从功能表还可看出,对输入代码 0000,译码条件是:LT 和 RBI 同时等于 1,而对其他输入代码则仅要求 $LT = 1$,这时候,译码器各段 $a \sim g$ 输出的电平是由输入 BCD 码决定的,并且满足显示字形的要求。

下面举一个利用 7448 实现多位数字译码显示的例子,通过它了解各控制端的使用,特别是如何动态灭零,实现无意义位的“消隐”。

该例如图 4.2.8 所示,图中 7 位显示器由 7 个译码器 7448 驱动。各片 7448 的 LT 均接高电平,由于第一片的 $RBI = 0$ 且 $DCBA = 0000$,所以第一片满足灭零条件,无字形显示,同时输出 $RBO = 0$;第一片的 RBO 与第二片的 RBI 相连,使第二片也满足灭零条件,无显示并输出 $RBO = 0$;同理,第三片的零也熄灭,无显示。由于第四、五、六、七片译码器的 $RBI = 1$,所以它们都正常译码,按输入 BCD 码去点亮各段电极。

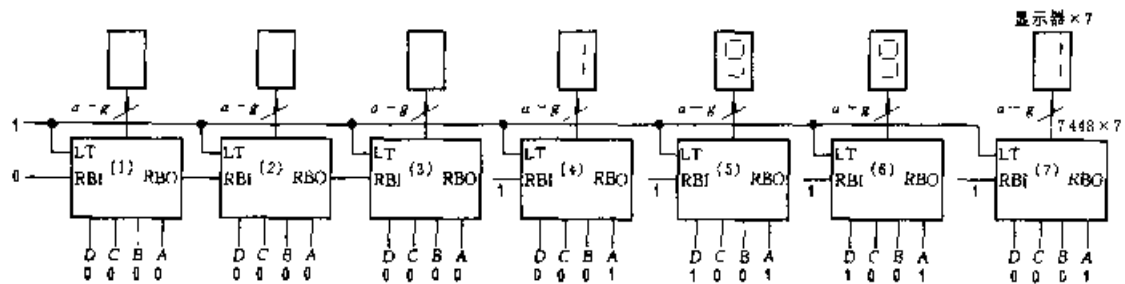


图 4.2.8 用 7448 实现多位数字译码显示

如果图 4.2.8 接法不变,但第一片 7448 的输入代码不是 0000 而是任何其他 BCD 码,则该片将正常译码并驱动显示,同时使 $RBO = 1$ 。这样,第二片、第三片就丧失了灭零条件,所以电路对最高位非零的数字仍然正常显示。

4.2.3 数据分配器

数据分配是将一个数据源来的数据根据需要送到多个不同的通道上去,实现数据分配功能的逻辑电路称为数据分配器。它的作用相当于多个输出的单刀多掷开关,其示意图如图 4.2.9 所示。

数据分配器可以用唯一地址译码器实现。如用 3 线-8 线译码器可以把一个数据信号分配到 8 个不同的通道上去。用 74138 作为数据分配器的逻辑原理图如图 4.2.10 所示。将 G_{2B} 接低电平, G_1 作为使能端, C 、 B 和 A 作为选择通道地址输入, G_{2A} 作为数据输入。例如,当 $G_1 = 1$,

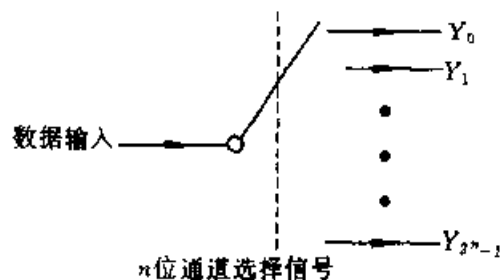


图 4.2.9 数据分配器示意图

$CBA = 010$ 时,由功能表(表 4.2.2)可得

$$Y_2 = (G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}}) \cdot \overline{C} \cdot B \cdot \overline{A}$$

$$= G_{2A}$$

而其余输出端均为高电平。因此,当地址 $CBA = 010$ 时,只有输出端 Y_2 得到与输入相同的数据波形。74138 译码器作为数据分配器的功能表如表 4.2.5 所示。

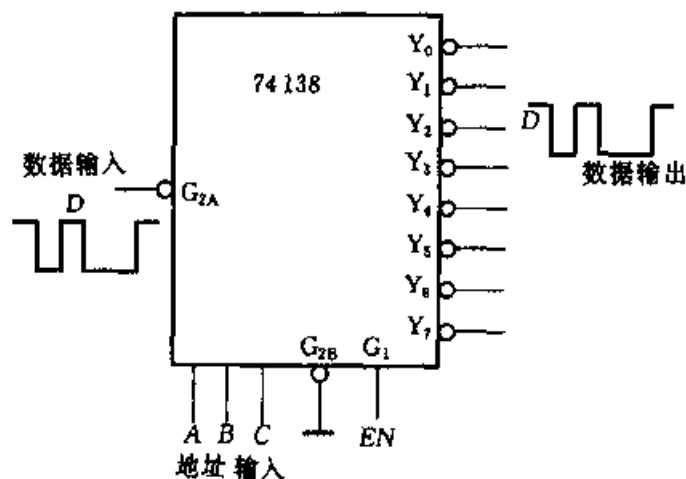


图 4.2.10 用 74138 作为数据分配器

表 4.2.5 74138 译码器作为数据分配器时的功能表

输 入						输 出							
G_1	G_{2B}	G_{2A}	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
L	L	×	×	×	×	H	H	H	H	H	H	H	H
H	L	D	L	L	L	D	H	H	H	H	H	H	H
H	L	D	L	L	H	H	D	H	H	H	H	H	H
H	L	D	L	H	L	H	H	D	H	H	H	H	H
H	L	D	L	H	H	H	H	H	D	H	H	H	H
H	L	D	H	L	L	H	H	H	H	D	H	H	H
H	L	D	H	L	H	H	H	H	H	H	D	H	H
H	L	D	H	H	L	H	H	H	H	H	H	D	H
H	L	D	H	H	H	H	H	H	H	H	H	H	D

数据分配器的用途比较多,比如用它将一台 PC 机与多台外部设备连接,将计算机的数据分送到外部设备中。它还可以与计数器结合组成脉冲分配器,用它与数据选择器连接组成分时数据传送系统。

复习思考题

4.2.1 什么是译码? 什么是唯一地址译码?

4.2.2 何种译码器可以作为数据分配器使用？为什么？

4.2.3 74138 译码器作为数据分配器使用时，对于 G_1 、 G_{2A} 、 G_{2B} 的设置办法，除了如图 4.2.10 外，你还有别的什么办法？

4.3 数据选择器

4.3.1 数据选择器的定义及功能

数据选择是指经过选择，把多个通道的数据传送到唯一的公共数据通道上去。实现数据选择功能的逻辑电路称为数据选择器。它的作用相当于多个输入的单刀多掷开关，其示意图如图 4.3.1 所示。

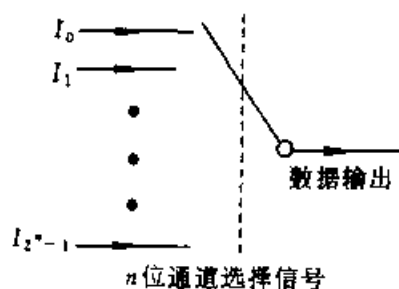


图 4.3.1 数据选择器示意图

下面以 4 选 1 数据选择器为例，说明工作原理及基本功能。其逻辑图如图 4.3.2 所示，功能表如表 4.3.1 所示。

为了对 4 个数据源进行选择，使用两位地址码 BA 产生 4 个地址信号，由 BA 等于 **00**、**01**、**10**、**11** 分别控制四个与门的开闭。显然，任何时候 BA 只有一种可能的取值，所以只有一个与门打开，使对应的那一路数据通过，送达 Y 端。输入使能端 G 是低电平有效，当 $G = 1$ 时，所有与门都被封锁，无论地址码是什么， Y 总是等于 **0**；当 $G = 0$ 时，封锁解除，由地址码决定哪一个与门打开。

同样原理，可以构成更多输入通道的数据选择器。被选数据源越多，所需地址码的位数也越多，若地址输入端为 n ，可选输入通道数为 2^n 。

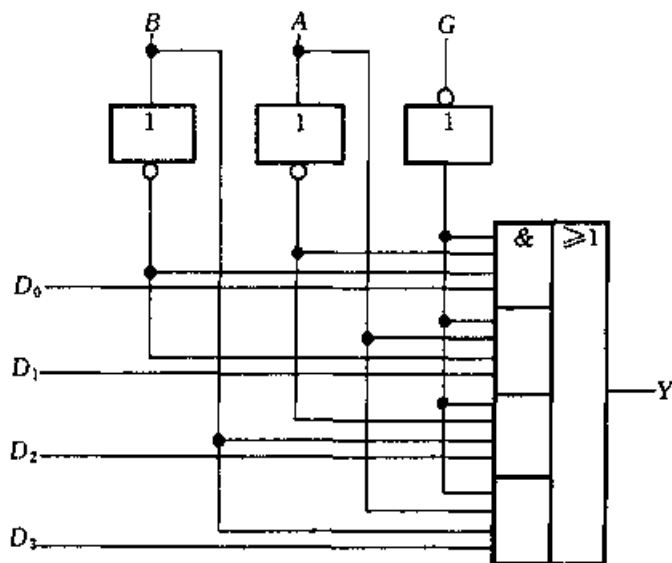


图 4.3.2 4 选 1 数据选择器逻辑图

表 4.3.1 4 选 1 数据选择器功能表

输 入			输 出
使能	地 址		
G	B	A	Y
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

4.3.2 集成电路数据选择器

1. 74LS151 集成电路数据选择器的功能

74LS151 是一种典型的集成电路数据选择器,它有 3 个地址输入端 CBA,可选择 $D_0 \sim D_7$ 8 个数据源,具有两个互补输出端,同相输出端 Y 和反相输出端 W。其逻辑图和引脚图分别如图 4.3.3a 和 4.3.3b 所示,功能表如表 4.3.2 所示。由图 4.3.3a 可知,该逻辑电路的基本结构为“与-或-非”形式。输入使能 G 为低电平有效。

表 4.3.2 74LS151 的功能表

输 入				输 出	
使 能	选 择			Y	W
G	C	B	A		
H	×	×	×	L	H
L	L	L	L	D_0	\bar{D}_0
L	L	L	H	D_1	\bar{D}_1
L	L	H	L	D_2	\bar{D}_2
L	L	H	H	D_3	\bar{D}_3
L	H	L	L	D_4	\bar{D}_4
L	H	L	H	D_5	\bar{D}_5
L	H	H	L	D_6	\bar{D}_6
L	H	H	H	D_7	\bar{D}_7

输出 Y 的表达式为
$$Y = \sum_{i=0}^7 m_i D_i$$

式中 m_i 为 CBA 的最小项。例如,当 $CBA = 010$ 时,根据最小项性质,只有 m_2

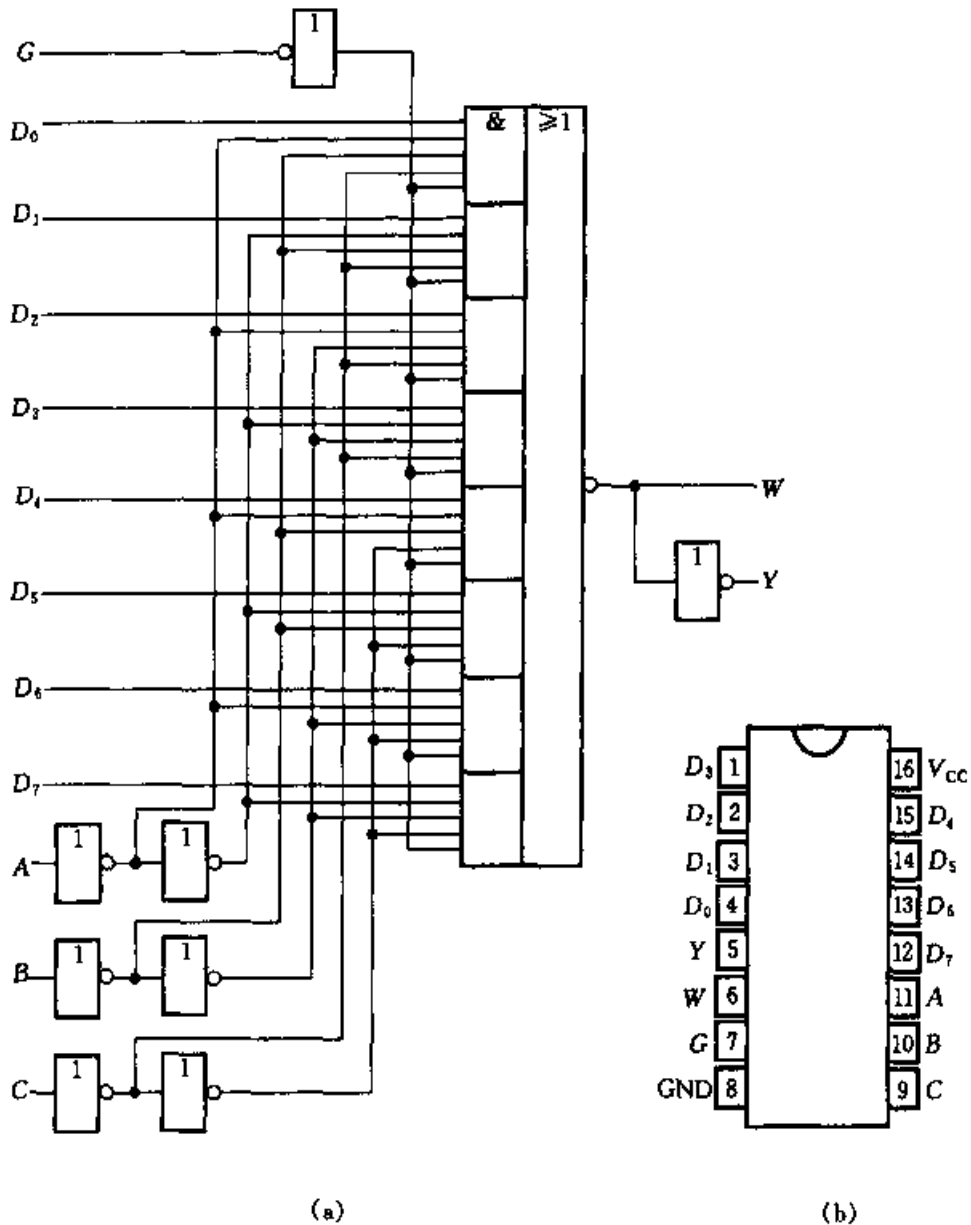


图 4.3.3 74LS151 的逻辑图和引脚分布图

(a) 逻辑图 (b) 引脚分布图

为 1, 其余各项为 0, 故得 $Y = D_2$, 即只有 D_2 传送到输出端。

上面所讨论的是 1 位数据选择器, 如需要选择多位数据时, 可由几个 1 位数据选择器并联组成, 即将它们的使能端连在一起, 相应的选择输入端连在一起。2 位 8 选 1 数据选择器的连接方法如图 4.3.4 所示。当需要进一步扩充位数时, 只需相应地增加器件的数目。

可以把数据选择器的使能端作为地址选择输入, 将两片 74LS151 连接成一个 16 选 1 的数据选择器, 其连接方式如图 4.3.5 所示。16 选 1 的数据选择器的地址选择输入有 4 位, 其最高位 D 与一个 8 选 1 数据选择器的使能端连接, 经过一反相器反相后与另一个数据选择器的使能端连接。低 3 位地址选择输入

端 CBA 由两片 74LS151 的地址选择输入端相对应连接而成。

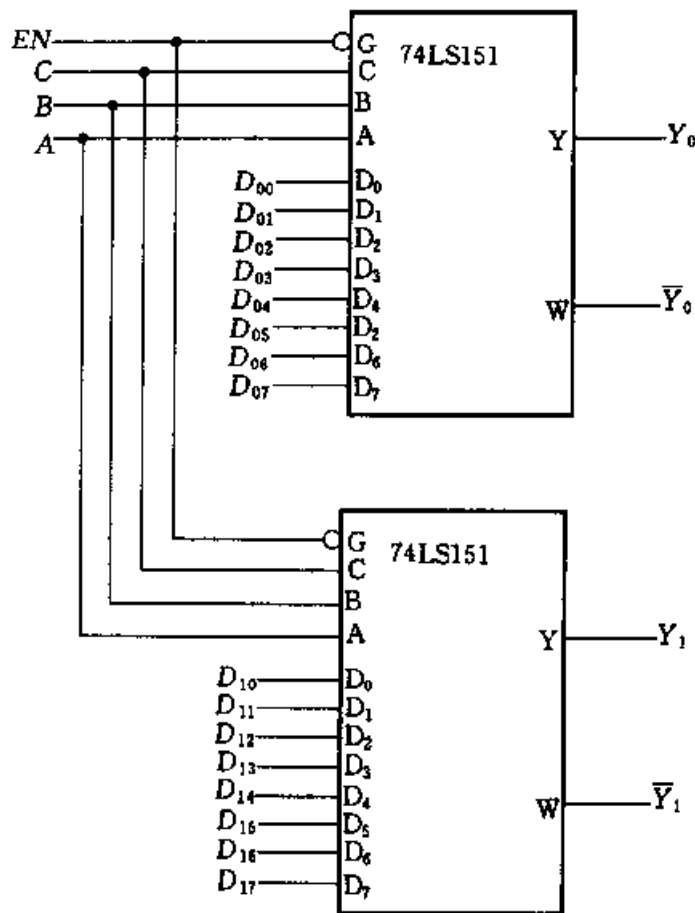


图 4.3.4 2 位八选一数据选择器的连接方法

2. 数据选择器的应用

数据选择器除实现有选择的传送数据外,还有其他用途,下面介绍几种典型应用。

(1) 逻辑函数产生器

从图 4.3.3a 可以看出,当使能端 $G=0$ 时, Y 是 C 、 B 、 A 和输入数据 $D_0 \sim D_7$ 的与或函数,它的表达式可以写成

$$Y = \sum_{i=0}^7 m_i D_i \quad (4.3.1)$$

式中 m_i 是 C 、 B 、 A 构成的最小项。显然,当 $D_i=1$ 时,其对应的最小项 m_i 在与或表达式中出现,当 $D_i=0$ 时,对应的最小项就不出现。利用这一点,不难实现组合逻辑函数。已知逻辑函数,利用数据选择器构成函数产生器的过程是,将函数变换成最小项表达式,根据最小项表达式确定各数据输入端的二元常量。将数据选择器的地址信号 C 、 B 、 A 作为函数的输入变量,数据输入 $D_0 \sim D_7$ 作为控制信号,控制各最小项在输出逻辑函数中是否出现,使能端 G 始终保

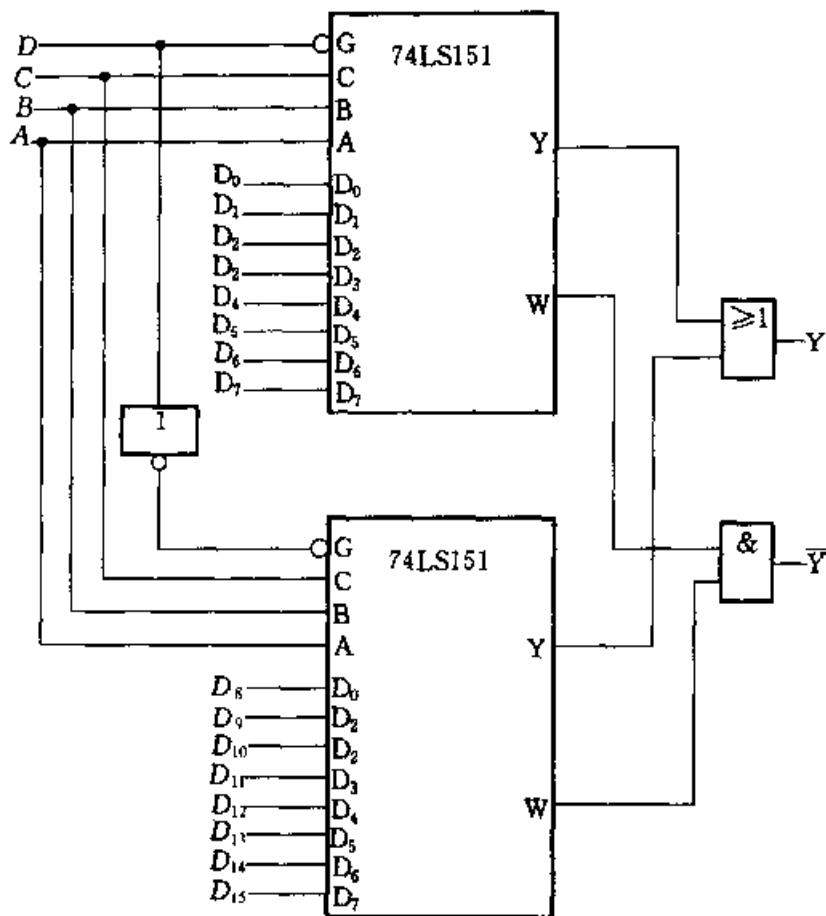


图 4.3.5 用两片 8 选 1 数据选择器连接成 16 选 1 数据选择器的逻辑图

持低电平,这样八选一数据选择器就成为一个 3 变量的函数产生器。

例 4.3.1 试用 8 选 1 数据选择器 74LS151 产生逻辑函数

$$L = \bar{X}YZ + X\bar{Y}Z + XY$$

解: 把式 $L = \bar{X}YZ + X\bar{Y}Z + XY$ 变换成最小项表达式

$$L = \bar{X}YZ + X\bar{Y}Z + XYZ + XY\bar{Z}$$

将上式写成如下形式:

$$L = m_3D_3 + m_5D_5 + m_6D_6 + m_7D_7$$

该式符合式(4.3.1)的标准形式,显然 D_3 、 D_5 、 D_6 、 D_7 都应该等于 1,而式中没有出现的最小项 m_0 、 m_1 、 m_2 、 m_3 的控制变量 D_0 、 D_1 、 D_2 、 D_4 都应该等于 0。由此可画出该逻辑函数产生器的逻辑图,如图 4.3.6 所示。

例 4.3.2 试用与上例相同的 8 选 1 数据选择器产生 $L = X \oplus Y \oplus Z$ 。

解: 根据表达式 $L = X \oplus Y \oplus Z$ 列出真值表,如表 4.3.3 所示。从表中可以看出,凡使 L 值为 1 的那些最小项,其控制变量应该等于 1,即 D_1 、 D_2 、 D_4 、 D_7 等于 1,其他控制变量均等于 0。由此可得逻辑函数产生器,如图 4.3.7 所示。

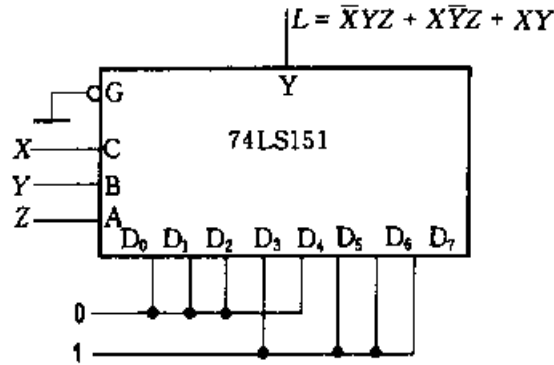


图 4.3.6 例 4.3.1 的图

表 4.3.3 例 4.3.2 的真值表

输入			输出	输入			输出
X	Y	Z	L	X	Y	Z	L
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	1	1	1

通过上面两例可以看出,与用各种逻辑门设计组合逻辑电路相比,使用数据选择器的好处是无需对函数化简。

(2) 实现并行数据到串行数据的转换

图 4.3.8 是实现并/串行转换的电路框图和输出信号时序图。电路由 8 选 1 数据选择器和 1 个 3 位二进制计数器构成。计数器的作用是累计时钟脉冲的个数,当时钟脉冲 CP 一个接一个送入时,计数器的输出端 $Q_2 Q_1 Q_0$ 从 $000 \rightarrow 001 \rightarrow 010 \rightarrow \dots \rightarrow 111$ 依次变化。由于 $Q_2 Q_1 Q_0$ 与选择器的地址输入端 C, B, A 相连,因此 C, B, A 就随时钟脉冲的逐个输入从 000 到 111 变化,选择器的输出 Y 随之接通 $D_0, D_1, D_2, \dots, D_7$ 。当选择器的数据输入端 $D_0 \sim D_7$ 与一个并行 8 位数 01001101 相连时,输出端得到的就是一串随时钟节拍变化的数据 $0-1-0-0-1-1-0-1$,这种数称为串行数据。为了简单起见,图 4.3.8b 时序图中的时钟脉冲 CP 画成了尖窄脉冲。

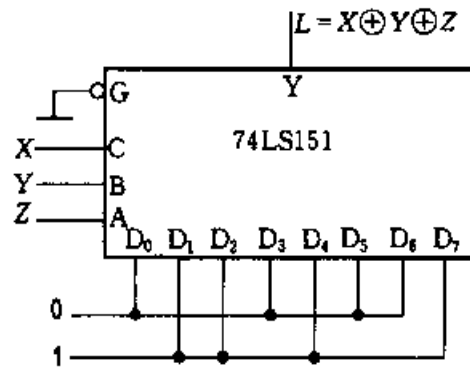


图 4.3.7 例 4.3.2 的逻辑函数产生器的逻辑图

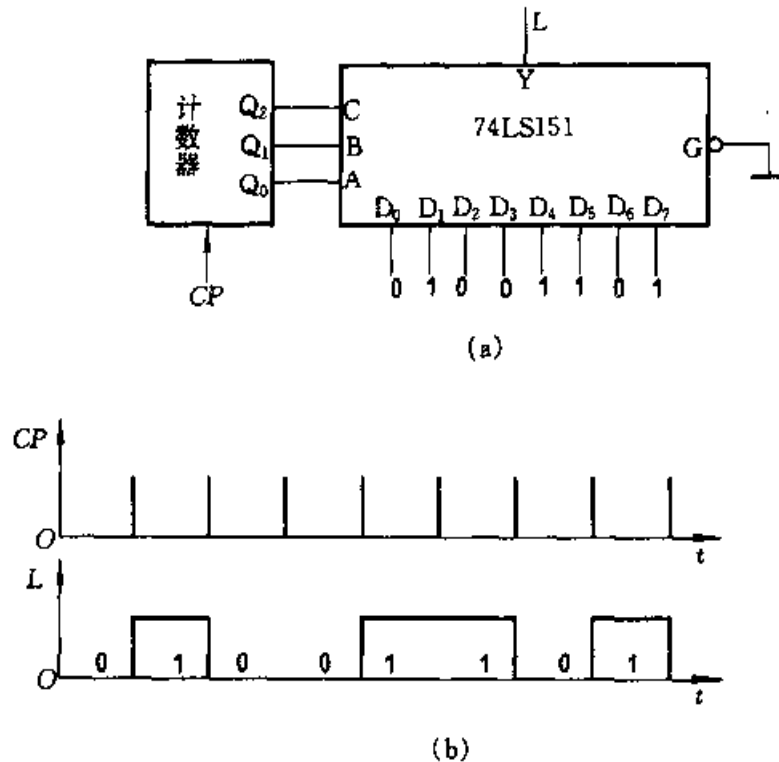


图 4.3.8 数据并行输入转换成串行输出

(a) 电路框图 (b) 时序图

复习思考题

4.3.1 试用十六进制数的方式写出 16 选 1 的数据选择器的各地址码。

4.3.2 用 32 选 1 数据选择器选择数据, 设选择的输入数据为 D_{20} 、 D_{17} 、 D_{18} 、 D_{27} 、 D_{31} , 试依次写出对应的地址码。

4.4 数值比较器

4.4.1 数值比较器的定义及功能

在数字系统中, 特别是在计算机中都具有运算功能, 一种简单的运算就是比较两个数 A 和 B 的大小。数值比较器就是对两数 A 、 B 进行比较, 以判断其大小的逻辑电路。比较结果有 $A > B$ 、 $A < B$ 以及 $A = B$ 三种情况。

1. 1 位数值比较器

1 位数值比较器是多位比较器的基础。当 A 和 B 都是 1 位数时, 它们只能取 0 或 1 两种值, 由此可写出 1 位数值比较器的真值表, 如表 4.4.1 所示。

表 4.4.1 1 位数值比较器真值表

输 入		输 出		
A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

由真值表得到如下逻辑表达式:

$$\left. \begin{aligned} F_{A>B} &= A\bar{B} \\ F_{A<B} &=\bar{A}B \\ F_{A=B} &=\bar{A}\bar{B} + AB \end{aligned} \right\} \quad (4.4.1)$$

由以上逻辑表达式可画出如图 4.4.1 所示的逻辑电路。实际应用中,可根据具体情况选用逻辑门。

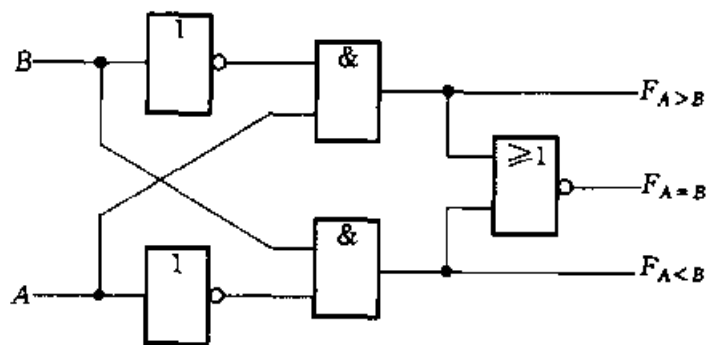


图 4.4.1 1 位数值比较器的逻辑图

2. 两位数值比较器

现在分析比较两位数字 A_1A_0 和 B_1B_0 的情况。利用 1 位比较器的结果,可以列出简化的真值表,如表 4.4.2 所示。为了减少符号的种类,不再使用字母 L ,而以 $(A_i > B_i)$ 、 $(A_i < B_i)$ 、 $(A_i = B_i)$ 直接表示逻辑函数。可以由真值表对两位比较器作如下简要概述。

表 4.4.2

输 入		输 出		
$A_1 B_1$	$A_0 B_0$	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1 > B_1$	×	1	0	0
$A_1 < B_1$	×	0	1	0
$A_1 = B_1$	$A_0 > B_0$	1	0	0
$A_1 = B_1$	$A_0 < B_0$	0	1	0
$A_1 = B_1$	$A_0 = B_0$	0	0	1

当高位(A_1 、 B_1)不相等时,无需比较低位(A_0 、 B_0),两个数的比较结果就是高位比较的结果。当高位相等时,两数的比较结果由低位比较的结果决定。由表 4.4.2 可以写出如下逻辑表达式:

$$F_{A>B} = (A_1 > B_1) + (A_1 = B_1)(A_0 > B_0)$$

$$F_{A<B} = (A_1 < B_1) + (A_1 = B_1)(A_0 < B_0)$$

$$F_{A=B} = (A_1 = B_1)(A_0 = B_0)$$

根据表达式画出逻辑图,如图 4.4.2 所示。电路利用了 1 位数值比较器的输出作为中间结果。它所依据的原理是,如果两位数 A_1A_0 和 B_1B_0 的高位不相等,则高位比较结果就是两数比较结果,与低位无关。这时,由于中间函数 $(A_1 = B_1) = 0$,使与门 G_1 、 G_2 、 G_3 均封锁,而或门都打开,低位比较结果不能影响或门,高位比较结果则从或门直接输出。如果高位相等,即 $(A_1 = B_1) = 1$,使与门 G_1 、 G_2 、 G_3 均打开,同时由 $(A_1 > B_1) = 0$ 和 $(A_1 < B_1) = 0$ 作用,或门也打开,低位的比较结果直接送达输出端,即低位的比较结果决定两数谁大、谁小或者相等。

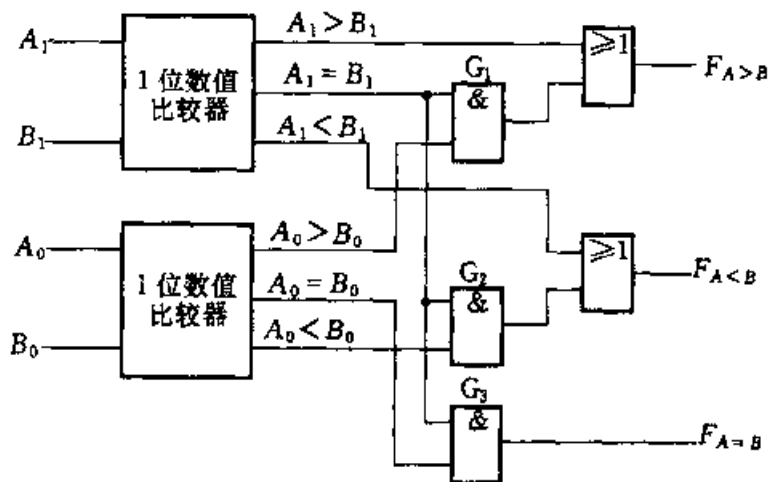


图 4.4.2 两位数值比较逻辑图

4.4.2 集成数值比较器

1. 集成数值比较器 74LS85 的功能

集成数值比较器 74LS85 是 4 位数值比较器,其功能如表 4.4.3 所示。从表 4.4.3 可以看出,该比较器的比较原理和两位比较器的比较原理相同。两个 4 位数的比较是从 A 的最高位 A_3 和 B 的最高位 B_3 进行比较,如果它们不相等,则该位的比较结果可以作为两数的比较结果。若最高位 $A_3 = B_3$,则再比较次高位 A_2 和 B_2 ,余类推。显然,如果两数相等,那么,比较步骤必须进行到最低

位才能得到结果。74LS85 的逻辑图和引脚图分别如图 4.4.3a 和 4.4.3b 所示。

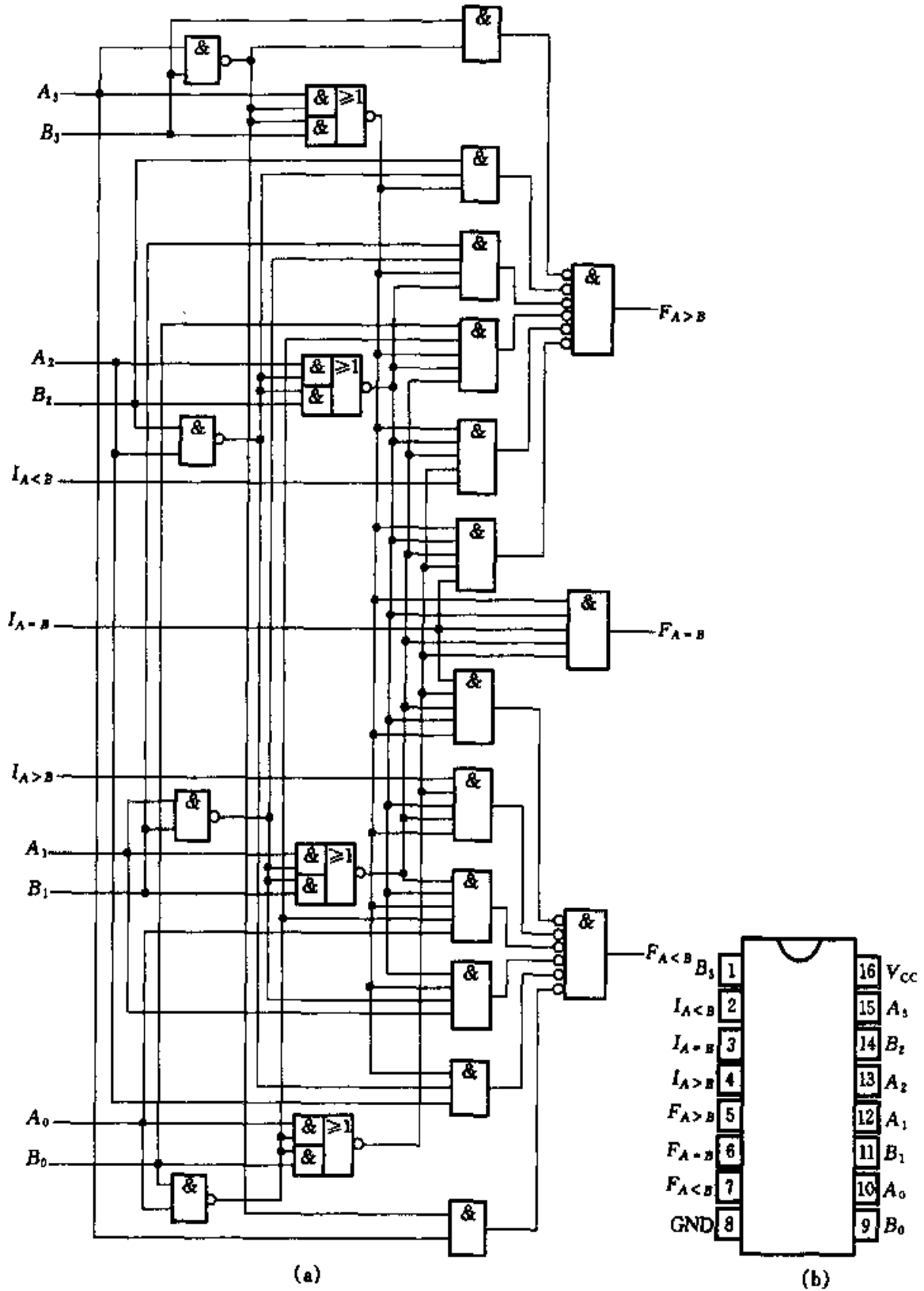


图 4.4.3 74LS85 的逻辑图和引脚图

(a) 逻辑图 (b) 引脚图

表 4.4.3 4 位数值比较器 74LS85 的功能表

输 入											输 出		
A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_3 > B_3$		×		×		×		×	×	×	H	L	L
$A_3 < B_3$		×		×		×		×	×	×	L	H	L
$A_3 = B_3$	$A_2 > B_2$			×		×		×	×	×	H	L	L
$A_3 = B_3$	$A_2 < B_2$			×		×		×	×	×	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$				×		×	×	×	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$				×		×	×	×	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$					×	×	×	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$					×	×	×	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$					H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$					L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$					×	×	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$					H	H	L	L	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$					L	L	L	H	H	L

真值表中的输入变量包括 A_3 与 B_3 、 A_2 与 B_2 、 A_1 与 B_1 、 A_0 与 B_0 和 A 与 B 的比较结果。其中 A 和 B 是另外两个低位数, $I_{A>B}$ 、 $I_{A<B}$ 和 $I_{A=B}$ 是它们的比较结果。设置低位数比较结果输入端是为了能与其他数值比较器连接, 以便组成位数更多的数值比较器。由式(4.4.1)可知

$$F_{A>B} = A\bar{B}, F_{A<B} = \bar{A}B, F_{A=B} = A\odot B$$

从表 4.4.3 可得

$$\left. \begin{aligned}
 F_{A>B} &= A_3\bar{B}_3 + (A_3\odot B_3)A_2\bar{B}_2 + (A_3\odot B_3)(A_2\odot B_2)(A_1\bar{B}_1) + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)A_0\bar{B}_0 + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)(A_0\odot B_0)I_{A>B}\bar{I}_{A=B}\bar{I}_{A<B} + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)(A_0\odot B_0)\bar{I}_{A>B}\bar{I}_{A<B}\bar{I}_{A=B} \\
 F_{A=B} &= (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)(A_0\odot B_0)I_{A=B} \\
 F_{A<B} &= \bar{A}_3B_3 + (A_3\odot B_3)\bar{A}_2B_2 + (A_3\odot B_3)(A_2\odot B_2)\bar{A}_1B_1 + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)\bar{A}_0B_0 + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)(A_0\odot B_0)I_{A<B}\bar{I}_{A>B}\bar{I}_{A=B} + \\
 &\quad (A_3\odot B_3)(A_2\odot B_2)(A_1\odot B_1)(A_0\odot B_0)\bar{I}_{A>B}\bar{I}_{A<B}\bar{I}_{A=B}
 \end{aligned} \right\} \quad (4.4.2)$$

上式与逻辑图一致。由上式可以看出, 仅对 4 位数进行比较时, 应对 $I_{A>B}$ 、

$I_{A<B}$ 、 $I_{A=B}$ 进行适当处理, 即 $I_{A>B} = I_{A<B} = 0$, $I_{A=B} = 1$ 。

2. 数值比较器的位数扩展

现在来讨论一下数值比较器的位数扩展问题。数值比较器的扩展方式有串联和并联两种。图 4.4.4 表示两个 4 位数值比较器串联而成为一个 8 位数值比较器。我们知道, 对于两个 8 位数, 若高 4 位相同, 它们的大小则由低 4 位的比较结果确定。因此, 低 4 位的比较结果应作为高 4 位的条件, 即低 4 位比较器的输出端应分别与高 4 位比较器的 $I_{A>B}$ 、 $I_{A<B}$ 和 $I_{A=B}$ 端连接。

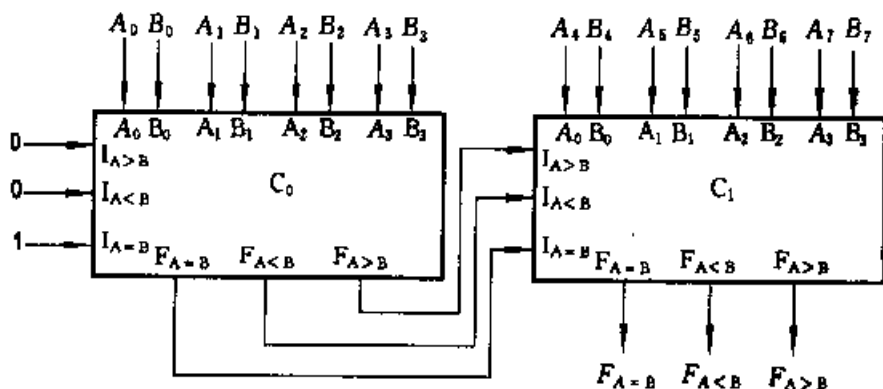


图 4.4.4 串联方式扩展数值比较器的位数

当位数较多且要满足一定的速度要求时, 可以采取并联方式。图 4.4.5 表示 16 位并联数值比较器的原理图。由图可以看出, 这里采用两级比较方法, 将 16 位按高低位次序分成 4 组, 每组 4 位, 各组的比较是并行进行的。将每组的比较结果再经 4 位比较器进行比较后得出结果。显然, 从数据输入到稳定输出只需两倍的 4 位比较器延迟时间, 若用串联方式, 则 16 位的数值比较器从输入到稳定输出需要 4 倍的 4 位比较器的延迟时间。

复习思考题

4.4.1 74LS85 的三个输入端, $I_{A>B}$ 、 $I_{A<B}$ 、 $I_{A=B}$ 有何作用?

4.4.2 用两片 74LS85 串联, 连接成 8 位数值比较器时, 低位片中的 $I_{A>B}$ 、 $I_{A<B}$ 、 $I_{A=B}$ 端应作何处理?

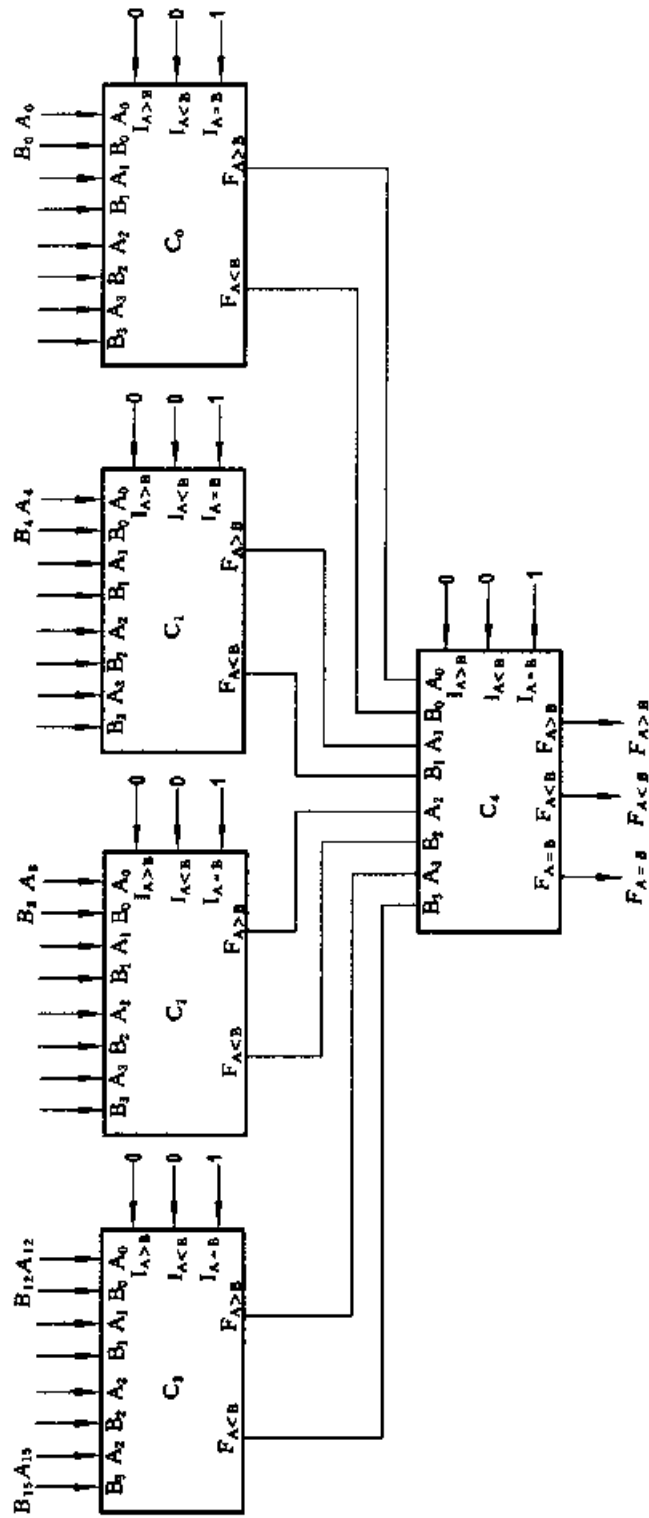


图 4.4.5 并联方式扩展数值比较器的位数

4.5 算术运算电路

算术运算是数字系统的基本功能,更是计算机中不可缺少的组成单元。本节介绍加法运算和减法运算的逻辑电路。(关于二进制算术运算的基本概念见附录 B)

4.5.1 半加器和全加器

1. 半加器

半加器和全加器是算术运算电路中的基本单元,它们是完成 1 位二进制数相加的一种组合逻辑电路。

两个 1 位二进制的加法运算可用真值表 4.5.1 表示,其中 S 表示和数, C 表示进位数。由表中逻辑关系可见,这种加法运算只考虑了两个加数本身,而没有考虑由低位来的进位,所以称为半加。半加器就是实现表 4.5.1 中逻辑关系的电路。

表 4.5.1 半加器真值表

被加数 A	加数 B	和数 S	进位数 C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

由真值表可得逻辑表达式

$$S = \bar{A}B + A\bar{B} \quad (4.5.1)$$

$$C = AB \quad (4.5.2)$$

利用加 $A\bar{A}$ 和摩根定律,可将上式变换成与非形式

$$S = \overline{\overline{\bar{A}B} \cdot \overline{A\bar{B}}} \quad (4.5.3)$$

$$C = \overline{\overline{AB}} \quad (4.5.4)$$

由式(4.5.3)和(4.5.4)可得由与非门组成的半加器,如图 4.5.1a 所示。

因为半加和 $S = \bar{A}B + A\bar{B}$ 是异或逻辑关系,所以半加器也可利用一个集成异或门和与门来实现,如图 4.5.1b 所示。图 4.5.1c 是半加器的代表符号。

2. 全加器

全加器能进行加数、被加数和低位来的进位信号相加,并根据求和结果给出该位的进位信号。

根据全加器的功能,可列出它的真值表,如表 4.5.2 所示。其中 A_i 和 B_i 分别是被加数及加数, C_{i-1} 为相邻低位来的进位数, S_i 为本位和数(称为全加和)

以及 C_i 为向相邻高位的进位数。

为了求出 S_i 和 C_i 的逻辑表达式, 首先分别画出 S_i 和 C_i 的卡诺图, 如图 4.5.2 所示。为了比较方便地获得与-或-非的表达式, 采用包围 0 的方法进行化简得

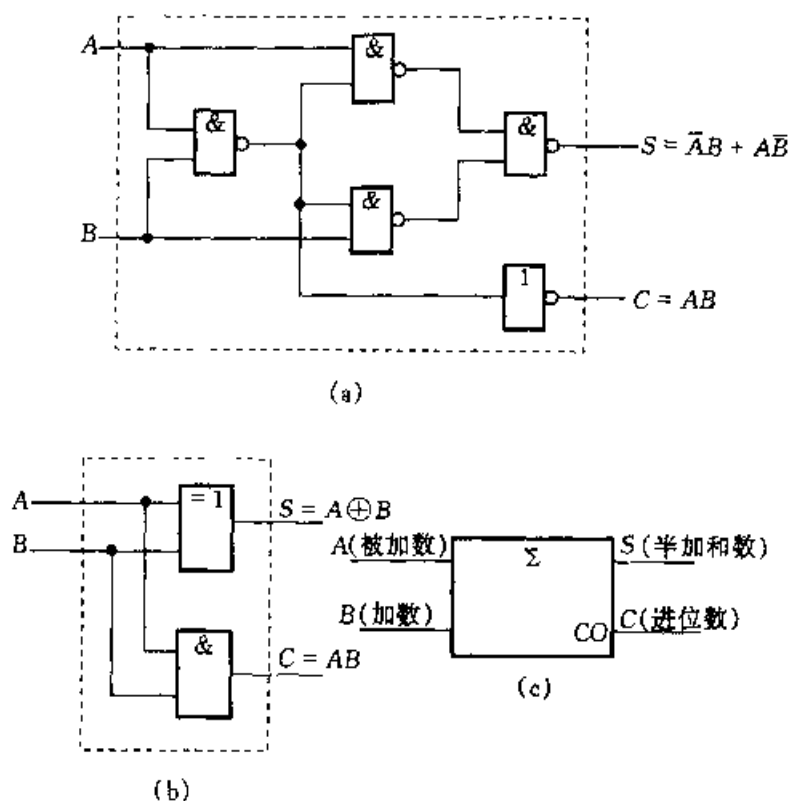


图 4.5.1 半加器

(a) 由与非门组成 (b) 由异或门及与门组成 (c) 半加器符号

$$\overline{S}_i = \overline{A}_i \overline{B}_i \overline{C}_{i-1} + \overline{A}_i B_i C_{i-1} + A_i \overline{B}_i C_{i-1} + A_i B_i \overline{C}_{i-1}$$

$$S_i = \overline{\overline{A}_i \overline{B}_i \overline{C}_{i-1} + \overline{A}_i B_i C_{i-1} + A_i \overline{B}_i C_{i-1} + A_i B_i \overline{C}_{i-1}} \quad (4.5.5)$$

$$\overline{C}_i = \overline{A}_i \overline{B}_i + \overline{B}_i \overline{C}_{i-1} + \overline{A}_i \overline{C}_{i-1}$$

$$C_i = \overline{\overline{A}_i \overline{B}_i + \overline{B}_i \overline{C}_{i-1} + \overline{A}_i \overline{C}_{i-1}} \quad (4.5.6)$$

表 4.5.2 全加器真值表

A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

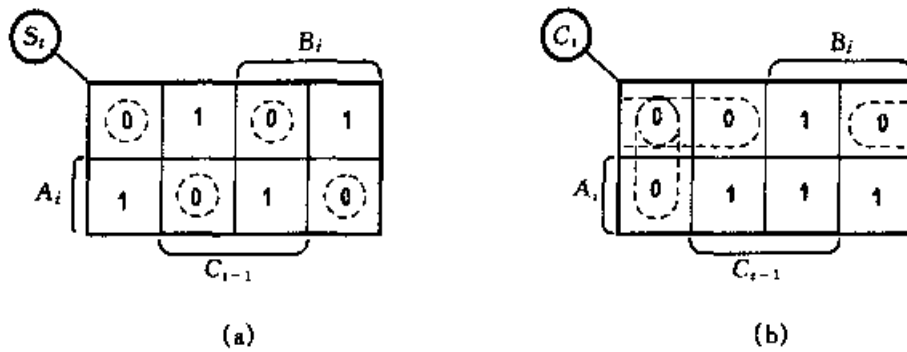


图 4.5.2 全加器的 S_i 和 C_i 卡诺图
(a) S_i 的卡诺图 (b) C_i 的卡诺图

由式(4.5.5)和式(4.5.6)可以画出 1 位全加器的逻辑图,如图 4.5.3a 所示,图 4.5.3b 是它的代表符号。

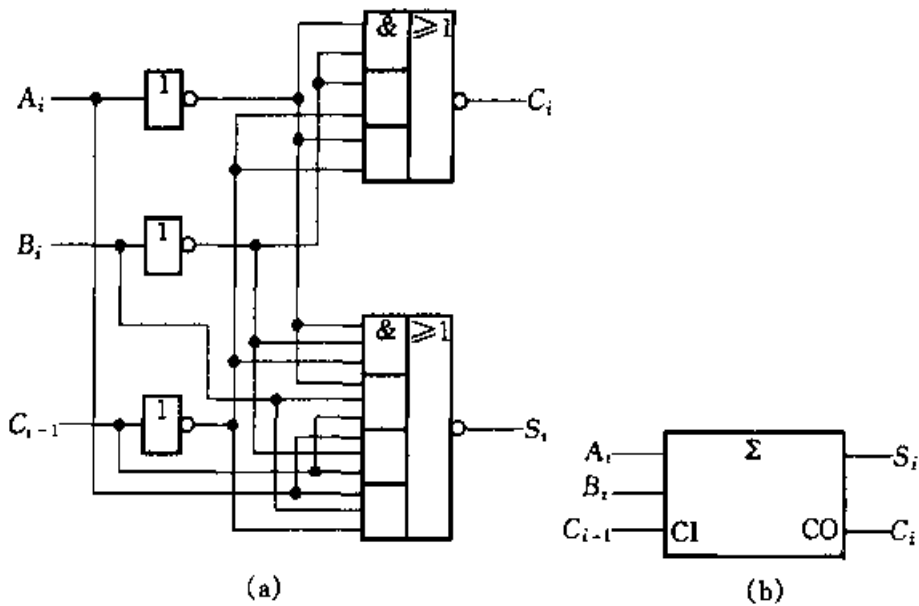


图 4.5.3 全加器
(a) 逻辑图 (b) 代表符号

4.5.2 多位数加法器

1. 串行进位加法器

若有多位数相加,则可采用并行相加串行进位的方式来完成。例如,有两个 4 位二进制数 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 相加,可以采用两片内含两个全加器或 1 片内含 4 个全加器的集成电路组成,其原理图如图 4.5.4 所示。由图可以看出,每 1 位的进位信号送给下 1 位作为输入信号,因此,任 1 位的加法运算必须

在低1位的运算完成之后才能进行,这种进位方式称为**串行进位**。这种加法器的逻辑电路比较简单,但它的运算速度不高。为克服这一缺点,可以采用**超前进位**等方式。

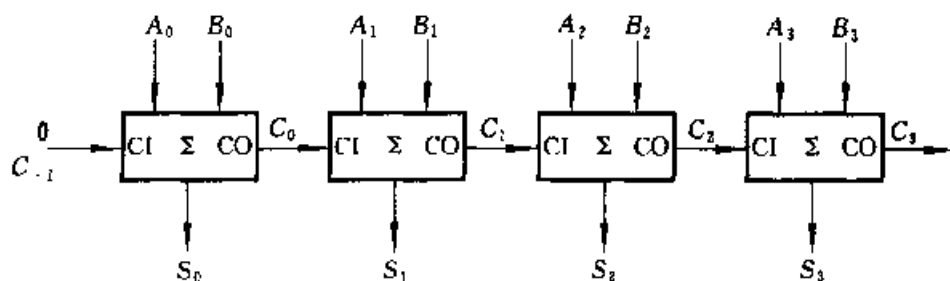


图 4.5.4 4位串行进位全加器

2. 超前进位集成4位加法器 74LS283

由于串行进位加法器的速度受到进位信号的限制,人们又设计了一种多位数超前进位加法逻辑电路,使每位的进位只由加数和被加数决定,而与低位的进位无关。现在介绍超前进位的概念。

由表 4.5.2 得 S_i 和 C_i 的逻辑表达式

$$\begin{aligned} S_i &= \overline{A_i} \overline{B_i} C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i} \overline{C_{i-1}} + A_i B_i C_{i-1} \\ &= (\overline{A_i} \oplus \overline{B_i}) C_{i-1} + (A_i \oplus B_i) \overline{C_{i-1}} = A_i \oplus B_i \oplus C_{i-1} \end{aligned} \quad (4.5.7)$$

$$\begin{aligned} C_i &= \overline{A_i} B_i C_{i-1} + A_i \overline{B_i} C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i B_i C_{i-1} \\ &= A_i B_i + (A_i \oplus B_i) C_{i-1} \end{aligned} \quad (4.5.8)$$

定义两个中间变量 G_i 和 P_i :

$$G_i = A_i B_i \quad (4.5.9)$$

$$P_i = A_i \oplus B_i \quad (4.5.10)$$

当 $A_i = B_i = 1$ 时, $G_i = 1$, 由式(4.5.8)得 $C_i = 1$, 即产生进位, 所以 G_i 称为**产生变量**。若 $P_i = 1$, 则 $A_i \cdot B_i = 0$, 由式(4.5.8)得 $C_i = C_{i-1}$, 即 $P_i = 1$ 时, 低位的进位能传送到高位的进位输出端, 故 P_i 称为**传输变量**。这两个变量都与进位信号无关。

将式(4.5.9)和式(4.5.10)代入式(4.5.7)和式(4.5.8), 得

$$S_i = P_i \oplus C_{i-1} \quad (4.5.11)$$

$$C_i = G_i + P_i C_{i-1} \quad (4.5.12)$$

由式(4.5.12)得各位进位信号的逻辑表达式如下:

$$C_0 = G_0 + P_0 C_{-1} \quad (4.5.13a)$$

$$C_1 = G_1 + P_1 C_0 = G_1 + P_1 G_0 + P_1 P_0 C_{-1} \quad (4.5.13b)$$

$$C_2 = G_2 + P_2 C_1 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1} \quad (4.5.13c)$$

$$C_3 = G_3 + P_3 C_2 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1} \quad (4.5.13d)$$

由式(4.5.13)可知,因为进位信号只与变量 G_i 、 P_i 和 C_{-1} 有关,而 C_{-1} 是向最低位的进位信号,其值为 0,所以各位的进位信号都只与两个加数有关,它们是可以并行产生的。

根据超前进位概念构成的集成 4 位加法器 74LS283 的逻辑图和引脚图分别如图 4.5.5a 和图 4.5.5b 所示。

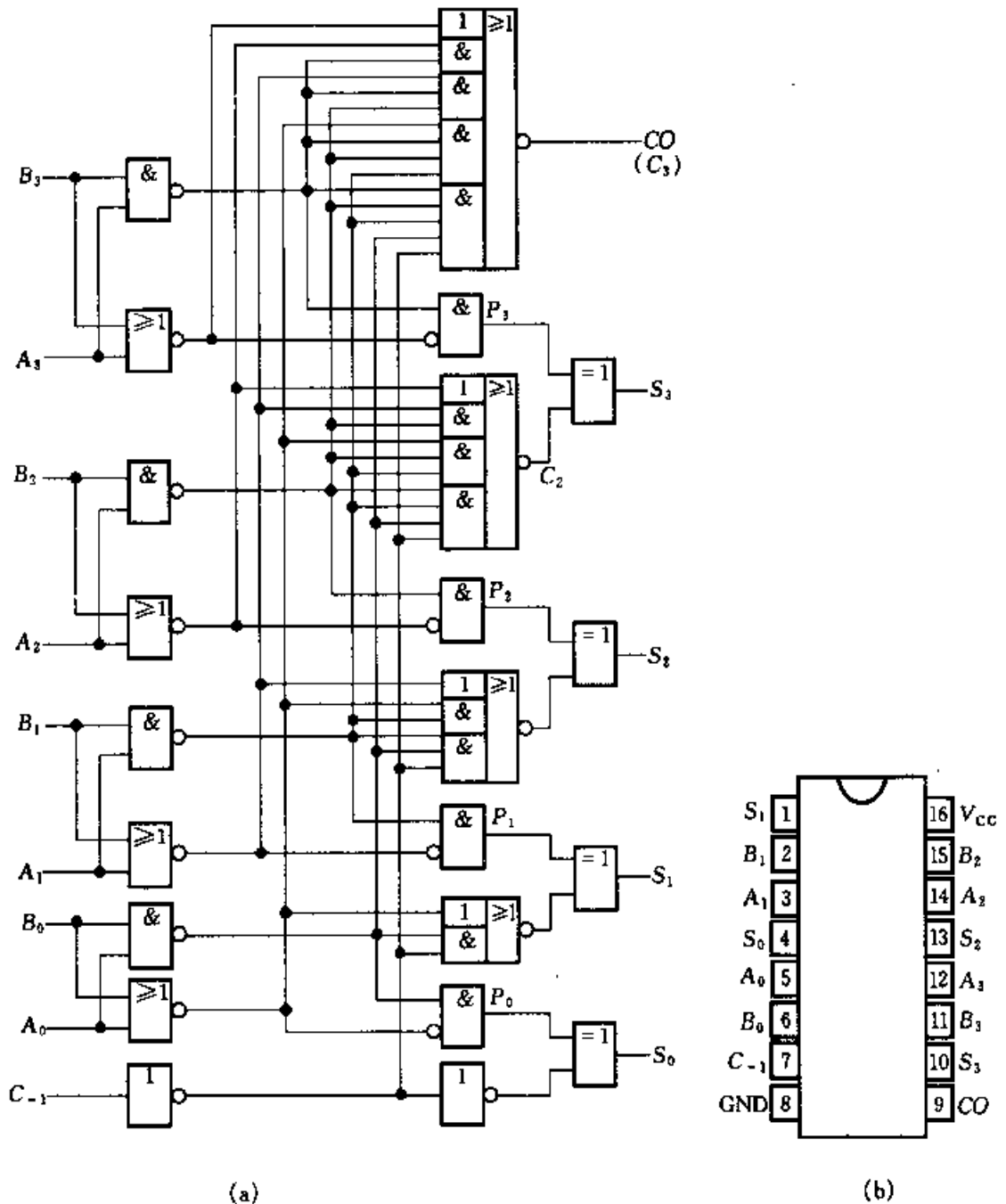


图 4.5.5 74LS283 逻辑图与引脚图

(a) 逻辑图 (b) 引脚图

根据式(4.5.11)和式(4.5.13a)~(4.5.13d)可以写出 $S_0 \sim S_3$ 的表达式。

例如

$$\begin{aligned} S_1 &= P_1 \oplus C_0 \\ &= P_1 \oplus (G_0 + P_0 C_{-1}) \end{aligned}$$

将式(4.5.9)和式(4.5.10)代入上式,得

$$S_1 = (A_1 \oplus B_1) \oplus [A_0 B_0 + (A_0 \bar{B}_0 + \bar{A}_0 B_0) C_{-1}]$$

经变换和化简得

$$S_1 = (A_1 \oplus B_1) \oplus \overline{A_0 B_0 \cdot C_{-1} + A_0 + B_0}$$

同理可推导出 S_0, S_2, S_3 和 $CO(C_3)$ 的表达式。

3. 超前进位产生器 74182

多位数的超前进位加法器的进位是并行产生的,大大提高了运算速度。但是,随着位数的增加,超前进位逻辑电路越来越复杂。为了解决这一矛盾,设计出了专用的超前进位产生器,用多个超前进位产生器连接,既可扩充位数而又不使逻辑电路太复杂。集成超前进位产生器 74182 的逻辑图和引脚图分别如图 4.5.6a 和 4.5.6b 所示。

由图 4.5.6 可以对 74182 的引出端信号说明如下。

74182 的输入、输出信号有进位输入端 C_n , 进位输出端 $C_{n+x}, C_{n+y}, C_{n+z}$, 进位产生输出端 F_G (低电平有效), 进位传输输出端 F_P (低电平有效), 进位产生输入端 $G_0 \sim G_3$ (低电平有效), 进位传输输入端 $P_0 \sim P_3$ (低电平有效)。

74182 的逻辑功能如表 4.5.3a~e 所示。

表 4.5.3(a) 74182 F_G 输出功能表

输 入							输出
G_3	G_2	G_1	G_0	P_3	P_2	P_1	F_G
L	×	×	×	×	×	×	L
×	L	×	×	L	×	×	L
×	×	L	×	L	L	×	L
×	×	×	L	L	L	L	L
所有其他组合							H

表 4.5.3(b) 74182 F_P 输出功能表

输 入				输出
P_3	P_2	P_1	P_0	F_P
L	L	L	L	L
所有其他组合				H

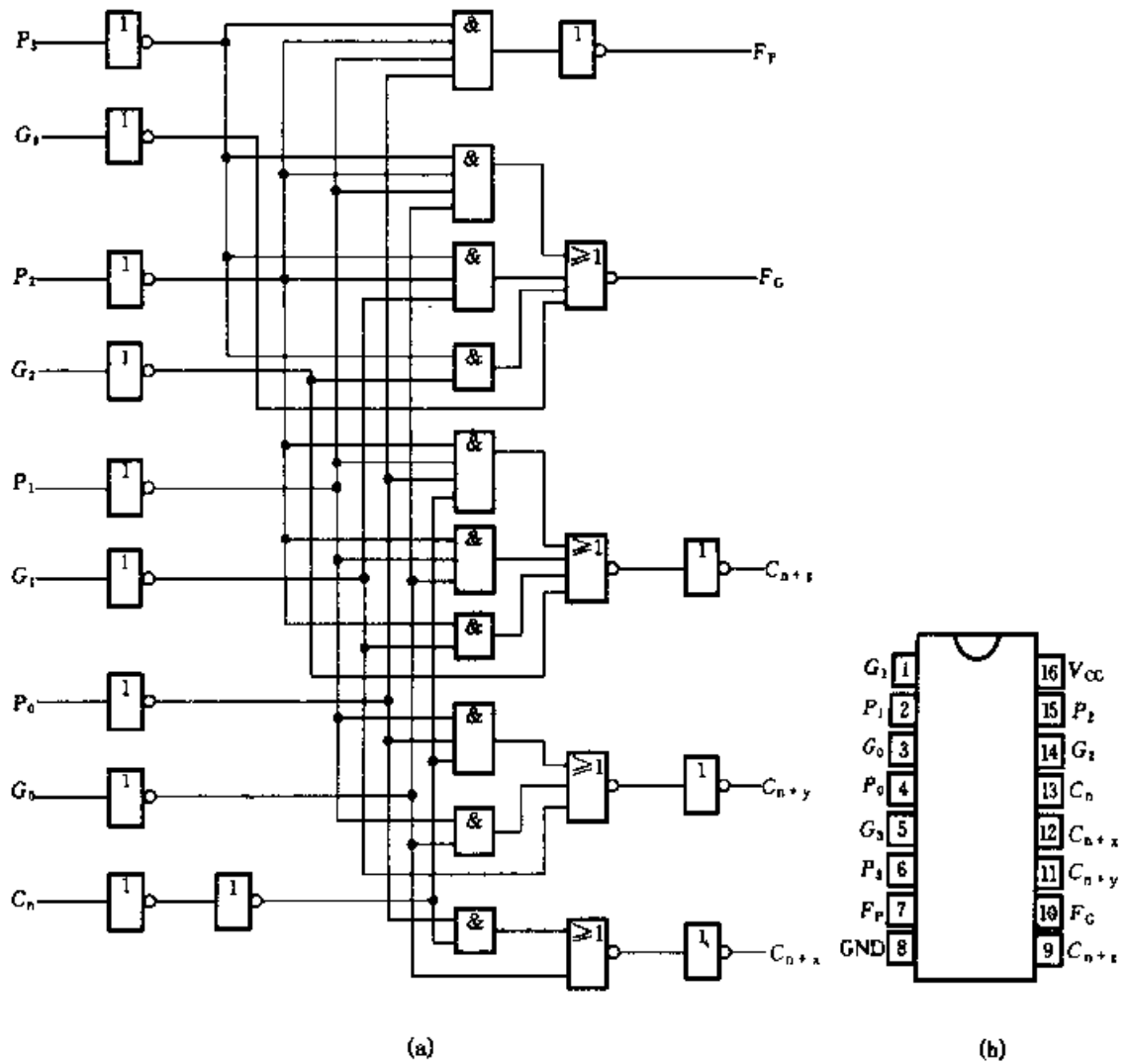


图 4.5.6 集成超前进位产生器 74182

(a) 逻辑图 (b) 引脚图

表 4.5.3(c) 74182 C_{n+x} 输出功能表

输入			输出
G_0	P_0	C_n	C_{n+x}
L	×	×	H
×	L	H	H
所有其他组合			L

表 4.5.3(d) 74182 C_{n+y} 输出功能表

输入						输出
G_1	G_0	P_1	P_0	C_n		C_{n+y}
L	×	×	×	×		H
×	L	L	×	×		H
×	×	L	L	H		H
所有其他组合						L

表 4.5.3(e) 74182 C_{n+z} 输出功能表

输 入							输出
G_2	G_1	G_0	P_2	P_1	P_0	C_n	C_{n+z}
L	×	×	×	×	×	×	H
×	L	×	L	×	×	×	H
×	×	L	L	L	×	×	H
×	×	×	L	L	L	H	H
所有其他组合							L

由图 4.5.6a 可得

$$C_{n+x} = \overline{P_0} C_n + \overline{G_0} \quad (4.5.14a)$$

$$C_{n+y} = \overline{P_1} \overline{P_0} C_n + \overline{P_1} \overline{G_0} + \overline{G_1} \quad (4.5.14b)$$

$$C_{n+z} = \overline{P_2} \overline{P_1} \overline{P_0} C_n + \overline{P_2} \overline{P_1} \overline{G_0} + \overline{P_2} \overline{G_1} + \overline{G_2} \quad (4.5.14c)$$

$$\overline{F_p} = \overline{P_3} \overline{P_2} \overline{P_1} \overline{P_0} \quad (4.5.14d)$$

$$\overline{F_G} = \overline{P_3} \overline{P_2} \overline{P_1} \overline{G_0} + \overline{P_3} \overline{P_2} \overline{G_1} + \overline{P_3} \overline{G_2} + \overline{G_3} \quad (4.5.14e)$$

式(4.5.14a~4.5.14c)与功能表一致,它们对应于式(4.5.13a~4.5.13c), C_{n+x} 、 C_{n+y} 、 C_{n+z} 为各位的进位信号, P_i 、 G_i 为低电平有效, C_n 是向最低位的进位信号。 F_p 和 F_G 可以用来实现多个超前进位产生器连接,这样既可以扩充位数,而又不使逻辑电路太复杂。具体连接方法见后面图 4.5.10。

4.5.3 减法运算

同加法运算一样,减法运算可采用减法器来实现。半减器和全减器的设计方法和步骤与设计加法器相同,读者可自行练习(见习题 4.5.6)。

实用上,为了简化系统结构,通常不另外设计减法器,而是将减法运算变为加法运算来处理,使运算器既能实现加法运算,又可实现减法运算。一般采用加补码的方法代替减法运算,下面先来介绍这种方法的原理。

1. 反码和补码

这里只讨论数值码,即数码中不包括符号位。以前应用的自然二进制码称为原码,所谓反码就是将原码中的所有 0 变为 1,所有 1 变为 0 后的代码。观察如下几组原码与反码之间的关系。

$$\begin{array}{r} \phantom{N_{\text{原}}} \\ N_{\text{原}} = \begin{array}{ccc} 1111 & 1111 & 1111 \\ \underline{0000} & \underline{0001} & \underline{0101} \\ 1111 & 1110 & 1010 \end{array} \\ N_{\text{反}} = \end{array}$$

显然,每组反码都是从 1111 中减去原码的结果,所以,可得如下反码与原码的一般关系式

$$N_{\text{反}} = (2^n - 1) - N_{\text{原}} \quad (4.5.15)$$

其中 n 等于数码的位数。

定义补码为

$$N_{\text{补}} = 2^n - N_{\text{原}} \quad (4.5.16)$$

于是,由式(4.5.15)和式(4.5.16)便可得到补码和反码的关系式

$$N_{\text{补}} = N_{\text{反}} + 1 \quad (4.5.17)$$

由以上分析可知,一个数的反码可将原码经反相器获得,而由反码加1就可得到补码。

2. 由加补码完成减法运算

由式(4.5.15)可得两数 A 、 B 相减的表达式

$$A - B = A + B_{\text{补}} - 2^n = A + B_{\text{反}} + 1 - 2^n \quad (4.5.18)$$

上式表明, A 减 B 可由 A 加 B 的补码并减 2^n 完成。4 位减法运算电路如图 4.5.7 所示。由 4 个反相器将 B 的各位反相(求反),并将进位输入端 C_{-1} 接逻辑 1 以实现加 1,由此求得 B 的补码。显然,只能由高位的进位信号与 2^n 相减。当最高位的进位信号为 1(2^n)时,它们的差为 0;最高位的进位信号为 0 时,它与 2^n 相减所得的差为 1,同时还应发出借位信号。因此,只要将最高位的进位信号反相即实现了减 2^n 的运算,反相器的输出 V 为 1 时需要借位,故 V 为借位信号。下面分两种情况分析减法运算过程。

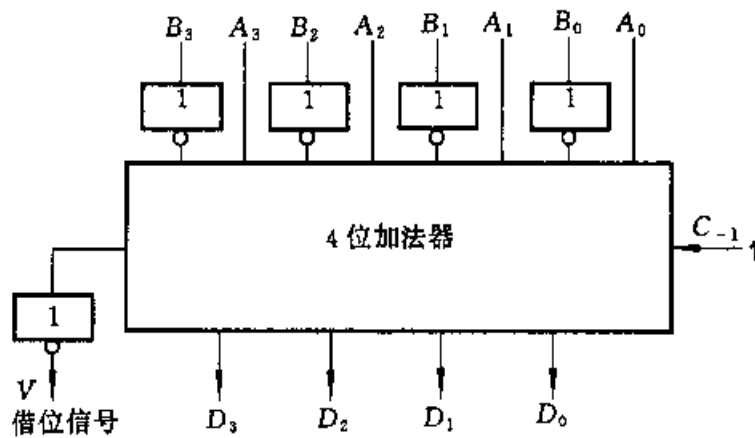


图 4.5.7 4 位减法运算逻辑图

(1) $A - B \geq 0$ 的情况。设 $A = 0101$, $B = 0001$ 。

求补相加演算过程如下:

$$\begin{array}{r}
 0101 \quad (A) \\
 1110 \quad (B_{\text{反}}) \\
 + \quad \quad 1 \quad (\text{加}1) \\
 \hline
 10100 \\
 \downarrow \\
 00100 \quad (\text{进位反相}) \\
 \uparrow \\
 (\text{借位})
 \end{array}$$

直接作减法演算,则有

$$\begin{array}{r} 0101 \\ - 0001 \\ \hline 0100 \end{array}$$

比较两种运算结果,它们完全相同。在 $A - B \geq 0$ 时,所得的差就是差的原码,借位信号为 0。

(2) $A - B < 0$ 的情况。设 $A = 0001, B = 0101$ 。

求补相加演算过程如下:

$$\begin{array}{r} 0001 (A) \\ 1010 (B_{\bar{}}) \\ + \quad 1 (加1) \\ \hline 01100 \\ 11100 (进位反相) \end{array}$$

(借位) \leftarrow

直接作减法演算,则有

$$\begin{array}{r} 0001 \\ - 0101 \\ \hline - 0100 \end{array}$$

(符号) \leftarrow

比较两种运算结果可知,前者正好是后者的绝对值的补码,借位信号 V 为 1 时表示差为负数, V 为 0 时差为正数。若要求差值以原码形式输出,则还需进行变换。由式(4.5.16)可知 $N_{\text{原}} = 2^n - N_{\text{补}}$,即将补码再求补得原码。这时的求补逻辑电路如图 4.5.8 所示,它和图 4.5.7 共同组成输出为原码的完整的 4 位减法运算电路。求补相加而得的差输入到不同的异或门的一个输入端,而另一输入端由借位信号 V 控制。当 $V = 1$ 时, $D_3 \sim D_0$ 反相,并和 $C_{-1} = 1$ 相加,实现求补运算; $V = 0$ 时, $D_3 \sim D_0$ 不反相,加法器也不实现加 1 运算,维持原码。

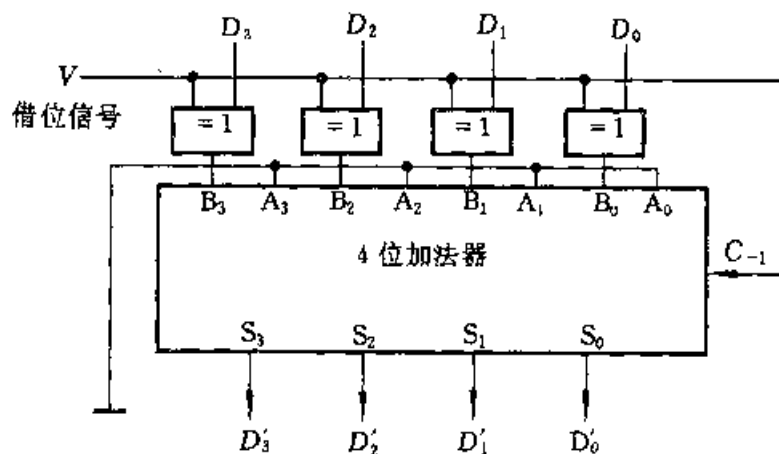


图 4.5.8 由符号决定求补的逻辑图

算术运算还包括乘法、除法运算,这里不作介绍。

4.5.4 集成算术/逻辑单元举例

集成算术/逻辑单元(ALU)^①能够完成一系列的算术运算和逻辑运算。74LS381是比较简单的双极型ALU,它的功能如表4.5.4所示,引脚图如图4.5.9所示。

表 4.5.4 74LS381 功能表

选 择			算术/逻辑操作
S_2	S_1	S_0	
0	0	0	清零
0	0	1	B 减 A
0	1	0	A 减 B
0	1	1	A 加 B
1	0	0	$A \oplus B$
1	0	1	$A + B$
1	1	0	$A \cdot B$
1	1	1	预置

该算术/逻辑单元可以对两个4位数据 A 和 B 进行6种算术或逻辑运算,并有清零和预置功能。所谓清零是使各数据输出端的状态为0,预置是使数据输出端处于预定的状态。输入信号 $S_2 \sim S_0$ 选择8种不同的运算功能。进行算术运算时,其输出 $F_3 F_2 F_1 F_0$ 为二进制数;进行逻辑运算时,则为含一定意义的代码。进行预置操作时,预定的状态从 A_3, A_2, A_1, A_0 端输入。

可以根据需要把若干片4位ALU与超前进位产生器连接,扩展ALU的位数。图4.5.10表示了16位的ALU的连接方法。这种运算电路从最低位到最高位都是超前进位,常称为全超前进位电路。

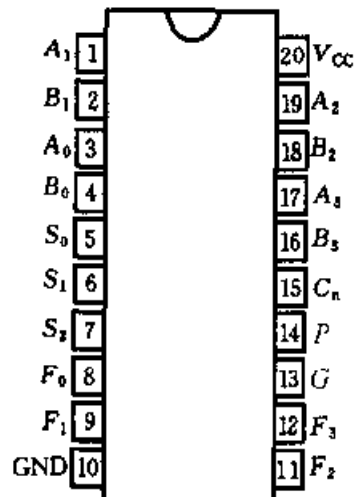


图 4.5.9 74LS381 引脚图

复习思考题

4.5.1 什么是半加器? 什么是全加器?

^① ALU为Arithmetic Logic Unit的缩写。

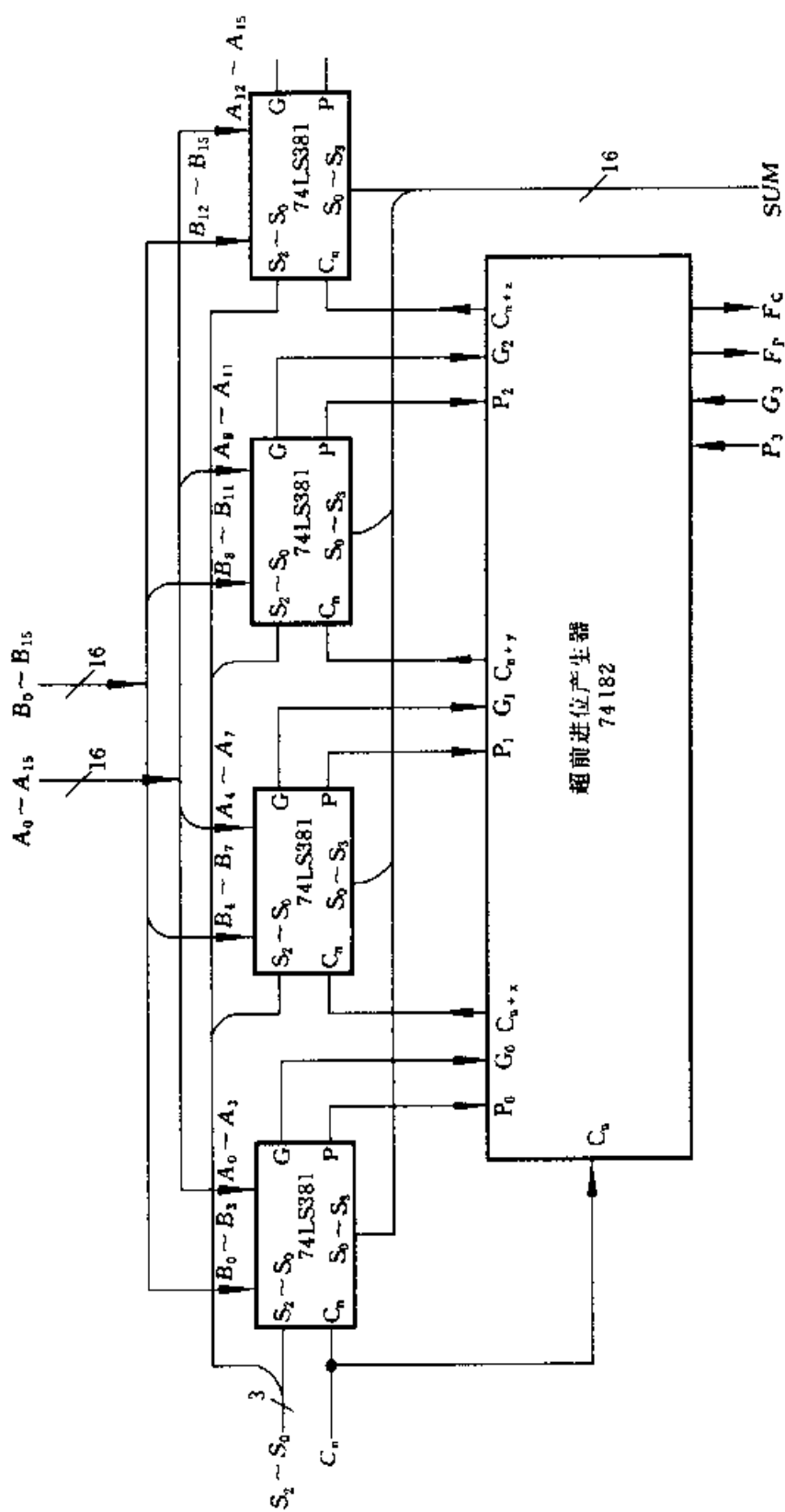


图 4.5.10 16 位全超前进位算术/逻辑运算电路

4.5.2 超前进位加法和串行进位加法的区别是什么?

4.5.3 74182 中的 F_P 和 F_G 的作用是什么?

4.5.4 说明反码和补码之间的关系。

4.5.5 简要说明由加补码完成减法运算的原理。

* 4.6 CAD 例题

例 CE4.6.1 试用 ABEL 语言设计一个七段显示译码器,其功能是将 4 位二进制数译为七段输出信号,驱动共阳极显示器显示数码。

解: (1) 逻辑设计。

驱动共阳极显示器的信号是低电平有效,即哪一段的驱动信号为低电平,则对应段发光。假设七段显示译码器的输入信号为 D_3 、 D_2 、 D_1 、 D_0 ,输出信号为 A 、 B 、 C 、 D 、 E 、 F 、 G ,分别与共阳极显示器的显示段相对应;另外,显示译码器还有一个使能输入控制信号 EN ,当 $EN=0$ 时译码器工作;当 $EN=1$ 时,译码器输出全 1 电平,显示器的各段不发光。于是,可以列出七段显示译码器的真值表,如表 4.6.1 所示。

表 4.6.1

输 入					输 出						
EN	D_3	D_2	D_1	D_0	A	B	C	D	E	F	G
1	×	×	×	×	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0	1	1	1	1
0	0	0	1	0	0	0	1	0	0	1	0
0	0	0	1	1	0	0	0	0	1	1	0
0	0	1	0	0	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	0	0	1	0	0	0
0	1	0	1	1	1	1	0	0	0	0	0
0	1	1	0	0	0	1	1	0	0	0	1
0	1	1	0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	1	1	0	0	0	0
0	1	1	1	1	0	1	1	1	0	0	0

(2) 输入设计文件。使用 ABEL 语言中的真值表结构,可以直接描述上述显示译码器真值表所表示的逻辑关系。

打开 ISP Synario 软件的文本编辑器,输入并保存下列 ABEL 语言的源文件(包括测试向量文件)。然后对源文件进行编译和逻辑简化。

显示译码器的 ABEL 语言源程序(包括测试向量)如下:

```

module BIN7ROM                                "模块开始
title 'SEVEN SEGMENT DISPLAY DECODER'
    D3,D2,D1,D0          pin;                "定义输入信号
    EN                   pin;                "定义输入信号
    A,B,C,D,E,F,G       pin istype 'com';   "定义输出信号
    BIN = [D3,D2,D1,D0]; "定义输入信号组
    LED = [A,B,C,D,E,F,G]; "定义输出信号组
    ON,OFF = 0,1;       "用 ON、OFF 表示共阳极显
                        "示器的亮或灭

    L,H,X = 0,1,..X.;   "定义常数

truth _ table([EN,BIN] -> LED)              "真值表结构段开始
"
    A      B      C      D      E      F      G
[L,H,X] -> [OFF, OFF, OFF, OFF, OFF, OFF, OFF];
[L,0] -> [ON, ON, ON, ON, ON, ON, OFF];
[L,1] -> [OFF, ON, ON, OFF, OFF, OFF, OFF];
[L,2] -> [ON, ON, OFF, ON, ON, OFF, ON];
[L,3] -> [ON, ON, ON, ON, OFF, OFF, ON];
[L,4] -> [OFF, ON, ON, OFF, OFF, ON, ON];
[L,5] -> [ON, OFF, ON, ON, OFF, ON, ON];
[L,6] -> [ON, OFF, ON, ON, ON, ON, ON];
[L,7] -> [ON, ON, ON, OFF, OFF, OFF, OFF];
[L,8] -> [ON, ON, ON, ON, ON, ON, ON];
[L,9] -> [ON, ON, ON, ON, OFF, ON, ON];
[L,10] -> [ON, ON, ON, OFF, ON, ON, ON]; "A
[L,11] -> [OFF, OFF, ON, ON, ON, ON, ON]; "b
[L,12] -> [ON, OFF, OFF, ON, ON, ON, OFF]; "C
[L,13] -> [OFF, ON, ON, ON, ON, OFF, ON]; "d
[L,14] -> [ON, OFF, OFF, ON, ON, ON, ON]; "E
[L,15] -> [ON, OFF, OFF, OFF, ON, ON, ON]; "F
test _ vectors([EN,BIN] -> LED)            "测试向量段开始

```

```

[L,0] ->X;           [L,9] ->X;
[L,1] ->X;           [L,10] ->X;
[L,2] ->X;          [L,11] ->X;
[L,3] ->X;          [L,12] ->X;
[L,4] ->X;          [L,13] ->X;
[L,5] ->X;          [L,14] ->X;
[L,6] ->X;          [L,15] ->X;
[L,7] ->X;          [H,0] ->X;
[L,8] ->X;          [H,9] ->X;

```

end bin7rom

(3) 进行逻辑功能仿真。对包含在源程序中的测试向量文件进行编译和逻辑功能仿真,得到如图 CE4.6.1 所示的逻辑功能仿真波形图。

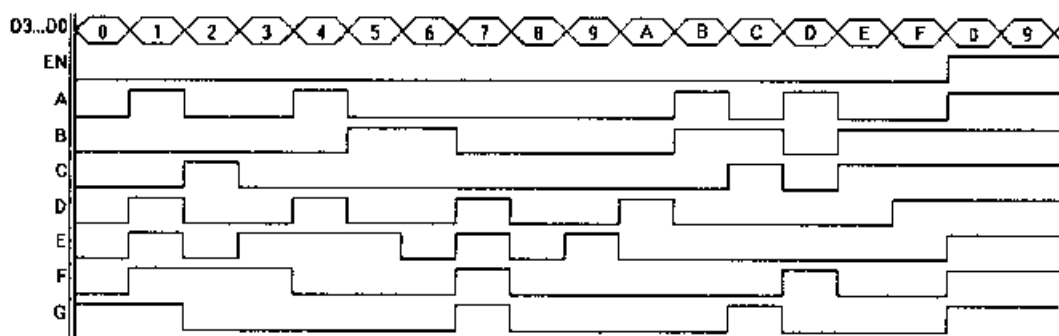


图 CE4.6.1 逻辑功能仿真波形图

例 CE4.6.2 试用 ABEL 语言设计一个数据分配器电路,并用 ISP Synario 软件对所设计的电路进行分析,给出逻辑功能仿真波形图。数据分配器的示意图如图 CE4.6.2 所示。图中,DATA 为数据输入信号,S2、S1、S0 为通道选择控制信号,Y0…Y7 为输出信号通道。假设所有的输出信号受输出使能信号 OE 控制,当 OE = 1 时数据分配器工作;当 OE = 0 时,数据分配器的输出呈现高阻抗状态。

解: (1) 逻辑设计

根据题意,可以列出数据分配器的逻辑方程如下:

$$\begin{aligned}
 Y7 &= EN \cdot (S2 \cdot S1 \cdot S0) \cdot DATA, & Y6 &= EN \cdot (S2 \cdot S1 \cdot \overline{S0}) \cdot DATA, \\
 Y5 &= EN \cdot (S2 \cdot \overline{S1} \cdot S0) \cdot DATA, & Y4 &= EN \cdot (S2 \cdot \overline{S1} \cdot \overline{S0}) \cdot DATA, \\
 Y3 &= EN \cdot (\overline{S2} \cdot S1 \cdot S0) \cdot DATA, & Y2 &= EN \cdot (\overline{S2} \cdot S1 \cdot \overline{S0}) \cdot DATA, \\
 Y1 &= EN \cdot (\overline{S2} \cdot \overline{S1} \cdot S0) \cdot DATA, & Y0 &= EN \cdot (\overline{S2} \cdot \overline{S1} \cdot \overline{S0}) \cdot DATA.
 \end{aligned}$$

(2) 输入设计文件。使用 ABEL 语言中的逻辑方程结构,可以直接描述上述数据分配器的逻辑关系。

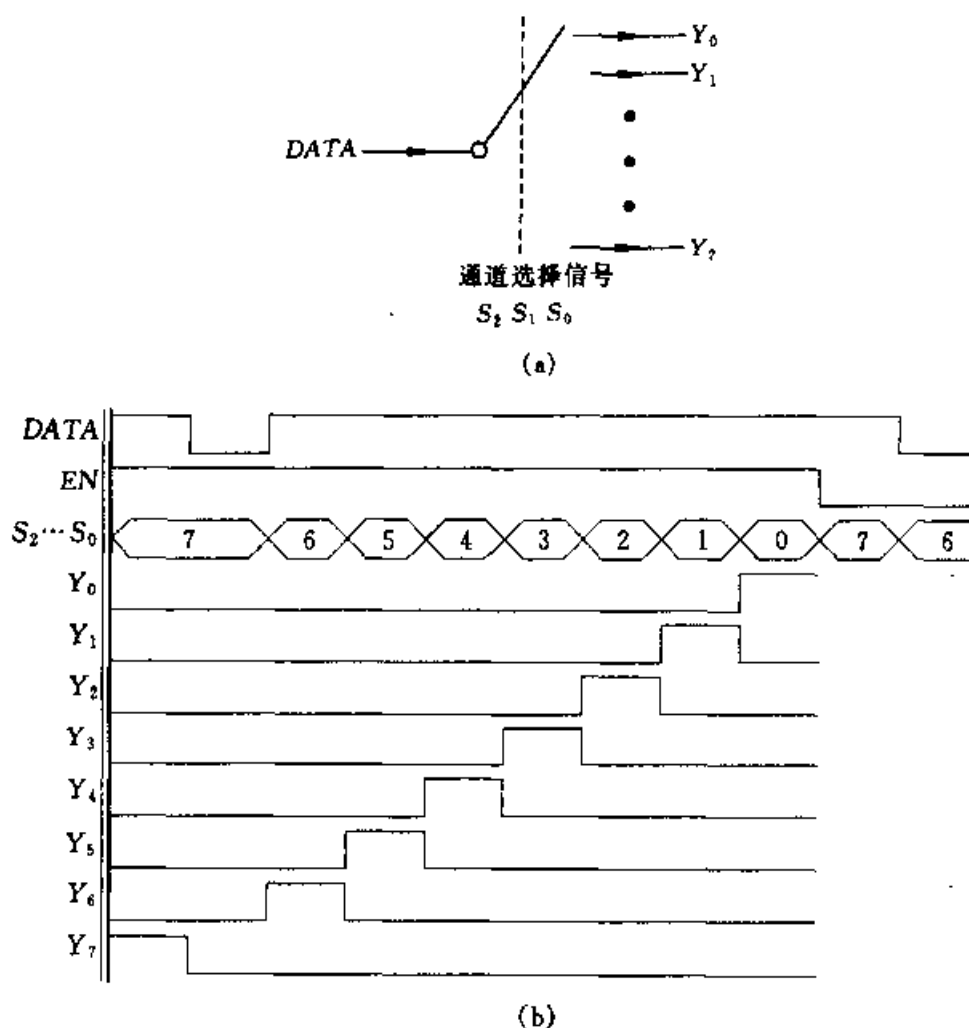


图 CE4.6.2 数据分配器的示意图及其逻辑功能仿真波形图
(a) 原理示意图 (b) 仿真波形图

打开 ISP Synario 软件的文本编辑器,输入并保存下列 ABEL 语言的源文件(包括测试向量文件)。然后对源文件进行编译和逻辑简化。

数据分配器的 ABEL 语言源程序(包括测试向量)如下:

```

module DMUX1TO8
    DATA                                pin;
    EN,S2,S1,S0                          pin;
    Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0            pin istype 'com';
    Z = .Z.;                             "定义常数
    SELECT = [S2,S1,S0];                 "定义输入信号组
    OUTPUTS = [Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0];
equations                                "逻辑方程段开始
    OUTPUTS.OE = EN;                    "定义所有的输出信号受 OE 的控制

```

```

Y7 = (SELECT == 7) & DATA;
Y6 = (SELECT == 6) & DATA;
Y5 = (SELECT == 5) & DATA;
Y4 = (SELECT == 4) & DATA;
Y3 = (SELECT == 3) & DATA;
Y2 = (SELECT == 2) & DATA;
Y1 = (SELECT == 1) & DATA;
Y0 = (SELECT == 0) & DATA;

```

```

test _ vectors          "测试向量段开始
([EN,SELECT,DATA] -> [Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0])
[1,7,1] -> [1,0,0,0,0,0,0,0]; "SELECT Y7
[1,7,0] -> [0,0,0,0,0,0,0,0]; "SELECT Y7
[1,6,1] -> [0,1,0,0,0,0,0,0]; "SELECT Y6
[1,5,1] -> [0,0,1,0,0,0,0,0]; "SELECT Y5
[1,4,1] -> [0,0,0,1,0,0,0,0]; "SELECT Y4
[1,3,1] -> [0,0,0,0,1,0,0,0]; "SELECT Y3
[1,2,1] -> [0,0,0,0,0,1,0,0]; "SELECT Y2
[1,1,1] -> [0,0,0,0,0,0,1,0]; "SELECT Y1
[1,0,1] -> [0,0,0,0,0,0,0,1]; "SELECT Y0

test _ vectors          "测试向量段开始
([EN,SELECT,DATA] -> [Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0])
[0,7,1] -> [Z,Z,Z,Z,Z,Z,Z,Z];
[0,6,0] -> [Z,Z,Z,Z,Z,Z,Z,Z];

end

```

(3) 进行逻辑功能仿真。对包含在源文件中的测试向量文件进行编译和逻辑功能仿真,得到如图 CE4.6.2b 所示的逻辑功能仿真波形图。波形图中的阴影区为高阻态。

小 结

• 常用的中规模组合逻辑器件包括编码器、译码器、数据选择器、数值比较器、奇偶校验/产生器、加法器及算术逻辑运算单元等。这些组合逻辑器件除了具有其基本功能外,通常还具有输入使能、输出使能、输入扩展、输出扩展功能,使其功能更加灵活,便于构成较复杂的逻辑系统。

• 应用组合逻辑器件进行组合逻辑电路设计时,所应用的原理和步骤与用门电路时是基本一致的,但也有其特殊之处。

a. 对逻辑表达式的变换与化简的目的是使其尽可能与组合逻辑器件的形式一致,而不是尽量简化。

b. 设计时应考虑合理充分应用组合器件的功能。同种类的组合器件有不同的型号,应尽量选用较少的器件数和较简单的器件满足设计要求。

c. 可能出现只需一个组合器件的部分功能就可以满足要求,这时需要对有关输入、输出信号作适当的处理。也可能会出现一个组合器件不能满足设计要求的情况,这就需要对组合器件进行扩展,直接将若干个器件组合或者由适当的逻辑门将若干个器件组合起来。

习 题

4.1.1 一编码器的真值表如表题 4.1.1 所示,试用或非门和反相器设计出该编码器的逻辑电路。

表题 4.1.1

输 入				输 出							
I_3	I_2	I_1	I_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1	0	1	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	0	0	1	1	1	0	0	1	1	0	1

4.1.2 试设计一个 10 位总线 $B = B_9 \cdots B_0$ 的状态标志逻辑电路。当任 1 位总线为 0 时,输出 P 为 0。另外还要求用 4 位输出状态 S_3, S_2, S_1, S_0 区分是哪 1 位总线为 0,当同时有几位总线都为 0 时,则指示出其中的最高位。

4.1.3 对例 4.1.2 加以完善,使得该十进制 BCD 码转换器具有输入使能和输出使能功能。

4.1.4 例 4.1.1 中的输出信号 D, C, B, A 及 GS 为低电平有效,如果将它们改为高电平有效,逻辑电路应作何修改? 按要求画出逻辑电路。

4.2.1 试用与非门设计一译码器,译出对应 $ABCD = 0010, 1010, 1110$ 状态的 3 个信号。

4.2.2 为了使 74138 译码器的第 10 脚输出为低电平,请标出各输入端应置的逻辑电平。

4.2.3 用译码器 74138 和适当的逻辑门实现函数 $F = \overline{A} \overline{B} \overline{C} + A \overline{B} \overline{C} + AB \overline{C} + ABC$ 。

4.2.4 应用译码器 74138 设计一个能对 32 个地址进行译码的译码系统。

4.2.5 译码器的功能如表题 4.2.5 所示。

(1) 用逻辑门设计该译码器;

(2) 用 74138 设计该译码器。

表题 4.2.5

选择输入				译码输出									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1	0

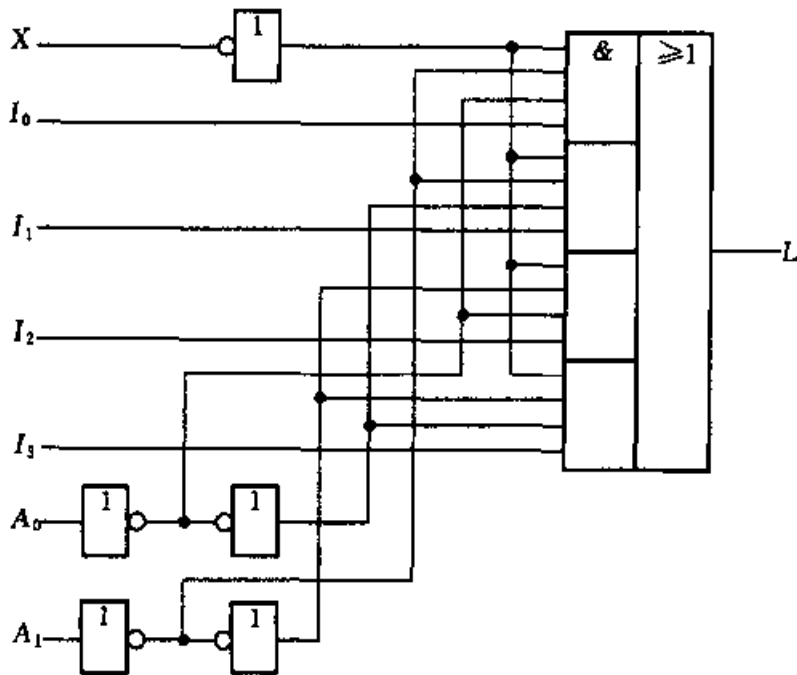
4.2.6 应用 74138 和其他逻辑门设计一地址译码器,要求地址范围是十六进制 00-3F。

4.2.7 指出题 4.2.6 中对应十六进制地址码 07、0E、13、2C、3B 的输入。

4.2.8 用逻辑门对 7442 的功能作修改,增加低电平使能输入功能。要求该输入为高电平时,所有输出为高电平。

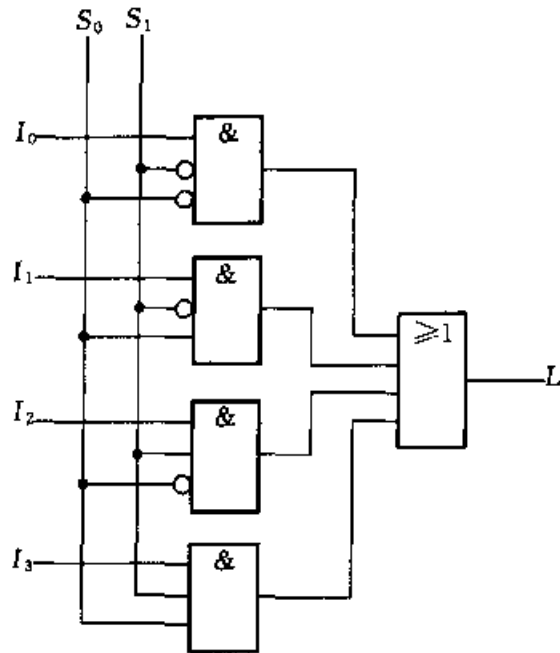
4.2.9 使用七段集成显示译码器 7448 和发光二极管显示器组成一个 7 位数字的译码显示电路,要求将 0099.120 显示成 99.12,各片的控制端应如何处理?画出外部接线图。(注:不考虑小数点的显示)

4.3.1 数据选择器如图题 4.3.1 所示,并行输入数据 $I_3 I_2 I_1 I_0 = 1010$,控制端 $X = 0$, $A_1 A_0$ 的态序为 00、01、10、11,试画出输出端 L 的波形。



图题 4.3.1

4.3.2 数据选择器如图题 4.3.2 所示。当 $I_3 = 0, I_2 = I_1 = I_0 = 1$ 时, 有 $L = \bar{S}_1 + S_1 \bar{S}_0$ 的关系, 证明该逻辑表达式的正确性。



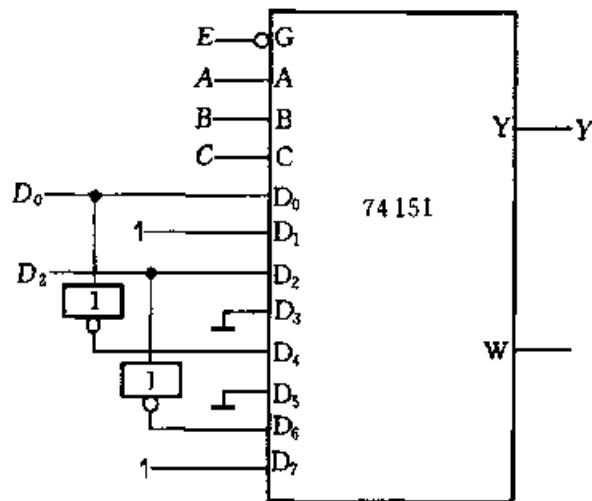
图题 4.3.2

4.3.3 应用图题 4.3.2 所示的电路产生逻辑函数 $F = S_1 + S_0$ 。

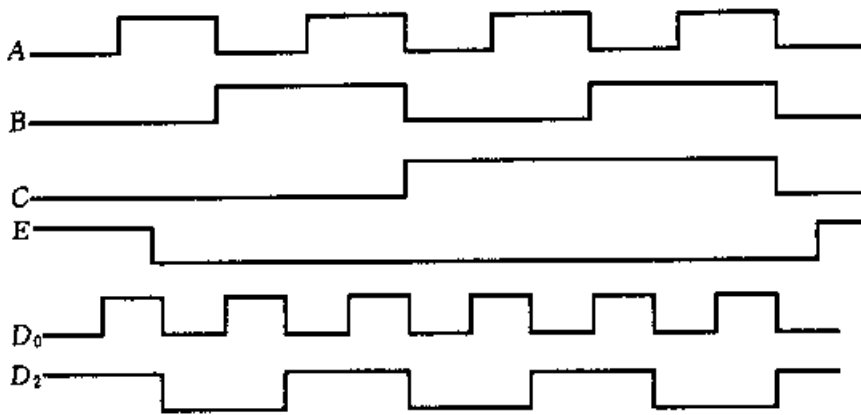
4.3.4 设计一 4 选 1 数据选择器。数据输入是 I_0, I_1, I_2, I_3 , 数据输出是 Y , 4 个控制信号为 S_0, S_1, S_2, S_3 。要求只当 $S_1 = 1$ 时, I_i 与 Y 接通, 且由另一控制信号 E 作为该选择器的使能信号。

- (1) 画出由反相器、2 输入与门和或门实现的逻辑电路;
- (2) 选择一合适的三态门作为输出级。

4.3.5 74151 的连接方式和各输入端的输入波形如图题 4.3.5 所示, 画出输出端 Y 的波形。



(a)



(b)

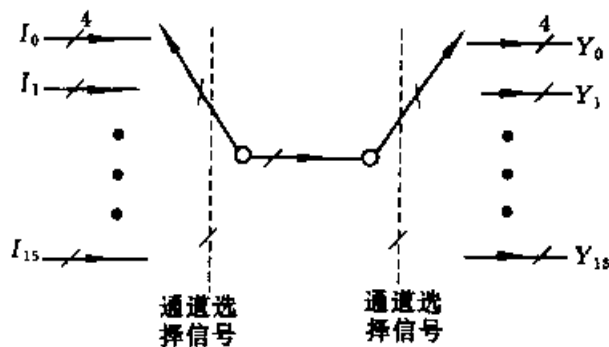
图题 4.3.5

4.3.6 应用 74151 实现如下逻辑功能

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

$$(2) Y = (A \odot B) \odot C$$

4.3.7 应用已介绍过的中规模组合逻辑电路设计一个 4 位数据传输电路,其功能是将 16 个输入数据中的任何一个传送到 16 个输出端中的任何一个输出端,其示意图如图题 4.3.7 所示。



图题 4.3.7

4.4.1 试用三个 3 输入端与门和一个或门实现语句“ $A > B$ ”, A 和 B 均为两位二进制数。

4.4.2 试用五个双输入端或门和一个与门实现语句“ $A > B$ ”, A 和 B 均为两位二进制数。

4.4.3 试设计一个 8 位相同数值比较器,当两数相等时,输出 $L = 1$,否则 $L = 0$ 。

4.4.4 在图 4.4.5 的基础上,增加一片同样的数值比较器构成二十四位的数值比较电路。

4.5.1 试用两个半加器和一个或门构成一全加器。

(1) 写出 S_i 和 C_i 的逻辑表达式;

(2) 画出逻辑图。

4.5.2 试证明等式

$$X \oplus Y = \overline{\overline{XYX} \cdot \overline{XYY}}$$

和

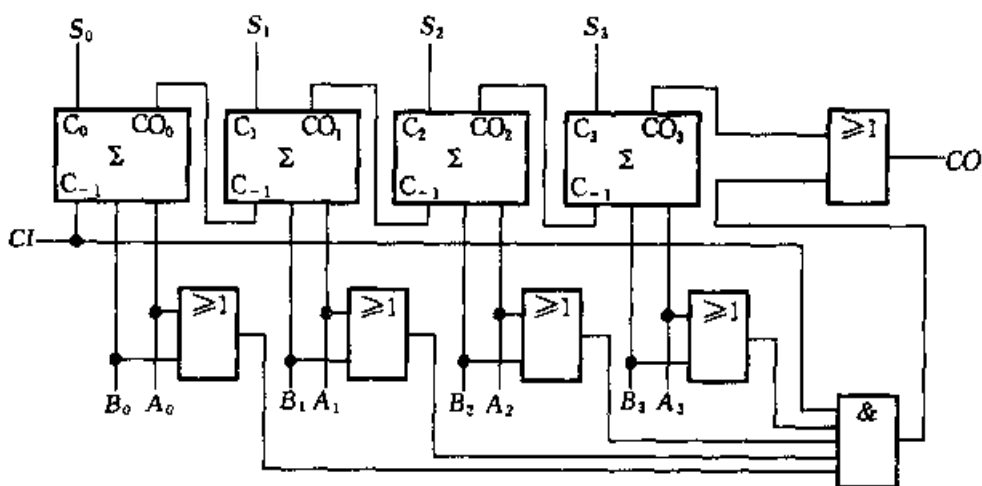
$$XY + XZ + YZ = \overline{\overline{XY} (X \oplus Y) Z}$$

根据上式,试用 9 个 2 输入与非门构成一个全加器。

4.5.3 4 位加法器如图 4.5.4 所示,全加器的逻辑电路如图 4.5.3 所示,假设每级门电路的延迟时间相等。 $C_{-1} = 0$ 时,且在 $t=0$ 的前一瞬间 $A = B = 0$ 。在 $t=0$ 时刻,使 $A = 1111$ 和 $B = 0001$ 。写出 $C_3 S_3 S_2 S_1 S_0$ 从 00000 到 10000 期间, $S_3 \sim S_0$ 和 $C_3 \sim C_0$ 的状态变化过程。

4.5.4 试用若干片 74182 构成一个十六位全超前进位产生器,画出逻辑示意图。

4.5.5 图题 4.5.5 是 4 位快速并行加法器。试分析其进位原理。



图题 4.5.5

4.5.6 仿照半加器和全加器的设计方法,试设计一半减器和一全减器,所用的门电路由自己选定。

* CAD 习题

C4.6.1 一编码器的真值表如表题 4.1.1 所示,试用 ABEL 语言中真值表的方法描述该编码器的逻辑电路,并用 ISP Synario 软件对所设计的电路进行分析,给出逻辑功能的仿真波形。

C4.6.2 译码器的功能如表题 4.2.5 所示。试用 ABEL 语言中真值表的方法描述该译码器的逻辑电路,并用 ISP Synario 软件对所设计的电路进行分析,给出逻辑功能的仿真波形。

C4.6.3 数据选择器如图题 4.3.1 所示,并行输入数据 $I_3 I_2 I_1 I_0 = 1010$,控制端 $X = 0$, $A_1 A_0$ 的态序为 00、01、10、11。试用 ISP Synario 软件对电路进行分析,并给出逻辑功能的仿真波形。

C4.6.4 试用 ABEL 语言设计一个能比较两个 4 位二进制数大小的比较器电路,并用 ISP Synario 软件对所设计的电路进行分析,给出逻辑功能的仿真波形。假设两个 4 位二进制

数为 $A(A_0 \cdots A_3)$ 和 $B(B_0 \cdots B_3)$, 比较的结果为 $A > B$ (用 GT 表示)、 $A < B$ (用 LT 表示) 和 $A = B$ (用 EQ 表示)。

C4.6.5 试用 ABEL 语言设计一个 4 位二进制数的并行加法器电路, 并用 ISP Synario 软件对所设计的电路进行分析, 给出逻辑功能的仿真波形。假设加数为 $A_0 \cdots A_3$, 被加数为 $B_0 \cdots B_3$, 和数为 $S_0 \cdots S_3$, 进位输出信号为 CO , 从低位来的进位输入信号为 CI 。

5 触 发 器

引言 在数字系统中,除了能够进行逻辑运算和算术运算的组合逻辑电路外,还需要具有记忆功能的时序逻辑电路。构成时序逻辑电路的基本单元是触发器。触发器是能够存储1位二进制码的逻辑电路,它有两个互补输出端,其输出状态不仅与输入有关,而且还与原先的输出状态有关。触发器具有不同的逻辑功能,在电路结构和触发方式方面也有不同的种类。在分析触发器的功能时,一般可用功能表、特性方程和状态图来描述其逻辑功能。研究触发方式时,主要是分析其输入信号的加入与触发脉冲之间的时间关系。

5.1 触发器的电路结构与工作原理

触发器的电路结构分为基本RS触发器、同步RS触发器、主从触发器和边沿触发器。

5.1.1 基本RS触发器

1. 电路结构和工作原理

把两个与非门 G_1 、 G_2 的输入、输出端交叉连接,即可构成基本RS触发器,其逻辑电路如图5.1.1a所示,它有两个输入端 R 、 S 和两个输出端 Q 、 \overline{Q} 。根据与非门的逻辑关系,触发器的逻辑表达式为

$$Q = \overline{S\overline{Q}} \quad (5.1.1)$$

$$\overline{Q} = \overline{RQ} \quad (5.1.2)$$

根据输入信号 R 、 S 不同状态的组合,触发器的输出与输入之间的关系有4种情况,现分析如下:

(1) $R=1$ 、 $S=0$

由式(5.1.1)可知,当 $S=0$ 时,不论 \overline{Q} 为何种状态,都有 $Q=1$;再由式(5.1.2)可得 $\overline{Q}=0$ 。

(2) $R=0$ 、 $S=1$

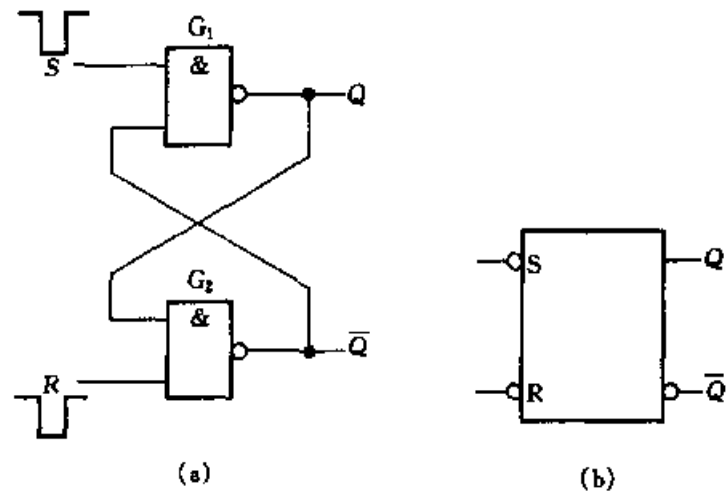


图 5.1.1 两与非门组成的基本 RS 触发器

(a) 逻辑图 (b) 逻辑符号

由于电路的对称性,这时 $Q=0$ 、 $\bar{Q}=1$ 。

如上所述,当触发器的两个输入端加入不同逻辑电平时,它的两个输出端 Q 和 \bar{Q} 有两种互补的稳定状态。一般规定触发器 Q 端的状态作为触发器的状态。通常称触发器处于某种状态,实际是指它的 Q 端的状态。 $Q=1$ 、 $\bar{Q}=0$ 时,称触发器处于1态,反之触发器处于0态。 $S=0$ 、 $R=1$ 使触发器置1,或称置位。因置位的决定性条件是 $S=0$,故称 S 端为置1端。 $R=0$ 、 $S=1$ 时,使触发器置0,或称复位。同理,称 R 端为置0端。

若触发器原来为1态,欲使之变为0态,必须令 R 端的电平由1变0, S 端的电平由0变1。这里所加的输入信号(低电平)称为触发信号,由它们导致的转换过程称为翻转。由于这里的触发信号是电平,因此这种触发器称为电平控制触发器。从功能方面看,它只能在 S 和 R 的作用下置1和置0,所以又称为置0置1触发器,或称为置位复位触发器。

基本 RS 触发器的逻辑符号如图 5.1.1b 所示。由于置1和置0都是低电平有效,因此在两输入端的边框外侧都画有小圆圈。

下面继续分析另外两种输入条件下的工作情况。

(3) $R=S=1$

当 R 、 S 都为1时,两个与非门 G_1 和 G_2 的状态由原来的 \bar{Q} 和 Q 的状态决定,不难推知,触发器维持原来状态不变。触发器保持状态时,输入端都加非有效电平(高电平),需要触发翻转时,按要求在某一输入端加一负脉冲,例如在 S 端加负脉冲使触发器置1,该脉冲信号回到高电平后,触发器仍维持1状态不变,相当于把 S 端某一时刻的电平信号存储起来,这体现了触发器具有记忆功能。

(4) $R=S=0$

显然,在此条件下,两个与非门的输出端 Q 和 \bar{Q} 全为1,在两个输入信号都同时撤去(回到1)后,触发器的状态将不能确定是1还是0,因此称这种情况为不定状态,这种情况应当避免。

综上所述,基本 RS 触发器的功能如表 5.1.1 所示。此外,还可以用或非门的输入、输出端交叉连接构成置0、置1触发器,其逻辑图和逻辑符号分别如图 5.1.2a和 5.1.2b 所示。这种触发器的触发信号是高电平有效,因此在逻辑符号方框外侧的输入端处没有小圆圈。

表 5.1.1 用两个与非门组成的基本 RS 触发器的功能表

R	S	Q
1	0	1
0	1	0
1	1	不变
0	0	不定

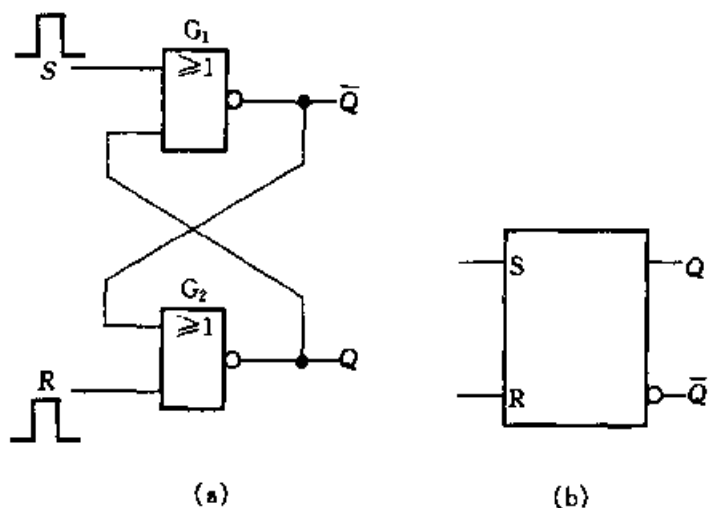


图 5.1.2 两或非门组成的基本 RS 触发器

(a) 逻辑图 (b) 逻辑符号

由图可得触发器的逻辑表达式为

$$Q = R + \bar{Q}$$

$$\bar{Q} = S + Q$$

根据上式可以分析出 R 、 S 为不同状态组合时触发器的状态,如表 5.1.2 所示。

表 5.1.2 用两个或非门组成的基本 RS 触发器的功能表

R	S	Q
0	1	1
1	0	0
0	0	不变
1	1	不定

2. 基本 RS 触发器的应用举例

例 5.1.1 用基本 RS 触发器和与非门构成 4 位二进制数码寄存器。

解： 在数字系统中，经常要用到可以存放数码的部件，这种部件称为数码寄存器。双稳态触发器就是一种具有记忆功能的单元电路，它能存储 1 位二进制码。如果要存放多位二进制码，可以用多个触发器完成。一个 4 位的数码寄存器逻辑图如图 5.1.3 所示。它由 4 个由与非门组成的基本 RS 触发器和 4 个与非门组成，其工作原理如下：

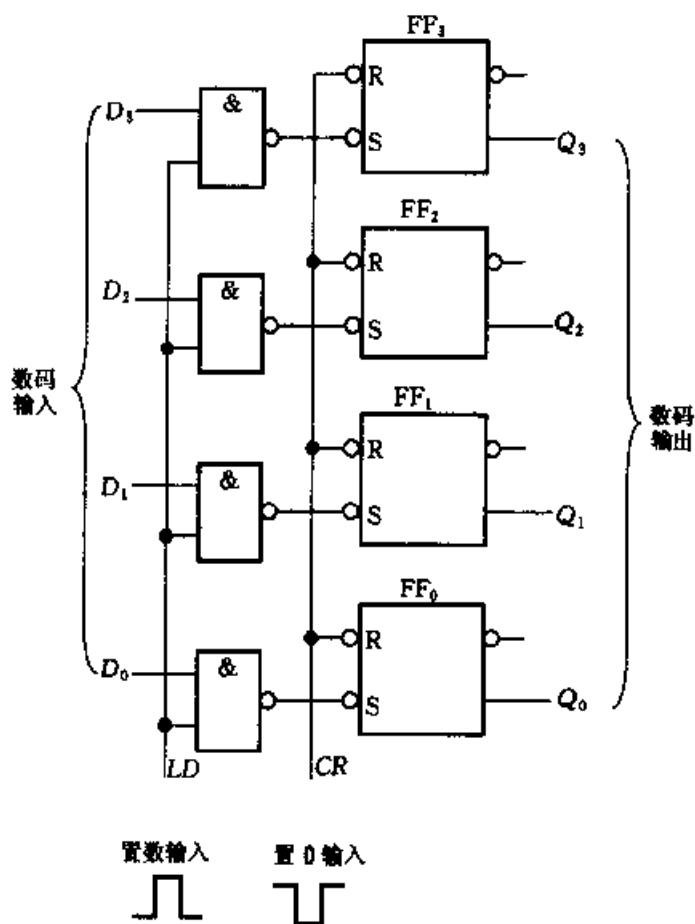


图 5.1.3 数码寄存器

数码寄存器有两个控制信号：清零指令 CR 和置数指令 LD ；4 个输入端 $D_0 \sim D_3$ ；4 个输出端 $Q_0 \sim Q_3$ 。清零是低电平有效，置数是高电平有效。

(1) 清零过程

清零时， CR 加低电平， LD 加低电平。这时 4 个与非门的输出均为高电平，即各触发器的 S 端为高电平，而 R 端均为低电平，使各触发器均为 0 态， CR 信号撤去（回到高电平）后， R 、 S 均为高电平，触发器转为不变状态。

(2) 置数过程

在清零之后， LD 端加有效电平（高电平）使各与非门打开， $D_0 \sim D_3$ 以反码

方式加入到对应触发器的 S 端, 根据触发器的功能可知, 各触发器的状态将与 $D_0 \sim D_3$ 的状态一致, 在 LD 信号撤去(回到低电平)后, 各触发器的 R 、 S 端均为 1, 又回到不变状态, 且 LD 将与非门封锁, 这就是置数过程。

还要指出的是, 置数必须在清零之后进行, 否则有可能出错。例如, 若 FF_0 原来的状态为 1, 现在要换成 0, 如果事先未置 0, 则由于 S 端为 1, 触发器处于不变状态, FF_0 不能翻转到 0。

例 5.1.2 运用基本 RS 触发器, 消除机械开关振动引起的脉冲。

解: 机械开关接通时, 由于振动会使电压或电流波形产生“毛刺”, 如图 5.1.4a 和 5.1.4b 所示。在电子电路中, 一般不允许出现这种现象, 因为这种干

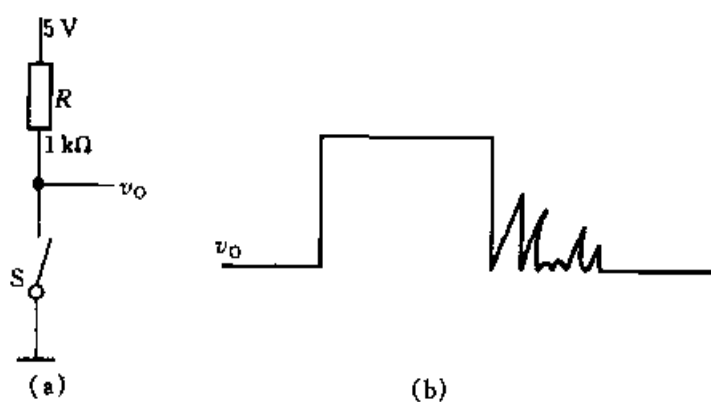


图 5.1.4 机械开关的工作情况

(a) 机械开关的接通 (b) 对电压波形的影响

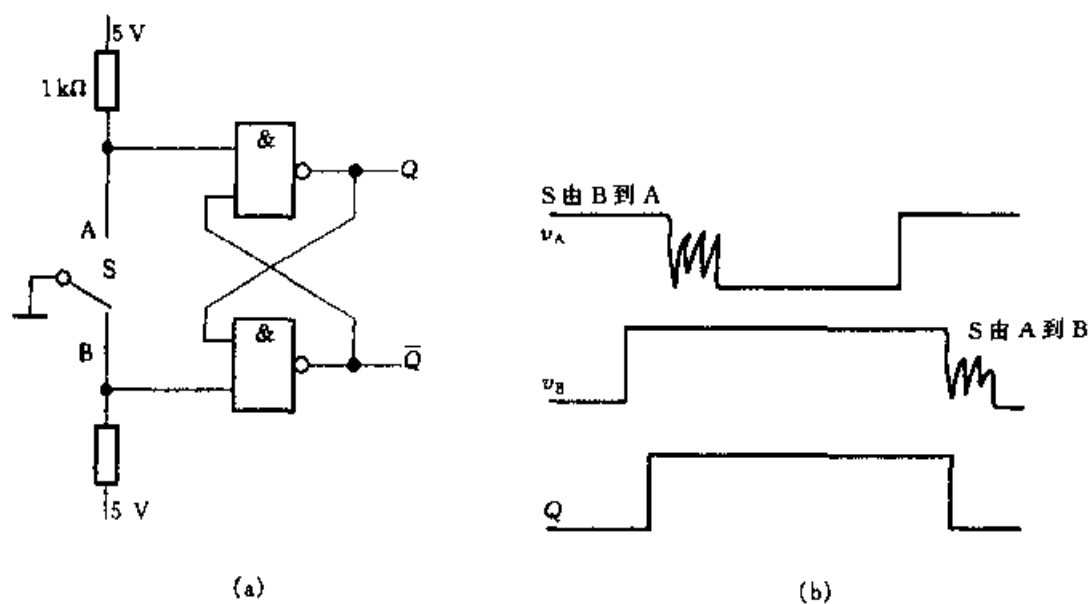


图 5.1.5 利用基本 RS 触发器消除机械开关振动的影响

(a) 电路 (b) 电压波形

扰信号会导致电路工作出错。

利用基本 RS 触发器的记忆作用可以消除上述开关振动所产生的影响,开关与触发器的连接方法如图 5.1.5a 所示。设单刀双掷开关原来与 B 点接通,这时触发器的状态为 0。当开关由 B 拨向 A 时,其中有一短暂的浮空时间,这时触发器的 R、S 均为 1, Q 仍为 0。中间触点与 A 接触时, A 点的电位由于振动而产生“毛刺”。但是,首先是 B 点已经为高电平, A 点一旦出现低电平,触发器的状态翻转为 1,即使 A 点再出现高电平,也不会再改变触发器的状态,所以 Q 端的电压波形不会出现“毛刺”现象,如图 5.1.5b 所示。

5.1.2 同步 RS 触发器

前面介绍的 RS 触发器的触发翻转过程直接由输入信号控制,而实际上,常常要求系统中的各触发器在规定的时刻按各自输入信号所决定的状态同步触发翻转,这个时刻可由外加时钟脉冲 CP^① 来决定。同步触发器如图 5.1.6 所示。

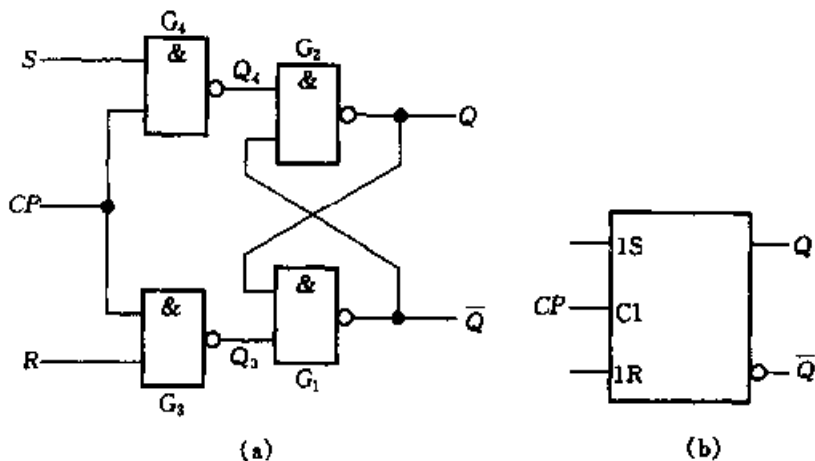


图 5.1.6 同步 RS 触发器

(a) 电路结构 (b) 逻辑符号

由图可知,输入信号要经过门 G_3 和 G_4 传递,这两个门同时受 CP 信号控制。当 CP 为 0 时, G_3 和 G_4 被封锁, R、S 不会影响触发器的状态,当 CP 为 1 时, G_3 和 G_4 打开,将 R、S 端的信号传送到基本 RS 触发器的输入端,触发器触发翻转。

我们可以先得出 Q_3 、 Q_4 和 R、S 的逻辑关系,然后再利用前面分析基本 RS 触发器的结果,列出同步 RS 触发器的功能表。考虑到 CP = 1 时,触发器可以翻转, G_3 和 G_4 的输出逻辑表达式为

$$Q_3 = \overline{R \cdot CP} = \overline{R} \quad (5.1.3)$$

① CP 是 Clock Pulse 的缩写。

$$Q_4 = \overline{S \cdot CP} = \overline{S} \quad (5.1.4)$$

将式(5.1.4)和式(5.1.3)分别代入式(5.1.1)和式(5.1.2)得

$$Q = \overline{Q_4 \cdot \overline{Q}} = \overline{\overline{S} \overline{Q}} \quad (5.1.5)$$

$$\overline{Q} = \overline{Q_3 \cdot \overline{Q}} = \overline{\overline{R} \overline{Q}} \quad (5.1.6)$$

式(5.1.5)和式(5.1.6)中等号左边的 Q 和等号右边 Q 的含义是不同的, 右边的 Q 表示每个 CP 作用前(高电平来到前)触发器的状态, 左边的 Q 则表示 CP 作用后触发器新的状态, 为了区分, 前者用 Q^n 表示, 称为触发器的现态, 后者用 Q^{n+1} 表示, 称为触发器的次态。故式(5.1.5)和式(5.1.6)可改写为

$$Q^{n+1} = \overline{\overline{S} \overline{Q}^n} \quad (5.1.7)$$

$$\overline{Q}^{n+1} = \overline{\overline{R} \overline{Q}^n} \quad (5.1.8)$$

由式(5.1.7)和(5.1.8)可以看出, 当 $R = S = 1$ 时, 触发器为不定状态, 应当避免出现这种情况。

根据以上分析, 利用表 5.1.1 的逻辑关系, 可得同步 RS 触发器的功能表, 如表 5.1.3 所示。

表 5.1.3 同步 RS 触发器的功能表

S	R	Q^n	Q^{n+1}	说 明
0	0	0	0	输出状态不变
		1	1	
0	1	0	0	输出状态与 S 的状态相同
		1	0	
1	0	0	1	输出状态与 S 的状态相同
		1	1	
1	1	0	—	输出状态不定
		1	—	

根据功能表, 同步 RS 触发器的逻辑功能可用如下表达式表示:

$$\begin{cases} Q^{n+1} = S + \overline{R}Q^n \\ SR = 0 \text{ (约束条件)} \end{cases} \quad (5.1.9)$$

式(5.1.9)称为触发器的特性方程。

例 5.1.3 图 5.1.6 所示同步 RS 触发器的 CP 、 S 、 R 的波形如图 5.1.7 所示, 触发器的原始状态为 $Q = 0, \overline{Q} = 1$, 试画出 Q_3 、 Q_4 、 Q 和 \overline{Q} 的波形。

解: 根据题意, 画出 Q_3 、 Q_4 、 Q 和 \overline{Q} 的波形示于同一图中。

在 CP 高电平时触发器翻转。根据式(5.1.3)和式(5.1.4)以及表 5.1.3 的逻辑关系即可画出 Q_3 、 Q_4 、 Q 和 \overline{Q} 的波形, 如图 5.1.7 所示。与基本 RS 触发器相比, 对触发翻转增加了时间控制。由图 5.1.6 可知, G_3 和 G_4 的输出 Q_3 和

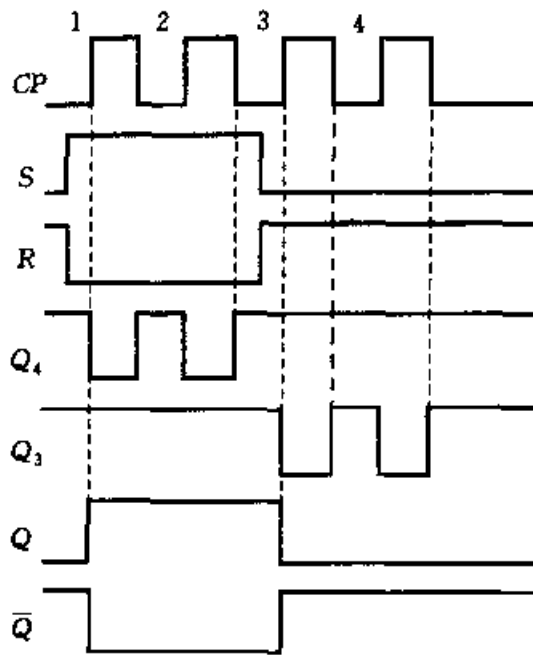


图 5.1.7 例 5.1.3 图

Q_4 直接影响触发器的状态,而在 CP 为 1 的时间间隔内,因为门 G_3 和 G_4 处于开启状态, R 、 S 的状态变化就会引起触发器状态的变化。因此,这种触发器的触发翻转只是被控制在一个时间间隔内,而不是控制在某一时刻进行。这种工作方式的触发器在应用中受到一定限制。下面介绍触发器翻转能控制在某一时刻(时钟脉冲的上升沿或下降沿)进行的触发器。

5.1.3 主从触发器

主从触发器由两级触发器构成,其中一级接收输入信号,其状态直接由输入信号决定,称为主触发器,还有一级的输入与主触发器的输出连接,其状态由主触发器的状态决定,称为从触发器。

1. 由两个同步 RS 触发器组成的主从触发器

由两个同步 RS 触发器组成的主从 RS 触发器的逻辑图和逻辑符号分别如图 5.1.8a 和 5.1.8b 所示,它由主触发器和从触发器两部分组成,它们都是同步 RS 触发器。反相器使这两个触发器加上互补时钟脉冲。下面分析其工作原理。

当 CP 为 1 时,主触发器的输入门 G_7 和 G_8 打开,主触发器根据 R 、 S 的状态触发翻转;而对于从触发器, CP 经 G_9 反相后加于它的输入门为逻辑 0 电平, G_3 和 G_4 封锁,其状态不受主触发器输出的影响,或者说这时保持状态不变。

CP 由 1 变 0 后,情况则相反, G_7 和 G_8 被封锁,输入信号 R 、 S 不影响主触发器的状态;而这时从触发器的 G_3 和 G_4 则打开,从触发器可以触发翻转。

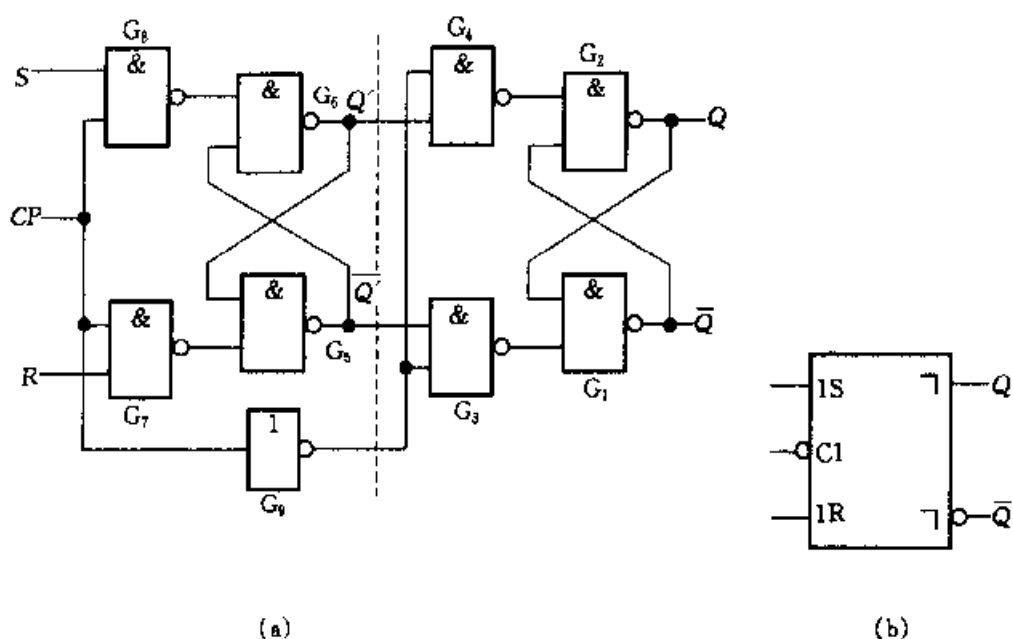


图 5.1.8 由两个同步 RS 触发器组成的主从 RS 触发器
(a) 逻辑图 (b) 逻辑符号

从触发器的翻转是在 CP 由 1 变 0 时刻 (CP 的下降沿) 发生的, CP 一旦达到 0 电平后, 主触发器被封锁, 其状态不受 R 、 S 的影响, 故从触发器的状态也不可能再改变, 即它只在 CP 由 1 变 0 时刻触发翻转。

由上分析可知, 主从触发器具有如下特点:

(1) 由两个同步 RS 触发器即主触发器和从触发器组成, 它们受互补时钟脉冲控制。

(2) 只在时钟脉冲的跳变沿 (本例为负跳沿) 触发翻转。

(3) 对于负跳沿触发的触发器, 输入信号在 CP 正跳沿前加入, 为主触发器触发翻转做好准备, 而 CP 正跳沿后的高电平要有一定的延迟时间, 以确保主触发器达到新的稳定状态; CP 负跳沿使触发器发生翻转后, CP 的低电平也必须有一定的延迟时间, 以确保从触发器达到新的稳定状态。这就是主从触发器对输入信号和时钟脉冲的要求, 常称为触发器的脉冲工作特性。有关触发器的脉冲工作特性还将在 5.3 节介绍。

2. 由传输门组成的 CMOS 主从触发器

CMOS 主从 D 触发器的逻辑图和逻辑符号分别如图 5.1.9a 和图 5.1.9b 所示。由图可知, D 触发器只有一个输入端, 它由两部分组成, 虚线左边为主触发器, 虚线右边为从触发器。主触发器和从触发器都是由传输门 (TG) 和反相器 (G) 经交叉连接构成的双稳态电路。由 TG_1 、 TG_2 、 G_1 和 G_2 组成主触发器, 由 TG_3 、 TG_4 、 G_3 和 G_4 组成从触发器。其中 CP 和 \overline{CP} 为互补时钟脉冲。

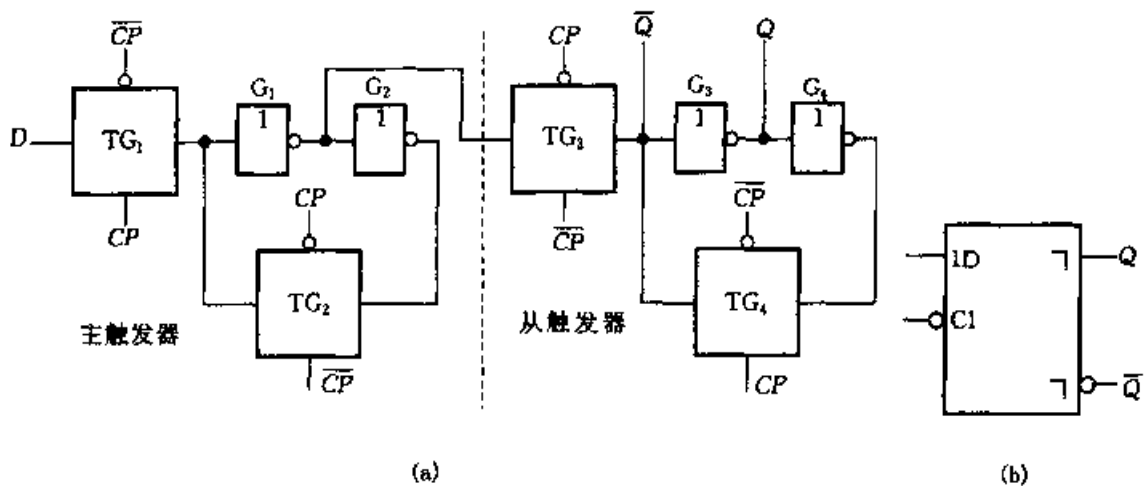


图 5.1.9 CMOS D 触发器逻辑图和逻辑符号

(a) 逻辑图 (b) 逻辑符号

工作过程分以下两个节拍：

(1) CP 正跳变后, TG_1 导通, TG_2 截止, 输入信号 D 送入主触发器。例如, D 为 1 时, 经 TG_1 传到 G_1 的输入端, 使 $\overline{Q'} = 0, Q' = 1$ 。同时, TG_3 截止, TG_4 导通, 显然 G_3 的输入端和 G_4 的输出端经 TG_4 连通, 使从触发器维持在原来的状态不变。

(2) CP 负跳变后, TG_1 截止, TG_2 导通, 由此切断了 D 端与主触发器的联系, 且同时 TG_2 将 G_1 的输入端和 G_2 的输出端连通, 使主触发器维持原态不变。从触发器的情况是, TG_3 导通, TG_4 截止, 主触发器的状态送入从触发器。 $\overline{Q} = 0$ 经 TG_3 传给 G_3 的输入, 于是 $\overline{Q} = 0, Q = 1$ 。

如上所述, 图 5.1.9 所示触发器是由 CP 的负跳沿触发翻转。如果把所有传输门上的控制信号 CP 和 \overline{CP} 对换, 那么就改成正跳沿触发。

5.1.4 边沿触发器

负跳沿触发的主从触发器工作时, 必须在正跳沿前加入输入信号。如果在 CP 高电平期间输入端出现干扰信号, 那么就有可能使触发器的状态出错。而边沿触发器允许在 CP 触发沿来到前一瞬间加入输入信号。这样, 输入端受干扰的时间大大缩短, 受干扰的可能性也就降低了。下面介绍两种不同工作方式的边沿触发器。

1. 维持-阻塞边沿触发器

维持-阻塞式边沿 D 触发器的逻辑图和逻辑符号分别如图 5.1.10a 和图 5.1.10b 所示。该触发器由 6 个与非门组成, 其中 G_1 和 G_2 构成基本 RS 触发器, 下面分析其工作原理。

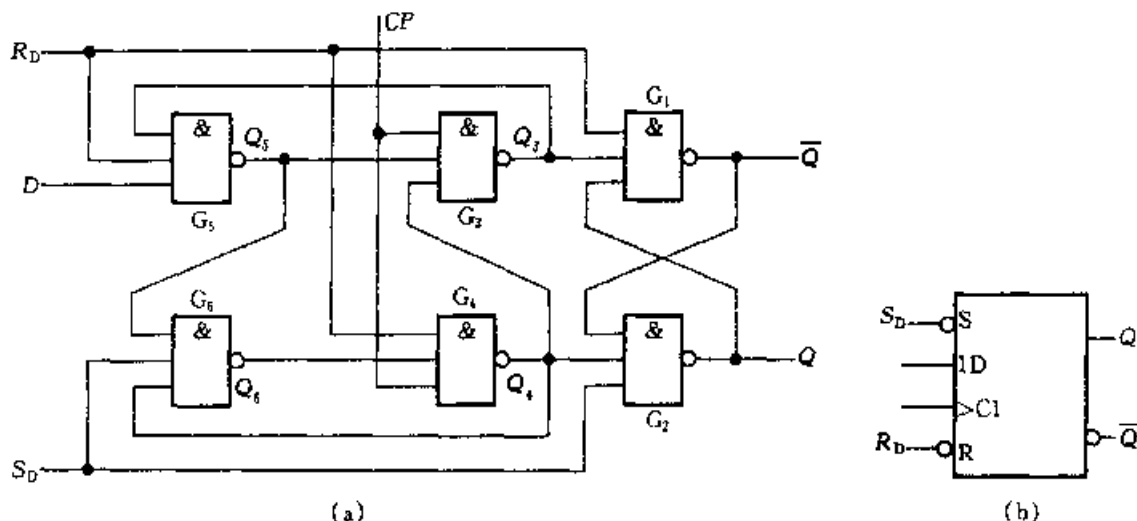


图 5.1.10 边沿 D 触发器的逻辑图和逻辑符号

(a) 逻辑图 (b) 逻辑符号

S_D 和 R_D 接至基本 RS 触发器的输入端, 它们分别是预置和清零端, 低电平有效。当 $S_D = 0$ 且 $R_D = 1$ 时, 不论输入端 D 为何种状态, 都会使 $Q = 1$, $\bar{Q} = 0$, 即触发器置 1; 当 $S_D = 1$ 且 $R_D = 0$ 时, 触发器的状态为 0, S_D 和 R_D 通常又称为直接置 1 和置 0 端。分析工作原理时, 设它们均已加入了高电平, 不影响电路的工作。工作过程如下:

(1) $CP = 0$ 时, 与非门 G_3 和 G_4 封锁, 其输出 $Q_3 = Q_4 = 1$, 触发器的状态不变。同时, 由于 Q_3 至 G_5 和 Q_4 至 G_6 的反馈信号将这两个门打开, 因此可接收输入信号 D , $Q_5 = \bar{D}$, $Q_6 = \bar{Q}_5 = D$ 。

(2) 当 CP 由 0 变 1 时触发器翻转。这时 G_3 和 G_4 打开, 它们的输出 Q_3 和 Q_4 的状态由 G_5 和 G_6 的输出状态决定。 $Q_3 = \bar{Q}_5 = D$, $Q_4 = \bar{Q}_6 = \bar{D}$ 。由基本 RS 触发器的逻辑功能可知, $Q = D$ 。

(3) 触发器翻转后, 在 $CP = 1$ 时输入信号被封锁。 G_5 和 G_6 打开后, 它们的输出 Q_3 和 Q_4 的状态是互补的, 即必定有一个是 0, 若 Q_3 为 0, 则经 G_3 输出至 G_5 输入的反馈线将 G_5 封锁, 即封锁了 D 通往基本 RS 触发器的路径; 该反馈线起到了使触发器维持在 0 状态和阻止触发器变为 1 状态的作用, 故该反馈线称为置 0 维持线, 置 1 阻塞线。 Q_4 为 0 时, 将 G_3 和 G_6 封锁, D 端通往基本 RS 触发器的路径也被封锁。 Q_4 输出端至 G_6 反馈线起到使触发器维持在 1 状态的作用, 称作置 1 维持线; Q_4 输出至 G_3 输入的反馈线起到阻止触发器置 0 的作用, 称为置 0 阻塞线。因此, 该触发器常称为维持-阻塞触发器。

总之, 该触发器是在 CP 正跳沿前接受输入信号, 正跳沿时触发翻转, 正跳沿后输入即被封锁, 三步都是在正跳沿前后完成, 所以有边沿触发器之称。

与主从触发器相比,同类工艺的边沿触发器有更强的抗干扰能力和更高的工作速度。

2. 利用传输延迟的边沿触发器

利用传输延迟的负跳变触发的边沿 JK 触发器的逻辑图和逻辑符号分别如图 5.1.11a 和图 5.1.11b 所示。其工作原理如下:

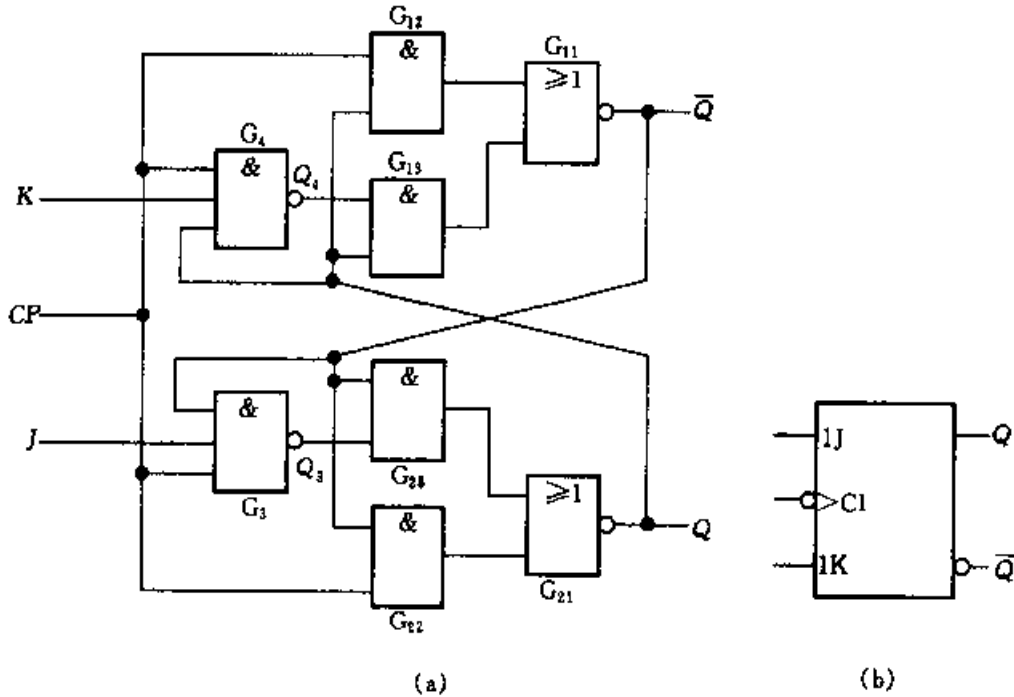


图 5.1.11 利用传输延迟的边沿 JK 触发器逻辑图和逻辑符号

(a) 逻辑图 (b) 逻辑符号

(1) $CP = 0$ 时, 触发器处于一个稳态

CP 为 0 时, G_3 、 G_4 被封锁, 不论 J 、 K 为何状态, Q_3 、 Q_4 均为 1, 另一方面, G_{12} 、 G_{22} 也被 CP 封锁, 因而由与非门组成的触发器处于一个稳定状态, 使输出 Q 、 \bar{Q} 状态不变。

(2) CP 由 0 变 1 时, 触发器不翻转, 为接收输入信号作准备

设触发器原状态为 $Q = 0$, $\bar{Q} = 1$ 。当 CP 由 0 变 1 时, 有两个信号通道影响触发器的输出状态, 一个是 G_{12} 和 G_{22} 打开, 直接影响触发器的输出, 另一个是 G_4 和 G_3 打开, 再经 G_{13} 和 G_{23} 影响触发器的状态。前一个通道只经一级与门, 而另一个通道则要经一级与非门和一级与门, 显然 CP 的跳变经前者影响输出比经后者要快得多。在 CP 由 0 变 1 时, G_{22} 的输出首先由 0 变 1, 这时无论 G_{23} 为何种状态 (即无论 J 、 K 为何状态), 都使 Q 仍为 0。由于 Q 同时连接 G_{12} 和 G_{13} 的输入端, 因此它们的输出均为 0, 使 G_{11} 的输出 $\bar{Q} = 1$, 触发器的状态不变。 CP 由 0 变 1 后, 打开 G_3 和 G_4 , 为接收输入信号 J 、 K 作好了准备。

(3) CP 由1变0时触发器翻转

设输入信号 $J=1, K=0$, 则 $Q_3=0, Q_4=1, G_{13}$ 和 G_{23} 的输出均为0。当 CP 下降沿到来时, G_{22} 的输出由1变0, 则有 $Q=1$, 使 G_{13} 输出为1, $\overline{Q}=0$, 触发器翻转。

虽然 CP 变0后, G_3, G_4, G_{12} 和 G_{22} 封锁, $Q_3=Q_4=1$, 但由于与非门的延迟时间比与门长(在制造工艺上予以保证), 因此 Q_3 和 Q_4 这一新状态的稳定是在触发器翻转之后。由此可知, 该触发器在 CP 下降沿触发翻转, CP 一旦到0电平, 则将触发器封锁, 处于(1)所分析的情况。

复习思考题

- 5.1.1 按电路结构和触发方式分类, 触发器可分为哪几类?
- 5.1.2 如果基本 RS 触发器由两个与非门组成, R、S 输入端加上什么电平时触发器出现不定状态?
- 5.1.3 负跳沿触发的主从触发器, 输入信号应该是在什么时刻(相对 CP)前加入?
- 5.1.4 边沿触发器与主从触发器比较, 具有什么主要优点?
- 5.1.5 边沿触发器输入信号的加入, 在时间(相对 CP)上有什么要求?

5.2 触发器的功能

触发器具有不同的功能, 通常可用真值表、状态方程和状态转换图表示。

5.2.1 RS 触发器

1. 功能分析

可以利用图 5.1.8 分析 RS 触发器的功能。由图可知, 其中从触发器的两个输入信号来自主触发器的输出, 显然它们是一对互补信号, 对照图 5.1.6 可知, 图 5.1.8 中的 Q' 对应 S, $\overline{Q'}$ 对应 R, 所以 $Q^{n+1} = Q'^{n+1} + \overline{Q'^{n+1}} Q^n = Q'^{n+1} + S + \overline{R} Q^n$, 既然 $Q^{n+1} = Q'^{n+1}$, 所以也有 $Q^n = Q'^n$, 故

$$Q^{n+1} = S + \overline{R} Q^n \quad (5.2.1)$$

再加上约束条件 $SR=0$, 就与式(5.1.9)一致, 所以, 主从 RS 触发器的逻辑功能与前面介绍的同步 RS 触发器的逻辑功能一致, 因此其功能表也与表 5.1.3 一致。

触发器的功能还可以用状态转换图表示, RS 触发器的状态转换图如图 5.2.1 所示。如图所示, 两个圆圈内标以 1 和 0, 表示触发器的两个状态, 带箭头的弧线表示状态转换的方向, 箭头指向触发器次态, 箭尾为触发器现态, 弧线旁边标出了状态转换的条件。

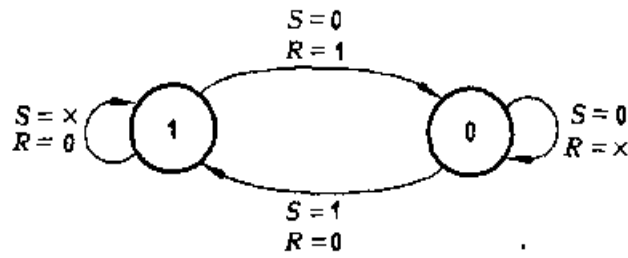


图 5.2.1 RS 触发器的状态转换图

2. 集成触发器

TTL 集成主从 RS 触发器 74L71 的逻辑符号和引脚分布分别如图 5.2.2a 和 5.2.2b 所示。该触发器分别有 3 个 S 端和 3 个 R 端,分别为与逻辑关系,即 $1R = R_1 \cdot R_2 \cdot R_3, 1S = S_1 \cdot S_2 \cdot S_3$ 。使用中如有多余的输入端,要将它们接至高电平。触发器带有清零端(置 0) R_D 和预置端(置 1) S_D ,它们的有效电平均为低电平。

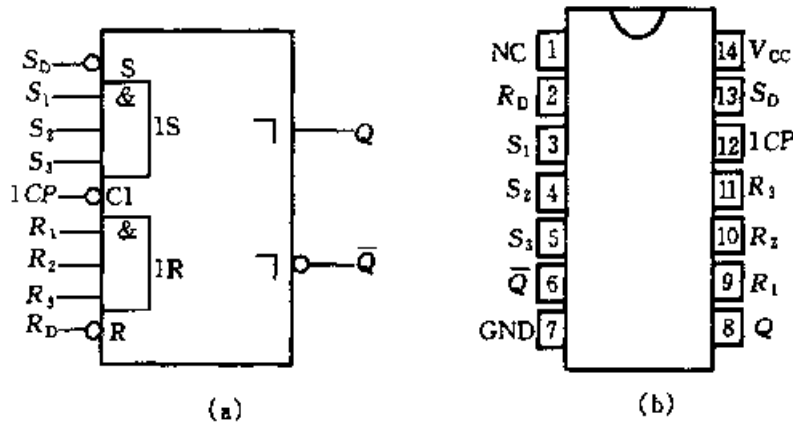


图 5.2.2 TTL 主从 RS 触发器

(a) 逻辑符号 (b) 引脚分布图

CT74LS71 的功能如表 5.2.1 所示,由表可将它的逻辑功能概括如下:

表 5.2.1 主从 RS 触发器 74LS71 功能表

输 入					输 出	
预置 S_D	清零 R_D	时钟 CP	1S	1R	Q	\bar{Q}
L	H	x	x	x	H	L
H	L	x	x	x	L	H
H	H	┌	L	L	Q^n	\bar{Q}^n
H	H	└	H	L	H	L
H	H	└	L	H	L	H
H	H	└	H	H	不	定

(1) 具有预置、清零功能,预置端加低电平,清零端加高电平时,触发器置1,反之触发器置0。预置和清零与 CP 无关,这种方式称为**直接预置**和**直接清零**。

(2) 正常工作时,预置端和清零端必须都加高电平,且要输入时钟脉冲。

(3) 触发器的功能和表 5.1.3 所示的 RS 触发器的功能一致。

5.2.2 JK 触发器

1. 功能分析

主从 JK 触发器是在 RS 触发器的基础上稍加改动而产生的,它的逻辑图如图 5.2.3 所示。

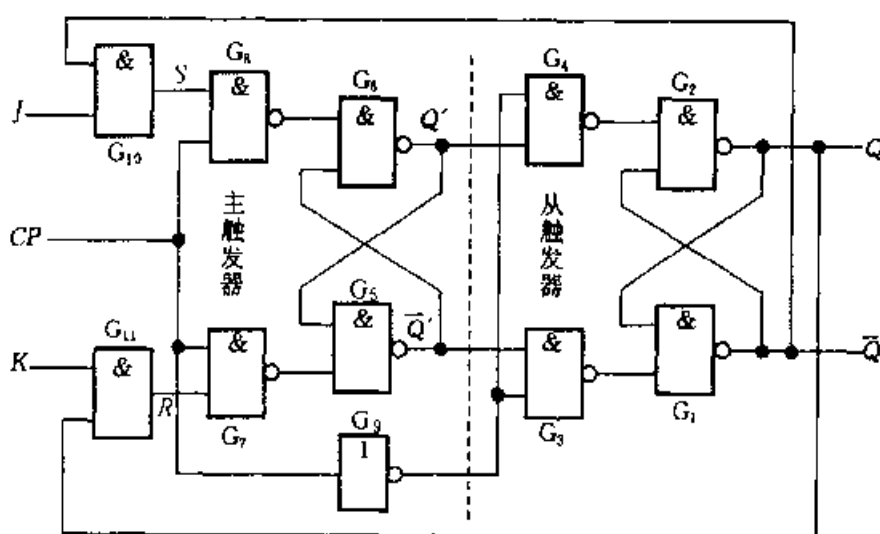


图 5.2.3 主从 JK 触发器的逻辑电路

由图可知,在 RS 触发器的 R 端和 S 端分别增加一个 2 输入端的与门 G_{11} 和 G_{10} ,将 \bar{Q} 端与原 S 端的与门 G_{10} 的一个输入端连接,另一输入端作为 J 端;将 Q 端与原 R 端的与门 G_{11} 的一个输入端连接,另一输入作为 K 端。由图 5.2.3 可得

$$S = J\bar{Q}$$

$$R = KQ$$

将上式代入式(5.2.1)得

$$\begin{aligned} Q^{n+1} &= J\bar{Q}^n + \bar{K}Q^nQ^n \\ &= J\bar{Q}^n + \bar{K}Q^n \end{aligned} \quad (5.2.2)$$

这是 JK 触发器的特性方程。

由式(5.2.2)可知,当 $J=1, K=0$ 时, $Q^{n+1}=1$; $J=0, K=1$ 时, $Q^{n+1}=0$; $J=K=1$ 时, $Q^{n+1}=\bar{Q}^n$ 。JK 触发器的功能如表 5.2.2 所示。

边沿 JK 触发器的功能与主从 JK 触发器的功能完全相同。

表 5.2.2 JK 触发器的功能表

J	K	Q^n	Q^{n+1}	说 明
0	0	0	0	输出状态不变
		1	1	
0	1	0	0	输出状态与 J 端状态相同
		1	0	
1	0	0	1	输出状态与 J 端状态相同
		1	1	
1	1	0	1	每输入一个脉冲输出状态改变一次
		1	0	

JK 触发器与 RS 触发器的不同之处是, 它没有约束条件。在 $J = K = 1$ 时, 每输入一个时钟脉冲后, 触发器翻转一次。触发器的这种工作状态称为计数状态。由触发器翻转的次数可以计算出输出时钟脉冲的个数。

JK 触发器的状态转换图如图 5.2.4 所示。

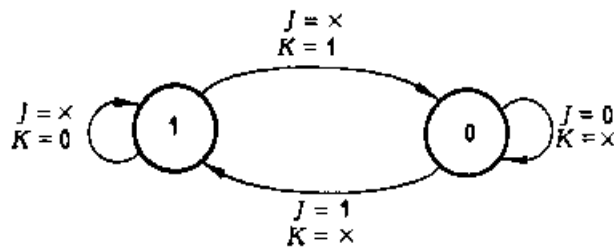


图 5.2.4 JK 触发器的状态转换图

例 5.2.1 设负跳沿触发的 JK 触发器的时钟脉冲和 J 、 K 信号的波形如图 5.2.5 所示, 画出输出端 Q 的波形。设触发器的初始状态为 0。

解: 根据式(5.2.2), 或表 5.2.2, 或图 5.2.4, 可画出 Q 端的波形, 如图 5.2.5 所示。

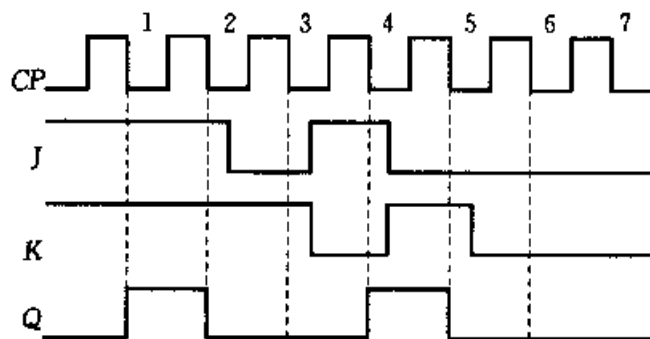


图 5.2.5 例 5.2.1 的波形图

从图 5.2.5 可以看出, 在第 1、2 个 CP 脉冲作用期间, J 、 K 均为 1, 每输入一个脉冲, Q 端的状态就改变一次, 这时 Q 端的方波频率是时钟脉冲频率的二分

之。若以 CP 端为输入, Q 端为输出, 则一个触发器就可作为二分频电路, 两个触发器串联就可获得四分频, 其余类推。

用画波形图的方法分析触发器的工作情况时, 必须注意几点:

- (1) 触发器的触发翻转发生在时钟脉冲的触发沿(这里是下跳沿)。
- (2) 判断主从触发器次态的依据是负跳沿前瞬间(触发沿为负跳沿时)输入端的状态。
- (3) 判断边沿触发器次态的依据是触发沿前瞬间输入端的状态。

例 5.2.2 负跳沿触发主从 JK 触发器的时钟信号 CP 和输入信号 J、K 的波形如图 5.2.6 所示, 信号 J 的波形图上用虚线标出了有一干扰信号, 画出考虑干扰信号影响的 Q 端的输出波形。设触发器的初始状态为 1。

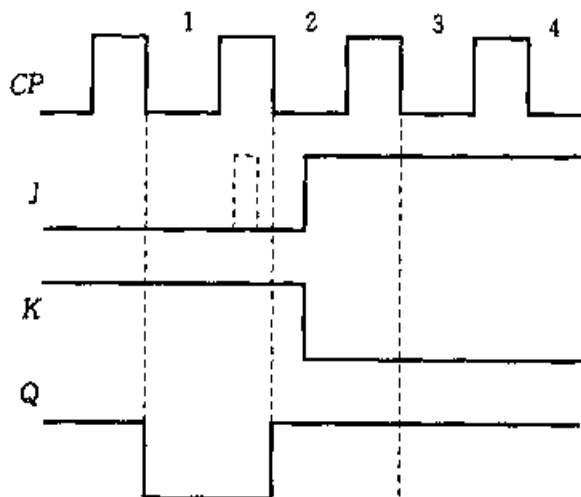


图 5.2.6 例 5.2.2 的波形图

解: (1) 第一个 CP 的正跳沿前, $J = 0, K = 1$, 因此负跳沿产生后触发器应翻转为 0。

(2) 第二个 CP 的高电平期间, 信号 J 有一个正跳变的干扰(如虚线所示)。利用图 5.2.3 分析干扰信号的影响。干扰信号出现前, 主触发器和从触发器的状态是 $Q' = 0, \overline{Q'} = 1$ 和 $Q = 0, \overline{Q} = 1$ 。干扰信号的出现会影响主触发器状态的变化, 具体情况是: G_{10} 的两个输入端都为 1, 其输出为 1, 使 G_8 输出变为 0, 又使 $Q' = 1, \overline{Q'} = 0$ 。由于干扰信号的产生使主触发器的状态由 0 变为 1。干扰信号消失后, 主触发器的状态是否能恢复到原来的状态呢? 由于 $\overline{Q'} = 0$, 已将 G_6 封锁, G_8 的输出变化不会影响 Q' 的状态, 也就是 J 端的干扰信号的消失不会使 Q' 恢复到 0。因此第二个 CP 的负跳沿到来后触发器的状态为 $Q = 1$ 。如果 J 端没有正跳变的干扰信号产生, 根据 $J = 0, K = 1$ 的条件, 触发器的正常状态应为 $Q = 0$ 。在上述条件下, 主触发器的状态只能根据输入信号改变一次, 这种现象称为一次变化现象。并非所有条件下都会出现一次变化现象。由于 JK 触发器电

路的对称性,不难理解,在触发器的状态为1时, $CP=1$ 期间信号 K 由0变1也会产生一次变化现象。也只有这两种条件下主从触发器会产生一次变化现象。

(3) 对应于第三、第四个 CP 的输入条件都是 $J=1, K=0$, 所以 $Q=1$ 。

2. 集成 JK 触发器

集成 JK 触发器的产品较多,以下介绍一种较典型的高速 CMOS 双 JK 触发器 HC76。该器件内含两个相同的 JK 触发器,它们都带有预置和清零输入,属于负跳沿触发的边沿触发器,其逻辑符号和引脚分布分别如图 5.2.7a 和图 5.2.7b 所示。如果在一片集成器件中有多个触发器,通常在符号前面(或后面)加上数字,以示不同触发器的输入、输出信号,比如 $C1$ 与 $1J, 1K$ 同属一个触发器。HC76 的逻辑功能如表 5.2.3 所示。'76 型号的产品种类较多,比如还有主从 TTL 的 7476、74H76、边沿 TTL 74LS76 等,它们的功能都一样,与表 5.2.3 基本一致,只是主从触发器与边沿触发器的触发方式不同。

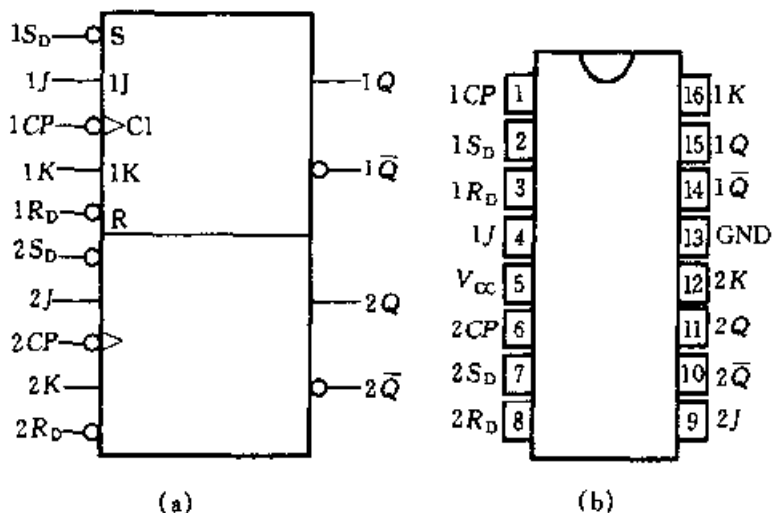


图 5.2.7 JK 触发器 HC76

(a) 逻辑符号 (b) 引脚分布

表 5.2.3 HC76 的逻辑功能表

输 入					输 出	
S_D	R_D	CP	J	K	Q	\bar{Q}
L	H	x	x	x	H	L
H	L	x	x	x	L	H
H	H	↓	L	L	Q^n	\bar{Q}^n
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	\bar{Q}^n	Q^n

5.2.3 T 触发器

对于式(5.2.2),若 $J = K = T$ 则变换为

$$Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n \quad (5.2.3)$$

这就是 T 触发器^①的特性方程。由式(5.2.3)得

$$Q^{n+1} = \begin{cases} \bar{Q}^n & T = 1 \\ Q^n & T = 0 \end{cases}$$

由此可知, T 触发器的功能是 T 为 1 时,为计数状态, T 为 0 时为保持状态。

表 5.2.4 T 触发器逻辑功能表

T	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

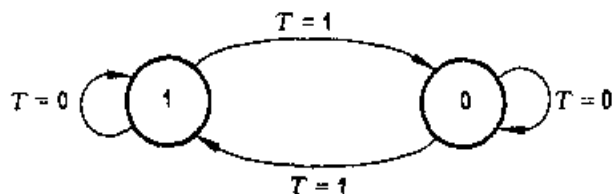


图 5.2.8 T 触发器的状态转换图

事实上,只要将 JK 触发器的 J、K 端连接在一起作为 T 端,就构成了 T 触发器,因此不必专门设计定型的 T 触发器产品。

5.2.4 D 触发器

1. 功能分析

在 5.1.4 节中已经分析了 D 触发器的工作原理,可得 D 触发器的特性方程为

$$Q^{n+1} = D \quad (5.2.4)$$

其逻辑功能表如表 5.2.5 所示,状态转换图如图 5.2.9 所示。

表 5.2.5 D 触发器的逻辑功能表

D	Q^n	Q^{n+1}	说明
0	0	0	输出状态与 D 端状态相同
0	1	0	
1	0	1	
1	1	1	

① T 系 Toggle 的字头。

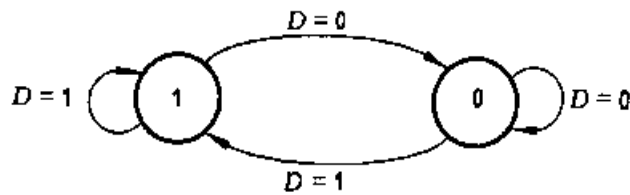


图 5.2.9 D 触发器的状态转换图

2. 集成 D 触发器

集成 D 触发器的定型产品种类比较多,这里介绍双 D 触发器 74HC74,实际上,74 型号的产品种类较多,比如还有 7474、74H74 等。由逻辑符号(图 5.2.10)和功能表(表 5.2.6)都可以看出,HC74 是带有预置、清零输入、上跳沿触发的边沿触发器。

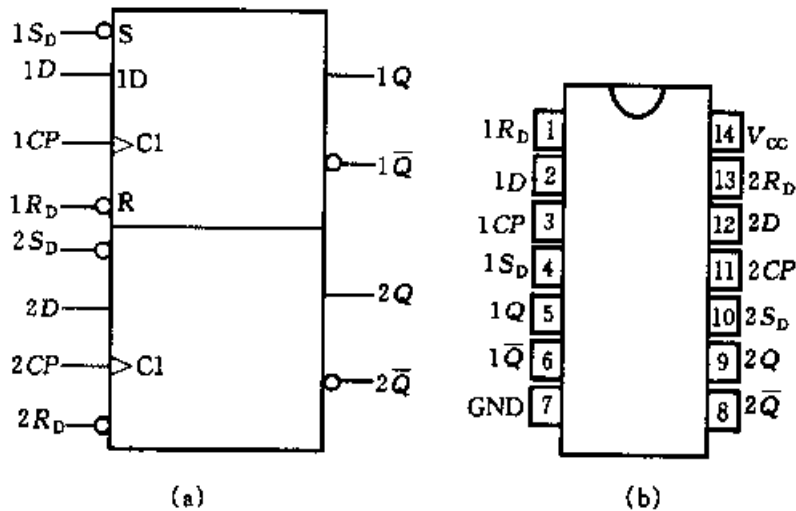


图 5.2.10 D 触发器 74HC74 的逻辑符号和引脚分布

(a) 逻辑符号 (b) 引脚分布

表 5.2.6 D 触发器 74HC74 功能表

输 入				输 出	
S_D	R_D	CP	D	Q	\bar{Q}
L	H	×	×	H	L
H	L	×	×	L	H
H	H	┌	H	H	L
H	H	└	L	L	H
H	H	L	×	Q_0	\bar{Q}_0

复习思考题

- 5.2.1 写出 RS 触发器、JK 触发器、T 触发器和 D 触发器的特性方程。
- 5.2.2 为什么 RS 触发器具有约束条件?
- 5.2.3 怎样利用 RS 触发器和逻辑门组成 JK 触发器?
- 5.2.4 怎样利用 JK 触发器组成 T 触发器?
- 5.2.5 怎样利用 JK 触发器和其他逻辑门组成 D 触发器?

5.3 触发器的脉冲工作特性及主要参数

5.3.1 集成触发器的脉冲工作特性

触发器对时钟脉冲、输入信号之间的时间关系的要求称为触发器的脉冲工作特性。掌握这种工作特性对触发器的应用非常重要。下面介绍几种触发器的脉冲工作特性。

1. JK 主从触发器的脉冲工作特性

如前所述,图 5.2.3 所示的 JK 主从触发器存在一次变化现象,因此 J、K 信号必须在 CP 正跳沿前加入,并且不允许在 CP=1 期间发生变化。为了工作可靠,CP 的 1 状态必须保持一段时间,直到主触发器的 Q' 和 \bar{Q} 端电平稳定,这段时间称为维持时间 t_{CPH} 。不难看出, t_{CPH} 应大于一级与门和三级与非门的传输延迟时间。

从 CP 负跳沿到触发器输出状态稳定,也需要一定的延迟时间 t_{CPL} 。我们把从时钟脉冲触发沿开始到一个输出端由 0 变 1 所需的延迟时间称为 t_{CPLH} , 把从 CP 触发沿开始到输出端由 1 变 0 的延迟时间称为 t_{CPLL} 。

为了使触发器可靠翻转,要求 $t_{CPL} > t_{CPLH}$ 。

综上分析,JK 主从触发器要求 CP 的最小工作周期 $T_{min} = t_{CPH} + t_{CPL}$ 。JK 主从触发器对 CP

和 J、K 信号的要求及触发翻转时间的示意图如图 5.3.1 所示。

2. JK 负边沿触发器的脉冲工作特性

以图 5.1.11 为例,来说明这类触发器的脉冲工作特性。如前所述,该触发器无一次变化现象,输入信号可在 CP 触发沿由 1 变 0 时刻前加入。由图 5.1.11 可知,该电路要求 J、K 信号先于 CP 信号触发沿传输到 G_3 、 G_4 的输出端,为此,

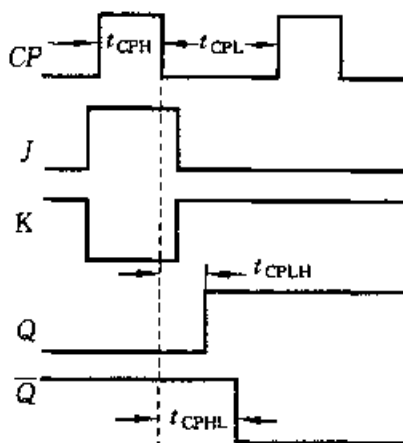


图 5.3.1 JK 主从触发器对 CP 和 J、K 信号的要求及触发器翻转时间示意图

它们的加入时间至少应比 CP 的触发沿提前一级与非门的延迟时间。这段时间称为建立时间 t_{set} 。

输入信号在负跳变触发沿来到后就不必保持,原因在于即使原来的 J 、 K 信号变化,还要经一级与非门的延迟才能传输到 G_3 和 G_4 的输出端,在此之前,触发器已由 G_{12} 、 G_{13} 、 G_{22} 、 G_{23} 的输出状态和触发器原先的状态决定翻转。所以这种触发器要求输入信号的维持时间极短,从而具有很高的抗干扰能力,且因缩短 t_{CH} 可提高工作速度。

从负跳变触发沿到触发器输出状态稳定,也需要一定的延迟时间 t_{CPL} 。显然,该延迟时间应大于两级与或非门的延迟时间。即 $t_{\text{CP1}} > 2.8t_{\text{pd}}$ 。

3. D 型正边沿维持-阻塞触发器的脉冲工作特性

由图 5.1.10 可知,在 CP 正跳变触发沿来到前,信号 D 必须经 G_5 和 G_6 传输到 G_3 和 G_4 的输入端,即这种触发器的建立时间为两级与非门的延迟时间, $t_{\text{set}} = 2t_{\text{pd}}$ 。

为了使触发器可靠翻转,信号 $D = 0$ 必须维持到 Q_3 由 1 变 0 后将 G_5 封锁为止,若在此之前 D 变为 1,则 Q_5 变为 0,将引起触发器误触发。在 CP 触发沿到来后,输入信号需要维持的时间称为保持时间 t_{H} 。在输入信号 $D = 0$ 时,保持时间 $t_{\text{H}} = 1t_{\text{pd}}$ 。 $D = 1$ 时,读者可自行分析,此时 t_{H} 为零。

为保证触发器可靠翻转, $CP = 1$ 的时间间隔 t_{CPH} 必须大于 t_{CPLH} 。显然,该触发器的 t_{CPLH} 为三级与非门 ($G_3 \rightarrow G_1 \rightarrow G_2$ 或 $G_4 \rightarrow G_2 \rightarrow G_1$) 延迟时间,即 $t_{\text{CPH}} > t_{\text{CPLH}} = 3t_{\text{pd}}$ 。对输入信号及触发脉冲 CP 的要求示意图如图 5.3.2 所示。

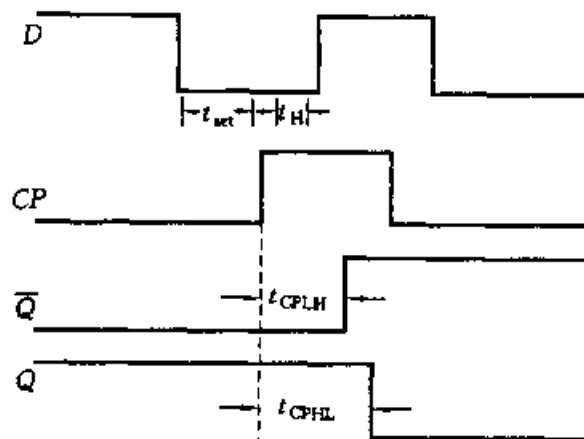


图 5.3.2 D 型正边沿维持-阻塞触发器对输入信号及触发脉冲的要求示意图

5.3.2 集成触发器的主要参数

和门电路一样,集成触发器的参数也可以分为直流参数和开关参数两大类,下面以 TTL 集成 JK 触发器为例分别予以简单介绍。

1. 直流参数

(1) 电源电流 I_{CC}

由于一个触发器由许多门构成,无论在0态或1态,总是一部分门处于饱和状态,另一部分处于截止状态,因此,电源电流的差别是不大的。但为明确起见,目前有些制造厂家规定,所有输入端和输出端悬空时电源向触发器提供的电流为电源电流 I_{CC} ,它表明该电路的空载功耗。

(2) 低电平输入电流(即输入短路电流) I_{IL}

某输入端接地,其他各输入、输出端悬空时,从该输入端流向地的电流为低电平输入电流 I_{IL} ,它表明对驱动电路输出为低电平时的加载情况。JK 触发器包括各 J、K 端,CP 端和直接置0、置1端的低电平输入电流。

(3) 高电平输入电流 I_{IH}

将各输入端(R_D 、 S_D 、J、K、CP 等)分别接 V_{CC} 时,测得的电流就是其高电平输入电流 I_{IH} ,它表明对驱动电路输出为高电平时的加载情况。

(4) 输出高电平 V_{OH} 和输出低电平 V_{OL}

Q 或 \bar{Q} 端输出高电平时的对地电压值为 V_{OH} ,输出低电平时的对地电压值为 V_{OL} 。

2. 开关参数

(1) 最高时钟频率 f_{max}

f_{max} 就是触发器在计数状态下能正常工作的最高工作频率,是表明触发器工作速度的一个指标。在测试 f_{max} 时,Q 和 \bar{Q} 端应带上额定的电流负载和电容负载,这在制造厂家的产品手册中均有明确规定。

(2) 对时钟信号的延迟时间(t_{CPLH} 和 t_{CPHL})

从时钟脉冲的触发沿到触发器输出端由0态变到1态的延迟时间为 t_{CPLH} ;从时钟脉冲的触发沿到触发器输出端由1态变到0态的延迟时间为 t_{CPHL} 。一般 t_{CPHL} 比 t_{CPLH} 约大一级门的延迟时间。它们表明对时钟脉冲 CP 的要求。

(3) 对直接置0(R_D)或置1(S_D)端的延迟时间(t_{RLH} 、 t_{RHL} 或 t_{SLH} 、 t_{SHL})

从置0脉冲触发沿到输出端由0变为1为 t_{RLH} ,到输出端由1变为0为 t_{RHL} ;从置1脉冲触发沿到输出端由0变1为 t_{SLH} ,到输出端由1变0为 t_{SHL} 。

CMOS 触发器的参数定义与以上介绍的参数基本一致,不再另作介绍。

复习思考题

- 5.3.1 什么是触发器的维持时间？
 5.3.2 什么是对 CP 所要求的最小工作周期？
 5.3.3 什么是建立时间？
 5.3.4 什么是保持时间？

* 5.4 CAD 例题

例 CE5.4.1 图 CE5.4.1a 是用 D 触发器模拟 JK 触发器的逻辑电路。试用 ISP Synario 软件对电路进行分析,并给出逻辑功能仿真波形图。

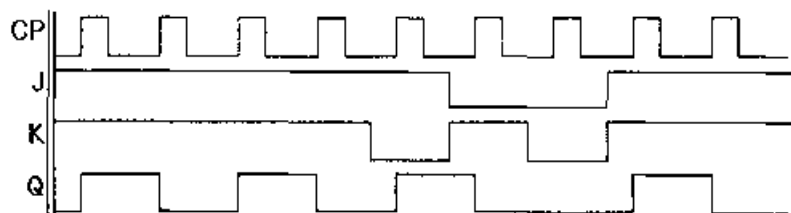
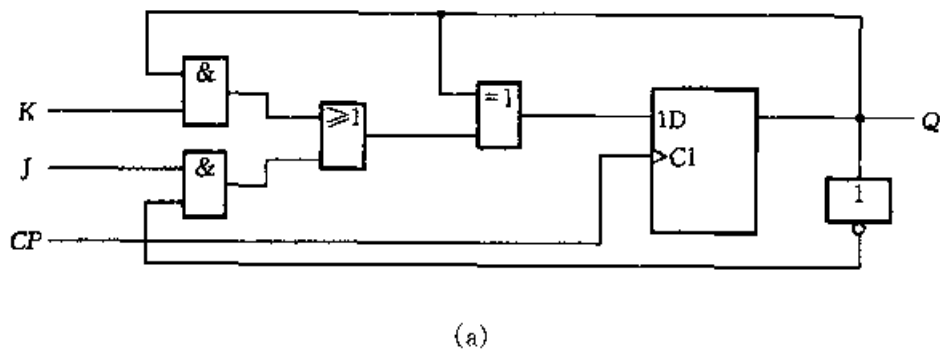


图 CE5.4.1 逻辑电路及其仿真波形图

(a) 逻辑电路 (b) 仿真波形图

解: (1) 输入设计文件

打开 ISP Synario 软件的原理图编辑器,输入图 CE5.4.1a 所示逻辑电路,并保存文件。然后对原理图文件进行编译和逻辑简化。

(2) 编写并输入测试向量文件

根据图 CE5.4.1a 所示逻辑电路,编写的测试向量文件如下:

```
module CE5_4_1           "模块开始
  J,K,CP    pin;        "定义输入信号
```

```

Q      pin istype 'com';      "定义输出信号
C,X=.C.,.X.;                "定义特殊常数
test _ vectors                "测试向量段开始

([CP,J,K]->Q)
@repeat 4{[C,1,1]->X;}      "将大括号内的内容重复4次
[C,1,0]->X;
[C,0,1]->X;
[C,0,0]->X;
@repeat 2{[C,1,1]->X;}
end

```

(3) 进行逻辑功能仿真

对测试向量文件进行编译和逻辑功能仿真,得到如图 CE5.4.1b 所示的逻辑功能仿真波形图。

例 CE5.4.2 试用 ABEL 语言设计一个 D 触发器、一个 JK 触发器和一个 T 触发器电路,并进行逻辑功能仿真。

解: (1) 输入设计文件

打开 ISP Synario 软件的文本编辑器,输入并保存下列 ABEL 语言的源文件(包括测试向量文件)。然后对源文件进行编译和逻辑简化。

```

module      CE5_4_2          "模块开始
CP,D,J,K,T   pin;          "定义输入信号
QD,QJK,QT    pin istype 'reg'; "定义输出信号
X,C=.X.,.C.; "定义特殊常数
equations    "逻辑方程段开始
QD.CLK=CP;   "D 触发器时钟方程
QJK.CLK=CP;  "JK 触发器时钟方程
QT.CLK=CP;   "T 触发器时钟方程
QD:=D;       "D 触发器特性方程
QJK:=J & !QJK # !K & QJK; "JK 触发器特性方程
QT:=(T & !QT) # (!T & QT); "T 触发器特性方程
test _ vectors "测试向量段开始
([CP,D,QD]->QD) "测试 D 触发器的逻辑功能
[C,0,0]->0;
[C,0,1]->0;
[C,1,0]->1;
[C,1,1]->1;

```

```

test _ vectors          "测试向量段开始
    ([CP,QT,T]->QT)    "测试 T 触发器的逻辑功能
    [C,0,0]->0;
    [C,0,1]->1;
    [C,1,0]->1;
    [C,1,1]->0;
test _ vectors          "测试向量段开始
    ([CP,QJK,J,K]->QJK) "测试 JK 触发器的逻辑功能
    [C,0,1,1]->1;
    [C,0,1,0]->1;
    [C,0,0,1]->0;
    [C,0,0,0]->0;
    [C,0,1,0]->1;
    [C,1,0,0]->1;
    [C,1,0,1]->0;
    [C,1,1,0]->1;
    [C,1,1,1]->0;
    [C,0,0,0]->0;
end

```

(2) 进行逻辑功能仿真

对测试向量文件进行编译和逻辑功能仿真,得到如图 CE5.4.2 所示的逻辑功能仿真波形图。

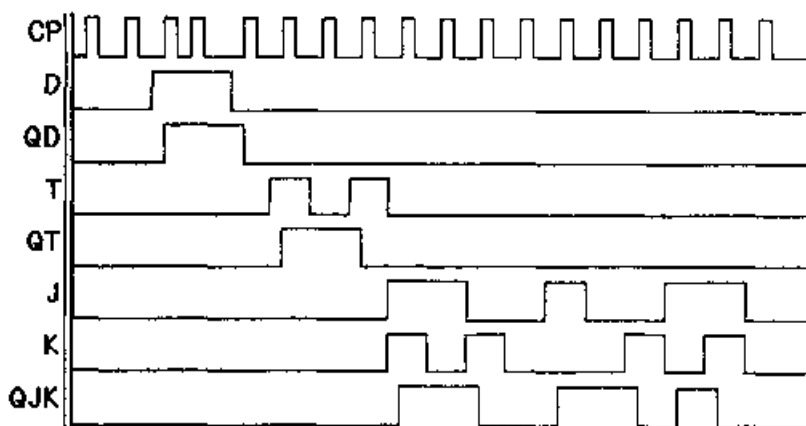


图 CE5.4.2 逻辑功能仿真波形图

小 结

• 触发器是具有记忆功能的逻辑电路,每个触发器能够存储 1 位二进制数据,是时序逻辑电路的基本单元。

• 按电路结构分类有基本 RS 触发器、同步触发器、主从触发器和边沿触发器。它们的触发翻转方式不同,基本 RS 触发器属于电平触发,同步触发器和主从触发器属于脉冲触发,边沿触发器是脉冲边沿触发,可以是正跳沿触发,也可以是负跳沿触发。主从触发器和边沿触发器的触发翻转虽然都发生在脉冲跳变时,但对加入输入信号的时间有所不同,对于主从触发器,如果是负跳变触发,输入信号必须在正跳变前加入,而边沿触发器可以在触发沿到来前(只要满足建立时间)加入。

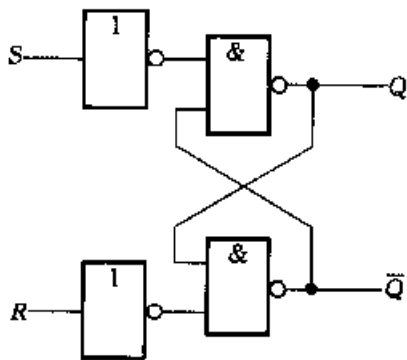
• 按功能分类有 RS 触发器、JK 触发器、T 触发器和 D 触发器。RS 触发器具有约束条件 $RS = 0$, T 触发器和 D 触发器的功能比较简单,JK 触发器的逻辑功能最为灵活,由它可以作 RS 触发器使用,也可以方便地转换成 T 触发器和 D 触发器。

• 电路结构和触发方式与功能没有必然的联系。比如 JK 触发器既有主从式的,也有边沿式的。主从式触发器和边沿触发器都有 RS、JK、D 功能触发器。

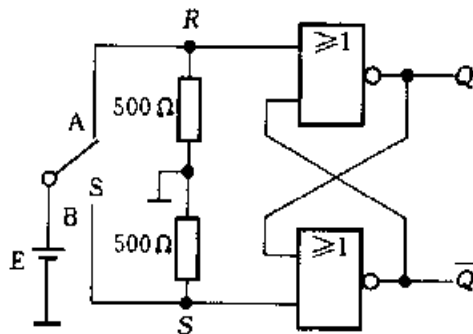
习 题

5.1.1 分析图题 5.1.1 所示电路的功能,列出功能表

5.1.2 用基本 RS 触发器消除手动开关因机械振动而产生的电压、电流波形毛刺的电路如图题 5.1.2 所示,试画出在按钮开关 S 由位置 A 到 B 有触点振动时的触发器 Q 、 \bar{Q} 端的波形。

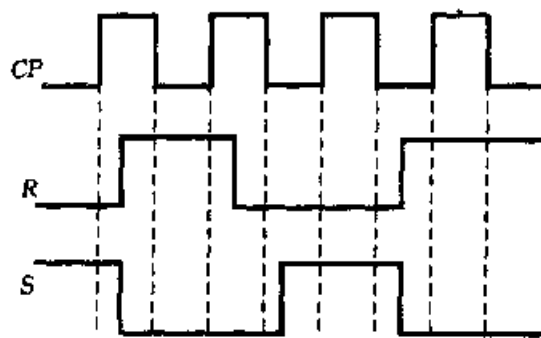


图题 5.1.1



图题 5.1.2

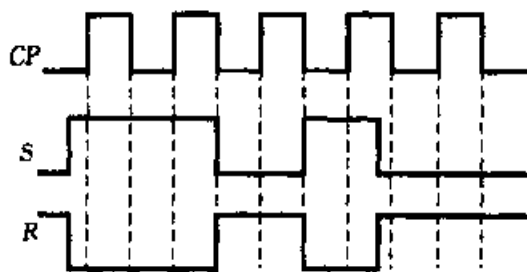
5.1.3 如图 5.1.6 所示触发器的 CP、R、S 信号波形如图题 5.1.3 所示,画出 Q 和 \bar{Q} 端的波形,设初态 $Q = 0$ 。



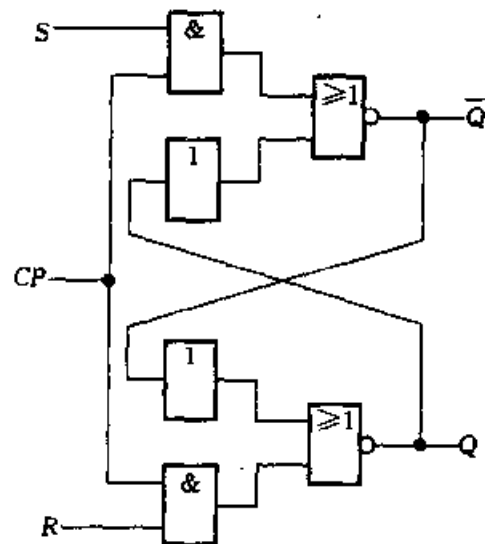
图题 5.1.3

5.1.4 同步 RS 触发器和基本 RS 触发器的主要区别是什么?

5.1.5 设图 5.1.6 所示电路的初始状态为 $Q=1$, R 、 S 端和 CP 端的输入信号如图题 5.1.5 所示, 试画出该同步 RS 触发器相应的 Q 和 \bar{Q} 端的波形。



图题 5.1.5



图题 5.1.6

5.1.6 由与或非门组成的同步 RS 触发器如图题 5.1.6 所示, 试分析其工作原理并列出功能表。

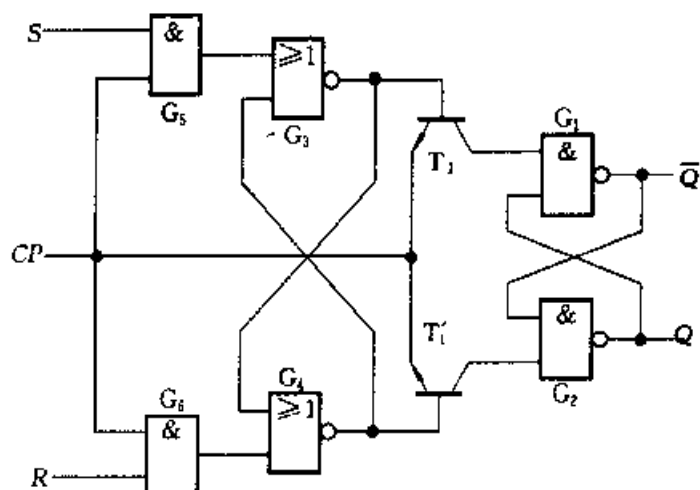
5.1.7 如果将图 5.1.8a 中的 G_5 、 G_6 改为或非门, 为了保持原来的输出、输入逻辑关系不变, 图中应作何修改?

5.1.8 触发器的逻辑电路如图题 5.1.8 所示。确定其为何种电路结构的触发器? 分析其工作原理, 列出功能表。

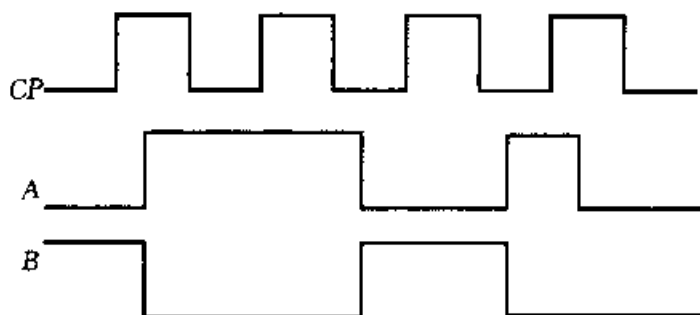
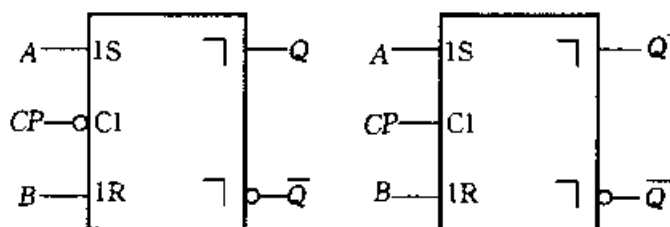
5.1.9 分析图 5.1.11a, 列出功能表。

5.1.10 归纳基本 RS 触发器、同步触发器、主从触发器和边沿触发器触发翻转的特点。

5.2.1 负跳沿触发和正跳沿触发两种触发方式的主从 RS 触发器的逻辑符号和 CP、S、R 的波形如图题 5.2.1 所示。分别画出它们的 Q 端的波形。设触发器的初始状态为 0。



图题 5.1.8



图题 5.2.1

5.2.2 设主从 JK 触发器的初始状态为 0, CP、J、K 信号如图题 5.2.2 所示,试画出触发器 Q 端的波形。

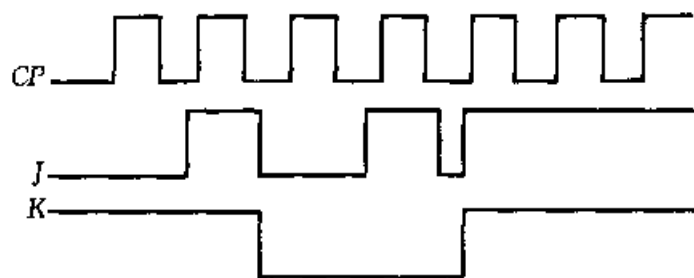
5.2.3 逻辑电路如图题 5.2.3 所示,试画出在 CP 作用下, ϕ_0 、 ϕ_1 、 ϕ_2 和 ϕ_3 的波形。

5.2.4 电路如图题 5.2.4 所示,设各触发器的初态为 0,画出在 CP 脉冲作用下 Q 端的波形。

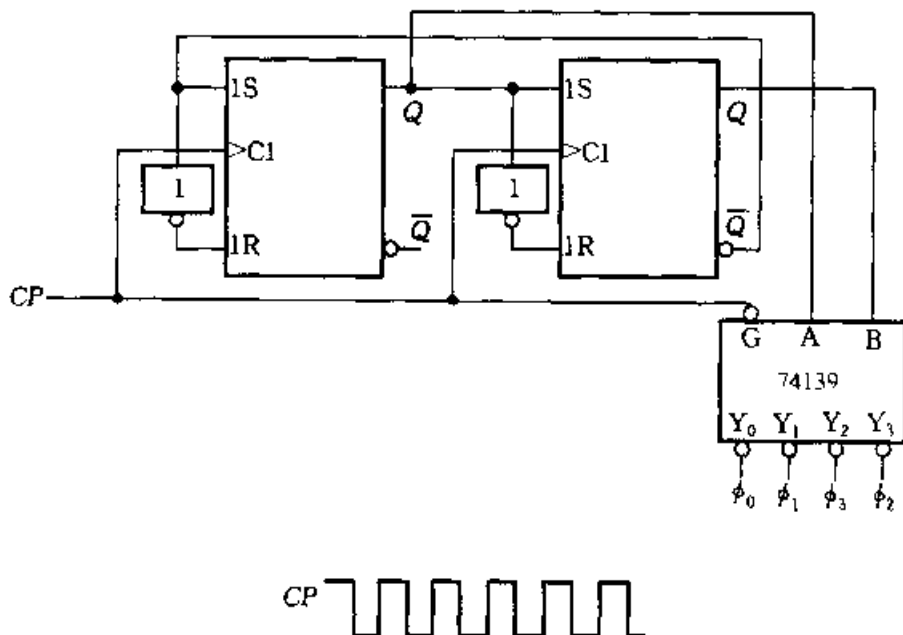
5.2.5 逻辑电路如图题 5.2.5 所示,已知 CP 和 X 的波形,试画出 Q_1 和 Q_2 的波形。触发器的初始状态均为 0。

5.2.6 逻辑电路如图题 5.2.6 所示,已知 CP 和 A 的波形,画出触发器 Q 端的波形,设触发器的初始状态为 0。

5.2.7 两相脉冲产生电路如图题 5.2.7 所示,试画出在 CP 作用下 ϕ_1 、 ϕ_2 的波形,并说



图题 5.2.2



图题 5.2.3

明 ϕ_1 、 ϕ_2 的相位差。各触发器的初始状态为 0。

5.2.8 3 种不同触发方式的 D 触发器的逻辑符号, 时钟 CP 和信号 D 的波形如图题 5.2.8 所示, 画出各触发器 Q 端的波形图。各触发器的初始状态为 0。

5.2.9 逻辑电路和各输入信号波形如图题 5.2.9 所示, 画出各触发器 Q 端的波形。各触发器的初始状态为 0。

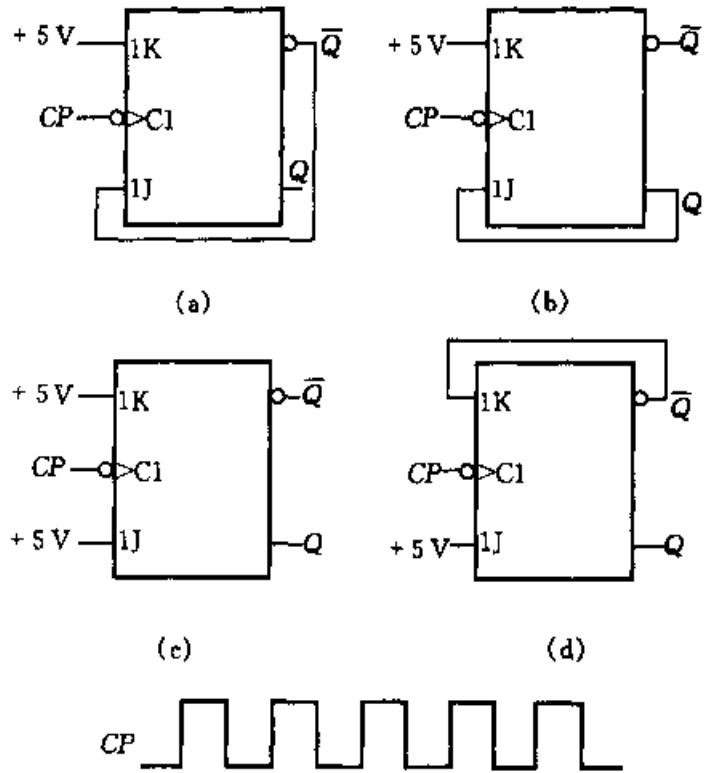
5.2.10 逻辑电路和输入信号波形如图题 5.2.10 所示, 画出各触发器 Q 端的波形。触发器的初始状态均为 0。

5.2.11 D 触发器逻辑符号如图题 5.2.11 所示, 用适当的逻辑门, 将 D 触发器转换成 T 触发器、RS 触发器和 JK 触发器。

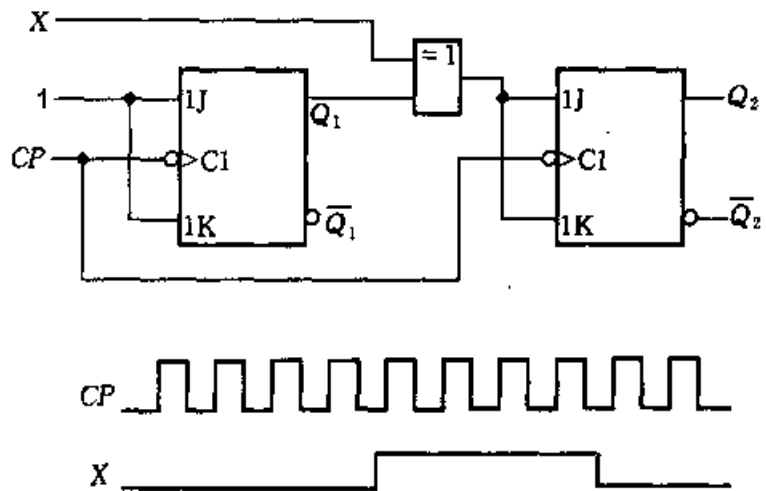
5.3.1 解释边沿触发器的工作速度高于主从触发器的原因。

5.3.2 分析 D 正跳沿触发的边沿触发器在 $D=1$ 时其保持时间为零的原因。

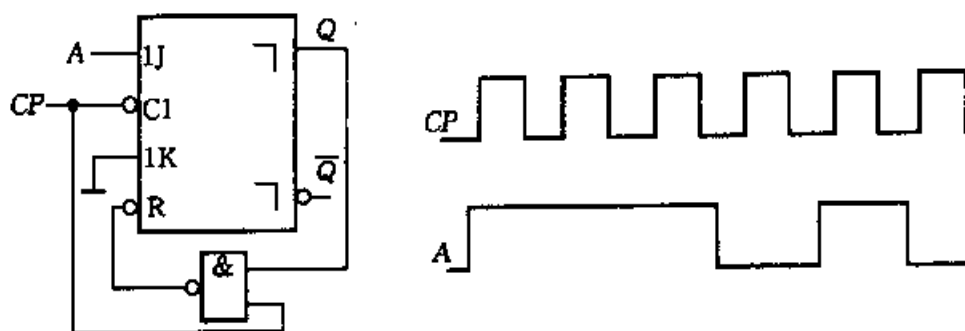
5.3.3 正跳沿触发的边沿触发器的脉冲特性参数为: $t_{CPH} = 10 \text{ ns}$, $t_{CPLH} = 15 \text{ ns}$, $t_{SET} = 5 \text{ ns}$ 。画出脉冲工作特性波形图。



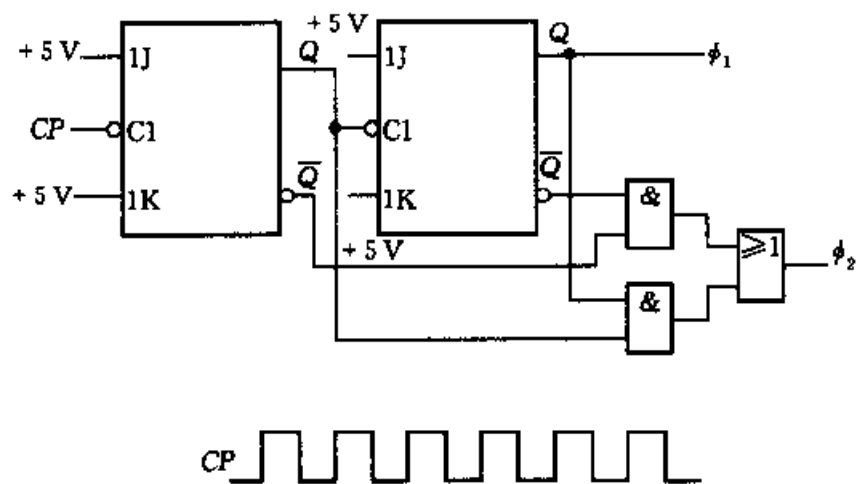
图题 5.2.4



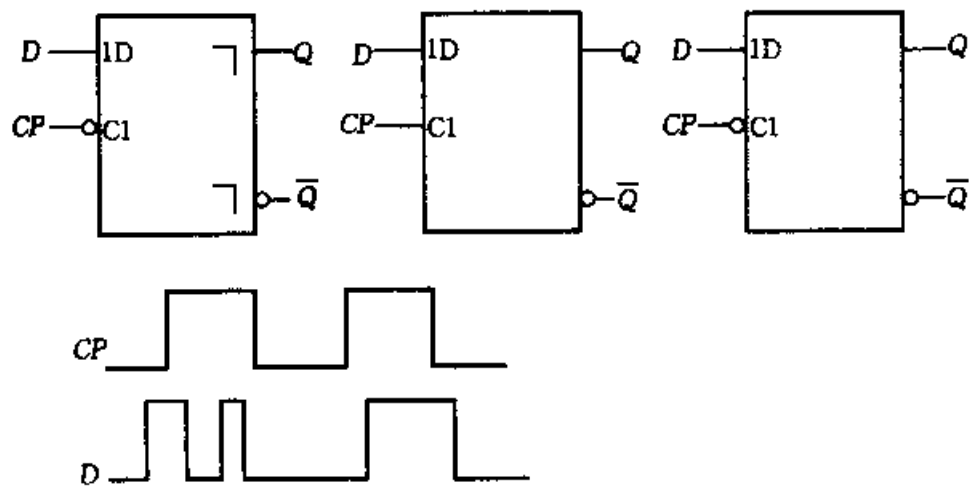
图题 5.2.5



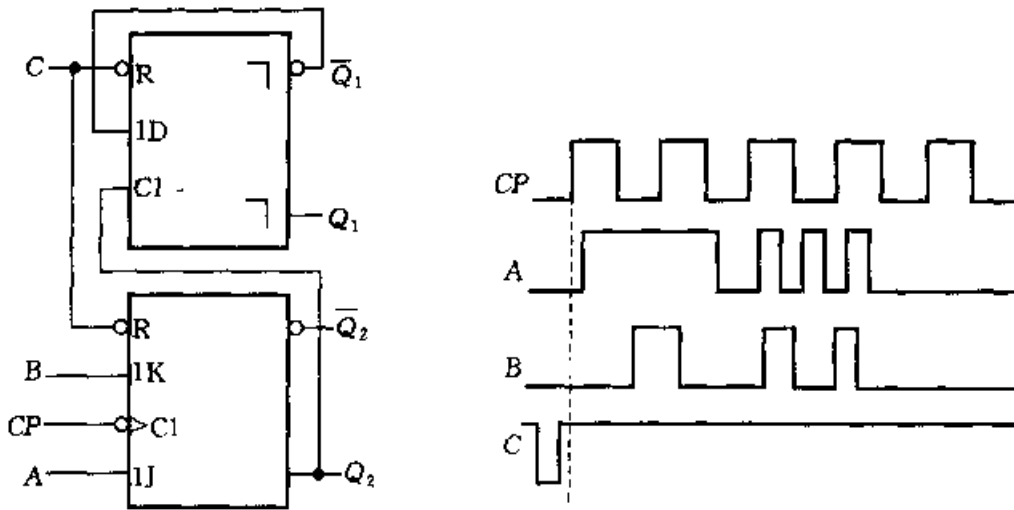
图题 5.2.6



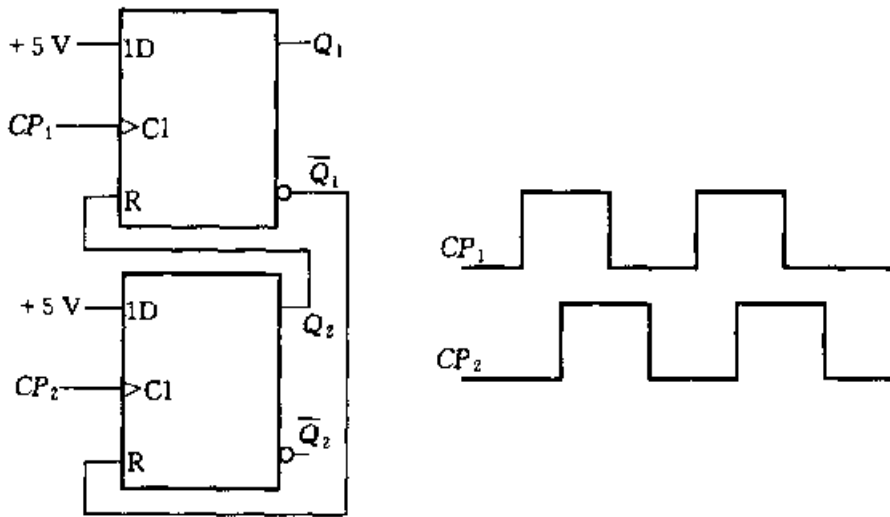
图题 5.2.7



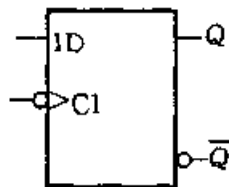
图题 5.2.8



图题 5.2.9



图题 5.2.10



图题 5.2.11

* CAD 习题

C5.4.1 试用 ISP Synario 软件对图题 5.1.1 所示的电路进行分析,并给出逻辑功能仿真波形图。

C5.4.2 试用 ISP Synario 软件分析图题 5.2.3 所示的逻辑电路,给出电路在 CP 作用下, F_0 、 F_1 、 F_2 和 F_3 的逻辑功能仿真波形图。

C5.4.3 试用 ISP Synario 软件对图题 5.2.5 所示的电路进行分析,给出电路在 CP 和 X 的作用下, Q1 和 Q2 的逻辑功能仿真波形。触发器的初始状态均为 0。

C5.4.4 两相脉冲产生电路如图题 5.2.7 所示,试用 ISP Synario 软件对电路进行逻辑功能仿真,给出电路在 CP 作用下 F_1 、 F_2 的波形,并说明 F_1 、 F_2 的相位差。各触发器的初始状态为 0。

6 时序逻辑电路的分析和设计

引言 前已介绍,逻辑电路可分为组合逻辑电路和时序逻辑电路两大类。从逻辑功能看,前面讨论的组合逻辑电路在任一时刻的输出信号仅仅与当时的输入信号有关;而时序逻辑电路在任一时刻的输出信号不仅与当时的输入信号有关,而且还与电路原来的状态有关。从结构上看,组合逻辑电路仅由若干逻辑门组成,没有存储电路,因而无记忆能力;而时序逻辑电路除包含组合电路外,还含有存储电路,因而有记忆能力。

本章将首先讲述时序逻辑电路的基本概念,然后讨论时序逻辑电路的分析和设计方法。

6.1 时序逻辑电路的基本概念

6.1.1 时序逻辑电路的基本结构及特点

由于时序逻辑电路(简称时序电路)在任一时刻的输出信号不仅与当时的输入信号有关,而且还与电路原来的状态有关,因此,时序逻辑电路中必须含有存储电路,由它将某一时刻之前的电路状态保存下来。存储电路可用延迟元件组成,也可用触发器构成。本章只讨论由触发器构成存储电路的时序电路。

时序电路的基本结构框图如图 6.1.1 所示,从总体上看,它由组合电路和存储电路两部分组成,其中 $X(X_1, \dots, X_n)$ 是时序逻辑电路的输入信号, $Q(Q_1, \dots, Q_r)$ 是存储电路的输出信号,它被反馈到组合电路的输入端,与输入信号共同决定时序逻辑电路的输出状态。 $Z(Z_1, \dots, Z_m)$ 是时序逻辑电路的输出信号^①, $Y(Y_1, \dots, Y_s)$ 是存储电路的输入信号。这些信号之间的逻辑关系可以表示为

^① 在有些时序电路中,输出信号不仅与存储电路的输出状态有关,而且还与时序电路的输入信号有关,这一类电路称为 Mealy 型电路;在另一些时序电路中,输出信号仅与存储电路的输出状态有关,这一类电路称为 Moore 型电路,这两种电路分别由 G. H. Mealy(1955)和 E. F. Moore(1956)提出的,详见文献[9]。

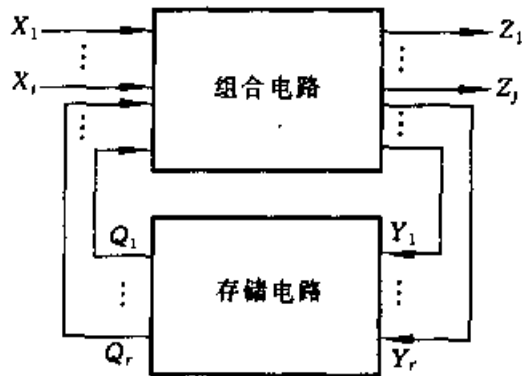


图 6.1.1 时序电路的结构框图

$$Z = F_1(X, Q^n) \quad (6.1.1)$$

$$Y = F_2(X, Q^n) \quad (6.1.2)$$

$$Q^{n+1} = F_3(Y, Q^n) \quad (6.1.3)$$

其中式(6.1.1)是输出方程。式(6.1.2)是存储电路的驱动方程(或称激励方程)。由于本章所用存储电路由触发器构成,即 Q_1, \dots, Q_r 表示的是构成存储电路的各个触发器的状态,所以式(6.1.3)是存储电路的状态方程,也就是时序逻辑电路的状态方程, Q^{n+1} 是次态, Q^n 是现态。

由上所述可知,时序逻辑电路具有以下特点:

- (1) 时序逻辑电路由组合电路和存储电路组成。
- (2) 时序逻辑电路中存在反馈,因而电路的工作状态与时间因素相关,即时序电路的输出由电路的输入和电路原来的状态共同决定。

6.1.2 时序逻辑电路的分类

时序逻辑电路可分为同步时序电路和异步时序电路两大类。在同步时序逻辑电路中,存储电路内所有触发器的时钟输入端都接于同一个时钟脉冲源,因而,所有触发器的状态(即时序逻辑电路的状态)的变化都与所加的时钟脉冲信号同步。在异步时序逻辑电路中,没有统一的时钟脉冲,有些触发器的时钟输入端与时钟脉冲源相连,只有这些触发器的状态变化才与时钟脉冲同步,而其他触发器状态的变化并不与时钟脉冲同步。由此可知,同步时序电路的速度高于异步时序电路,但电路结构一般较后者复杂。这两类时序逻辑电路的区别在下一节中将理解得更清楚。

以上是时序逻辑电路的主要分类方法,还可从另外的角度进行分类,限于篇幅,不再赘述。

6.1.3 时序逻辑电路功能的描述方法

1. 逻辑方程式

在 6.1.1 节里,根据时序电路的结构图,写出了时序电路的输出方程、驱动方程和状态方程。从理论上讲,有了这 3 个方程,时序电路的逻辑功能就被唯一地确定了,所以逻辑方程式可以描述时序电路的逻辑功能。值得提出的是,对许多时序逻辑电路而言,由 $Z = F_1(X, Q^n)$ 、 $Y = F_2(X, Q^n)$ 、 $Q^{n+1} = F_3(Y, Q^n)$ 这 3 个逻辑方程式还不能直观地看出时序电路的逻辑功能到底是什么,此外,设计时序逻辑电路时,往往很难根据给出的逻辑要求而直接写出电路的驱动方程、状态方程和输出方程。因此,下面再介绍几种能够反映时序电路状态变化全过程的描述方法。

2. 状态表

反映时序逻辑电路的输出 Z 、次态 Q^{n+1} 和电路的输入 X 、现态 Q^n 间对应取值关系的表格称为**状态表**。如表 6.1.1 所示,时序逻辑电路的全部输入信号列在状态表的顶部,表的左边列出现态,表的内部列出次态和输出。状态表的读法是,处在现态 Q^n 的时序电路,当输入为 X 时,该电路将进入输出为 Z 的次态 Q^{n+1} 。

表 6.1.1 时序逻辑电路的状态表

次态/输出 现态		输入	
		X	
Q ⁿ	Q ⁿ⁺¹ /Z		

3. 状态图

反映时序逻辑电路状态转换规律及相应输入、输出取值关系的图形称为**状态图**,如图 6.1.2 所示。在状态图中,圆圈及圈内的字母或数字表示电路的各个状态,连线及箭头表示状态转换的方向(由现态到次态),当箭头的起点和终点都在同一个圆圈上时,则表示状态不变。标在连线一侧的数字表示状态转换前输入信号的取值和输出值。通常将输入信号的取值写在斜线以上,输出值写在斜线以下。它清楚地表明,在该输入取值作用下,将产生相应的输出值,同时,电路将发生如箭头所指的状态转换。

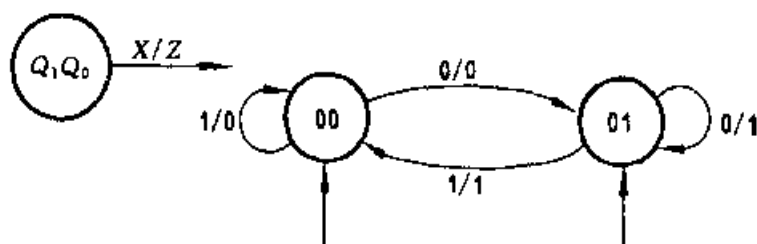


图 6.1.2 时序电路的状态图

4. 时序图

时序图即时序电路的工作波形图。它能直观地描述时序电路的输入信号、时钟信号、输出信号及电路的状态转换等在时间上的对应关系。

上面介绍的描述时序电路逻辑功能的4种方法可以互相转换。在介绍时序逻辑电路的分析和设计方法时,将具体讲述以上4种描述方法的应用。

复习思考题

6.1.1 时序逻辑电路由哪几部分组成?它和组合电路的区别是什么?时序电路可分为哪两大类?

6.1.2 描述时序电路逻辑功能的方法有哪几种?它们有何关系?状态表和状态图怎样构成?

6.1.3 两个多位数相加时,除了采用第4章中介绍的并行加法方式外,还可采用串行加法,即从低位到高位逐位相加的方式完成加法运算。试说明串行加法器是属于组合电路还是属于时序电路,并画出一种串行加法器电路。

6.2 时序逻辑电路的分析方法

时序逻辑电路的分析就是根据给定的时序逻辑电路图,通过分析,求出它的输出 Z 的变化规律,以及电路状态 Q 的转换规律,进而说明该时序电路的逻辑功能和工作特性。

下面先介绍分析时序逻辑电路的一般步骤,然后通过例题分析加深对分析方法的理解。

6.2.1 分析时序逻辑电路的一般步骤

1. 根据给定的时序电路图写出下列各逻辑方程式:

- (1) 各触发器的时钟信号 CP 的逻辑表达式;
- (2) 时序电路的输出方程;
- (3) 各触发器的驱动方程。

2. 将驱动方程代入相应触发器的特性方程,求得各触发器的次态方程,也就是时序逻辑电路的状态方程。

3. 根据状态方程和输出方程,列出该时序电路的状态表,画出状态图或时序图。

4. 用文字描述给定时序逻辑电路的逻辑功能。

需要说明的是,上述步骤不是必须执行的固定程序,实际应用中可根据具体情况加以取舍,例如,在同步时序电路中,各触发器的时钟输入端都接至同一个

时钟脉冲源,因此各触发器的时钟信号 CP 的逻辑表达式就可以不写。

6.2.2 同步时序逻辑电路的分析举例

例 6.2.1 试分析图 6.2.1 所示时序电路。

解: 分析过程如下:

(1) 写出各逻辑方程式

① 这是一个同步时序电路,各触发器 CP 信号的逻辑表达式可以不写。

② 输出方程

$$Z = Q_1^n Q_0^n$$

③ 驱动方程

$$\begin{aligned} J_0 &= 1 & K_0 &= 1 \\ J_1 &= X \oplus Q_0^n & K_1 &= X \oplus Q_0^n \end{aligned}$$

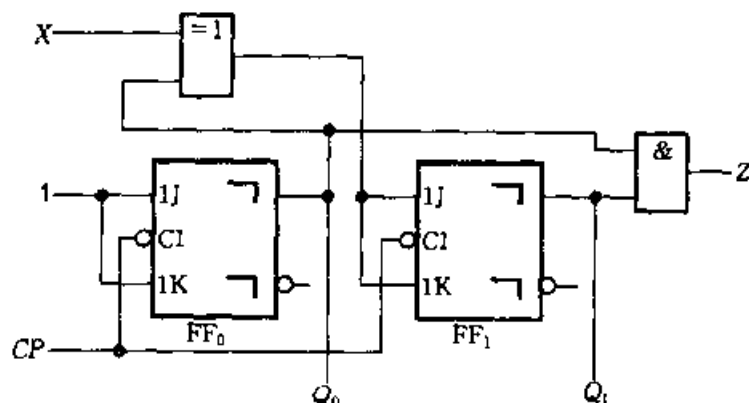


图 6.2.1 例 6.2.1 的逻辑电路图

(2) 将驱动方程代入相应 JK 触发器的特性方程,求得各触发器的次态方程为

$$\begin{aligned} Q_0^{n+1} &= J_0 \bar{Q}_0^n + \bar{K}_0 Q_0^n = \bar{Q}_0^n \\ Q_1^{n+1} &= J_1 \bar{Q}_1^n + \bar{K}_1 Q_1^n \\ &= (X \oplus Q_0^n) \bar{Q}_1^n + \overline{X \oplus Q_0^n} Q_1^n \\ &= X \oplus Q_0^n \oplus Q_1^n \end{aligned}$$

(3) 列状态表、画状态图和时序图

列状态表是分析时序逻辑电路的关键性的一步,其具体做法是:先填入电路现态 Q^n (本例中为 Q_1^n, Q_0^n) 的所有组合状态以及输入信号 X 的所有组合状态,然后根据输出方程及状态方程,逐行填入当前输出 Z 的相应值,以及次态 Q^{n+1} (Q_1^{n+1}, Q_0^{n+1}) 的相应值。照此做法,可列出例 6.2.1 的状态表,如表 6.2.1 所示。

表 6.2.1 例 6.2.1 的状态表

$Q_1^n Q_0^n$		X	
		0	1
$Q_1^{n+1} Q_0^{n+1} / Z$	0 0	01/0	11/0
	0 1	10/0	00/0
	1 0	11/0	01/0
	1 1	00/1	10/1

根据状态表可以画出状态图。例 6.2.1 的状态图如图 6.2.2 所示,它展示的电路状态变化的规律是:

若输入信号 $X = 0$, 当现态 $Q_1^n Q_0^n = 00$ 时, 则当前输出 $Z = 0$, 在一个 CP 脉冲作用后, 电路转向次态 $Q_1^{n+1} Q_0^{n+1} = 01$; 当 $Q_1^n Q_0^n = 01$ 时, 则 $Z = 0$, 在 CP 作用后, $Q_1^{n+1} Q_0^{n+1} = 10$; 当 $Q_1^n Q_0^n = 10$ 时, 则 $Z = 0$, 在 CP 作用后, $Q_1^{n+1} Q_0^{n+1} = 11$; 当 $Q_1^n Q_0^n = 11$ 时, 则 $Z = 1$, 在 CP 作用后, $Q_1^{n+1} Q_0^{n+1} = 00$ 。

若输入信号 $X = 1$, 电路状态转换的方向则与上述方向相反。

设电路的初始状态为 $Q_1^n Q_0^n = 00$, 根据状态表和状态图, 可画出在一系列 CP 脉冲作用下, 例 6.2.1 电路的时序图, 如图 6.2.3 所示。

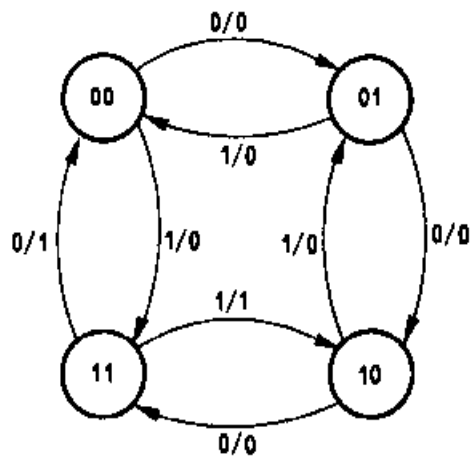


图 6.2.2 例 6.2.1 的状态图

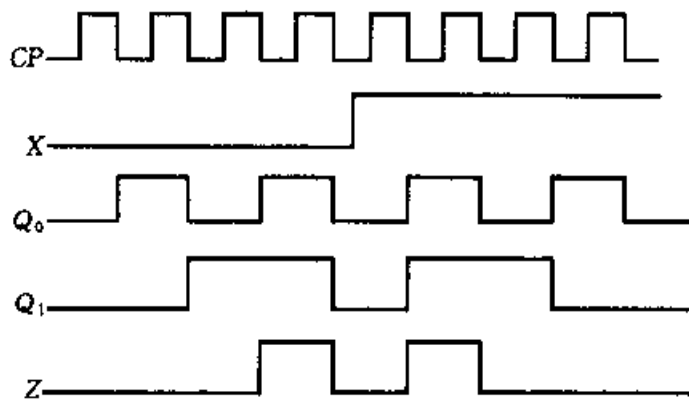


图 6.2.3 例 6.2.1 电路的时序图

(4) 逻辑功能分析

实际上, 由该例的状态图就可看出, 此电路是一个可控计数器。当 $X = 0$

时,进行加法计数,在时钟脉冲作用下, $Q_1 Q_0$ 的数值从 00 到 11 递增,每经过 4 个时钟脉冲作用后,电路的状态循环一次。同时在 Z 端输出一个进位脉冲,因此, Z 是进位信号。当 $X=1$ 时,进行减 1 计数, Z 是借位信号。有关计数器的详细内容将在第 7 章加以介绍。

例 6.2.2 分析图 6.2.4 所示的逻辑电路。

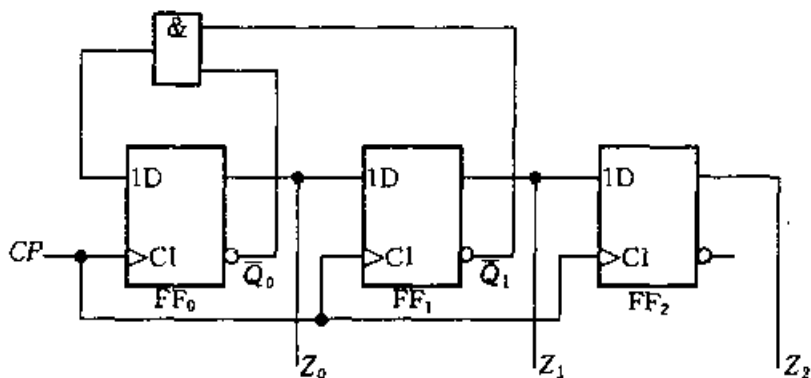


图 6.2.4 例 6.2.2 的逻辑电路图

解: 由图 6.2.4 可见,这是一个同步时序逻辑电路,电路中没有输入信号 X ,而且电路的输出直接由各触发器的 Q 端取出。分析过程如下:

(1) 写出各逻辑方程

① 输出方程

$$Z_0 = Q_0^n \quad Z_1 = Q_1^n \quad Z_2 = Q_2^n$$

② 驱动方程

$$D_0 = \overline{Q_0^n} \overline{Q_1^n} \quad D_1 = Q_0^n \quad D_2 = Q_1^n$$

(2) 将驱动方程代入相应 D 触发器的特性方程,求得各 D 触发器的次态方程

$$Q_0^{n+1} = D_0 = \overline{Q_0^n} \overline{Q_1^n}$$

$$Q_1^{n+1} = D_1 = Q_0^n$$

$$Q_2^{n+1} = D_2 = Q_1^n$$

(3) 列状态表、画状态图和时序图

由于此电路中 $Z_2 Z_1 Z_0 = Q_2^n Q_1^n Q_0^n$,输出与现态完全相同,所以状态表中可以不再列出输出。又因这个电路中没有输入信号 X ,状态表中的次态仅有一列,因而此电路的状态表可简化成表 6.2.2 所示的形式。

根据状态表可画出这个电路的状态图,如图 6.2.5 所示。由状态图可见,001、010、100 这 3 个状态形成了闭合回路,在电路正常工作时,电路状态总是按照回路中的箭头方向循环变化,因此这 3 个状态构成了有效序列,称它们为有效状态,其余的 5 个状态称为无效状态。

表 6.2.2 例 6.2.2 的状态表

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

设电路的初始状态为 $Q_2 Q_1 Q_0 = 000$, 根据状态表或状态图, 可画出在一系列 CP 脉冲作用下该电路的时序图, 如图 6.2.6 所示。

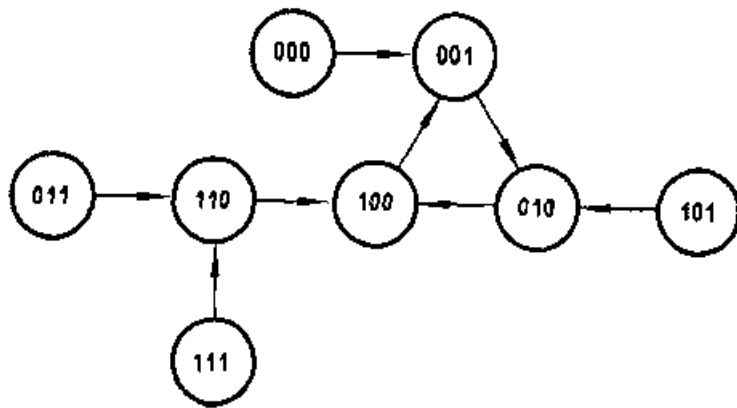


图 6.2.5 例 6.2.2 的状态图

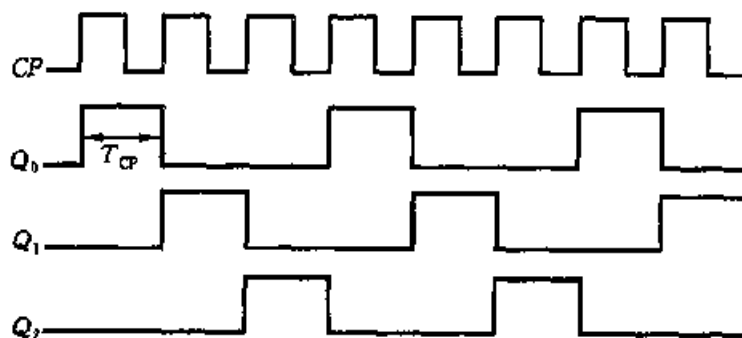


图 6.2.6 例 6.2.2 的时序图

(4) 逻辑功能分析

由该电路的状态表和状态图不太容易直接看出此电路的逻辑功能, 而由它的时序图可见, 这个电路在正常工作时, 各触发器的 Q 端轮流出现一个脉冲信

号,其宽度为一个CP周期,即 $1T_{CP}$,循环周期为 $3T_{CP}$,这个动作可以看作是在CP脉冲作用下,电路把宽度为 $1T_{CP}$ 的脉冲依次分配给 Q_0 、 Q_1 、 Q_2 各端,所以此电路的功能为脉冲分配器或节拍脉冲产生器。还应指出的是,由状态图可知,若此电路由于某种原因进入无效状态时,在CP脉冲作用后,电路能自动回到有效序列,这种能力称为电路具有自启动能力。

6.2.3 异步时序逻辑电路的分析举例

在异步时序逻辑电路中,由于没有统一的时钟脉冲,分析时必须注意,触发器只有在加到其CP端上的信号有效时,才有可能改变状态。否则,触发器将保持原有状态不变。因此,在考虑各触发器状态转换时,除考虑驱动信号的情况外,还必须考虑其CP端的情况,即根据各触发器的时钟信号CP的逻辑表达式及触发方式,确定各CP端是否有触发信号作用(对于由上升沿触发的触发器而言,当其CP端的信号由0变1时,则有触发信号作用;对于由下降沿触发的触发器而言,当其CP端的信号由1变0时,则有触发信号作用)。有触发信号作用的触发器能改变状态;无触发信号作用的触发器则保持原有的状态不变。举例如下。

例 6.2.3 分析图 6.2.7 所示逻辑电路。

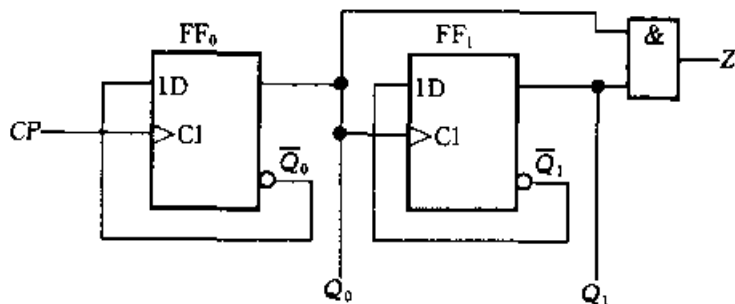


图 6.2.7 例 6.2.3 的逻辑电路图

解: 在此电路中, CP_1 未与时钟脉冲源CP相连,属异步时序电路。

(1) 写出各逻辑方程式

① 各触发器的时钟信号的逻辑方程

$CP_0 = CP$ (时钟脉冲源),上升沿触发。

$CP_1 = Q_0$,仅当 Q_0 由0→1时, Q_1 才可能改变状态,否则 Q_1 将保持原有状态不变。

② 输出方程

$$Z = Q_1^n Q_0^n$$

③ 驱动方程

$$D_0 = \overline{Q_0^n}, \quad D_1 = \overline{Q_1^n}$$

(2) 各触发器的次态方程

$$Q_0^{n+1} = D_0 = \overline{Q_0^n} \quad (\text{CP 由 } 0 \rightarrow 1 \text{ 时此式有效})$$

$$Q_1^{n+1} = D_1 = \overline{Q_1^n} \quad (\text{Q}_0 \text{ 由 } 0 \rightarrow 1 \text{ 时此式有效})$$

(3) 列状态表、画状态图和时序图

列状态表的方法与同步时序电路基本相似,只是还应注意各触发器 CP 端的状况(是否有上升沿作用),因此,可在状态表中增加各触发器 CP 端的状况,无上升沿作用时的 CP 用 0 表示。例 6.2.3 的状态表如表 6.2.3 所示。

表 6.2.3 例 6.2.3 的状态表

Q_1^n	Q_0^n	CP_0	CP_1	$Q_1^{n+1} Q_0^{n+1} / Z$
0	0			11/0
0	1		0	00/0
1	0			01/0
1	1		0	10/1

由状态表可画出状态图,如图 6.2.8 所示。此电路的时序图如图 6.2.9 所示。

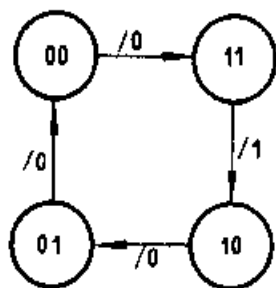


图 6.2.8 例 6.2.3 的状态图

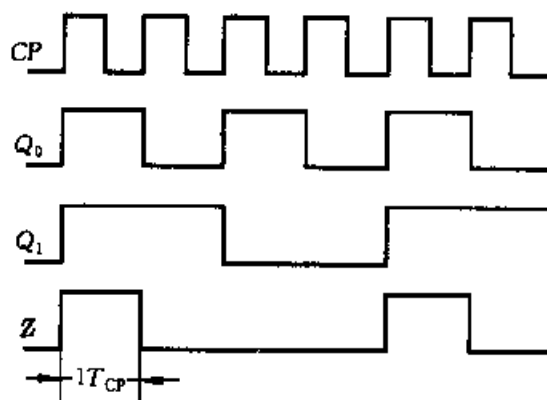


图 6.2.9 例 6.2.3 的时序图

(4) 逻辑功能分析

由状态图和时序图可知,此电路是一个异步四进制减法计数器,Z 是借位信号。也可把该电路看作一个序列信号发生器。输出序列脉冲信号 Z 的重复周期为 $4T_{CP}$,脉宽为 $1T_{CP}$ 。

复习思考题

6.2.1 时序逻辑电路的分析过程大致分哪几步?

6.2.2 在分析方法上,异步时序电路与同步时序电路的不同之处何在?

6.3 同步时序逻辑电路的设计方法

时序电路设计又称时序电路综合,它是时序电路分析的逆过程,即根据给定的逻辑功能要求,选择适当的逻辑器件,设计出符合要求的时序逻辑电路。本节仅介绍用触发器及门电路设计同步时序电路的方法,这种设计方法的基本指导思想是用尽可能少的时钟触发器和门电路来实现待设计的时序电路。

6.3.1 同步时序逻辑电路设计的一般步骤

设计同步时序电路的一般过程如图 6.3.1 所示。

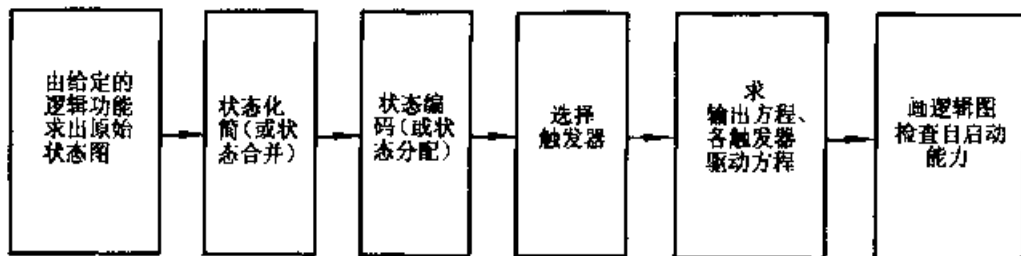


图 6.3.1 同步时序电路的设计过程

下面对设计过程中的主要步骤加以说明。

1. 由给定的逻辑功能求出原始状态图

由于时序电路在某一时刻的输出信号,不仅与当时的输入信号有关,而且还与电路原来的状态有关。因此设计时序电路时,首先必须分析给定的逻辑功能,从而求出对应的状态转换图。这种直接由要求实现的逻辑功能求得的状态转换图叫做原始状态图。正确画出原始状态图,是设计时序电路的最关键的一步,具体做法是:

(1) 分析给定的逻辑功能,确定输入变量、输出变量及该电路应包含的状态,并用字母 S_0 、 S_1 、 \dots 表示这些状态。

(2) 分别以上述状态为现态,考察在每一个可能的输入组合作用下应转入哪个状态及相应的输出,便可求得符合题意的状态图。

2. 状态化简

根据给定要求得到的原始状态图不一定是简的,很可能包含有多余的状态,即可以合并的状态,因此需要进行状态化简或状态合并。状态化简是建立在状态等价这个概念的基础上的。所谓状态等价,是指在原始状态图中,如果有两个或两个以上的状态,在输入相同的条件下,不仅有相同的输出,而且向同一个次态转换,则称这些状态是等价的。凡是等价状态都可以合并。如图 6.3.2 中的状态 S_2 和 S_3 ,当输入 $X=0$ 时,输出 Z 都是 0,且都向同一个次态 S_0 转换;

当 $X=1$ 时, 输出 Z 都是 1, 次态都是 S_3 , 所以 S_2 和 S_3 是等价状态, 可以合并为 S_2 , 取消 S_3 , 即将图 6.3.2 中代表 S_3 的圆圈及由该圆圈出发的所有连线去掉, 将原先指向 S_3 的连线改而指向 S_2 , 得到化简后的状态图如图 6.3.3 所示。

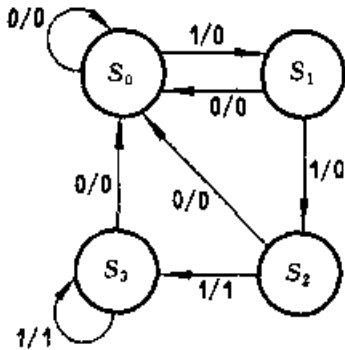


图 6.3.2 原始状态图

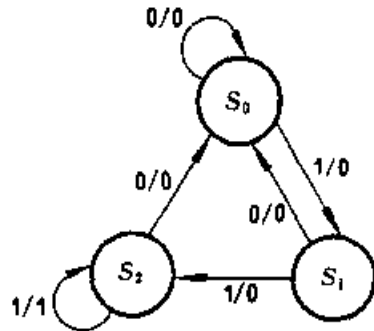


图 6.3.3 图 6.3.2 的简化状态图

显然, 状态化简使状态数目减少, 从而可以减少电路所需触发器的个数或门电路的个数。

3. 状态编码、并画出编码形式的状态图及状态表

在得到简化的状态图后, 要对每一个状态指定 1 个二进制代码, 这就是状态编码(或称状态分配)。编码的方案不同, 设计的电路结构也就不同。编码方案选择得当, 设计结果可以很简单。为此, 选取的编码方案应该有利于所选触发器的驱动方程及电路输出方程的简化。为便于记忆和识别, 一般选用的状态编码都遵循一定的规律, 如用自然二进制码。编码方案确定后, 根据简化的状态图, 画出编码形式的状态图及状态表。

4. 选择触发器的类型及个数

按照下式选择触发器的个数 n :

$$2^{n-1} < M \leq 2^n \quad (6.3.1)$$

其中 M 是电路包含的状态个数。

5. 求电路的输出方程及各触发器的驱动方程

根据编码后的状态表及触发器的驱动表可求得电路的输出方程和各触发器的驱动方程。

6. 画逻辑电路图, 并检查自启动能力

6.3.2 同步时序逻辑电路设计举例

例 6.3.1 试设计一序列脉冲检测器, 当连续输入信号 110 时, 该电路输出为 1, 否则输出为 0。

解: 由设计要求可知, 要设计的电路有一个输入信号 X 和一个输出信号 Z , 电路功能是对输入信号进行检测。

(1) 由给定的逻辑功能确定电路应包含的状态,并画出原始状态图。

因为该电路在连续收到信号 **110** 时,输出为 **1**,其他情况下输出为 **0**,因此要求该电路能记忆收到的输入为 **0**、收到 **1** 个 **1**、连续收到两个 **1**、连续收到 **110** 后的状态,由此可见该电路应有 4 个状态,用 S_0 表示输入为 **0** 时的电路状态(或称初始状态), S_1 、 S_2 、 S_3 分别表示收到一个 **1**、连续收到两个 **1** 和连续收到 **110** 时的状态。先假设电路处于状态 S_0 ,在此状态下,电路可能的输入有 $X=0$ 和 $X=1$ 两种情况。若 $X=0$,则输出 $Z=0$,且电路应保持在状态 S_0 不变;若 $X=1$,则 $Z=0$,但电路应转向状态 S_1 ,表示电路收到了一个 **1**。现在以 S_1 为现态,若这时输入 $X=0$,则输出 $Z=0$,且电路应回到 S_0 ,重新开始检测;若 $X=1$,则输出 $Z=0$,且电路应进入 S_2 ,表示已连续收到了二个 **1**。又以 S_2 为现态,若输入 $X=0$,则输出 $Z=1$,电路应进入 S_3 状态,表示已连续收到了 **110**;若 $X=1$,则 $Z=0$,且电路应保持在状态 S_2 不变。再以 S_3 为现态,若输入 $X=0$,则输出 $Z=0$,电路应回到状态 S_0 ,重新开始检测;若 $X=1$,则 $Z=0$,电路应转向状态 S_1 ,表示又重新收到了一个 **1**。根据上述分析,可以画出该例题的原始状态图,如图 6.3.4 所示。

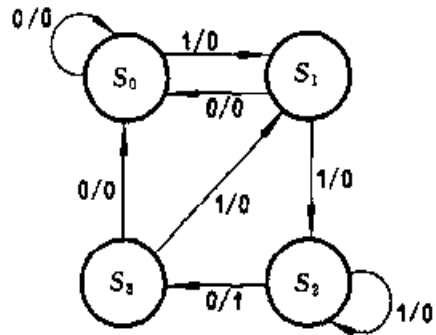


图 6.3.4 例 6.3.1 的原始状态图

(2) 状态化简

观察图 6.3.4 便知, S_0 和 S_3 是等价状态,因为当输入 $X=0$ 时,输出 Z 都为 **0**,而且次态均转向 S_0 ;当 $X=1$ 时,输出 Z 都为 **0**,而且次态均转向 S_1 ,所以 S_0 和 S_3 可以合并。去掉 S_3 的圆圈及由此圆圈出发的连线,将指向 S_3 的连线指向 S_0 ,得到简化后的状态图,如图 6.3.5 所示。

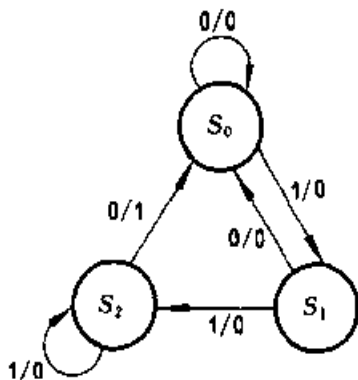


图 6.3.5 例 6.3.1 的简化状态图

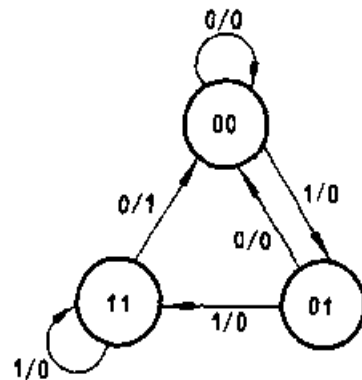


图 6.3.6 例 6.3.1 的编码形式的状态图

(3) 状态编码及画编码形式的状态图和状态表

由图 6.3.5 可知,该电路有 3 个状态,可以用 2 位二进制代码组合(00,01,10,11)中的任意三个代码表示,这里取 00,01,11 分别表示 S_0 、 S_1 、 S_2 ,即令 $S_0 = 00$, $S_1 = 01$, $S_2 = 11$ 。图 6.3.6 是该例的编码形式的状态图。

由编码形式的状态图可画出编码后的状态表如表 6.3.1 所示。

表 6.3.1 例 6.3.1 的编码状态表

$Q_1^{n+1} Q_0^{n+1} / Z$		X	
		0	1
$Q_1^n Q_0^n$			
0	0	00/0	01/0
0	1	00/0	11/0
1	1	00/1	11/0

(4) 选择触发器

根据式(6.3.1)可知,本例需用两个触发器,可选用前面介绍的集成 JK 触发器 HC76。

(5) 确定各触发器的驱动方程及电路的输出方程。

根据编码状态表 6.3.1 及 JK 触发器的驱动表,画出各触发器驱动信号及电路输出信号的真值表,如表 6.3.2 所示,由此表画出各触发器 J、K 端和电路输出端 Z 的卡诺图,如图 6.3.7 所示,利用卡诺图化简得各触发器的驱动方程及电路的输出方程。

$$\begin{cases} J_0 = X & K_0 = \bar{X} \\ J_1 = XQ_0 & K_1 = \bar{X} \\ Z = \bar{X}Q_1 \end{cases} \quad (6.3.2)$$

表 6.3.2 驱动信号及输出信号的真值表

输入 X	现 态		次 态		输出 Z	驱 动 信 号			
	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}		J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	×	0	×
0	0	1	0	0	0	0	×	×	1
0	1	1	0	0	1	×	1	×	1
1	0	0	0	1	0	0	×	1	×
1	0	1	1	1	0	1	×	×	0
1	1	1	1	1	0	×	0	×	0

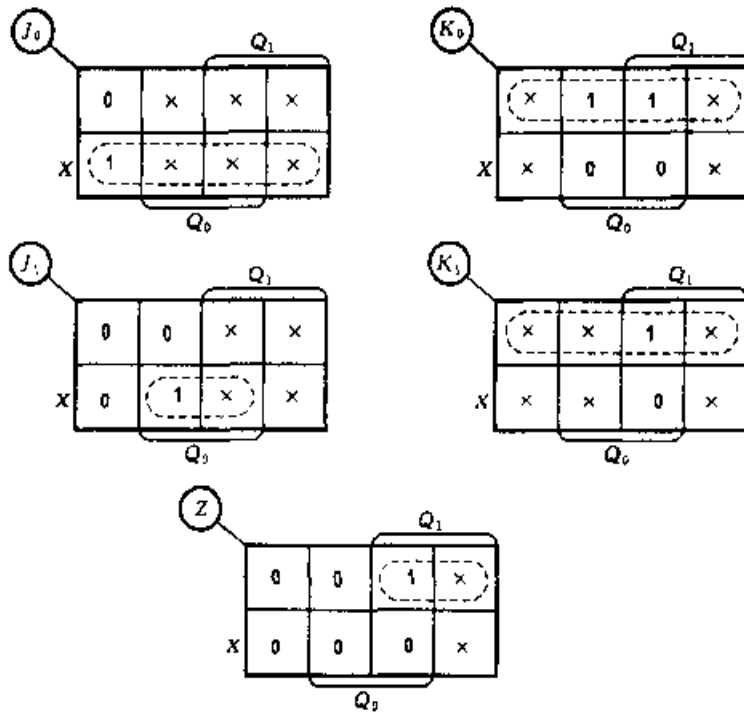


图 6.3.7 驱动信号及输出信号的卡诺图

(6) 根据式(6.3.2)画出逻辑电路图,如图 6.3.8 所示。

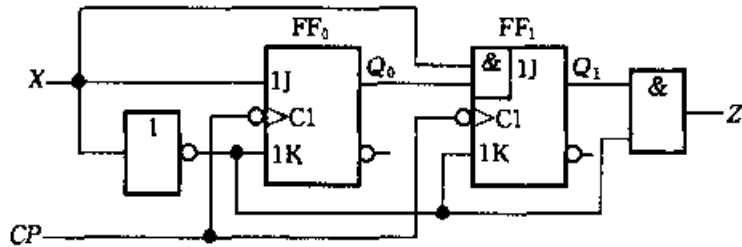


图 6.3.8 例 6.3.1 的逻辑电路图

最后还应检查该电路的自启动能力。当电路进入无效状态 10 后,由式(6.3.2)可知,若 $X=0$,则次态为 00;若 $X=1$,则次态为 11,电路能自动进入有效序列。但从输出来看,若电路在无效状态 10,当 $X=0$ 时, $Z=1$,这是错误的。为了消除这个错误输出,需要对输出方程作适当修改,即将图 6.3.7 中输出信号 Z 卡诺图内的无关项 $\bar{X}Q_1\bar{Q}_0$ 不画在包围圈内,则输出方程变为 $Z = \bar{X}Q_1Q_0$,根据此式对图 6.3.8 也作相应修改即可。

如果发现设计的电路没有自启动能力,则应对设计进行修改。其方法是:在驱动信号之卡诺图的包围圈中,对无效状态 \times 的处理作适当修改,即原来取 1 画入包围圈的,可试改为取 0 而不画入包围圈,或者相反。得到新的驱动方程和逻辑图,再检查其自启动能力,直到能够自启动为止。

复习思考题

- 6.3.1 同步时序电路的设计大致分哪几步?
- 6.3.2 什么叫原始状态图? 怎样画出原始状态图?
- 6.3.3 什么叫状态等价? 若状态 S_A 与 S_B 等价, S_B 又与 S_C 等价, 那么状态 S_A 与 S_C 等价吗?

*6.4 CAD 例题

例 CE6.4.1 图 CE6.4.1(即图 6.2.2)是某时序电路的状态转换图, 决定状态机状态的寄存器为 Q_1 、 Q_0 , 用集合 $[Q_1, Q_0]$ 表示; 电路状态转换的方向由输入信号 M 决定, 当 $M=0$ 时, 电路状态按顺时针方向变化, 反之, 则按逆时针方向变化。假定电路的初始状态为 01 , 试用 ABEL 中状态机的结构进行描述, 并给出逻辑功能的仿真波形。

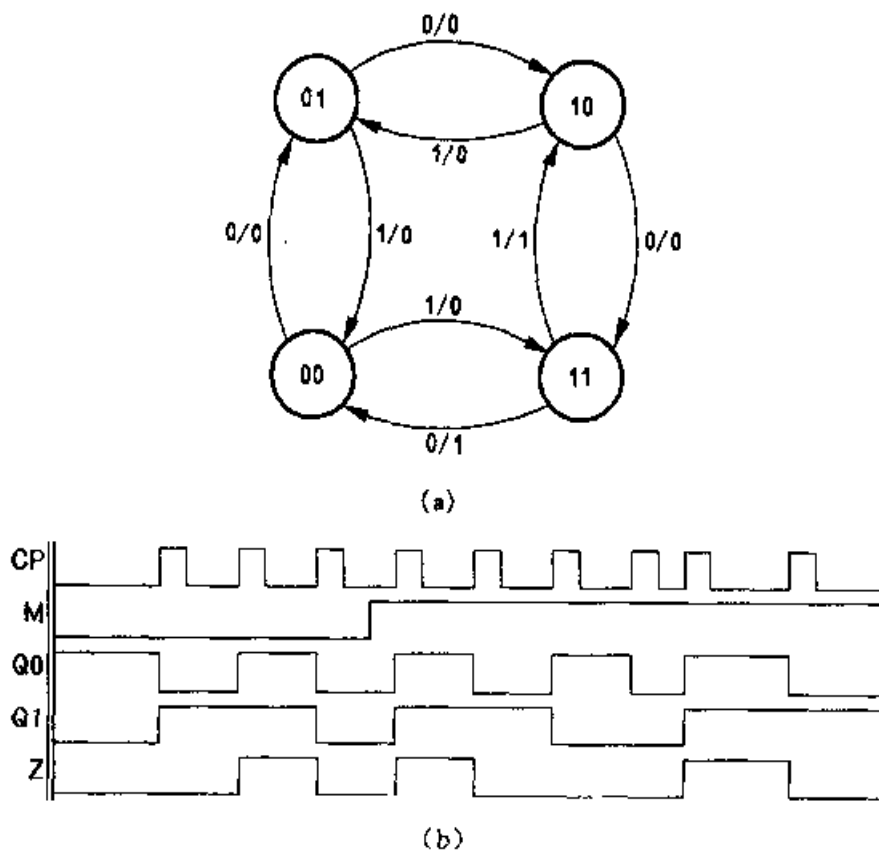


图 CE6.4.1 电路的状态转换图及逻辑功能仿真波形图

(a) 状态转换图 (b) 逻辑功能仿真波形图

解: (1) 输入设计文件

首先对电路的状态转换图用 ABEL 语言进行描述, 将编写的 ABEL 程序

(包括测试向量)输入到 ISP Synario 软件中,然后对程序进行编译和逻辑简化。

上述状态转换的 ABEL 语言程序:

```

module CE6_4_1      "模块开始
title 'State _ machine example
declarations
  M, CP          pin;
  Q1, Q0        pin istype 'reg';
  Z             pin istype 'com';
  Q=[Q1,Q0];    "定义状态寄存器变量
  S0 = ^b00;   S1 = ^b01;  "定义计数器的状态
  S2 = ^b10;   S3 = ^b11;
  C,P,X = .C., .P., .X.; "定义特殊常量
equations
  Q.clk = CP;   "定义两个输出寄存器的时钟信号由输入信号 CP 提供。
state _ diagram Q
  state S0; if M = 0 then S1 with Z = 0;
             else S3 with Z = 0;
  state S1; if M = 0 then S2 with Z = 0;
             else S0 with Z = 0;
  state S2; if M = 0 then S3 with Z = 0;
             else S1 with Z = 0;
  state S3; if M = 0 then S0 with Z = 1;
             else S2 with Z = 1;
test _ vectors      "定义测试向量
  ([CP,M,Q1,Q0] -> [Q1,Q0,Z])
  [P,0,1,0] -> X;   "预置触发器的初始状态(反码预置)
  [0,0,X,X] -> [0,1,0];
  [C,0,X,X] -> [1,0,0];
  [C,0,X,X] -> [1,1,1];
  [C,0,X,X] -> [0,0,0];
  [C,1,X,X] -> [1,1,1];
  [C,1,X,X] -> [1,0,0];
  [C,1,X,X] -> [0,1,0];
  [C,1,X,X] -> [0,0,0];
  [C,1,X,X] -> [1,1,1];

```



```
[C,1,X,X] ->[1,0,0];
end
```

(2) 逻辑功能仿真

在 ISP Synario 软件中,对包含在上述程序中的测试向量文件进行编译和逻辑功能仿真,得到图 CE6.4.1b 所示的逻辑功能仿真波形图。

例 CE6.4.2 用 ISP Synario 软件分析图 CE6.4.2a(即图题 6.2.4)所示电路,说明其逻辑功能,并给出逻辑功能仿真波形。

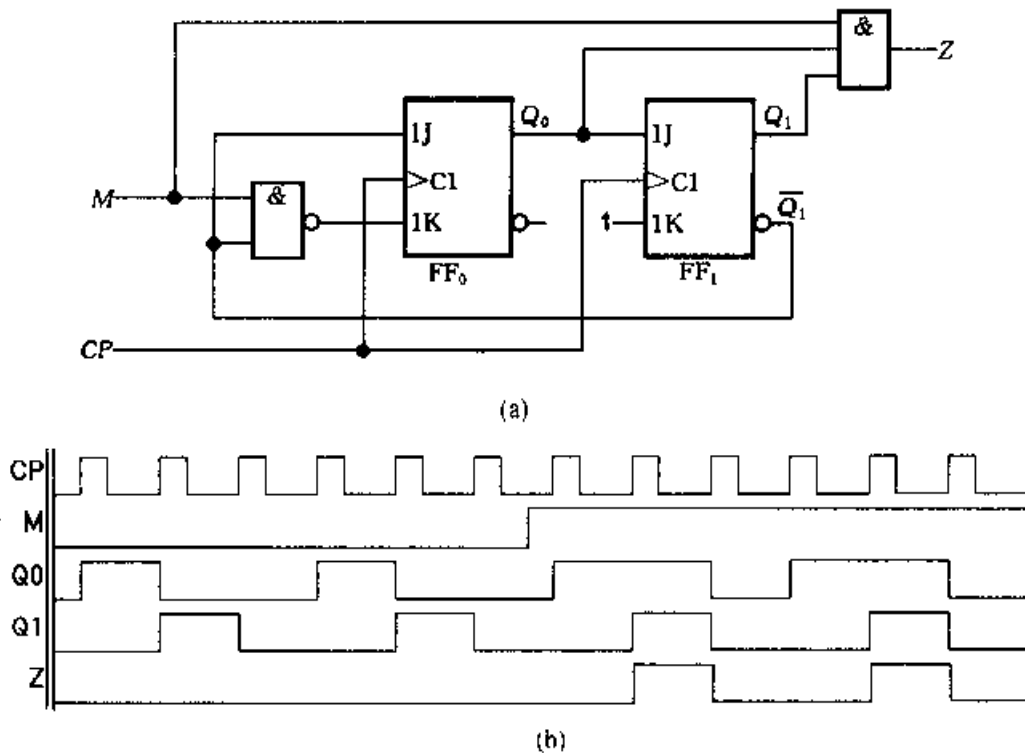


图 CE6.4.2 逻辑电路及仿真波形

(a) 逻辑电路图 (b) 逻辑功能仿真波形图

解: (1) 输入设计文件

调用 ISP Synario 软件中的原理图编辑器,输入图 CE6.4.2a 所示逻辑电路,并保存文件。然后对原理图文件进行编译和逻辑简化。

(2) 编写并输入测试向量文件

根据图 CE6.4.2a 所示逻辑电路,编写的测试向量文件如下:

```
module CE6_4_2
    CP,M    pin;
    Q1,Q0   pin istype 'reg';
    Z       pin istype 'com';
    C,X = .C.,.X.;
```

```

test_vectors
    ([CP,M] -> |Q1,Q0,Z])
@ repeat 6 |[C,0] -> X;|
@ repeat 6 |[C,1] -> X;|
end

```

(3) 进行逻辑功能仿真

在 ISP Synario 软件中,对测试向量文件进行编译和逻辑功能仿真,得到图 CE6.4.2b 所示的逻辑功能仿真波形图。

小 结

• 时序逻辑电路通常由组合电路及存储电路两部分组成。其中存储电路能将电路的状态记忆下来,并和当前的输入信号一起决定电路的输出信号。这是时序电路在结构上的特点,这个特点决定了时序电路的逻辑功能,即时序电路在任一时刻的输出信号不仅和当时的输入信号有关,而且还与电路原来的状态有关。

• 描述时序电路逻辑功能的方法有逻辑方程组(含驱动方程、状态方程和输出方程)、状态表、状态图和时序图,它们各具特色,各有所用,且可以相互转换。逻辑方程组是和具体时序电路直接对应的,状态表和状态图能给出时序电路的全部工作过程,时序图能更直观地显示电路的工作过程。为进行时序电路的分析和设计,应该熟练地掌握这几种描述方法。

• 就工作方式而言,时序电路可分为同步时序电路和异步时序电路两类。它们的主要区别是,在同步时序电路的存储电路中,所有触发器的 CP 端均受同一时钟脉冲源控制,而在异步时序电路中,各触发器 CP 端受不同的触发脉冲控制。

• 时序电路的分析和设计是两个相反的过程,时序电路的分析是由给定的时序电路,写出逻辑方程组,列出状态表,画出状态图或时序图,指出电路逻辑功能的过程。时序电路的设计是根据要求实现的逻辑功能,作出原始状态图或原始状态表,然后进行状态化简(状态合并)和状态编码(状态分配),再求出所选触发器的驱动方程、时序电路的状态方程和输出方程,最后画出设计好的逻辑电路图的过程。其中画出正确的原始状态图或原始状态表是关键的一步,是后面几个设计步骤的基础。即使是用可编程逻辑器件来设计时序电路,这一步也是不可缺少的。

习 题

6.1.1 已知一时序电路的状态表如表题 6.1.1 所示,试作出相应的状态图。

表题 6.1.1

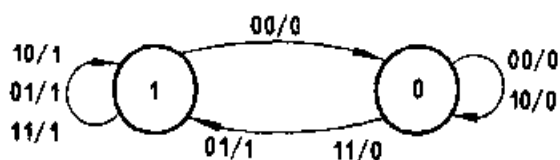
现态	次态/输出	输入 X	
		0	1
S_0		$S_3/1$	$S_1/0$
S_1		$S_3/1$	$S_2/0$
S_2		$S_3/1$	$S_0/0$
S_3		$S_1/1$	$S_2/0$

6.1.2 已知状态表如表题 6.1.2 所示,试作出相应的状态图。

表题 6.1.2

现态	次态/输出 Z_1				输出 Z_2
	$X_2X_1=00$	$X_2X_1=01$	$X_2X_1=11$	$X_2X_1=10$	
S_0	$S_0/0$	$S_1/0$	$S_2/1$	$S_3/0$	1
S_1	$S_1/0$	$S_2/1$	$S_0/0$	$S_3/1$	1
S_2	$S_2/0$	$S_1/0$	$S_3/0$	$S_3/0$	1
S_3	$S_3/0$	$S_0/1$	$S_2/0$	$S_2/0$	1

6.1.3 已知状态图如图题 6.1.3 所示,试作出它的状态表。



图题 6.1.3

6.1.4 试画出 101 序列检测器的状态图,已知此检测器的输入、输出序列如下:

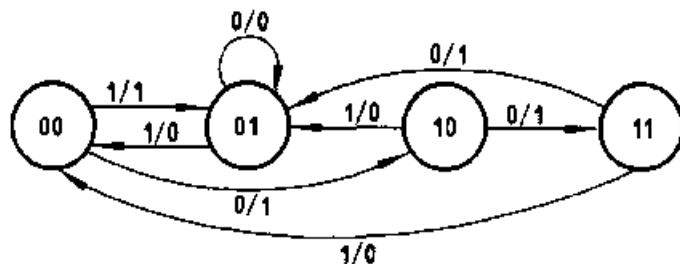
1. 输入 X:010101101

输出 Z:000101001

2. 输入 X:0101011010

输出 Z:0001000010

6.1.5 图题 6.1.5 是某时序电路的状态转换图,设电路的初始状态为 01,当序列 X = 100110(自左至右输入)时,求该电路输出 Z 的序列。



图题 6.1.5

6.1.6 已知某时序电路的状态表如表题 6.1.6 所示,试画出它的状态图。如果电路的初始状态在 S_2 , 输入信号依次是 0、1、0、1、1、1、1, 试求其相应的输出。

表题 6.1.6

现态	次态/输出	X	
		0	1
S_1		$S_1/0$	$S_2/0$
S_2		$S_3/1$	$S_4/1$
S_3		$S_2/1$	$S_5/1$
S_4		$S_4/0$	$S_3/0$
S_5		$S_2/1$	$S_1/1$

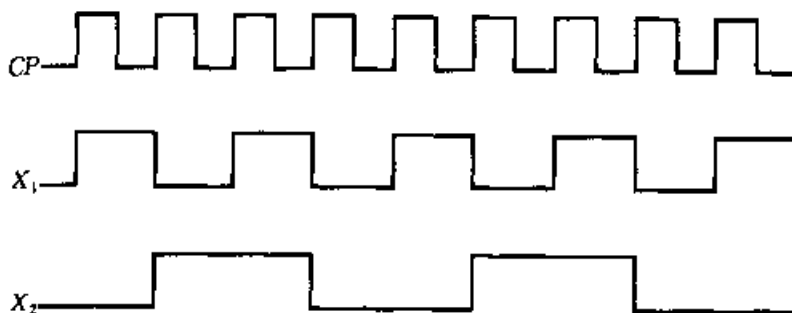
6.1.7 已知某同步时序电路含有两个正边沿 D 触发器, 其驱动方程、输出方程为

$$D_0 = X_1 X_2 + X_1 Q_0^n + X_2 Q_0^n$$

$$D_1 = X_1 \oplus X_2 \oplus Q_0^n$$

$$Z = Q_1^n$$

输入信号的波形如图题 6.1.7 所示, 设电路的初始状态为 00, 试画出 Q_1 、 Q_0 的波形, 并分析其逻辑功能。

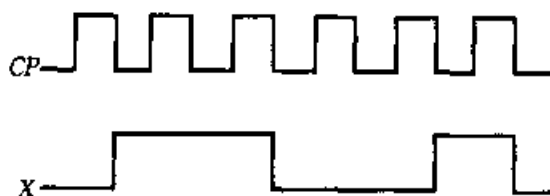


图题 6.1.7

6.1.8 已知状态表如表题 6.1.8 所示, 若电路的初始状态为 $Q_1 Q_0 = 00$, 输入信号波形如图题 6.1.8 所示, 试画出 Q_1 、 Q_0 的波形(设触发器响应于负跳变)。

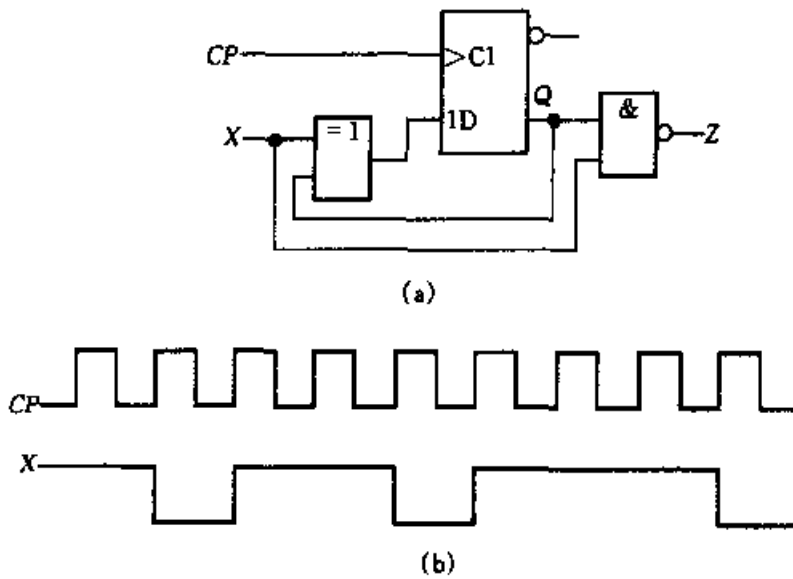
表题 6.1.8

$Q_1^{n+1} Q_0^{n+1} / Z$		X	
		0	1
$Q_1^n Q_0^n$	0 0	01/1	11/1
	0 1	10/0	10/0
	1 0	10/0	11/0
	1 1	01/1	00/1



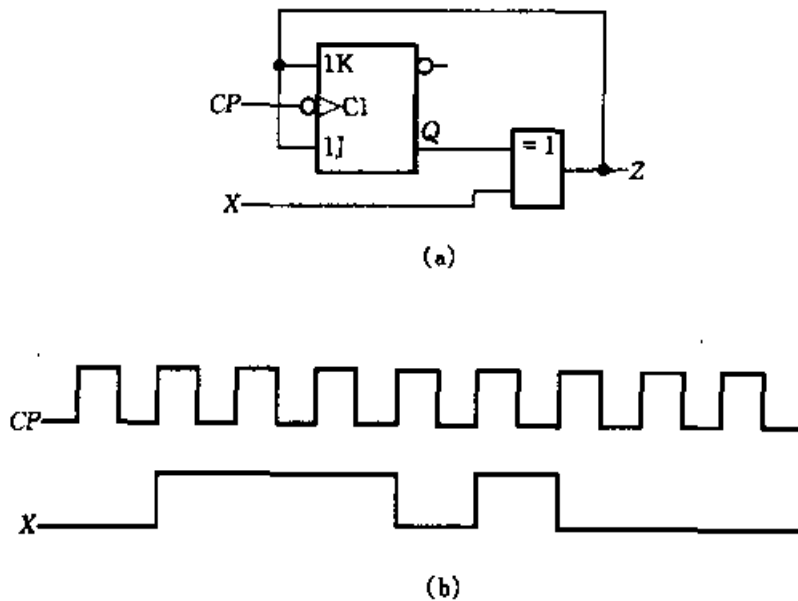
图题 6.1.8

6.2.1 试分析图题 6.2.1a 所示时序电路, 画出其状态表和状态图。设电路的初始状态为 0, 试画出在图题 6.2.1b 所示波形作用下, Q 和 Z 的波形图。



图题 6.2.1

6.2.2 试分析图题 6.2.2a 所示时序电路,画出其状态表和状态图。设电路的初始状态为 0,画出在图题 6.2.2b 所示波形作用下, Q 和 Z 的波形图。



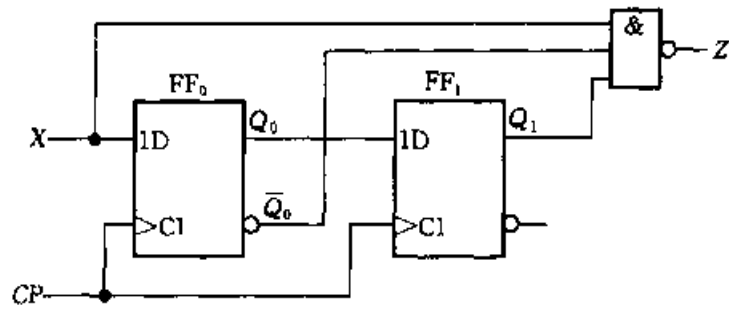
图题 6.2.2

6.2.3 试分析图题 6.2.3 所示时序电路,画出状态图。

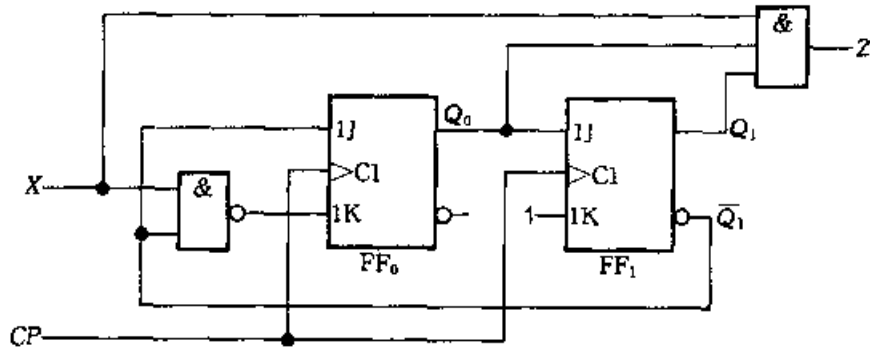
6.2.4 分析图题 6.2.4 所示电路,写出它的驱动方程、状态方程、输出方程,画出状态表和状态图。

6.2.5 分析图题 6.2.5 所示同步时序电路,写出各触发器的驱动方程、电路的状态方程和输出方程,画出状态表和状态图。

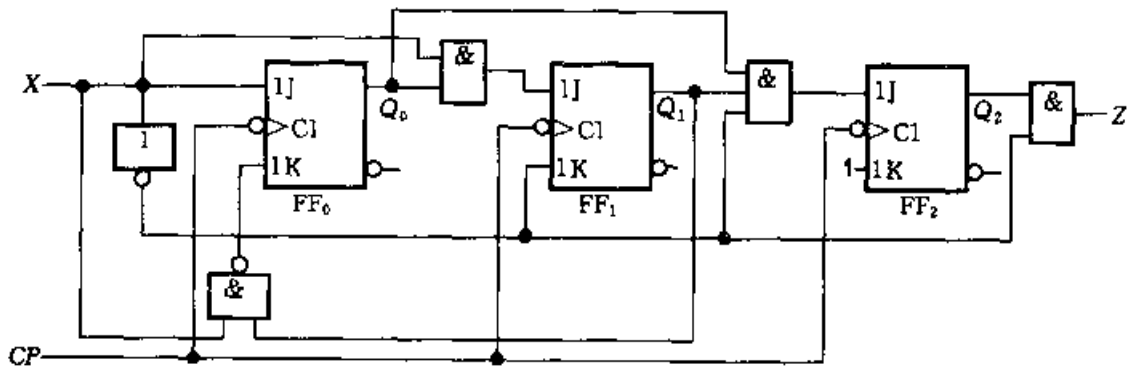
6.2.6 试画出图题 6.2.6a 所示时序电路的状态转换图,并画出对应于 CP 的 Q_1 、 Q_0 和输出 Z 的波形。设电路的初始状态为 00。



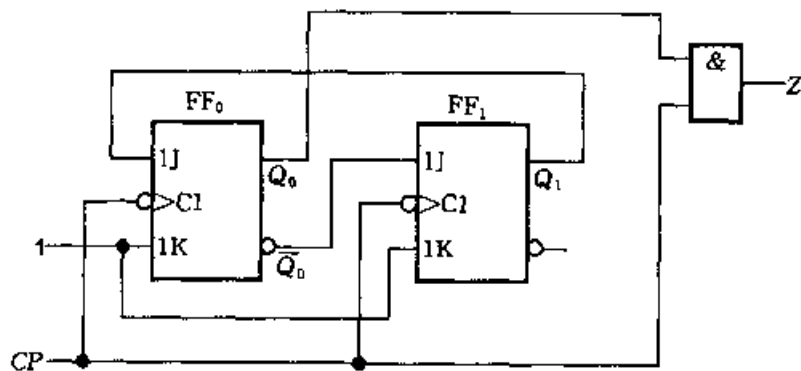
图题 6.2.3



图题 6.2.4



图题 6.2.5



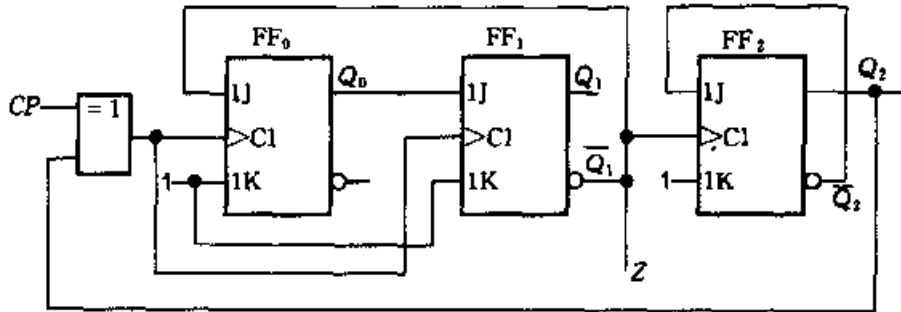
(a)



(b)

图题 6.2.6

6.2.7 一时序电路如图题 6.2.7a 所示,试画出在 CP 作用下, Q_0 、 Q_1 、 Q_2 和 Z 端的波形。设各触发器的初态均为零。



(a)

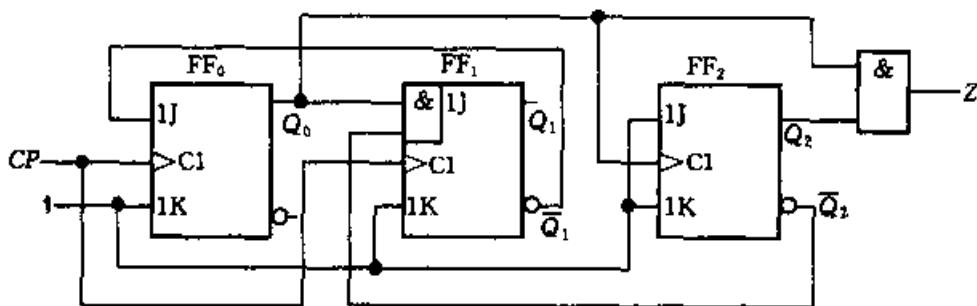


(b)

图题 6.2.7

6.2.8 分析图题 6.2.8 所示时序电路。

1. 写出各触发器 CP 信号的方程和驱动方程;
2. 写出电路的状态方程和输出方程;
3. 画出状态表及状态图;
4. 画出电路的时序图。



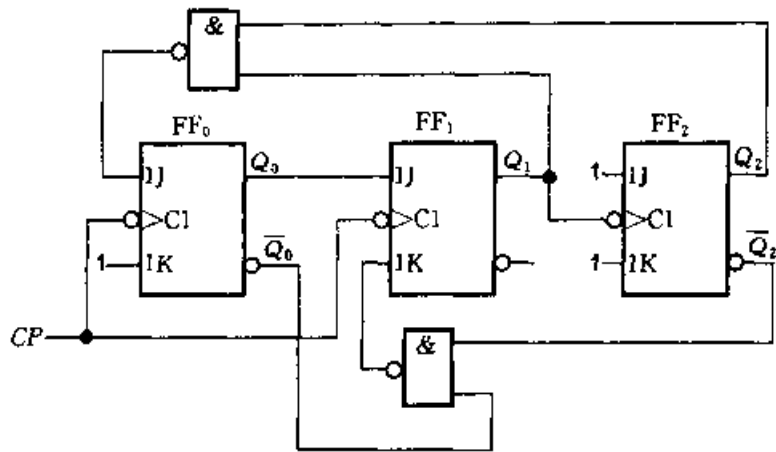
(a)



(b)

图题 6.2.8

6.2.9 试分析图题 6.2.9 所示时序电路的逻辑功能。



图题 6.2.9

6.3.1 按表题 6.3.1a 所示的最简状态表和表题 6.3.1b 所示的状态编码方案,用主从 JK 触发器设计此同步时序电路。

表题 6.3.1a

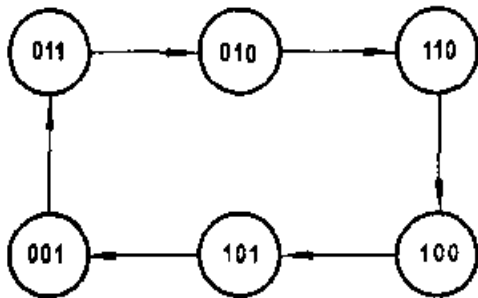
次态/输出 现态	输入	X	
		0	1
S_0		$S_1/0$	$S_3/0$
S_1		$S_2/0$	$S_0/0$
S_2		$S_3/0$	$S_1/0$
S_3		$S_0/1$	$S_2/1$

表题 6.3.1b

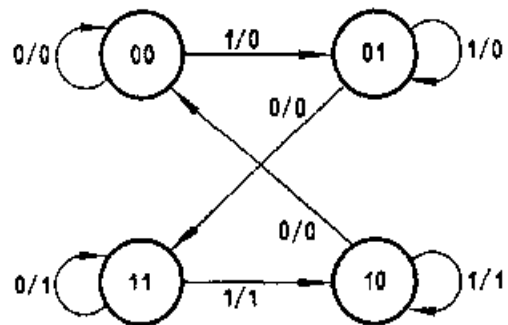
$S_0 = 00$
 $S_1 = 01$
 $S_2 = 10$
 $S_3 = 11$

6.3.2 某同步时序电路的编码状态图如图题 6.3.2 所示,试写出用 D 触发器设计此电路时的最简驱动方程。

6.3.3 试用正边沿 JK 触发器设计一同步时序电路,其状态转换图如图题 6.3.3 所示,要求电路最简。

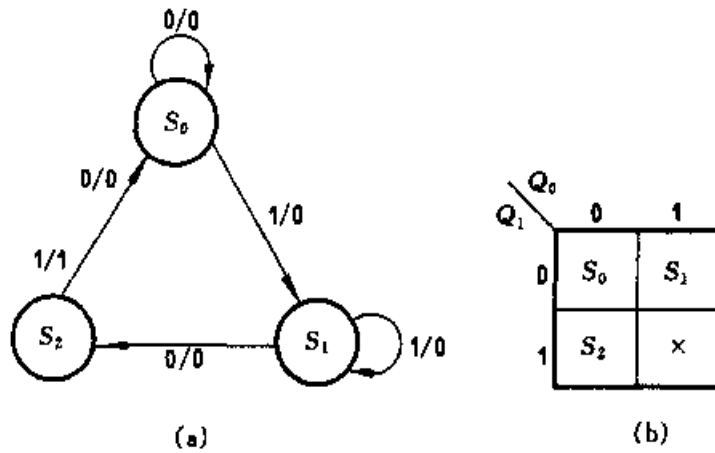


图题 6.3.2



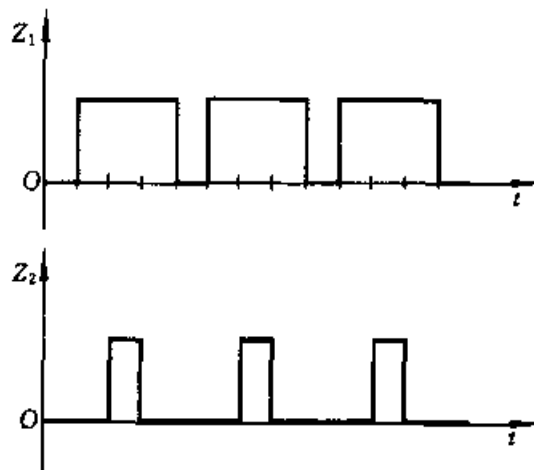
图题 6.3.3

6.3.4 试用负边沿 D 触发器设计一同步时序电路,其状态转换图如图题 6.3.4a 所示, S_0, S_1, S_2 的编码如图 b 所示。



图题 6.3.4

6.3.5 试用负边沿 JK 触发器和最少的门电路,实现图题 6.3.5 所示的 Z_1 和 Z_2 输出波形。



图题 6.3.5

6.3.6 试画出 01 序列检测器的原始状态图。

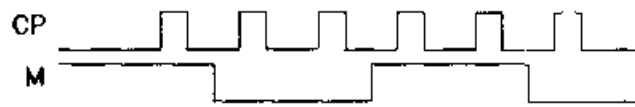
6.3.7 试用正边沿 D 触发器设计一个 1101 序列检测器,它有一个输入端和一个输出端。

* CAD 习题

C6.4.1 某时序电路的状态转换图如图题 6.1.5 所示,试用 ABEL 中状态机的结构对电路进行描述,并进行逻辑功能仿真,在如图题 C6.4.1 所示输入信号 CP 和 M 作用下,给出电路 Q_1 、 Q_0 和 Z 的输出波形。假定电路的初始状态为 01,并设 Q_1 、 Q_0 响应于 CP 的上升沿。

C6.4.2 试用 ISP Synario 软件对图题 6.2.10 所示电路进行逻辑功能仿真,给出仿真波形图,并说明电路的逻辑功能。

C6.4.3 图题 6.3.2 给出了最简状态表和状态编码方案,试用 ABEL 语言进行描述,



图题 C6.4.1 电路的输入信号波形图

并给出逻辑功能仿真波形。

7 常用时序逻辑功能器件

引言 本章介绍在计算机和其他数字系统中广泛应用的两种时序逻辑功能器件——计数器和寄存器。计数器的基本功能是统计时钟脉冲的个数,即实现计数操作,也可用于分频、定时、产生节拍脉冲等。寄存器的基本功能是存储或传输用二进制数码表示的数据或信息,即完成代码的寄存、移位、传输操作。

7.1 计数器

计数器是数字系统中用得较多的基本逻辑器件。它不仅能记录输入时钟脉冲的个数,还可以实现分频、定时、产生节拍脉冲和脉冲序列等。例如,计算机中的时序发生器、分频器、指令计数器等都要使用计数器。

计数器的种类很多。按时钟脉冲输入方式的不同,可分为同步计数器和异步计数器;按进位体制的不同,可分为二进制计数器和非二进制计数器;按计数过程中数字增减趋势的不同,可分为加计数器、减计数器和可逆计数器。

7.1.1 二进制计数器

1. 二进制异步计数器

(1) 二进制异步加计数器

图 7.1.1 是由 3 个上升沿触发的 D 触发器组成的 3 位二进制异步加计数器。图中各个触发器的 \overline{Q} 输出端与该触发器的 D 输入端相连(即 $D_i = \overline{Q}_i$),把 D 触发器转换成计数型触发器 T 。同时,各 \overline{Q} 端又与相邻高 1 位触发器的时钟脉冲输入端相连。计数脉冲 CP 加至触发器 FF_0 的时钟脉冲输入端,因而每输入一个计数脉冲, FF_0 就翻转一次。当 Q_0 由 1 变 0, \overline{Q}_0 由 0 变 1(Q_0 的进位信号)时, FF_1 翻转。当 Q_1 由 1 变 0, \overline{Q}_1 由 0 变 1(Q_1 的进位信号)时, FF_2 翻转。显然,这是一个异步时序电路,分析其工作过程,不难得到其状态图和时序图,它们分别如图 7.1.2 和图 7.1.3 所示。其中虚线是考虑触发器的传输延迟时间 t_{pd} 后的波形。

由状态图可以清楚地看到,从初态 000(由清零脉冲所置)开始,每输入一个

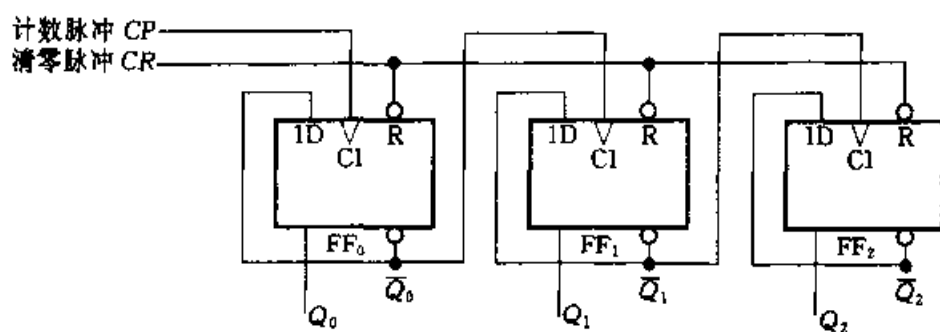


图 7.1.1 3 位二进制异步加计数器

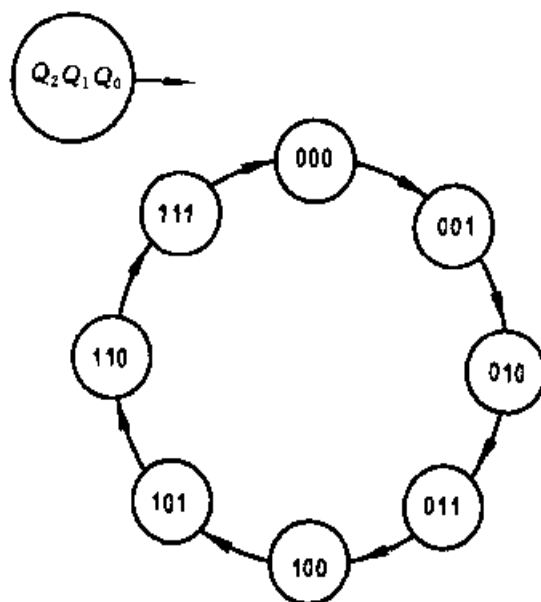


图 7.1.2 图 7.1.1 所示电路的状态图

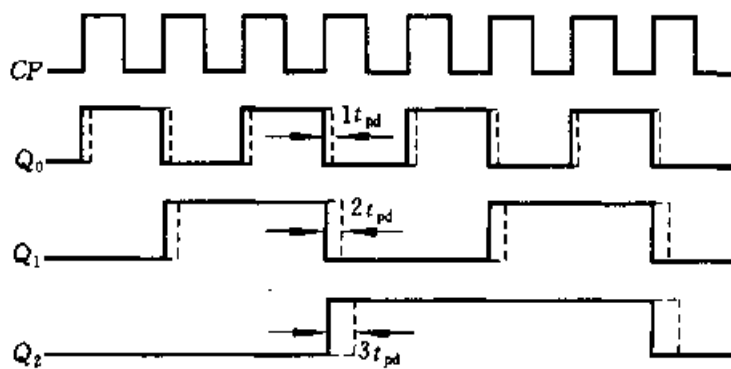


图 7.1.3 图 7.1.1 所示电路的时序图

计数脉冲,计数器的状态按二进制递增(加1),输入第8个计数脉冲后,计数器又回到000状态。因此它是 2^3 进制加计数器,也称模^①八($M=8$)加计数器。

从时序图可以清楚地看到, Q_0 、 Q_1 、 Q_2 的周期分别是计数脉冲(CP)周期的2倍、4倍、8倍,也就是说, Q_0 、 Q_1 、 Q_2 分别对 CP 波形进行了二分频、四分频、八分频,因而计数器也可作为分频器。

需要说明的是,由图7.1.3中的虚线波形可知,在考虑各触发器的传输延迟时间 t_{pd} 时,对于一个 n 位的二进制异步计数器来说,从一个计数脉冲(设为上升沿起作用)到来,到 n 个触发器都翻转稳定,需要经历的最长时间是 nt_{pd} ,为保证计数器的状态能正确反映计数脉冲的个数,下一个计数脉冲(上升沿)必须在 nt_{pd} 后到来,因此计数脉冲的最小周期 $T_{min} = nt_{pd}$ 。

(2) 二进制异步减计数器

图7.1.4是3位二进制异步减计数器的逻辑图和状态图。从初态000开始,在第一个计数脉冲作用后,触发器 FF_0 由0翻转为1(Q_0 的借位信号),此上升沿使 FF_1 也由0翻转为1(Q_1 的借位信号),这个上升沿又使 FF_2 由0翻转为1,即计数器由000变成了111状态。在这一过程中, Q_0 向 Q_1 进行了借位, Q_1 向 Q_2 进行了借位。此后,每输入1个计数脉冲,计数器的状态按二进制递减(减1)。输入第8个计数脉冲后,计数器又回到000状态,完成一次循环。因此,该计数器是 2^3 进制(模八)异步减计数器,它也同样具有分频作用。

综上所述,可对二进制异步计数器归纳出以下两点:

① n 位二进制异步计数器由 n 个处于计数工作状态(对于 D 触发器,使 $D_i = \overline{Q}_i$;对于 JK 触发器,使 $J_i = K_i = 1$)的触发器组成。各触发器之间的连接方式由加、减计数方式及触发器的触发方式决定。对于加计数器,若用上升沿触发的触发器组成,则应将低位触发器的 \overline{Q} 端与相邻高1位触发器的时钟脉冲输入端相连(即进位信号应从触发器的 \overline{Q} 端引出);若用下降沿触发的触发器组成,则应将低位触发器的 Q 端与相邻高1位触发器的时钟脉冲输入端连接。对于减计数器,各触发器间的连接方式则相反。

② 在二进制异步计数器中,高位触发器的状态翻转必须在低1位触发器产生进位信号(加计数)或借位信号(减计数)之后才能实现。故又称这种类型的计数器为串行计数器。也正因为如此,异步计数器的工作速度较低。

2. 二进制同步计数器

为了提高计数速度,可采用同步计数器,其特点是,计数脉冲同时接于各位

^① 计数器运行时总是从某个起始状态开始,依次经过所有应包含的不重复的状态后完成一次循环。我们把一次循环所包含的状态数称为计数器的“模”,用 M 表示。 M 系Modulo的字头。

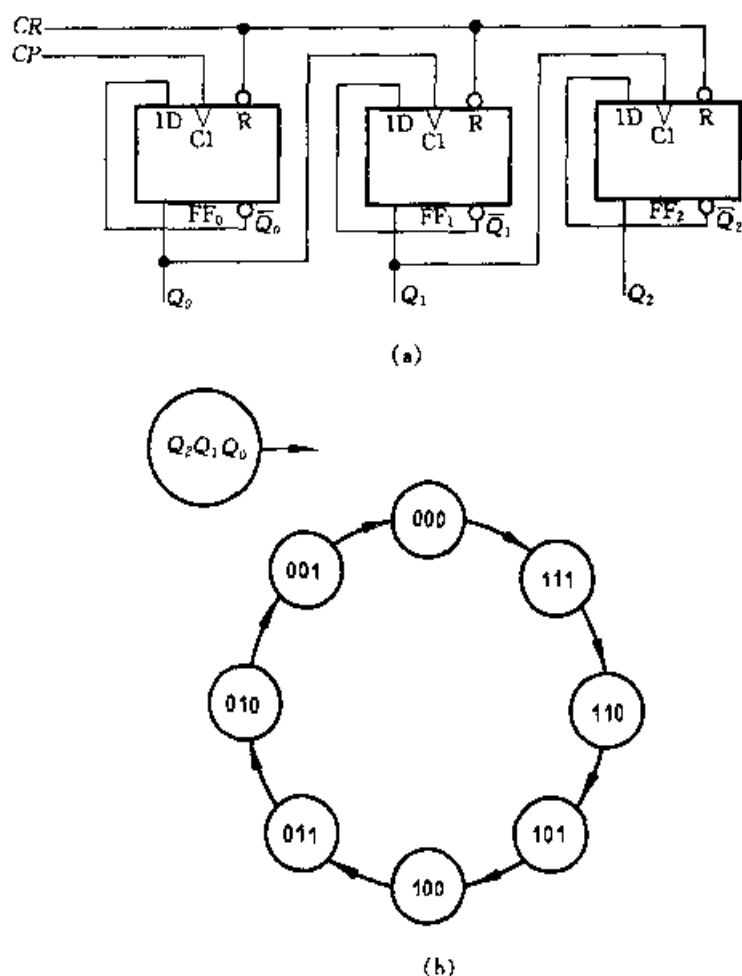


图 7.1.4 3 位二进制异步减计数器

(a) 逻辑图 (b) 状态图

触发器的时钟脉冲输入端,当计数脉冲到来时,应该翻转的触发器是同时翻转的,没有各级延迟时间的积累问题。同步计数器也可称为并行计数器。

(1) 二进制同步加计数器

图 7.1.5 是用 JK 触发器(但已令 $J=K$)组成的 4 位二进制($M=16$)同步加计数器。由图可见,各位触发器的时钟脉冲输入端接同一计数脉冲 CP ,各触发器的驱动信号分别为 $J_0=K_0=1$ 、 $J_1=K_1=Q_0$ 、 $J_2=K_2=Q_0Q_1$ 、 $J_3=K_3=Q_0Q_1Q_2$ 。根据同步时序电路的分析方法,可得到该电路的状态表,如表 7.1.1^① 所示。设从初态 000 开始,因为 $J_0=K_0=1$,所以每输入一个计数脉冲 CP ,最低位触发器 FF_0 就翻转一次,其他位的触发器 FF_i 仅在 $J_i=K_i=Q_{i-1}Q_{i-2}\cdots Q_0=1$ 的条件下,在 CP 下降沿到来时才翻转。

^① 表 7.1.1 是时序电路状态表的另一种形式。该表给出了在一系列时钟脉冲信号作用下电路状态的转换顺序。

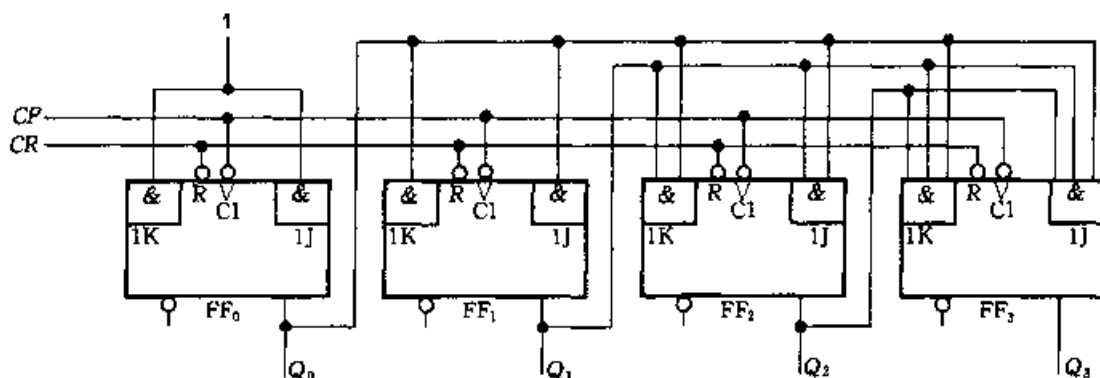


图 7.1.5 4 位二进制同步加计数器

表 7.1.1 图 7.1.5 电路的状态表

计数脉冲 CP 的顺序	电路状态				等效十进制数
	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15
16	0	0	0	0	0

图 7.1.6 是图 7.1.5 电路的时序图,其中虚线是考虑触发器的传输延迟时间 t_{pd} 后的波形。由此波形图可知,在同步计数器中,由于计数脉冲 CP 同时作用于各个触发器,所有触发器的翻转是同时进行的,都比计数脉冲 CP 的作用时间滞后一个 t_{pd} ,因此其工作速度一般要比异步计数器高。

应当指出的是,同步计数器的电路结构较异步计数器复杂,需要增加一些输入控制电路,因而其工作速度也要受这些控制电路的传输延迟时间的限制。

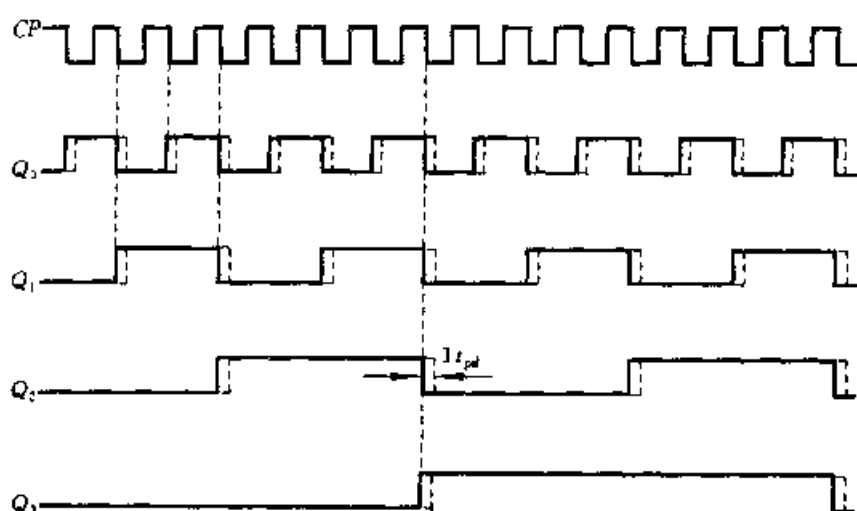


图 7.1.6 图 7.1.5 电路的时序图

如果将图 7.1.5 电路中触发器 FF_1 、 FF_2 和 FF_3 的驱动信号分别改为 $J_1 = K_1 = \overline{Q_0}$ 、 $J_2 = K_2 = \overline{Q_1} \overline{Q_0}$ 、 $J_3 = K_3 = \overline{Q_2} \overline{Q_1} \overline{Q_0}$, 即可构成 4 位二进制同步减计数器, 其工作过程请读者自行分析。

(2) 二进制同步可逆计数器

实际应用中, 有时要求一个计数器既能作加计数又能作减计数。同时兼有加和减两种计数功能的计数器称为可逆计数器。

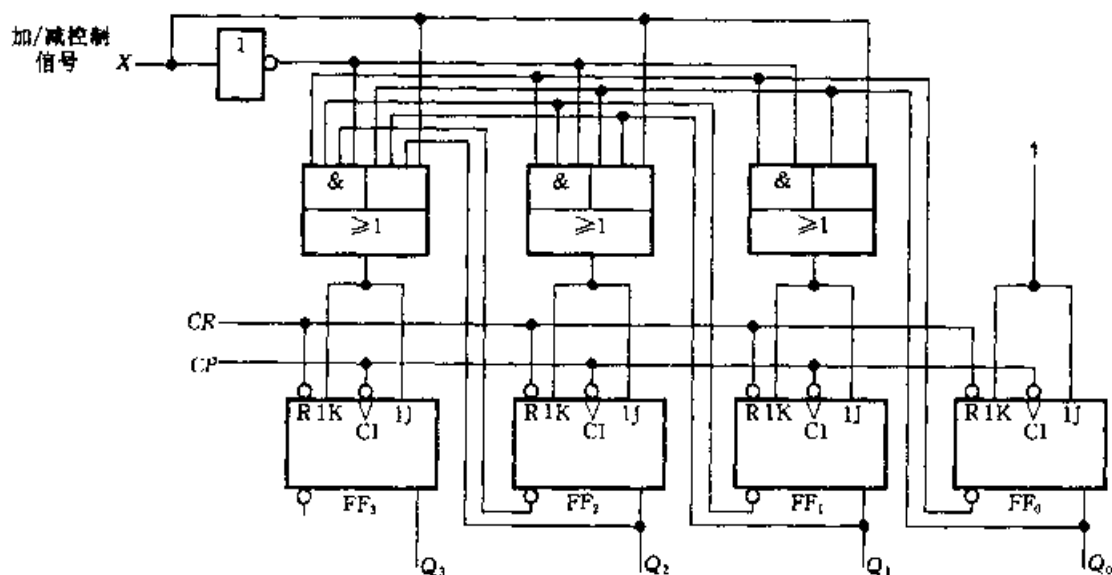


图 7.1.7 4 位二进制同步可逆计数器

4 位二进制同步可逆计数器如图 7.1.7 所示, 它是在前面介绍的 4 位二进制同步加和减计数器的基础上, 增加一控制电路构成的。由图可知, 各触发器的驱动信号分别为

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = XQ_0 + \overline{X}\overline{Q_0}$$

$$J_2 = K_2 = XQ_1Q_0 + \overline{X}\overline{Q_1}\overline{Q_0}$$

$$J_3 = K_3 = XQ_2Q_1Q_0 + \overline{X}\overline{Q_2}\overline{Q_1}\overline{Q_0}$$

当加/减控制信号 $X=1$ 时, $FF_1 \sim FF_3$ 中的各 J 、 K 端分别与低位各触发器的 Q 端接通, 进行加计数; 当 $X=0$ 时, 各 J 、 K 端分别与低位各触发器的 \overline{Q} 端接通, 进行减计数, 实现了可逆计数器的功能。

7.1.2 非二进制计数器

在非二进制计数器中, 最常用的是十进制计数器, 其他进制的计数器习惯上被称为任意进制计数器。非二进制计数器也有同步和异步, 加、减和可逆计数器等各种类型。这里不再一一介绍, 仅以 8421 码十进制同步计数器为例, 介绍非二进制同步计数器的设计问题。

例 7.1.1 用 D 触发器设计一个 8421 码十进制同步加计数器。

解: 由于计数器在工作过程中所经历的状态个数、状态转换关系及状态编码等都是明确的, 因此其设计过程较其他时序电路简单, 同步计数器更加简单一些。

(1) 列出状态表和驱动表, 如表 7.1.2 所示。

表 7.1.2 8421 码十进制同步计数器的状态表和驱动表

计数脉冲 CP 的顺序	现 态				次 态				驱动信号			
	Q_3	Q_2	Q_1	Q_0	Q_3^{*+1}	Q_2^{*+1}	Q_1^{*+1}	Q_0^{*+1}	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	1	0	0	0	1	0
2	0	0	1	0	0	0	1	1	0	0	1	1
3	0	0	1	1	0	1	0	0	0	1	0	0
4	0	1	0	0	0	1	0	1	0	1	0	1
5	0	1	0	1	0	1	1	0	0	1	1	0
6	0	1	1	0	0	1	1	1	0	1	1	1
7	0	1	1	1	1	0	0	0	1	0	0	0
8	1	0	0	0	1	0	0	1	1	0	0	1
9	1	0	0	1	0	0	0	0	0	0	0	0
	1	0	1	0	×	×	×	×	×	×	×	×
	1	0	1	1	×	×	×	×	×	×	×	×
	1	1	0	0	×	×	×	×	×	×	×	×
	1	1	0	1	×	×	×	×	×	×	×	×
	1	1	1	0	×	×	×	×	×	×	×	×
	1	1	1	1	×	×	×	×	×	×	×	×

1 个十进制计数器有 10 个状态, 至少需要 4 位触发器组成。4 位触发器共

有 16 个组合状态(0000~1111),其中有 6 个状态(1010~1111)在 8421 码十进制计数器中是无效的组态,但我们仍将这 6 个状态列在状态表中,并将其对应的次态全都以无关项 \times 表示,以便于得到驱动信号的最简表达式。

(2) 用卡诺图法化简,如图 7.1.8 所示。求得各位 D 触发器的驱动信号的表达式。

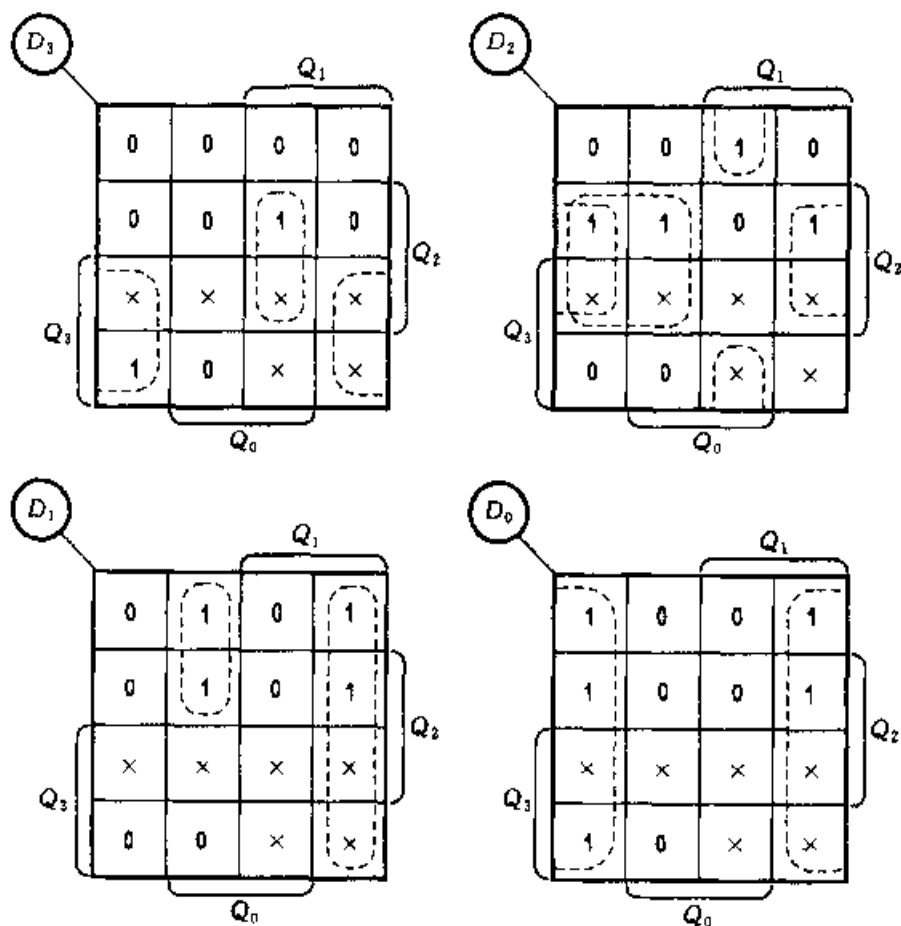


图 7.1.8 例 7.1.1 的卡诺图

$$D_3 = Q_3 \bar{Q}_0 + Q_2 Q_1 Q_0 \quad D_2 = Q_2 \bar{Q}_1 + Q_2 \bar{Q}_0 + \bar{Q}_2 Q_1 Q_0$$

$$D_1 = Q_1 \bar{Q}_0 + \bar{Q}_3 \bar{Q}_1 Q_0 \quad D_0 = \bar{Q}_0$$

(3) 画出该计数器的逻辑电路图,如图 7.1.9 所示。

(4) 画出完整的状态图,检查设计的计数器能否自启动。

对于非二进制计数器来说,当有效状态数 M 和所用触发器的位数 n 之间存在 $M < 2^n$ 关系时,必然存在 $2^n - M$ 个多余状态,即无效状态(如本例中的 1010~1111 六个状态)。在实际工作中,当由于某种原因(如干扰信号等)使计数器进入某一无效状态时,要求计数器能够自动地由无效状态返回到有效状态的循环中来,这就是说,要求设计的计数器具有自启动能力。检查自启动的方法

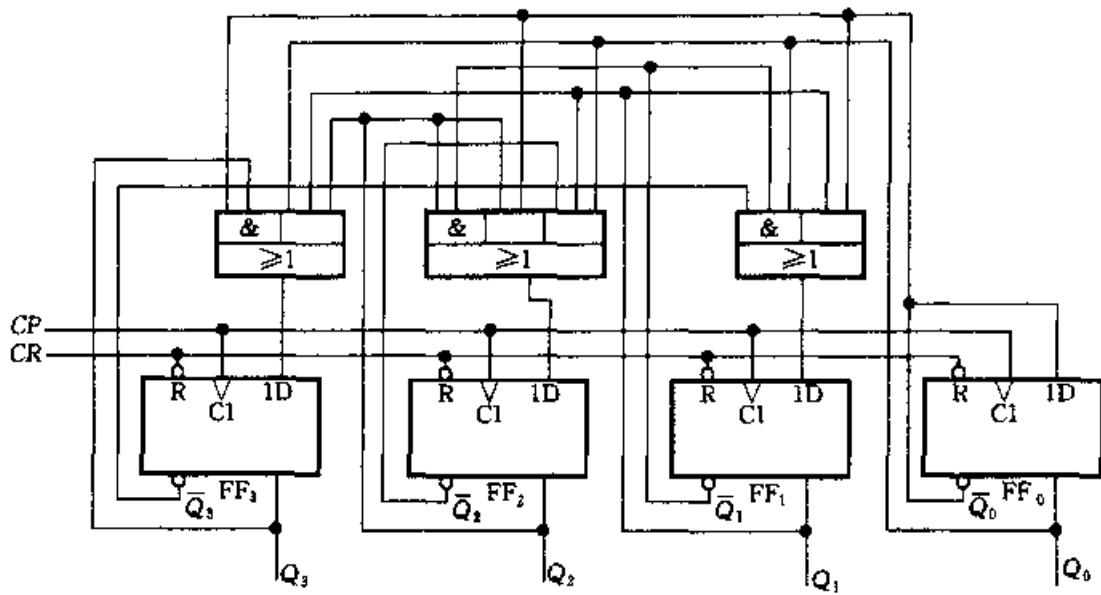


图 7.1.9 例 7.1.1 的逻辑电路

是:画出包括无效状态在内的完整的状态图,看能否从无效状态进入有效状态。由图 7.1.10 的状态图可知,图 7.1.9 电路能够自启动。

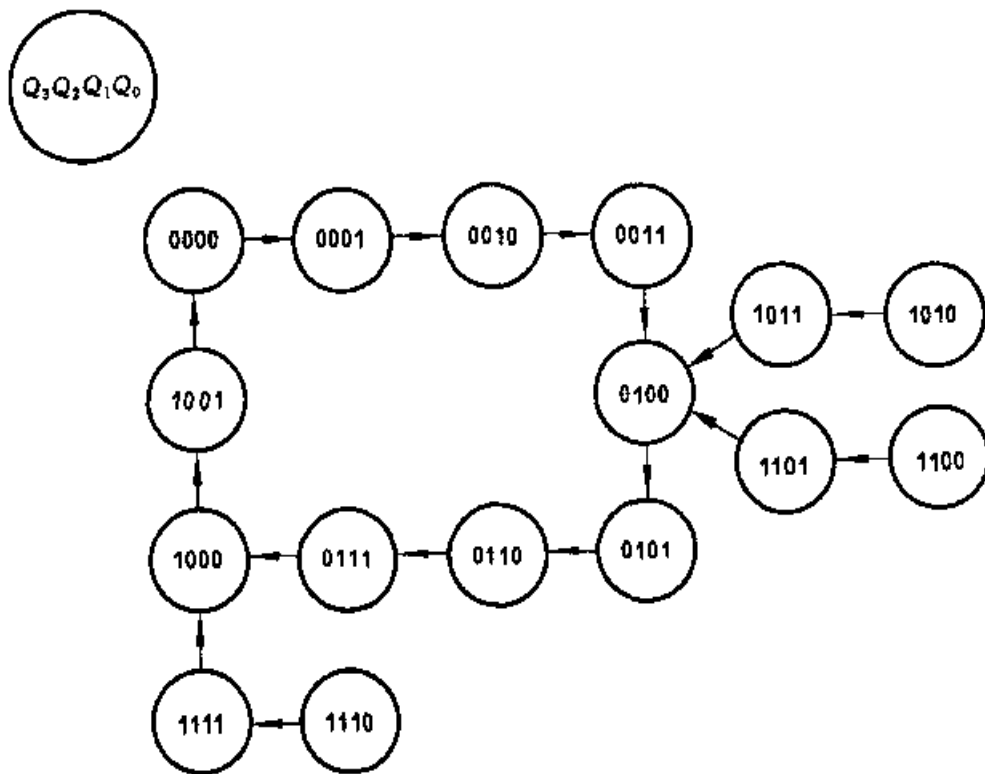


图 7.1.10 图 7.1.9 电路的状态图

7.1.3 集成计数器

目前,集成计数器在一些简单小型数字系统中仍被广泛应用,因为它们具有体积小、功耗低、功能灵活等优点。集成计数器的类型很多,表 7.1.3 列举了若干集成计数器产品。限于篇幅,本节仅介绍其中几个较典型产品的功能和应用。

表 7.1.3 几种集成计数器

CP 脉冲引入方式	型 号	计数模式	清零方式	预置数方式
同步	74161	4 位二进制加法	异步(低电平)	同步
	74HC161	4 位二进制加法	异步(低电平)	同步
	74HCT161	4 位二进制加法	异步(低电平)	同步
	74LS191	单时钟 4 位二进制可逆	无	异步
	74LS193	双时钟 4 位二进制可逆	异步(高电平)	异步
	74160	十进制加法	异步(低电平)	同步
	74LS190	单时钟十进制可逆	无	异步
异步	74LS293	双时钟 4 位二进制加法	异步	无
	74LS290	二-五-十进制加法	异步	异步

1. 集成计数器 74161、74LS193、74LS290

(1) 74161 的功能

74161 是 4 位二进制同步加计数器。图 7.1.11a、b 分别是它的逻辑电路图和引脚图,其中 R_D 是异步清零端, LD 是预置数控制端, A 、 B 、 C 、 D 是预置数据输入端, EP 和 ET 是计数使能(控制)端, $RCO(=ET \cdot Q_A \cdot Q_B \cdot Q_C \cdot Q_D)$ 是进位输出端,它的设置为多片集成计数器的级联提供了方便。

表 7.1.4 是 74161 的功能表。由表可知,74161 具有以下功能:

① 异步清零 当 $R_D = 0$ 时,不管其他输入端的状态如何(包括时钟信号 CP),计数器输出将被直接置零,称为异步清零。

表 7.1.4 74161 的功能表

清零 R_D	预置 LD	使能		时钟 CP	预置数据输入				输 出			
		EP	ET		A	B	C	D	Q_A	Q_B	Q_C	Q_D
L	x	x	x	x	x	x	x	x	L	L	L	L
H	L	x	x	\downarrow	A	B	C	D	A	B	C	D
H	H	L	x	x	x	x	x	x	保			持
H	H	x	L	x	x	x	x	x	保			持
H	H	H	H	\downarrow	x	x	x	x	计			数

② 同步并行预置数 在 $R_D = 1$ 的条件下,当 $LD = 0$ 、且有时钟脉冲 CP 的

上升沿作用时, A 、 B 、 C 、 D 输入端的数据将分别被 $Q_A \sim Q_D$ 所接收。由于这个置数操作要与 CP 上升沿同步, 且 $A \sim D$ 的数据同时置入计数器, 所以称为同步并行预置。

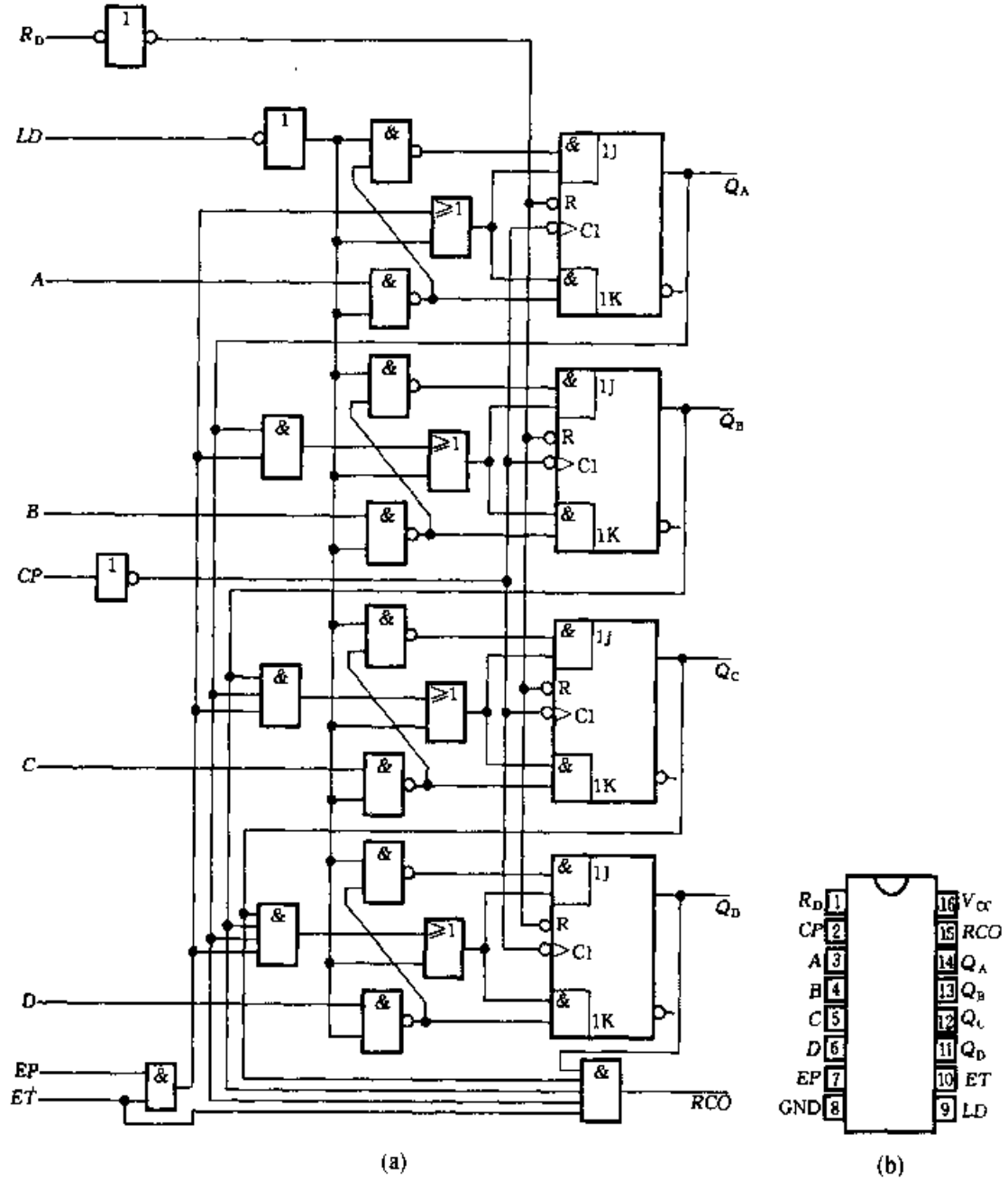


图 7.1.11 74161 的逻辑电路图和引脚图

(a) 逻辑电路图 (b) 引脚图

③ 保持 在 $R_D = LD = 1$ 的条件下, 当 $ET \cdot EP = 0$, 即两个计数使能端中有 0 时, 不管有无 CP 脉冲作用, 计数器都将保持原有状态不变(停止计数)。需要说明的是, 当 $EP = 0, ET = 1$ 时, 进位输出 RCO 也保持不变; 而当 $ET = 0$ 时, 不管 EP 状态如何, 进位输出 $RCO = 0$ 。

④ 计数 当 $R_D = LD = EP = ET = 1$ 时, 74161 处于计数状态, 其状态表与表 7.1.1 相同。

图 7.1.12 是 74161 的时序图。由时序图可以清楚地看到 74161 的功能和各控制信号间的时序关系。

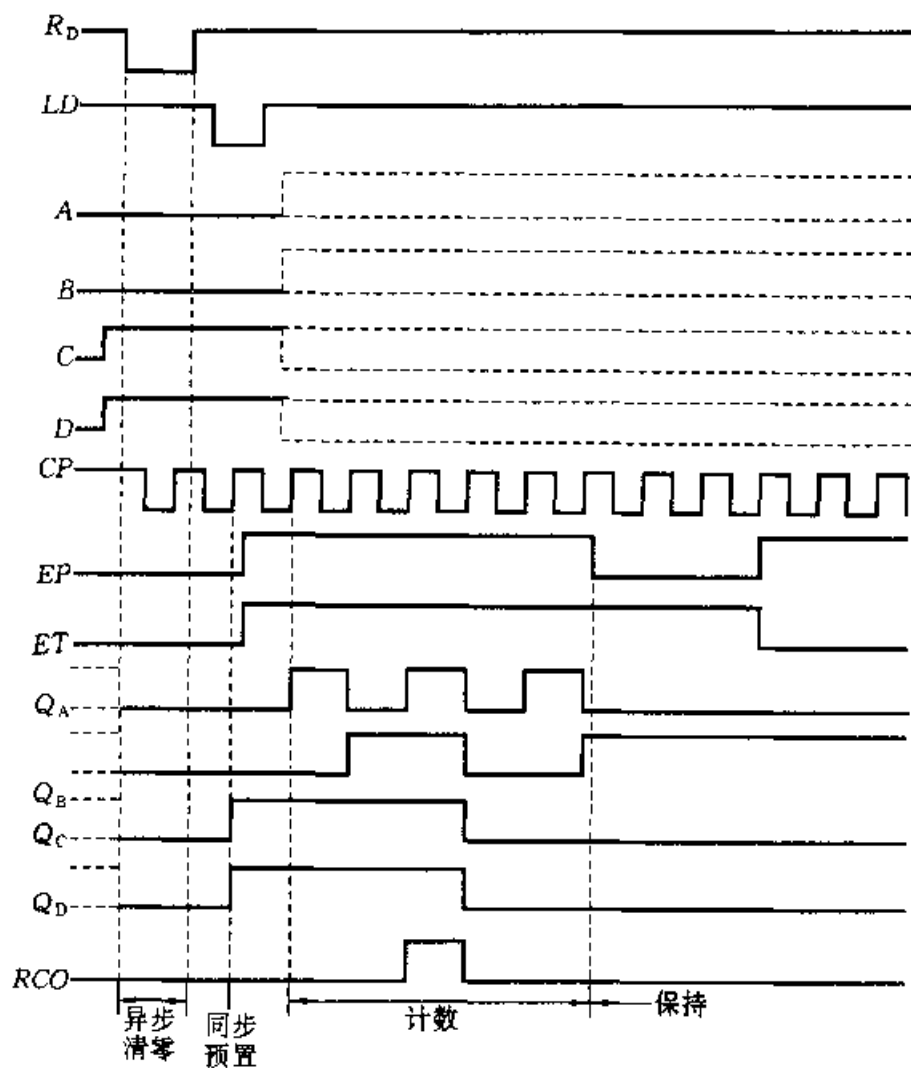


图 7.1.12 74161 的时序图

由图可知, 首先加入一清零信号 $R_D = 0$, 使各触发器的状态为 0, 即计数器清零。 R_D 变为 1 后, 加入一置数信号 $LD = 0$, 该信号需维持到下一个时钟脉冲的正跳变到来后。在这个置数信号和时钟脉冲上升沿的共同作用下, 各触发器的输出状态与预置的输入数据相同 (图中为 $DCBA = 1100$), 这就是预置操作。接着是 $EP = ET = 1$, 在此期间 74161 处于计数状态。这里是从预置的 $DCBA = 1100$ 开始计数, 直到 $EP = 0$, $ET = 1$, 计数状态结束, 转为保持状态, 计数器输出保持 EP 负跳变前的状态不变, 图中为 $Q_D Q_C Q_B Q_A = 0010$, $RCO = 0$ 。

高速 CMOS 集成器件 74HC161、74HCT161 的逻辑功能、外形和尺寸、引脚排列顺序等与 74161 完全相同。

(2) 74LS193 的功能

74LS193 是双时钟 4 位二进制同步可逆计数器。图 7.1.13a、b 分别是它的逻辑电路图和引脚图,表 7.1.5 是它的功能表。74LS193 的特点是有两个时钟

表 7.1.5 74LS193 的功能表

清零 R_D	预置 LD	时钟		预置数据输入				输出			
		CP_U	CP_D	A	B	C	D	Q_A	Q_B	Q_C	Q_D
H	×	×	×	×	×	×	×	L	L	L	L
L	L	×	×	A	B	C	D	A	B	C	D
L	H	\downarrow	H	×	×	×	×	加计数			
L	H	H	\downarrow	×	×	×	×	减计数			

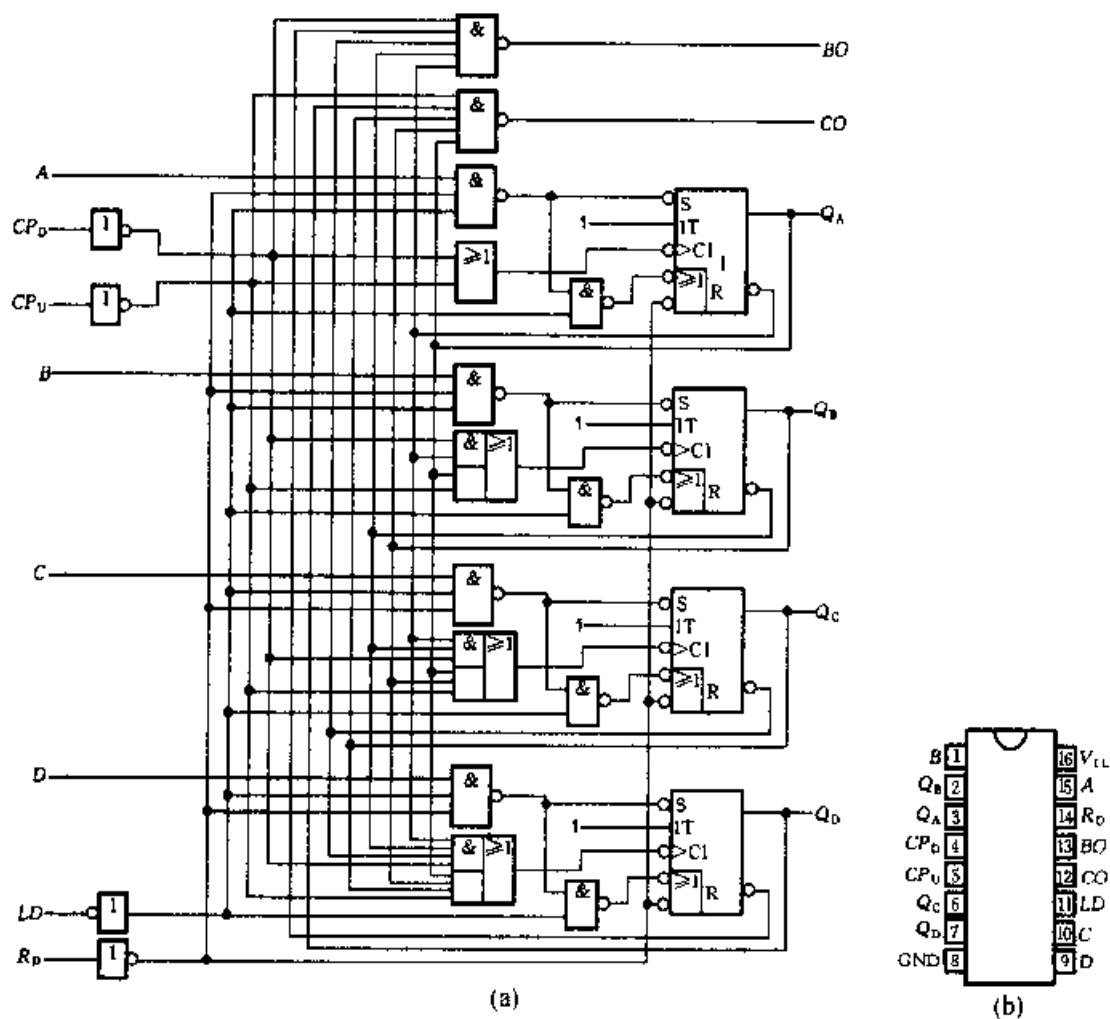


图 7.1.13 双时钟 4 位二进制同步可逆计数器 74LS193

(a) 逻辑电路图 (b) 引脚图

脉冲(计数脉冲)输入端 CP_C 和 CP_D 。在 $R_D = 0$ 、 $LD = 1$ 的条件下,作加计数时,令 $CP_D = 1$,计数脉冲从 CP_C 输入;作减计数时,令 $CP_C = 1$,计数脉冲从 CP_D 输入。此外,74LS193 还具有异步清零和异步预置数的功能。当清零信号 $R_D = 1$ 时,不管时钟脉冲的状态如何,计数器的输出将被直接置零;当 $R_D = 0$ 、 $LD = 0$ 时,不管时钟脉冲的状态如何,将立即把预置数据输入端 A 、 B 、 C 、 D 的状态置入计数器的 Q_A 、 Q_B 、 Q_C 、 Q_D 端,称为异步预置数。

74HC193、74HCT193 的逻辑功能及引脚图与 74LS193 完全相同。

(3) 74LS290 的功能

74LS290 是异步十进制计数器,其逻辑电路图和引脚图如图 7.1.14a、b 所

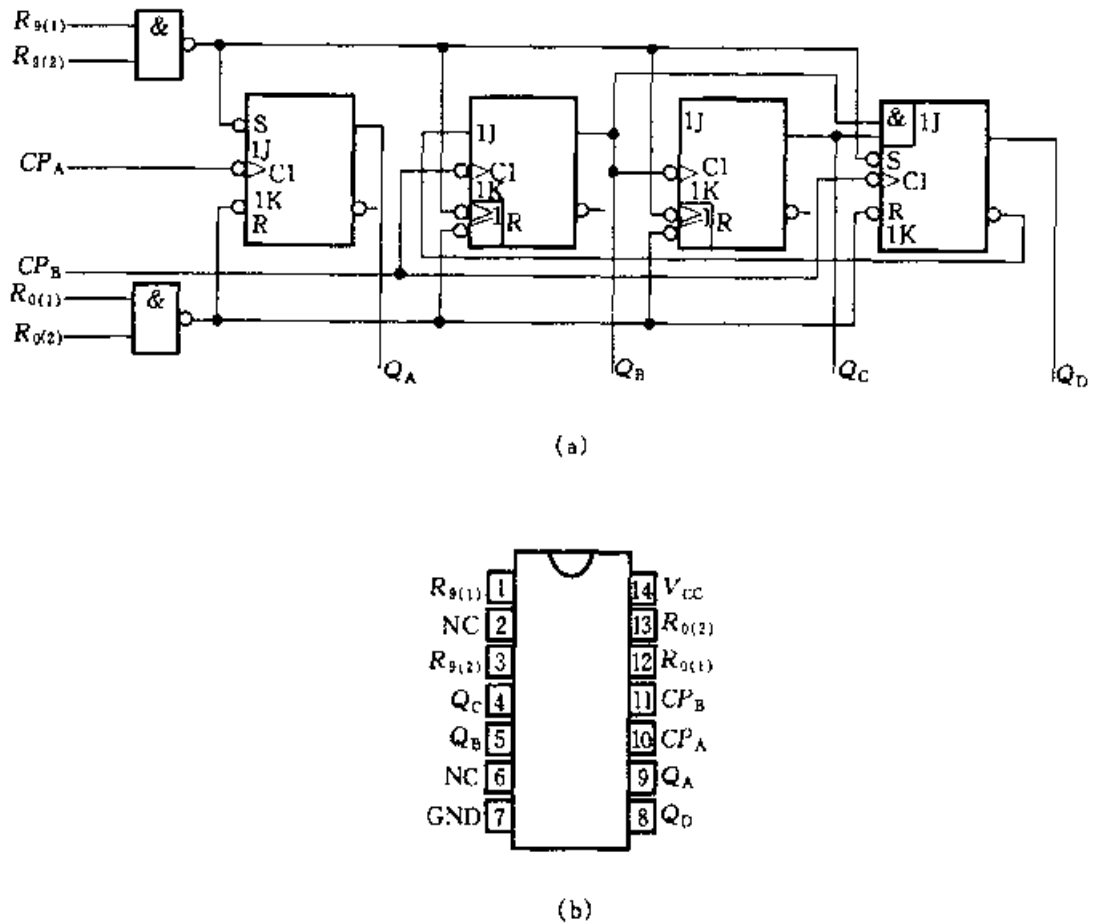


图 7.1.14 异步十进制计数器 74LS290

(a) 逻辑电路图 (b) 引脚图

示,它由 1 个 1 位二进制计数器和 1 个异步五进制计数器组成。如果计数脉冲由 CP_A 端输入,输出由 Q_A 端引出,即得二进制计数器;如果计数脉冲由 CP_B 端输入,输出由 $Q_B \sim Q_D$ 引出,即是五进制计数器;如果将 Q_A 与 CP_B 相连,计数脉冲由 CP_A 输入,输出由 $Q_A \sim Q_D$ 引出,即得 8421 码十进制计数器。因此,又称

此电路为二-五-十进制计数器。

表 7.1.6 是 74LS290 的功能表。由表可以看出,当复位输入 $R_{0(1)} = R_{0(2)} = 1$,且置位输入 $R_{9(1)} = R_{9(2)} = 0$ 时,74LS290 的输出被直接置零;只要置位输入 $R_{9(1)} = R_{9(2)} = 1$,则 74LS290 的输出将被直接置 9,即 $Q_D Q_C Q_B Q_A = 1001$;只有同时满足 $R_{0(1)} = R_{0(2)} = 0$ 和 $R_{9(1)} = R_{9(2)} = 0$ 时,才能在计数脉冲(下降沿)作用下实现二-五-十进制加计数。

表 7.1.6 74LS290 的功能表

复位输入		置位输入		时钟	输出			
$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	CP	Q_A	Q_B	Q_C	Q_D
H	H	L	×	×	L	L	L	L
H	H	×	L	×	L	L	L	L
×	×	H	H	×	H	L	L	H
L	×	L	×	↓		计	数	
L	×	×	L	↓		计	数	
×	L	L	×	↓		计	数	
×	L	×	L	↓		计	数	

74HC290、74HCT290 的逻辑功能和引脚图与 74LS290 完全相同。

2. 用集成计数器构成任意进制计数器

尽管集成计数器的品种很多,但也不可能任一进制的计数器都有其对应的集成产品。在需要用到它们时,只能用现有的成品计数器外加适当的电路连接而成。

用现有的 M 进制集成计数器构成 N 进制计数器时,如果 $M > N$,则只需一片 M 进制计数器;如果 $M < N$,则要用多片 M 进制计数器。下面结合例题分别介绍这两种情况的实现方法。

例 7.1.2 用 74161 构成九进制加计数器。

解: 九($N=9$)进制计数器有 9 个状态,而 74161 在计数过程中有 16($M=16$)个状态,因此属于 $M > N$ 的情况。此时必须设法跳过 $M - N (=16 - 9 = 7)$ 个状态。通常用两种方法实现,即反馈清零法和反馈置数法。

(1) 反馈清零法

反馈清零法适用于有清零输入端的集成计数器。74161 具有异步清零功能,在其计数过程中,不管它的输出处于哪一状态,只要在异步清零输入端加一低电平电压,使 $R_D = 0$,74161 的输出会立即从那个状态回到 0000 状态。清零信号($R_D = 0$)消失后,74161 又从 0000 状态开始重新计数。

图 7.1.15a 所示的九进制计数器,就是借助 74161 的异步清零功能实现的。图 7.1.15b 是该九进制计数器的主循环状态图。由图可知,74161 从 0000 状态开始计数,当输入第九个 CP 脉冲(上升沿)时,输出 $Q_D Q_C Q_B Q_A = 1001$,通过

与非门译码后,反馈给 R_D 端一个清零信号,立即使 $Q_D Q_C Q_B Q_A$ 返回 **0000** 状态,接着, R_D 端的清零信号也随之消失,74161 重新从 **0000** 状态开始新的计数周期。要说明的是,此电路一进入 **1001** 状态后,立即又被置成 **0000** 状态,即 **1001** 状态仅在极短的瞬间出现,因此,在主循环状态图中用虚线表示。这样就跳过了 **1001**~**1111** 七个状态,获得了九进制计数器。

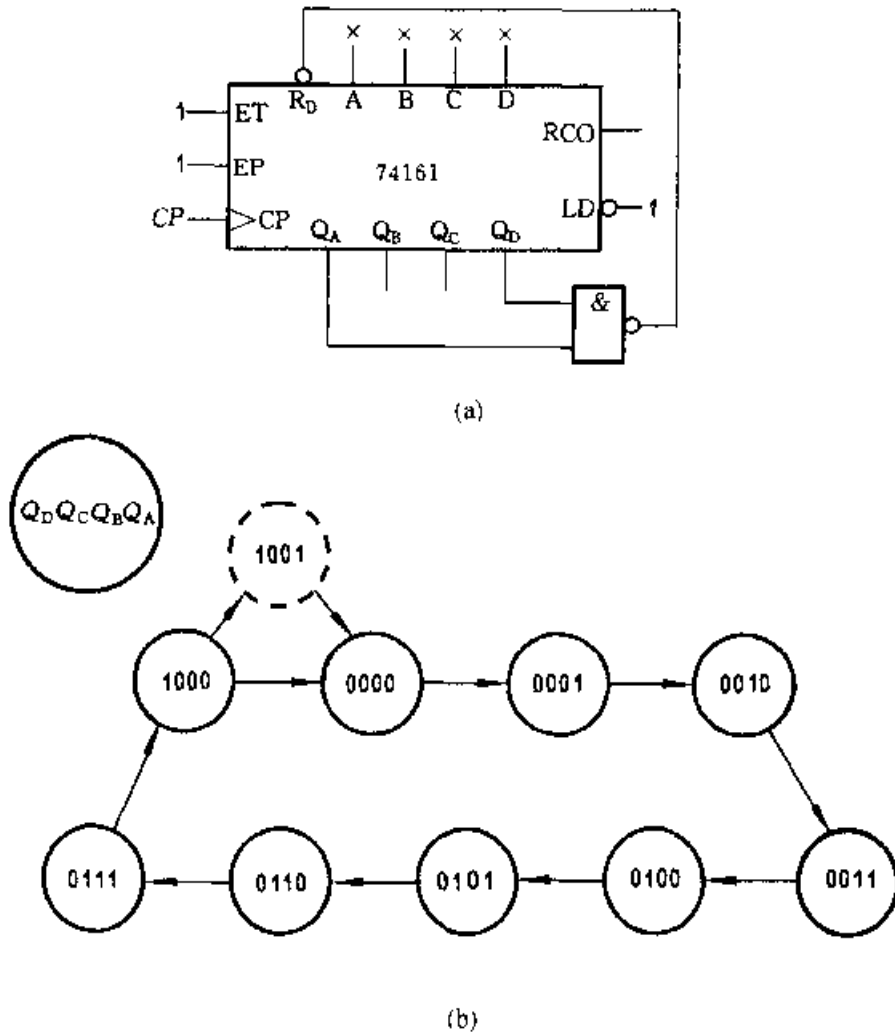


图 7.1.15 用反馈清零法将 74161 接成九进制计数器

(a) 逻辑电路图 (b) 主循环状态图

具有同步清零功能的 M 进制集成计数器也可用反馈清零法构成 N 进制计数器。这里不再举例,读者可自行分析两者的差异。

(2) 反馈置数法

反馈置数法适用于具有预置数功能的集成计数器。对于具有同步预置数功能的计数器而言,在其计数过程中,可以将它输出的任何一个状态通过译码,产生一个预置数控制信号反馈至预置数控制端,在下一个 CP 脉冲作用后,计数器

就会把预置数输入端 A 、 B 、 C 、 D 的状态置入输出端。预置数控制信号消失后,计数器就从被置入的状态开始重新计数。

图 7.1.16a 和图 7.1.17 都是借助同步预置数功能,采用反馈置数法,用 74161 构成九进制加计数器的。其中图 7.1.16a 的接法是把输出 $Q_D Q_C Q_B Q_A = 1000$ 状态译码产生预置数控制信号 0,反馈至 LD 端,在下一个 CP 脉冲的上升沿到达时置入 0000 状态。图 7.1.16b 是图 7.1.16a 电路的主循环状态图。其中 $0001 \sim 1000$ 这 8 个状态是 74161 进行加 1 计数实现的, 0000 是由反馈(同步)置数得到的。由此可以推知,在图 7.1.16a 中,反馈置数操作可在 74161 计数循环状态 ($0000 \sim 1111$) 中的任何一个状态下进行。例如可将 $Q_D Q_C Q_B Q_A = 1111$ 状态的译码信号加至 LD 端,这时预置数据输入端应为 $0111 (= 1111 \sim 1000)$ 状态。

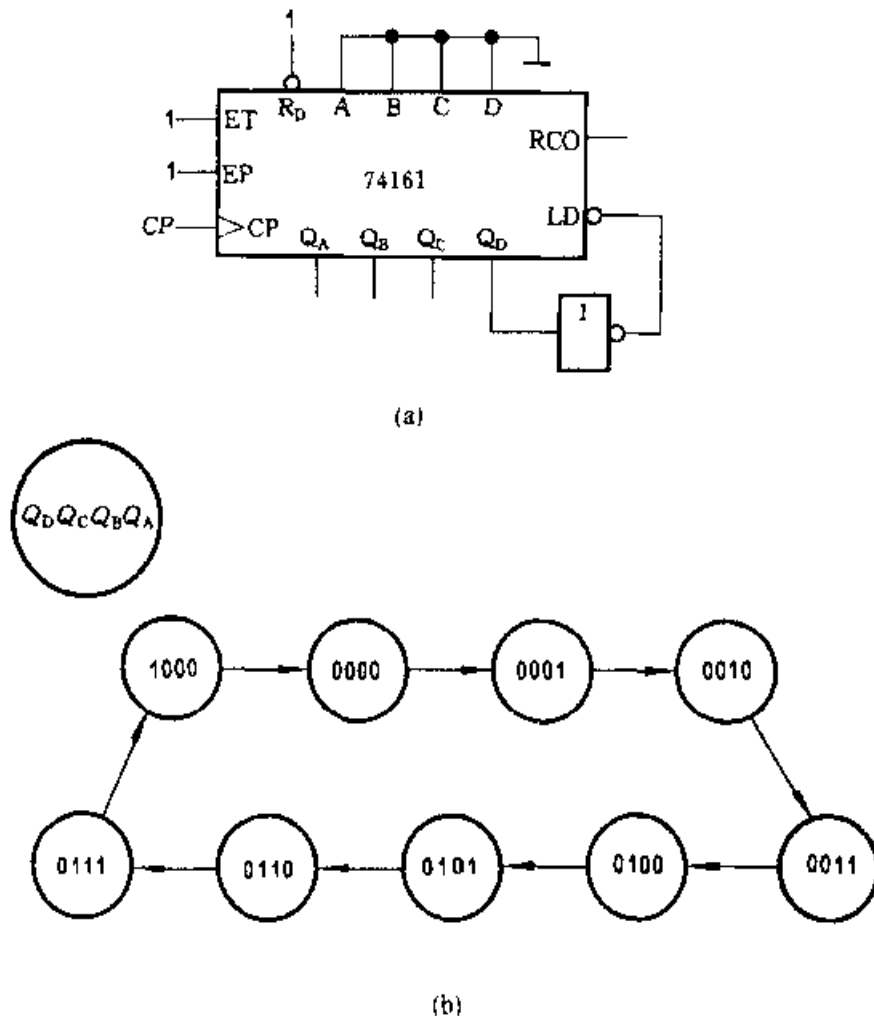


图 7.1.16 用反馈置数法将 74161 接成九进制计数器

(a) 逻辑电路图 (b) 主循环状态图

图 7.1.17 电路的接法是将 74161 计数到 1111 状态时产生的进位信号译码后,反馈到预置数控制端。预置数据输入端置成 0111 状态。该电路从 0111 状态

开始加1计数,输入第8个CP脉冲后到达1111状态,此时 $RCO = ET \cdot Q_D \cdot Q_C \cdot Q_B \cdot Q_A = 1$, $LD = 0$,在第9个CP脉冲作用后, $Q_D Q_C Q_B Q_A$ 被置成0111状态,同时使 $RCO = 0$, $LD = 1$ 。新的计数周期又从0111开始。

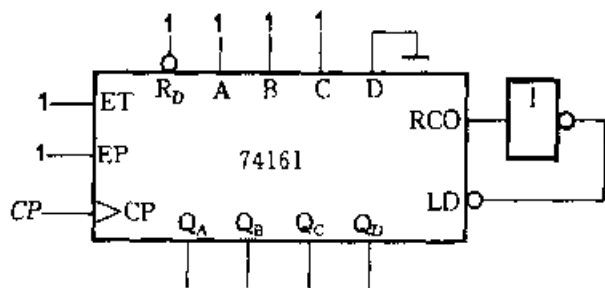


图 7.1.17 反馈置数法的另一种电路

具有异步置数功能的 M 进制集成计数器也可用反馈置数法构成 N 进制计数器。读者可自行分析两者的差异。

例 7.1.3 用 74HCT161 组成 256 进制计数器。

解: 因为 $N (= 256) > M (= 16)$, 且 $256 = 16 \times 16$, 所以要用两片 74HCT161 构成此计数器。每片均接成十六进制。片与片之间的连接方式有并行进位(低位片的进位信号作为高位片的使能信号)和串行进位(低位片的进位信号作为高位片的时钟脉冲,即异步计数方式)两种。

图 7.1.18a 是以并行进位的方式连接的 256 进制计数器。两片 74HCT161 的 CP 端均与计数脉冲 CP 连接,因而是同步计数器。低位片(片 1)的使能端 $ET = EP = 1$,因而它总处于计数状态;高位片(片 2)的使能端接至低位片的进位信号输出端 RCO,因而只有当片 1 计数至 1111 状态,使其 $RCO = 1$ 时,片 2 才能处于计数状态。在下一个计数脉冲作用后,片 2 计入一个脉冲,片 1 由 1111 状态变成 0000 状态,它的进位信号也变成 0,使片 2 停止计数。

图 7.1.18b 是以串行进位的方式连接的 256 进制计数器。其中,片 1 的进位输出信号 RCO 经反相器反相后作为片 2 的计数脉冲 CP_2 。显然,这是一个异步计数器。虽然两芯片的使能控制信号都为 1,但只有当片 1 由 1111 变成 0000 状态,使其 RCO 由 1 变为 0, CP_2 由 0 变为 1 时,片 2 才能计入一个脉冲。其他情况下,片 2 都将保持原有状态不变。

例 7.1.4 用 74LS290 构成二十四进制计数器。

解: 因为 $M = 10$, $N = 24$, 所以要用两片 74LS290。先将两芯片均接成十进制计数器,然后将它们连接成 100 进制计数器,在此基础上,再借助 74LS290 的异步清零功能,用反馈清零法将片 1 的 Q_C 和片 2 的 Q_B 分别接至两芯片的

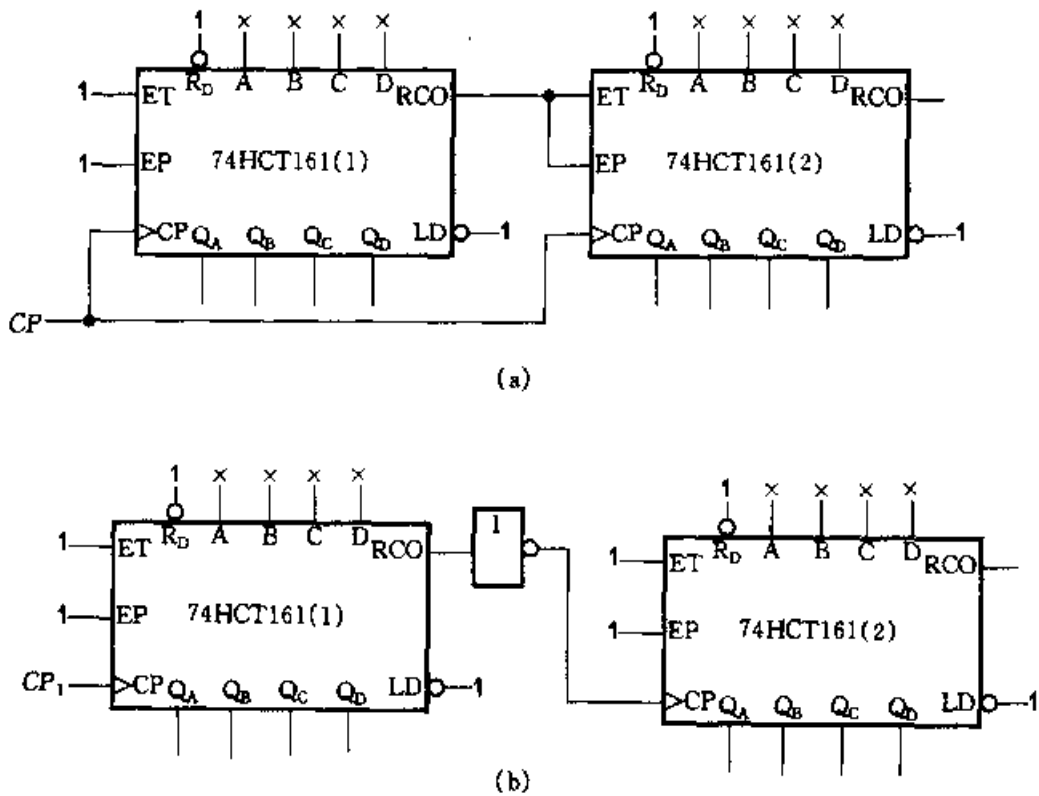


图 7.1.18 例 7.1.3 的逻辑电路图
(a) 并行进位方式 (b) 串行进位方式

$R_{0(1)}$ 和 $R_{0(2)}$ 端, 在第 24 个计数脉冲作用后, 计数器输出为 **00100100** 状态, 片 2 的 Q_B 与片 (1) 的 Q_C 同时为 1, 使计数器立即返回到 **00000000** 状态。状态 **00100100** 仅在较短的瞬间出现一下。这样, 就构成了二十四进制计数器。其逻辑电路如图 7.1.19 所示。

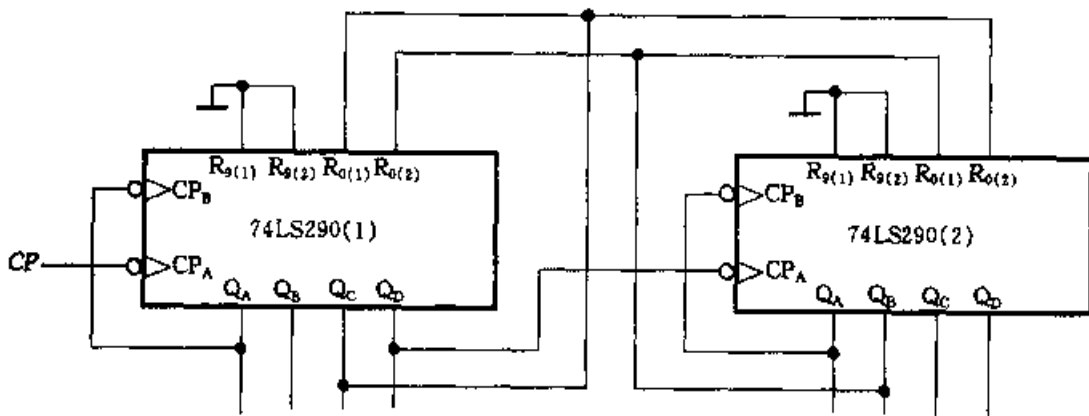


图 7.1.19 例 7.1.4 的逻辑电路图

这种连接方式可称为整体反馈清零法, 其原理与 $M > N$ 时的反馈清零法相

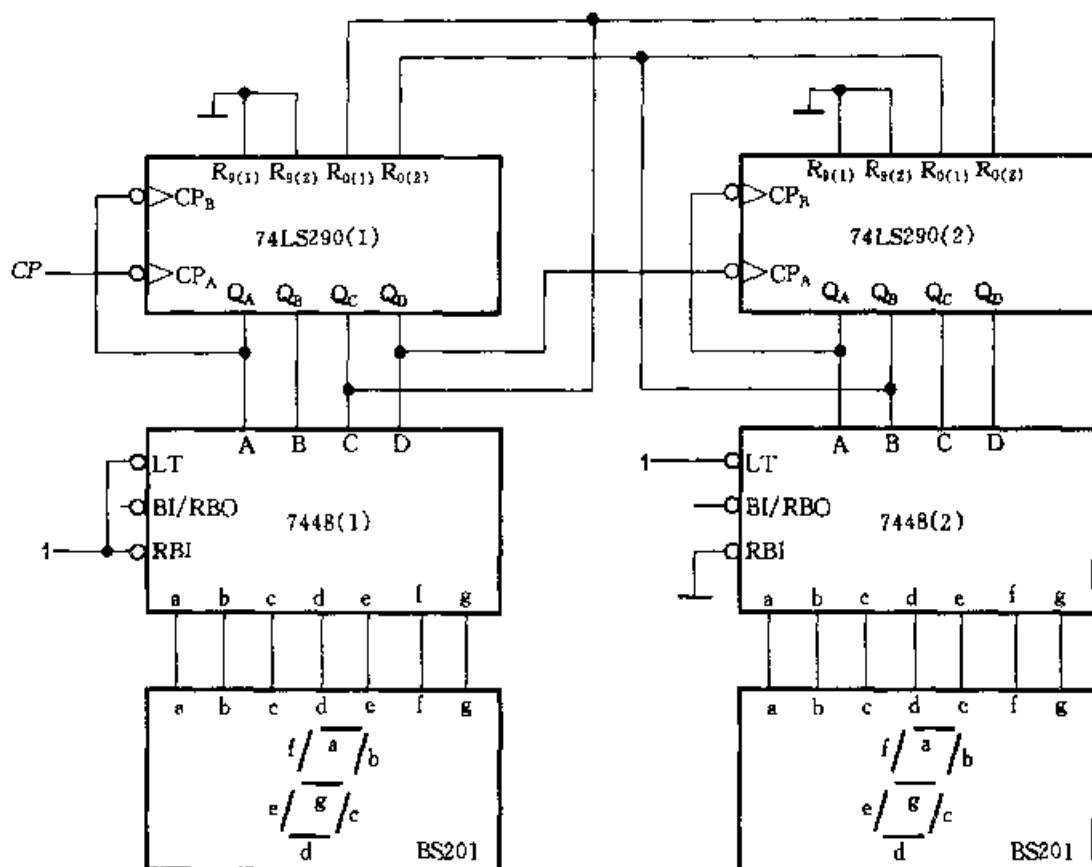


图 7.1.20 数字电子钟的时计数、译码、显示电路

同。也可以用具有预置数功能的集成计数器,采取整体反馈置数的方法构成二十四进制计数器,其原理与 $M > N$ 时的反馈置数法相似。读者可自行分析、设计。

二十四进制计数器是数字电子钟里必不可少的组成部分,用来累计小时数。将图 7.1.19 所示电路与 BCD-七段显示译码器 7448 及共阴极七段数码管显示器 BS201 连接起来,就组成了数字电子钟里时计数、译码及显示电路,如图 7.1.20 所示。

复习思考题

- 7.1.1 设计二进制同步计数器的主要步骤是什么?
- 7.1.2 如何将 74LS290 连接成 5421 码十进制计数器?
- 7.1.3 能否用具有同步清零功能的二进制计数器采取反馈清零法来构成 N 进制计数器?
- 7.1.4 用 74LS290 构成 $N > 10$ 进制计数器时,能采取并行进位的连接方式吗?
- 7.1.5 某电视图像系统中需要 512 进制计数器,如何用本节介绍的集成计数器来构成?

7.2 寄存器和移位寄存器

7.2.1 寄存器

寄存器是计算机和其他数字系统中用来存储代码或数据的逻辑部件。它的主要组成部分是触发器。一个触发器能存储 1 位二进制代码,所以要存储 n 位二进制代码的寄存器就需要用 n 个触发器组成。

一个 4 位的集成寄存器 74LS175 的逻辑电路图和引脚图分别如图 7.2.1a、b 所示。其中, R_D 是异步清零控制端。在往寄存器中寄存数据或代码之前,必须先将寄存器清零,否则有可能出错。1D~4D 是数据输入端,在 CP 脉冲上升沿作用下,1D~4D 端的数据被并行地存入寄存器。输出数据可以并行从 1Q

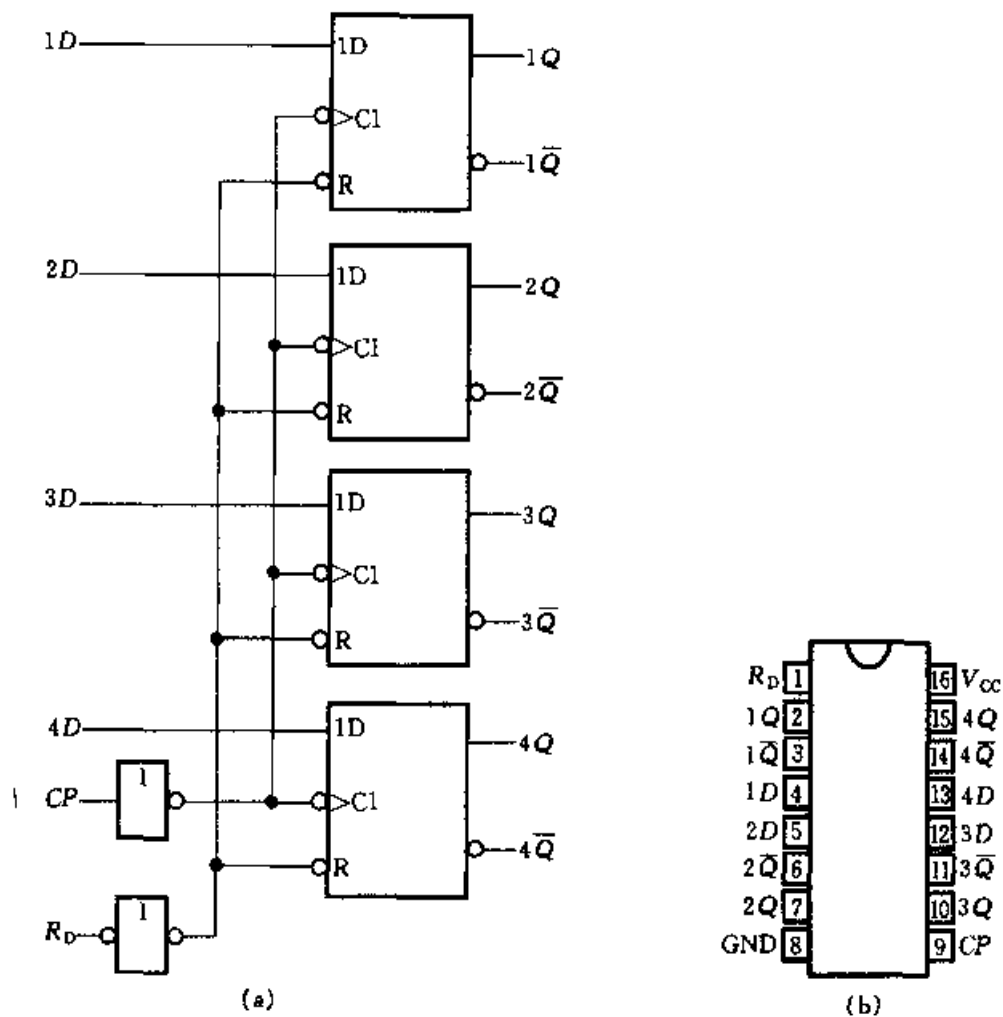



图 7.2.1 集成寄存器 74LS175

(a) 逻辑电路图 (b) 引脚图

$\sim 4Q$ 端引出,也可以并行从 $1\bar{Q}\sim 4\bar{Q}$ 端引出反码输出。

74LS175 的功能示于表 7.2.1 中。

表 7.2.1 74LS175 的功能表

输 入						输 出			
R_D	CP	1D	2D	3D	4D	1Q	2Q	3Q	4Q
L	×	×	×	×	×	L	L	L	L
H		1D	2D	3D	4D	1D	2D	3D	4D
H	H	×	×	×	×	保持			
H	L	×	×	×	×				

7.2.2 移位寄存器

上面介绍的寄存器只有寄存数据或代码的功能。有时为了处理数据,需要将寄存器中的各位数据在移位控制信号作用下,依次向高位或向低位移动 1 位。具有移位功能的寄存器称为移位寄存器。

1. 移位寄存器的工作原理

把若干个触发器串接起来,就可以构成一个移位寄存器。由 4 个边沿 D 触发器构成的 4 位移位寄存器逻辑电路如图 7.2.2 所示。数据从串行输入端 D_1 输入。左边触发器的输出作为右邻触发器的数据输入。

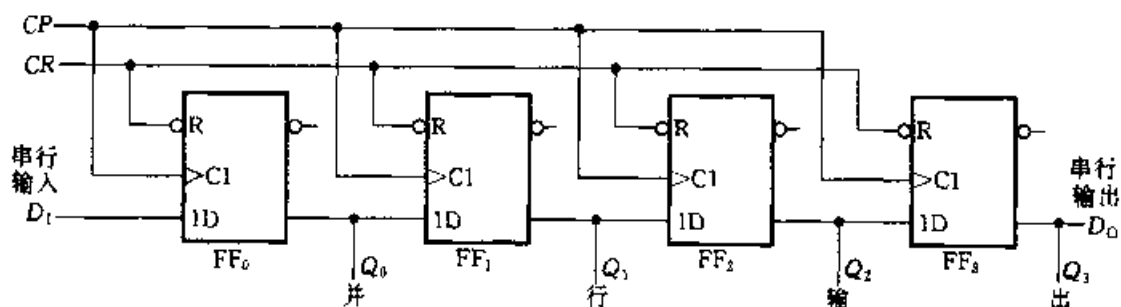


图 7.2.2 用边沿 D 触发器构成的 4 位移位寄存器

假设移位寄存器的初始状态为 0000, 现将数码 $D_3D_2D_1D_0$ (1101) 从高位 (D_3) 至低位依次送到 D_1 端, 经过第一个时钟脉冲后, $Q_0 = D_3$ 。由于跟随数码 D_3 后面的数码是 D_2 , 则经过第二个时钟脉冲后, 触发器 FF_0 的状态移入触发器 FF_1 , 而 FF_0 变为新的状态, 即 $Q_1 = D_3, Q_0 = D_2$ 。依此类推, 可得 4 位右向移位寄存器的状态, 如表 7.2.2 所示。由表可知, 输入数码依次地由低位触发器移到高位触发器, 作右向移动。经过 4 个时钟脉冲后, 4 个触发器的输出状态 $Q_3Q_2Q_1Q_0$ 与输入数码 $D_3D_2D_1D_0$ 相对应。为了加深理解, 在图 7.2.3 中画

出了数码**1101**(相当于 $D_3 = 1, D_2 = 1, D_1 = 0, D_0 = 1$) 在寄存器中移位的波形, 经过 4 个时钟脉冲后, **1101** 出现在寄存器的输出端 $Q_3 Q_2 Q_1 Q_0$ 。这样, 就可将串行输入(从 D_1 端输入)的数码转换为并行输出(从 Q_3, Q_2, Q_1, Q_0 端输出)的数码。这种转换方式特别适用于将遥测到的串行输入信号转换为并行输出信号, 以便于打印或由计算机处理。

表 7.2.2 图 7.2.2 电路的状态表

CP	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	D_3	0	0	0
2	D_2	D_3	0	0
3	D_1	D_2	D_3	0
4	D_0	D_1	D_2	D_3

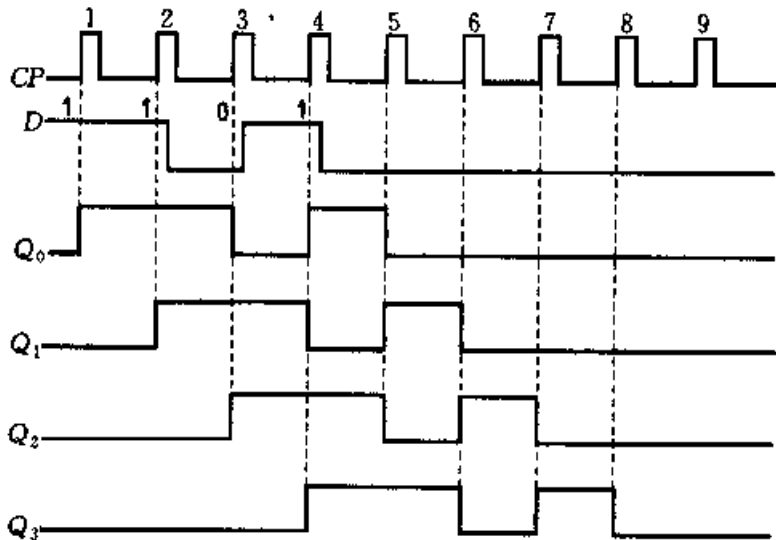


图 7.2.3 图 7.2.2 电路的时序图

在图 7.2.3 中还画出了第 5 到第 8 个时钟脉冲作用下, 输入数码在寄存器中移位的波形。由图可见, 在第 8 个时钟脉冲作用后, 数码从 Q_3 端已全部移出寄存器。这说明存入该寄存器中的数码也可以从 D_0 端串行输出。

根据需要, 可用更多的触发器组成多位移位寄存器。

除了用边沿 D 触发器外, 还可用其他类型的触发器来组成移位寄存器, 例如, 用主从 JK 触发器来组成移位寄存器, 其级间连接方式如图 7.2.4 所示。

根据 JK 触发器的特性方程, 由图 7.2.4 可得 $Q_0^{n+1} = D_1 \overline{Q_0^n} + D_1 Q_0^n = D_1$, $Q_1^{n+1} = J_1 \overline{Q_1^n} + \overline{K_1} Q_1^n = Q_0 \overline{Q_1^n} + Q_0 Q_1^n = Q_0$, FF_2 和 FF_3 的接法与 FF_1 完全相似, 所以各 JK 触发器均以 D 触发器的功能工作, 也就是说, 图 7.2.4 和图 7.2.2 所示电路具有相同的功能。

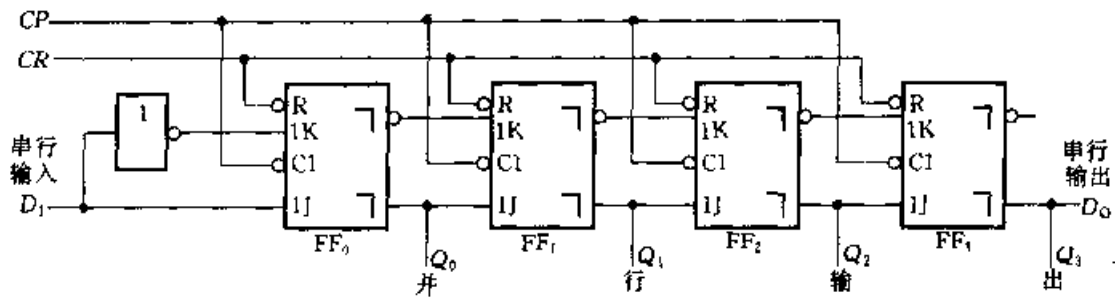


图 7.2.4 用主从 JK 触发器组成的 4 位移位寄存器

2. 双向移位寄存器

若将图 7.2.2 所示电路中各触发器间的连接顺序调换一下,让右边触发器的输出作为左邻触发器的数据输入,则可构成左向移位寄存器。若再增添一些控制门,则可构成既能右移(由低位向高位)、又能左移(由高位至低位)的双向移位寄存器。图 7.2.5 是双向移位寄存器的一种方案,它是利用边沿 D 触发器组成的,每个触发器的数据输入端 D 同与非门组成的转换控制门相连,移位方向取决于移位控制端 S 的状态。以触发器 FF₀、FF₁ 为例,其数据输入端 D 的逻辑表达式分别为

$$D_0 = \overline{S} \overline{D_{SR}} + \overline{S} \overline{Q_1}$$

$$D_1 = \overline{S} \overline{Q_0} + \overline{S} \overline{Q_2}$$

当 $S=1$ 时, $D_0 = D_{SR}$, $D_1 = Q_0$, 即 FF₀ 的 D₀ 端与右移串行输入端 D_{SR} 连通, FF₁ 的 D₁ 端与 Q₀ 连通, 在时钟脉冲 CP 作用下, 由 D_{SR} 端输入的数据将作右向移位; 反之, 当 $S=0$ 时, $D_0 = Q_1$, $D_1 = Q_2$, 在时钟脉冲 CP 作用下, Q₂、Q₁ 的状态将作左向移位。同理, 可以分析其他两位触发器间的移位情况。

由此可见, 图 7.2.5 所示寄存器可作双向移位。当 $S=1$ 时, 数据作右向移位; 当 $S=0$ 时, 数据作左向移位。可实现串行输入-串行输出(由 D_{OR} 或 D_{OL} 输出)、串行输入-并行输出工作方式(由 Q₃~Q₀ 输出)。

7.2.3 集成移位寄存器 74194

集成移位寄存器 74194 由 4 个 RS 触发器及它们的输入控制电路组成。图 7.2.6a、b 分别是它的逻辑电路图和引脚图。与图 7.2.5 相比, 除增加了 4 个并行输入端 A~D 外, 还有两个控制输入端 S₁、S₀。如表 7.2.3 所示, 它们的状态组合可以完成 4 种控制功能, 其中左移和右移两项是指串行输入, 数据是分别从左移输入端 D_{SL} 和右移输入端 D_{SR} 送入寄存器的。R_D 为异步清零输入端。表 7.2.4 是 74194 的功能表。其第 1 行表示寄存器异步清零; 第 2 行表示当 R_D =

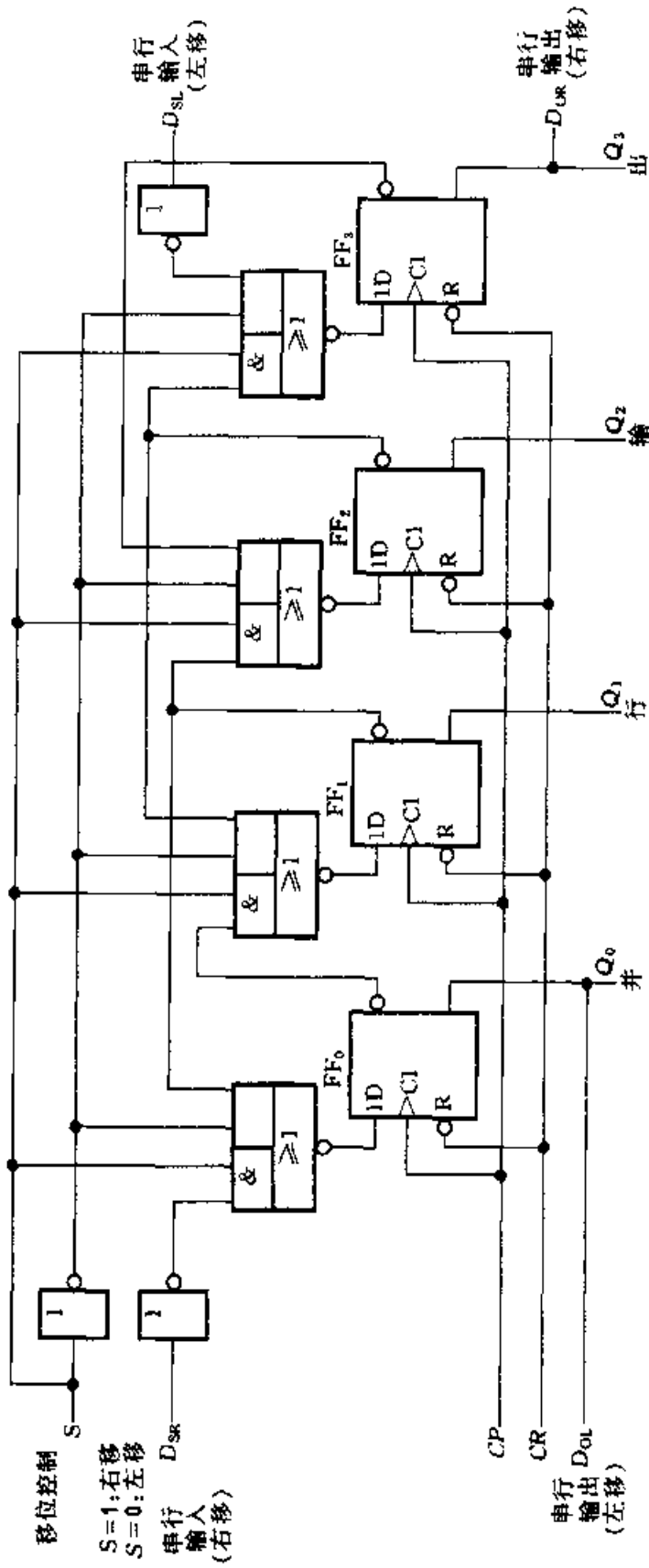


图 7.2.5 用边沿 D 触发器组成的双向移位寄存器

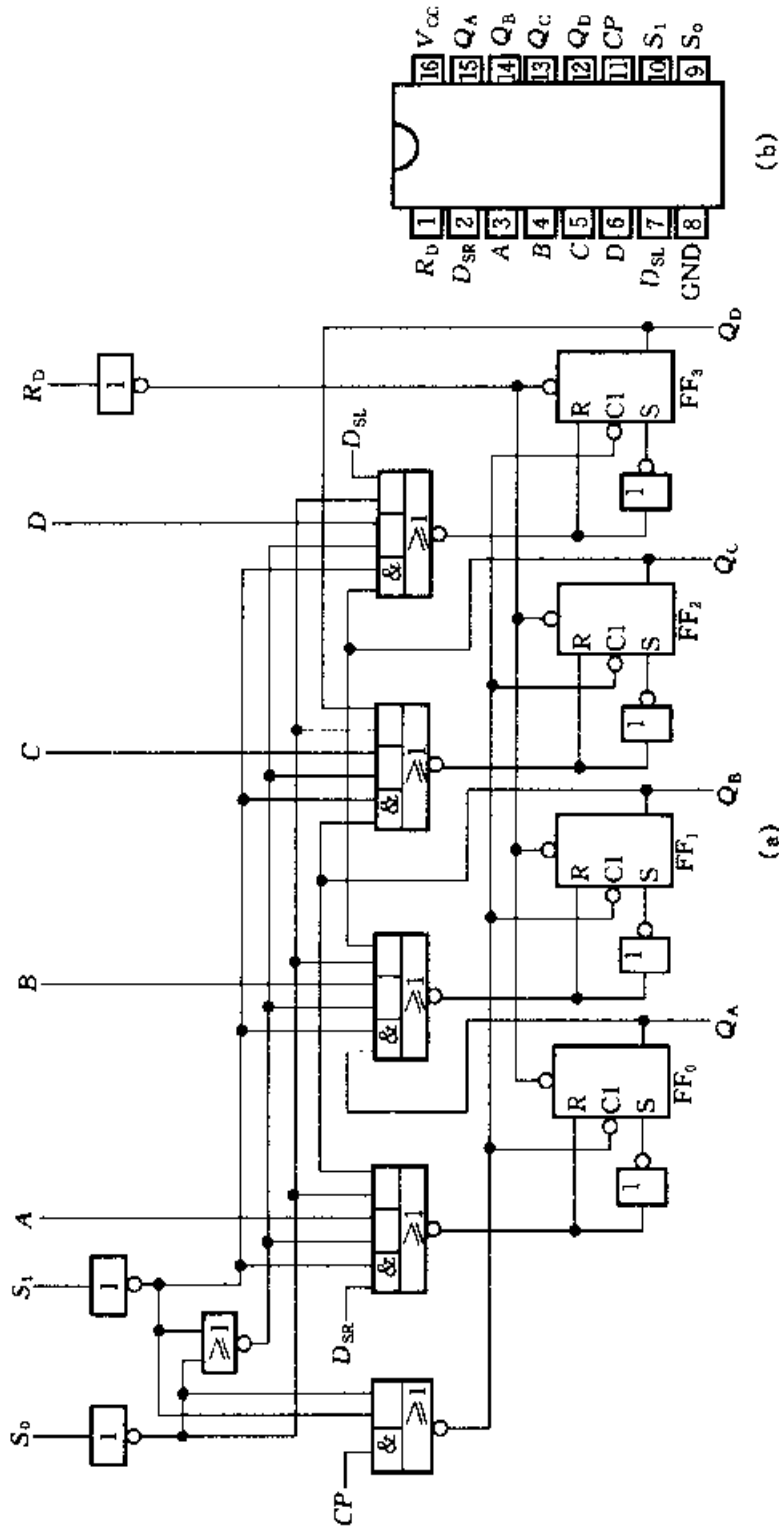


图 7.2.6 4 位双向移位寄存器 74194

(a) 逻辑电路图 (b) 引脚图

1, $CP=1$ (或0)时, 寄存器处于原来状态; 第3行表示为并行输入同步预置数; 第4、5行为串行输入左移; 第6、7行为串行输入右移; 第8行为保持状态。

表 7.2.3 74194 双向移位寄存器控制端的逻辑功能

控 制 信 号		完成的 功能
S_1	S_0	
0	0	保 持
0	1	右 移
1	0	左 移
1	1	并行输入

表 7.2.4 74194 的功能

序号	清零 R_D	输 入					输 出							
		控制信号		串行输入		时钟脉 冲 CP	并行输入							
		S_1	S_0	左移 D_{SL}	右移 D_{SR}		D	C	B	A	Q_D	Q_C	Q_B	Q_A
1	L	×	×	×	×	×	×	×	×	L	L	L	L	
2	H	×	×	×	×	H(L)	×	×	×	×	Q_D^c	Q_C^c	Q_B^c	Q_A^c
3	H	H	H	×	×	\downarrow	D	C	B	A	D	C	B	A
4	H	H	L	H	×	\downarrow	×	×	×	×	H	Q_D^c	Q_C^c	Q_B^c
5	H	H	L	L	×	\downarrow	×	×	×	×	L	Q_D^c	Q_C^c	Q_B^c
6	H	L	H	×	H	\downarrow	×	×	×	×	Q_C^c	Q_B^c	Q_A^c	H
7	H	L	H	×	L	\downarrow	×	×	×	×	Q_C^c	Q_B^c	Q_A^c	L
8	H	L	L	×	×	×	×	×	×	×	Q_D^c	Q_C^c	Q_B^c	Q_A^c

有时要求在移位过程中数据不要丢失, 仍然保持在寄存器中。此时, 只要将移位寄存器的最高位的输出接至最低位的输入端, 或将最低位的输出接至最高位的输入端, 即将移位寄存器的首尾相连就可实现上述功能。这种寄存器称为循环移位寄存器, 它也可以作为计数器用, 称为环形计数器。

复习思考题

7.2.1 用 74LS175 能实现数据的串行-并行或并行-串行转换吗?

7.2.2 若用同步 RS 触发器组成移位寄存器, 能否正常工作? 会产生什么现象?

7.2.3 用 1 片全加器和 1 个 D 触发器及两片 8 位移位寄存器 A、B 构成一-串行加法器, 全加器的进位延迟用 D 触发器实现。A 寄存器存放加数, B 寄存器存放被加数及和数。试画出其逻辑图。(注: 加数和被加数均并行置入寄存器)

* 7.3 CAD 例题

例 CE7.3.1 试用正边沿 D 触发器设计一个十二进制同步递增计数器。

解：(1) 逻辑设计

① 十二进制同步递增计数器有 12 个状态, 假设在工作过程中所经历的状态为(0000~1011), 而状态(1100~1111)5 个是无效的组, 则设计十二进制计数器需用 4 个 D 触发器。

② 根据计数器状态转换关系, 列出状态表和驱动表, 求得各位 D 触发器的驱动信号的表达式。

③ 画出该计数器的逻辑电路图, 如图 CE7.3.1a 所示。

(2) 输入设计文件

调用 ISP Synario 软件中的原理图编辑器, 输入图 CE7.3.1a 所示逻辑电路, 并保存文件。然后对原理图文件进行编译和逻辑简化。

(3) 编写并输入测试向量文件

根据图 CE7.3.1a 所示逻辑电路, 编写的测试向量文件如下:

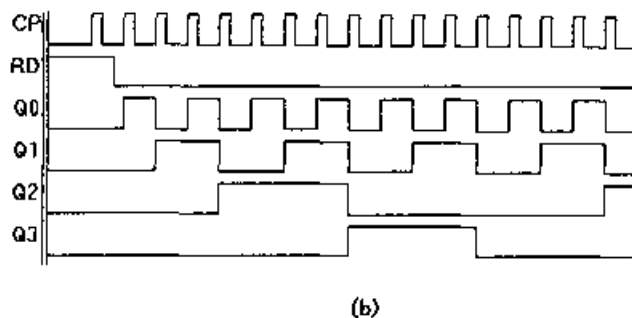
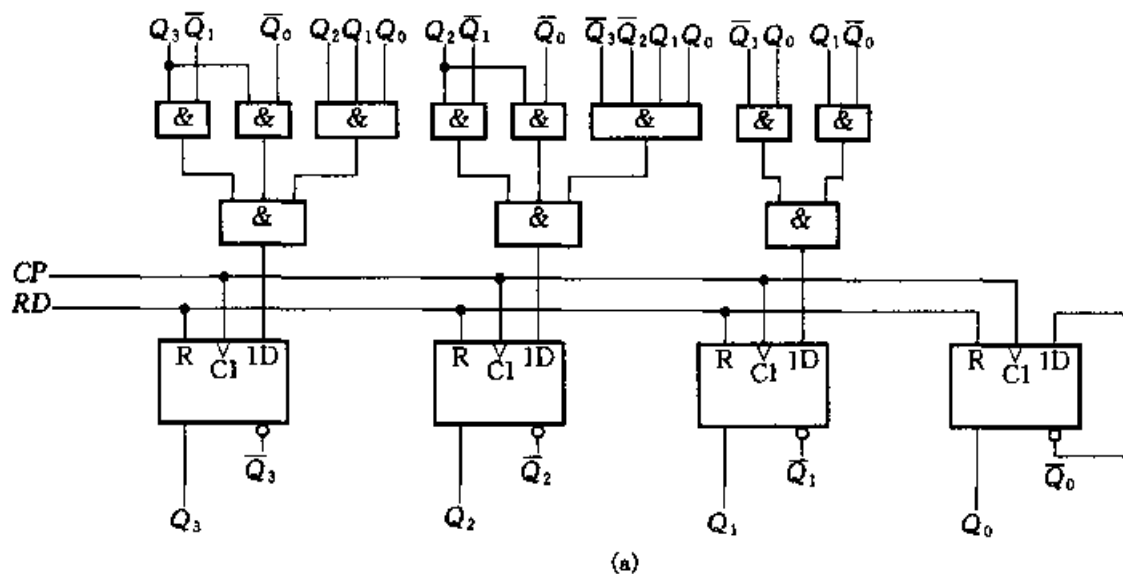


图 CE7.3.1 十二进制递增计数器电路图及逻辑功能仿真波形图

(a) 电路图 (b) 仿真波形图

```

module CE7_3_1
    CP, RD pin;
    Q0..Q3 pin istype 'reg';
    C, X, P = .C., .X., .P.;
test_vectors
    ([CP, RD] -> [Q3..Q0])
    [X, 1] -> X;
    [C, 1] -> X;
    @repeat 16 { [C, 0] -> X; }
end

```

(4) 进行逻辑功能仿真

在 ISP Synario 软件中,对测试向量文件进行编译和逻辑功能仿真,得到图 CE7.3.1b 所示的逻辑功能仿真波波形图。

例 CE7.3.2 用 ISP Synario 软件分析图 CE7.3.2a(即图题 7.1.15)所示电路,给出逻辑功能仿真波形。并说明它是多少进制的计数器,采用了何种进位方式。

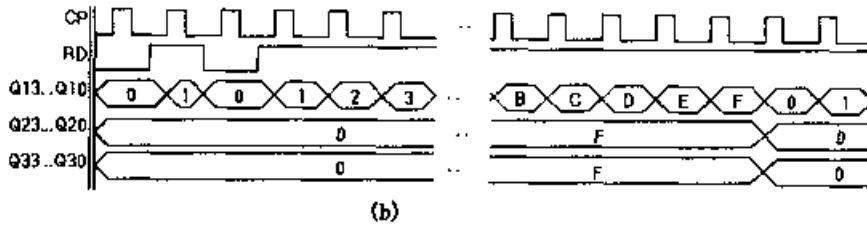
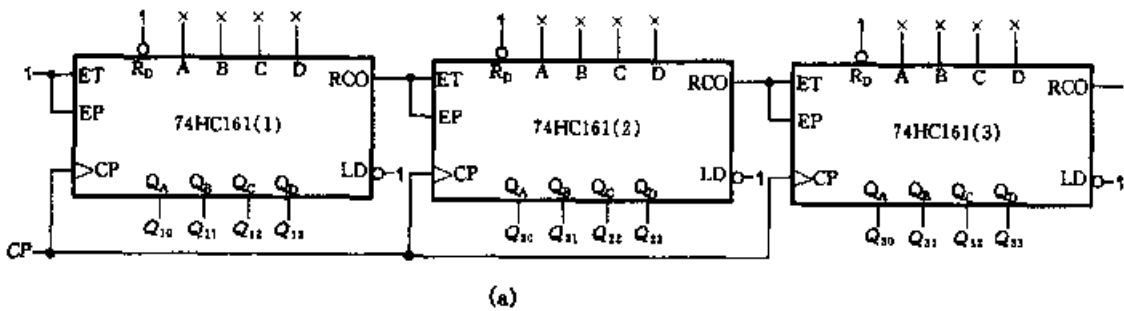


图 CE7.3.2 逻辑电路图及其仿真波形图

(a) 电路图 (b) 仿真波形图

解: (1) 输入设计文件

调用 ISP Synario 软件的原理图编辑器,输入图 CE7.3.2a 所示逻辑电路,并保存文件。然后对原理图文件进行编译和逻辑简化。

(2) 编写并输入测试向量文件

根据图 CE7.3.2a 所示逻辑电路,编写的测试向量文件如下:

```
module CE7_3_2
declarations
    CP,RD                                pin;
    Q10..Q13,Q20..Q23,Q30..Q33          pin istype 'buffer,reg';
    C, X = .C., .X.;
test_vectors
    ([CP,RD]->[Q10..Q13,Q20..Q23,Q30..Q33])
    [C,0]->X;
    [C,1]->X;
    [C,0]->X;
    @repeat 5000{[C,1]->X;}
end
```

(3) 进行逻辑功能仿真

对测试向量文件进行编译和逻辑功能仿真,得到如图 CE7.3.2b 所示的仿真波形。

(4) 由图 CE7.3.2b 所示的波形图可知,该电路是 $16 \times 16 \times 16 = 4096$ 进制计数器。图中的计数器采用了并行进位输出方式。

小 结

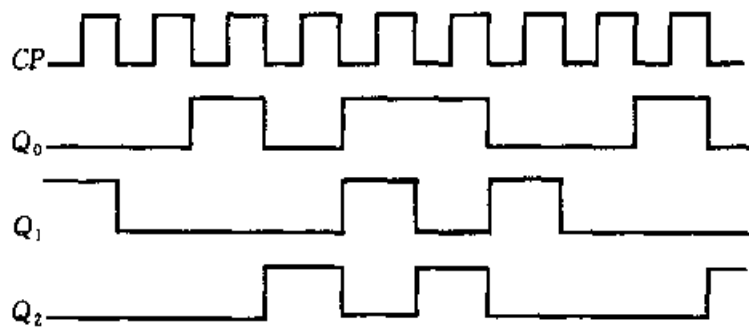
• 计数器和寄存器是简单而又最常用的时序逻辑器件。它们在计算机和其他数字系统中的作用往往超过了它们自身的功能。计数器不仅能用于统计输入时钟脉冲的个数,还能用于分频、定时、产生节拍脉冲等。寄存器的功能是存储代码。移位寄存器不但可以存储代码,还可用来实现数据的串行-并行转换、数据处理及数值的运算。

• 分析和设计(用触发器和门电路构成的)计数器、寄存器及由它们组成的电路时,仍然是用第6章中介绍的时序电路的分析和设计方法。

• 用已有的 M 进制集成计数器产品可以构成 N (任意)进制的计数器。当 $M > N$ 时,用1片 M 进制计数器,采取反馈清零法或反馈置数法,跳过 $M - N$ 个状态,就可以得到 N 进制的计数器了;当 $M < N$ 时,要用多片 M 进制计数器组合起来,才能构成 N 进制计数器。各级之间的连接方式可分为并行进位、串行进位、整体反馈清零和整体反馈置数等几种方式。

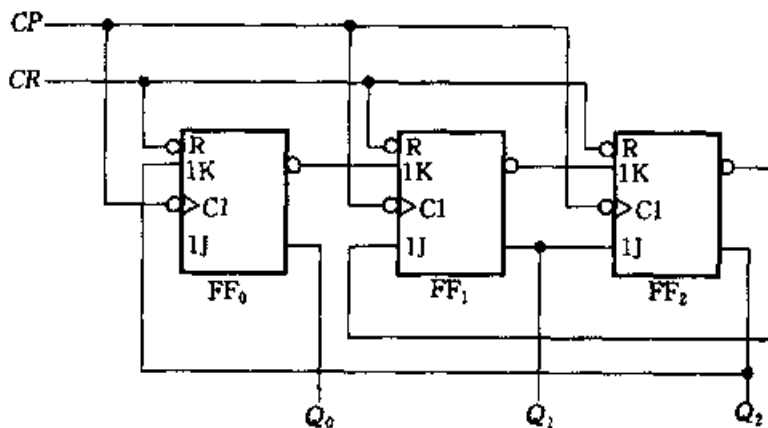
习 题

7.1.1 在某计数器的输出端观察到如图题 7.1.1 所示的波形,试确定该计数器的模。



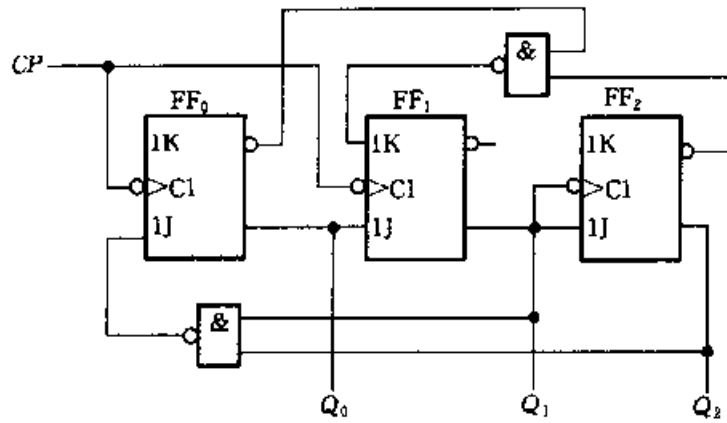
图题 7.1.1

- 7.1.2 试用负边沿 JK 触发器组成 4 位二进制异步减计数器,画出逻辑图。
 7.1.3 试用负边沿 D 触发器组成 4 位二进制异步加计数器,画出逻辑图。
 7.1.4 试用正边沿 D 触发器及门电路组成 3 位二进制同步加计数器,画出逻辑图。
 7.1.5 试分析图题 7.1.5 电路是几进制计数器,画出各触发器输出端的波形图。

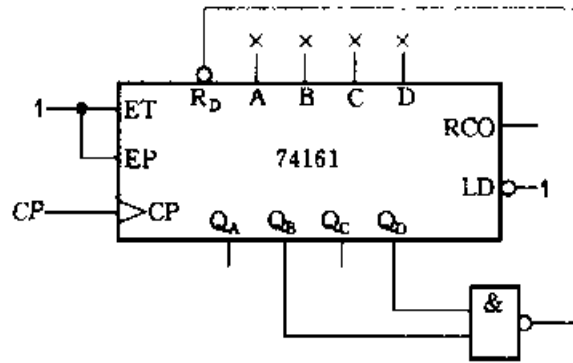


图题 7.1.5

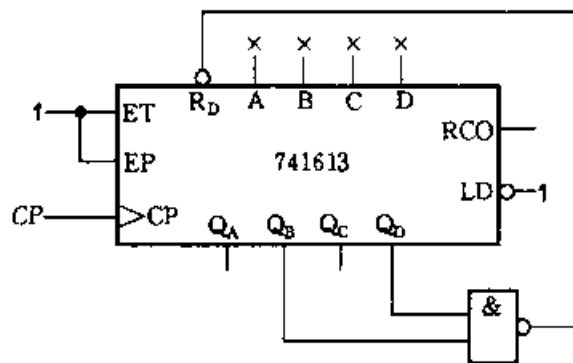
- 7.1.6 电路如图题 7.1.6 所示,试分析它是几进制计数器,画出它的状态转换图。
 7.1.7 试用正边沿 D 触发器和门电路设计一个同步三进制减计数器。
 7.1.8 试用主从 JK 触发器设计一个同步六进制加计数器。
 7.1.9 试分析图题 7.1.9 所示电路,画出它的状态图,说明它是几进制计数器。
 7.1.10 试分析图题 7.1.10 所示电路,画出它的状态图,说明它是几进制计数器。
 (74163 是具有同步清零功能的 4 位二进制同步加计数器,其他功能与 74161 相同)
 7.1.11 试分析图题 7.1.11 所示电路,画出它的状态图,说明它是几进制计数器。
 7.1.12 试分析图题 7.1.12 所示电路,画出它的状态图,说明它是几进制计数器。



图题 7.1.6



图题 7.1.9

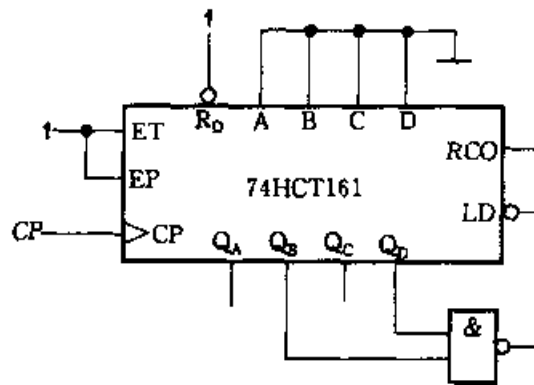


图题 7.1.10

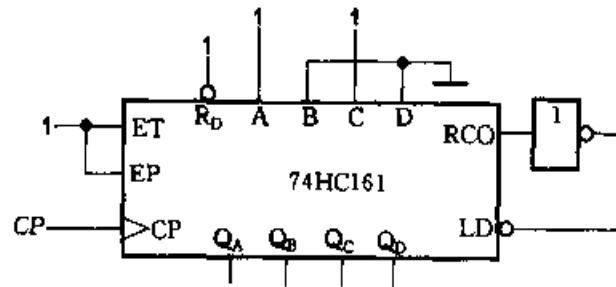
7.1.13 试用 74HCT161 设计一个计数器,其计数状态为自然二进制数 1001~1111。

7.1.14 试用 74LS290 设计一个六进制计数器,要求采取两种不同的方法。

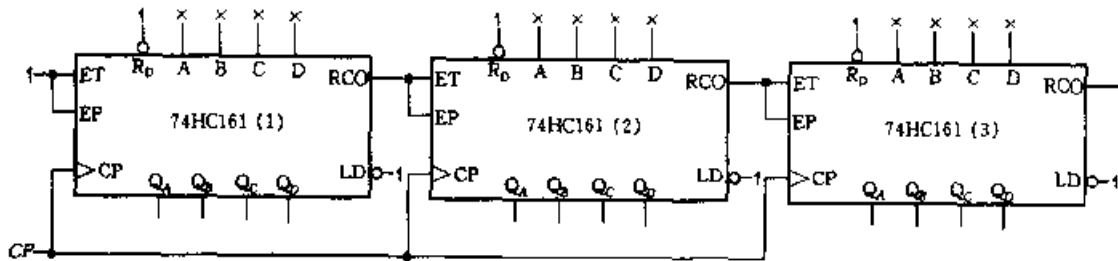
7.1.15 试分析图题 7.1.15 所示电路,说明它是多少进制的计数器,采用了何种进位方式。



图题 7.1.11

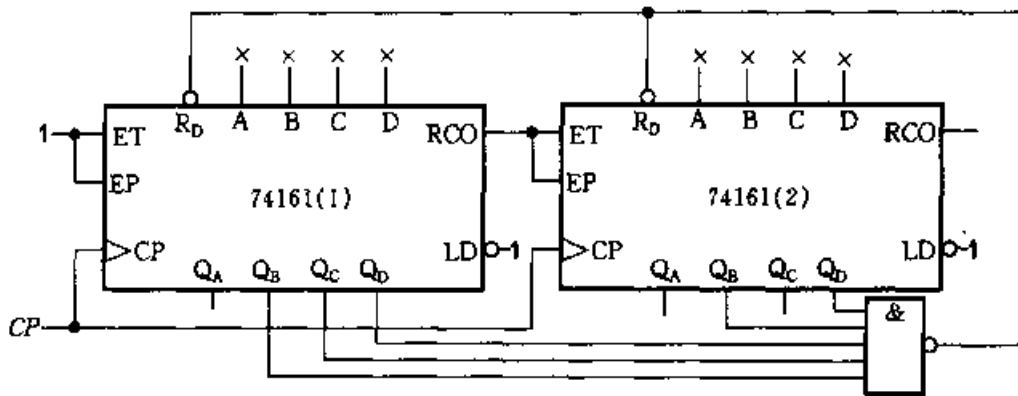


图题 7.1.12

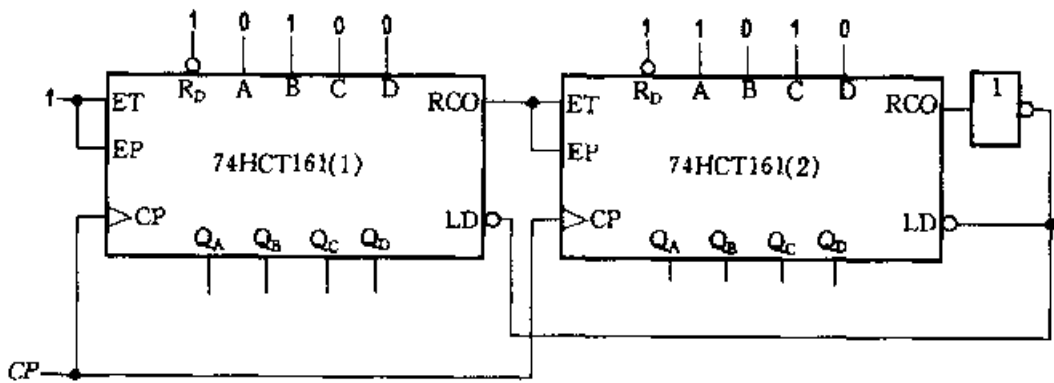


图题 7.1.15

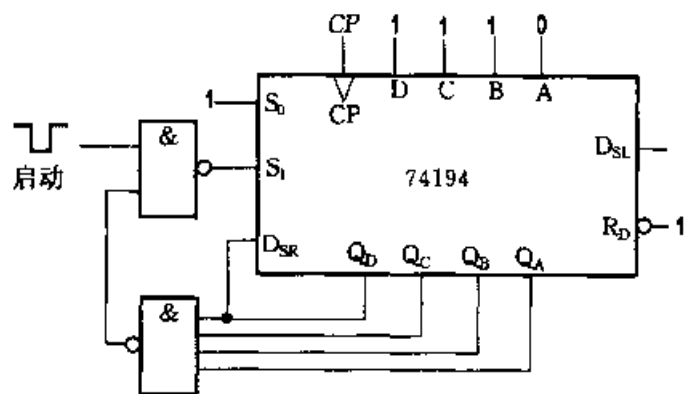
- 7.1.16 试分析图题 7.1.16 所示电路,说明它是多少进制的计数器。
- 7.1.17 试分析图题 7.1.17 所示电路,说明它是多少进制的计数器。
- 7.1.18 试用 74161 构成同步二十四进制计数器,要求采用两种不同的方法。
- 7.2.1 试画出图题 7.2.1 所示逻辑电路的输出($Q_A \sim Q_D$)波形,并分析该电路的逻辑功能。
- 7.2.2 试用两片 74194 构成 8 位双向移位寄存器。



图题 7.1.16



图题 7.1.17



图题 7.2.1

* CAD 习题

C7.3.1 试用 ISP Synario 软件分析图题 7.1.5 所示电路, 给出各触发器输出端的波形图, 并说明它是多少进制的计数器。

C7.3.2 试用 74HC161 以两种不同的方法构成二十四进制同步递增计数器,然后用 ISP Synario 软件对所设计的电路进行逻辑功能仿真,给出仿真波形图。

C7.3.3 试用 ISP Synario 软件分析图题 7.2.1 所示电路,给出电路的输出(QA~QD)波形图,并分析该电路的逻辑功能。

C7.3.4 试用两片 74HC194 构成 8 位双向移位寄存器,然后用 ISP Synario 软件对所设计的电路进行逻辑功能仿真,给出仿真波形图。

8 半导体存储器和可编程逻辑器件

引言 半导体存储器几乎是当今数字系统中不可缺少的组成部分,它用来存储大量的二值数据。按照集成度划分,半导体存储器属于大规模集成电路。

另一类功能特殊的大规模集成电路是70年代后期发展起来的可编程逻辑器件。前面各章介绍的中小规模标准集成器件性能好、价格低,但是仅仅采用这些器件构成一个大型复杂的数字系统,常常可能导致系统功耗高、占用空间大和系统可靠性差等问题。可编程逻辑器件较好地解决了以上问题,并在工业控制和产品开发等方面得到了广泛的应用。可编程逻辑器件是一种可以由用户定义和设置逻辑功能的器件。该类器件具有结构灵活、集成度高、处理速度快和可靠性高等特点。

本章^①首先介绍半导体存储器分类方法、电路结构和工作原理,然后介绍几种典型的可编程逻辑器件的基本结构和实现逻辑功能的编程原理。

8.1 随机存取存储器(RAM)

根据使用功能的不同,半导体存储器可分为随机存取存储器(RAM^②,又称为读写存储器)和只读存储器(ROM^③)。按照存储机理的不同,RAM又可分为静态RAM(Static RAM,简称SRAM)和动态RAM(Dynamic RAM,简称DRAM)。RAM使用灵活、方便,可以随时从其中任一指定地址读出(取出)或写入(存入)数据。但是RAM具有易失性,一旦失电,所存储的数据立即丢失。

① 为便于与国际市场上的产品相适应,本章所用图形和文字符号采用国外符号。

② Random Access Memory的缩写。

③ Read Only memory的缩写。

8.1.1 RAM 的电路结构与工作原理

1. RAM 存储单元

存储单元是存储器的最基本存储细胞,它可以存放 1 位二值数据。

(1) 静态 RAM 存储单元

静态 RAM 中存储单元的结构如图 8.1.1 所示。虚线框中的存储单元为六管 SRAM 存储单元,其中 $T_1 \sim T_4$ 构成一个基本 RS 触发器,用来存储 1 位二值数据。 T_5 、 T_6 为本单元控制门,由行选择线 X_i 控制。 $X_i = 1$, T_5 、 T_6 导通,触发器与位线接通; $X_i = 0$, T_5 、 T_6 截止,触发器与位线隔离。 T_7 、 T_8 为一列存储单元公用的控制门,用于控制位线与数据线的连接状态,由列选择线 Y_j 控制。显然,当行选择线和列选择线均为高电平时, $T_5 \sim T_8$ 都导通,触发器的输出才与数据线接通,该单元才能通过数据线传送数据。因此,存储单元能够进行读/写操作的条件是,与它相连的行、列选择线均须呈高电平。

由静态存储单元构成的静态 RAM 的特点是,数据由触发器记忆,只要不断电,数据就能永久保存。

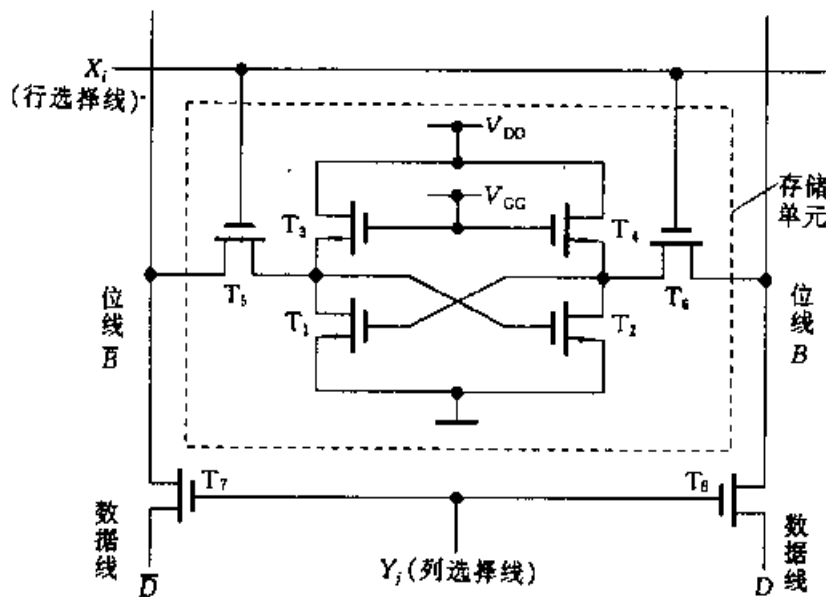


图 8.1.1 六管静态存储单元

(2) 动态 RAM 存储单元

静态 RAM 存储单元所用的管子数目多,功耗大,集成度受到限制,为了克服这些缺点,人们研制出了动态 RAM。动态 RAM 存储数据的原理是基于 MOS 管栅极电容的电荷存储效应。由于漏电流的存在,电容上存储的数据(电荷)不能长久保存,因此必须定期给电容补充电荷,以避免存储数据的丢失,这种操作称为再生或刷新。

常见的动态 RAM 存储单元有三管和单管两种。图 8.1.2 所示为三管动态存储单元,存储单元是以 MOS 管 T_2 及其栅极电容 C 为基础构成的,数据存于栅极电容 C 中。若电容 C 充有足够的电荷,使 T_2 导通,这一状态为逻辑 0,否则为逻辑 1。图中除了存储单元外,还画出了该列存储单元公用的写入刷新控制电路。

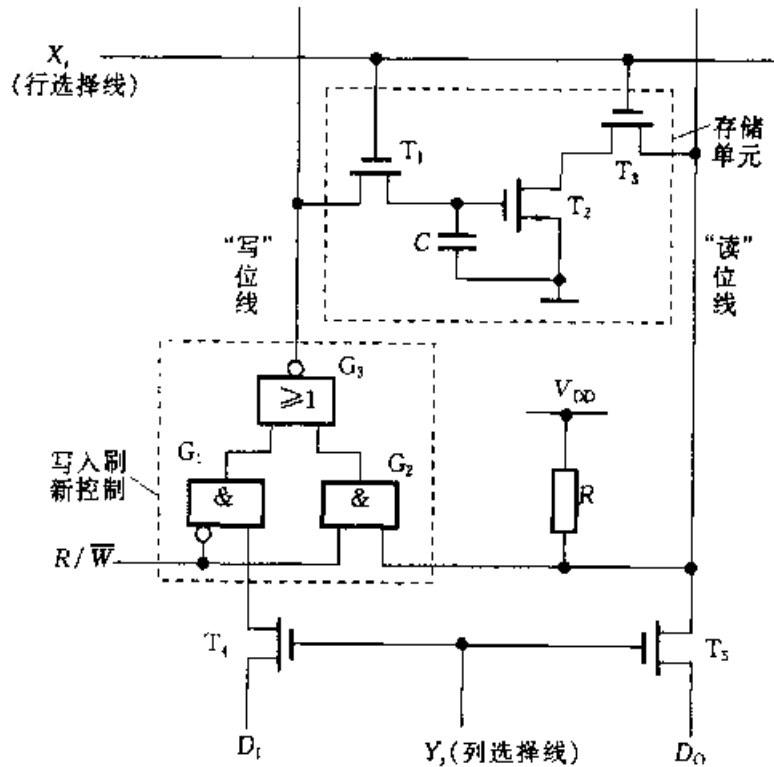


图 8.1.2 三管动态存储单元

图 8.1.2 中行、列选择线 X_j 、 Y_j 均为高电平时,存储单元被选中;读/写控制信号 R/\overline{W} 为高电平时进行读操作,低电平时进行写操作; G_1 门输入端的小圆圈“○”表示反相。

在进行读操作时,地址信号使门控管 T_3 导通,此时若 C 上充有电荷且使 T_2 导通,则读出数据为 0;反之, T_2 截止,使“读”位线获得高电平,输出数据为 1。由图可以看出,“读”位线信号分为两路,一路经 T_3 由 D_0 输出,另一路经写入刷新控制电路对存储单元刷新。

进行写操作时, R/\overline{W} 为低电平($R/\overline{W} = 0$),此时 G_2 被封锁,由于 Y_j 为高电平, T_4 导通,输入数据 D_1 经 T_4 并由写入刷新控制电路反相,再经 T_1 写入到电容器 C 中。这样,当输入数据为 0 时,电容充电;而输入数据为 1 时,电容放电。

除了读、写操作可以进行刷新外,刷新操作也可以通过只选通行选择线来

实现。例如,当行选择线 X_i 为高电平,且 R/\overline{W} 读有效时($R/\overline{W}=1$), C 上的数据经 T_2 、 T_3 到达“读”位线,然后经写入刷新控制电路对存储单元刷新。此时 X_i 有效的整个一行存储单元被刷新。由于列选择线 Y_j 无效,因此数据不被读出。

为了提高集成度,目前大容量动态 RAM 的存储单元普遍采用单管结构,其电路如图 8.1.3 所示。0 或 1 数据存于电容 C_s 中, T 为门控管,通过控制 T 的导通与截止,可以把数据从存储单元送至位线上或者将位线上的数据写入到存储单元。

为了节省芯片面积,存储单元的电容 C_s 不能做得很大,而位线上连接的元件较多,杂散电容 C_w 远大于 C_s 。当读出数据时,电容 C_s 上的电荷向 C_w 转移,位线上的电压 V_w 远小于读出操作前 C_s 上的电压 V_s (即 $V_w = V_s C_s / (C_s + C_w)$)。因此,需经读出放大器^① 对信号放大。同时,由于 C_s 上的电荷减少,存储的数据被破坏,故每次读出后,必须及时对读出单元刷新。

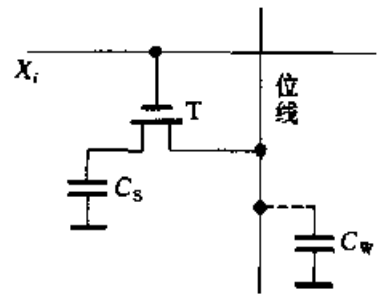


图 8.1.3 单管动态存储单元

2. RAM 的基本结构

一般而言,存储器由存储矩阵、地址译码器和输入/输出控制电路 3 部分组成,其结构如图 8.1.4 所示,由此看出进出存储器有 3 类信号线,即地址线、数据线和控制线。

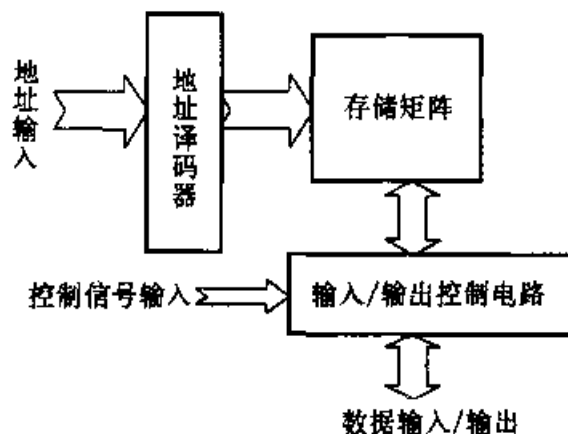


图 8.1.4 RAM 电路的基本结构

(1) 存储矩阵

^① 也称为灵敏再生放大器,见本书第三版第 299 页。

一个存储器由许多存储单元组成,每个存储单元存放1位二值数据。通常存储单元排列成矩阵形式。存储器以字为单位组织内部结构,1个字含有若干个存储单元。1个字中所含的位数称为字长。在实际应用中,常以字数和字长的乘积表示存储器的容量,存储器的容量越大,意味着存储器存储的数据越多。

例如,一个容量为 256×4 (256个字,每字4位)的存储器,有1024个存储单元,这些单元可以排成 32×32 列的矩阵形式,如图8.1.5所示。图中每行有32个存储单元,每4列存储单元连接在相同的列地址译码线上,组成一个字列,由此看出,每行可存储8个字,每个字列可储存32个字。每根行地址选择线选中一行,每根列地址选择线选中一个字列。因此,图示阵列有32根行地址选择线和8根列地址选择线。

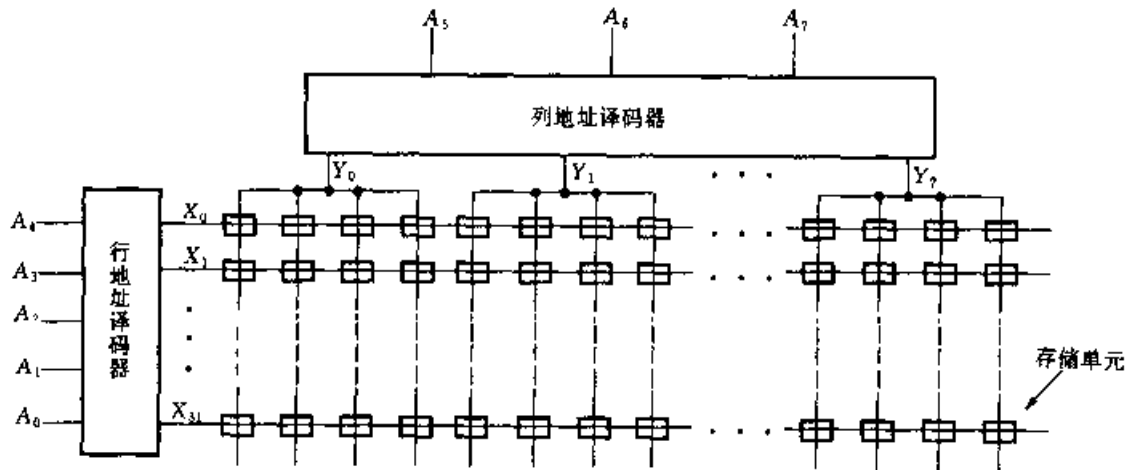


图 8.1.5 256×4 RAM 存储矩阵

(2) 地址译码

通常 RAM 以字为单位进行数据的读出与写入(每次写入或读出一个字),为了区别各个不同的字,将存放同一个字的存储单元编为一组,并赋予一个号码,称为地址。不同的字单元具有不同的地址,从而在进行读写操作时,可以按照地址选择欲访问(读写操作)的单元。字单元也称为地址单元。

地址译码电路实现地址的选择。在大容量的存储器中,通常采用双译码结构,即将输入地址分为行地址和列地址两部分,分别由行、列地址译码电路译码。行、列地址译码电路的输出作为存储矩阵的行、列地址选择线,由它们共同确定欲选择的地址单元。地址单元的个数 N 与二进制地址码的位数 n 满足关系式 $N=2^n$ 。

对于图 8.1.5 所示的存储矩阵,256 个字需要 8 位二进制地址码($A_7 \sim A_0$)。地址译码有多种形式。例如,可以将地址码 $A_7 \sim A_0$ 的低 5 位 $A_4 \sim A_0$

作为行地址,经过 5 线-32 线译码电路,产生 32 根行地址选择线;地址码的高 3 位 $A_7 \sim A_5$ 作为列译码输入,产生 8 根列地址选择线。只有被行地址选择线和列地址选择线同时选中的单元,才能被访问。例如,若输入地址码 $A_7 \sim A_0$ 为 00011111 时, X_{31} 和 Y_0 输出有效电平,位于 X_{31} 和 Y_0 交叉处的字单元可以进行读出或写入操作,而其余任何字单元都不会被选中。

(3) 输入/输出控制电路

图 8.1.6 给出了一个简单的输入/输出控制电路。在系统中为了便于控制,电路不仅有读/写控制信号 R/\overline{W} , 还有片选控制信号 CS 。当片选信号有效时,芯片被选中,可以进行读/写操作,否则芯片不工作。片选信号仅解决芯片是否工作的问题,而芯片的读、写操作则由读/写控制信号 R/\overline{W} 决定。

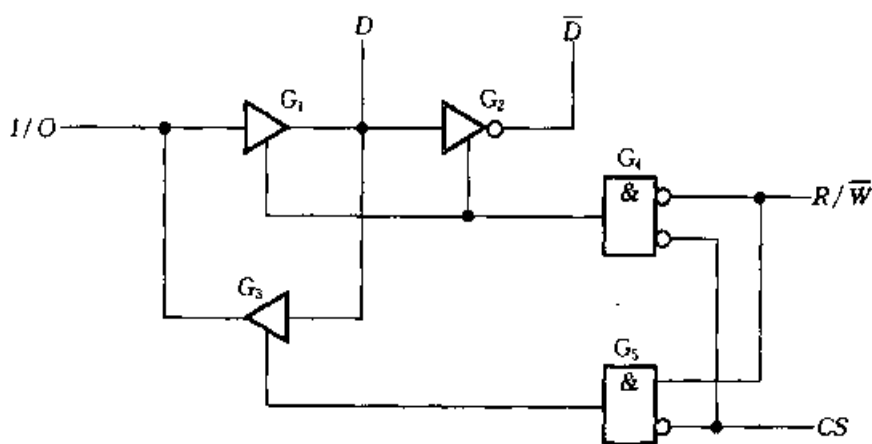


图 8.1.6 输入/输出控制电路

在图 8.1.6 中,当片选信号 $CS = 1$ 时, G_5 、 G_4 输出为 0, 三态门 G_1 、 G_2 、 G_3 均处于高阻状态,输入/输出 (I/O) 端与存储器内部完全隔离,存储器禁止读/写操作,即不工作;而当 $CS = 0$ 时,芯片被选通,根据读/写控制信号 R/\overline{W} 的高低,执行读或写操作。当 $R/\overline{W} = 1$ 时, G_5 输出高电平, G_3 被打开,于是被选中的单元所存储的数据出现在 I/O 端,存储器执行读操作;反之, $R/\overline{W} = 0$ 时, G_4 输出高电平, G_1 、 G_2 被打开,此时加在 I/O 端的数据以互补的形式出现在内部数据线上,并被存入到所选中的存储单元,存储器执行写操作。

3. RAM 的操作与定时

为了保证存储器准确无误地工作,加到存储器的地址、数据和控制信号必须遵守几个时间边界条件。下面以静态 RAM 为例加以说明。

图 8.1.7 示出了读出过程的定时关系。读出过程操作如下:

- (1) 欲读取单元的地址加到存储器的地址输入端;
- (2) 加入有效的片选信号 CS ;

(3) 在 R/\overline{W} 线上加高电平, 经过一段延时后, 所选择单元的内容出现在 I/O 端。

(4) 让片选信号 CS 无效, I/O 端呈高阻态, 本次读出结束。

由于地址缓冲器、译码器及输入/输出电路存在延时, 在地址信号加到存储器上之后, 必须等待一段时间 t_{AA} , 数据才能稳定地传输到数据输出端, 这段时间称为地址存取时间。如果在 RAM 的地址输入端已经有稳定地址的条件下, 加入片选信号, 从片选信号有效到数据稳定输出, 这段时间间隔记为 t_{ACS} 。显然在进行存储器读操作时, 只有在地址和片选信号加入, 且分别等待 t_{AA} 和 t_{ACS} 以后, 被读单元的内容才能稳定地出现在数据输出端, 这两个条件必须同时满足。图中 t_{RC} 为读周期, 它表示该芯片连续进行两次读操作必须的时间间隔。

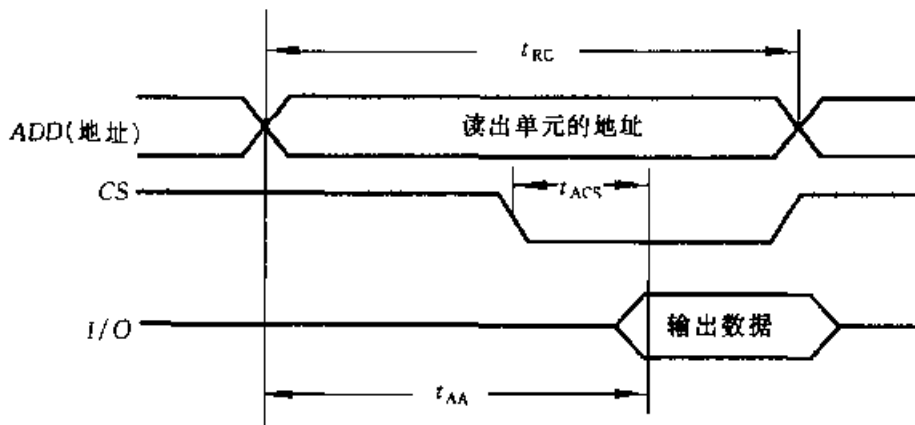


图 8.1.7 读操作时序图

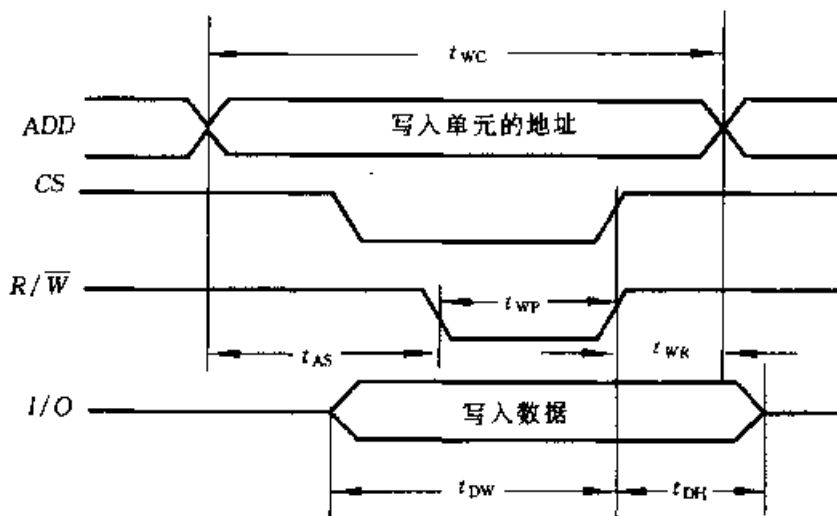


图 8.1.8 写操作时序图

写操作的定时波形如图 8.1.8 所示。写操作过程如下：

- (1) 将欲写入单元的地址加到存储器的地址输入端；
- (2) 在片选信号 CS 端加上有效逻辑电平,使 RAM 工作；
- (3) 将待写入的数据加到数据输入端；
- (4) 在 R/\overline{W} 线上加入低电平,进入写工作状态；
- (5) 使片选信号无效,数据输入线回到高阻状态。

由于地址改变时,新地址的稳定要经过一段时间,如果在这段时间内加入写控制信号(即 R/\overline{W} 变低),可能将数据错误地写入其他单元。为了防止这种情况出现,在写控制信号有效前,地址必须稳定一段时间 t_{AS} ,这段时间称为地址建立时间。同时,在写信号失效后,地址至少还要维持一段写恢复时间 t_{WR} 。为了保证速度最慢的存储器芯片的写入,写信号有效的时间不得小于写脉冲宽度 t_{WP} 。此外,对于写入的数据,应在写信号失效前 t_{Dw} 时间内保持稳定,且在写信号失效后继续保留 t_{DH} 时间。在时序图中还给出了写周期 t_{WC} ,它反映了连续进行两次写操作所需要的最小时间间隔。对于大多数静态半导体存储器来说,读周期和写周期是相等的,一般为十几到几十 ns。

由于动态 RAM 的地址大都采用行、列地址分时送人的方法,因此,其读、写操作定时与静态 RAM 稍有不同,详见 8.1.3 节。

8.1.2 RAM 存储容量的扩展

在数字系统或计算机中,单个存储器芯片往往不能满足存储容量的要求,因此,必须把若干个存储器芯片连接在一起,以扩展存储容量。扩展存储容量的方法可以通过增加字长(位数)或字数来实现。存储器的字数通常采用 K、M 或 G 为倍率,其中 $1K = 2^{10} = 1024$, $1M = 2^{20} = 1024K$, $1G = 2^{30} = 1024M$ 。

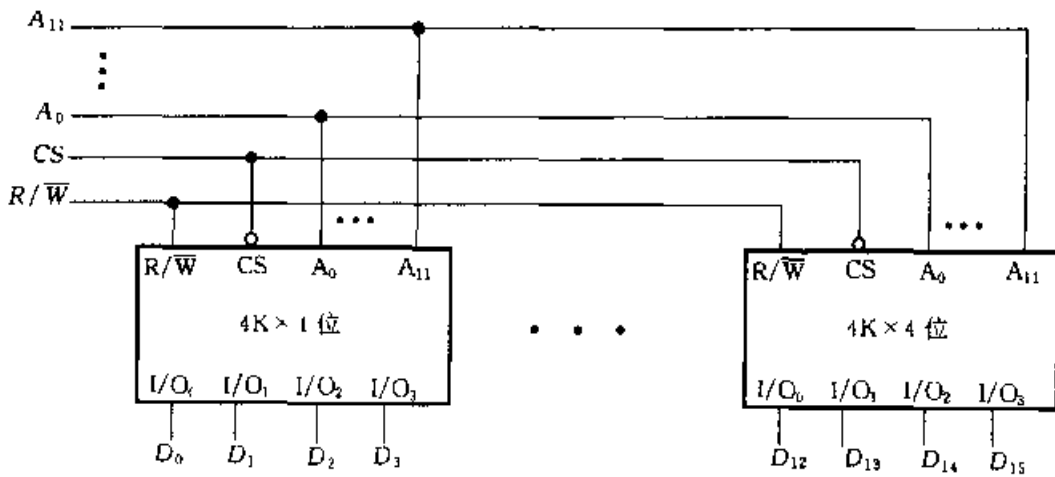
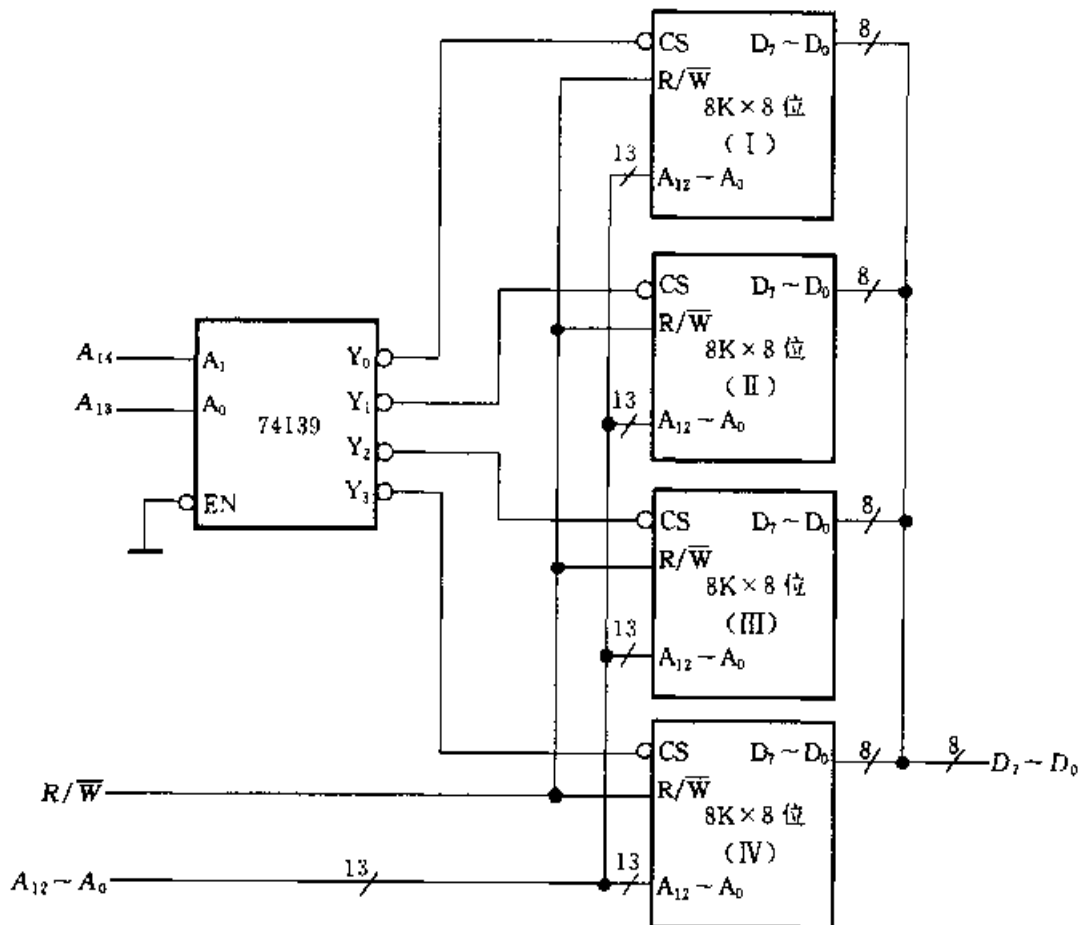
1. 字长(位数)的扩展

通常 RAM 芯片的字长为 1 位、4 位、8 位、16 位和 32 位等。当实际的存储器系统的字长超过 RAM 芯片的字长时,需要对 RAM 实行位扩展。

位扩展可以利用芯片的并联方式实现,即将 RAM 的地址线、读/写控制线和片选信号对应地并联在一起,而各个芯片的数据输入/输出端作为字的各个位线。例如,用 4 个 $4K \times 4$ 位 RAM 芯片可以扩展成 $4K \times 16$ 位的存储系统,如图 8.1.9 所示。

2. 字数的扩展

字数的扩展可以利用外加译码器,控制存储器芯片的片选输入端来实现。例如,利用 2 线-4 线译码器 74139 将 4 个 $8K \times 8$ 位的 RAM 芯片扩展为 $32K \times 8$ 位的存储器系统。扩展方式如图 8.1.10 所示。图中,存储器扩展所要增加的地址线 A_{14} 、 A_{13} 与译码器的 74139 的输入相连,译码器的输出 $Y_0 \sim Y_3$ 分别接

图 8.1.9 用 $4K \times 4$ 位 RAM 芯片构成 $4K \times 16$ 位的存储器系统图 8.1.10 用 $8K \times 8$ 位 RAM 芯片构成 $32K \times 8$ 位的存储器系统

至 4 片 RAM 的片选信号控制端 CS, 这样, 当输入一个地址码 ($A_{14} \sim A_0$) 时, 只有一片 RAM 被选中, 从而实现了字的扩展。芯片的地址分布, 如表 8.1.1 所示。

表 8.1.1 32K×8 位存储器系统的地址分配表

各 RAM 芯片	译码器有效输出端	扩展的地址输入端 $A_{14} A_{13}$	8K×8 位 RAM 芯片地址输入端	对应的十六进制地址码
			$A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$	
I	Y_0	0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0000H
			0 0 0 0 0 0 0 0 0 0 0 0 0 1	0001H
			0 0 0 0 0 0 0 0 0 0 0 0 1 0	0002H
			⋮	⋮
			1 1 1 1 1 1 1 1 1 1 1 1 1 1	1FFFH
II	Y_1	0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0	2000H
			0 0 0 0 0 0 0 0 0 0 0 0 0 1	2001H
			0 0 0 0 0 0 0 0 0 0 0 0 1 0	2002H
			⋮	⋮
			1 1 1 1 1 1 1 1 1 1 1 1 1 1	3FFFH
III	Y_2	1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	4000H
			0 0 0 0 0 0 0 0 0 0 0 0 0 1	4001H
			0 0 0 0 0 0 0 0 0 0 0 0 1 0	4002H
			⋮	⋮
			1 1 1 1 1 1 1 1 1 1 1 1 1 1	5FFFH
IV	Y_3	1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0	6000H
			0 0 0 0 0 0 0 0 0 0 0 0 0 1	6001H
			0 0 0 0 0 0 0 0 0 0 0 0 1 0	6002H
			⋮	⋮
			1 1 1 1 1 1 1 1 1 1 1 1 1 1	7FFFH

实际应用中,常将两种方法相互结合,以达到字和位均扩展的要求。可见,无论需要多大容量的存储器系统,均可利用容量有限的存储器芯片,通过位数和字数的扩展来构成。

8.1.3 RAM 举例

目前,市场上的 RAM 品种繁多,且没有一个统一的命名标准。不同厂商生产的功能相同的产品,其型号也不尽相同。这里给出了 MOTOROLA 公司生产的 MCM6264 和 NEC 公司生产的 μ PD41256 两个芯片例子,例子展示的内容为器件手册中内容的一部分。

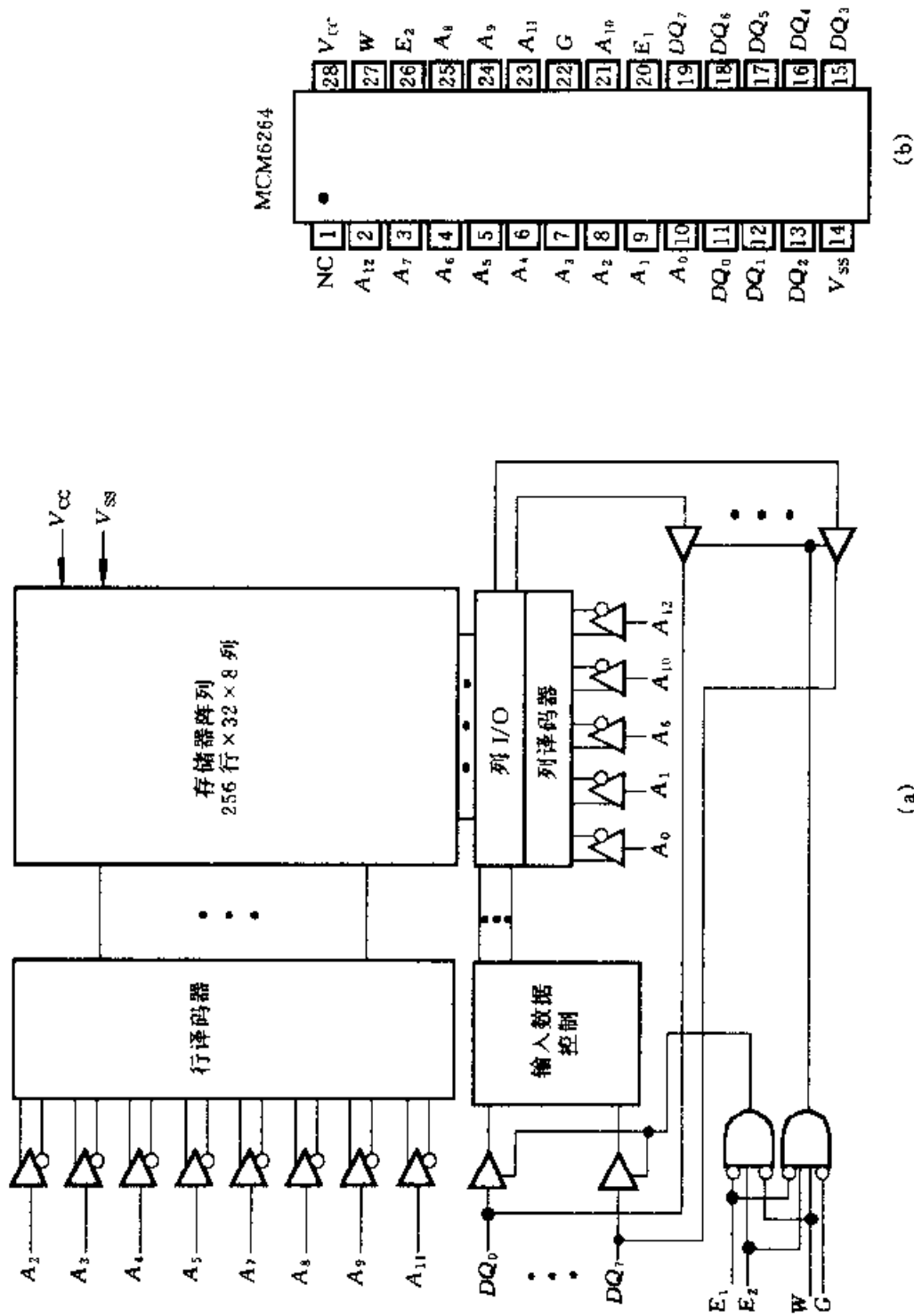


图 8.1.11 MCM6264 SRAM 的方框图

(a) MCM6264 的逻辑结构图 (b) 引脚排列图

图中: $A_0 \sim A_{12}$ 为地址输入, $DQ_0 \sim DQ_7$ 为数据输入/输出, W 为写允许, E_1 、 E_2 为片选, G 为无效引脚。

1. MCM6264

MCM6264是8K×8位的SRAM。该芯片采用20引脚塑料双列直插封装，单电源+5V供电。图8.1.11给出了它的逻辑结构框图和引脚排列图，表8.1.2为功能表，图8.1.12为读、写操作时序图。

表 8.1.2 功能表

E_1	E_2	G	W	方式	I/O	周期
H	x	x	x	无选择	高阻态	—
x	L	x	x	无选择	高阻态	—
L	H	H	H	输出禁止	高阻态	—
L	H	L	H	读	D_0	读
L	H	x	L	写	D_1	写

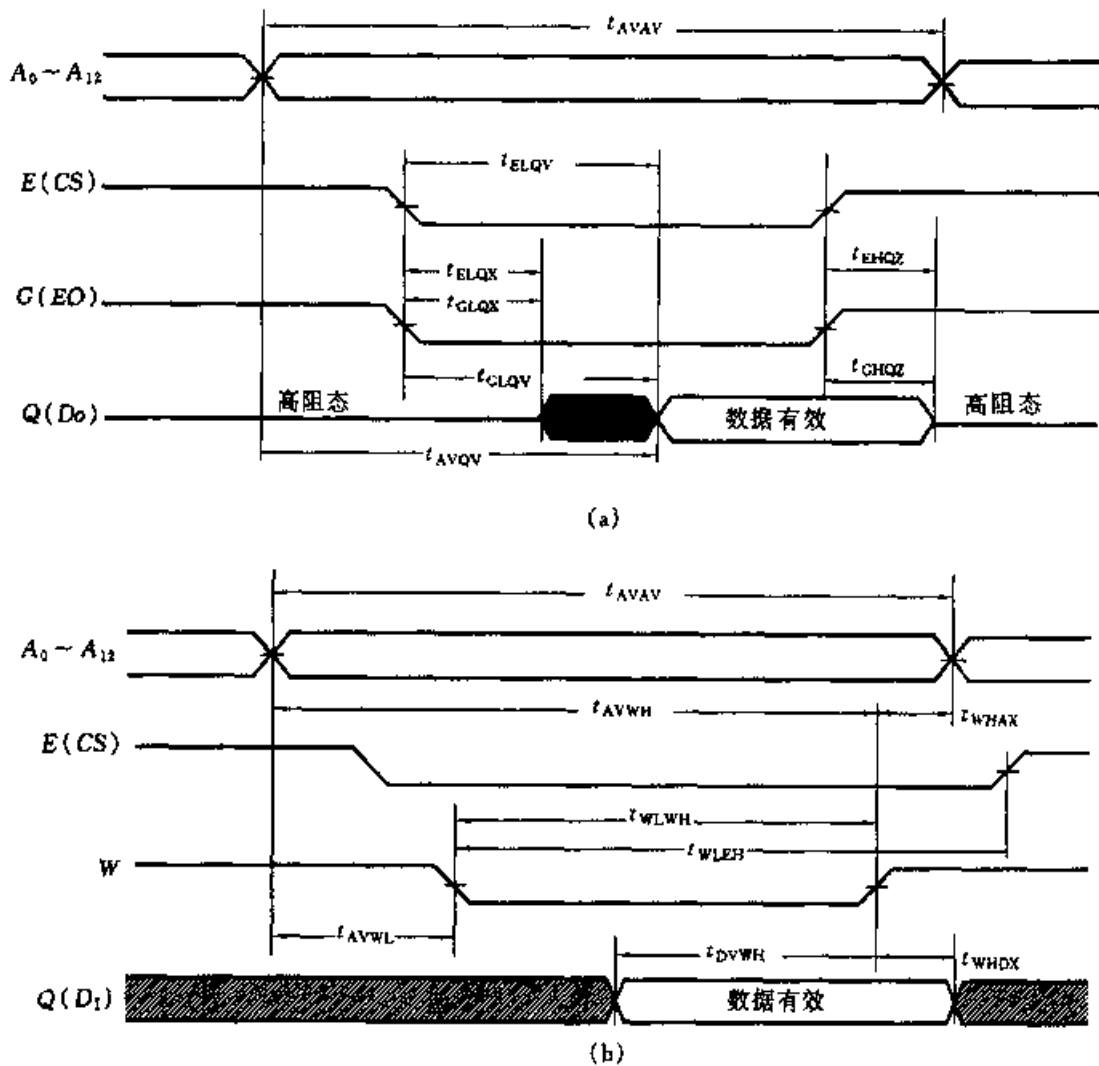


图 8.1.12 读、写操作时序图

(a) 读操作时序图 (b) 写操作时序图

注：(1) 图中用 E 表示 E_1 和 E_2 ，且 $E = E_1, E_2$ 和 E_1 极性相反。(2) MCM6264P20 读写定时参数见本章附录中附表 8.1 和附表 8.2。

2. μ PD41256

μ PD41256 是 $256K \times 1$ 位的 DRAM 芯片。由于 DRAM 的集成度很高, 存储容量大, 因此需要较多的地址线。但引线数目增多, 势必会加大芯片尺寸, 为

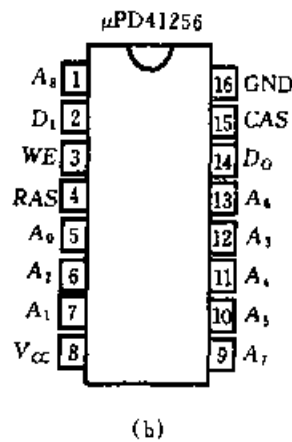
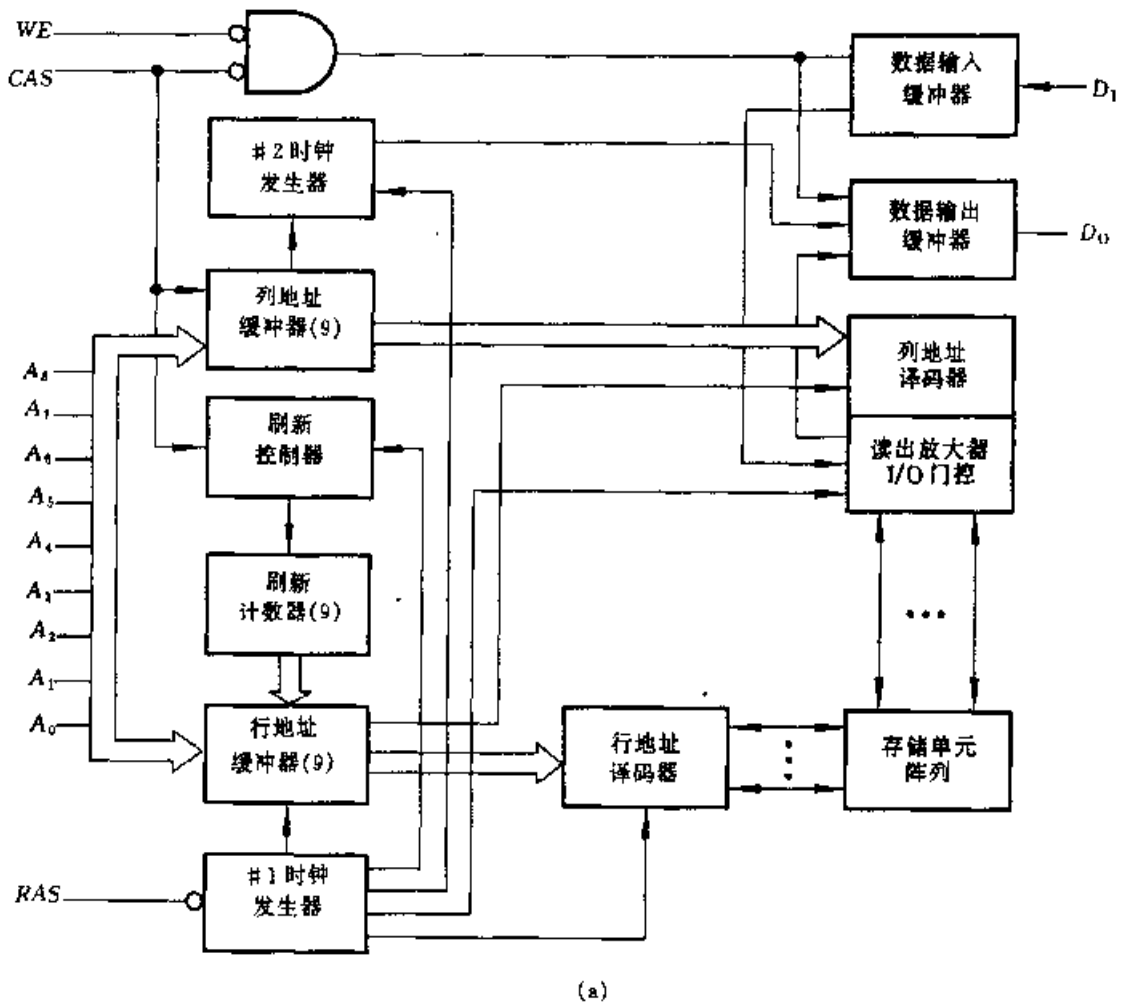
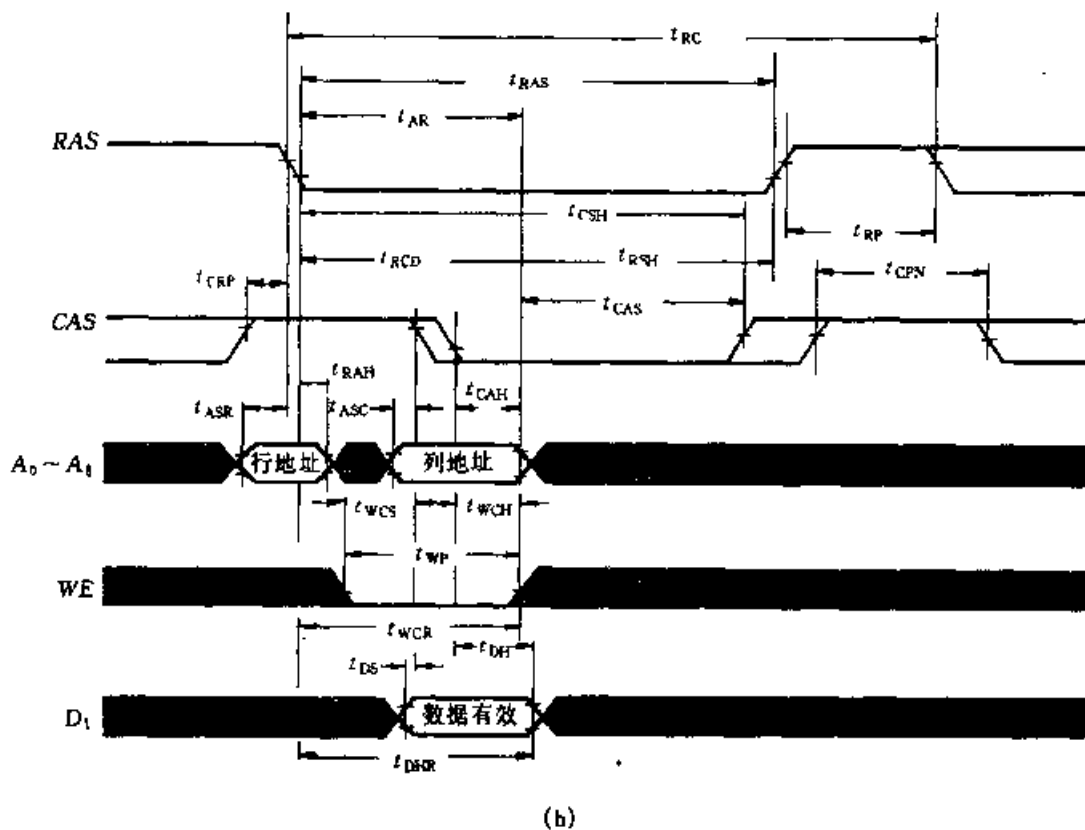
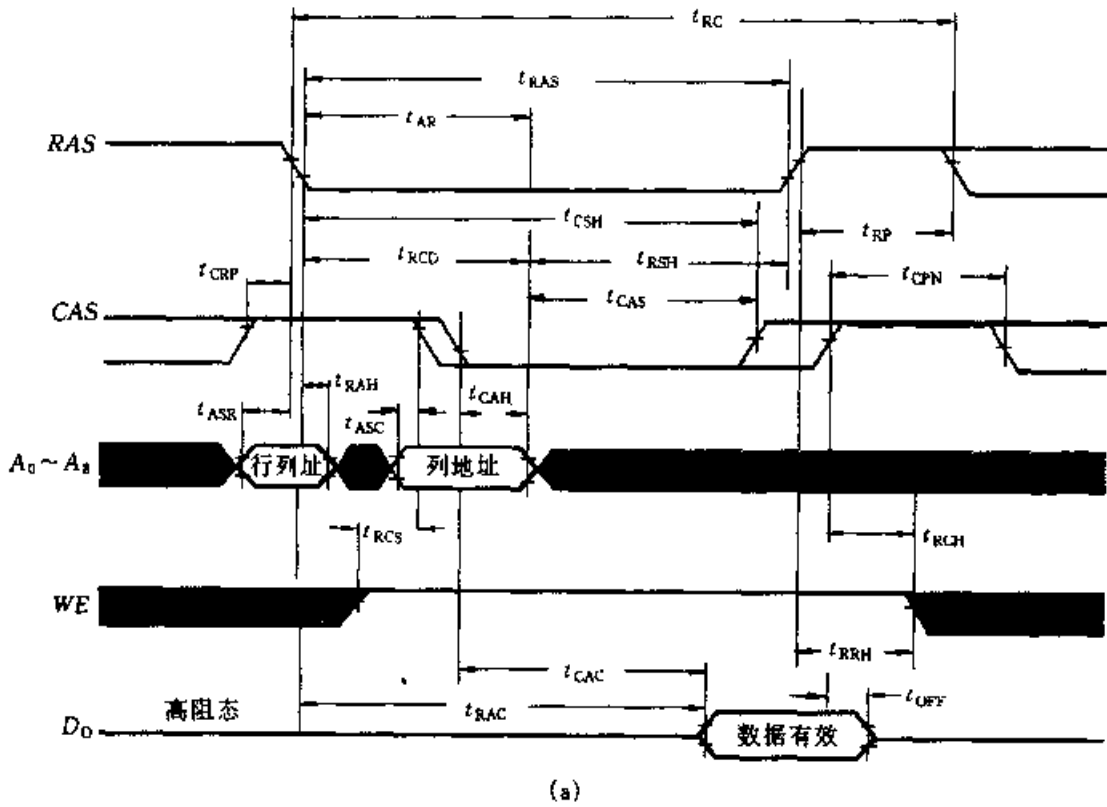


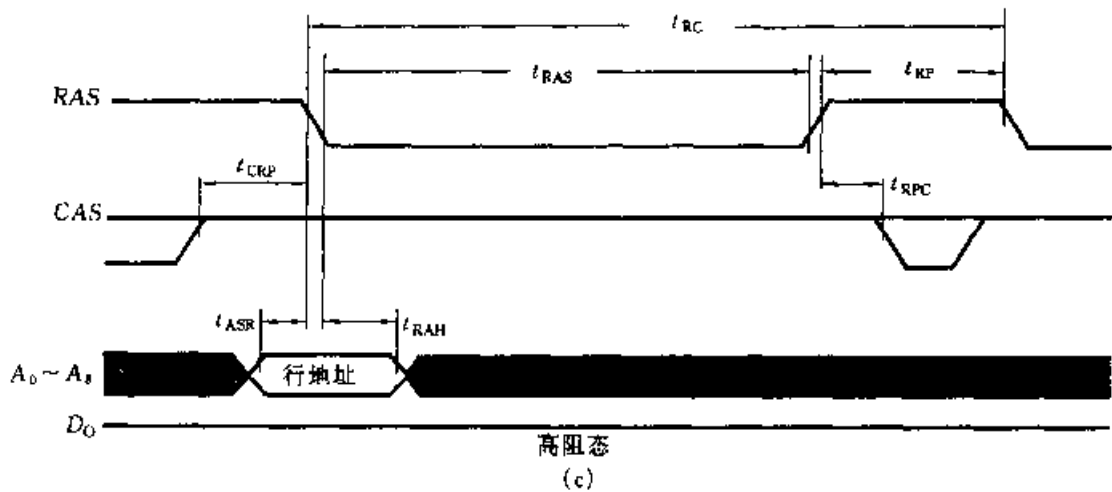
图 8.1.13 μ PD41256 DRAM 的方框图

(a) μ PD41256 的逻辑结构图 (b) 引脚排列图

图中: $A_0 \sim A_8$ 为地址输入, D_1 为数据输入, D_0 为数据输出, WE 为写允许, RAS 为行地址选通, CAS 为列地址选通。

了解决这一矛盾, DRAM 大都采用行、列地址分时送入的方法。μPD41256 逻辑结构框图如图 8.1.13a 所示, 它具有独立的数据输入、输出线, 9 根地址线, 18 位地址分两次输入。芯片内部设有行、列两个地址锁存器, 分别用于锁存行、列地



图 8.1.14 μ PD41256 操作时序图

(a) 读操作时序图 (b) 写操作时序图 (c) RAS 只刷新操作时序图

注： μ PD41256 读写定时参数见本章附录中附表 8.3

址。行、列地址先后由行地址选通 RAS^① 和列地址选通 CAS^② 信号控制,送入各自的锁存器。此外, μ PD41256 没有单独的片选控制信号,片选工作由 RAS 提供。芯片内部还设有时钟发生器,用于产生内部时钟信号,这些时钟信号控制芯片的读、写和刷新等操作。时钟发生器受 RAS 和 CAS 制约。 μ PD41256 的读、写及刷新有多种操作方式,这里各举出一种方式示于图 8.1.14 中。

复习思考题

8.1.1 在存储器的结构中,什么叫“字”? 什么叫“字长”? 如何标注存储器的容量?

8.1.2 DRAM 中存储的数据如果不进行周期性的刷新,其数据将会丢失;而 SRAM 中存储的数据无须刷新,只要电源不断电就可以永久保存,为什么?

8.1.3 一般情况下,DRAM 的集成度比 SRAM 的集成度高,为什么?

8.1.4 对于一个存储容量为 $32\text{K} \times 16$ 位的 RAM,下列哪些说法是正确的?

- (1) 该存储器有 512K 个存储单元;
- (2) 每次可同时读/写 8 位数据;
- (3) 该存储器有 16 根地址线;
- (4) 该存储器有 32 根数据线;
- (5) 该存储器的字长为 16 位;
- (6) 访问该存储器的某个存储单元时需要 15 位地址码;
- (7) 该存储器的十六进制数地址范围是 $0000\text{H} \sim \text{FFFFH}$ 。

① Row Address Strobe 的缩写。

② Column Address Strobe 的缩写。

8.1.5 已知某计算机系统的 CPU 读/写数据的时间为 100 ns, 试判断能否用 $\mu\text{PD}41256-10$ DRAM 作为该系统的数据存储器。($\mu\text{PD}41256-10$ 读写定时参数见本章附录中附表 8.3.)

8.1.6 用容量为 $16\text{K} \times 8$ 位存储器芯片构成 1 个 $64\text{K} \times 8$ 位的存储系统, 需要多少根地址线? 多少根数据线? 多少个 $16\text{K} \times 8$ 位的存储器芯片?

8.2 只读存储器(ROM)

前面讨论的 RAM, 无论是 SRAM 还是 DRAM 都具有易失性, 即当电源断电后, 存储的数据便随之消失。然而在数字系统及计算机中, 常常需要储存一些固定不变的数据, 例如数据转换表以及保证计算机能正常运行的操作系统程序等, 希望断电后, 这些数据能够保留下来, 通电后又可以继续使用。采用 ROM 即可满足上述要求。

与 RAM 不同, ROM 一般由专用的装置写入数据, 数据一旦写入, 不能随意改写, 在切断电源之后, 数据也不会消失, 即具有非易失性。ROM 器件的种类很多, 从制造工艺上看, 有二极管 ROM、双极型 ROM 和 MOS 型 ROM 三种, 按存储内容存入方式的不同, 又可以分成固定 ROM 和可编程 ROM。可编程 ROM 又可以细分为一次可编程存储器 PROM^①、光可擦除可编程存储器 EPROM^②、电可擦除可编程存储器 E²PROM^③ 和快闪存储器^④ 等。

ROM 的结构与 RAM 类似, 本节主要介绍以上几种 ROM 存储数据的基本原理。

固定 ROM 又称为掩模 ROM, 这种 ROM 在制造时, 生产厂家利用掩模技术把数据写入存储器中, 一旦 ROM 制成, 其存储的数据也就固定不变了。而 PROM 在出厂时, 存储内容全为 1 (或者全为 0), 用户可以根据自己的需要, 利用通用或专用的编程器, 将某些单元改写为 0 (或 1)。图 8.2.1 是一个简单的 PROM 结构示意图, 它采用熔断丝结构, 译码器输出高电平有效。出厂时, 熔丝是连通的, 也就是全部存储单元为 1, 如欲使某些单元改写为 0, 只要通过编程, 并给这些单元通以足够大的电流将熔丝烧断即可。熔丝烧断后不能恢复, 因此, PROM 只能改写一次。

EPROM 是采用浮栅技术生产的可编程存储器, 它的存储单元多采用 N 沟道叠栅 MOS 管 (SIMOS^⑤), 其结构及符号如图 8.2.2a 所示。除控制栅外, 还有一个没有外引线的栅极, 称为浮栅。当浮栅上没有电荷时, 给控制栅 (接在行选

① Programmable Read-Only Memory 的缩写

② Erasable Programmable Read-Only Memory 的缩写。

③ Electrical Erasable Programmable Read-Only Memory 的缩写。

④ Flash Memory。

⑤ Stacked gate avalanche Injection Metal Oxide Semiconductor 的缩写。

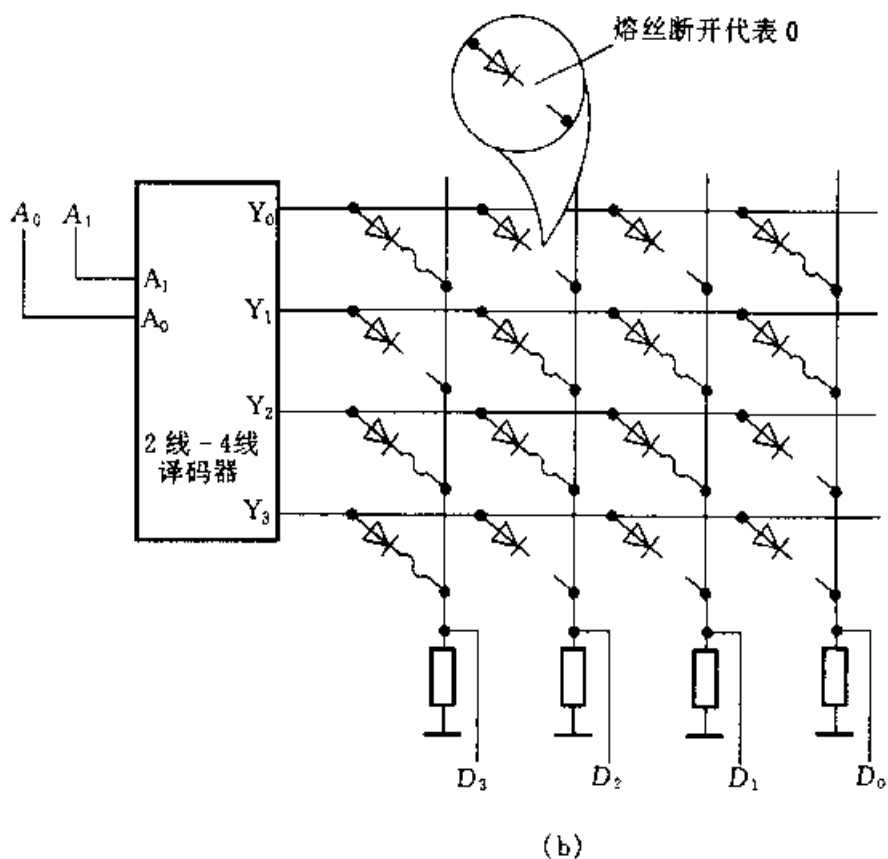
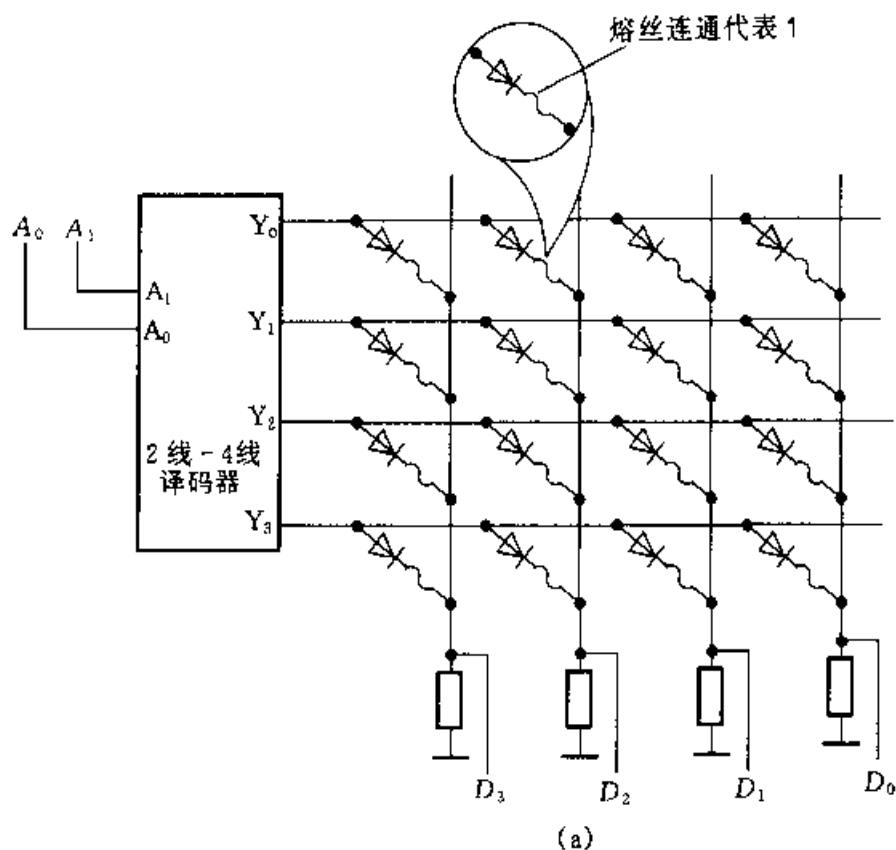


图 8.2.1 PROM 结构示意图

(a) 编程前的 PROM (b) 编程后的 PROM

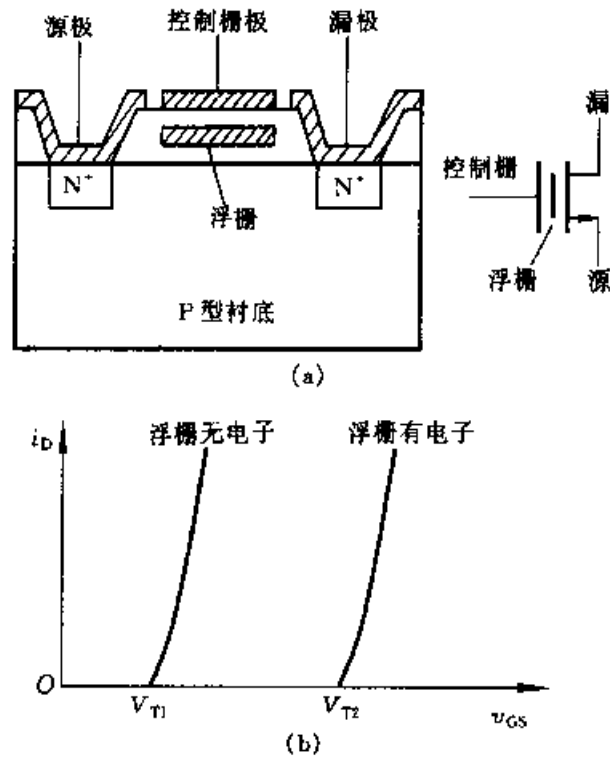


图 8.2.2 叠栅 MOS 管

(a) 结构及符号 (b) 浮栅上累积电子与开启电压的关系

择线上)加上控制电压, MOS 管导通;而当浮栅上带有负电荷时,则衬底表面感应的是正电荷,这使得 MOS 管的开启电压变高(如图 8.2.2b 所示),如果给控制栅加上同样的控制电压, MOS 管仍处于截止状态。由此可见, SIMOS 管可以利用浮栅是否累积有负电荷来存储二值数据。

在写入数据前,浮栅是不带电的,要使浮栅带负电荷,必须在 SIMOS 管的漏、栅极加上足够高的电压(如 25 V),使漏极及衬底之间的 PN 结反向击穿,产生大量的高能电子。这些电子穿过很薄的氧化绝缘层堆积在浮栅上,从而使浮栅带有负电荷。当移去外加电压后,浮栅上的电子由于没有放电回路,所以能够长期保存。当用紫外线或 X 射线照射时,浮栅上的电子形成光电流而泄放,从而恢复写入前的状态。照射一般需要 15 到 20min(分钟)。为了便于照射擦除,芯片的封装外壳装有透明的石英盖板。EPROM 的擦除为一次全部擦除,其数据写入需要通用或专用的编程器。

E^2 PROM 也是采用浮栅技术生产的可编程存储器,其构成存储单元的 MOS 管的结构如图 8.2.3 所示。它与叠栅 MOS 管的不同之处在于浮栅延长区与漏区 N^+ 之间的交叠处有一个厚度约为 80 Å(埃)的薄绝缘层,当漏极接地,控制栅加上足够高的电压时,交叠区将产生一个很强的电场,在强电场作用下,电子通过绝缘层到达浮栅,使浮栅带负电荷。这一现象称为“隧道效应”,因此,该 MOS 管也称为隧道 MOS 管。相反,当控制栅接地,漏极加一正电压,则产生与上述

相反的过程,即浮栅放电。与 SIMOS 管相比,隧道 MOS 管也是利用浮栅是否累积有负电荷来存储二值数据的,不同的是隧道 MOS 管是用电擦除的,并且擦除的速度要快得多(一般为毫秒数量级)。

E^2 PROM 电擦除的过程就是改写过程,它是以字为单位进行的。 E^2 PROM 具有 ROM 的非易失性,又具备类似 RAM 的功能,可以随时改写(可重复擦写 1 万次以上)。目前,大多数 E^2 PROM 芯片内部都备有升压电路。因此,只需提供单电源供电,便可进行读、擦除/写操作,这为数字系统的设计和在线调试提供了极大的方便。

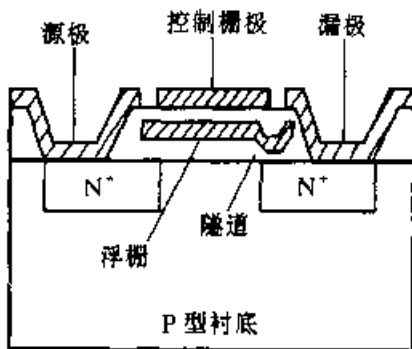


图 8.2.3 隧道 MOS 管剖面结构示意图

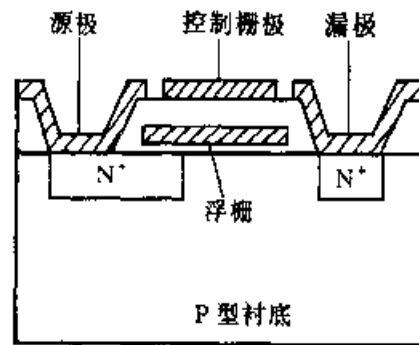


图 8.2.4 快闪存储器存储单元 MOS 管剖面结构示意图

快闪存储器存储单元的 MOS 管结构与 SIMOS 管类似,如图 8.2.4。但有两点不同,一是快闪存储器存储单元 MOS 管的源极 N^+ 区大于漏极 N^+ 区,而 SIMOS 管的源极 N^+ 区和漏极 N^+ 区是对称的;二是浮栅到 P 型衬底间的氧化绝缘层比 SIMOS 管的更薄。这样,可以通过在源极上加一正电压,使浮栅放电,从而擦除写入的数据。由于快闪存储器中存储单元 MOS 管的源极是连在一起的,所以不能象 E^2 PROM 那样按字擦除,而是类似 EPROM 那样整片擦除或分块擦除。一般整片擦除只需要几秒钟,不像 EPROM 那样需要照射 15 到 20min。

快闪存储器中数据的擦除和写入是分开进行的,数据写入方式与 EPROM 相同,需要输入一个较高的电压,因此要为芯片提供两组电源。一个字的写入时间约为 $200 \mu s$,一般可以擦除/写入 100 次以上。

下面举一例子以说明 ROM 的一种简单应用。图 8.2.5 给出了一个用 ROM 实现的十进制数码显示电路。图中 8421 BCD 码接至 ROM 的地址输入线,ROM 的七根数据线依次接到七段数码显示器的 $a \sim g$ 端。这样,地址单元 0000 的内容对应七段数码 0。1001 的内容对应七段数码 9,从而实现十进制数显示。

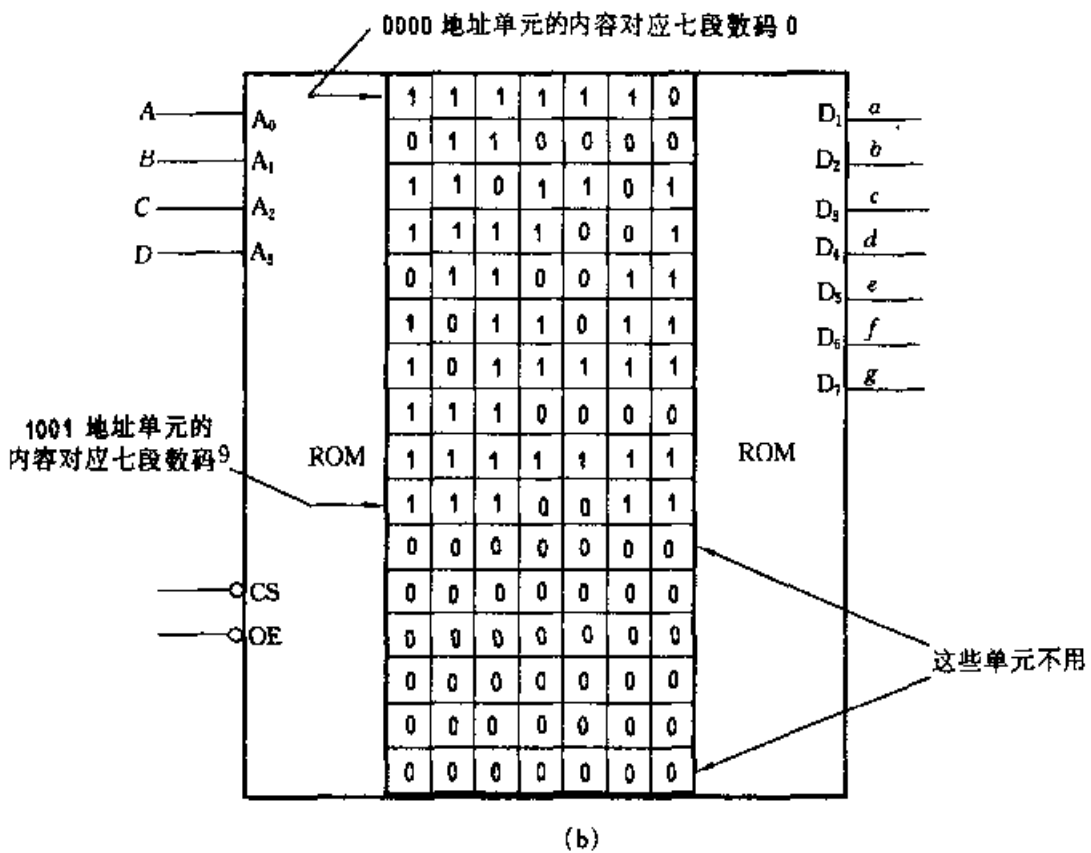
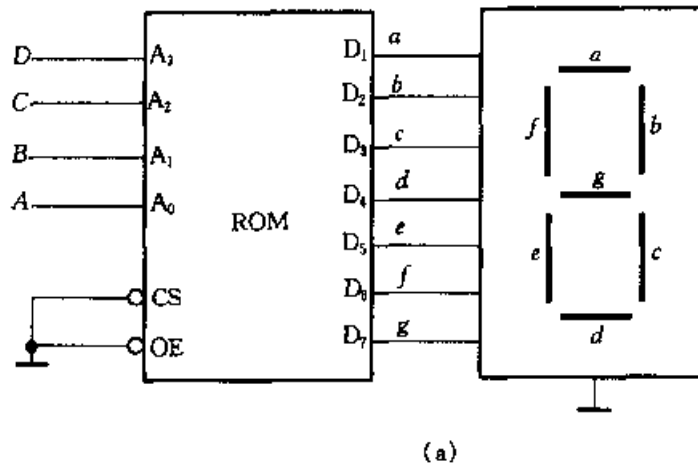


图 8.2.5 用 ROM 显示十进制数
(a) 电路原理图 (b) ROM 的内容

复习思考题

8.2.1 哪几种 ROM 具有多次擦除重写功能？哪种 ROM 的擦除过程就是数据写入过程？

8.2.2 EPROM, E²PROM 和快闪存储器有什么共同之处？它们的主要区别是什么？

8.3 可编程逻辑器件(PLD)

8.3.1 PLD 的电路表示法

前面各章已经介绍了逻辑电路的一般表示方法,但那里的方法并不适合于描述可编程逻辑器件 PLD^① 内部结构和功能。为此,本章将介绍一种新的逻辑表示法——PLD 表示法。这种表示法在芯片内部配置和逻辑图之间建立了一一对应的关系,并将逻辑图和真值表结合起来,构成了一种紧凑而易于识读的表达式。本节介绍几种比较简单的 PLD。

1. 连接方式

PLD 电路由与门和或门阵列两种基本的门阵列组成。图 8.3.1a 是一个基本的 PLD 结构图。从图中可以看出,门阵列交叉点上的连接方式共有三种情况:

- (1) 硬线连接:硬线连接是固定连接,不可以编程改变。
- (2) 可编程“接通”单元:它依靠用户编程来实现“接通”连接。

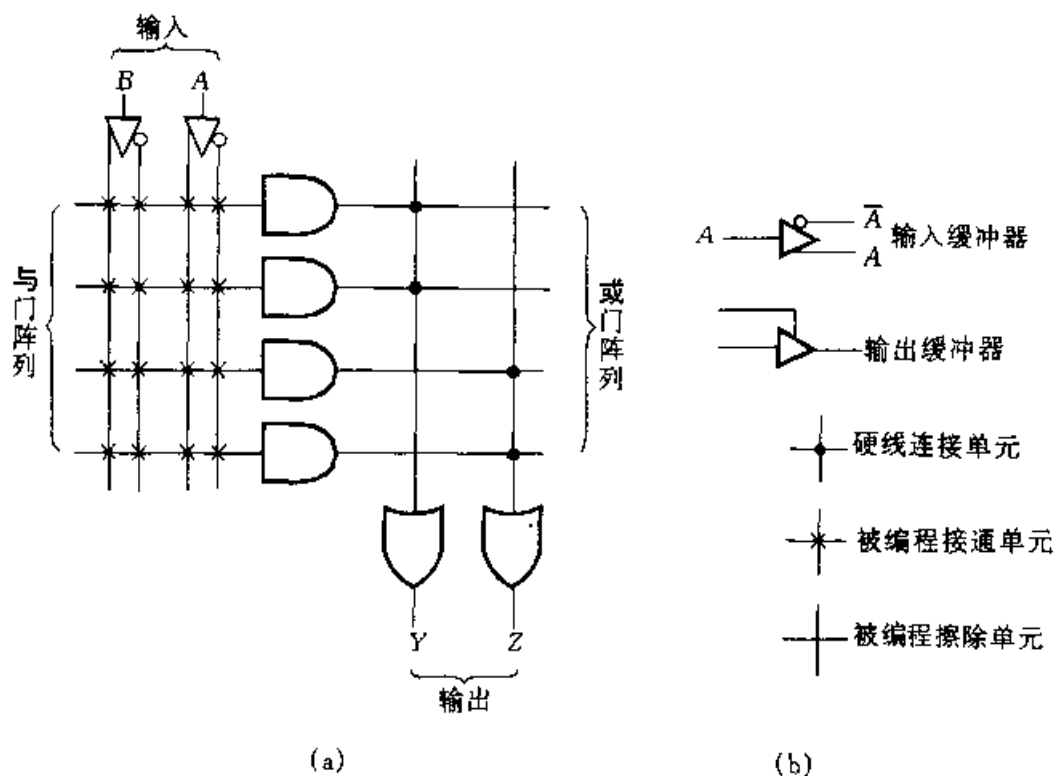


图 8.3.1 PLD 表示法

(a) 基本的 PLD 结构图 (b) PLD 连接方式

① Programmable Logic Device 的缩写。

(3) 可编程“断开”单元:编程实现断开状态。这种单元又称为被编程擦除单元。

硬线连接单元、可编程接通单元和可编程断开单元的图形符号如图 8.3.1b 所示。

2. 基本门电路的 PLD 表示法

PLD 的输入缓冲器(或反馈缓冲器)采用互补输出结构,其图形符号如图 8.3.1b 所示,其真值表列于表 8.3.1

表 8.3.1 PLD 输入缓冲器真值表

输入	输出	
	A	\bar{A}
0	0	1
1	1	0

一个 4 输入端与门的 PLD 表示法如图 8.3.2a 所示。图中 $L_1 = ABCD$, 通常把 A、B、C、D 称为输入项, L_1 称为乘积项(或简称积项)。4 输入端或门如图 8.3.2b 所示,其中 $L_2 = A + B + C + D$ 。

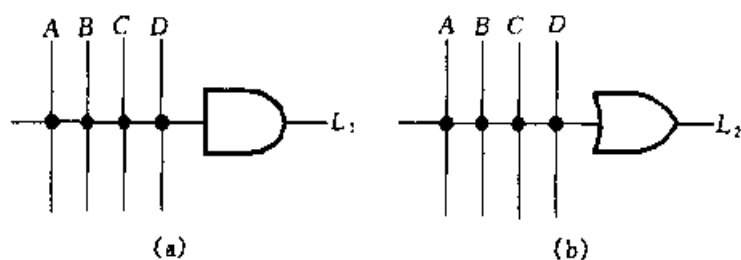


图 8.3.2 PLD 表示法的图形符号
(a) 与门符号 (b) 或门符号

在图 8.3.3 中,逻辑电路的输出变量 L_1 、 L_2 和 L_3 为

$L_1 = A \cdot \bar{A} \cdot B \cdot \bar{B} = 0$ 输入项 A、 \bar{A} 、B、 \bar{B} 被编程接通

$L_2 = 1$ 与门的所有输入项均不接通,保持“悬浮”的 1 状态

$L_3 = \bar{A} \cdot B$ 输入项 $\bar{A} \cdot B$ 硬线连接

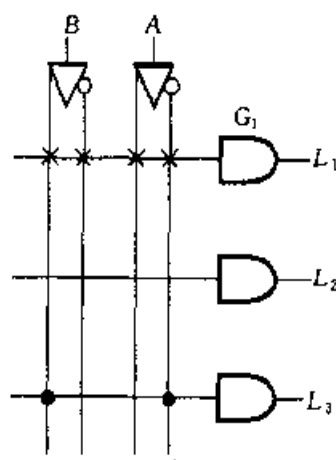


图 8.3.3 PLD 表示的与门阵列

图 8.3.3 中与门 G_1 对应的所有输入项被编程接通,输出项恒等于 0,这种状态为与门编程的默认状态,如图 8.3.4a 所示。可以在与门 G_1 中划一个“×”取代

各输入项对应的“×”，其图形符号如图 8.3.4b 所示。

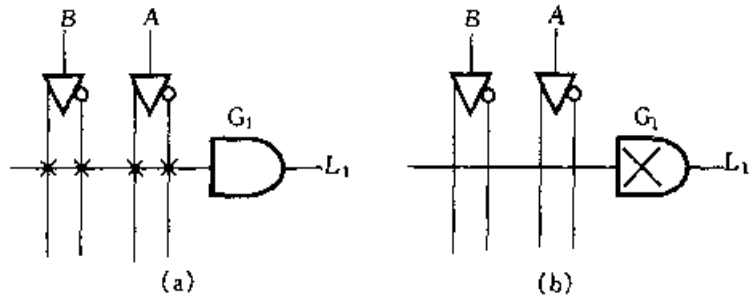


图 8.3.4 与门的默认状态
(a) 与门默认状态符号 (b) 图 a 的等效符号

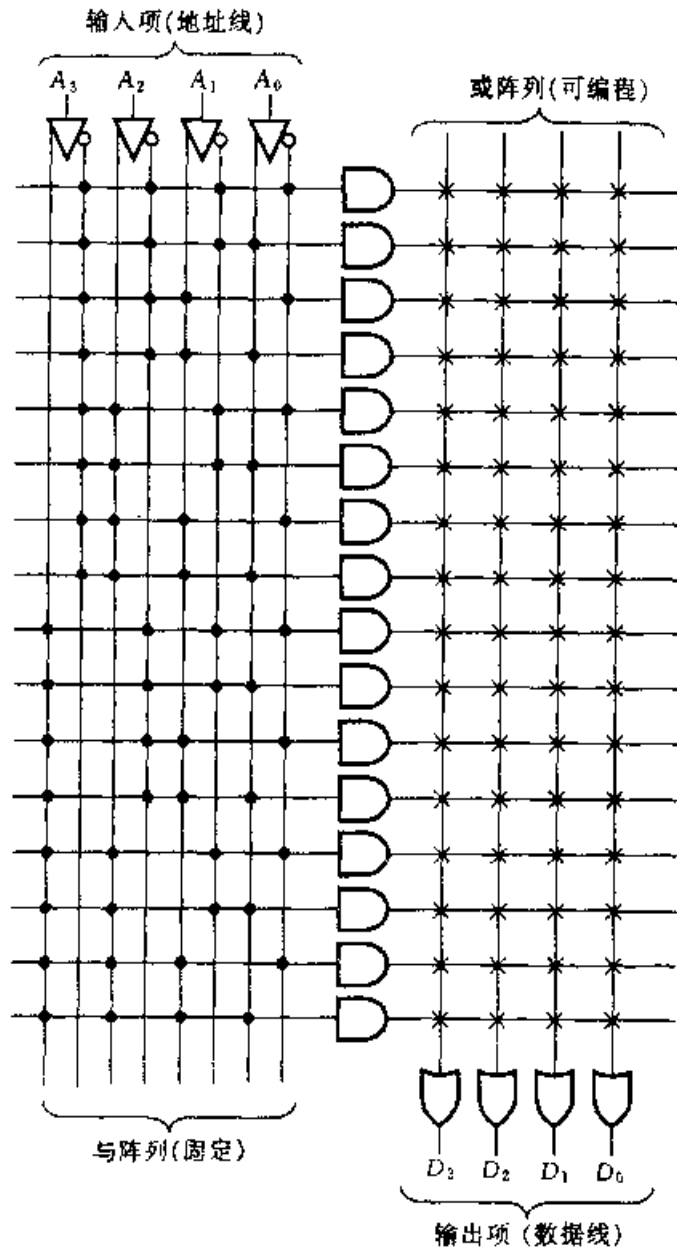


图 8.3.5 PROM 电路的 PLD 表示法

3. PROM 的 PLD 表示法

前面提到的 PROM 实质上是可编程逻辑器件,它包含一个固定连接的与门阵列(该与门阵列就是全译码的地址译码器)和一个可编程的或门阵列。相应地,4 位输入地址码的 PROM 可用图 8.3.5 所示的 PLD 表示法描述。若将图 8.3.5 中 PROM 的输入项推广到 m 个,则实现地址译码的与门数为 2^m 个,输入项数提高,与门阵列增大。而与门阵列增大,则开关时间变长,速度减慢。因此,一般只有小规模 PROM 才作为可编程逻辑器件使用,密度高达 2 百万位/片的大规模 PROM,一般只作为存储器用。

8.3.2 可编程阵列逻辑器件(PAL)简介

可编程阵列逻辑器件 PAL^① 是 70 年代后期推出的 PLD 器件。它采用可编

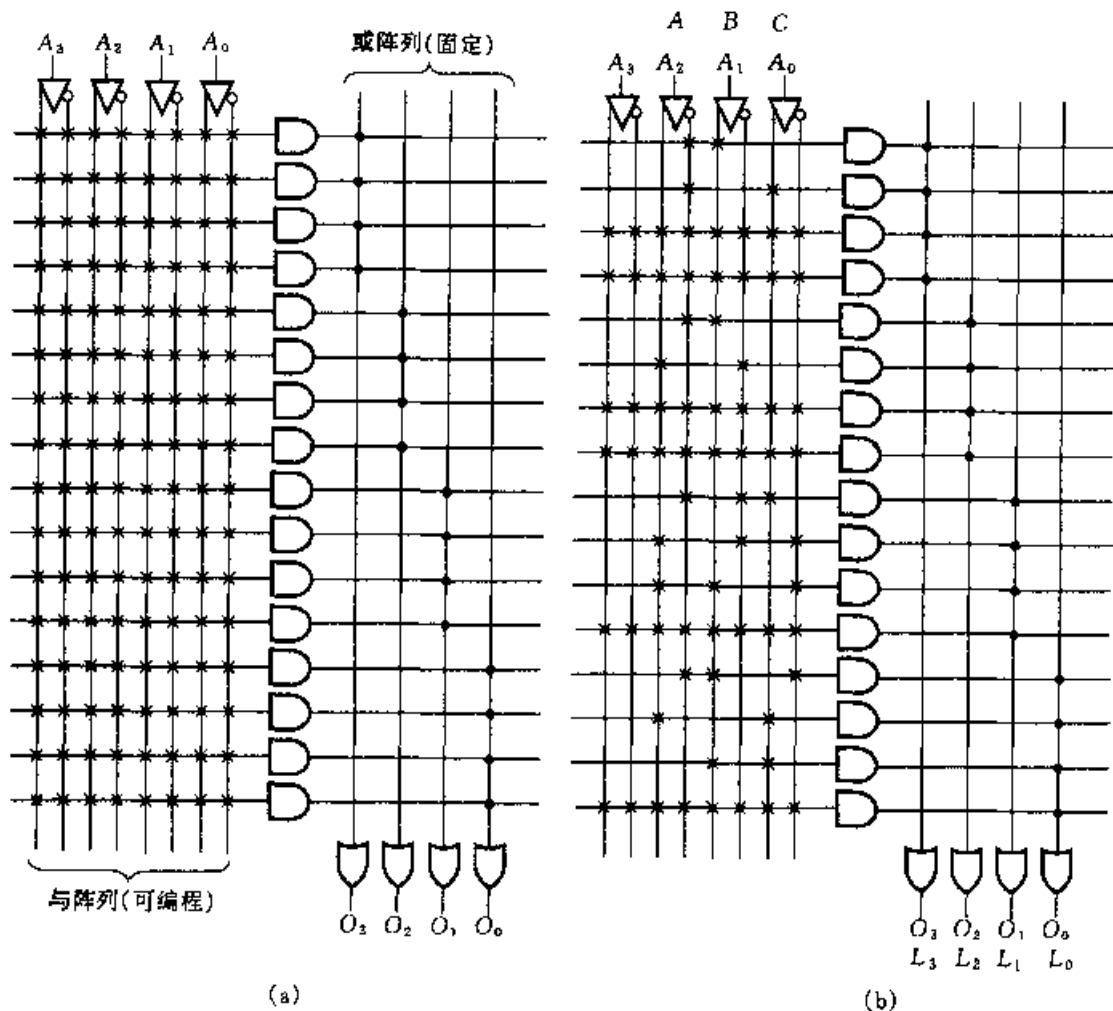


图 8.3.6 PAL 的基本结构

(a) 编程前的内部结构 (b) 编程后的内部结构

① Programmable Array Logic 的缩写。

程与门阵列和固定连接或门阵列的基本结构形式,一般采用熔丝编程技术实现与门阵列的编程。各种型号 PAL 的门阵列规模有大有小,但基本结构类似。用 PAL 门阵列实现逻辑函数时,每个输出是若干个乘积之和,即用乘积之和的形式实现逻辑函数,其中乘积项数目固定不变。图 8.3.6a 是某一个 PAL 编程前的结构图,它的每个输出信号包含 4 个乘积项。若用它来实现下列 4 个逻辑函数: $L_3 = \overline{A}B + \overline{A}C$, $L_2 = \overline{A}B + A\overline{B}$, $L_1 = \sum m(1,4,6) = \overline{A}\overline{B}C + A\overline{B}\overline{C} + AB\overline{C}$, $L_0 = \overline{A}B\overline{C} + AC + BC$,则编程后的 PAL 连接形式如图 8.3.6b 所示。

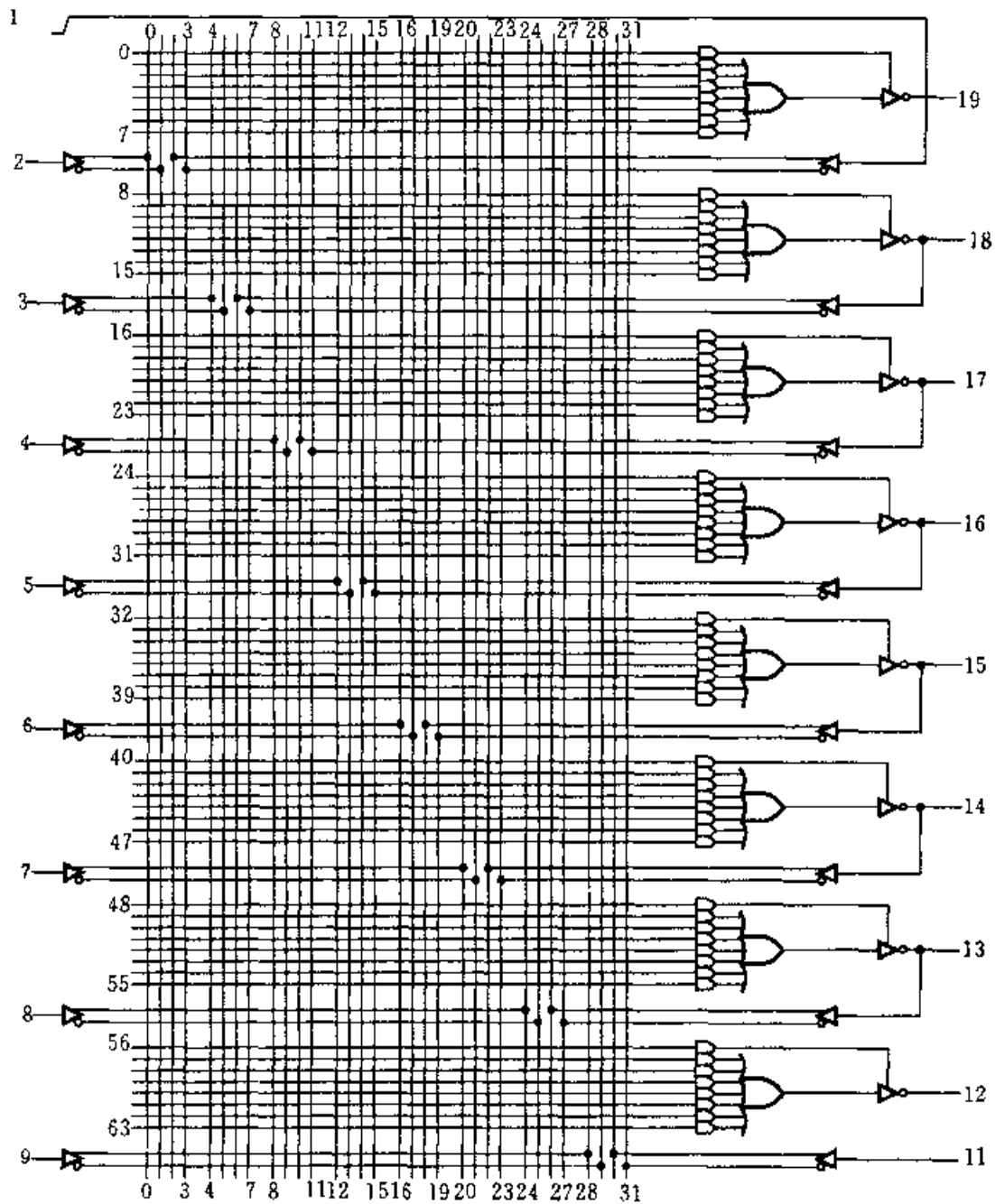


图 8.3.7 PAL16L8 的逻辑电路图

通常典型的逻辑函数要求 3~4 个乘积项,在 PAL 现有产品中,乘积项最多可达 8 个,对于大多数逻辑函数,这种结构基本上能满足要求。

PAL16L8 是一种典型的 PAL 器件,图 8.3.7 给出了它的逻辑电路图。电路内部包括 8 个与-或阵列和 8 个三态反相输出缓冲器。每个与-或阵列由 32 输入端的与门和 7 输入端的或门组成。引脚 1~9 以及引脚 11 作为输入端,用户可以根据自己的需要将引脚 13~18 用作输出端,或者是输入端。例如当引脚 14 的三态反相输出缓冲器的输出呈高阻态时,引脚 14 可以用作输入端,否则,它将用作输出端,并且低电平有效。引脚 12 和 19 只能用作输出端。

8.3.3 可编程通用阵列逻辑器件(GAL)

PAL 器件的发展给逻辑设计带来了很大的灵活性,但它还存在着不足之处。一方面,它采用熔丝连接工艺,靠熔丝烧断达到编程的目的,一旦编程便不能改写;另一方面,不同输出结构的 PAL 对应不同型号的 PAL 器件,不便于用户使用。而通用阵列逻辑器件 GAL^① 是在 PAL 器件的基础上发展起来的新一代增强型器件,它直接继承了 PAL 器件的与-或阵列结构,利用灵活的输出逻辑宏单元 OLMC^② 结构来增强输出功能,同时采用电子标签和宏单元结构字等新技术和 E²CMOS 新工艺^③,使 GAL 器件具有可擦除、可重新编程和可重新配置其结构等功能。用 GAL 器件设计逻辑系统,不仅灵活性大,而且能对 PAL 器件进行仿真,并能完全兼容。GAL 和 PAL 器件都需要通用或专用编程器进行编程。

1. GAL 的基本结构

根据 GAL 器件的门阵列结构,可以把现有的 GAL 器件分为两大类:一类与 PAL 器件基本相似,即与门阵列可编程,或门阵列固定连接,这类器件有 GAL16V8,ispGAL16Z8 和 GAL20V8 等;另一类 GAL 器件的与门阵列和或门阵列都可编程,GAL39V18 就属于这类器件。前一类 GAL 器件具有基本相同的电路结构。

通用型 GAL 包括 GAL16V8 和 GAL20V8 两种器件。其中 GAL16V8 是 20 脚器件,器件型号中的 16 表示最多有 16 个引脚作为输入端,器件型号中的 8 表示器件内含有 8 个 OLMC,最多可有 8 个引脚作为输出端。同理,GAL20V8 的最大输入引脚数是 20,GAL20V8 是 24 脚器件。

下面以 GAL16V8 为例,说明 GAL 的电路结构和工作原理。图 8.3.8 为

① Generic Array Logic 的缩写,GAL 是美国 LATTICE 公司所生产的通用阵列逻辑器件的专用商标。

② Output Logic Macro Cell 的缩写。

③ 利用浮栅技术,以 CMOS 器件为基础的电可擦除可编程生产工艺。

GAL16V8 的逻辑结构图,它由五部分组成:

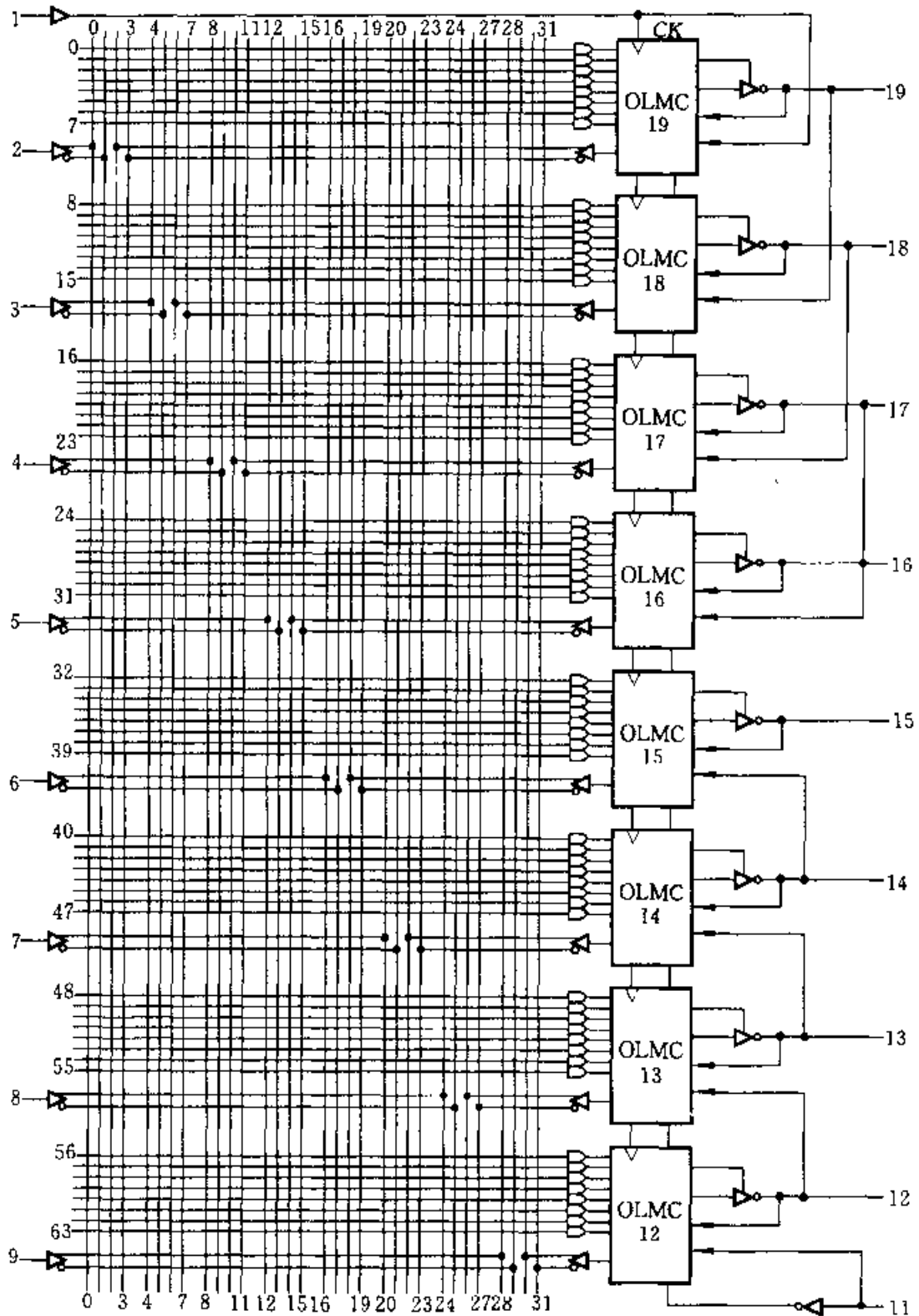


图 8.3.8 GAL16V8 的逻辑结构图

- (1) 8 个输入缓冲器(引脚 2~9 作固定输入);
- (2) 8 个输出缓冲器(引脚 12~19 作为输出缓冲器的输出);

- (3) 8 个输出逻辑宏单元(OLMC12~19,或门阵列包含在其中);
- (4) 可编程与门阵列(由 8×8 个与门构成,形成 64 个乘积项,每个与门有 32 个输入端);
- (5) 8 个输出反馈/输入缓冲器(中间一列 8 个缓冲器)。

除以上 5 个组成部分外,该器件还有 1 个系统时钟 CK 的输入端(引脚 1),一个输出三态控制端 OE (引脚 11),一个电源 V_{CC} 端和一个接地端(引脚 20 和引脚 10,图中未画出。通常 $V_{CC} = 5V$)。

2. 输出逻辑宏单元(OLMC)

GAL 的每一个输出端都对应一个输出逻辑宏单元 OLMC,它的逻辑结构示于图 8.3.9。它主要由 4 部分组成:

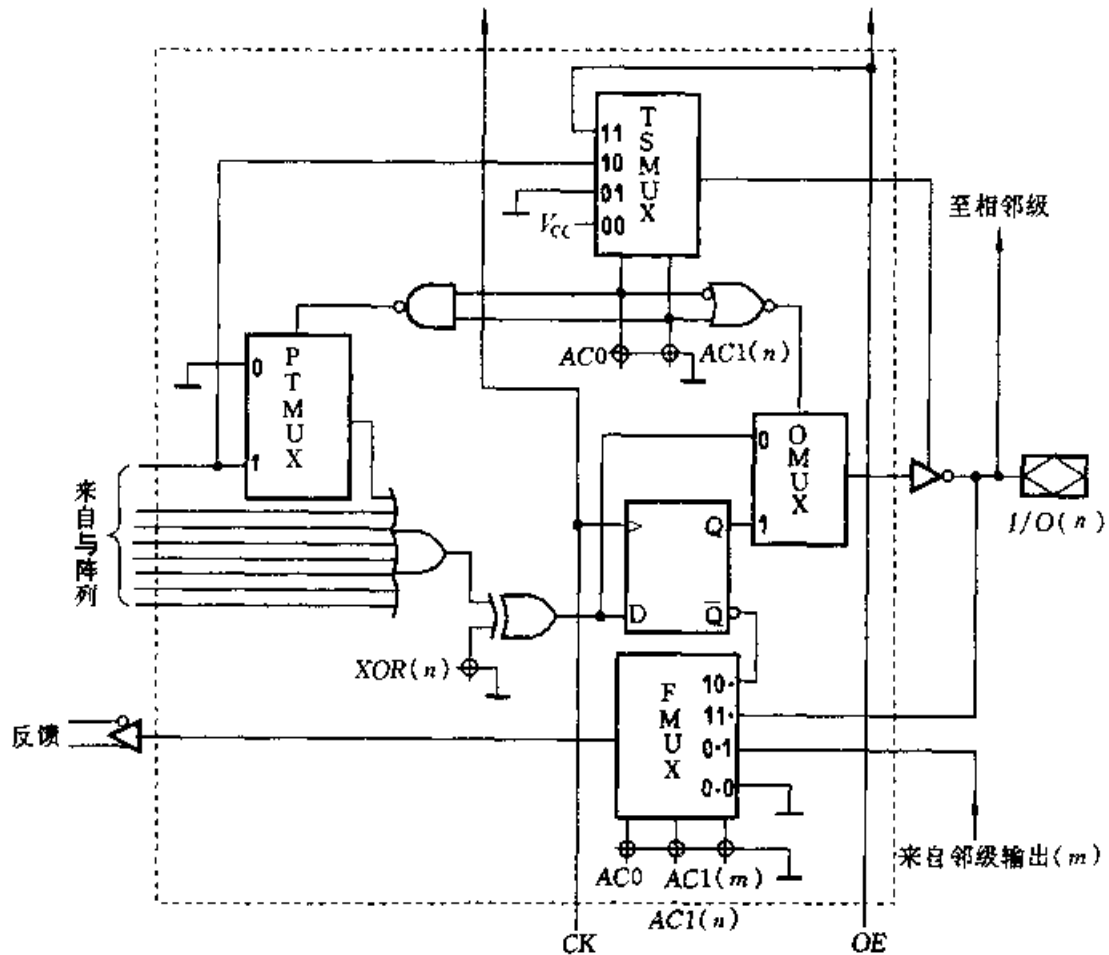


图 8.3.9 输出逻辑宏单元 OLMC

图中: \oplus 表示 E^2CMOS 编程单元

- (1) 或阵列:是一个 8 输入或阵列,构成了 GAL 的或门阵列;
- (2) 异或门:异或门用于控制输出信号的极性,8 输入或门的输出与结构控制字(见本小节第 3 点)中的控制位 $XOR(n)$ 异或后,输出到 D 触发器的 D 端。

通过将 $XOR(n)$ 编程为1或0来改变或门输出的极性; $XOR(n)$ 中的 n 表示该宏单元对应的 I/O 引脚号。

(3) 正边沿触发的 D 触发器; 锁存或门的输出状态, 使 GAL 适用于时序逻辑电路;

(4) 4 个数据多路开关(数据选择器 MUX);

① 乘积项数据选择器 PTMUX; 用于控制来自与阵列的第一乘积项。除了 OLMCI2 和 OLMCI9 两个输出逻辑宏单元外, PTMUX 的控制信号是结构控制字中控制位 $AC0$ 和 $AC1(n)$ 的与非。当 $\overline{AC0} \cdot \overline{AC1(n)} = 1$ 时, 第一乘积项作为或门的一个输入项。

② 三态数据选择器 TSMUX; 用于选择输出三态缓冲器的选通信号。其 4 个数据输入端受 $AC0$ 和 $AC1(n)$ 的控制, TSMUX 的工作情况列于表 8.3.2。

表 8.3.2 $AC0$ 和 $AC1(n)$ 对输出三态缓冲器的控制情况

控制信号		输出三态缓冲器的三态控制信号
$AC0$	$AC1(n)$	C
0	0	1(V_{CC})
0	1	0
1	0	第一乘积项
1	1	OE

表 8.3.3 OLMC 的功能组合

功能	SYN	$AC0$	$AC1(n)$	$XOR(n)$	输出极性	备注
专用输入	1	0	1	—	—	1 和 11 脚为数据输入, 三态门不通
专用组合型输出	1	0	0	0	低电平有效	1 和 11 脚为数据输入, 所有输出是组合的, 三态门总是选通
				1	高电平有效	
反馈组合型输出	1	1	1	0	低电平有效	1 脚和 11 脚为数据输入, 所有输出是组合的, 但三态门由第一乘积项选通
				1	高电平有效	
时序电路中的组合型输出	0	1	1	0	低电平有效	1 脚接 CK , 11 脚接 \overline{OE} ; 这个宏单元输出是组合的, 但其余宏单元至少有一个输出是寄存的
				1	高电平有效	
寄存器型输出	0	1	0	0	低电平有效	1 脚接 CK , 11 脚接 \overline{OE}
				1	高电平有效	

③ 反馈数据选择器 FMUX: 用于决定反馈信号的来源。它根据控制信号 $AC0$ 、 $AC1(n)$ 和 $AC1(m)$ 的值, 分别选择 4 路不同的信号反馈到与阵列的输入端。4 路不同的信号是: 地电平、相邻 OLMC 的输出、本级 OLMC 输出和本级 D 触发器的输出 Q 。 $AC1(m)$ 中的 m 表示邻级宏单元对应的 I/O 引脚号。

④ 输出数据选择器 OMUX: 用于控制输出信号是否锁存。当控制信号 $AC0$ 和 $AC1(n)$ 满足 $\overline{AC0} + AC1(n) = 0$ 时, 异或门的输出信号直接送到输出缓冲器; 但当 $\overline{AC0} + AC1(n) = 1$ 时, 已被 D 触发器锁存的异或门输出信号 (即 D 触发器的 Q 端信号) 送到输出缓冲器, 输出信号是被 D 触发器锁存了的信号。

表 8.3.3 给出了 5 种 OLMC 配置情况, 可以看出, 在结构控制字同步位 SYN 、控制位 $AC0$ 和 $AC1(n)$ 的控制下, 可将 OLMC 设置成 5 种不同的功能组合。

3. 结构控制字

GAL16V8 的各种配置是由结构控制字来控制的。结构控制字如图 8.3.10 所示。图中 $XOR(n)$ 和 $AC1(n)$ 字段下面的数字分别表示它们控制该器件中各个 OLMC 的输出引脚号。

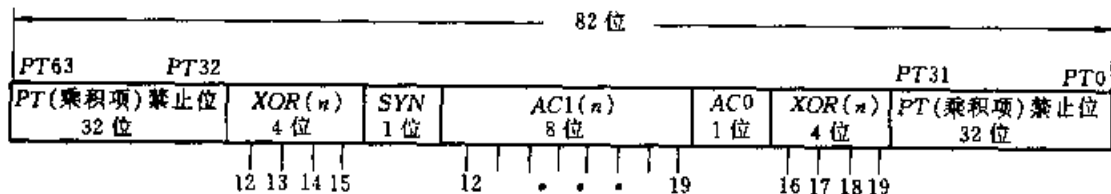


图 8.3.10 GAL16V8 的结构控制字

结构控制字各位功能如下:

(1) 同步位 SYN

该位用以确定 GAL 器件具有组合型输出能力还是寄存器型输出能力。当 $SYN = 1$ 时, 具有组合型输出能力; 当 $SYN = 0$ 时, 具有寄存器型输出能力。此外, 对于 GAL16V8 中的 OLMC(12) 和 OLMC(19), \overline{SYN} 代替 $AC0$, SYN 代替 $AC1(m)$ 作为 FMUX 的控制信号。

(2) 结构控制位 $AC0$

这 1 位对于 8 个 OLMC 是公共的, 它与各个 OLMC(n) 各自的 $AC1(n)$ 配合, 控制 OLMC(n) 中的各个多路开关。

(3) 结构控制位 $AC1$

共有 8 位。每个 OLMC(n) 有单独的 $AC1(n)$ 。

(4) 极性控制位 $XOR(n)$

通过 OLMC 中间的异或门, 控制逻辑操作结果的输出极性:

$XOR(n) = 0$ 时, 输出信号 $O(n)$ 低电平有效;

$XOR(n)=1$ 时,输出信号 $O(n)$ 高电平有效。

(5) 乘积项(PT)禁止位

共有 64 位,分别控制逻辑图中与门阵列的 64 个乘积项($PT0\sim PT63$),以便屏蔽某些不用的乘积项。

通过对结构控制字的编程,便可控制 GAL 的工作方式。

4. GAL 的工作模式

由于 OLMC 提供了灵活的输出功能,因此编程后的 GAL 器件可以替代所有其他固定输出级的 PLD。GAL16V8 有 3 种工作模式,即简单型、复杂型和寄存器型。适当连接该器件的引脚线,由 OLMC 的输出/输入特性可以决定其工作模式。

表 8.3.4 给出了 GAL16V8 的简单型工作模式。处于这种模式时,该器件有多条输入和输出线,没有任何反馈通路。15 和 16 脚仅仅作为输出端,12~14 和 17~19 脚既能作为输入端也能作为输出端,其输出逻辑表达式最多有 8 个乘积项。

表 8.3.4 GAL16V8 的简单型工作模式

引脚号	功能
20	V_{CC}
10	地
1~9,11	仅作为输入
15,16	仅作为输出(无反馈通路)
12~14,17~19	输入或输出(无反馈通路)

表 8.3.5 GAL16V8 的复杂型工作模式

引脚号	功能
20	V_{CC}
10	地
1~9,11	仅作为输入
12,19	仅作为输出(无反馈通路)
13~18	输入或输出(有反馈通路)

表 8.3.5 给出了 GAL16V8 的复杂型工作模式。处于该模式时,它有多条输入和输出线,输出 12 和 19 脚不存在任何反馈通路,输出 13~18 脚和与门阵列之间有一条反馈通路。其输出逻辑表达式最多有 7 个乘积项,另一个乘积项用于输出使能控制。

表 8.3.6 给出了 GAL16V8 的寄存器型工作模式。

表 8.3.6 GAL16V8 的寄存器型工作模式

引脚号	功能	引脚号	功能
20	V_{CC}	1	时钟脉冲输入
10	地	11	使能输入(低电平有效)
2~9	仅作为输入	12~19	输入或输出(有反馈通路)

例 8.3.1 试用 GAL16V8 设计一个四 2 选 1 多路数据选择器,其功能表如

表 8.3.7 所示。

表 8.3.7 多路数据选择器功能表

选择控制 S	输出 Y ₃ Y ₂ Y ₁ Y ₀	选择控制 S	输出 Y ₃ Y ₂ Y ₁ Y ₀
0	A ₃ A ₂ A ₁ A ₀	1	B ₃ B ₂ B ₁ B ₀

解：根据四 2 选 1 多路数据选择器的功能可知：电路需要 1 个选择输入端，8 个数据输入端和 4 个数据输出端。下面在可编程逻辑器件开发软件 ISP Synario 中，用 ABEL 语言设计该选择器。

四个 2 选 1 的数据选择器 ABEL 源文件：

```

module multiple                                "模块开始
title 'Quad Two - Input Multiplexer'
declarations
    S,A3,B3,A2,B2,A1,B1,A0,B0  PIN  19,2,3,4,5,6,7,8,9;
                                "定义输入
    Y3,Y2,Y1,Y0  PIN  15,14,13,12  istype 'com'; "定义输出
    X = .X.;      "定义常数
equations
    Y3 = S & B3 # !S & A3;      "描述逻辑
    Y2 = S & B2 # !S & A2;
    Y1 = S & B1 # !S & A1;
    Y0 = S & B0 # !S & A0;
test _ vectors
    ([S,A0,B0] -> Y0) "测试向量
    [0,0,0] -> X;
    [0,0,1] -> X;
    [0,1,0] -> X;
    [0,1,1] -> X;
    [1,0,0] -> X;
    [1,0,1] -> X;
    [1,1,0] -> X;
    [1,1,1] -> X;
test _ vectors
    ([S,A1,B1] -> Y1)

```

```

[0,0,0] -> X;
[0,0,1] -> X;
[0,1,0] -> X;
[0,1,1] -> X;
[1,0,0] -> X;
[1,0,1] -> X;
[1,1,0] -> X;
[1,1,1] -> X;

```

end

GAL16V8 实现的四 2 选 1 数据选择器的管脚分配如图 8.3.11 所示,功能测试的部分仿真波形如图 8.3.12 所示。可以看出设计满足要求。然后在 ISP Synario 中生成 JEDEC 文件,最后用编程器对 GAL16V8 进行编程。编程后的 GAL16V8 便具有了四 2 选 1 数据选择器的功能。

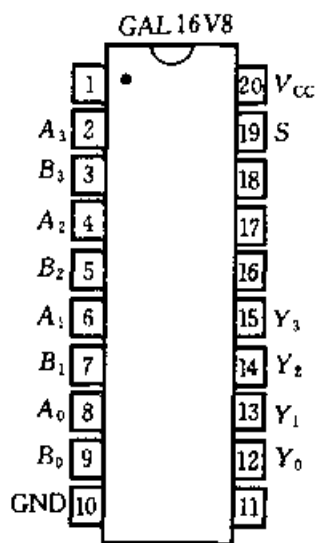


图 8.3.11 选择器的管脚分配

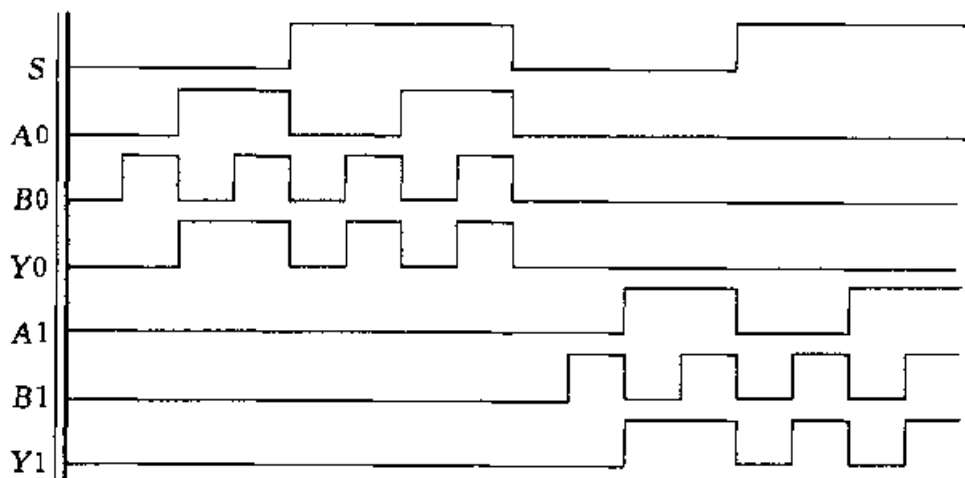


图 8.3.12 四 2 选 1 数据选择器功能测试部分仿真波形

复习思考题

- 8.3.1 PAL 和 GAL 实现组合逻辑函数的基本原理是什么?
- 8.3.2 GAL 中的输出三态缓冲器可由哪几个信号控制?
- 8.3.3 在 GAL 的 OLMC 中可以将哪几个信号反馈到与阵列中?

① 为了减少篇幅,在 ABEL 源文件只给出了 Y_0 和 Y_1 的测试向量,因此仿真波形也只有其中两个选择器的波形

8.3.4 在现代数字系统中, PAL 和 GAL 两种器件应用的局限性是什么?

8.4 复杂的可编程逻辑器件(CPLD)

8.4.1 CPLD 的结构

CPLD^① 将简单 PLD(PAL、GAL 等)的概念作了进一步的扩展,并提高了器件的集成度。和简单的 PLD 相比,CPLD 允许有更多的输入信号、更多的乘积项和更多的宏单元,CPLD 器件内部含有多个逻辑单元块,每个逻辑块就相当于一个 GAL 器件,这些逻辑块之间可以使用可编程内部连线实现相互连接。目前,生产 CPLD 器件著名的公司有多家^②,尽管各个公司的器件结构千差万别,但它们仍有共同之处,图 8.4.1 给出了通用的 CPLD 器件的结构框图。

下面以 LATTICE 公司生产的在系统可编程大规模集成逻辑器件 ispLSI 1016 为例,介绍 CPLD 的电路结构及其工作原理。这种器件的最大特点是“在系统可编程(ISP^③)”特性。所谓在系统可编程是指未编程的 ISP 器件可以直接焊接在印制电路板上,然后通过计算机的并行口和专用的编程电缆对焊接在电路板上的 ISP 器件直接多次编程,从而使器件具有所需要的逻辑功能。这种编程不需要使用专用的编程器,因为已将原来属于编程器的编程电路及升压电路集成在 ISP 器件内部了。ISP 技术使得调试过程不需要反复拔插芯片,从而不会产生引脚弯曲变形现象,提高了可靠性,而且可以随时对焊接在电路板上的 ISP 器件的逻辑功能进行修改,从而加快了数字系统的调试过程。除了 LATTICE 公司外,其他公司生产的 CPLD 器件也具有在系统编程的功能。

ispLSI 1016 的结构如图 8.4.2 所示。它由 16 个相同的通用逻辑块 GLB^④ (A0~A7、B0~B7)、32 个相同的输入、输出单元(I/O0~I/O31)、可编程的集总布线区 GRP^⑤、时钟分配网络以及在系统编程控制电路等部分组成(图中未画出编程控制电路)。在 GRP 的左边和右边各形成一个宏模块。每个宏模块包括:8 个 GLB、16 个 I/O 单元、两个专用输入引脚(SDI/IN0, SDO/IN1 或 MODE/IN2, IN3)、一个输出布线区 ORP 以及 16 位的输入总线。

① Complex Programmable Logic Device 之缩写。

② 如 ALTERA、AMD/VANTIS、LATTICE、CYPRESS 和 XILINX 公司等。

③ In-System Programmability 之缩写。

④ Generic Logic Block 之缩写。

⑤ Global Routing Pool 之缩写。

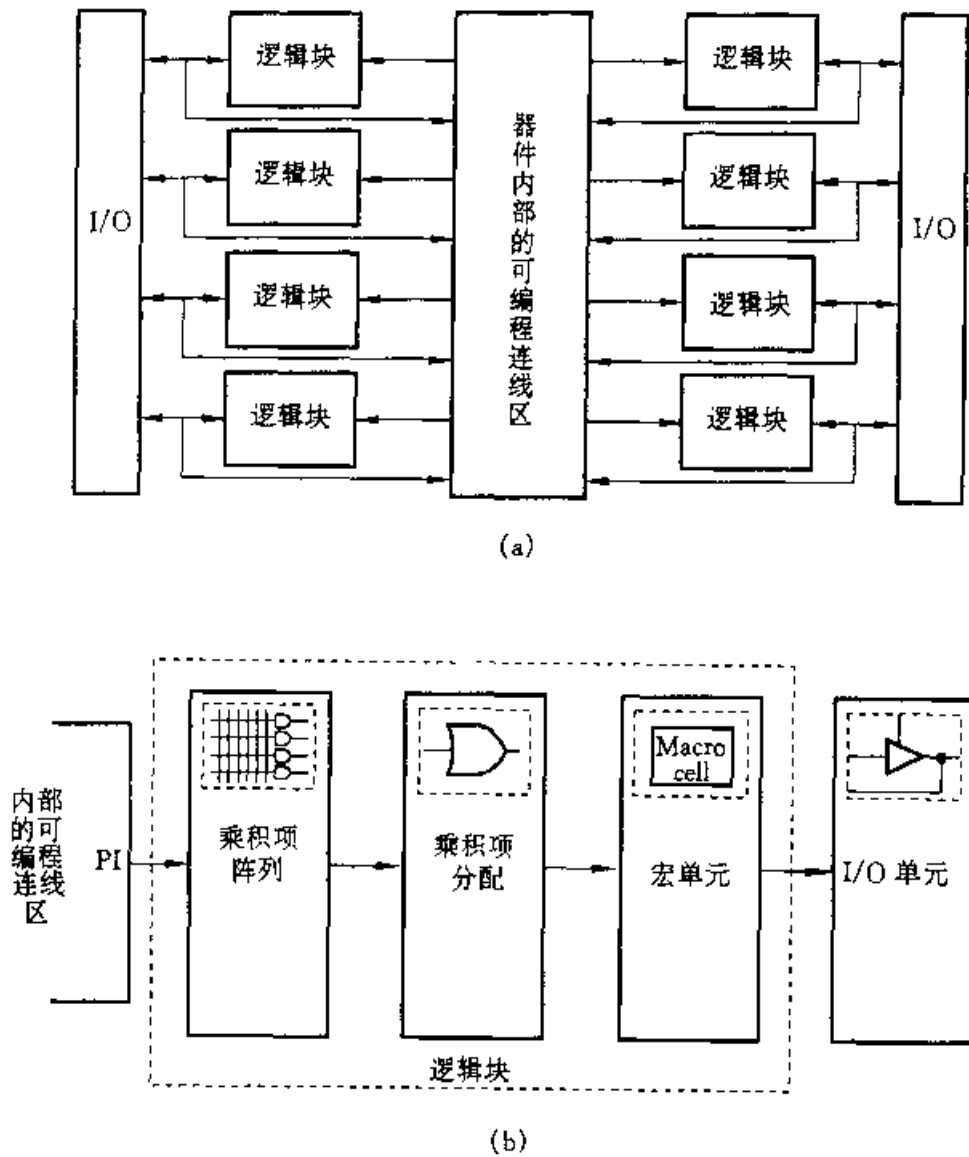


图 8.4.1 通用的 CPLD 器件的结构框图

(a) CPLD 结构 (b) 逻辑块结构

集总布线区 GRP 位于两个宏模块的中央,它由众多的可编程 E²CMOS 构成,内部逻辑的连接都是通过这一区域完成的。它接收输入总线送来的输入信号和各 GLB 的输出信号,同时向每个宏模块输出信号。因此,任何一个 GLB 的输出信号和任何一个通过 I/O 单元的输入信号都能送到任何一个 GLB 的输入端。这种结构使得信号的传输延迟时间是可预知的,有利于获得高性能的数字系统。

下面简要介绍通用逻辑块 GLB、输入输出 I/O 单元、输出布线区和时钟分配网络的结构和功能。

1. 通用逻辑块(GLB)的结构

GLB 由与阵列、乘积项共享阵列、输出逻辑宏单元 OLMC 和功能控制 4 部

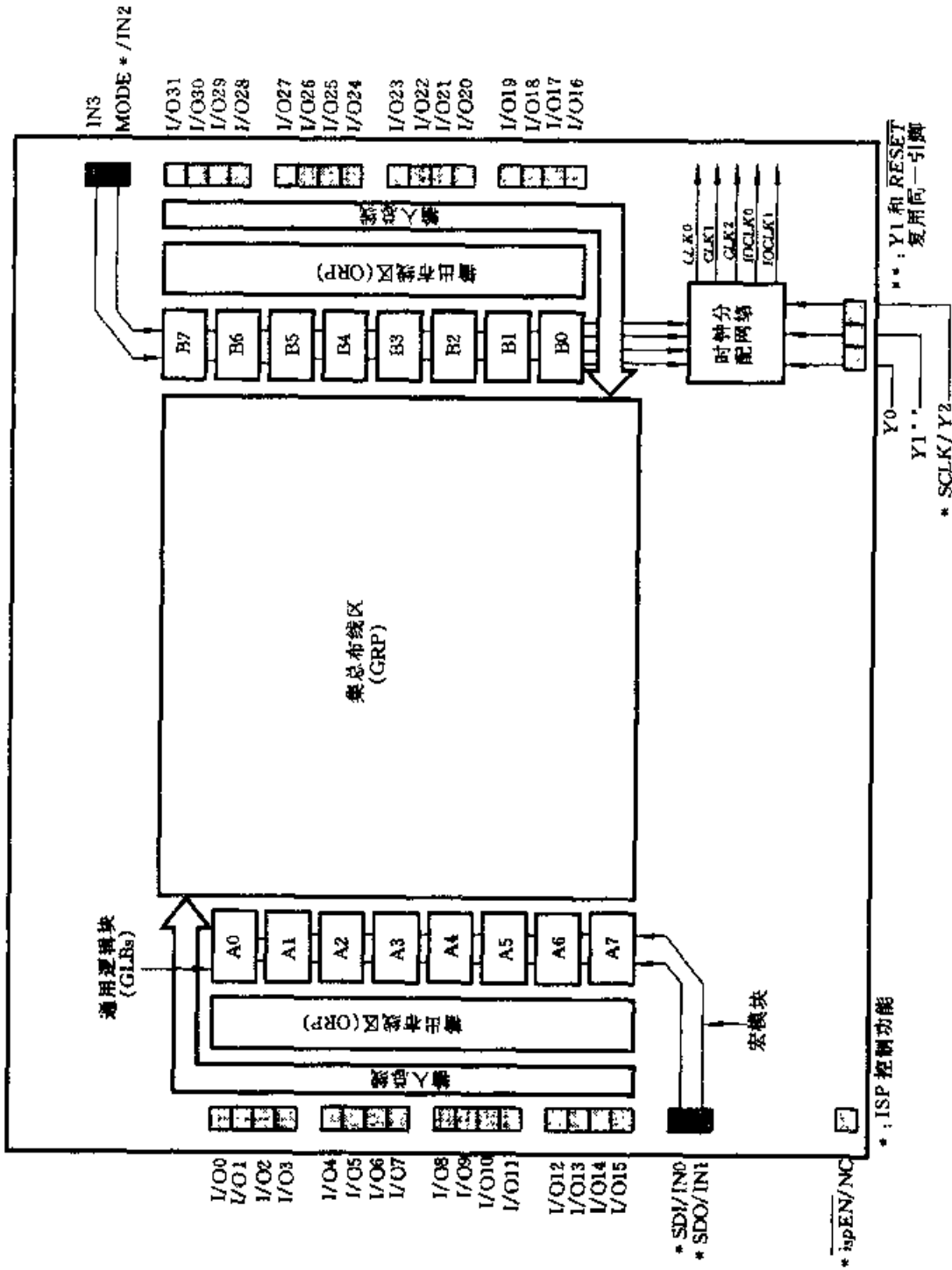


图 8.4.2 ispLSI 1016 的结构框图

分组成。它可实现类似 GAL 的功能。简化的 GLB 逻辑图如图 8.4.3 所示, 4 部分具体功能如下:

(1) 与阵列

与阵列有 18 个输入, 其中有 16 个来自集总布线区 GRP(它们可以是来自 I/O 引脚的信号, 也可以是 GLB 的反馈信号), 另外两个来自专用输入引脚, 它们经过输入缓冲器后, 都产生互补信号。通过对与阵列编程, 可以产生 20 个乘积项(0~19)。

(2) 乘积项共享阵列

这一阵列可以把 20 个乘积项分组送到 4 个或门, 其输出经过乘积项共享阵列的编程, 可以按需要连至 GLB 的任何一个输出。乘积项共享阵列具有“线或”功能, 如果输出函数需要的乘积项多于 7 个, 可以将两个或两个以上的或门输出的乘积项合并, 最多可以实现 20 个乘积项的输出。这种相同的乘积项可以被多个输出宏单元使用的情况, 称为乘积项共享。

乘积项共享阵列可以灵活地配置以满足用户不同的需要, 同一个 GLB 中的 4 个输出可以采用相同的配置形式, 也可以采用不同的配置形式(混合配置), 图 8.4.4 采用的是混合配置。图中, O3 配置为异或模式, 第 1 个或门输出的 3 个乘积项与第 3 个或门输出的 4 个乘积项进行“线或”组成 7 个乘积项, 然后再与第 0 个乘积项“异或”, “异或”的结果送到属于 O3 的 D 触发器输入端。O2 配置为高速旁路模式, 第 2 个或门的输出直接送到 OLMC。O1 配置为单乘积项旁路模式, 可以获得最快的信号传输速度。O0 配置为共享下面两个或门的 11 个乘积项。

(3) 输出逻辑宏单元 OLMC

与 GAL 中的 OLMC 类似, 它由 4 个 D 触发器构成, 其输入端接异或门(图 8.4.3 中未画出)。异或门可以作为逻辑单元来使用, 也可以把它与 D 触发器结合构成 JK 触发器或 T 触发器。如果需要组合逻辑输出, 可以通过数据选择器把触发器旁路掉。

(4) 功能控制

寄存器的时钟信号分为同步时钟和异步时钟信号两种。同步时钟信号由时钟分配网络供给, 它可以在 CLK0、CLK1 及 CLK2 中选择一个; 异步时钟信号由 GLB 中的第 12 乘积项提供。寄存器的复位信号由全局复位引脚或 GLB 中的第 12 或 19 乘积项提供。另外, 第 19 乘积项还可以作为输出三态门的输出使能控制信号。因此, 若在设计中使用第 12 或第 19 乘积项作为控制信号, 那么这一乘积项就不能用于实现其他逻辑功能。乘积项时钟是通过输入项相“与”产生的时钟, 也是 ispLSI 器件最有特色的性质之一。

2. I/O 单元结构

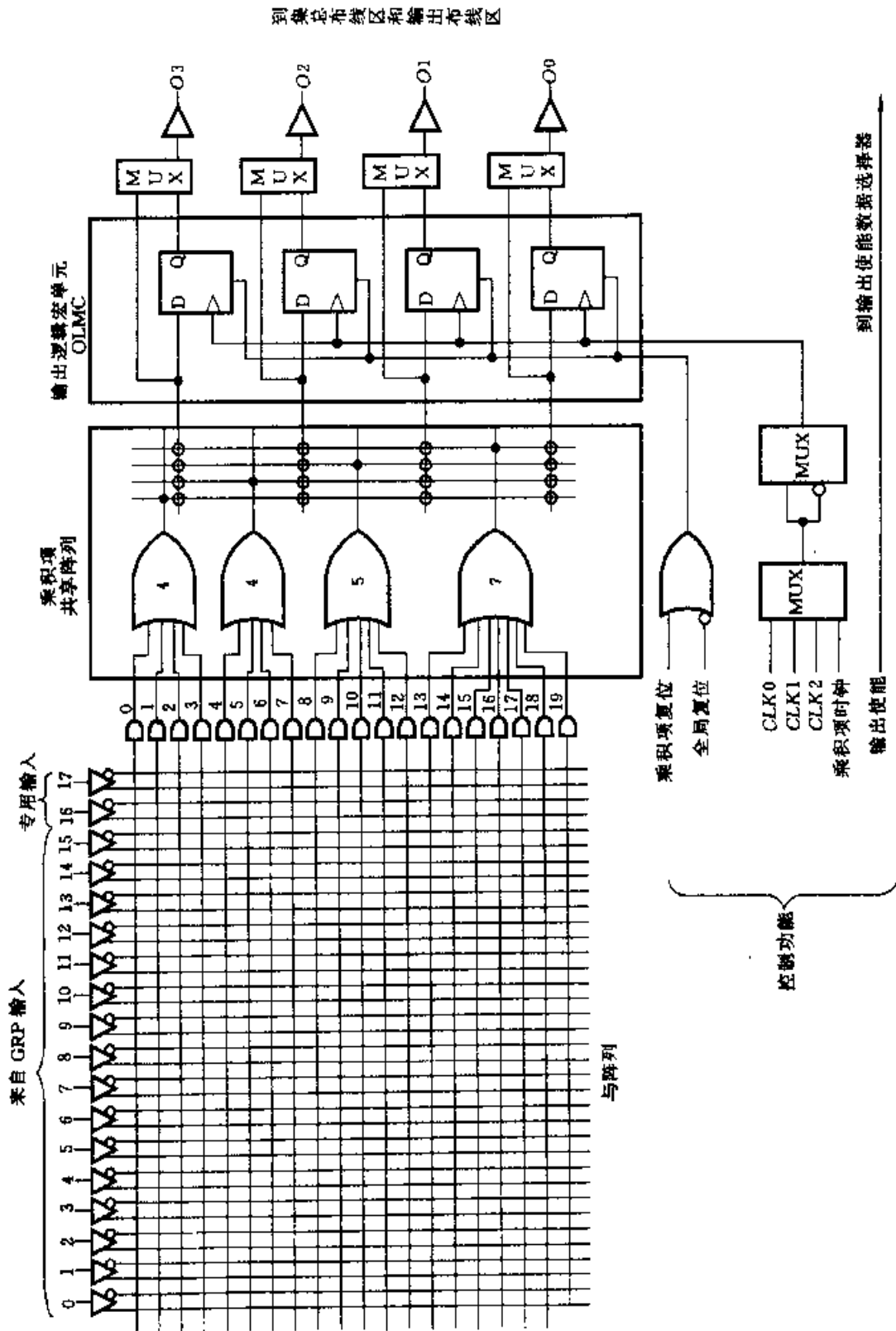


图 8.4.3 ispLSI 1016 器件通用逻辑块(GLB)的结构

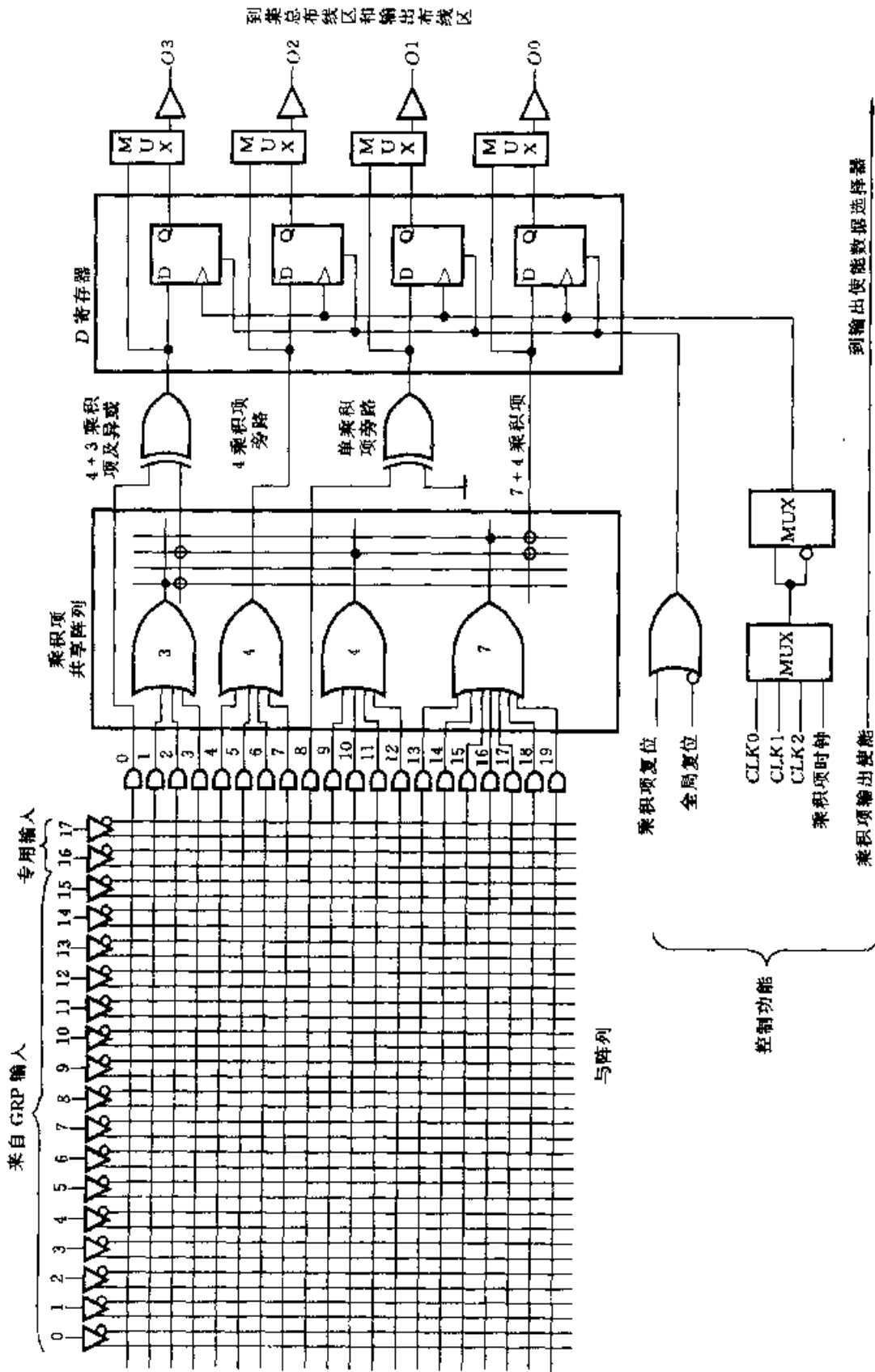


图 8.4.4 ispLSI 1016 器件通用逻辑块(GLB)的配置举例

I/O 单元是 CPLD 外部封装引脚和内部逻辑间的接口。每个 I/O 单元对应一个封装引脚,通过对 I/O 单元中可编程单元的编程,可将引脚定义为输入、输出和双向功能。ispLSI 1016 I/O 单元的简化原理框图如图 8.4.5 所示。

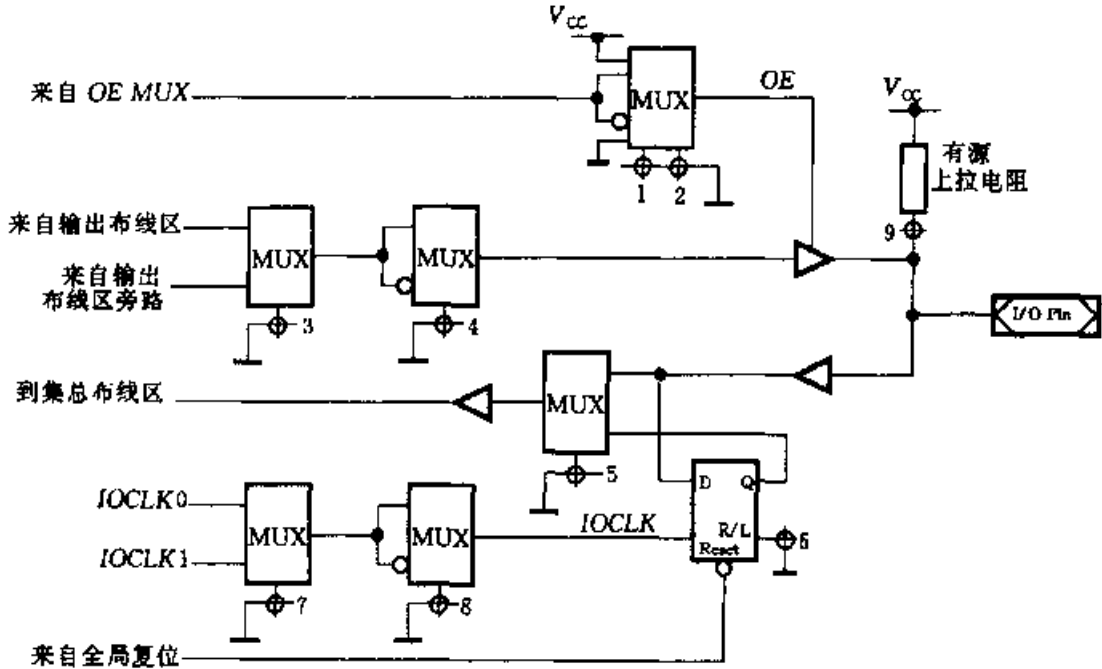


图 8.4.5 ispLSI 1016 器件 I/O 单元的结构图

I/O 单元中有输入和输出两条信号通路。当 I/O 引脚作输出时,三态输出缓冲器的输入信号来自输出布线区,由可编程单元 3、4 控制数据选择器 MUX,选择输入信号的来源及极性。三态输出缓冲器的使能控制信号 OE,由可编程单元 1、2 控制数据选择器 MUX 选择其来源;当 OE 为低电平时,I/O 引脚可用作输入,引脚上的输入信号经过输入缓冲器,由可编程单元 5 控制数据选择器

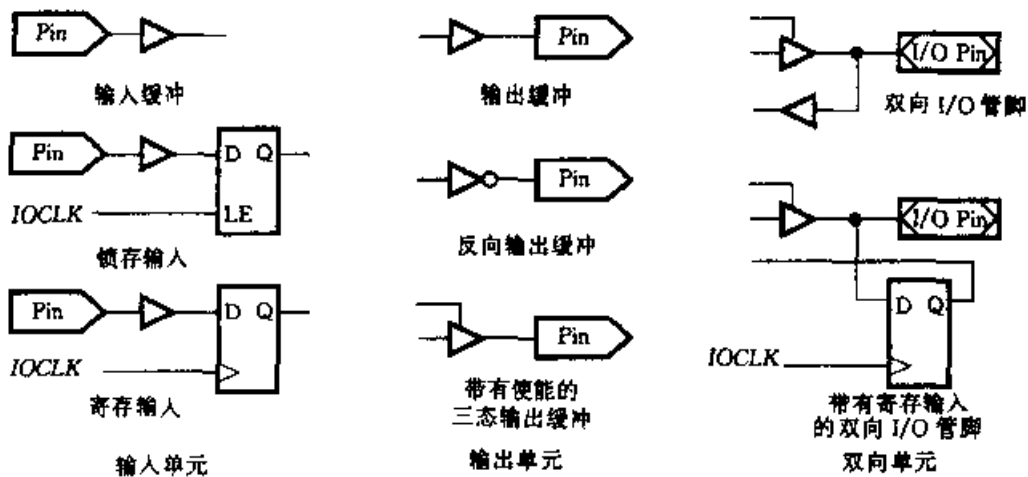


图 8.4.6 ispLSI 1016 器件 I/O 单元的配置形式

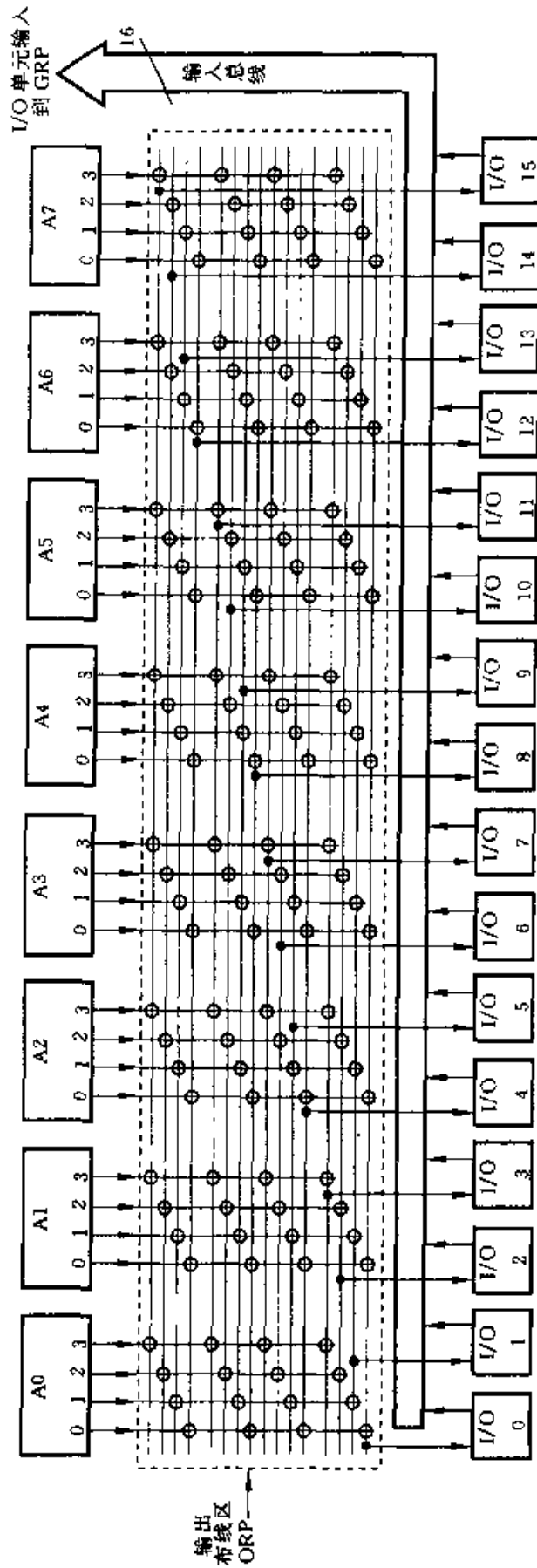


图 8.4.7 ispLSI 1016 器件输出布线区(ORP)的结构

MUX 选择是直接送到集总布线区,还是经 D 触发器寄存后输入到集总布线区。由可编程单元 6 控制 D 寄存器工作在寄存方式或锁存方式。可编程单元 7、8 控制 D 触发器的时钟信号 $IOCLK$ 的来源及极性。通过对上述 8 个可编程单元的编程,可以使 I/O 单元配置为如图 8.4.6 所示的 8 种形式。

每一个 I/O 单元都接有上拉电阻,如果某一个 I/O 引脚未使用,通过可编程单元 9 可以使上拉电阻接至该引脚,防止该引脚浮空,避免了噪声进入该电路及消耗额外的功率。

3. 输出布线区

输出布线区(ORP)的结构如图 8.4.7 所示,它的作用是把 GLB 的输出信号接到 I/O 单元。8 个通用逻辑块及 16 个 I/O 单元共用一个输出布线区,每个 GLB 的输出可以分别接到 4 个 I/O 单元。例如,通过对输出布线区的编程,各个 GLB 的输出 $O3$ 都可以接到 I/O3、I/O7、I/O11 及 I/O15 中的任一个。而对 GLB 中乘积项共享阵列的编程,可以使 GLB 中的 4 个输出位置互换。因此,实际上能够做到把每个 GLB 的输出送到本宏模块内任意 1 个 I/O 单元。这些工作是由开发软件的布线程序自动完成的。

4. 时钟分配网络

ispLSI 1016 器件的时钟分配网络如图 8.4.8 所示。它有 3 个外部时钟引脚,其中 $Y0$ 脚直接连至 $CLK0$, $Y1$ 连至全局复位及时钟分配网络, $Y2$ 也连至时钟分配网络。在每个器件的内部都有一个确定的 GLB 与时钟分配网络相连,这个 GLB 既可以作为普通的 GLB 使用(此时不与时钟分配网络相连),又可以

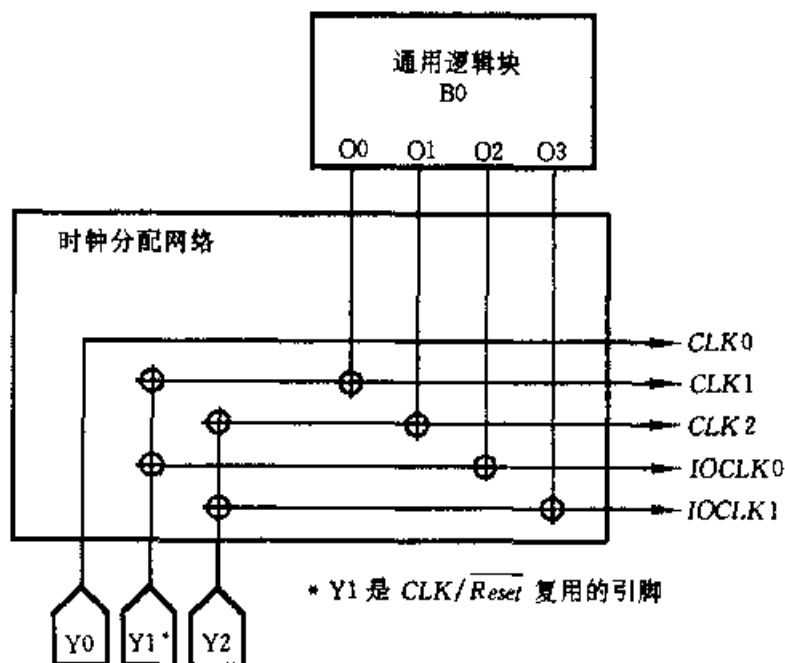


图 8.4.8 ispLSI 1016 器件的时钟分配网络

用来产生时钟。在 ispLSI 1016 内部, GLB B0 的 4 个输出 O0~O3 与时钟分配网络相连, 产生 CLK1、CLK2、IOCLK0 及 IOCLK1 时钟。在这种情况下, 这 4 个时钟是用户定义的内部时钟。其中 IOCLK0 及 IOCLK1 用作 I/O 单元的时钟。

8.4.2 CPLD 的编程

通过 8.4.1 节的介绍可以看出, CPLD 的各种逻辑功能的实现, 都是由其内部的可编程单元控制的。这些单元均为 E²CMOS 结构, 它们按照一定的规则排列成阵列形式。编程过程就是将编程数据写入 E²CMOS 单元阵列的过程。下面以 ispLSI 器件为例进行说明。

1. ispLSI 器件 E²CMOS 单元的编程方法

每个 ispLSI 器件有一个预先规定的 E²CMOS 单元阵列。此阵列的行数是 n , 每行的数据位数是 m 。两者的乘积 $m \times n$ 就是要编程的总位数。

表 8.4.1 列出了 Lattice 公司生产的若干 ispLSI 器件 E²CMOS 单元阵列的有关数据。

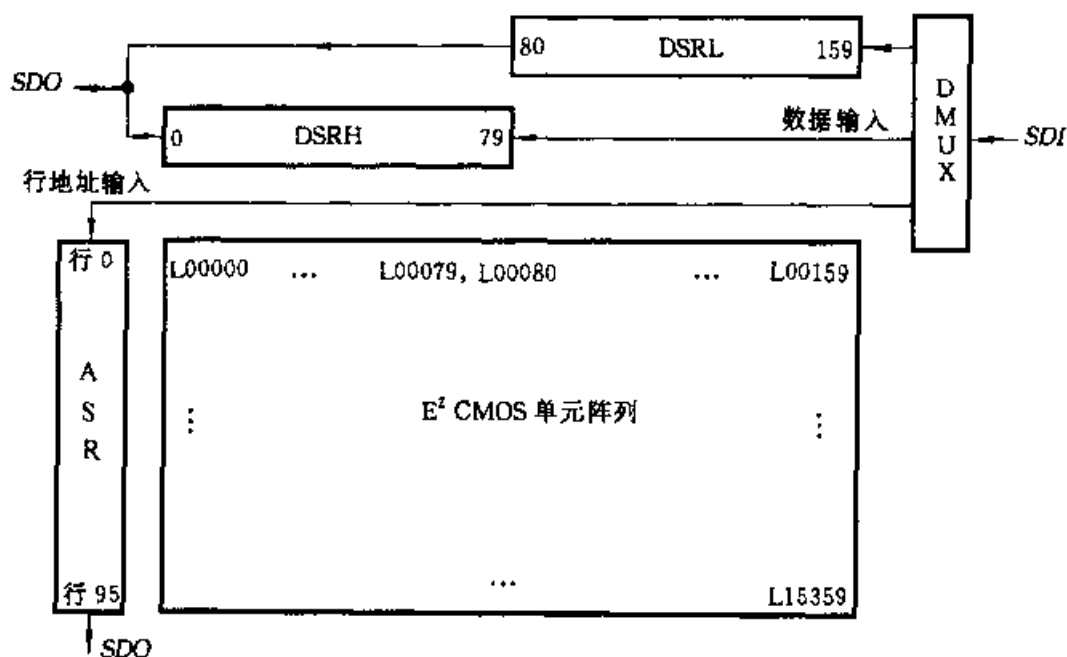
表 8.4.1 ispLSI 器件 E²CMOS 单元阵列的数据

	ispLSI 1016	ispLSI 1032	ispLSI 2032	ispLSI 3256
行数 n	96	108	102	180
每行数据位数 m	160	320	80	676
编程总位数	15360	34560	8160	121680

ispLSI 1016 器件的 E²CMOS 单元阵列如图 8.4.9 所示。图中还画出了编程所需的 96 位地址移位寄存器 ASR^①, 两个数据移位寄存器 DSRH(高位数据移位寄存器)和 DSRL(低位数据移位寄存器), 两者的长度都是 80 位。ASR、DSRH 及 DSRL 都集成在 ispLSI 1016 器件内部。

编程时, 行地址信号首先从串行数据输入端 SDI 输入, 经过数据分配器 DMUX 送至地址移位寄存器 ASR。行地址信号中只有 1 位为 1, 其余全为 0。最初地址中的 1 处于 ASR 的行 0 位置, 此时只能对第 0 行的 E²CMOS 单元编程。行地址送完后, 接着送第 0 行编程数据, 160 位编程数据分两次送入。先将 80 位编程数据 L00000~L00079 经 DMUX 送至 DSRH, 对行 0 的左边 80 位 E²CMOS 单元进行编程; 再将另 80 位数据 L00080~L00159 经 DMUX 送至

① Address Shift Register 之缩写。

图 8.4.9 ispLSI 1016 器件的 E²CMOS 单元编程结构示意图

DSRL,对行 0 的右边 80 位 E²CMOS 单元进行编程。第 0 行编程完毕,地址移位寄存器 ASR 下移一位,ASR 中唯一的 1 移至行 1 位置,接着将第 1 行的编程数据 L00160~L00319 移入 DSRH 和 DSRL 对行 1 的各 E²CMOS 单元编程。如此下去,直至对器件上行 95 的各 E²CMOS 单元编程完毕。其中,L00000 表示 E²CMOS 阵列中开始的可编程单元编号是 00000。

2. ispLSI 1000 及 2000 系列器件的编程接口

ISP 器件的编程必须具备三个条件:①ISP 专用编程电缆;②PC 机;③ISP 编程软件。编程时,用户首先将 ISP 编程电缆的一端接到 PC 机的并行口,另一端接到电路板上被编程器件的 ISP 接口上,然后通过编程软件发出编程命令,将编程数据文件(*.JED)中的数据转换成串行数据传送到芯片中。在系统编程时,ispLSI 1000 及 2000 系列器件所使用的接口电路如图 8.4.10 所示。其中 \overline{ispEN} 是编程使能信号,MODE 是模式控制信号,SCLK 是串行时钟输入信号,SDI 是串行数据和命令输入端,SDO 是串行数据输出端。

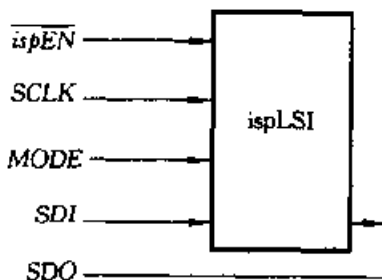


图 8.4.10 ispLSI 器件的编程接口

当 $\overline{ispEN} = 1$ 时,MODE、SCLK、SDI 端均为高阻状态,在印制电路板上的 ispLSI 器件正常工作,称之为正常工作模式。在正常工作模式下,MODE、SCLK、SDI 变成专用输入引脚,有正常输入信号的功能。当 $\overline{ispEN} = 0$ 时,各种

控制信号和编程数据经过编程电缆直接送到印制板上 ispLSI 器件的 *MODE*、*SCLK*、*SDI* 端,从而控制器件内部的一个“编程状态机”完成编程工作。编程数据可以从 *SDO* 端移出并回送到计算机以便进行校验。这种工作方式称之为编程模式。在编程模式下,所有 *I/O* 引脚以及编程时不用的输入引脚均处于高阻态。

另外,除了对单个 ISP 器件能够进行在系统编程外,还可以将印制电路板上多个 ISP 器件以串行的方式连接起来,一次完成多个器件的编程。这种连接方式称之为菊花链连接,其电路连接举例如图 8.4.11 所示。

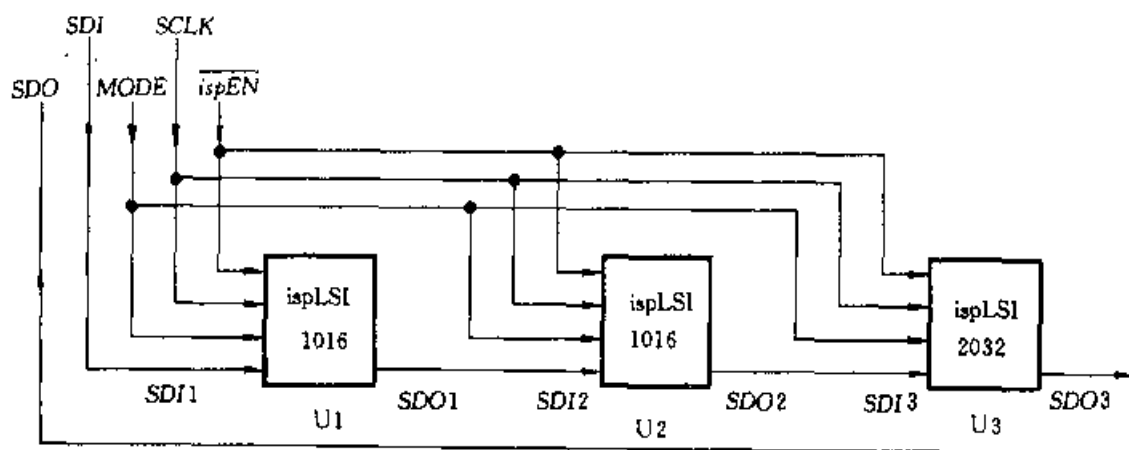


图 8.4.11 多个 ispLSI 器件的菊花链连接举例

3. ispLSI 3000 及 3000 以上系列器件的编程接口

自 3000 系列开始,LATTICE 公司生产的在系统可编程器件均增加了边界扫描^①测试功能,为此 ISP 器件专门设计了测试端口。而器件的编程端口设计成与测试端口复用的形式。端口定义与用途如表 8.4.2 所示。当 *BSCAN*/

表 8.4.2 边界扫描测试端口与编程端口复用关系

端口引脚	用于边界扫描时的功能	用于编程时的功能
<i>BSCAN</i> / \overline{ispEN}	边界扫描使能,高电平有效	编程使能,低电平有效
<i>TMS</i> / <i>MODE</i>	测试模式选择	编程模式选择
<i>TCLK</i> / <i>SCLK</i>	边界扫描时钟	编程时钟
<i>TDI</i> / <i>SDI</i>	测试数据输入	编程数据输入
<i>TDO</i> / <i>SDO</i>	测试数据输出	编程数据输出
\overline{TRST}	复位信号	—

^① Boundary Scan,一种测试技术,用来解决高密度引线器件和高密度电路板上的元件测试问题。具有国际标准 IEEE1149.1。

$\overline{ispEN} = 0$ 时,器件处于编程状态,引脚 $TMS/MODE$ 、 $TCLK/SCLK$ 、 TDI/SDI 、 TDO/SDO 与 ispLSI 1000 系列的 $MODE$ 、 $SCLK$ 、 SDI 和 SDO 具有相同的作用和功能;当 $BSCAN/\overline{ispEN} = 1$ 时,器件处于边界扫描测试状态。

例 8.4.1 试用 ispLSI 1016 设计一个用来记录短跑运动员成绩的秒表电路。要求:(1)秒表的计时范围为 $0.01\text{ s} \sim 59.99\text{ s}$;(2)具有清零、启动、停止功能;(3)输入时钟脉冲的频率为 100 Hz ,输出为 8421 BCD 码。

解: (1) 逻辑设计

根据设计要求可知,电路需要输出 4 组 8421BCD 码。从秒表的计时过程可以看出,电路实际上可以由 3 个十进制(模 10)计数器和 1 个六进制(模 6)计数器串接构成。各控制信号应满足下述关系:①每次启动前必须清零,且一旦启动后,再来启动信号电路不受影响;②停止信号到来时,秒表停止计时,输出保持停止前的状态,且此时再来停止信号或启动信号,电路状态不变。

根据上述分析,可以设计出电路原理图,如图 8.4.12 所示。图中 CNT10 表

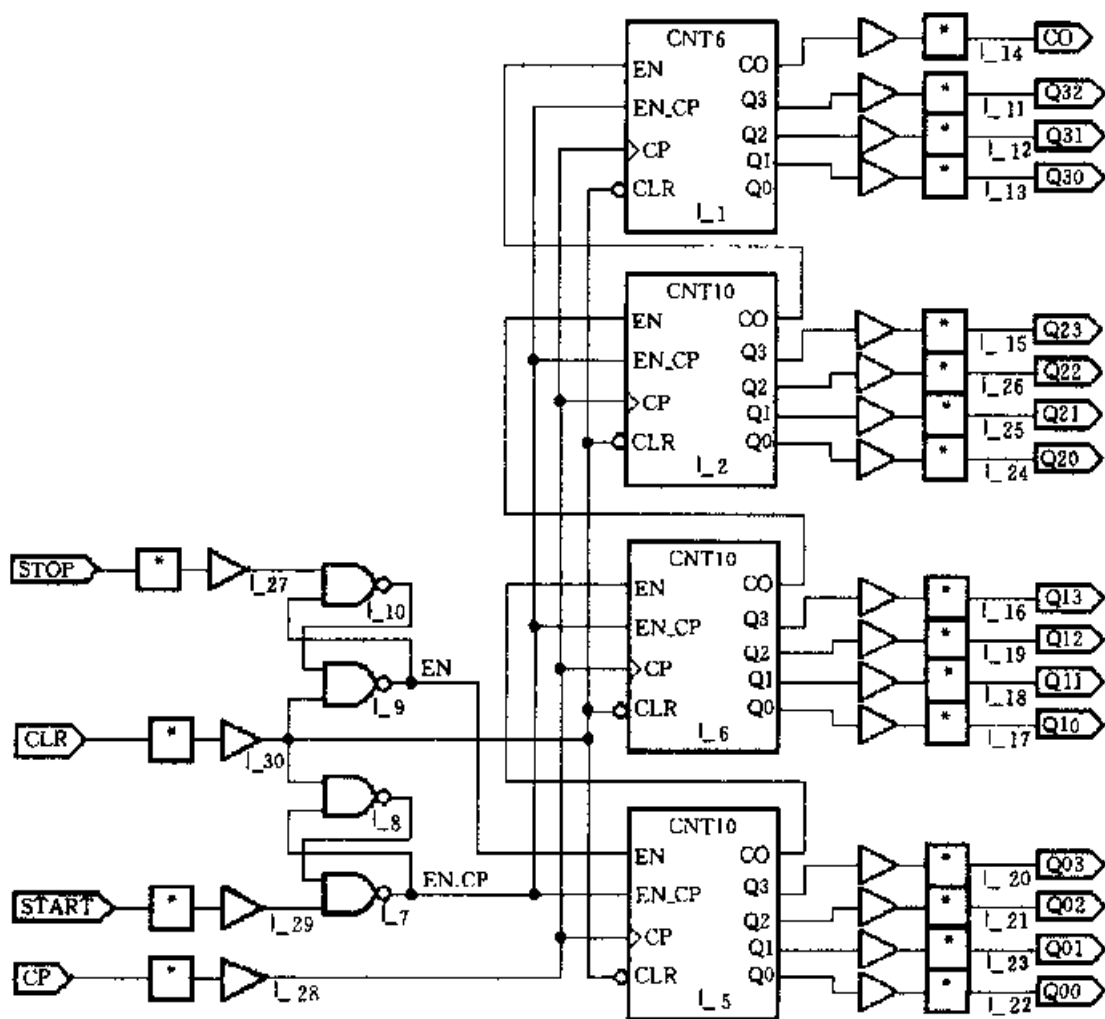


图 8.4.12 秒表电路原理图

注:带 * 号的小方框和三角形为 ISP Synario 软件中的输入、输出端口符号。

示8421BCD码十进制计数器模块,CNT6表示六进制计数器模块,EN为计数使能,EN_CP为时钟使能。为了满足各控制信号的控制关系,图中设计了两个由与非门构成的基本RS触发器。

(2) 设计实现

图8.4.12中CNT10和CNT6只给出了框图,其具体电路采用ABEL语言描述。由此看出,电路是采用层次化设计方法实现的,即顶层采用了电路原理图,底层采用了ABEL语言。CNT10和CNT6模块的ABEL语言源程序如下:

8421BCD码十进制递增计数器模块程序:

```

module CNT10
title '0~9 BCD COUNTER'
declarations
    EN,EN_CP,CP,CLR    pin;
    Q3..Q0              pin istype 'reg';
    CO                  pin istype 'com';
    COUNT=[Q3..Q0];
equations
    COUNT.CLK=CP;
    COUNT.CE=EN_CP;
    COUNT.RE=!CLR;
    when EN & (COUNT<9) then COUNT:=COUNT+1;
    else when(!EN) then COUNT:=COUNT;
    else COUNT:=0;
    CO=EN & Q3 & Q0;
end

```

六进制递增计数器模块程序:

```

module CNT6
title '0~5 BCD COUNTER'
declarations
    EN,EN_CP,CP,CLR    pin;
    Q2..Q0              pin istype 'reg';
    CO                  pin istype 'com';
    COUNT=[Q2..Q0];
equations
    COUNT.CLK=CP;

```

```

COUNT.CE = EN _ CP;
COUNT.RE = ! CLR;
when (EN & (COUNT < 5)) then COUNT := COUNT + 1
else when (!EN) then COUNT := COUNT
else COUNT := 0;
CO = EN & Q2 & Q0;
end

```

编写测试向量对电路进行功能仿真, 仿真波形如图 8.4.13 所示。由此看出, 电路功能满足设计要求。

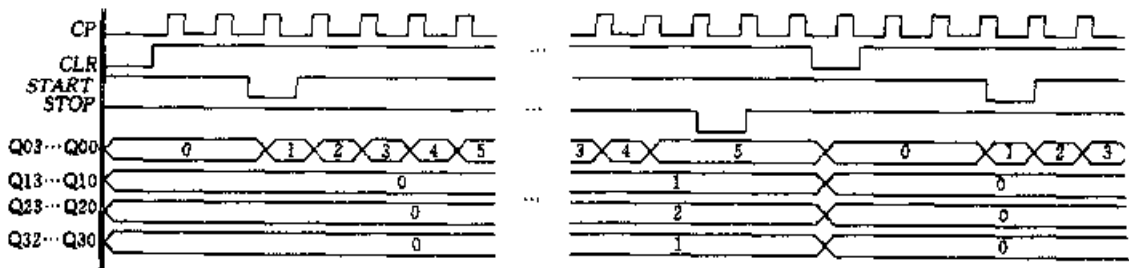


图 8.4.13 秒表电路功能仿真波形

接着利用开发软件对 ispLSI 1016 进行布局 and 布线, 生成 JEDEC 文件。ispLSI 1016 引脚分配也由软件自动完成, 其结果如表 8.4.3 所示。最后, 用下载软件 ispDCD 对 ispLSI 1016 器件进行在系统编程, 编程后 ispLSI 1016 就是一块秒表电路。

表 8.4.3 ispLSI 1016 引脚分配

引脚号	引脚名	类型	引脚号	引脚名	类型	引脚号	引脚名	类型
5	Q30	Output	18	Q10	Output	39	Q22	Output
8	START	Input	19	Q03	Output	40	Q21	Output
10	CLR	Input	20	Q02	Output	41	Q32	Output
11	CP	Clock Input	21	Q01	Output	42	Q31	Output
15	Q13	Output	22	Q00	Output	43	STOP	Input
16	Q12	Output	37	Q23	Output	44	CO	Output
17	Q11	Output	38	Q20	Output			

复习思考题

8.4.1 CPLD 在结构上可分为哪几个部分? 各部分的主要功能是什么?

8.4.2 CPLD 的可编程特性是基于什么编程技术? 这种编程技术有什么特点?

8.4.3 通过对 CPLD 的 I/O 单元的编程,可将其配置成哪几种形式?

8.4.4 LATTICE 公司的 ispLSI 器件的编程接口共有几个引脚?它们各有什么作用?

* 8.5 现场可编程门阵列(FPGA)

FPGA^① 是 80 年代中期发展起来的另一种类型的可编程器件。与前面讨论过的可编程器件相比, FPGA 不受“与-或”阵列结构上的限制以及含有触发器和 I/O 端数量上的限制, 可以靠内部的逻辑单元以及它们的连接构成任何复杂的逻辑电路, 更适合实现多级逻辑功能, 并且具有更高的密度和更大的灵活性。目前已成为设计数字电路或系统的首选器件之一。

8.5.1 FPGA 中编程实现逻辑功能的基本原理

在 FPGA 中, 通过编程实现多种组合逻辑功能的原理图如图 8.5.1 所示。图中 $T_0 \sim T_7$ 为 NMOS 开关管或 CMOS 传输门构成的门阵列, $M_0 \sim M_3$ 为静态随机存取存储器 SRAM。A、B 为两个输入逻辑变量, F 为输出逻辑变量。通过向存储单元 $M_0 \sim M_3$ 写入不同的数据(编程), 便可实现各种逻辑运算关系。例如, 在 $M_3M_2M_1M_0 = 0011$ 的条件下, 当 AB 为 10 时, T_0 、 T_1 导通, F 为 1; 同样, AB 为 01 时, T_2 、 T_3 导通, F 也为 1, 因此 F 的逻辑函数表达式为 $F = A\bar{B} + \bar{A}B$ 。表 8.5.1 给出了 $M_0 \sim M_3$ 不同取值时输出与输入的逻辑关系。

表 8.5.1 图 8.5.1 中的输出与输入的逻辑关系

$M_3M_2M_1M_0$	F	$M_3M_2M_1M_0$	F
0 0 0 0	0	1 0 0 0	AB
0 0 0 1	$A\bar{B}$	1 0 0 1	A
0 0 1 0	$\bar{A}B$	1 0 1 0	B
0 0 1 1	$A\bar{B} + \bar{A}B = A \oplus B$	1 0 1 1	$A \cdot B$
0 1 0 0	$\bar{A}\bar{B}$	1 1 0 0	$AB + \bar{A}\bar{B} = \overline{A \oplus B}$
0 1 0 1	\bar{B}	1 1 0 1	$A + \bar{B}$
0 1 1 0	\bar{A}	1 1 1 0	$\bar{A} + B$
0 1 1 1	$\bar{A} + \bar{B} = \overline{AB}$	1 1 1 1	1

图 8.5.2 为多路选择器的电路原理图。当 M_1 、 M_0 中写入不同的数据时, 在输出端便得到不同的输入信号。功能表如表 8.5.2 所示。

① Field Programmable Gate Array 的缩写。

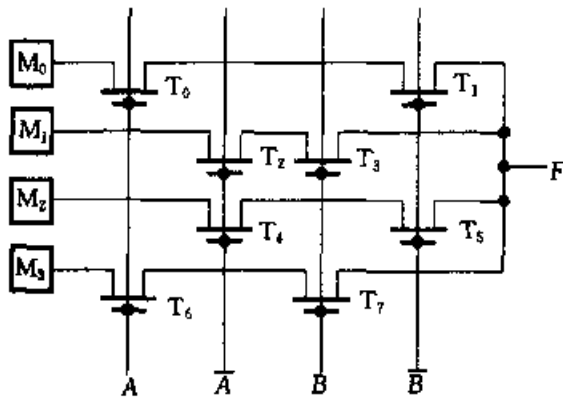


图 8.5.1 2 输入变量的逻辑函数产生电路

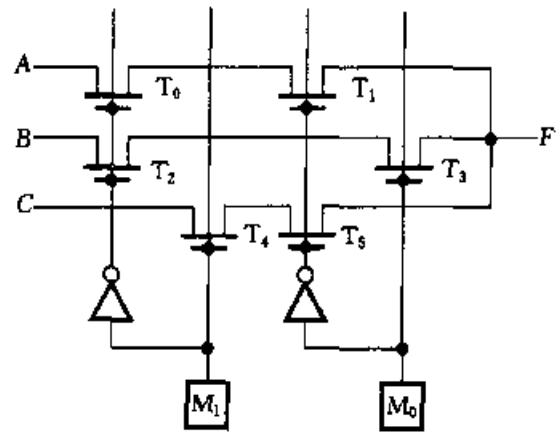


图 8.5.2 多路选择器电路

表 8.5.2 图 8.5.2 的功能表

$M_1 M_0$	F	$M_1 M_0$	F
0 0	A	1 0	C
0 1	B		

由上述两例看出, FPGA 中实现各种组合逻辑功能的原理是, 通过对各存储单元的编程, 来控制门阵列中门的“开”与“关”, 从而实现不同的逻辑功能。编程过程实际上是对各存储单元写入数据的过程, 这些数据也称为编程数据。存储单元中的编程数据一旦确定, 门阵列的逻辑关系也就确定了。

在上述门阵列的基础上再增加触发器, 便可构成既可实现组合逻辑功能又可实现时序逻辑功能的基本逻辑单元电路。FPGA 中就是由很多类似这样的基本逻辑单元来完成各种复杂逻辑功能的。

由于 SRAM 中的数据理论上可以进行无限次写入, 所以, 基于 SRAM 技术的 FPGA 可以进行无限次的编程。

8.5.2 现场可编程门阵列结构

目前, 虽然生产现场可编程门阵列器件的厂家较多, 且产品种类亦多, 但它们的基本组成大致相似。这里以 Xilinx 公司的 XC4000E 系列为例, 介绍 FPGA 的内部结构及各模块的功能。

FPGA 的结构示意图如图 8.5.3 所示。它主要由可编程逻辑模块 CLB^①、输入/输出模块 IOB^②、可编程连线资源 3 个部分组成。可编程逻辑模块 CLB 是

① Configurable Logic Block 的缩写, 也称为可配置逻辑模块。

② Input/Output Block 的缩写。

实现各种逻辑功能的基本单元,其中包括组合逻辑、时序逻辑、RAM及各种运算功能。CLB以 $n \times n$ 阵列形式分布在FPGA中,同一系列中不同型号的FPGA,其阵列规模也不同,详见表8.5.3;可编程的输入/输出模块IOB是芯片外部引

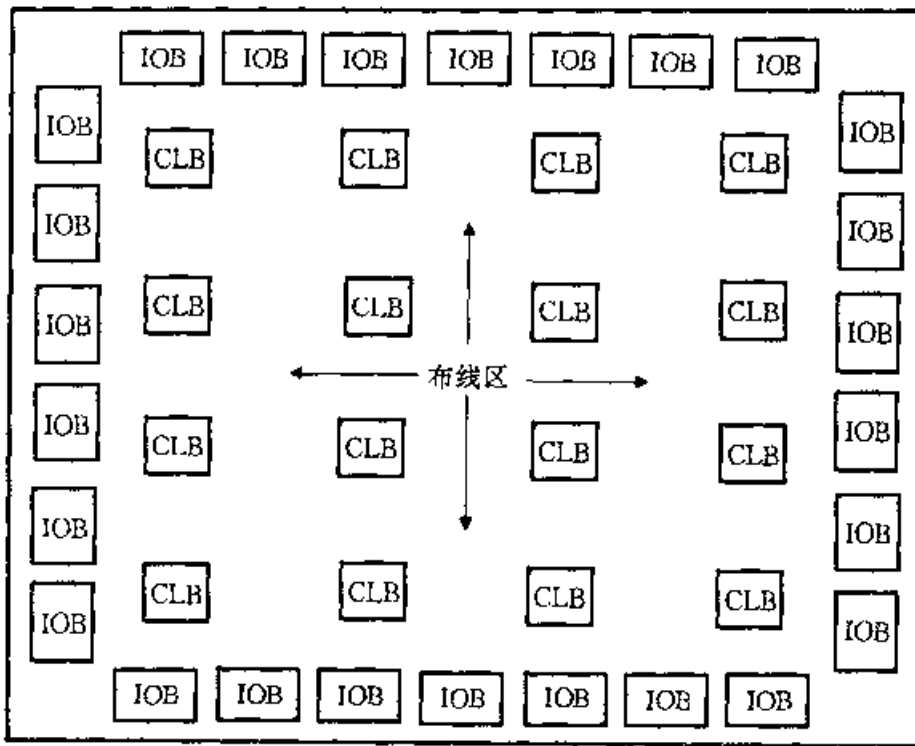


图 8.5.3 FPGA 的结构示意图

表 8.5.3 XC4000E 系列产品规模

器件	XC4003E	XC4005E	XC4006E	XC4008E	XC4010E	XC4013F	XC4020E	XC4025E
门数	3 000	5 000	6 000	8 000	10 000	13 000	20 000	25 000
CLB 个数 (行×列)	100 (10×10)	196 (14×14)	256 (16×16)	324 (18×18)	400 (20×20)	576 (24×24)	784 (28×28)	1 024 (32×32)
IOB 个数	80	112	128	144	160	192	224	256
触发器 (个)	360	616	768	936	1 120	1 536	2 016	2 560
数据结构 长度(Bit)	126	166	186	206	226	266	306	346
数据结构 数量(个)	428	572	644	716	788	932	1 076	1 220
编程数据 总量(Bit)	53 936	94 960	119 792	147 504	178 096	247 920	329 264	442 128
PROM 容量(Bit)	53 984	95 008	119 840	147 552	178 144	247 968	329 312	442 176

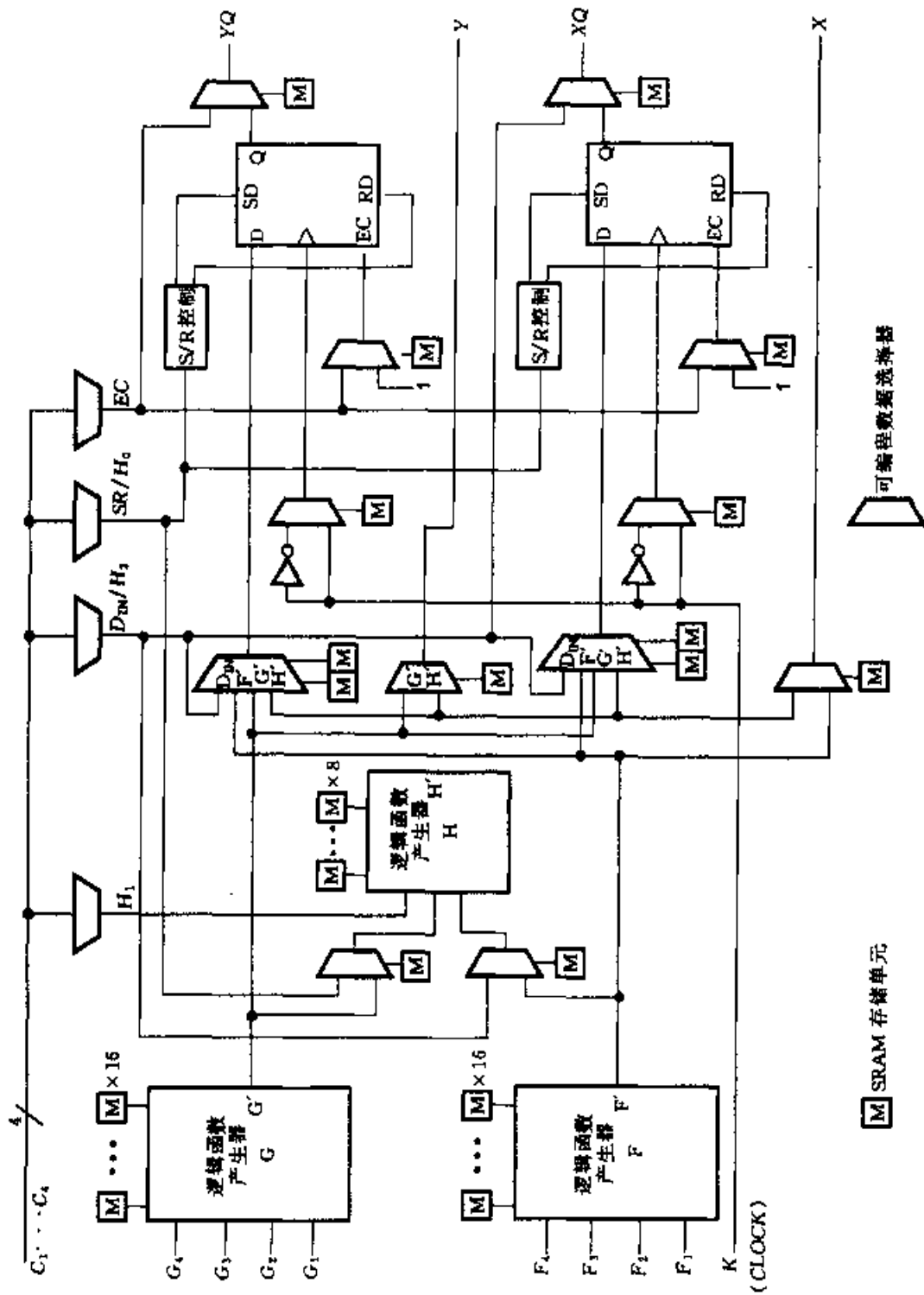


图 8.5.4 简化的 CLB 原理框图

脚数据与内部数据进行交换的接口电路,通过编程可将 I/O 引脚设置成输入、输出和双向等不同的功能。IOB 分布在芯片的四周;CLB 之间的空隙部分是布线区,分布着可编程连线资源,这些资源包括金属导线、可编程开关点和可编程开关矩阵。金属导线以纵横交错的格栅状结构分布在两个层面(一层为横向线段,一层为纵向线段),有关的交叉点上连接着可编程开关或可编程开关矩阵,通过对可编程开关和可编程开关矩阵的编程实现 CLB 与 CLB 之间、CLB 与 IOB 之间、以及全局信号与 CLB 和 IOB 之间的连接。

通过对内部静态存储单元的编程来确定每个模块的功能,存储在这些存储单元中的编程数据决定了 FPGA 的功能。

下面简要介绍这 3 部分的结构和功能。

1. 可编程逻辑模块 CLB

可编程逻辑模块 CLB 是 FPGA 中的基本逻辑单元电路,它可实现绝大多数的逻辑功能。XC4000E 系列 CLB 简化的原理框图如图 8.5.4 所示。CLB 主要由逻辑函数产生器、触发器、进位逻辑(图中未画出)、编程数据存储单元、数据选择器及其他控制电路组成,有 13 个输入端和 4 个输出端(不包括进位逻辑电路的输入、输出信号)。其中 13 个输入包括两组 4 变量逻辑函数输入端 $G_1 \sim G_4$ 和 $F_1 \sim F_4$ 、4 个控制信号输入端 $C_1 \sim C_4$ 和 1 个时钟输入端 K ;4 个输出包括 2 个组合逻辑输出端 X 和 Y 和 2 个时序逻辑输出端 XQ 和 YQ , XQ 和 YQ 还可分别作为 D_N/H_2 和 EC 信号的直通输出。

控制信号 $C_1 \sim C_4$ 通过选择器转换为 4 个内部信号 H_1 、 D_N/H_2 、 SR/H_0 和 EC 。 H_1 、 D_N/H_2 和 SR/H_0 可作为 CLB 中逻辑函数产生的输入信号, SR/H_0 也可作为 CLB 中 D 触发器的置位/复位控制信号, EC 为 D 触发器的时钟使能控制信号。

3 个逻辑函数产生器 G 、 F 和 H 的工作原理如 8.5.1 节所述, G 、 F 是两个独立的 4 变量逻辑函数产生器, $G_1 \sim G_4$ 和 $F_1 \sim F_4$ 分别为 G 和 F 的输入, G' 和 F' 为输出。它们各有 16 个编程数据存储单元,当给这些存储单元写入特定的数据时,便可实现各自特定的逻辑运算。这些编程数据存储单元也称为查找表^①。

第 3 个逻辑函数产生器 H 为 3 变量逻辑函数产生器,其中的两个输入分别由两个数据选择器控制,可以选择 G 的输出 G' 或外部输入 H_0 以及 F 的输出 F' 或外部输入 H_2 ,第 3 个输入取自外部输入 H_1 。由此可见,经过 3 个逻辑函数产生器的两级组合,在 H' 端可以实现多达 9 个变量的组合逻辑函数。

^① Look-up Table。

逻辑函数产生器产生的信号可从两个输出端送到 CLB 外,即 F' 或 H' 通过 2 选 1 选择器可以连到 X 输出端, G' 或 H' 通过 2 选 1 选择器可以连到 Y 输出端。

CLB 中有两个边沿 D 触发器,通过 4 选 1 选择器可分别选择 D_{IN} 、 F' 、 G' 和 H' 之一作为触发器的输入信号。两个 D 触发器共用时钟脉冲 K ,可以通过各自的选择器选择上升沿触发或下降沿触发。时钟使能 EC 也可通过选择器选择直接受 CLB 内部控制信号 EC 控制或接高电平。异步置位/复位信号 SD/RD 共用一个 CLB 内部控制信号 SR ,通过各自的 S/R 控制电路,提供两触发器各自的置位/复位信号。

为了提高 FPGA 算术运算的速度,每个 CLB 的 G 和 F 两个逻辑函数产生器

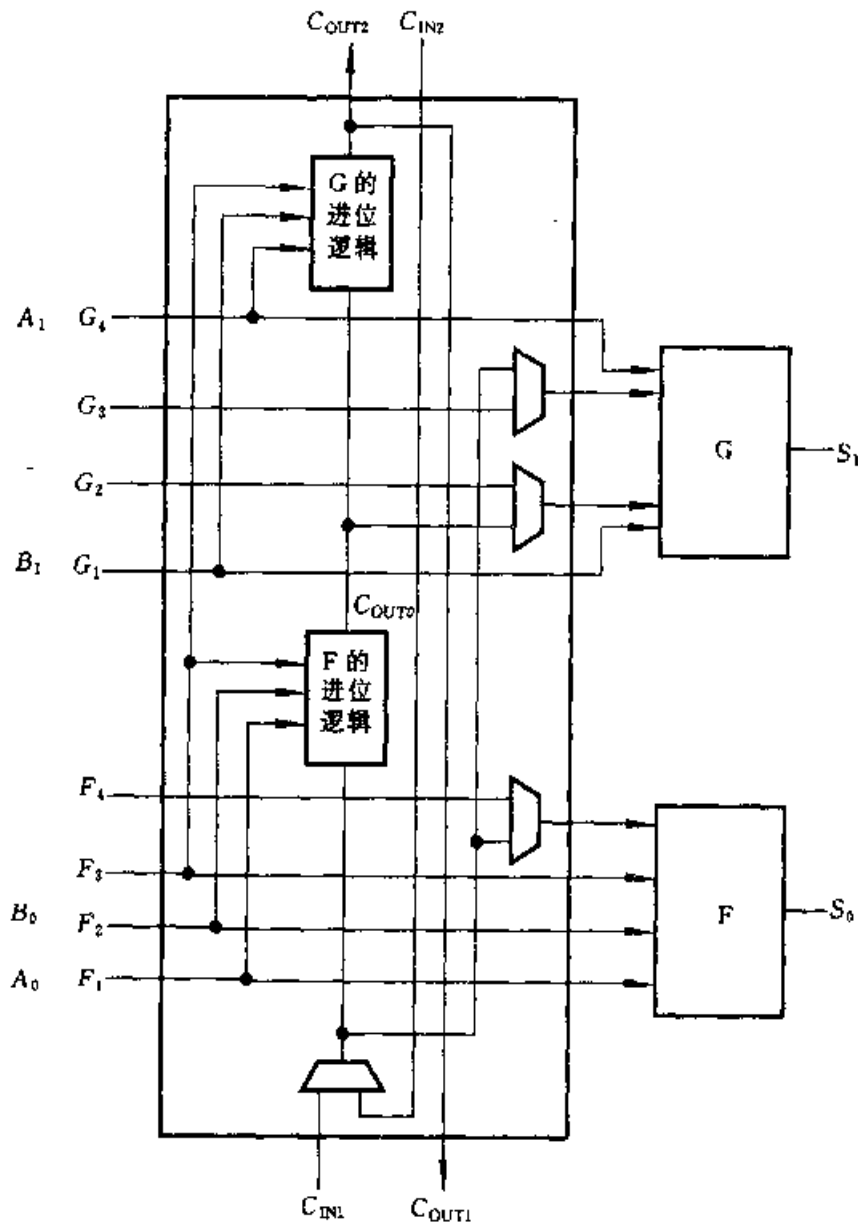


图 8.5.5 具有快速进位的 2 位二进制加法器时的原理框图

中还设计了快速进位逻辑电路。G、F 作为具有快速进位的 2 位二进制加法器时的原理框图如图 8.5.5 所示。除了被加数和加数输入以外,还有两个进位输入端和两个进位输出端。设计两组进位输入、输出主要为了串接方便,使用时只选择其中的一组。将多个 CLB 串接起来,便可完成多位二进制数的快速加法运算。

CLB 除了可以实现一般的组合逻辑功能和时序逻辑功能外,G 和 F 的各 16 个编程数据存储单元还可作为读/写存储器使用。此时,一个 CLB 可以构成两个容量为 16×1 位的 RAM 或一个 32×1 位 RAM。数据写入可采用边沿触发或电平触发两种方式。函数产生器 G 的编程数据存储单元还可设置成 16×1 位的双口 RAM(可以同时读操作和写操作)。XC4000 系列支持的 1 个 CLB 的存储器容量、触发方式和单口及双口模式的关系如表 8.5.4 所示。CLB 构成两个 16×1 位单口 RAM 的原理框图如图 8.5.6 所示。当作为 RAM 使用时,CLB 控制信号的功能将有所变化, D_{IN}/H_2 、 H_1 和 SR/H_0 变为两个数据输入 D_1 、 D_0 和写使能输入 WE ,时钟脉冲 K 作为 RAM 的写入脉冲, $G_1 \sim G_4$ 、 $F_1 \sim F_4$ 分别为两个 RAM 的地址信号, G' 、 F' 为两个数据输出。写入时,地址信号 $G_1 \sim G_4$ 、 $F_1 \sim F_4$ 经写地址译码器译码,选通要写入数据的存储单元,在写使能

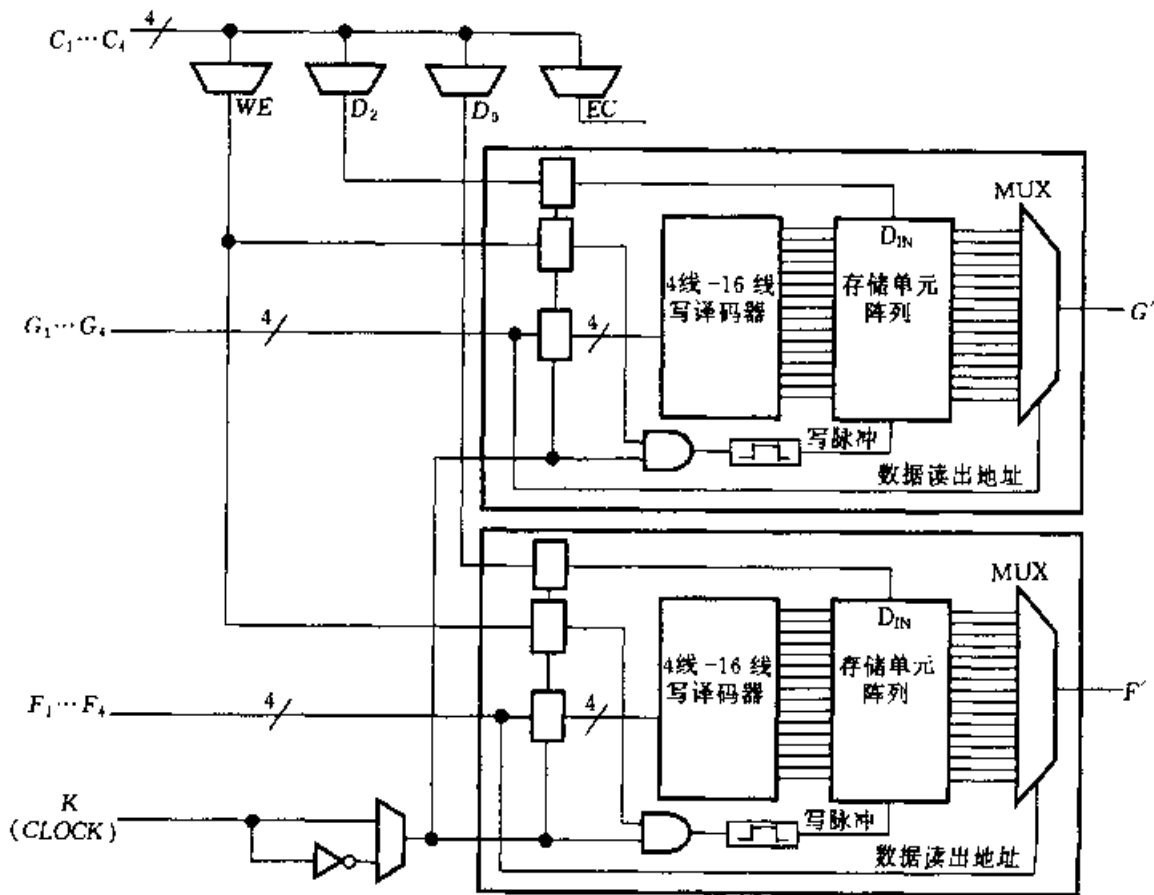


图 8.5.6 CLB 构成的两个 16×1 位单口 RAM 原理框图

WE 和写时钟 K 的控制下将 D_1 、 D_0 分别写入 G 和 F 的存储单元；读出时，只要给出读地址信号，数据便可从 RAM 中立即读出。将地址信号 $G_1 \sim G_4$ 和 $F_1 \sim F_4$ 并接在一起，便可构成 16×2 位单口 RAM。

表 8.5.4 存储器容量、触发方式和单口及双口模式的关系

	16×1	16×2	32×1	边沿触发	电平触发
单口	✓	✓	✓	✓	✓
双口	✓			✓	

双口 RAM 原理框图如图 8.5.7 所示。此时， D_0 为 16×1 RAM 阵列的数据输入端， G' 为数据输出端， $G_1 \sim G_4$ 为函数产生器 G 存储单元阵列的读地址线，写地址信号由 $F_1 \sim F_4$ 提供，即 G 存储阵列的读、写地址线是分开的。这样便可对函数产生器 G 中的 RAM 同时进行读写。

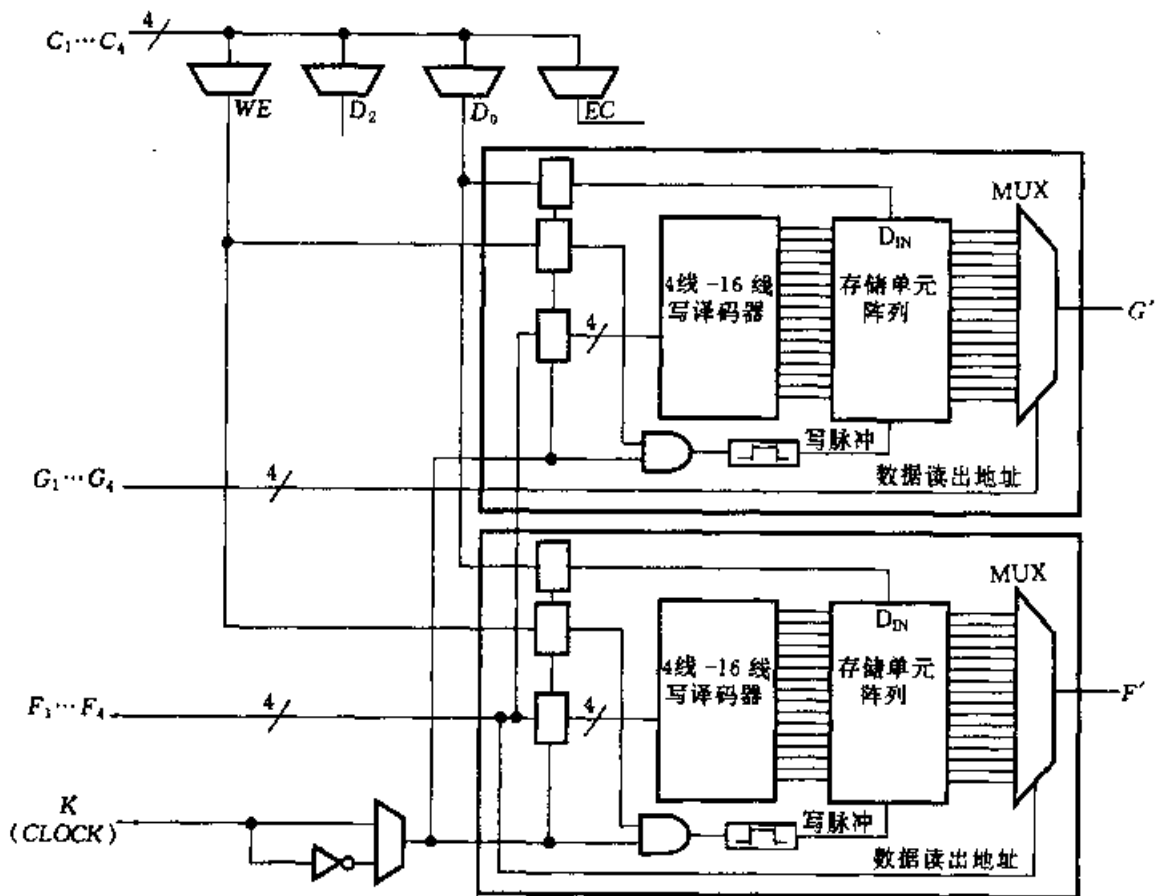


图 8.5.7 双口 RAM 原理框图

2. 输入/输出模块 IOB

IOB 是 FPGA 外部封装引脚和内部逻辑间的接口。每个 IOB 对应一个封装引脚,通过在 IOB 有关的编程数据存储单元中写入不同的数据,可将引脚定义为输入、输出和双向功能。XC4000E 系列 IOB 的简化原理框图如图 8.5.8 所示。

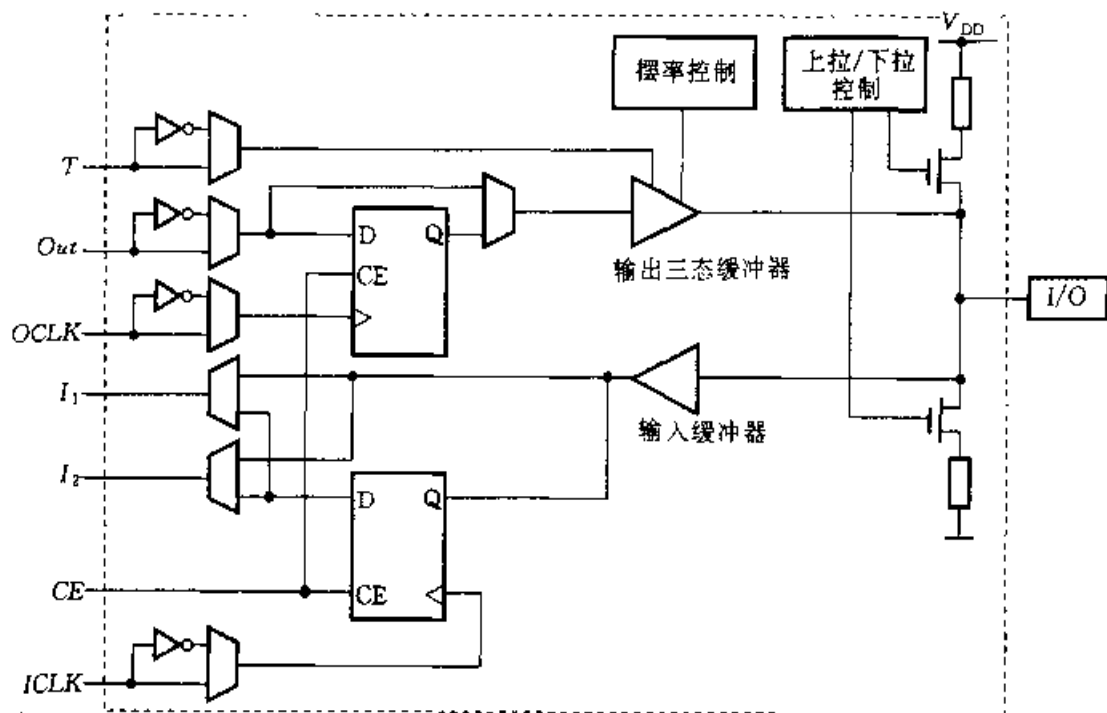


图 8.5.8 简化的 IOB 原理框图

IOB 中有输入和输出两条信号通路。当 I/O 引脚用作输出时,内部逻辑信号由 Out 端进入 IOB 模块,通过对选择器编程可选择是否反相,再由下一个选择器选择是直接送三态缓冲器还是经过 D 触发器寄存后再送三态缓冲器。三态输出缓冲器的使能控制信号 T 也可以通过编程定义为高电平有效或低电平有效,当 T 有效时,输出信号经缓冲器输出到 I/O 引脚。另外,对输出缓冲器还可进行摆率(电平跳变的速率)控制,可以选择快速和慢速两种方式,选择快速方式可适应频率较高的信号输出;而选择慢速方式则可减小功耗和降低噪声。当 I/O 引脚用作输入时,引脚上的输入信号经过输入缓冲器,可以直接由 I_1 、 I_2 进入内部逻辑电路,也可以经 D 触发器寄存后输入到内部逻辑电路中。输入通路和输出通路中的两个 D 触发器共用一个时钟使能控制信号 CE ,但它们有各自的时钟脉冲 $ICLK$ 和 $OCLK$,且可编程为上升沿触发或下降沿触发。

没有用到的引脚可由上、下拉控制电路控制,通过上拉电阻接电源电压或下拉电阻接地,以免引脚浮空产生震荡而增加附加功耗和系统噪声。上、下拉电阻

的阻值为 $50 \sim 100 \text{ k}\Omega$ 。

3. 可编程连线资源

可编程连线资源分布在 CLB 阵列的行、列间隙上,由水平和垂直的两层金属线段组成栅状结构。XC4000E 系列中有 5 种类型的可编程连线:单长线、双长线、长线、全局时钟线和进位链。图 8.5.9 给出了这些连线资源的示意图(图中没有画出进位链,PSM^① 是可编程开关矩阵,其结构如图 8.5.10 所示)。

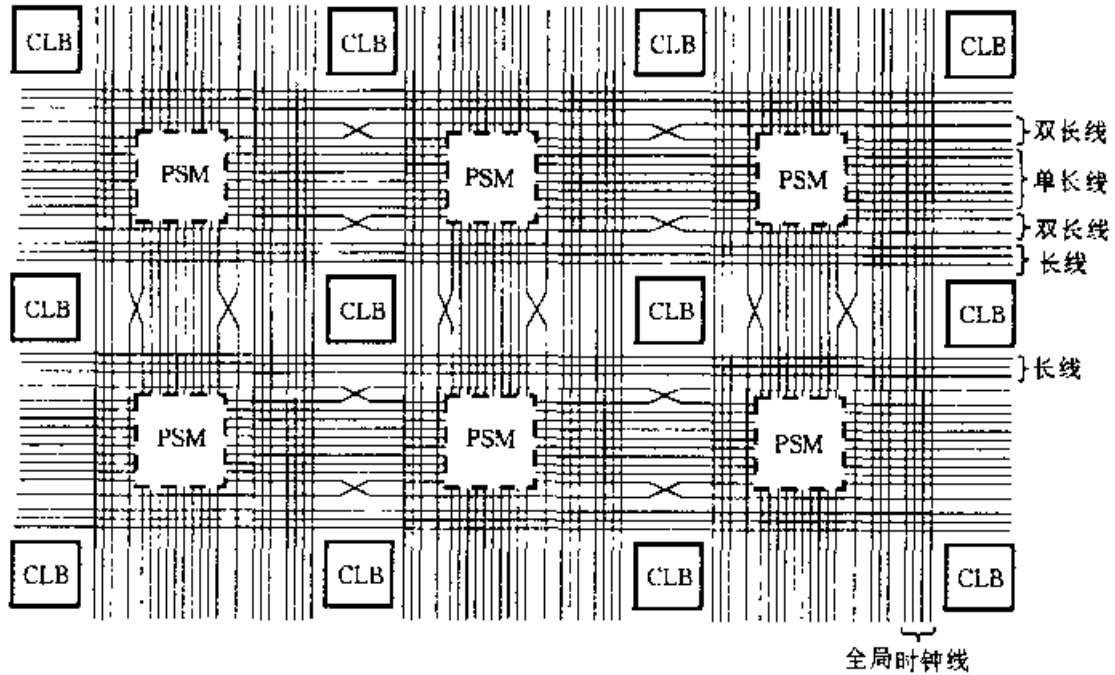


图 8.5.9 可编程连线资源示意图

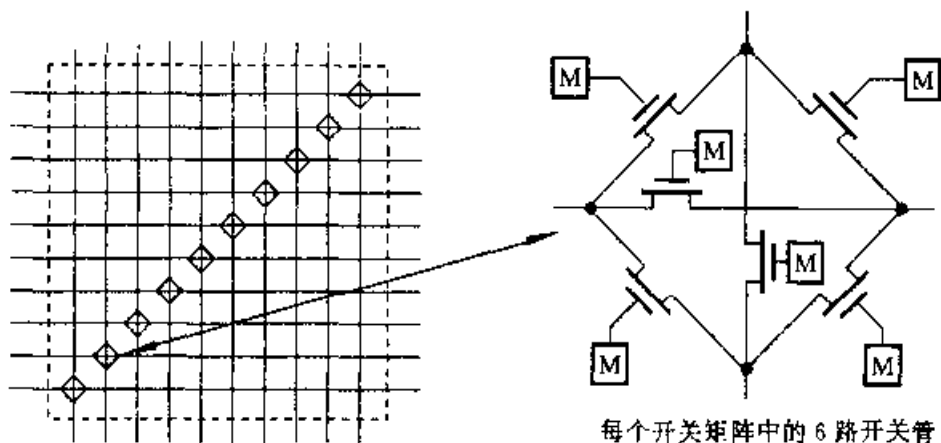


图 8.5.10 可编程开关矩阵及结构

① Programmable Switching Matrix 的缩写。

单长线的长度相当于两个 CLB 之间的距离,且通过 PSM 与其他单长线相连。双长线的长度相当于 2 倍的单长线的长度,每根双长线都从一个开关矩阵出发,绕过相邻的开关矩阵进入下一个开关矩阵。双长线在电路中成对出现。长线不经过开关矩阵,其长度可穿越整个芯片,每条长线的中点处有一可编程的分离开关,可将长线分成 2 条独立的布线通道。全局时钟线只分布在垂直方向上。

CLB 输入、输出的详细布线连接图如图 8.5.11 所示(图中未画出进位逻辑连线)。由图看出,CLB 的输入、输出均匀地分布在四周,通过对可编程开关点的编程,可以将它们连接到单长线、双长线或长线上。阴影部分为可编程开关矩阵,XC4000E 系列中一个 CLB 的连线资源如表 8.5.5 所示。其中,全局时钟线只有 4 根垂直线,进位逻辑有 2 根垂直线。

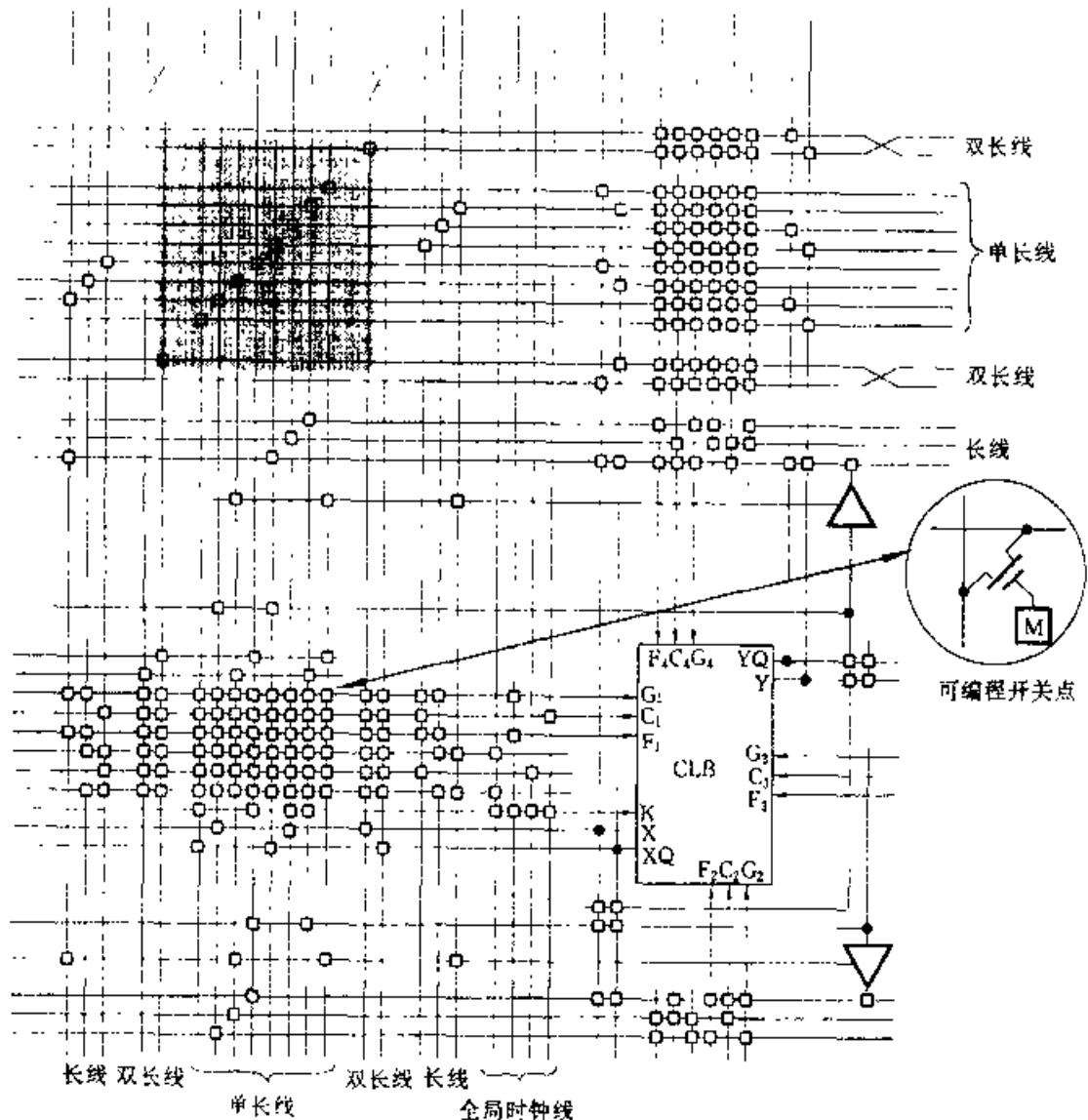


图 8.5.11 CLB 输入、输出的布线连接图

单长线与 CLB 输入、输出间有许多直接连接点,因此有很高的布线成功率。单长度连线提供了最好的互连灵活性和相邻模块的快速布线。由于信号每经过一个开关矩阵都要产生一定的延时,所以单长线不适合长距离传输的信号。与单长线相比,双长线减少了经过矩阵开关的数量,更有效地提供了中等距离的信号通路,提高了系统工作速度。长线通常用于高扇出和时间要求苛刻的信号网。全局时钟垂直长线主要用来提供全局的时钟信号和高扇出的控制信号。

表 8.5.5 XC4000E 系列中一个 CLB 的连线资源

	单长线	双长线	长线	全局时钟线	进位逻辑线	合计
垂直线	8	4	6	4	2	24
水平线	8	4	6	0	0	18

8.5.3 编程实现原理简介

1. 编程数据存储单元阵列结构及编程数据流

FPGA 中的 CLB、IOB 的功能和连线资源的连接方式,都是由它们相应的存储单元中的数据确定的,将编程数据装入 FPGA 中存储单元的过程称为编程。数据的写入只在编程过程中完成,在正常工作期间,存储单元只用来提供固定的连续的控制信号。编程存储单元以阵列形式分布在 FPGA 中,其结构如图 8.5.12 所示。可以看出,数据以串行方式移入移位寄存器,而地址移位寄存器选中存储单元阵列的 1 根字线,由写信号控制将数据移位寄存器中 1 个字的数据写入存储单元。存储单元为 5 管 SRAM,只有 1 根位线,不像图 8.1.1 中 6 管 SRAM 那样有 1 对互逆的位线(B 和 \bar{B})。

存储单元阵列的规模与 FPGA 中 CLB 阵列的规模有着直接的联系,XC4000E 系列中一个 CLB 对应 10×36 个存储单元,另外,还要加上四周 IOB 对应存储单元。

编程数据以某种数据结构形式组成数据流装入 FPGA 中,数据结构的长度和个数与器件类型有关(见表 8.5.3)。在 XC4000E 系列中:

结构的长度 = 1 个起始位(0) + (FPGA 中 CLB 阵列行数 \times 10) + 顶部 7 位 + 底部 13 位 + 4 位校验位或结构结束位(xxxx 或 0110)。

结构的数量 = (FPGA 中 CLB 阵列的列数 \times 36) + 左边 26 位 + 右边 41 位 + 1 个附加位。

数据流的构成形式如下:

11111111	伪数据位	} 数据流头
0010	前同步码	
<24 - Bit Length Count>	24 位的编程数据长度计数器	
1111	伪数据结束位	
0<DATA (0)>xxxx	第 1 个结构	} 数据流体
0<DATA (1)>xxxx	第 2 个结构	
0<DATA (2)>xxxx	第 3 个结构	
⋮		
0<DATA (n-1)>xxxx	第 n 个结构	
01111111	后同步码(Postamble Code)	} 数据流结束码

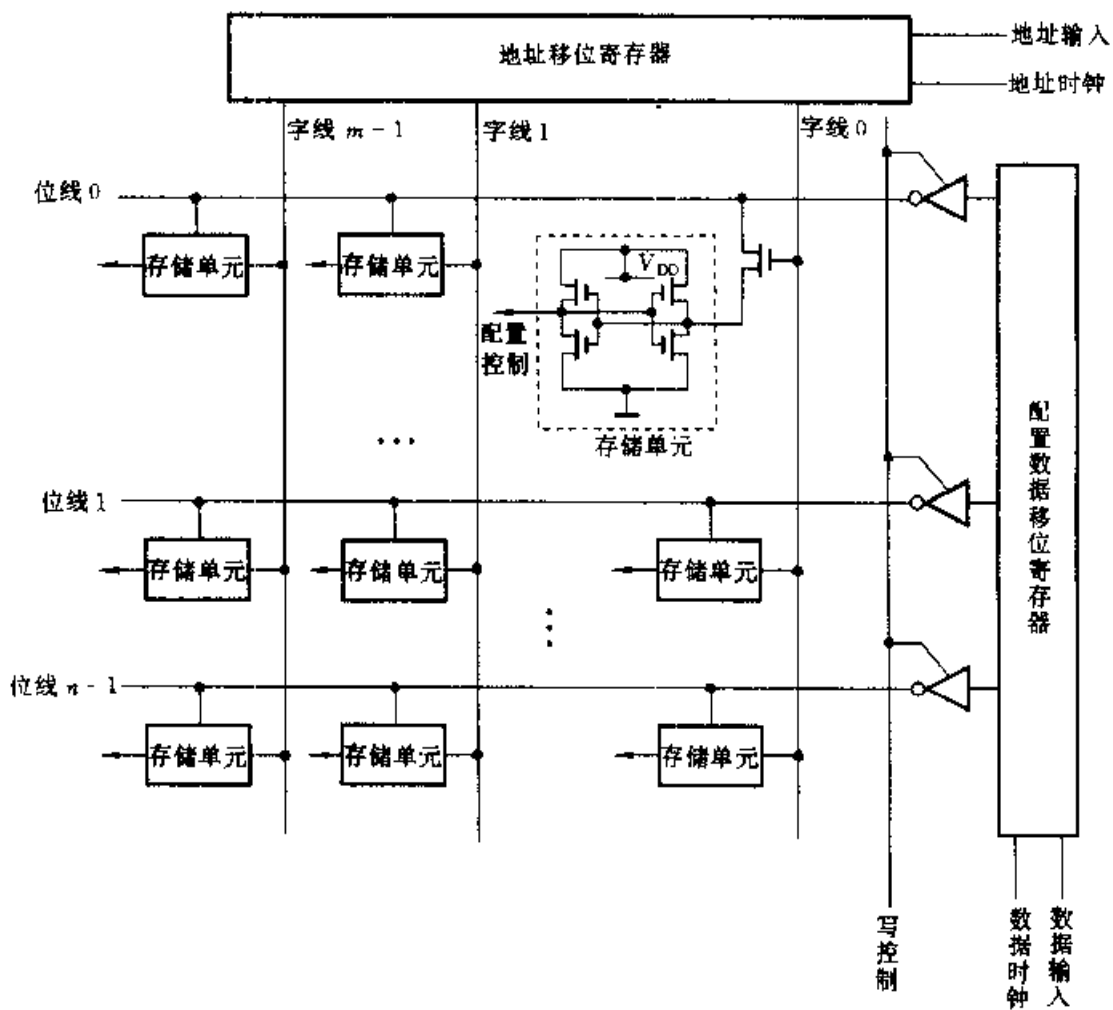


图 8.5.12 编程数据存储单元阵列结构

由此可见,编程数据流的总长度 = 数据流头部位数(40位) + 结构的长度 × 结构的数量 + 流结束码(8位)。

编程数据流由开发系统(如 Xilinx 公司的 Foundation 软件开发系统)自动生成。开发系统将设计输入转换成网表文件,并自动对逻辑电路进行划分、布局和布线,然后按 PROM 格式产生编程数据流,并将编程数据流存入 PROM 中。

2. 编程数据流的装入

由于 SRAM 在掉电后其内部的数据会丢失,所以基于 SRAM 的 FPGA 必须配置一个 PROM 芯片,用以存放 FPGA 的编程数据。这里,假设 PROM 中已经存放了编程数据流,下面简要介绍如何将 PROM 中的编程数据装入到 FPGA 中。

XC4000E 系列有 6 种编程模式,当在 FPGA 的 3 个专用引脚 M2、M1 和 M0 上输入不同的逻辑电平时,便可选择一种编程模式进行数据装入。6 种编程方式如表 8.5.6 所示。有 3 种自行装入的主模式,2 种外部模式和 1 种从串模式。

主模式利用 FPGA 内部高速反相器和外部晶振构成的振荡器产生编程时钟信号(CCLK)来驱动装有编程数据的 PROM。当选择主串模式时,编程数据从 PROM 中以串行方式装入 FPGA 中,此时必须用有串行功能的 PROM。主串装入模式电路如图 8.5.13 所示。当选择主并模式时,FPGA 除了提供时钟信号外,还提供地址信号,编程数据从 PROM 中以并行方式输出,FPGA 收到后,在内部转换成串行数据结构,再写入编程数据存储单元。主并模式电路如图 8.5.14 所示。上行和下行的区别主要是地址步进的方式不同,上行时,地址从 00000H 开始递增,下行时,地址从 3FFFFH 递减。

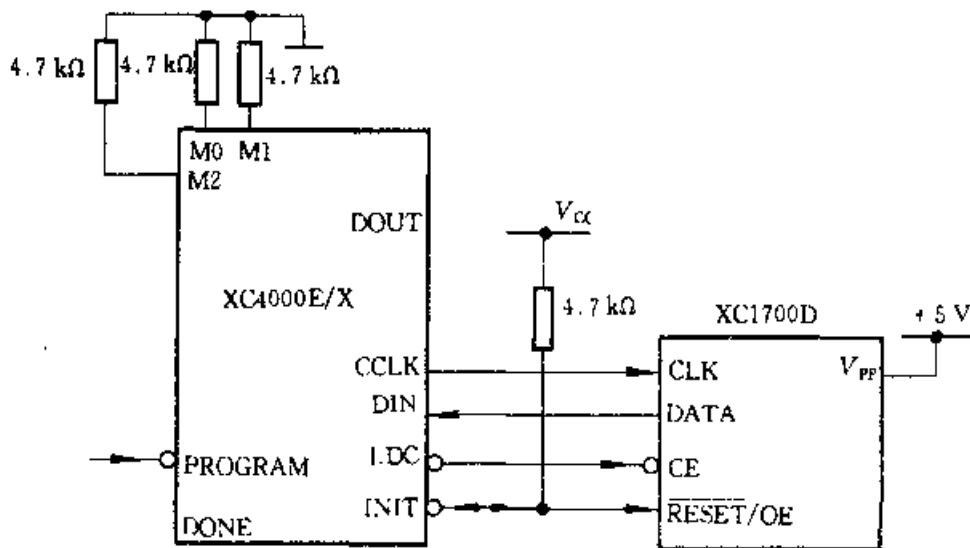


图 8.5.13 主串模式电路

表 8.5.6 XC4000E 系列的 6 种编程模式

模式	M2	M1	M0	CCLK	数据
主串	0	0	0	输出	串行位
从串	1	1	1	输入	串行位
主并上行	1	0	0	输出	并行字节,地址由 00000H 递增
主并下行	1	1	0	输出	并行字节,地址由 3FFFFH 递减
外部同步 ^①	0	1	1	输入	并行字节
外部异步	1	0	1	输出	并行字节

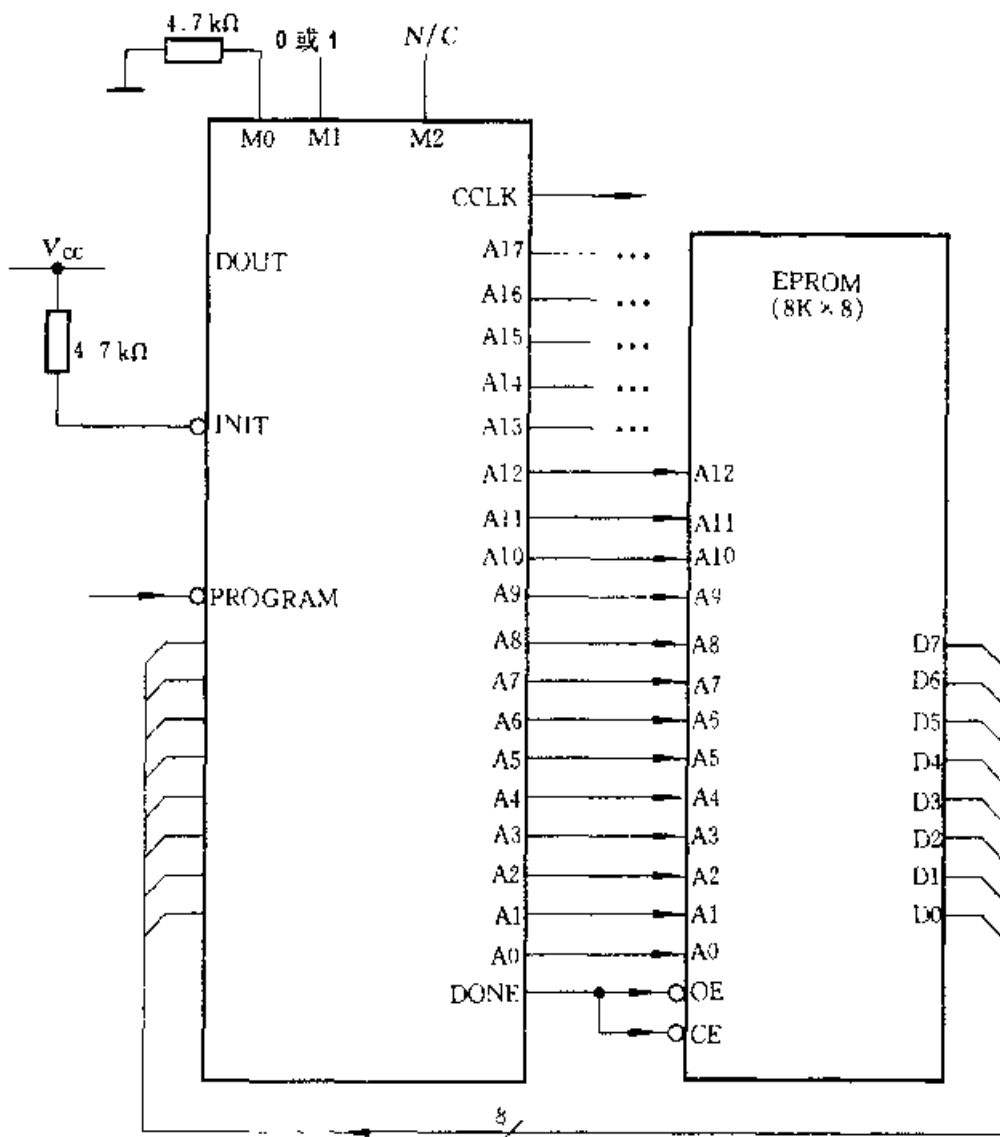


图 8.5.14 主并模式电路

① 外部同步模式也可认为是从并模式

从串模式和外部同步模式必须用外设产生时钟信号来驱动 FPGA 编程数据的装入,其他与主模式类似。

复习思考题

- 8.5.1 FPGA 中实现各种组合逻辑功能的原理是什么?
- 8.5.2 FPGA 在结构上由哪几个部分组成?各部分的主要功能是什么
- 8.5.3 XC4000E 系列中,1 个 CLB 能否从组合输出端同时输出 G' 、 F' 和 H' 3 个逻辑函数,为什么?
- 8.5.4 可编程开关点和可编程开关矩阵中的连接点有何区别?
- 8.5.5 为什么在 FPGA 构成的数字系统中要配备 1 个 PROM 或 EPROM?
- 8.5.6 FPGA 与 CPLD 上要有哪些区别?它们各有什么特点?

小 结

• 半导体存储器是现代数字系统特别是计算机中的重要组成部分,它可分为 RAM 和 ROM 两大类,绝大多数属于 MOS 工艺制成的大规模集成电路。

• RAM 是一种时序逻辑电路,具有记忆功能。它存储的数据随电源断电而消失,因此是一种易失性的读写存储器。它包含有 SRAM 和 DRAM 两种类型,前者用触发器记忆数据,后者靠 MOS 管栅极电容存储数据。因此,在不停电的情况下,SRAM 的数据可以长久保持,而 DRAM 则必须定期刷新。

• ROM 是一种非易失性的存储器,它存储的是固定数据,一般只能被读出。根据数据写入方式的不同,ROM 又可以分成固定 ROM 和可编程 ROM。后者又可以细分为 PROM、EPROM、 E^2 PROM 和快闪存储器等。特别是 E^2 PROM 和快闪存储器可以进行电擦写,已兼有了 RAM 的特性。

• 目前,可编程逻辑器件(PLD)的使用越来越广泛,用户可以自行设计该类器件的逻辑功能。它们具有集成度高、可靠性高、处理速度快和保密性好等特点。PAL 和 GAL 是两种典型的可编程逻辑器件,其电路结构的核心都是与-或阵列。而 GAL 器件的输出部分增加了输出逻辑宏单元 OLMC,因此比 PAL 具有更强的功能和灵活性。

• CPLD 是在 GAL 基础上发展起来的复杂可编程逻辑器件,采用先进的 E^2 CMOS 工艺,集成度更高,且可以在线编程。

• FPGA 是基于 SRAM 的可编程器件,它以功能很强的 CLB 为基本逻辑单元,可以实现各种复杂的逻辑功能,同时还可以兼作 RAM 使用。FPGA 是目前规模最大、密度最高的可编程器件。

本章附录:

附表 8.1 MCM6264P20 读周期定时参数

参数	标准符号	替代符号	最小	最大
读周期时间/ns	t_{AVAV}	t_{RC}	20	—
地址存取时间/ns	t_{AVQV}	t_{AA}	—	20
片允许选取时间/ns	t_{ELQV}	t_{ACS}	—	20
输出允许选取时间/ns	t_{GLQV}	t_{OE}	—	10
片允许低电平到输出有效/ns	t_{ELQX}	t_{LZ}	4	—
输出允许低电平到输出有效/ns	t_{GLQX}	t_{LZ}	0	9
允许高电平到输出高阻态/ns	t_{EHQZ}	t_{HZ}	0	—
输出允许高电平到输出高阻态/ns	t_{GHQZ}	t_{HZ}	0	8

注: MCM6264P20 中的 P20 是型号中描述读/写周期长短的参数, 此处表示该型号的读/写周期为 20 ns。

附表 8.2 MCM6264P20 写周期定时参数

参数	标准符号	替代符号	最小
写周期时间/ns	t_{AVAV}	t_{WC}	20
地址建立时间/ns	t_{AVWL}	t_{AS}	0
地址有效到写结束/ns	t_{AVWH}	t_{AW}	15
写脉冲宽度/ns	t_{WLWH} t_{WLEH}	t_{WP}	15
写脉冲宽度, G 为高电平/ns	t_{WCLWH} t_{WLEH}	t_{WP}	12
数据有效到写结束/ns	t_{DVWH}	t_{DW}	8
数据保持/ns	t_{WHDX}	t_{DH}	0
写恢复时间/ns	t_{WHAX}	t_{WR}	0

附表 8.3 $\mu\text{PD41256}-10$ 的定时参数

参数	符号	最小	最大
随机读或写周期时间/ns	t_{RC}	200	
从 RAS 起存取时间/ns	t_{RAC}		100
从 CAS 起存取时间/ns	t_{CAC}		50
输出缓冲关断延迟/ns	t_{OFF}	0	25
RAS 预充电时间/ns	t_{RP}	90	
RAS 脉冲宽度/ns	t_{RAS}	100	10k
RAS 保持时间/ns	t_{RSH}	50	
CAS 脉冲宽度/ns	t_{CAS}	50	10k
CAS 保持时间/ns	t_{CSH}	100	
RAS 到 CAS 延迟时间/ns	t_{RCD}	20	50
CAS 到 RAS 预充电时间/ns	t_{CRP}	10	
CAS 预充电时间/ns	t_{CPN}	25	
RAS 预充电 CAS 保持时间/ns	t_{RPH}	0	
行地址建立时间/ns	t_{ASR}	0	
行地址保持时间/ns	t_{RAH}	10	
列地址建立时间/ns	t_{ASC}	0	
列地址保持时间/ns	t_{CAH}	15	
列地址保持时间(参考 RAS)/ns	t_{AR}	65	
读命令建立时间/ns	t_{RCS}	0	
读命令保持时间(参考 RAS)/ns	t_{RRH}	10	
读命令保持时间(参考 CAS)/ns	t_{RCH}	0	
写命令保持时间/ns	t_{WCH}	25	
写命令保持时间(参考 RAS)/ns	t_{WCN}	75	
写命令脉冲宽度/ns	t_{WP}	15	
数据输入建立时间/ns	t_{DS}	0	
数据输入保持时间/ns	t_{DH}	25	
数据输入保持时间(参考 RAS)/ns	t_{DHR}	75	
写命令建立时间/ns	t_{WCS}	0	

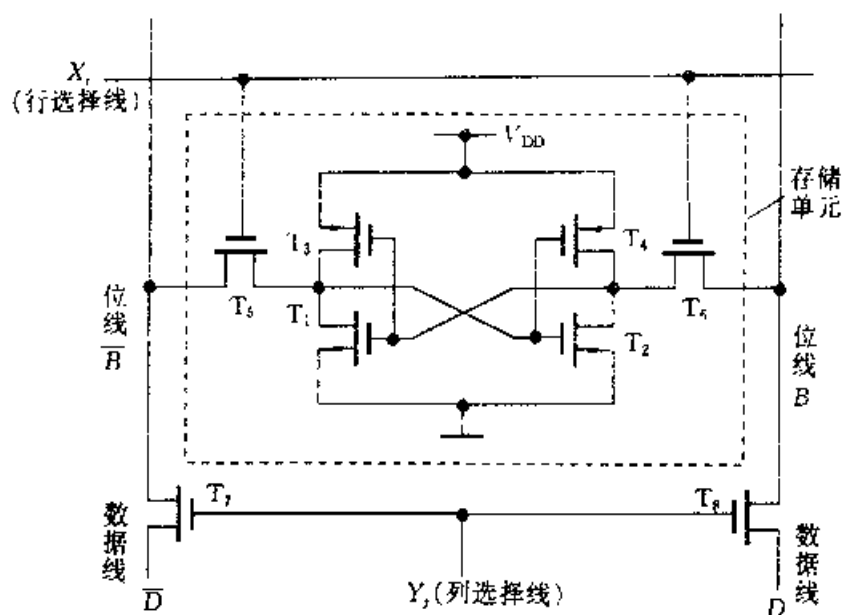
习 题

8.1.1 一个 CMOS 存储单元如图题所示,试分析其工作原理。

8.1.2 一个有 4096 位的 DRAM,如果存储矩阵为 64×64 结构形式,且每个存储单元刷新时间为 400 ns,则存储单元全部刷新一遍需要多长时间?

8.1.3 指出下列存储系统各具有多少个存储单元,至少需要几根地址线和数据线。

- (1) $64\text{K} \times 1$ (2) $256\text{K} \times 4$ (3) $1\text{M} \times 1$ (4) $128\text{K} \times 8$



图题 8.1.1

8.1.4 设存储器的起始地址为全 0, 试指出下列存储系统的最高地址为多少?

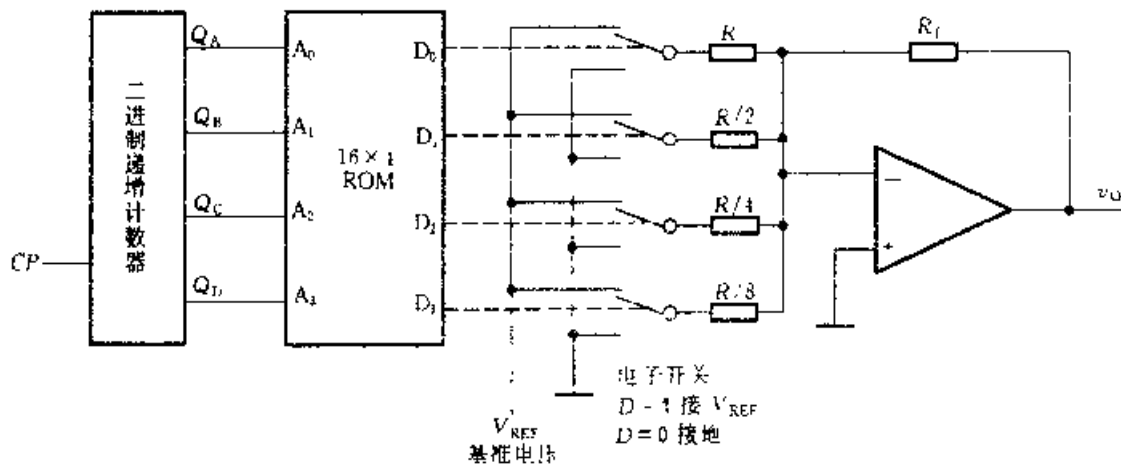
- (1) $2K \times 1$ (2) $16K \times 4$ (3) $256K \times 32$

8.1.5 试用 MCM6264 SRAM 芯片设计一个 $16K \times 16$ 位的存储器的系统, 画出其逻辑图。

8.1.6 一个有 $1M \times 1$ 位的 DRAM, 采用地址分时送入的方法, 芯片应具有几根地址线?

8.2.1 用一片 128×8 位的 ROM 实现各种码制之间的转换。要求用从全 0 地址开始的前 16 个地址单元实现 8421BCD 码到余 3 码的转换; 接下来的 16 个地址单元实现余 3 码到 8421BCD 码的转换。试求: (1) 列出 ROM 的地址与内容对应关系的真值表; (2) 确定输入变量和输出变量与 ROM 地址线和数据线的对应关系; (3) 简要说明将 8421BCD 码的 0101 转换成余 3 码和将余 3 码的 1001 转换成 8421BCD 码的过程

8.2.2 试确定用 ROM 实现下列逻辑函数时所需的容量: (1) 实现两个 3 位二进制数相乘的乘法器; (2) 将 8 位二进制数转换成十进制数 (用 BCD 码表示) 的转换电路



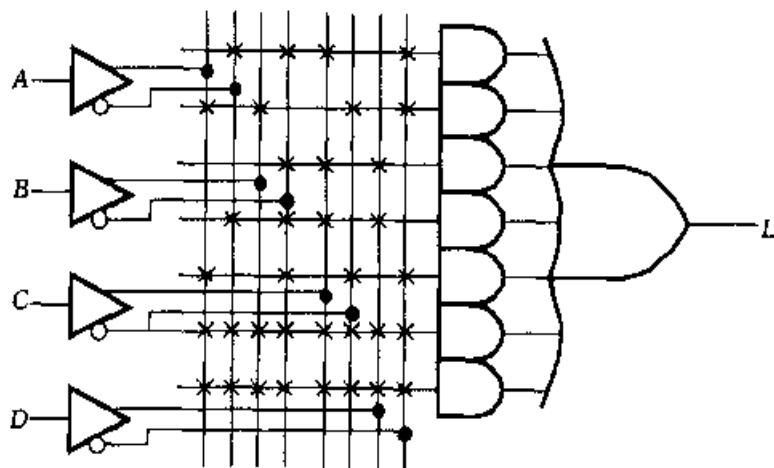
图题 8.2.3

8.2.3 利用 ROM 构成的任意波形发生器如图题 8.2.3 所示, 改变 ROM 的内容, 即可改变输出波形。当 ROM 的内容如表题 8.2.3 所示时, 画出输出端随 CP 变化的波形。

表题 8.2.3

A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	1
0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	0	1
1	0	0	0	0	1	0	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	1	0
1	0	1	1	0	0	0	1
1	1	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	0	1	1

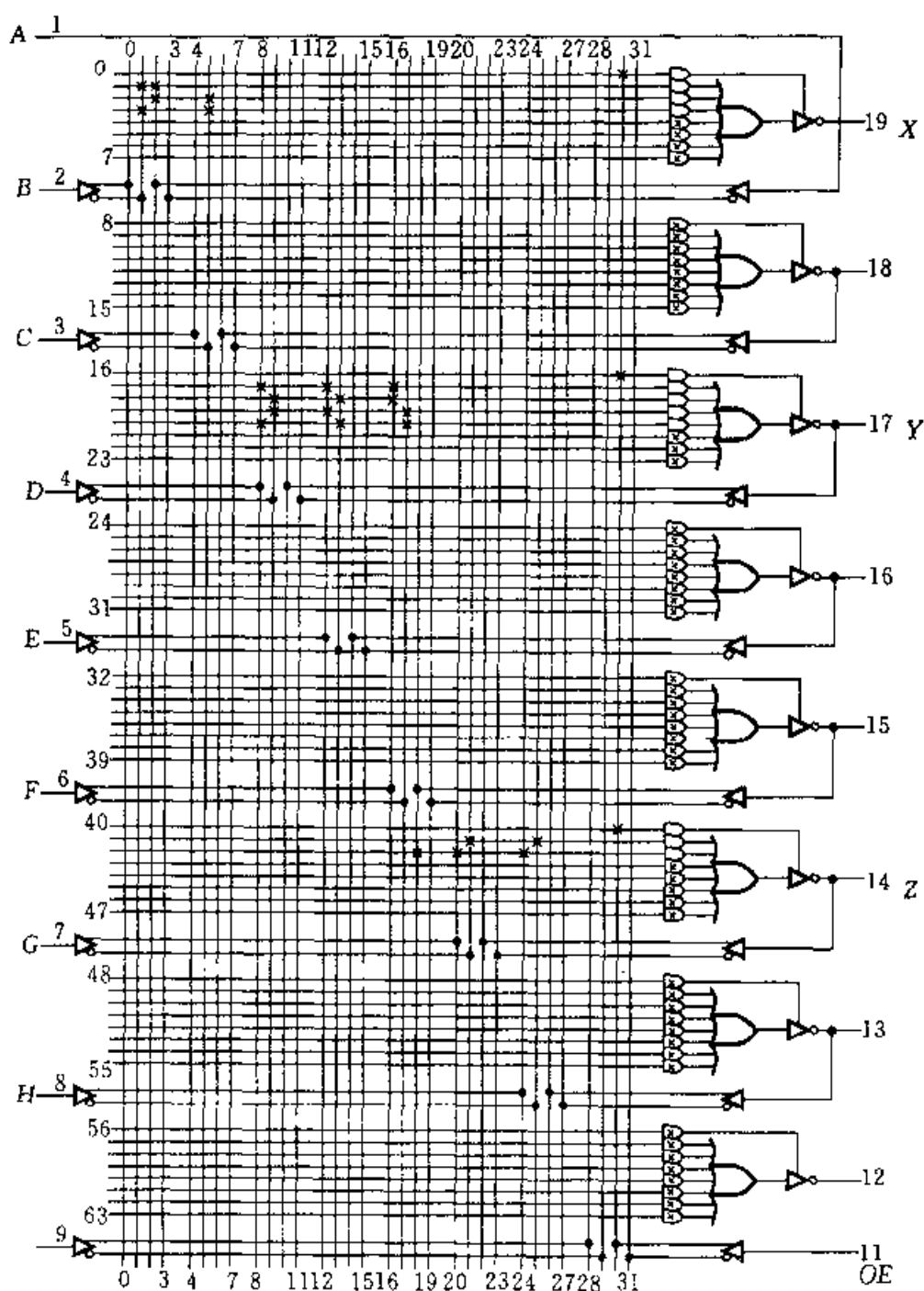
8.3.1 试分析图题 8.3.1 的逻辑电路, 写出输出逻辑函数表达式。



图题 8.3.1

8.3.2 PAL16L8 编程后的电路如图题 8.3.2 所示, 试写出 X、Y 和 Z 的逻辑函数表达式。

8.3.3 设输入逻辑变量为 A、B、C 和 D, 用图 8.3.7 所示的 PAL16L8 实现逻辑函数 L_1



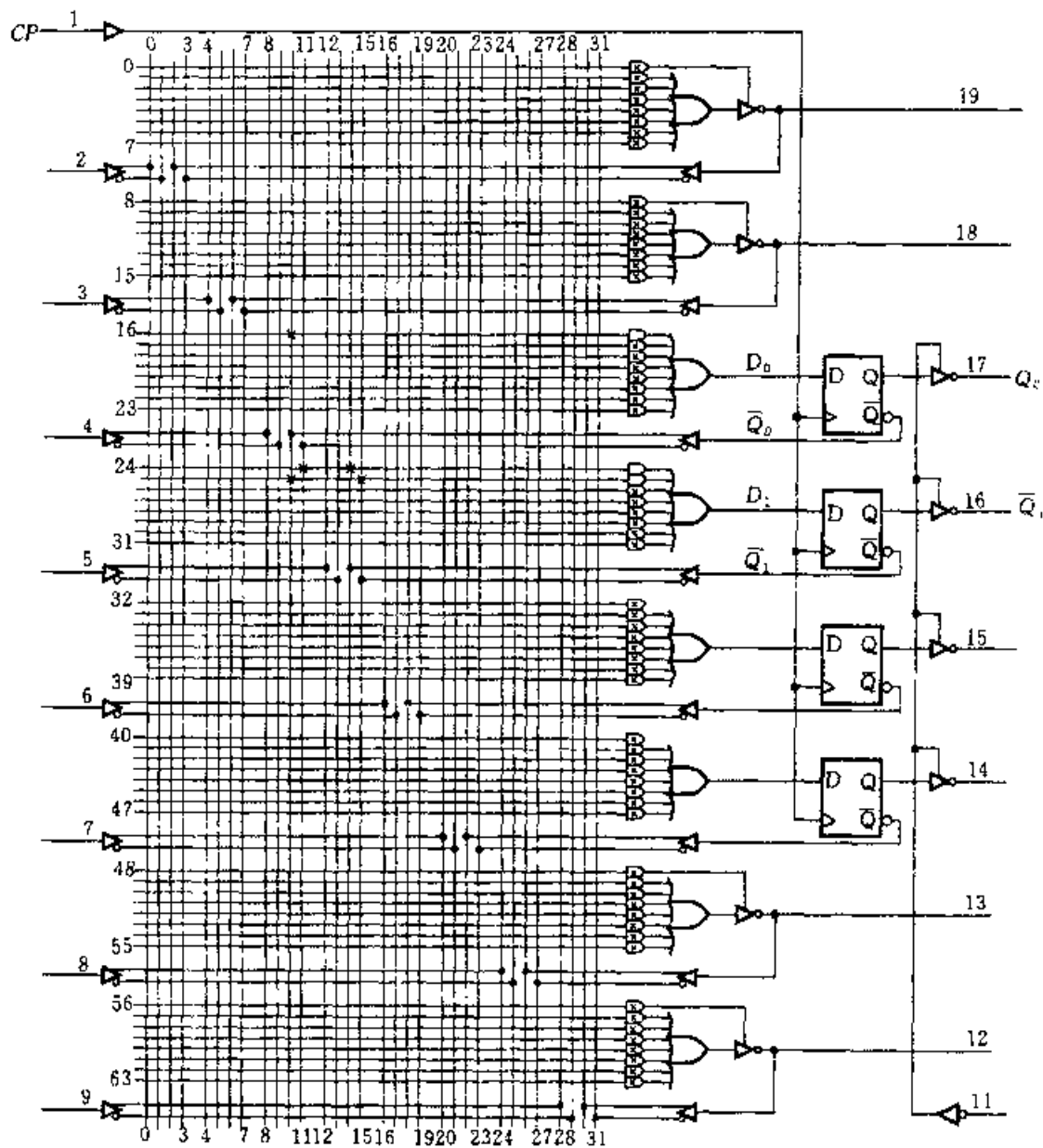
图题 8.3.2

$(A, B, C, D) = \sum m(0, 5, 10, 11)$, $L_2(A, B, C, D) = \sum m(4, 7, 11, 14)$, $L_3(A, B, C, D) = \sum m(1, 3, 5, 15)$ 。试画出编程后的电路图

8.3.4 试分析图题 8.3.4 所示电路,说明该电路的逻辑功能

8.3.5 对于图 8.3.9 所示的 OLMC,试画出当 $AC0 = 1$, $AC1(n) = 1$, $XOR(n) = 1$ 时的等效逻辑电路。

8.5.1 简要说明在 XC4000E 系列的 1 个 CLB 中,怎样同时输出一个 9 输入变量的组合



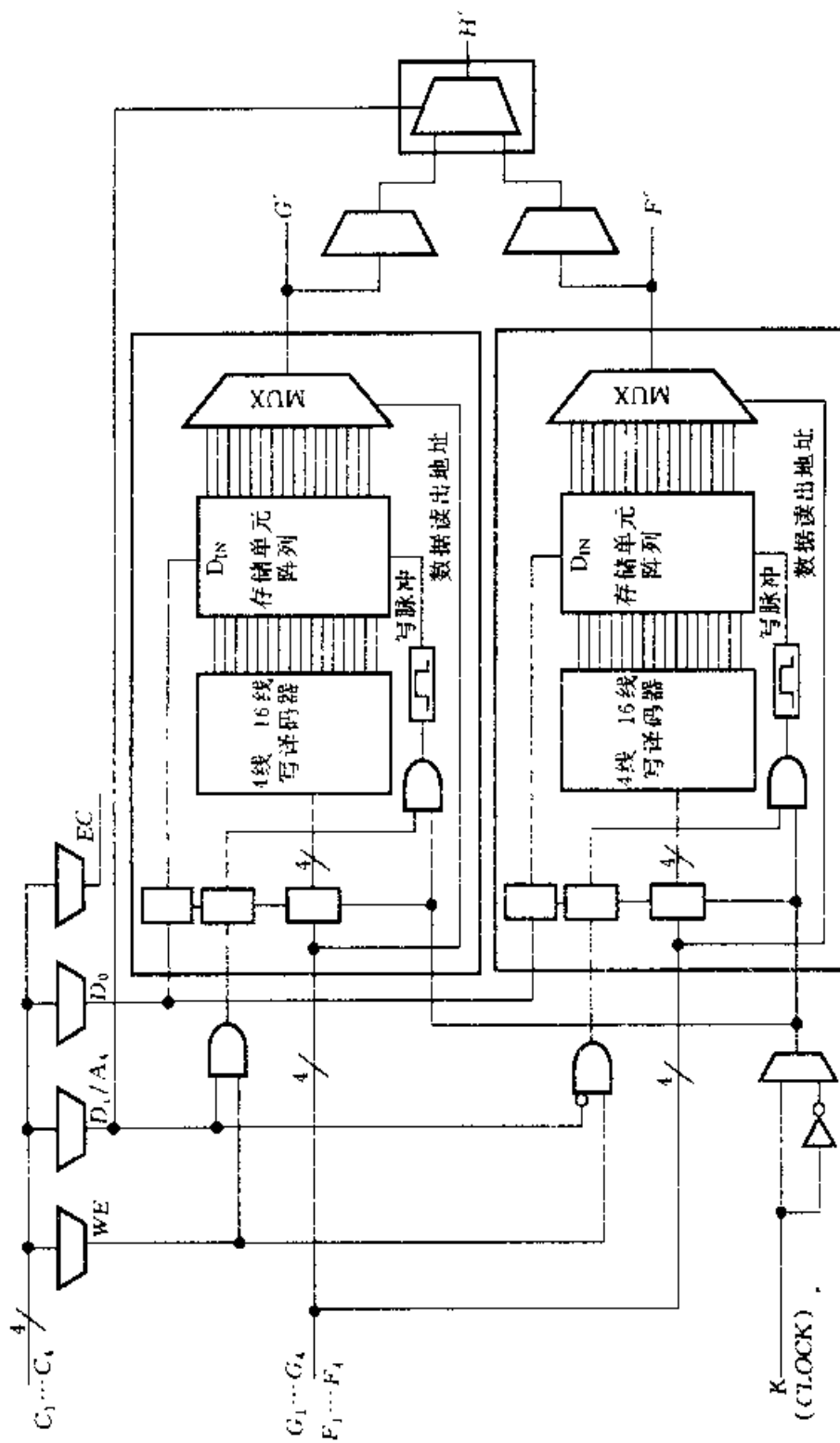
图题 8.3.4

逻辑函数和一个 4 输入变量的组合逻辑函数。

8.5.2 一个 CLB 构成的 32×1 位 RAM 电路如图题 8.5.2 所示, 试分析工作原理。

* CAD 习题

- C8.1 试用 ispLSI1016 设计一个 1 位十进制数(用 8421BCD 码表示)加法器。
- C8.2 试用 ispLSI1016 设计一个模可变递增计数器, 在 X、Y 的控制下, 分别实现模 5、模 8、模 10 和模 12 的递增计数。X、Y 与模的控制关系如表题 8.2 所示。
- C8.3 试用 ispLSI1016 设计一个 4 位共阴极显示器的动态扫描译码显示控制电路。
- C8.4 试用 ispLSI1032E 设计一个具有时、分、秒的电子钟电路, 按 24 小时制计时。要求: (1) 输出时、分、秒的 8421BCD 码, 计时输入脉冲频率为 1 kHz; (2) 具有分、时校正功能, 校



图题 8.5.2

表题 8.2

X	Y	功能
0	0	模 5 递增计数
0	1	模 8 递增计数
1	0	模 10 递增计数
1	1	模 12 递增计数

正输入脉冲频率为 5 Hz;(3)具有整点报时功能。每逢 59 分 51 秒、53 秒、55 秒、57 秒输出 500 Hz 的低音信号,在 59 分 59 秒时输出 1 kHz 的高音信号,输出信号的持续时间均为 1 s;(4)采用层次式的设计方法。顶层采用原理图,标明各模块功能及其输入/输出信号的连接关系;下层采用原理图或者 ABEL 语言进行设计。

C8.5 试用 ispLSI1032E 设计一个 4 位数字显示、量程可变的频率计。要求:(1)能够测量 1 kHz~9999kHz 正方波信号的频率;(2)手动按键切换量程;(3)当输入信号的频率大于实际量程时,有溢出显示;(4)测量值以 8421BCD 码形式输出;(5)采用层次式的设计方法。顶层采用原理图,标明各模块功能及其输入/输出信号的连接关系;下层采用原理图或 ABEL 语言进行设计。

9 脉冲波形的产生与变换

引言 在数字电路或系统中,常常需要各种脉冲波形,例如时钟脉冲、控制过程中的定时信号等。这些脉冲波形的获取,通常采用两种方法:一种是利用脉冲信号产生器直接产生;另一种则是对已有的信号进行变换,使之满足系统的要求。

本章主要讨论几种脉冲信号产生器以及脉冲变换的基本单元电路,如多谐振荡器、施密特触发器、单稳态触发器及定时器等。并对它们的功能、特点及其主要应用作简要的介绍。

9.1 多谐振荡器

多谐振荡器是一种自激振荡电路,该电路在接通电源后无需外接触发信号就能产生一定频率和幅值的矩形脉冲波或方波。由于多谐振荡器在工作过程中不存在稳定状态,故又称为无稳态电路。

9.1.1 门电路组成的多谐振荡器

1. 电路组成及工作原理

由门电路组成的多谐振荡器虽有多种电路形式,但它们无一例外地均具有如下共同的特点。首先,电路中含有开关器件,如门电路、电压比较器、BJT等。这些器件主要用作产生高、低电平;其次,具有反馈网络,将输出电压恰当地反馈给开关器件使之改变输出状态;另外,还要有延迟环节,利用RC电路的充、放电特性可实现延时,以获得所需要的振荡频率。在许多实用电路中,反馈网络兼有延时的作用。一种由CMOS门电路组成的多谐振荡器如图9.1.1所示。其原理图和工作波形图分别如图9.1.2a、b所示。图a中 D_1 、 D_2 、

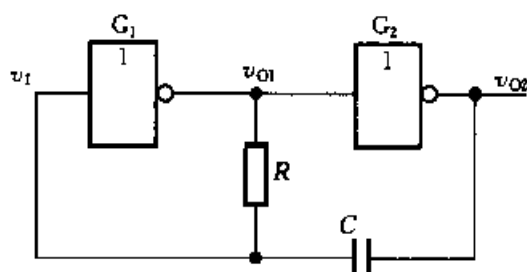


图 9.1.1 由 CMOS 门电路组成的多谐振荡器

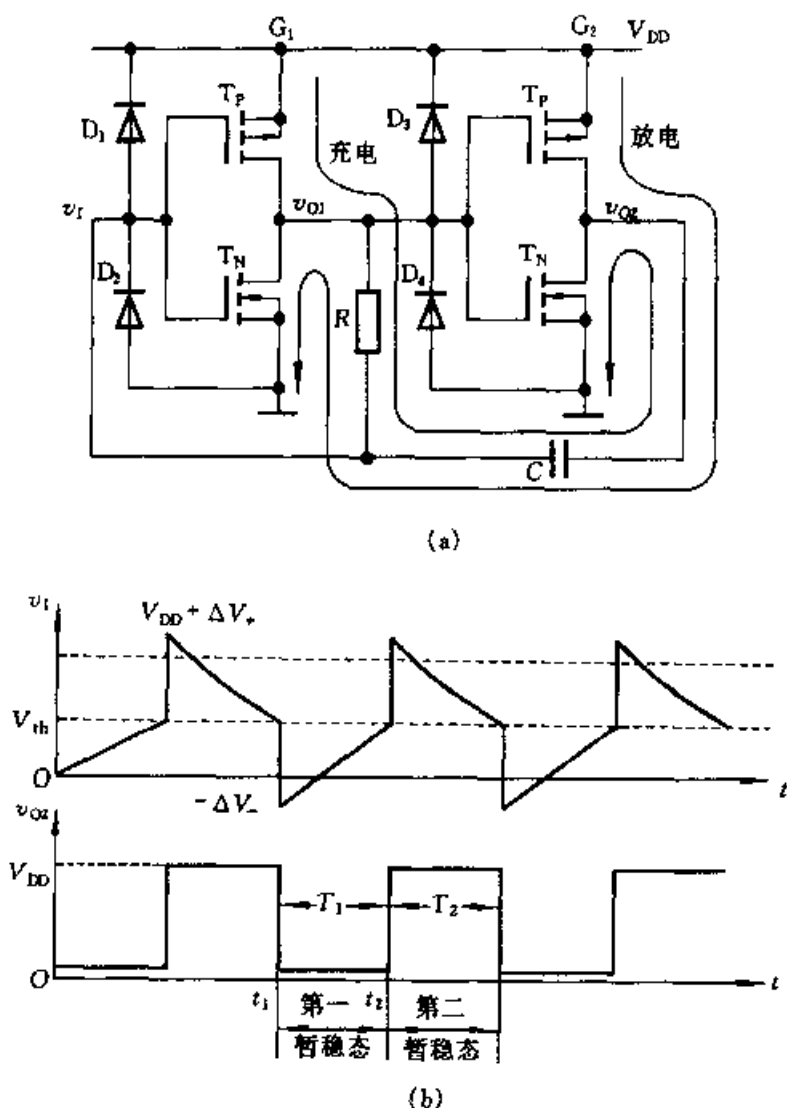


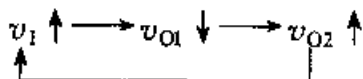
图 9.1.2 多谐振荡器原理图和波形图
(a) 多谐振荡器原理图 (b) 多谐振荡器波形图

D_3 、 D_4 均为保护二极管。

为了讨论方便,在电路分析中,假定门电路的电压传输特性曲线为理想化的折线,即开门电平(V_{ON})和关门电平(V_{OFF})相等,这个理想化的开门电平或关门电平称为门坎电平(或阈值电平),记为 V_{th} 且设 $V_{th} = V_{DD}/2$ 。

(1) 第一暂稳态及电路自动翻转的过程

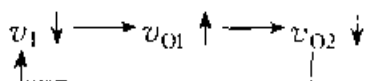
假定在 $t=0$ 时接通电源,电容 C 尚未充电,电路初始状态为 $v_{O1} = V_{OH}$, $v_1 = v_{O2} = V_{OL}$ 状态,即第一暂稳态。此时,电源 V_{DD} 经 G_1 的 T_P 管、 R 和 G_2 的 T_N 管给电容 C 充电,如图 9.1.2a 所示。随着充电时间的增加, v_1 的值不断上升,当 v_1 达到 V_{th} 时,电路发生下述正反馈过程:



这一正反馈过程瞬间完成,使 G_1 导通, G_2 截止,电路进入第二暂稳态,即 $v_{O1} = V_{OL}$, $v_{O2} = V_{OH}$ 。

(2) 第二暂稳态及电路自动翻转的过程

电路进入第二暂稳态瞬间, v_{O2} 由 0 V 上跳至 V_{DD} , 由于电容两端电压不能突变, 则 v_1 也将上跳 V_{DD} , 本应升至 $V_{DD} + V_{th}$, 但由于保护二极管的钳位作用, v_1 仅上跳至 $V_{DD} + \Delta V_{+c}$ 。随后, 电容 C 通过 G_2 的 T_P 、电阻 R 和 G_1 的 T 、放电, 使 v_1 下降, 当 v_1 降至 V_{th} 后, 电路又产生如下正反馈过程:



从而使 G_1 迅速截止, G_2 迅速导通, 电路又回到第一暂稳态, $v_{O1} = V_{OH}$, $v_{O2} = V_{OL}$ 。此后, 电路重复上述过程, 周而复始地从一个暂稳态翻转到另一个暂稳态, 在 G_2 的输出端得到方波。

由上述分析不难看出, 多谐振荡器的两个暂稳态的转换过程是通过电容 C 充、放电作用来实现的, 电容的充、放电作用又集中体现在图中 v_1 的变化上。因此, 在分析中要着重注意 v_1 的波形。

2. 振荡周期的计算

在振荡过程中, 电路状态的转换主要取决于电容的充、放电, 而转换时刻则决定于 v_1 的数值。根据以上分析所得电路在状态转换时 v_1 的几个特征值, 可以计算出图 9.1.2b 中的 T_1 、 T_2 的值。

(1) T_1 的计算

对应于第一暂稳态, 将图 9.1.2b 中 t_1 作为时间起点, $T_1 = t_2 - t_1$, $v_1(0^+) = -\Delta V_- \approx 0$ V, $v_1(\infty) = V_{DD}$, $\tau = RC$ 。根据 RC 电路瞬态响应的分析, 有

$$T_1 = RC \ln \frac{V_{DD}}{V_{DD} - V_{th}} \quad (9.1.1)$$

(2) T_2 的计算

对应于图 9.1.2b, 在第二暂稳态, 将 t_2 作为时间起点, 则有

$$v_1(0^+) = V_{DD} + \Delta V_+ \approx V_{DD}, v_1(\infty) = 0, \tau = RC$$

由此可求出

$$T_2 = RC \ln \frac{V_{DD}}{V_{th}} \quad (9.1.2)$$

所以

$$T = T_1 + T_2 = RC \ln \left[\frac{V_{DD}^2}{(V_{DD} - V_{th}) \cdot V_{th}} \right] \quad (9.1.3)$$

将 $V_{th} = V_{DD}/2$ 代入, 上式变为

$$T = RC \ln 4 \approx 1.4RC \quad (9.1.4)$$

图9.1.1是一种最简型多谐振荡器, 式(9.1.4)仅适于 $R \gg R_{ON(P)} + R_{ON(N)}$ [$R_{ON(P)}$ 、 $R_{ON(N)}$ 分别为 CMOS 门中 NMOS、PMOS 管的导通电阻]、 C 远大于电路分布电容的情况。当电源电压波动时, 会使振荡频率不稳定, 在 $V_{th} \neq V_{DD}/2$ 时, 影响尤为严重。一般在图9.1.1中增加一个补偿电阻 R_s , 如图9.1.3所示。 R_s 可减小电源电压变化对振荡频率的影响。当 $V_{th} = V_{DD}/2$ 时, 取 $R_s \gg R$ (一般取 $R_s = 10R$)。

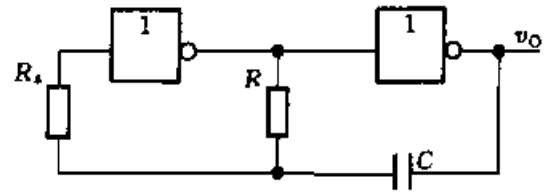


图 9.1.3 加补偿电阻的 CMOS 多谐振荡器

9.1.2 石英晶体振荡器

前面介绍的多谐振荡器的振荡周期或重复频率不仅与时间常数 RC 有关, 而且还取决于门电路的阈值电压 V_{th} 。由于 V_{th} 容易受温度、电源电压及干扰的影响, 因此频率稳定性较差, 不能适应在对频率稳定性要求较高的场合。

为得到频率稳定性很高的脉冲波形, 多采用由石英晶体组成的石英晶体振荡器。石英晶体的电路符号和阻抗频率响应在本书模拟部分第 9 章已作过介绍, 现重绘于图 9.1.4 中。由阻抗频率响应可知, 石英晶体的选频特性非常好, 它有一个极为稳定的串联谐振频率 f_s , 且等效品质因数 Q 值很高。只有频率为 f_s 的信号最容易通过, 而其他频率的信号均会被晶体所衰减。

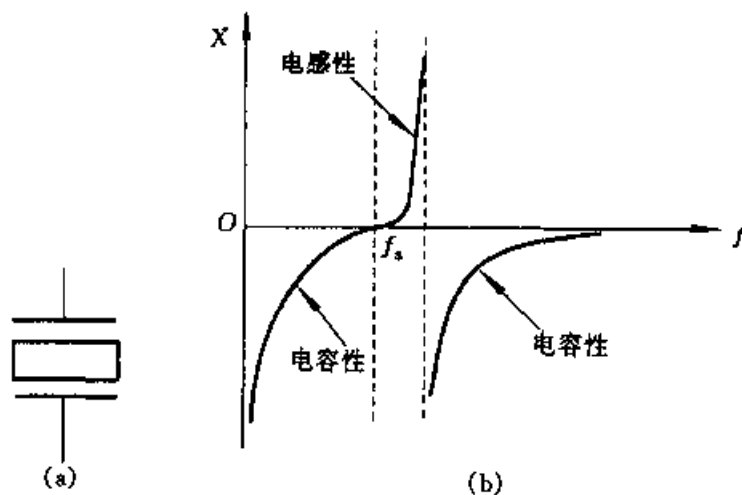


图 9.1.4 石英晶体的电路符号及阻抗频率特性

(a) 电路符号 (b) 阻抗频率特性

石英晶体振荡器电路如图 9.1.5 所示。图中, 并联在两个反相器输入、输出

间的电阻 R 的作用是使反相器工作在线性放大区。 R 的阻值,对于 TTL 门电路通常在 $0.7 \sim 2 \text{ k}\Omega$ 之间;对于 CMOS 门则常在 $10 \sim 100 \text{ M}\Omega$ 之间。电路中,电容 C_1 用于两个反相器间的耦合,而 C_2 的作用,则是抑制高次谐波,以保证稳定的频率输出。电容 C_2 的选择应使 $2\pi RC_2 f_s \approx 1$,从而使 RC_2 并联网络在 f_s 处产生极点,以减少谐振信号损失。 C_1 的选择应使 C_1 在频率为 f_s 时的容抗可以忽略不计。

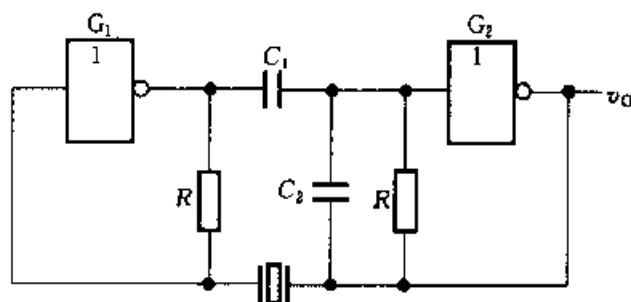
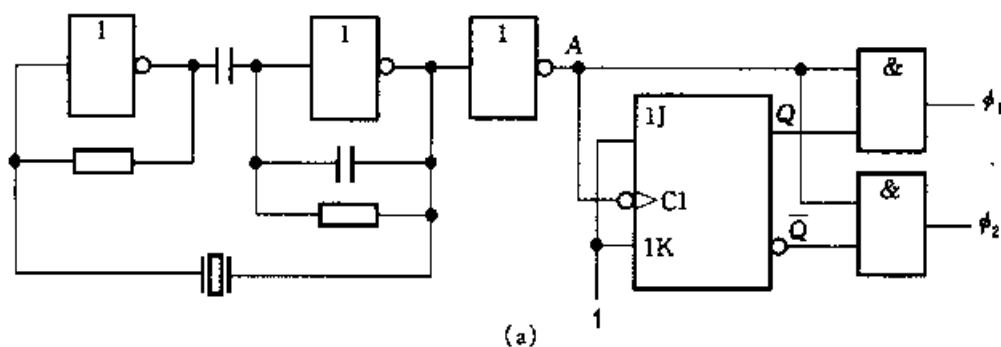
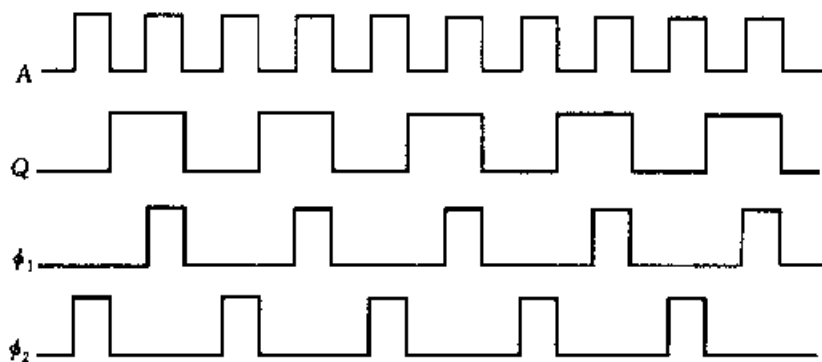


图 9.1.5 石英晶体振荡器



(a)



(b)

图 9.1.6 双相时钟发生器

(a) 逻辑图 (b) 波形图

图 9.1.5 所示电路的振荡频率仅取决于石英晶体的串联谐振频率 f_s ,而与

电路中的 R 、 C 的数值无关。这是因为电路对 f_s 频率所形成正反馈最强而易于维持振荡。

为了改善输出波形,增强带负载的能力,通常在振荡器的输出端再加一级反相器。作为一个应用实例,两相时钟产生电路如图 9.1.6a 所示,其波形如图 b 所示。

9.2 单稳态触发器

单稳态触发器与前面介绍的触发器不同,它具有下述特点:

- ① 电路有一个稳态、一个暂稳态。
- ② 在外来触发信号作用下,电路由稳态翻转到暂稳态。
- ③ 暂稳态是一个不能长久保持的状态,由于电路中 RC 延时环节的作用,经过一段时间后,电路会自动返回到稳态。暂稳态的持续时间取决于 RC 电路的参数值。

单稳态触发器的这些特点被广泛地应用于脉冲波形的变换与延时中。

9.2.1 门电路组成的微分型单稳态触发器

1. 电路组成及工作原理

微分型单稳态触发器可由与非门或或非门电路构成,图 9.2.1a、b 分别为由与非门和或非门构成的单稳态触发器。与基本 RS 触发器不同,构成单稳态触发器的两个逻辑门是由 RC 耦合的,由于 RC 电路为微分电路的形式,故称为微分型单稳态触发器。

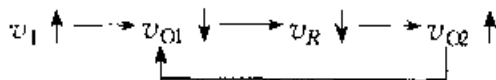
下面以 CMOS 或非门构成的单稳态触发器为例,来说明它的工作原理。

(1) 没有触发信号时,电路处于一种稳态

没有触发信号时, v_1 为低电平。由于门 G_2 的输入端经电阻 R 接至 V_{DD} ,因此 v_{O2} 为低电平; G_1 的两个输入均为 0,故输出 v_{O1} 为高电平,电容两端的电压接近 0 V,这是电路的“稳态”。在触发信号到来之前,电路一直处于这个状态: $v_{O1} = V_{OH}$, $v_{O2} = V_{OL}$ 。

(2) 外加触发信号,电路由稳态翻转到暂稳态

当 v_1 正跳变时, G_1 的输出 v_{O1} 由高变低,经电容 C 耦合,使 v_R 为低电平,于是 G_2 的输出 v_{O2} 变为高电平。 v_{O2} 的高电平接至 G_1 门的输入端,从而在此瞬间导致如下正反馈过程:



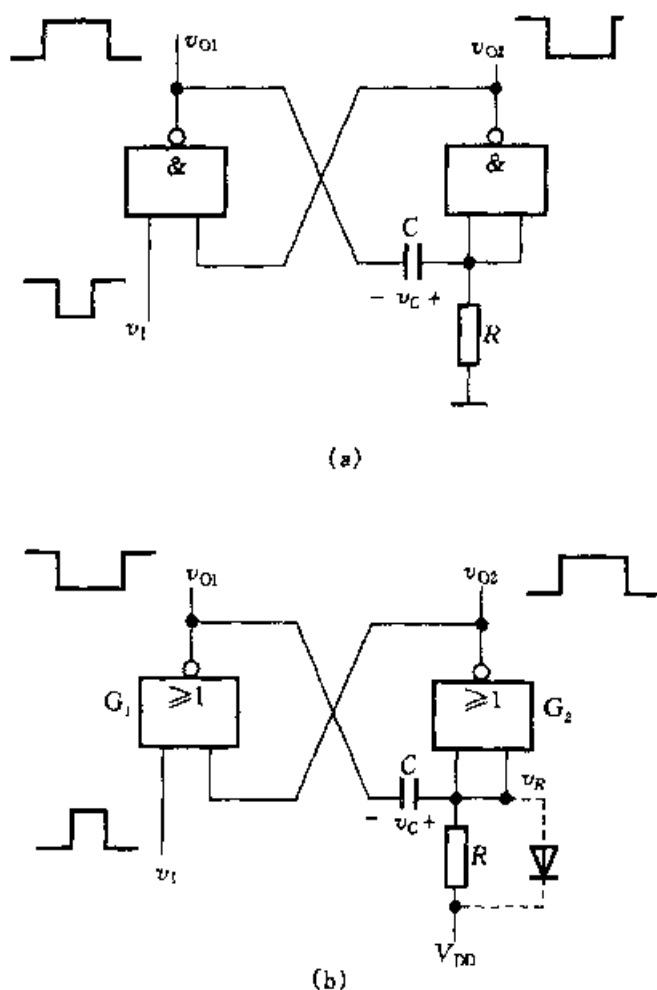


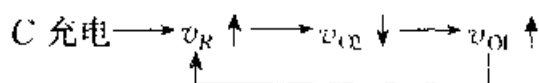
图 9.2.1 微分型单稳态触发器

(a) 由与非门构成的微分型单稳态触发器 (b) 由或非门构成的微分型单稳态触发器

这样 G_1 导通, G_2 截止在瞬间完成。此时, 即使触发信号 v_i 撤除 (v_i 变为低电平), 由于 v_{O2} 的作用, v_{O1} 仍维持低电平。然而, 电路的这种状态是不能长久保持的, 故称之为暂稳态。暂稳态时, $v_{O1} = V_{OL}$, $v_{O2} = V_{OH}$ 。

(3) 电容充电, 电路由暂稳态自动返回至稳态

在暂稳态期间, 电源经电阻 R 和门 G_1 的导通工作管对电容 C 充电, 随着充电时间的增加, v_C 增加, 使 v_R 升高, 当 v_R 达到阈值电压 V_{th} 时, 电路发生下述正反馈过程 (设此时触发器脉冲已消失):



于是 G_1 门迅速截止, G_2 门很快导通, 最后使电路由暂稳态返回至稳态, $v_{O1} = V_{OH}$, $v_{O2} = V_{OL}$ 。

暂稳态结束后, 电容将通过电阻 R 放电, 使 C 上的电压恢复到稳定状态时

的初始值。在整个过程中,电路各点工作波形如图 9.2.2 所示。

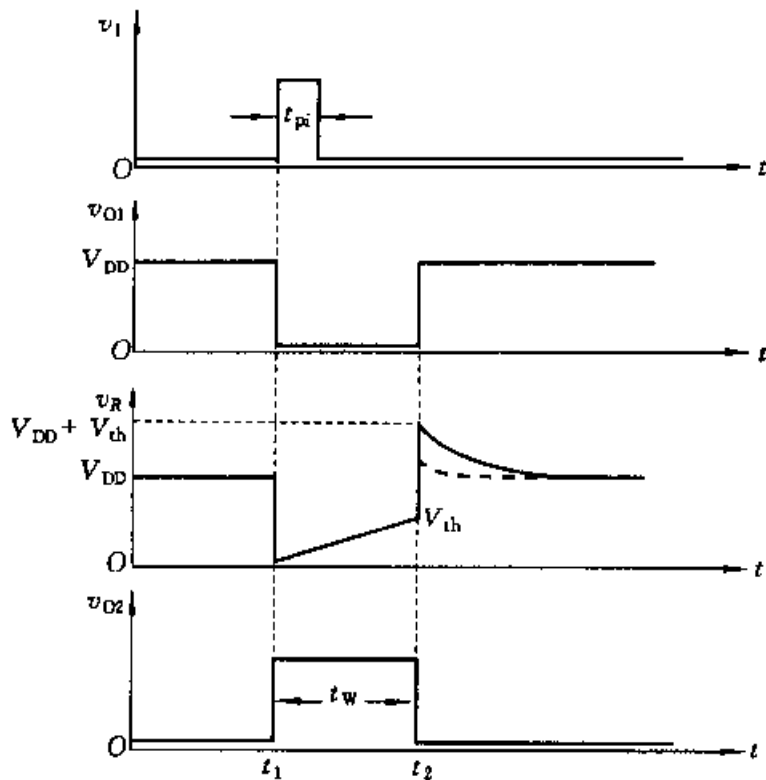


图 9.2.2 微分型单稳态触发器各点工作波形

2. 主要参数的计算

(1) 输出脉冲宽度 t_w

输出脉冲宽度 t_w ,也就是暂稳态的维持时间,可以根据 v_R 的波形进行计算。为了计算方便,对于图 9.2.2 的 v_R 的波形,将触发脉冲作用的起始时刻 t_1 作为时间起点,于是有

$$\begin{aligned} v_R(0^+) &= 0 \\ v_R(\infty) &= V_{DD} \\ \tau &= RC \end{aligned}$$

根据 RC 电路瞬态过程的分析,可得到

$$v_R(t) = V_R(\infty) + [V_R(0^+) - V_R(\infty)]e^{-t/\tau} \quad (9.2.1)$$

当 $t = t_w$ 时, $v_R(t_w) = V_{th}$,代入上式可求得

$$v_R(t_w) = V_{th} = V_{DD}(1 - e^{-t_w/RC})$$

$$t_w = RC \ln \frac{V_{DD}}{V_{DD} - V_{th}} \quad (9.2.2)$$

当 $V_{th} = V_{DD}/2$,则

$$t_w \approx 0.7RC \quad (9.2.3)$$

(2) 恢复时间 t_{re}

暂稳态结束后,还需要一段恢复时间,以便电容 C 在暂稳态期间所充的电荷释放完,使电路恢复到初始状态。一般要经过 $3\tau_d$ (τ_d 为放电时间常数)的时间,放电才基本结束,故 t_{re} 约为 $3\tau_d$ 。

(3) 最高工作频率 f_{max}

设触发信号 v_1 的时间间隔为 T ,为了使单稳电路能正常工作,应满足 $T > t_w + t_{re}$ 的条件,即最小时间间隔 $T_{min} = t_w + t_{re}$ 。因此,单稳态触发器的最高工作频率为

$$f_{max} = \frac{1}{T_{min}} < \frac{1}{t_w + t_{re}} \quad (9.2.4)$$

显然,上述关系式是在作了某些近似之后得到的(例如,忽略了导通管的漏源电阻等),因而只能作为选择参数的初步依据,准确的参数还要通过实验调整得到。

3. 讨论

(1) 如图 9.2.2 所示,在暂稳态结束($t = t_2$)瞬间,门 G_2 的输入电压 v_R 达到 $V_{DD} + V_{th}$,这么高的输入电压可能损坏 CMOS 门。为了避免这种现象发生,在 CMOS 器件内部设有保护二极管 D ,如图 9.2.1b 中的虚线所示。在电容 C 充电期间,二极管 D 开路。而当 $t = t_2$ 时,二极管 D 导通,于是 v_R 被钳制在 $V_{DD} + 0.6V$ 的电位上(见图 9.2.2 中的虚线)。同时,在恢复期间,电容 C 放电的时间常数 $\tau_d = (R \parallel R_f)C$ (R_f 为二极管 D 的正向电阻),由于 $R_f \ll R$,因此电容放电的时间很短。

(2) 当输入 v_1 的脉冲宽度 $t_{pi} > t_w$ 时,则在 v_{O2} 变为低电平后, G_1 没有响应,不能形成前述的正反馈过程,使 v_{O2} 的输出边沿变缓。因此,当输入脉冲宽度 t_{pi} 很宽时,可在单稳态触发器的输入端加入 R_d 、 C_d 组成的微分电路。同时为了改善输出波形,可在图 9.2.1 中 G_2 的输出端再加一级反相器 G_3 ,如图 9.2.3 所示。

(3) 若采用 TTL 与非门构成如图 9.2.1a 所示的单稳电路时,由于 TTL 门存在输入电流,因此,为了保证稳态时 G_2 的输入为低电平,电阻 R 要小于 $0.7k\Omega$ 。如果输入端采用 R_d 、 C_d 微分电路时, R_d 的数值应大于 $2k\Omega$,使得稳态时 v_D 大于门 G_1 的开门电平(V_{ON}),而 CMOS 门由于不存在输入电流,故不受此限制。

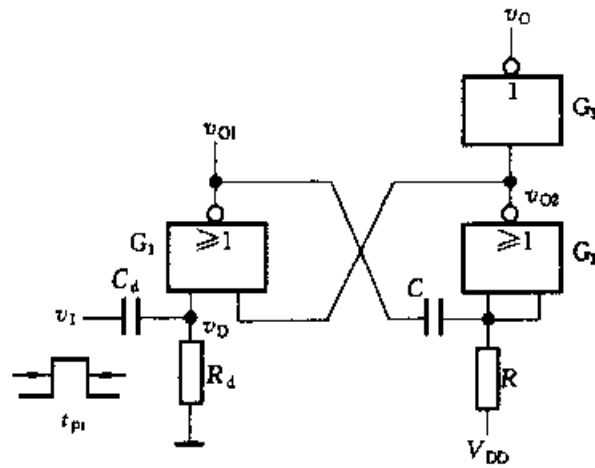


图 9.2.3 宽脉冲触发的单稳电路

9.2.2 集成单稳态触发器

用门电路组成的单稳态触发器虽然电路简单,但输出脉宽的稳定性差,调节范围小,且触发方式单一。为适应数字系统中的广泛应用,现已生产出单片集成单稳态触发器。集成单稳态触发器根据电路及工作状态不同分为可重复触发和不可重复触发两种。

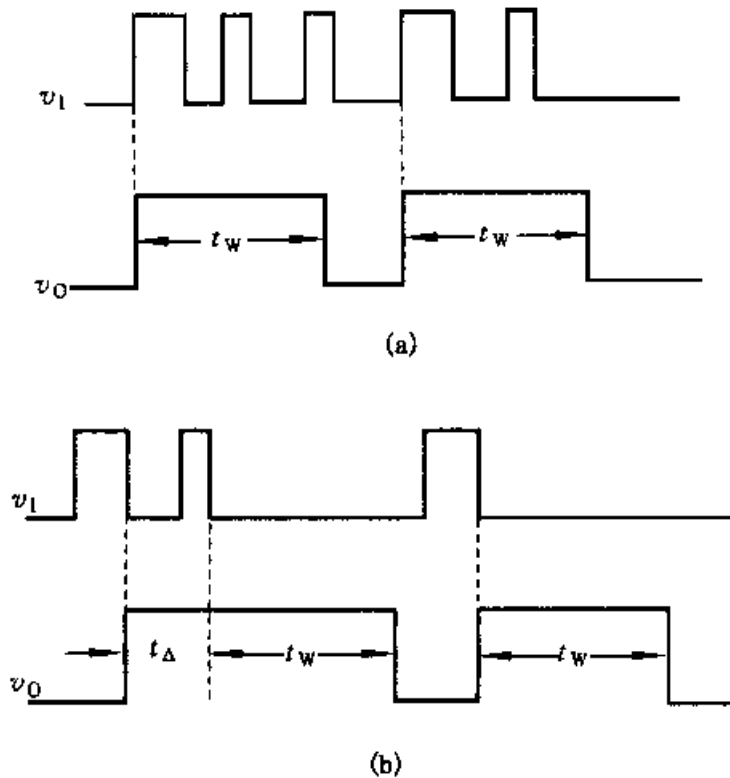


图 9.2.4 两种单稳电路工作波形

(a) 不可重复触发单稳态触发器工作波形 (b) 可重复触发单稳态触发器工作波形

两种不同触发特性的单稳态触发器的主要区别是：不可重复触发单稳态触发器，在进入暂稳态期间，如有触发脉冲作用，电路的工作过程不受其影响，只有当电路的暂稳态结束后，输入触发脉冲才会影响电路状态。电路输出脉宽由 R 、 C 参数确定。

而可重复触发单稳态触发器在暂稳态期间，如有触发脉冲作用，电路会重新被触发，使暂稳态继续延迟一个 t_s 时间，直至触发脉冲的间隔超过单稳输出脉宽，电路才返回稳态。

两种单稳态触发器的工作波形分别如图 9.2.4a、b 所示。

1. 不可重复触发的集成单稳态触发器

TTL 集成器件 74121 是一种不可重复触发集成单稳态触发器，其逻辑图和引脚图分别如图 9.2.5a、b 所示。

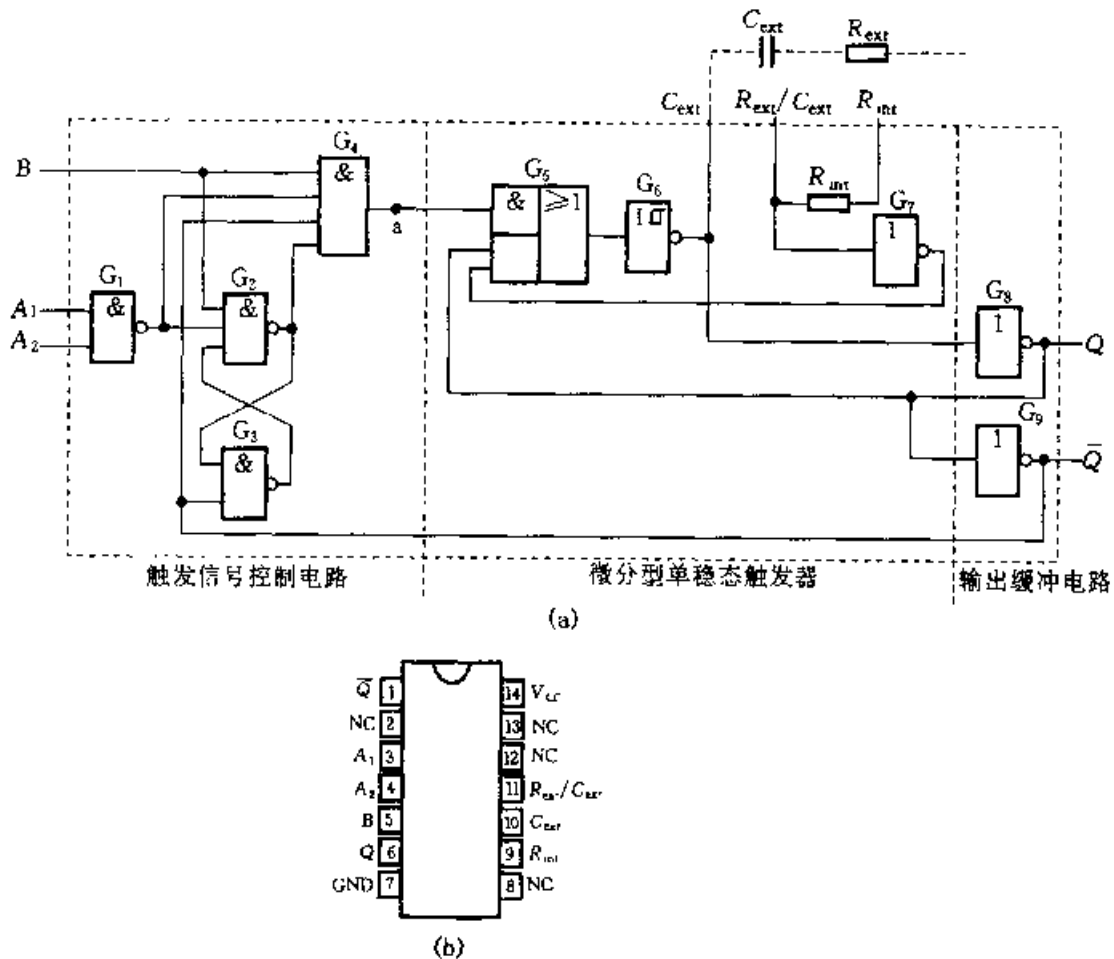


图 9.2.5 TTL 集成器件 74121

(a) 逻辑图 (b) 引脚图

(1) 电路组成及工作原理

电路由触发信号控制电路、微分型单稳态触发器及输出缓冲电路组成。

将具有迟滞特性^①的非门 G_6 与 G_5 门合起来看成是一个或非门,它与 G_7 门及外接电阻 R_{ext} (或 R_{int})、电容 C_{ext} 即组成微分型单稳态触发器,其电路工作原理与 9.2.1 节介绍的微分型单稳态触发器基本相同。电路只有一个稳态 $Q=0, \bar{Q}=1$ 。当图中 a 点有正脉冲触发时,电路进入暂稳态 $Q=1, \bar{Q}=0$ 。 \bar{Q} 为低电平后使触发信号控制电路中 RS 触发器的 G_2 门输出低电平,将 G_4 门封锁,这样即使有触发信号输入,在 a 点也不会产生微分型单稳态触发器的触发信号,只有等电路返回稳态后,电路才会在输入触发信号作用下被再次触发,根据上述分析,电路属于不可重复触发单稳态触发器。

(2) 触发与定时











① 触发方式

74121 集成单稳态触发器有 3 个触发输入端,由触发信号控制电路分析可知,在下述情况下,电路可由稳态翻转到暂稳态:

- 若 A_1 、 A_2 两个输入中有一个或两个为低电平, B 发生由 0 到 1 的正跳变。
- 若 B 和 A_1 、 A_2 中的一个为高电平,输入中有一个或两个产生由 1 到 0 的负跳变。

74121 的功能如表 9.2.1 所示。

表 9.2.1 74121 功能表

输 入			输 出	
A_1	A_2	B	Q	\bar{Q}
L	×	H	L	H
×	L	H	L	H
×	×	L	L	H
H	H	×	L	H
H	↓	H		
↓	H	H		
↓	↓	H		
L	×	↑		
×	L	↑		

② 定时

单稳电路的定时取决于定时电阻和定时电容的数值。74121 的定时电容连接在芯片的 10、11 引脚之间。若输出脉冲宽度较宽,而采用电解电容时,电容 C

① 见本书模拟部分 P417~419 和 9.3 节。

的正极接在 C_{ext} 输入端(10 脚)。对于定时电阻,使用者可以有两种选择:

- 利用内部定时电阻(2 kΩ),此时将 9 号引脚(R_{int})接至电源 V_{CC} (14 脚)。
- 采用外接定时电阻(阻值在 1.4~40 kΩ 之间),此时 9 脚应悬空,电阻接在 11、14 脚之间。

74121 的输出脉冲宽度

$$t_w \approx 0.7RC \tag{9.2.5}$$

通常 R 的数值取在 2~30 kΩ 之间, C 的数值取在 10 pF~10 μF 之间,得到的 t_w 的取值范围可达到 20 ns~200 ms。

式(9.2.5)中的 R 可以是外接电阻 R_{ext} ,也可以是芯片内部电阻 R_{int} (约 2 kΩ),如希望得到较宽的输出脉冲,一般使用外接电阻。

2. 可重复触发集成单稳态触发器

以常用 CMOS 集成器件 MC14528 为例,介绍可重复触发单稳态触发器工

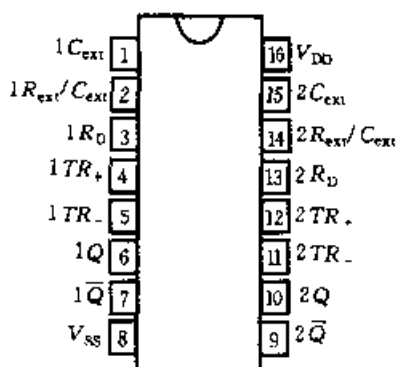
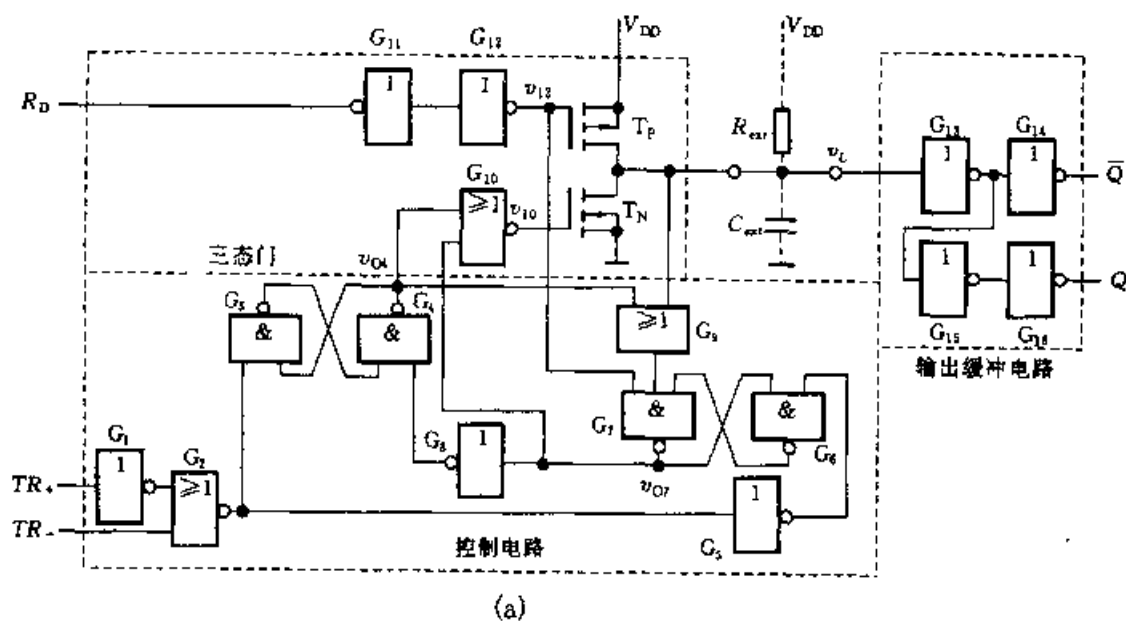


图 9.2.6 集成单稳态触发器 MC14528 逻辑图

(a) 逻辑图 (b) 引脚图

作原理。

该器件的逻辑图和引脚图分别如图 9.2.6a、b 所示。下面分析电路的工作原理。

由图 9.2.6 可见,电路主要由三态门、积分电路、控制电路组成的积分型单稳态触发器及输出缓冲电路组成。





(1) 稳态

令 $R_D = \times$, 无触发信号时 $TR_+ = 1, TR_- = \times$, 设接通电源后电容还未充电, $v_C = 0 \text{ V}$, 此时 G_4 输出 v_{O4} 一定为高电平。若 v_{O4} 为低电平, 则在 $v_C = V_{OL}$ 的共同作用下, G_9 输出低电平并使 G_7 输出高电平, G_8 输出低电平, 于是 v_{O4} 被置为高电平, 这样图中 $v_{10} = V_{OH}, v_{12} = V_{OH}, T_N, T_P$ 同时截止, V_{DD} 经 R_{ext} 向 C_{ext} 充电, 当 v_C 大于 V_{th3} 时, $Q = 0, \bar{Q} = 1$, 电路处于稳态。同样, 当 $R_D = \times$, 输入信号 $TR_+ = \times, TR_- = 0$ 时, G_5 门输出低电平, 使 G_6, G_7 门组成的基本 RS 触发器的 v_{O7} 为低电平, 经 G_8 反相后使 v_{O4} 处于高电平。电路维持稳态不变。

(2) 触发与定时

以 $R_D = 1, TR_+ = 1, TR_-$ 端加正触发脉冲情况为例说明电路触发工作情况。在 TR_- 端上升沿到来时 $v_{O4} = V_{OL}$, 由于 $v_{O7} = V_{OL}$, 于是 G_{10} 输出 $v_{10} = V_{OH}, T_N$ 导通, C_{ext} 开始放电, 当 v_C 下降至 G_{13} 门的阈值电压 V_{th3} 时, 电路进入暂态 $Q = 1, \bar{Q} = 0$ 。此暂态不能持续下去, 当 v_C 进一步下降至 G_9 门的阈值电压 V_{th9} 时, G_9 门输出低电平, 经 G_8 使 G_3, G_4 组成的基本 RS 触发器的 G_4 输出高电平, G_{10} 输出为低电平, 这样 T_N 又截止, C 又重新开始充电, 当 v_C 值大于 G_{13} 门的阈值电压 V_{th3} 时, 电路自动返回到稳态, 即 $Q = 0, \bar{Q} = 1$ 的状态, C_{ext} 继续充电至 V_{DD} 以后电路又恢复为稳态。MC14528 功能表如表 9.2.2 所示, 同理, 读者可自行分析表中其他状态。电路工作波形如图 9.2.7 所示。

表 9.2.2 MC14528 功能表

输 入			输 出		功能
R_D	TR_+	TR_-	Q	\bar{Q}	
L	\times	\times	L	H	清除
\times	H	\times	L	H	禁止
\times	\times	L	L	H	禁止
H	H	\uparrow			单稳
H	\downarrow	L			单稳

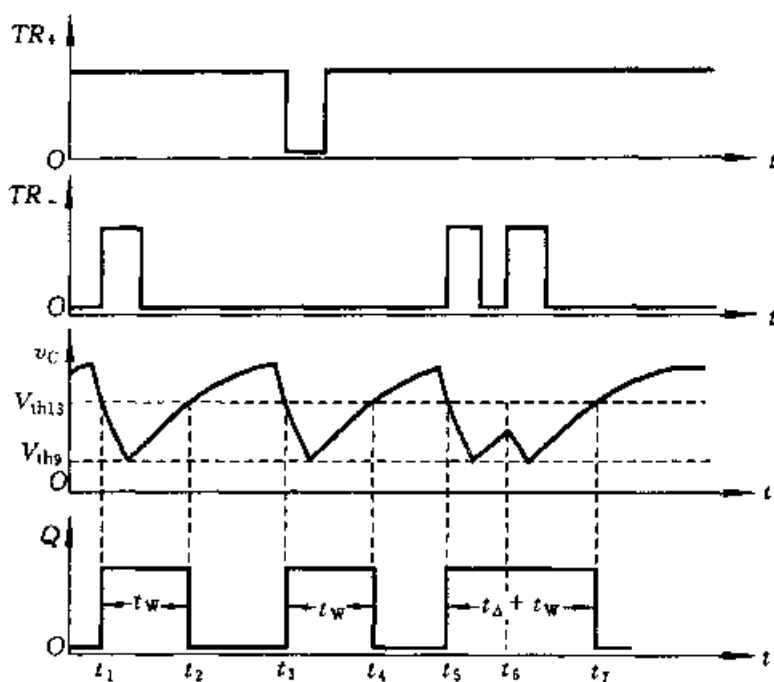


图 9.2.7 MCI4528 可重复触发单稳态工作波形

由图 9.2.7 可见,输出脉宽 t_w 等于 v_C 由 V_{th13} 下降至 V_{th9} 的时间与 v_C 由 V_{th9} 充电至 V_{th13} 两个时间之和。为获得较宽的输出脉冲,一般都将 V_{th13} 设计得较高而将 V_{th9} 设计得较低。

为说明 MCI4528 的可重复触发特性,分析图中 $t_5 \sim t_7$ 时的工作情况。如前所述,在 t_5 时刻电路被触发进入暂态,电容很快放电后,又进入充电状态。当 v_C 尚未充至 V_{th13} 时, t_6 时刻电路被再次触发, G_2 门的低电平使 $v_{O4} = V_{OL}$, 门 G_{10} 输出高电平, T_N 管导通,电容 C 又放电,当放电使 $v_C \ll V_{th9}$ 时, G_{10} 门输出低电平, T_N 管截止,电容又充电,一直充到 V_{th13} 且在没有触发信号作用时,电路才返回至稳态。显然,在这两个重复脉冲触发下,输出脉冲宽度 $t_w = t_\Delta + t_w$ 。这种可重复触发单稳态可利用在暂稳态加触发脉冲的方法增加输出脉宽。

9.2.3 单稳态触发器的应用

单稳态触发器是数字电路中常用的基本单元电路,典型应用介绍如下:

1. 定时

由于单稳态触发器能生产一定宽度 t_w 的矩形输出脉冲,如利用这个矩形脉冲作为定时信号去控制某电路,使其在 t_w 时间内动作(或不动作)。例如,利用单稳输出的矩形脉冲作为与门输入的控制信号(见图 9.2.8),则只有这个矩形波的 t_w 时间内,信号 v_A 才有可能通过与门。

2. 延时

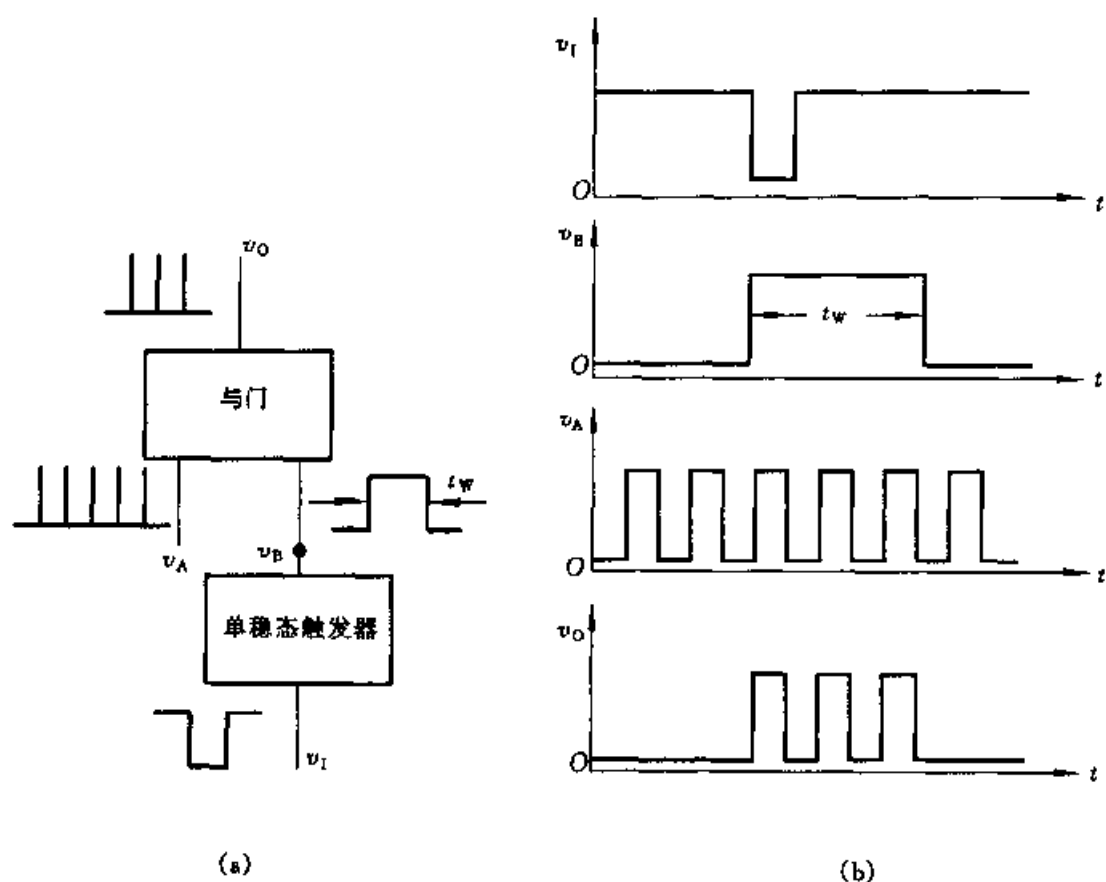


图 9.2.8 单稳态触发器作定时电路的应用

(a) 逻辑图 (b) 波形图

单稳态触发器的延时作用不难从图 9.2.2 所示微分型单稳态触发器的工作波形可看出。图中输出端 v_{O1} 的上升沿相对输入信号 v_1 的上升沿延迟了 t_w —

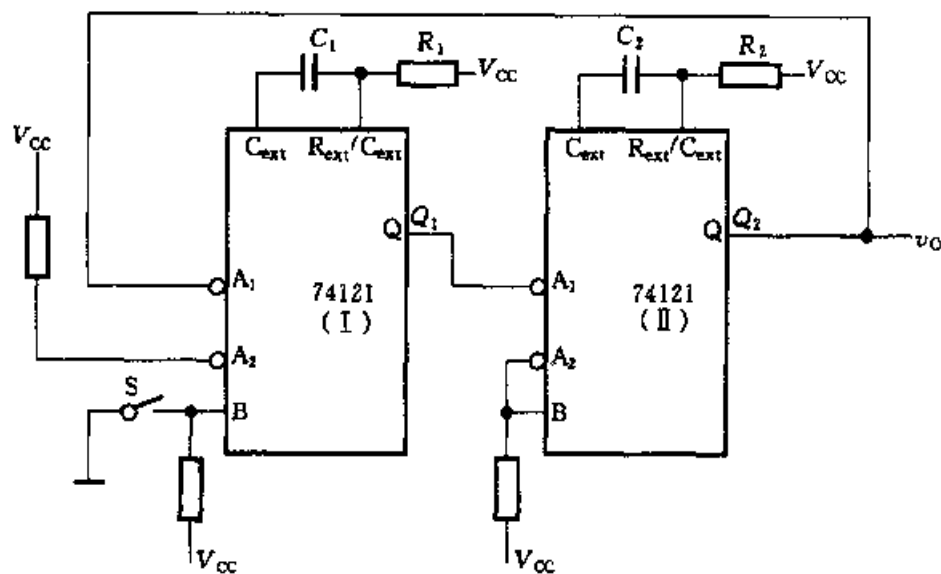


图 9.2.9 由单稳构成的多谐振荡器

段时间。单稳态的延时作用常被应用于时序控制。

3. 多谐振荡器

利用两个单稳态触发器可以构成多谐振荡器。由两片 74121 集成单稳态触发器组成的多谐振荡器如图 9.2.9 所示,图中开关 S 为振荡器控制开关。

设当电路处于 $Q_1 = 0, Q_2 = 0$ 时,将开关 S 打开,电路开始振荡,其工作过程如下:在起始时,单稳态触发器 I 的 A_1 为低电平,开关 S 打开瞬间,B 端产生正跳变,单稳 I 被触发, Q_1 输出正脉冲,其脉冲宽度 $0.7R_1C_1$,当单稳 I 暂稳态结束时, Q_1 的下跳沿触发单稳 II, Q_2 端输出正脉冲,此后, Q_2 的下跳沿又触发单稳 I,如此周而复始地产生振荡,其振荡周期为

$$T = 0.7(R_1C_1 + R_2C_2)$$

4. 噪声消除电路

利用单稳态触发器可以构成噪声消除电路(或称脉宽鉴别电路)。通常噪声多表现为尖脉冲,宽度较窄,而有用的信号都具有一定的宽度。利用单稳电路,将输出脉宽调节到大于噪声宽度而小于信号脉宽,即可消除噪声。由单稳态触发器组成的噪声消除电路及波形如图 9.2.10 所示。

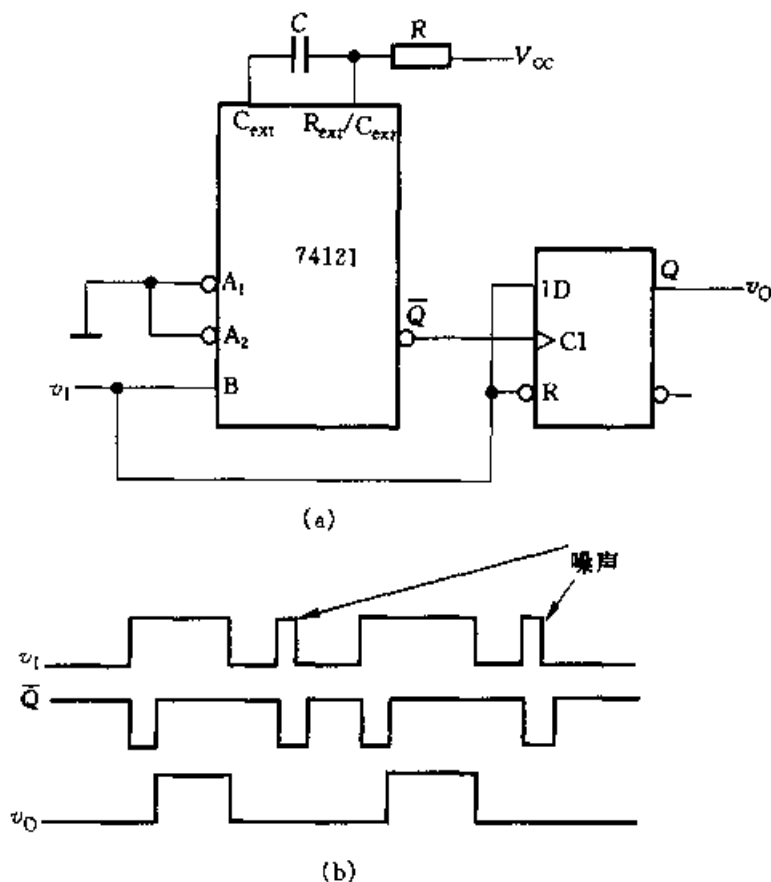


图 9.2.10 噪声消除电路

(a) 逻辑图 (b) 波形图

图中,输入信号接至单稳态触发器的触发输入端和 D 触发器的数据输入端及直接置 0 端。由于有用信号大于单稳态输出脉宽,因此单稳 \bar{Q} 输出上升沿使 D 触发器置 1 ,而当信号消失后, D 触发器被清 0 。若输入中含有噪声,其噪声前沿使单稳触发翻转,但由于单稳输出脉宽大于噪声宽度,故单稳 \bar{Q} 输出上升沿时,噪声已消失,从而在输出信号中消除了噪声成分。

复习思考题

9.2.1 与触发器相比较,单稳态触发器在电路结构和工作原理方面有什么特点?

9.2.2 用与非门构成的单稳与用或非门组成的单稳,它们的工作原理是否相同?它们的触发脉冲和输出脉冲有何区别?

9.2.3 图 9.2.1 所示单稳态触发器对触发脉冲有何要求?如触发脉冲宽度大于单稳态触发器输出脉宽,试问电路会产生什么现象?应如何解决?

9.2.4 图 9.2.1 所示单稳态触发器的输出脉宽与电路中哪些参数有关?对 R 的取值是否应有限制?为什么?

9.2.5 集成单稳分为哪两类?它们的区别是什么?

9.2.6 用集成单稳 74121 产生输出脉宽等于 3 ms 的脉冲信号,如选 $R = 2\text{ k}\Omega$ (内部电阻),试问外接电容 C_{ext} 应取何值?

9.2.7 你能否通过外加连线的方法将 MC14528 改变为不可重复触发单稳?试画出电路并说明其工作原理。

9.3 施密特触发器

施密特触发器^①不同于前述的各类触发器,它具有下述特点:

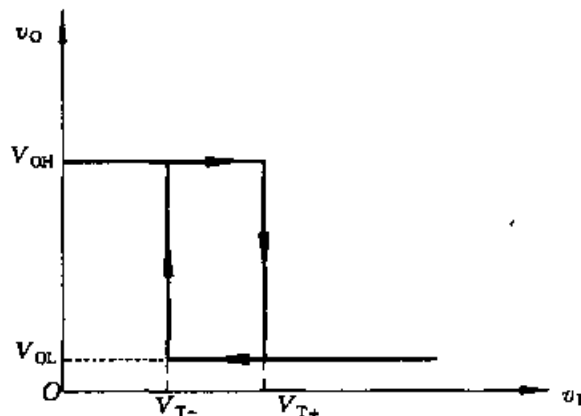


图 9.3.1 施密特电路的传输特性

① 施密特触发器系 Schmitt trigger 的译称,为纪念该电路的发明人 OTTO H. Schmitt 而命名。

① 施密特触发器属于电平触发,对于缓慢变化的信号仍然适用,当输入信号达到某一定电压值时,输出电压会发生突变。

② 输入信号增加和减少时,电路有不同的阈值电压,它具有如图 9.3.1 所示的传输特性。

在模拟电路中,曾经讨论过由集成运放构成的施密特触发器(带正反馈的迟滞比较器),这里将介绍数字技术中常用的施密特触发器。

9.3.1 门电路组成的施密特触发器

由 CMOS 门组成的施密特触发器如图 9.3.2 所示。电路中两个 CMOS 反相器串接,分压电阻 R_1 、 R_2 将输出端的电压反馈到输入端对电路产生影响。

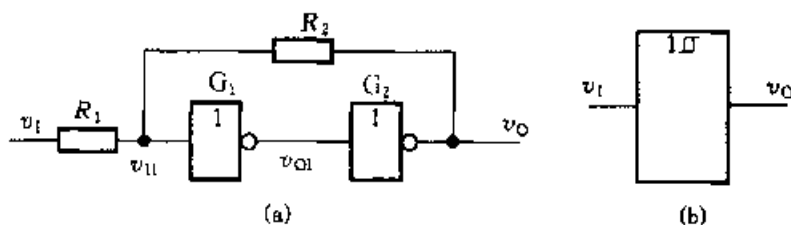


图 9.3.2 CMOS 反相器组成的施密特触发器
(a) 电路 (b) 图形符号

假定电路中 CMOS 反相器的阈值电压 $V_{th} \approx V_{DD}/2$, $R_1 < R_2$ 且输入信号 v_i 为三角波,下面分析电路的工作过程。

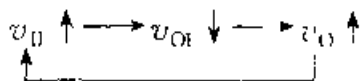
由电路不难看出, G_1 门的输入电平 v_{i1} 决定着电路的状态,根据叠加原理有

$$v_{i1} = \frac{R_2}{R_1 + R_2} \cdot v_i + \frac{R_1}{R_1 + R_2} \cdot v_O \quad (9.3.1)$$

当 $v_i = 0$ V 时, G_1 门截止, G_2 门导通, 输出端 $v_O = 0$ V, 此时 $v_{i1} \approx 0$ V。

输入从 0 V 电压逐渐增加, 只要 $v_{i1} < V_{th}$, 则电路保持 $v_O = 0$ V 不变。

当 v_i 上升使得 $v_{i1} = V_{th}$ 时, 使电路产生如下正反馈过程:



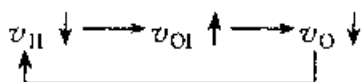
这样, 电路状态很快转换为 $v_O \approx V_{DD}$, 此时 v_i 的值即为施密特触发器在输入信号正向增加时的阈值电压, 称为正向阈值电压, 用 V_{T+} 表示。即由式(9.3.1)得

$$v_{i1} = V_{th} \approx \frac{R_2}{R_1 + R_2} \cdot V_{T+} \quad (9.3.2)$$

所以
$$V_{T+} = \left(1 + \frac{R_1}{R_2}\right) V_{th} \quad (9.3.3)$$

当 $v_{i1} > V_{th}$ 时, 电路状态维持 $v_O = V_{DD}$ 不变。

v_{i1} 继续上升至最大值后开始下降,当 $v_{i1} = V_{th}$ 时,电路产生如下正反馈过程:

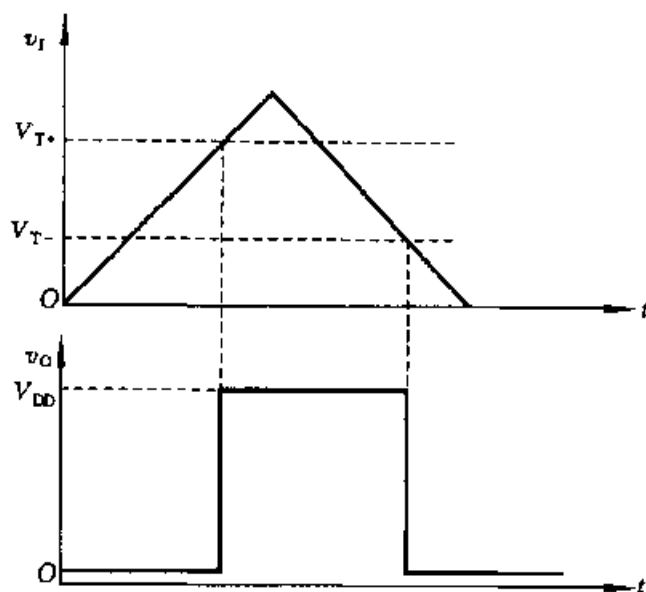


这样电路又迅速转换为 $v_o \approx 0$ V 的状态,此时的输入电平为 v_i 减小时的阈值电压,称为负向阈值电压,用 V_{T-} 表示。根据式(9.3.1),此时有

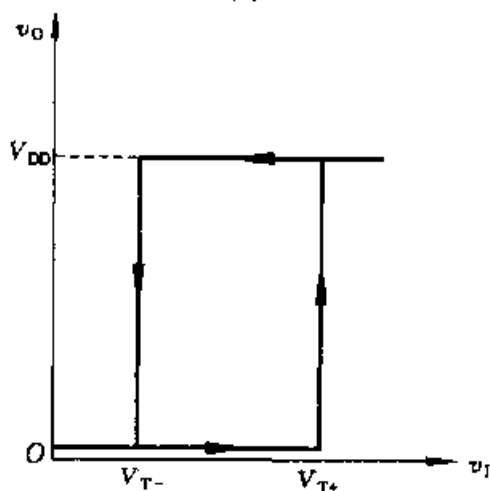
$$v_{i1} \approx V_{th} = \frac{R_2}{R_1 + R_2} \cdot V_{1-} + \frac{R_1}{R_1 + R_2} \cdot V_{DD}$$

将 $V_{DD} = 2 V_{th}$ 代入可得

$$V_{T-} \approx \left(1 - \frac{R_1}{R_2}\right) V_{th} \quad (9.3.4)$$



(a)



(b)

图 9.3.3 施密特触发器工作波形及传输特性曲线

(a) 工作波形 (b) 传输特性曲线

只要满足 $v_1 < V_{T-}$, 施密特电路就稳定在 $v_0 \approx 0$ V 的状态。

由式(9.3.3)和式(9.3.4)可求得回差电压为

$$\begin{aligned} \Delta V_T &= V_{T+} - V_{T-} \\ &\approx 2 \frac{R_1}{R_2} V_{th} \end{aligned} \quad (9.3.5)$$

上式表明, 电路回差电压与 R_1/R_2 成正比, 改变 R_1 、 R_2 的比值即可调节回差电压的大小。

电路的工作波形及传输特性如图 9.3.3 所示。

9.3.2 集成施密特触发器

集成施密特触发器性能稳定, 应用广泛, 下面以 CMOS 集成施密特触发器 CC40106(见图 9.3.4)为例介绍其工作原理。

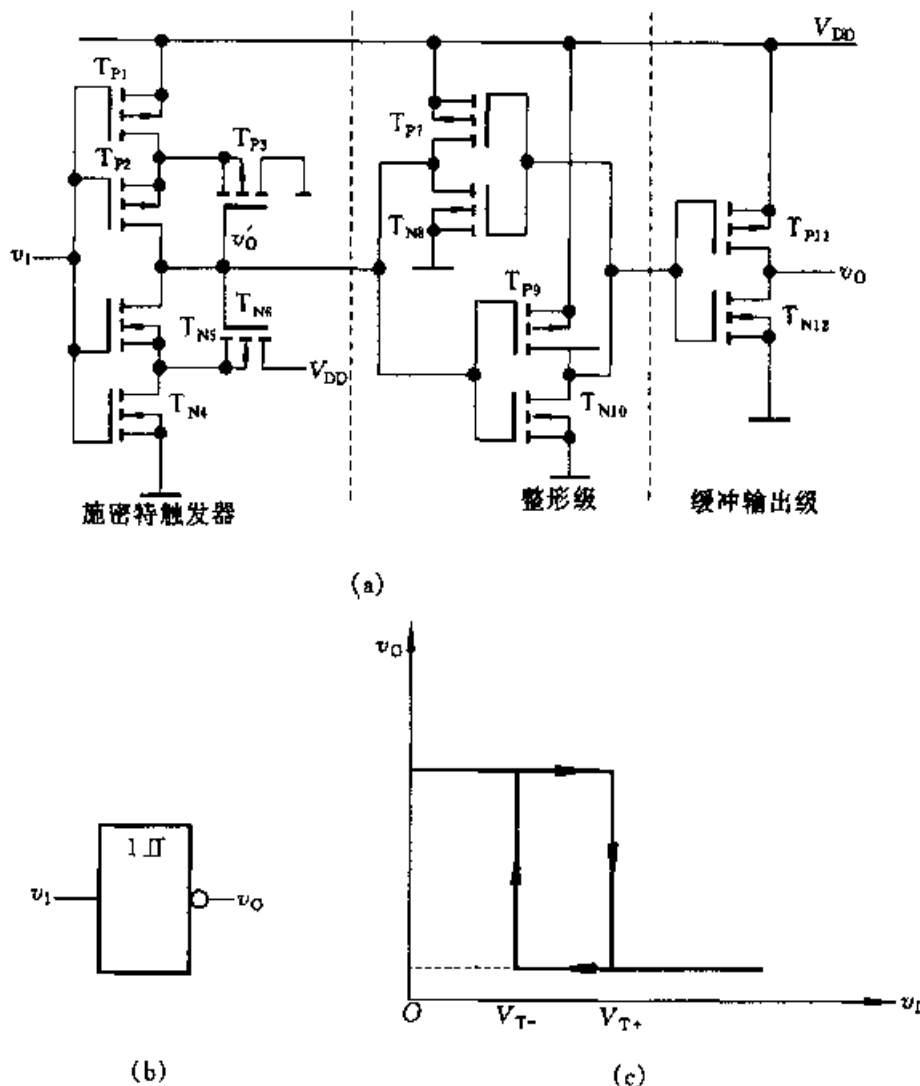


图 9.3.4 CMOS 集成施密特触发器电路
(a) 电路图 (b) 逻辑符号 (c) 传输特性曲线

由图 9.3.4a 可见,电路由施密特电路、整形级和缓冲输出级组成,其核心部分为施密特电路。

1. 施密特电路

施密特电路由 P 沟道 MOS 管 $T_{P1} \sim T_{P3}$ 、N 沟道 MOS 管 $T_{N4} \sim T_{N6}$ 组成,设 P 沟道 MOS 管的开启电压为 V_{TP} ,N 沟道 MOS 管开启电压为 V_{TN} ,输入信号 v_i 为三角波。

当 $v_i = 0$ 时, T_{P1} 、 T_{P2} 导通, T_{N4} 、 T_{N5} 截止,电路中 v'_O 为高电平 ($v'_O \approx V_{DD}$), $v_O = V_{OH}$ 。 v'_O 的高电平同时使 T_{P3} 截止, T_{N6} 导通且工作于源极输出状态。 T_{N5} 的源极电位 $v_{S5} \approx V_{DD} - V_{TN}$, 该电位较高。

v_i 电位逐渐升高,当 $v_i > V_{TN}$ 时, T_{N4} 先导通,由于 T_{N5} 其源极电压 v_{S5} 较大,即使 $v_i > V_{DD}/2$, T_{N5} 仍不能导通,直至 v_i 继续升高直至 T_{P1} 、 T_{P2} 趋于截止时,随着其内阻增大, v'_O 和 v_{S5} 才开始相应减少。

当 $v_i - v_{S5} \geq V_{TN}$ 时, T_{N5} 导通,并引起如下正反馈过程:

$$v'_O \downarrow \rightarrow v_{S5} \downarrow \rightarrow v_{GS5} \uparrow \rightarrow R_{GS5} \text{ (} T_{N5} \text{ 导通电阻) } \downarrow$$

于是 T_{P1} 、 T_{P2} 迅速截止, v'_O 为低电平,电路输出状态转换为 $v_O = 0$ 。

v'_O 的低电平使 T_{N6} 截止, T_{P3} 导通且工作于源极输出器状态, T_{P2} 的源极电压 $v_{S2} \approx 0 - V_{TP}$ 。

同理可分析,当 v_i 逐渐下降时,电路工作过程与 v_i 上升过程类似,只有当 $|v_i - v_{S2}| > |V_{TP}|$ 时,电路又转换为 v'_O 为高电平, $v_O = V_{OH}$ 的状态。

在 $V_{DD} \gg V_{TN} + |V_{TP}|$ 的条件下,电路的正向阈值电压 V_{T+} 远大于 $V_{DD}/2$,且随着 V_{DD} 增加而增加。在 v_i 下降过程中的负向阈值电压 V_{T-} 也要比 $V_{DD}/2$ 低得多。

由上述分析可知,电路在 v_i 上升和下降过程分别有不同的两个阈值电压,具有施密特电压传输特性。其传输特性如图 9.3.4c 所示。

2. 整形级

整形级由 T_{P7} 、 T_{N8} 、 T_{P9} 、 T_{N10} 组成,电路为两个首尾相连的反相器。在 v'_O 上升和下降过程中,利用两级反相器的正反馈作用可使输出波形有陡直的上升沿和下降沿。

3. 输出级

输出级为 T_{P11} 和 T_{N12} 组成的反相器,它不仅能起到与负载隔离的作用,而且提高了电路带负载能力。

9.3.3 施密特触发器的应用

施密特触发器的用途很广,其典型应用举例如下:

(1) 波形的整形及变换

利用施密特触发器将正弦波、三角波变换成方波,已在模拟电路中讨论过,不再赘述。这里主要讨论整形。通常由测量装置来的信号,经放大后可能是不规则的波形,必须经施密特触发器整形。作为整形电路时,如果要求输出与输入同相,则可在上述集成施密特反相器后再加一级反相器。整形电路对回差电压又有什么要求呢?如果输入信号具有如图 9.3.5a 所示的顶部干扰,而又希望得到如图 9.3.5c 所示的波形,若回差电压较小,将出现图 9.3.5b 所示波形,顶部干扰造成了不良影响。此时,应选择回差电压较大的施密特触发器,以提高电路的抗干扰性能。

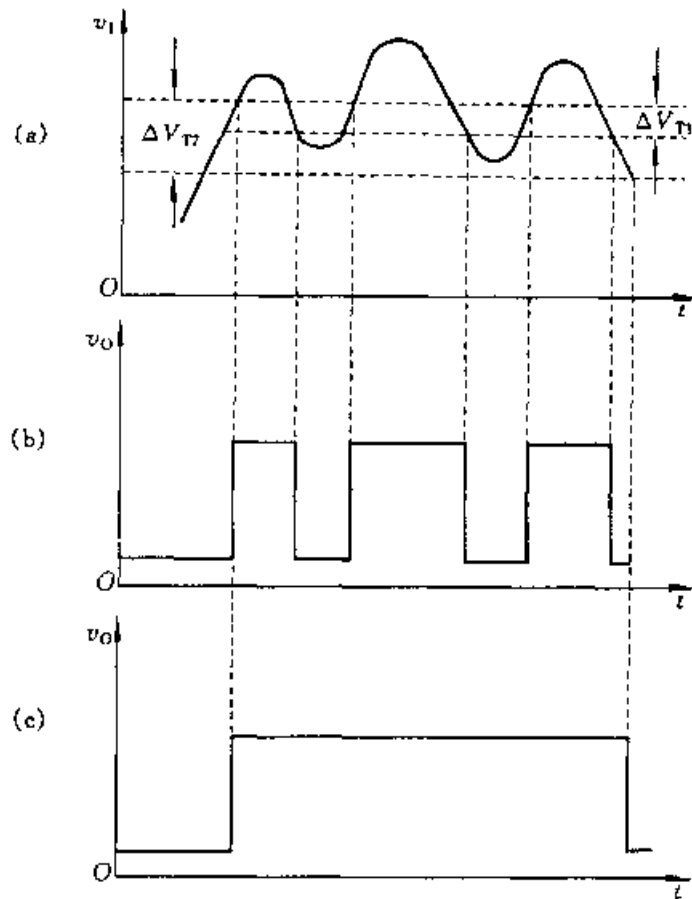


图 9.3.5 利用回差电压抗干扰

(a) 具有顶部干扰的输入信号 (b) 回差电压小时的输出波形 (c) 回差电压大于顶部干扰时输出波形

(2) 幅度鉴别

利用施密特触发器输出状态取决于输入信号 v_i 幅度的工作特点,可以用它来作为幅度鉴别电路。例如,输入信号为幅度不等的一串脉冲,需要消除幅度较小的脉冲,而保留幅度大于 V_{th} (见图 9.3.6)的脉冲,只要将施密特触发器的正向阈值电压 V_{T+} 调整到规定的幅度 V_{th} ,这样,幅度超过 V_{th} 的脉冲就使电路动作,有脉冲输出;而对于幅度小于 V_{th} 的脉冲,电路则无脉冲输出,从而达到幅度鉴别的目的。

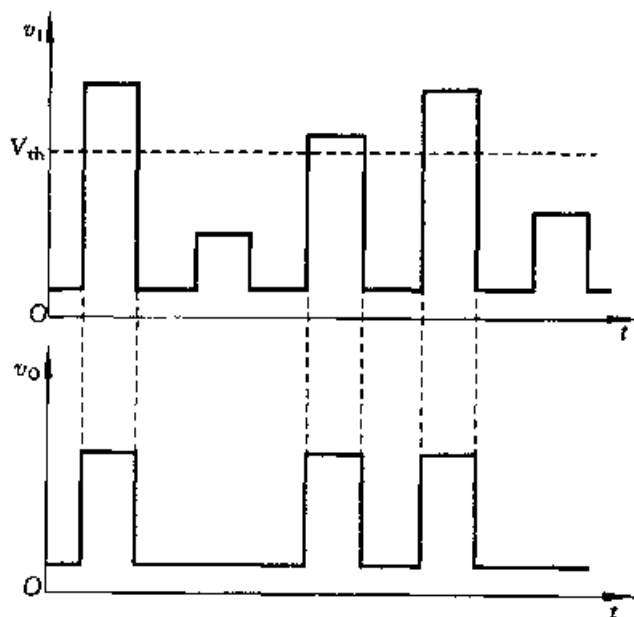


图 9.3.6 脉冲幅度鉴别

(3) 多谐振荡器

利用施密特触发器也可以构成多谐振荡器。其电路如图 9.3.7 所示。

接通电源瞬间,电容 C 上的电压为 0V ,输出 v_o 为高电平。 v_o 通过电阻 R 对电容 C 充电,当 v_i 达到 V_{T+} 时,施密特触发器翻转,输出为低电平,此后电容 C 又开始放电, v_i 下降,当 v_i 下降到 V_{T-} 时,电路又发生翻转,如此周而复始地形成振荡。其输入、输出波形如图 9.3.8 所示。

若在图 9.3.7 中采用的是 CMOS 施密特触发器,且 $V_{OH} \approx V_{DD}$, $V_{OL} \approx 0$,根据图 9.3.8 的电压波形得到振荡周期计算公式为

$$\begin{aligned} T &= T_1 + T_2 \\ &= RC \ln \frac{V_{DD} - V_{T-}}{V_{DD} - V_{T+}} + RC \ln \frac{V_{T+}}{V_{T-}} \end{aligned}$$

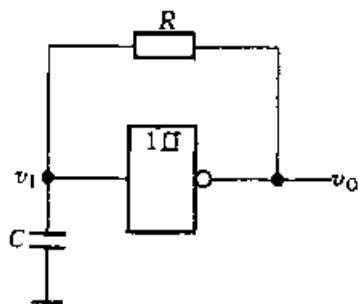


图 9.3.7 用施密特触发器构成的多谐振荡器

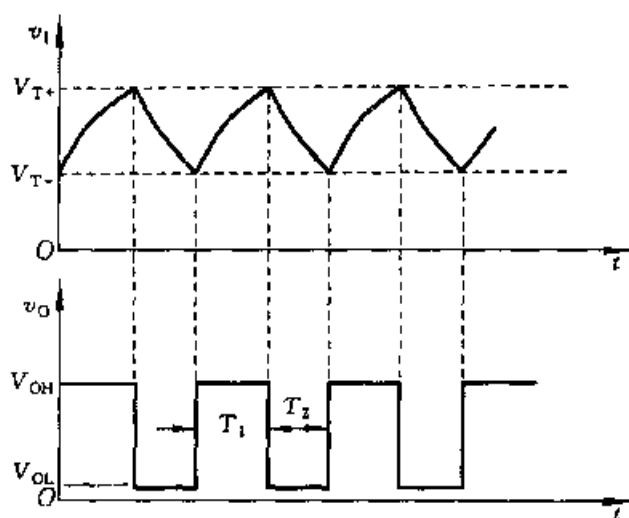


图 9.3.8 图 9.3.7 的波形

$$= RC \ln \left(\frac{V_{DD} - V_{T-}}{V_{DD} - V_{T+}} \cdot \frac{V_{T+}}{V_{T-}} \right) \quad (9.3.6)$$

当采用 TTL 施密特触发器(例如 7414)时,电阻 R 不能大于 470Ω ,以保证输入端能够达到负向阈值电平。 R 的最小值由门的扇出数确定(不得小于 100Ω)。对于典型的参数值($V_{T-} = 0.8 \text{ V}$, $V_{T+} = 1.6 \text{ V}$, 输出电压摆幅为 3 V),其输出的振荡频率为

$$f \approx 0.7/RC \quad (9.3.7)$$

最大可能的振荡频率为 10 MHz 。

复习思考题

- 9.3.1 施密特触发器工作特点如何? 它具有怎样的传输特性?
- 9.3.2 在图 9.3.2 中,为什么一般取 $R_1 < R_2$?
- 9.3.3 改变图 9.3.2 中 R_1 的取值,对电路的传输特性有何影响?
- 9.3.4 试简述施密特电路具有抗干扰特性的原理。
- 9.3.5 为消除图 9.3.5a 中的顶部幅度为 ΔV 的干扰,得到如图 c 的输出波形,应如何正确选择施密特触发器的正、负向阈值电压 V_{T+} 和 V_{T-} ?

9.4 555 定时器及其应用

555 定时器是一种应用极为广泛的中规模集成电路。该电路使用灵活、方便,只需外接少量的阻容元件就可以构成单稳、多谐和施密特触发器。因而广泛用于信号的产生、变换、控制与检测。

目前生产的定时器有双极型和 CMOS 两种类型,其型号分别有 NE555(或 5G555)和 C7555 等多种。它们的结构及工作原理基本相同。通常,双极型定时器具有较大的驱动能力,而 CMOS 定时电路具有低功耗、输入阻抗高等优点。555 定时器工作的电源电压很宽,并可承受较大的负载电流。双极型定时器电源电压范围为 5~16 V,最大负载电流可达 200 mA;CMOS 定时器电源电压范围为 3~18 V,最大负载电流在 4 mA 以下。

9.4.1 555 定时器

555 定时器内部结构的简化原理图如图 9.4.1 所示。它由 3 个阻值为 5 kΩ 的电阻组成的分压器,两个电压比较器 C_1 和 C_2 、基本 RS 触发器、放电 BJT T 以及缓冲器 G 组成。

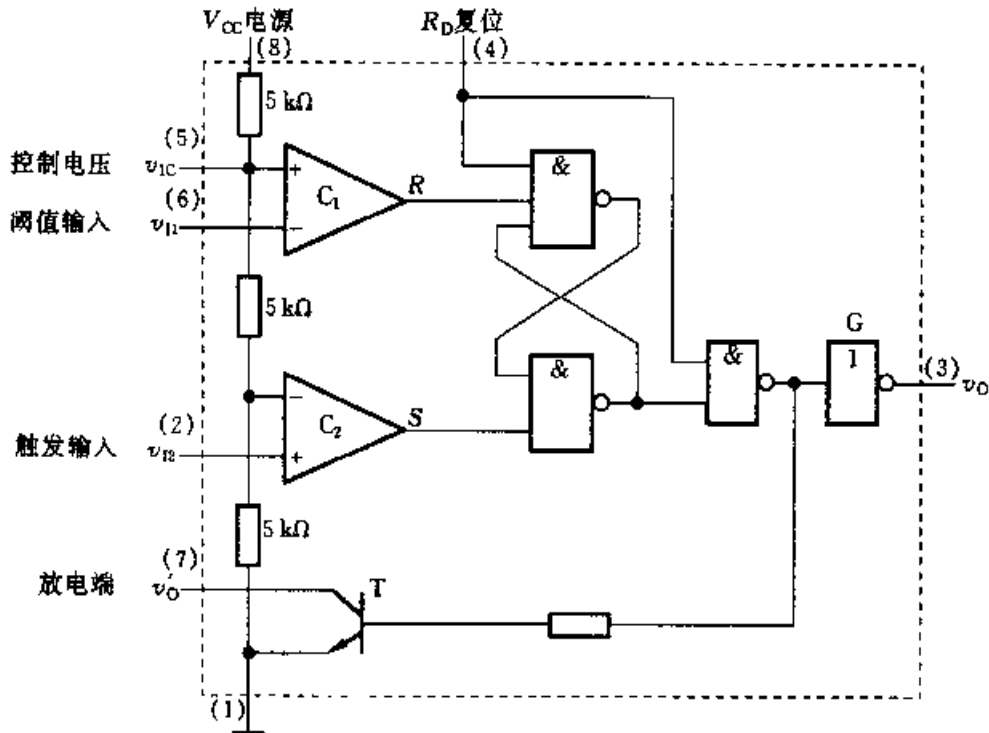


图 9.4.1 555 定时器原理图

定时器的主要功能取决于比较器,比较器的输出控制 RS 触发器和放电 BJT T 的状态。图中 R_D 为复位输入端,当 R_D 为低电平时,不管其他输入端的状态如何,输出 v_o 为低电平。因此在正常工作时,应将其接高电平。

由图可知,当 5 脚悬空时,比较器 C_1 和 C_2 的比较电压分别为 $\frac{2}{3} V_{CC}$ 和 $\frac{1}{3} V_{CC}$ 。

当 $v_{11} > \frac{2}{3} V_{CC}$, $v_{12} > \frac{1}{3} V_{CC}$ 时, 比较器 C_1 输出低电平, 比较器 C_2 输出高电平, 基本 RS 触发器被置 0, 放电三极管 T 导通, 输出端 v_o 为低电平。

当 $v_{11} < \frac{2}{3} V_{CC}$, $v_{12} < \frac{1}{3} V_{CC}$ 时, 比较器 C_1 输出高电平, C_2 输出低电平, 基本 RS 触发器置 1, 放电三极管截止, 输出端当 v_o 为高电平。

当 $v_{11} < \frac{2}{3} V_{CC}$, $v_{12} > \frac{1}{3} V_{CC}$ 时, 基本 RS 触发器 $R = 1$, $S = 1$, 触发器状态不变, 电路亦保持原状态不变。

综合上述分析, 可得 555 定时器功能表如表 9.4.1 所示。

表 9.4.1 555 定时器功能表

输 入			输 出	
阈值输入(v_{11})	触发输入(v_{12})	复位(R_D)	输出(v_o)	放电管 T
×	×	0	0	导通
$< \frac{2}{3} V_{CC}$	$< \frac{1}{3} V_{CC}$	1	1	截止
$> \frac{2}{3} V_{CC}$	$> \frac{1}{3} V_{CC}$	1	0	导通
$< \frac{2}{3} V_{CC}$	$> \frac{1}{3} V_{CC}$	1	不变	不变

如果在电压控制端(5脚)施加一个外加电压(其值在 $0 - V_{CC}$ 之间), 比较器的参考电压将发生变化, 电路相应的阈值、触发电平也将随之变化, 并进而影响电路的工作状态。读者可自行分析。

9.4.2 定时器应用举例

1. 单稳态触发器

由 555 构成的单稳态触发器及工作波形如图 9.4.2 所示。电源接通瞬间, 电路有一个稳定的过程, 即电源通过电阻 R 向电容 C 充电, 当 v_c 上升到 $\frac{2}{3} V_{CC}$ 时, 触发器复位, v_o 为低电平, 放电 BJT T 导通, 电容 C 放电, 电路进入稳定状态。

若触发输入端施加触发信号($v_i < \frac{1}{3} V_{CC}$), 触发器发生翻转, 电路进入暂稳态, v_o 输出高电平, 且 BJT T 截止。此后电容 C 充电至 $v_c = \frac{2}{3} V_{CC}$ 时, 电路又发生翻转, v_o 为低电平, T 导通, 电容 C 放电, 电路恢复至稳定状态。

如果忽略 T 的饱和压降, 则 v_c 从零电平上升到 $\frac{2}{3} V_{CC}$ 的时间, 即为输出电

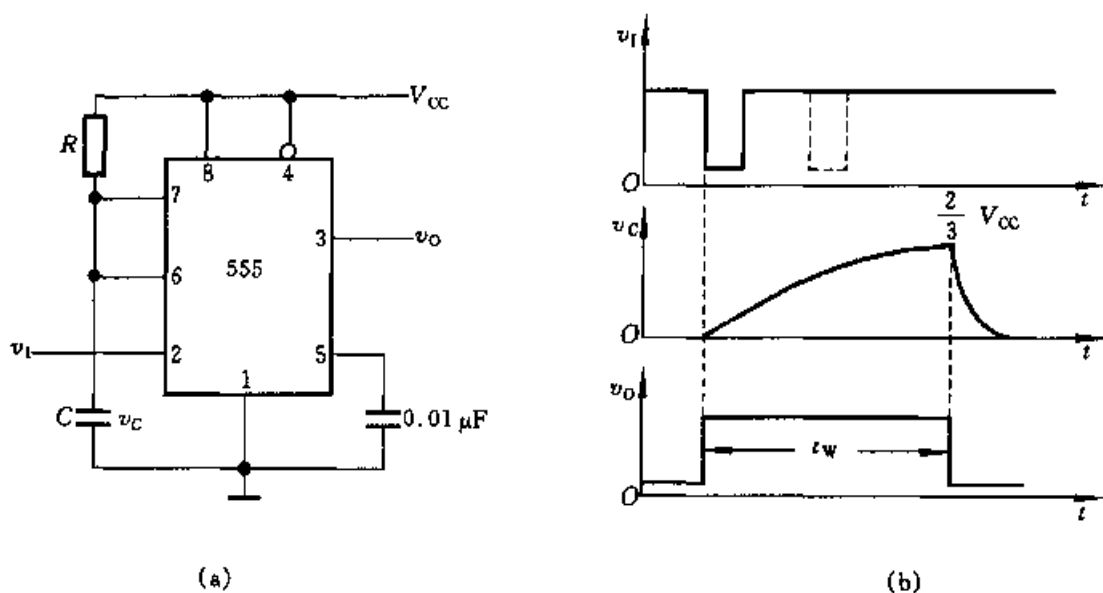


图 9.4.2 由 555 定时器构成的单稳态触发器

(a) 电路图 (b) 工作波形

压 v_o 的脉宽 t_w 。

$$t_w = RC \ln 3 \approx 1.1RC \quad (9.4.1)$$

这种电路产生的脉冲宽度可从几个微秒到数分钟,精度可达 0.1%。

通常 R 的取值在几百欧姆至几兆欧姆之间,电容取值为几百皮法到几百微法。由图 9.4.2 可知,如果在电路的暂稳态持续时间内,加入新的触发脉冲,如图 9.4.2b 中的虚线所示,则该脉冲不起作用,电路为不可重复触发单稳。

由 555 定时器构成的可重复触发单稳电路如图 9.4.3 所示。

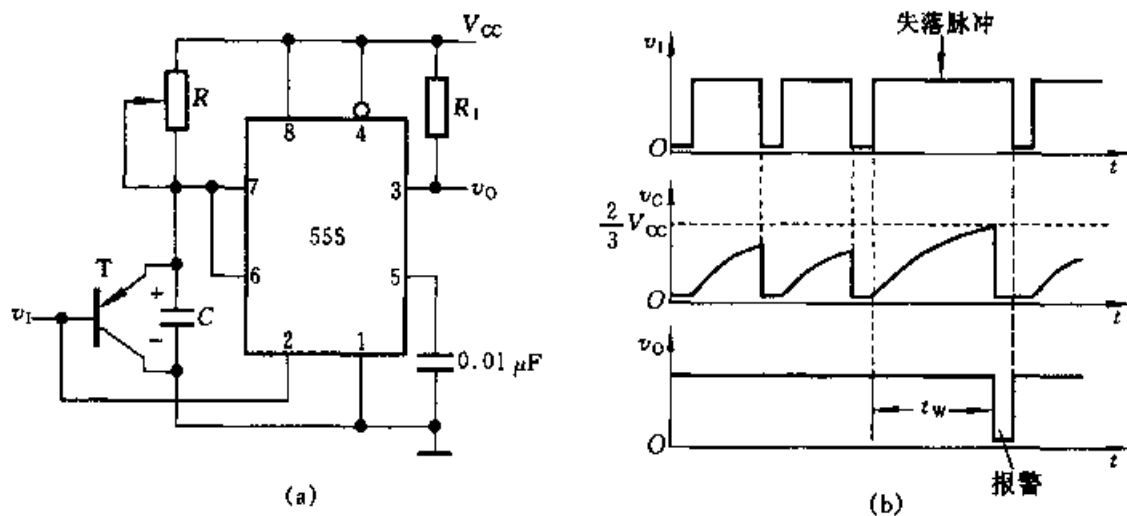


图 9.4.3 由 555 定时器构成的可重复触发单稳电路

(a) 电路图 (b) 工作波形

当 v_i 输入负向脉冲后, 电路进入暂稳态, 同时 BJT T 导通, 电容 C 放电。输入脉冲撤除后, 电容 C 充电, 在 v_c 未充到 $\frac{2}{3} V_{CC}$ 之前, 电路处于暂稳态。如果在此期间, 又加入新的触发脉冲, BJT T 又导通, 电容 C 再次放电, 输出仍然维持在暂稳态。只有在触发脉冲撤除后且在输出脉宽 t_w 时间间隔内没有新的触发脉冲, 电路才返回到稳定状态。这种电路可作为失落脉冲检出电路, 对机器的转速或人体的心律进行监视, 当机器转速降到一定限度或人体的心律不齐时就发出报警信号。

如果在控制电压端(5脚)施加一个变化电压, 由 555 构成的单稳电路可作为脉冲宽度调制器, 如图 9.4.4 所示。

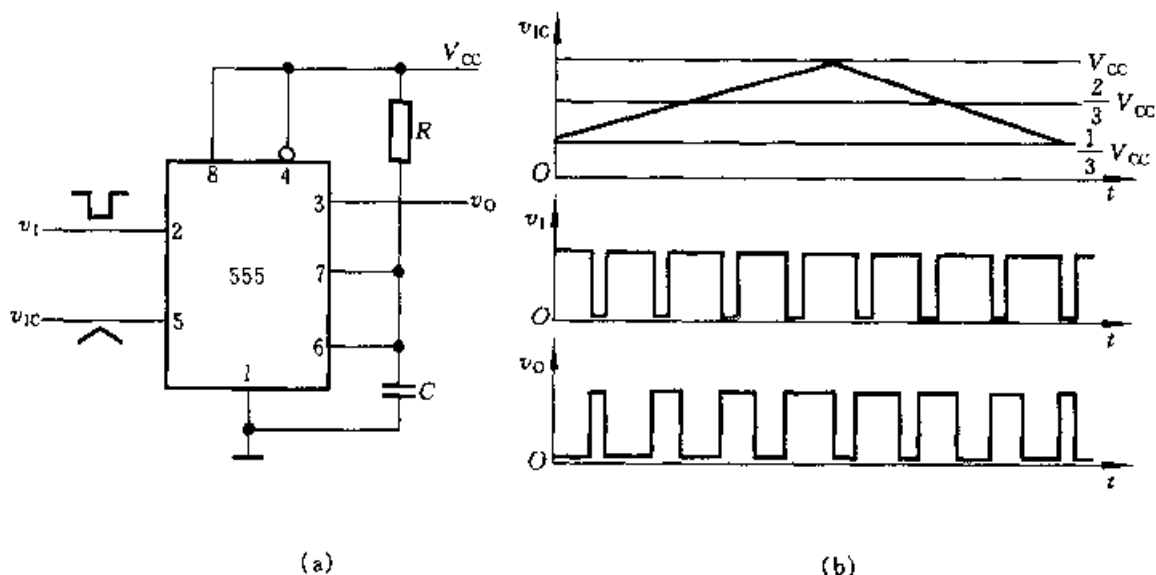


图 9.4.4 脉冲宽度调制器

(a) 逻辑图 (b) 波形图

当控制电压升高时, 电路的阈值电压也升高, 输出的脉冲宽度随之增加; 而当控制电压降低时, 电路的阈值电压也降低, 单稳的输出脉宽则随之减小。因此, 若控制电压如图 b 所示的三角波时, 在单稳的输出端便得到一串随控制电压变化的脉冲宽度调制波。从 v_{IC} 与 v_o 波形关系可看出, 该电路可实现电压-频率转换。

2. 多谐振荡器

由 555 定时器构成的多谐振荡器如图 9.4.5a 所示, 其工作波形见图 b。

接通电源后, 电容 C 被充电, v_c 上升, 当 v_c 上升到 $\frac{2}{3} V_{CC}$ 时, 触发器被复位, 同时放电 BJT T 导通, 此时 v_o 为低电平, 电容 C 通过 R_2 和 T 放电, 使 v_c

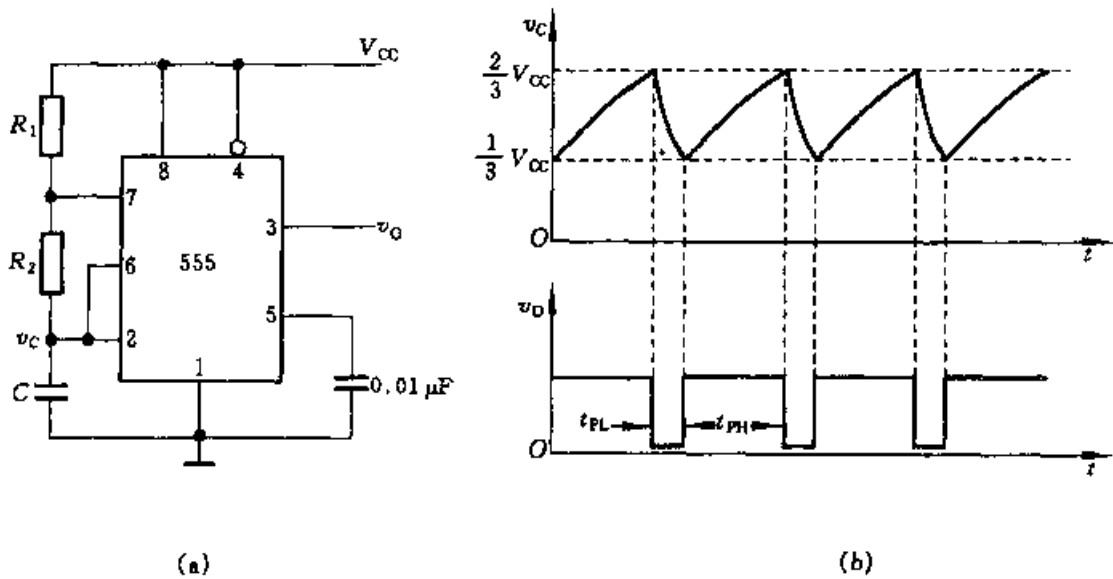


图 9.4.5 由 555 定时器构成的多谐振荡器

(a) 电路图 (b) 工作波形

下降。当 v_c 下降到 $\frac{1}{3}V_{CC}$ 时, 触发器又被置位, v_o 翻转为高电平。电容器 C 放电所需的时间为

$$t_{PL} = R_2 C \ln 2 \approx 0.7 R_2 C \quad (9.4.2)$$

当 C 放电结束时, T 截止, V_{CC} 将通过 R_1 、 R_2 向电容器 C 充电, v_c 由 $\frac{1}{3}V_{CC}$ 上升到 $\frac{2}{3}V_{CC}$ 所需的时间为

$$t_{PH} = (R_1 + R_2) C \ln 2 \approx 0.7(R_1 + R_2) C \quad (9.4.3)$$

当 v_c 上升到 $\frac{2}{3}V_{CC}$ 时, 触发器又发生翻转, 如此周而复始, 在输出端就得到一个周期性的方波, 其频率为

$$f = \frac{1}{t_{PL} + t_{PH}} \approx \frac{1.43}{(R_1 + 2R_2) C} \quad (9.4.4)$$

由于 555 内部的比较器灵敏度较高, 而且采用差分电路形式, 它的振荡频率受电源电压和温度变化的影响很小。

图 9.4.5 所示电路的 $t_{PL} \neq t_{PH}$, 而且占空比固定不变。如果将电路改成如图 9.4.6 所示的形式, 电路利用 D_1 、 D_2 单向导电特性将电容器 C 充、放电回路分开, 再加上电位器调节, 便构成了占空比可调的多谐振荡器。图中, V_{CC} 通过 R_A 、 D_1 向电容 C 充电, 充电时间为

$$t_{PH} \approx 0.7 R_A C \quad (9.4.5)$$

电容器 C 通过 D_2 、 R_B 及 555 中的 BJT T 放电, 放电时间为

$$t_{PL} \approx 0.7R_H C \quad (9.4.6)$$

因而,振荡频率为

$$f = \frac{1}{t_{PH} + t_{PL}} \approx \frac{1.43}{(R_A + R_B)C} \quad (9.4.7)$$

可见,这种振荡器输出波形的占空比为

$$q(\%) = \frac{R_A}{R_A + R_B} \times 100\% \quad (9.4.8)$$

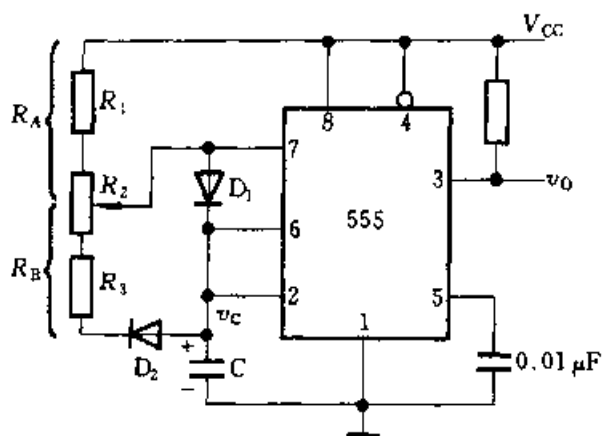


图 9.4.6 占空比可调的方波发生器

3. 施密特触发器

将 555 定时器的阈值输入端和触发输入端连在一起,便构成了施密特触发器,如图 9.4.7a 所示。当输入如图

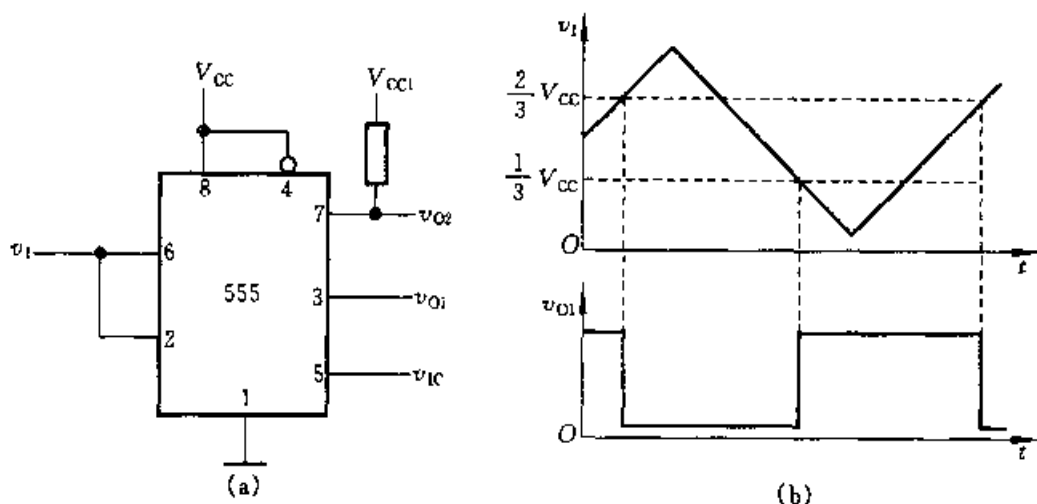


图 9.4.7 由 555 定时器构成的施密特触发器

(a) 电路图 (b) 波形图

9.4.7b 所示的三角波信号时,则从施密特触发器的 v_{O1} 端可得到方波输出。

如将图中 5 脚外接控制电压 v_{IC} , 改变 v_{IC} 的大小,可以调节回差电压的范围。如果在 555 定时器的放电 BJT 输出端(7 脚)外接一电阻,并与另一电源 V_{CC} 相连,则由 v_{O2} 输出的信号可实现电平转换。

上面仅讨论了由 555 定时器组成的单稳态触发器、多谐振荡器和施密特触发器,实际上,由于 555 定时器的比较器灵敏度高,输出驱动电流大,功能灵活,因而在电子电路中获得广泛应用,限于篇幅,这里就不一一枚举了。

复习思考题

9.4.1 555 定时器具有哪些应用特点？其典型应用电路有哪几种？

9.4.2 555 定时器中缓冲器 G 有什么作用？

9.4.3 如将图 9.4.2a 中 555 定时器的控制电压输入端(5 脚)改接电压 V_R 上,试问电路的输出脉宽 t_w 有无改变? t_w 与 V_R 的关系如何?

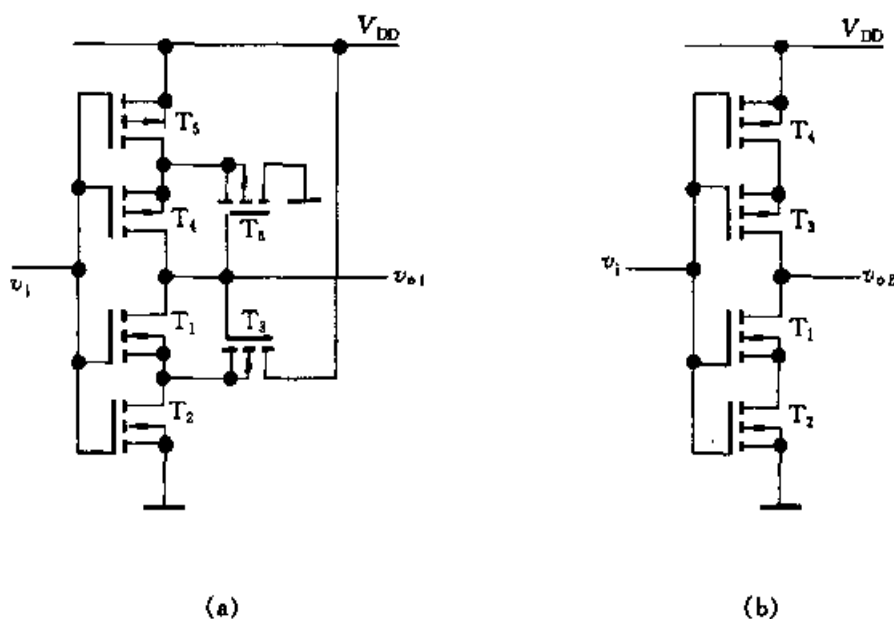
9.4.4 能否采用改变图 9.4.5 中 555 定时器控制电压的方法调节电路的振荡频率? 为什么? 如要实现振荡、停振可控,你会采用哪些方法实现?

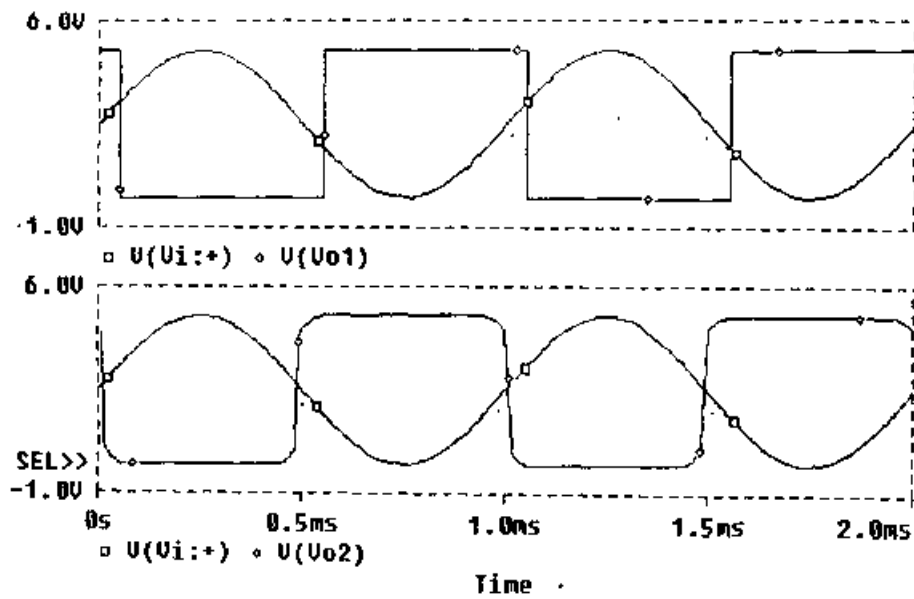
*9.5 CAD 例题

例 CE9.5.1 施密特触发器和 CMOS 反相器分别如图 CE9.5.1a、b 所示。设电路中 $V_{DD} = 5\text{ V}$, MOS 管开启电压 $|V_T| = 1.5\text{ V}$, 输入为含 2.5 V 的直流分量的正弦信号, 其频率为 1 kHz , 幅值为 2.5 V 。试用 PSPICE 程序分析两电路的输出电压波形, 并指出两电路的差别。

解: 电路中, PMOS 管选用模型 IRF9140, NMOS 管选用 IRF150, 并分别将它们的开启电压 V_o 设置为 -1.5 V 和 1.5 V 。其他参数按题目要求设置(正弦信号源 VSIN 的 VOFF = 2.5 V , 表示 2.5 V 的直流分量)。设置瞬态分析, 仿真后输入、输出波形如图 CE9.5.1c 所示。

由图看出, 施密特触发器输出波形的翻转沿比反相器的翻转沿陡, 且施密特触发器有正、负向两个阈值电压, 而反相器只有一个阈值电压。





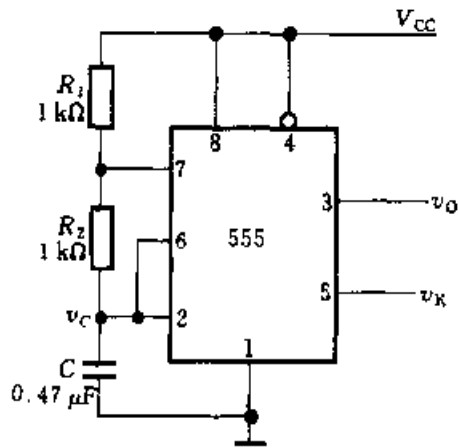
(c)

图 CE9.5.1

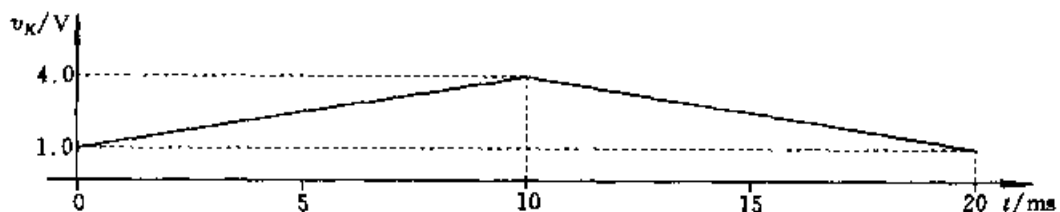
(a) 施密特触发器 (b) CMOS 反相器 (c) 仿真后的输入输出波形

例 CE9.5.2 由 555 构成的压控频率振荡电路如图 CE9.5.2a 所示。当控制电压 v_K 为图 b 所示波形时, 试用 PSPICE 程序分析 v_C 、 v_O 的电压波形。

解: 设置瞬态分析, 仿真后的波形如图 c 所示。



(a)



(b)

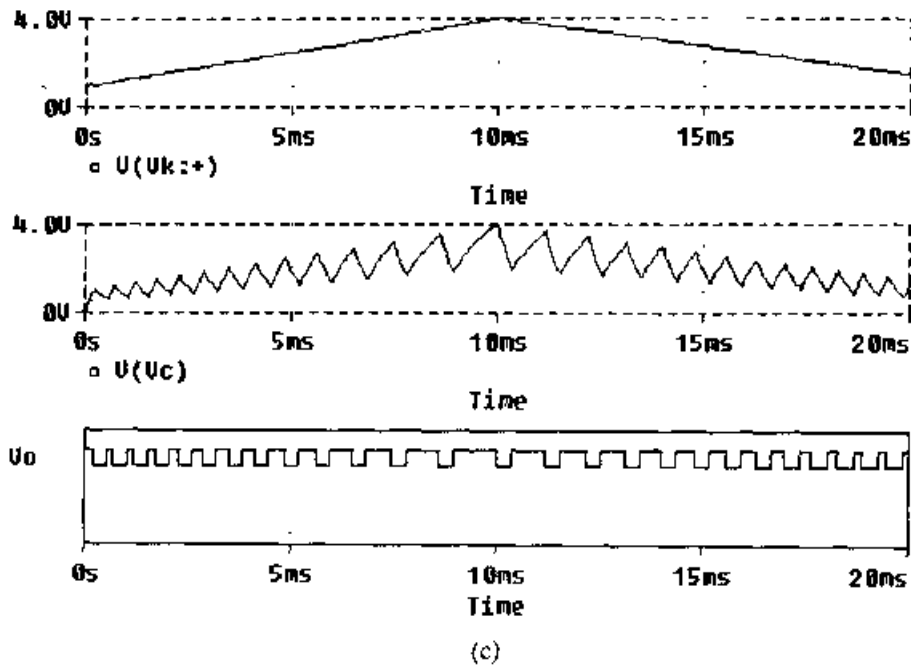


图 CE9.5.2

(a) 由 555 定时器构成的压控频率振荡电路 (b) 控制电压 U_K 的波形 (c) 仿真后 U_K 、 U_C 、 U_O 的波形

小 结

• 由门电路构成的单稳、多谐振荡器电路和基本 RS 触发器在结构上极其相似,都属于图 9.5.1 所示的反馈结构,只是耦合网络不同,因而 RS 触发器具有两个稳态,单稳触发器只有一个稳态,多谐振荡器则没有稳态。

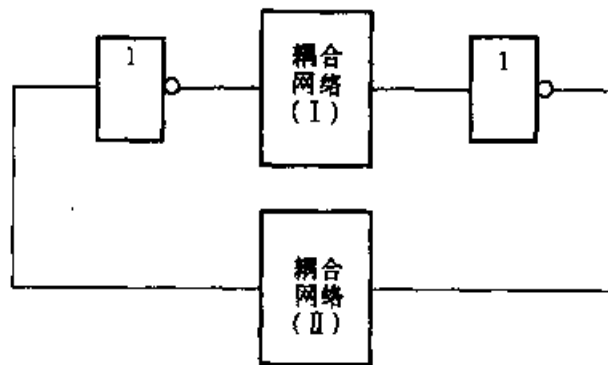


图 9.5.1 由门电路构成的多谐单稳和 RS 触发器的结构框图

• 多谐振荡器无需外加输入信号就能在接通电源后自行产生矩形波输出。在频率稳定性要求较高的场合通常采用石英晶体振荡器。

• 在单稳和多谐振荡器中,电路由暂稳态过渡到另一个状态,其“触发”信号是由电路内部电容充(放)电提供的,因此无需外部触发脉冲。暂稳态持续的时

间是脉冲电路的主要参数,它与电路的阻容元件取值有关。电路中,RC 电路充、放电过程对相应门输入电平的影响是分析电路的关键。

- 在数字电路中,施密特触发器实质上是具有滞后特性的逻辑门,它有两个阈值电压。电路状态与输入电压有关,不具备记忆功能。除施密特反相器外还有施密特与非门,或非门等。

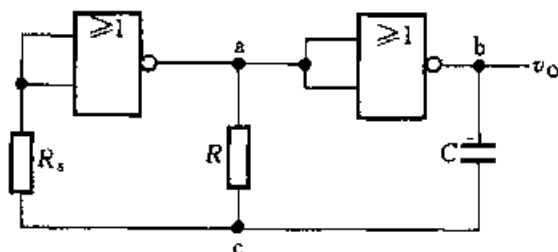
- 集成单稳态触发器分为非重复触发和可重复触发两大类,在暂稳态期间,出现的触发信号对非重复触发单稳电路没有影响,而对可重复触发单稳电路可起到连续触发作用。

- 定时器是一种应用广泛的集成器件,多用于脉冲产生、整形及定时等。除 555 定时器外,目前还有 556(双定时器)、558(四定时器)等。

习 题

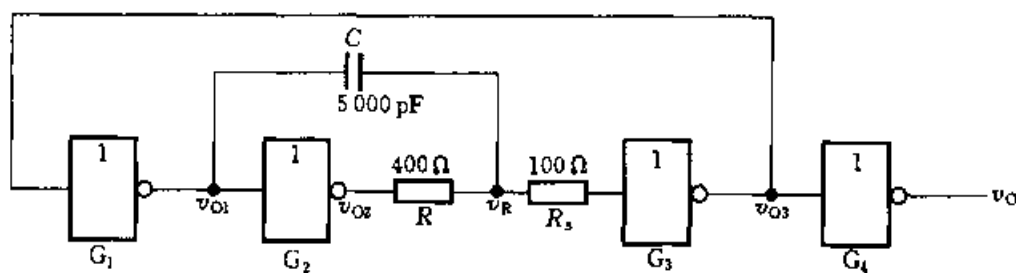
9.1.1 图示电路为 CMOS 或非门构成的多谐振荡器,图中 $R_s = 10R$ 。

(1) 画出 a、b、c 各点的波形;(2) 计算电路的振荡周期;(3) 当阈值电压 V_{th} 由 $\frac{1}{2} V_{DD}$ 改变至 $\frac{2}{3} V_{DD}$ 时,电路的振荡频率如何变化? 与图 9.1.1 电路相比,说明 R_s 的作用。



图题 9.1.1

9.1.2 RC 环形多谐振荡电路如图所示,试分析电路的振荡过程,画出 v_{O1} 、 v_{O2} 、 v_R 、 v_{O3} 及 v_O 的波形。

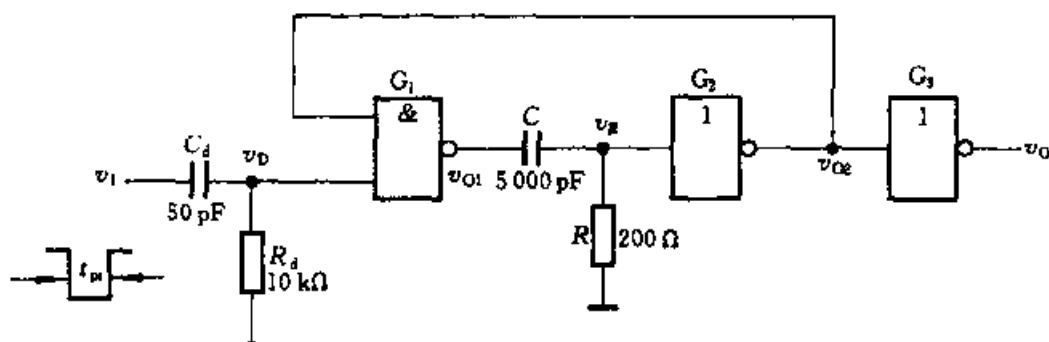


图题 9.1.2

9.2.1 微分型单稳电路如图所示。其中 t_p 为 $3 \mu s$, $C_d = 50 \text{ pF}$, $R_d = 10 \text{ k}\Omega$, $C = 5000$

μF 、 $R = 200 \Omega$ ，试对应地画出 v_1 、 v_D 、 v_{O1} 、 v_R 、 v_{O2} 、 v_O 的波形，并求出输出脉冲宽度。

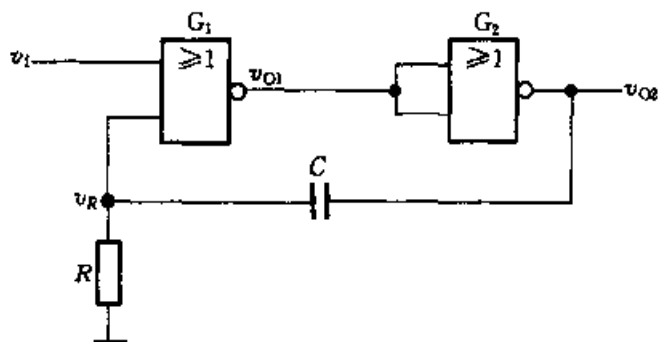
提示：TTL 与非门的门坎电平为 1.4 V，当 G_1 开通时， v_D 被钳在约 1.4 V 上。



图题 9.2.1

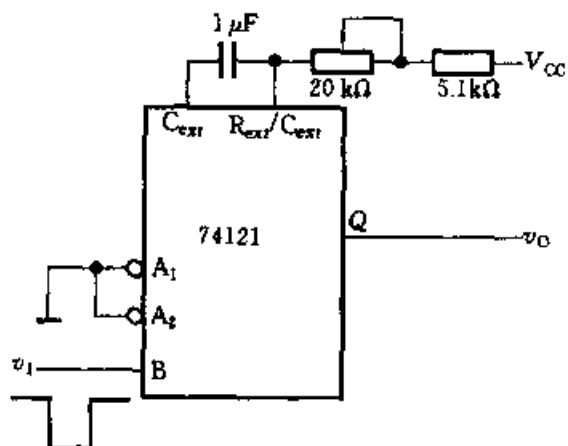
9.2.2 图题 9.2.2 所示电路为由 CMOS 或非门构成的单稳态触发器的另一种形式。试回答下列问题：

(1) 分析电路的工作原理；(2) 画出加入触发脉冲后 v_{O1} 、 v_{O2} 及 v_R 的工作波形；(3) 写出输出脉宽 t_w 表达式。



图题 9.2.2

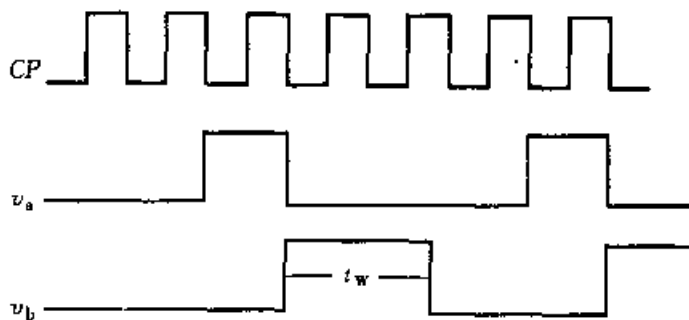
9.2.3 由集成单稳态触发器 74121 组成的延时电路及输入波形如图题 9.2.3 所示。



图题 9.2.3

(1) 计算输出脉宽的变化范围;(2) 解释为什么使用电位器时要串接一个电阻。

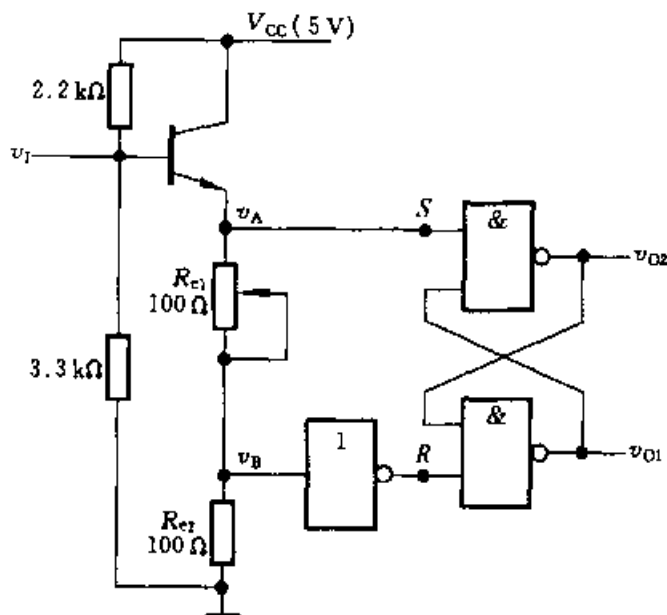
9.2.4 某控制系统要求产生的时序信号 v_a 、 v_b 与系统时钟 CP 的时序关系如图题 9.2.4 所示。试用 4 位二进制计数器 74163、集成单稳 74121 设计该信号产生电路,画出电路图。



图题 9.2.4

9.3.1 图题 9.3.1 所示电路为一个回差可调的施密特电路,它是利用射极跟随器的发射极电阻来调节回差的。试:

- (1) 分析电路的工作原理;
- (2) 当 R_{e1} 在 $50 \sim 100 \Omega$ 的范围内变动时,回差的变化范围。

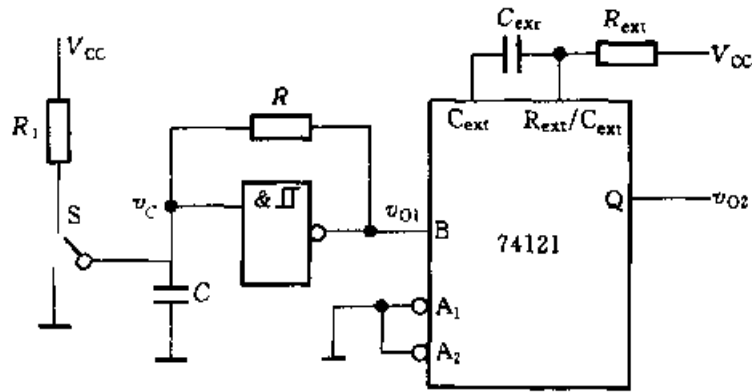


图题 9.3.1

9.3.2 集成施密特电路和集成单稳态触发器 74121 构成的电路如图题 9.3.2 所示。已知集成施密特电路的 $V_{DD} = 10 \text{ V}$, $R = 100 \text{ k}\Omega$, $C = 0.01 \mu\text{F}$, $V_{T+} = 6.3 \text{ V}$, $V_{T-} = 2.7 \text{ V}$; $C_{ext} = 0.01 \mu\text{F}$, $R_{ext} = 30 \text{ k}\Omega$ 。

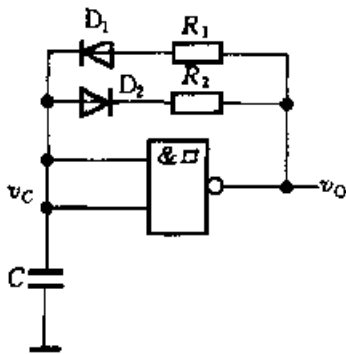
- (1) 分别计算 v_{O1} 的周期及 v_{O2} 的脉宽;

(2) 根据计算结果,画出 v_C 、 v_{O1} 、 v_{O2} 的波形。

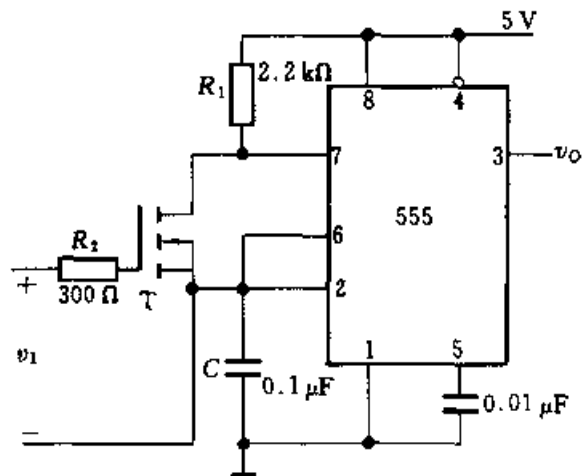


图题 9.3.2

9.3.3 由集成施密特 CMOS 与非门电路组成的脉冲占空比可调多谐振荡器如图题 9.3.3 所示。已知电路中 R_1 、 R_2 、 C 及 V_{DD} 、 V_{T+} 、 V_{T-} 的值。(1) 定性画出 v_C 及 v_O 波形;(2) 写出输出信号频率的表达式。



图题 9.3.3

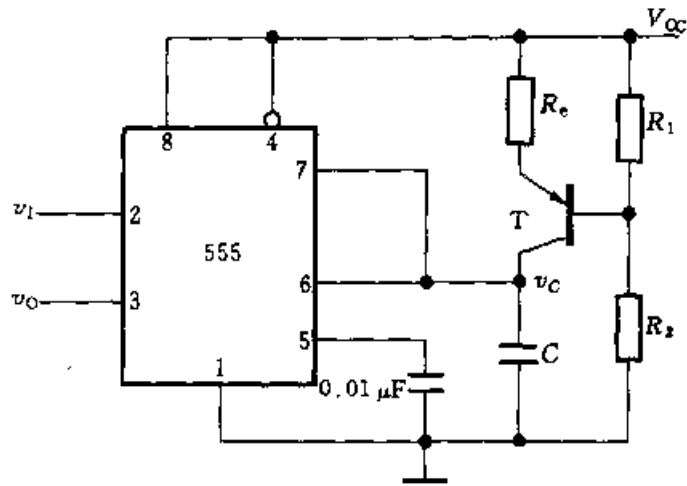


图题 9.4.1

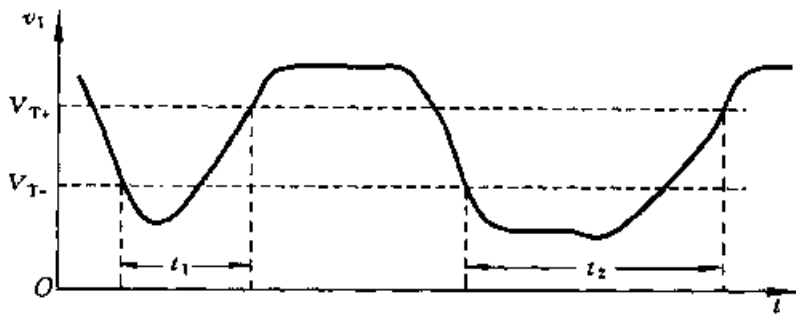
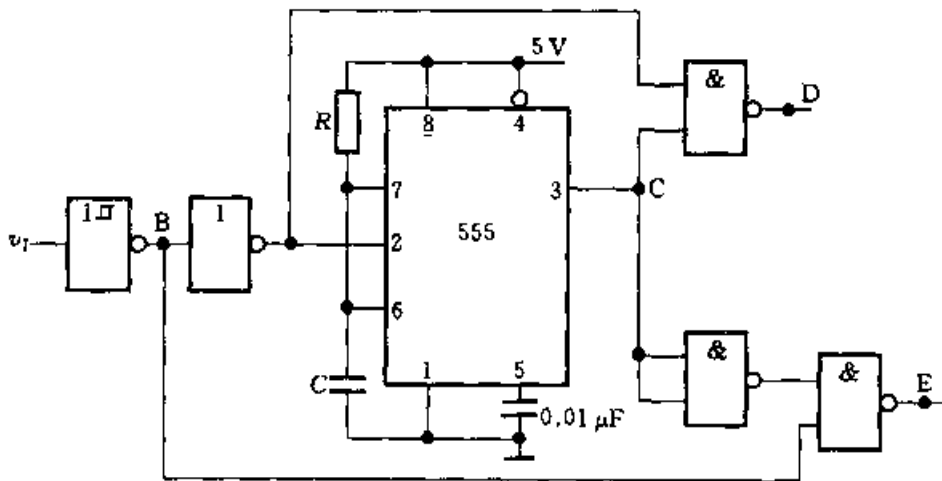
9.4.1 由 555 定时器及场效应管 T 组成的某功能电路如图所示,电路中 T 工作于可变电阻区,其导通电阻为 R_{DS} 。(1) 说明电路功能;(2) 写出输出端 v_O 频率的表达式。

9.4.2 图示电路为由 555 定时器构成的锯齿波发生器,BJT T 和电阻 R_1 、 R_2 、 R_c 构成恒流源,给定时电容 C 充电,当触发输入端输入负脉冲后,画出电容电压 v_C 及 555 输出端 v_O 的波形,并计算电容 C 充电的时间。

9.4.3 由 555 定时器组成的脉冲宽度鉴别电路及输入 v_1 波形如图题 9.4.3 所示。集成施密特电路的 $V_{T+} = 3\text{ V}$ 、 $V_{T-} = 1.6\text{ V}$,单稳的输出脉宽 t_w 有 $t_1 < t_w < t_2$ 的关系。对应 v_1 画出电路中 B、C、D、E 各点波形。

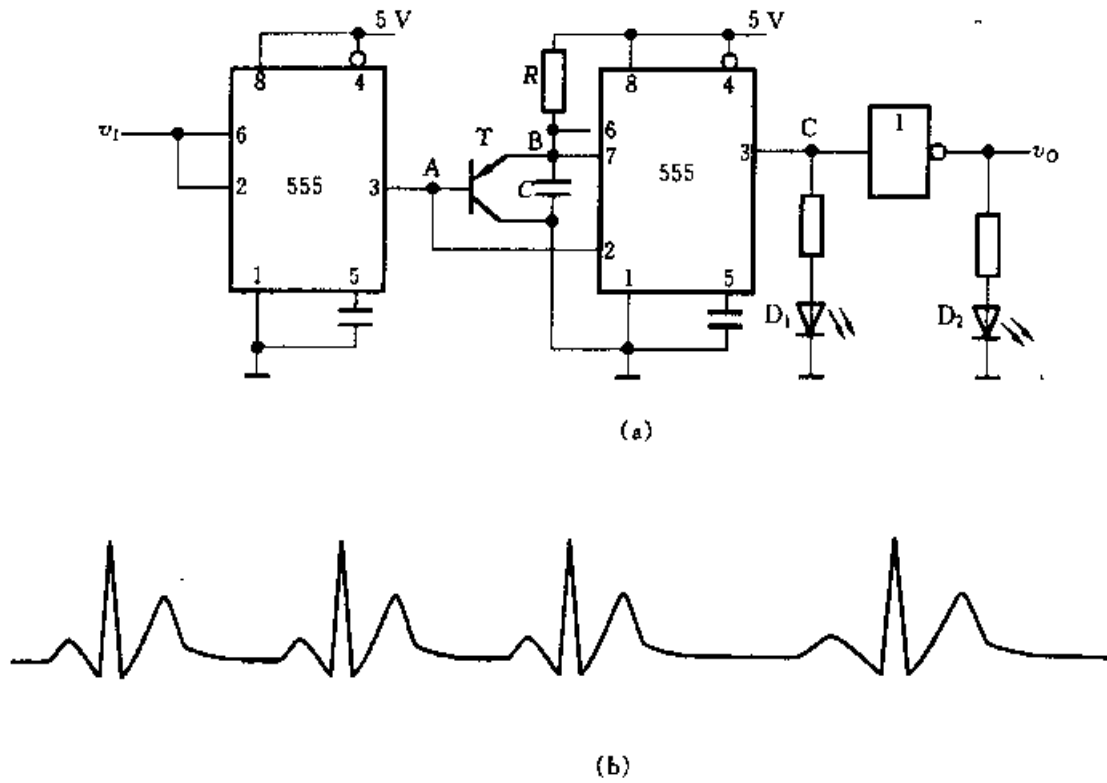


图题 9.4.2



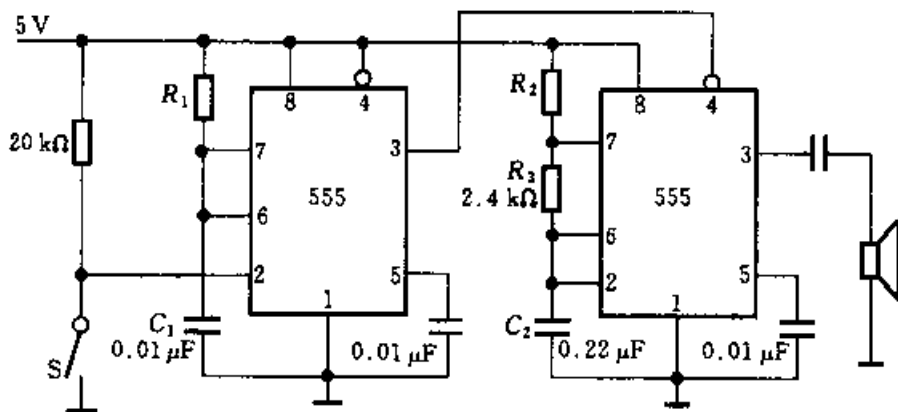
图题 9.4.3

9.4.4 图题 9.4.4a 为心律失常报警电路,经放大后的心电信号 v_i 如图 b 所示, v_i 的幅值 $v_m = 4\text{ V}$ 。(1) 对应 v_i 分别画出图中 A、B、C 三点波形;(2) 说明电路的组成及工作原理。



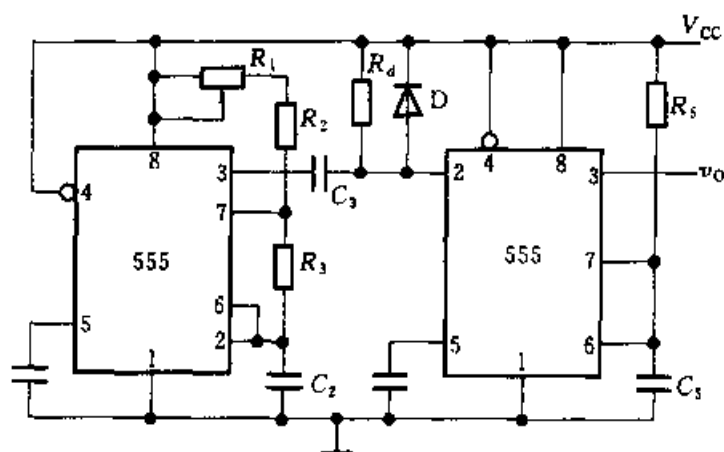
图题 9.4.4

9.4.5 分析如图题 9.4.5 所示电路,简述电路组成及工作原理。若要求扬声器在开关 S 按下后,以 1.2 kHz 的频率持续响 10 s,试确定图中 R_1 、 R_2 的阻值。



图题 9.4.5

9.4.6 图题 9.4.6 所示电路为两个 555 定时器构成的频率可调、而脉宽不变的方波发生器,试说明工作原理;确定频率变化的范围和输出脉宽;解释二极管 D 在电路中的作用。



图题 9.4.6

* CAD 习题

C9.5.1 电路与图 CE9.5.1a、b 相同,应用 PSPICE 程序对其进行仿真。求电压传输特性 $v_1 - v_{O1}$ 、 $v_1 - v_{O2}$ 和施密特触发器正、负向阈值电压。

C9.5.2 电路如图题 9.4.6 所示。设 $R_2 = 10 \text{ k}\Omega$, $R_3 = 10 \text{ k}\Omega$, $R_4 = 4.7 \text{ k}\Omega$, $R_5 = 2 \text{ k}\Omega$, $C_1 = 0.1 \mu\text{F}$, $C_2 = 4.7 \mu\text{F}$, $C_3 = 0.1 \mu\text{F}$, $C_4 = 0.1 \mu\text{F}$, $C_5 = 1 \mu\text{F}$ 。试用 PSPICE 程序分析, (1) 当 $R_1 = 50 \text{ k}\Omega$, 滑动触头处于中心位置时, 求 v_{O1} 、 v_{O2} 和 v_{O3} 的波形。(2) 当 R_1 的滑动触头处于最左端和最右端时, 求 v_{O1} 的波形。

C9.5.3 电路如图题 9.4.2 所示。设 $R_1 = 10 \text{ k}\Omega$, $R_2 = 15 \text{ k}\Omega$, $R_c = 1 \text{ k}\Omega$, $C = 0.1 \mu\text{F}$, v_1 输入频率为 100 Hz、脉宽为 0.5 ms 的负脉冲。当 C 为 $1 \mu\text{F}$ 和 $2.7 \mu\text{F}$ 时, 用 PSPICE 程序分析 v_c 的电压波形。

10 数模与模数转换器

引言 随着数字技术,特别是计算机技术的飞速发展与普及,在现代控制、通信及检测领域中,为提高系统的性能指标,对信号的处理无不广泛地采用了数字计算机技术。由于系统的实际对象往往都是一些模拟量(如温度、压力、位移、图像等),要使计算机或数字仪表能识别、处理这些信号,必须首先将这些模拟信号转换成数字信号;而经计算机分析、处理后输出的数字量也往往需要将其转换为相应模拟信号才能为执行机构所接收。这样,就需要一种能在模拟信号与数字信号之间起桥梁作用的电路——模数转换器和数模转换器。

能将模拟信号转换成数字信号的电路,称为模数转换器(简称 A/D^① 转换器);而将能把数字信号转换为模拟信号的电路称为数模转换器(简称为 D/A^② 转换器),A/D 转换器和 D/A 转换器已成为计算机系统中不可缺少的接口电路。

为确保系统处理结果的精确度,A/D 转换器和 D/A 转换器必须具有足够的转换精度;如要实现对快速变化的信号的实时控制与检测,A/D 与 D/A 转换器还要求具有较高的转换速度。转换精度与转换速度是衡量 A/D 与 D/A 转换器的重要技术指标。

随着集成技术的发展,现已研制和生产出许多单片的和混合集成型的 A/D 和 D/A 转换器。它们具有愈来愈先进的技术指标。在本章中,将介绍几种常用 A/D 与 D/A 转换器的电路结构、工作原理及其应用。

10.1 D/A 转换器

大家知道,数字量是用代码按数位组合起来表示的,对于有权码,每位代码都有一定的权。为了将数字量转换成模拟量,必须将每 1 位的代码按其权的大

① A/D 系 Analog to Digital Converter 的缩写。

② D/A 系 Digital to Analog Converter 的缩写。

小转换成相应的模拟量,然后将这些模拟量相加,即可得与数字量成正比的总模拟量,从而实现了数字-模拟转换。这就是组成 D/A 转换器的基本指导思想。

n 位 D/A 转换器的方框图如图 10.1.1 所示。

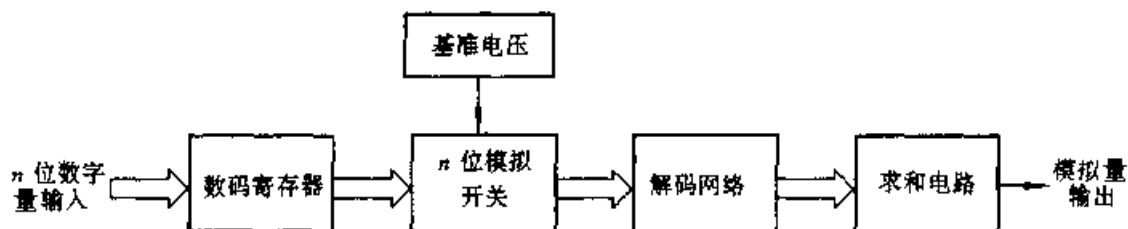


图 10.1.1 n 位 D/A 转换器方框图

D/A 转换器由数码寄存器、模拟电子开关电路、解码网络、求和电路及基准电压几部分组成。数字量以串行或并行方式输入并存储于数码寄存器中,寄存器输出的每位数码驱动对应数位上的电子开关将在电阻解码网络中获得的相应数位权值送入求和电路。求和电路将各位权值相加便得到与数字量对应的模拟量。

D/A 转换器按解码网络结构不同分为 T 形电阻网络、倒 T 形电阻网络 D/A 转换器、权电流 D/A 转换器及权电阻网络 D/A 转换器等。按模拟电子开关电路的不同,D/A 转换器又可分为 CMOS 开关型和双极型开关 D/A 转换器。其中双极型开关 D/A 转换器又分为电流开关型和 ECL 电流开关型两种,在速度要求不高的情况可选用 CMOS 开关型 D/A 转换器。如要求较高的转换速度则应选用双极型电流开关 D/A 转换器或转换速度更高的 ECL 电流开关型 D/A 转换器。

10.1.1 倒 T 形电阻网络 D/A 转换器

在单片集成 D/A 转换器中,使用最多的是倒 T 形电阻网络 D/A 转换器。以下以 4 位 D/A 转换器为例说明其工作原理。

4 位倒 T 形电阻网络 D/A 转换器的原理图如图 10.1.2 所示。图中 $S_0 \sim S_3$ 为模拟开关, $R-2R$ 电阻解码网络呈倒 T 形,运算放大器 A 组成求和电路。模拟开关 S_i 由输入数码 D_i 控制,当 $D_i = 1$ 时, S_i 接运算放大器反相端,电流 I_i 流入求和电路;当 $D_i = 0$ 时, S_i 则将电阻 $2R$ 接地。根据运算放大器线性运用时虚地的概念可知,无论模拟开关 S_i 处于何种位置,与 S_i 相连的 $2R$ 电阻均将接“地”(地或虚地)。这样,流经 $2R$ 电阻的电流与开关位置无关,为确定值。分析 $R-2R$ 电阻网络可以发现,从每个节点向左看的二端网络等效电阻均为 R ,流入每个 $2R$ 电阻的电流从高位到低位按 2 的整数倍递减。设由基准电压源提供的总电流为 $I(I = V_{REF}/R)$,则流过各开关支路(从右到左)的电流分别为 $I/2$ 、

$I/4$ 、 $I/8$ 和 $I/16$ 。

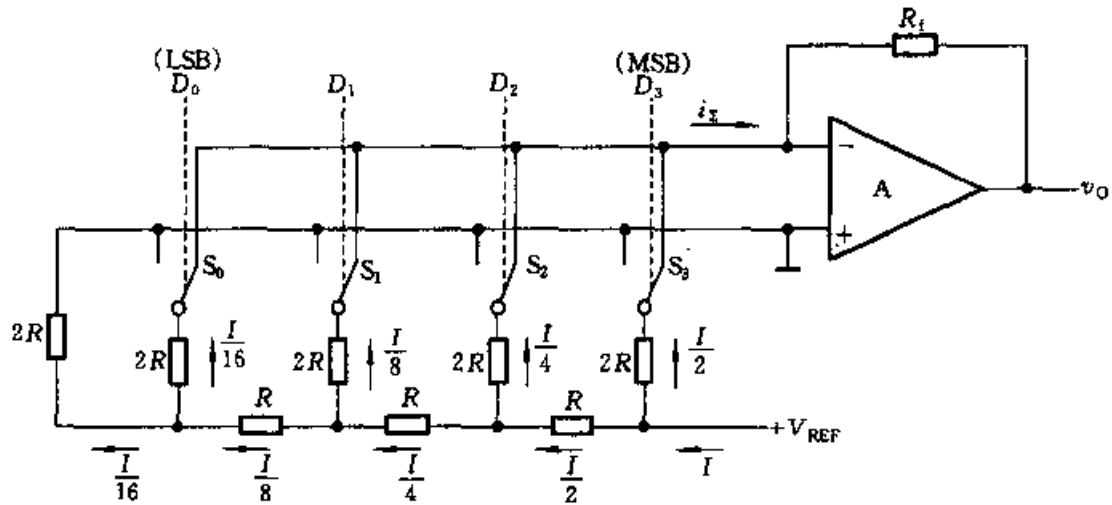


图 10.1.2 倒 T 形电阻网络 D/A 转换器

于是可得总电流

$$i_{\Sigma} = \frac{V_{\text{REF}}}{R} \left(\frac{D_0}{2^4} + \frac{D_1}{2^3} + \frac{D_2}{2^2} + \frac{D_3}{2^1} \right) = \frac{V_{\text{REF}}}{2^4 \times R} \sum_{i=0}^3 (D_i \cdot 2^i) \quad (10.1.1)$$

输出电压

$$v_O = -i_{\Sigma} R_f = -\frac{R_f}{R} \cdot \frac{V_{\text{REF}}}{2^4} \sum_{i=0}^3 (D_i \cdot 2^i) \quad (10.1.2)$$

将输入数字量扩展到 n 位, 可得 n 位倒 T 形电阻网络 D/A 转换器输出模拟量与输入数字量之间的一般关系式

$$v_O = -\frac{V_{\text{REF}}}{2^n} \cdot \frac{R_f}{R} \left[\sum_{i=0}^{n-1} (D_i \cdot 2^i) \right] \quad (10.1.3)$$

若将式中 $\frac{V_{\text{REF}}}{2^n} \cdot \frac{R_f}{R}$ 用 K 表示, 中括号内的 n 位二进制数用 N_B 表示, 则式 (10.1.3) 可改写为

$$v_O = -KN_B \quad (10.1.4)$$

式 (10.1.4) 表明, 对于在图 10.1.2 电路中输入的每一个二进制数 N_B , 均能在其输出端得到与之成正比的模拟电压 v_O 。

通过以上分析看到, 要使 D/A 转换器具有较高的精度, 对电路中的参数有以下要求: (1) 基准电压稳定性好; (2) 倒 T 形电阻网络中 R 和 $2R$ 电阻比值的精度要高; (3) 每个模拟开关的开关电压降要相等。为实现电流从高位到低位按 2 的整数倍递减, 模拟开关的导通电阻也相应地按 2 的整数倍递增。

由于在倒 T 形电阻网络 D/A 转换器中, 各支路电流直接流入运算放大器的输入端, 它们之间不存在传输上的时间差。电路的这一特点不仅提高了转换速

度,而且也减小了动态过程中输出端可能出现的尖脉冲。它是目前广泛使用的D/A转换器中速度较快的一种。常用的CMOS开关倒T形电阻网络D/A转换器的集成电路有AD7520(10位)、DAC1210(12位)及AK7546(16位高精度)等。

10.1.2 权电流型D/A转换器

尽管倒T形电阻网络D/A转换器具有较高的转换速度,但由于电路中存在模拟开关电压降,当流过各支路的电流稍有变化时,就会产生转换误差。为进一步提高D/A转换器的精度,可采用权电流型D/A转换器。4位权电流D/A转换器原理电路如图10.1.3所示。电路中,用一组恒流源代替了图10.1.2中倒T形电阻网络。这组恒流源从高位到低位电流的大小依次为 $I/2$ 、 $I/4$ 、 $I/8$ 、 $I/16$ 。

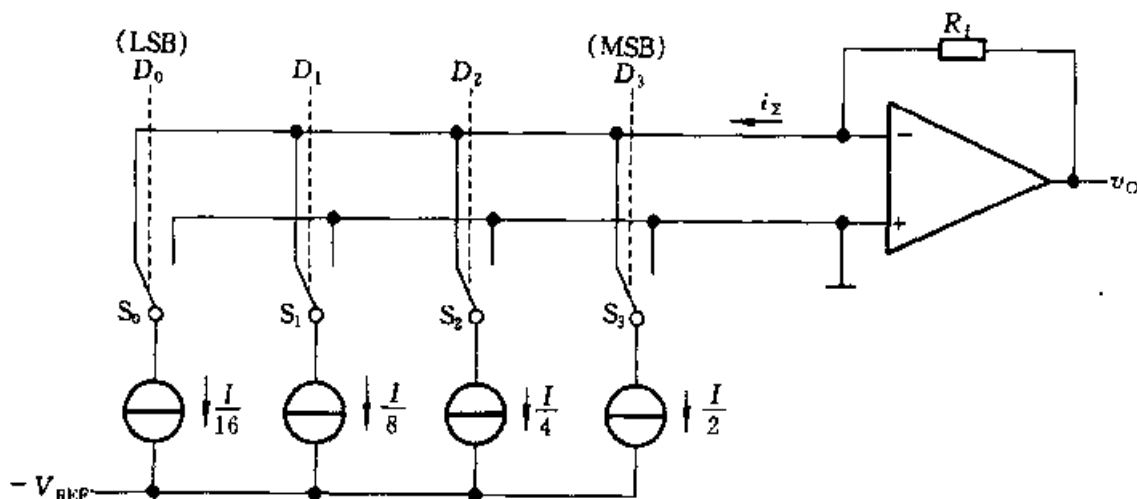


图 10.1.3 权电流 D/A 转换器的原理电路

在图 10.1.3 所示电路中,当输入数字量的某一位代码 $D_i = 1$ 时,开关 S_i 接运算放大器的反相端,相应权电流流入求和电路;当 $D_i = 0$ 时,开关 S_i 接地。分析该电路,可得出

$$\begin{aligned} v_o &= i_z R_f = R_f \left(\frac{I}{2} D_3 + \frac{I}{4} D_2 + \frac{I}{8} D_1 + \frac{I}{16} D_0 \right) \\ &= \frac{I}{2^4} \cdot R_f (D_3 \cdot 2^3 + D_2 \cdot 2^2 + D_1 \cdot 2^1 + D_0 \cdot 2^0) \quad (10.1.5) \\ &= \frac{I}{2^4} \cdot R_f \sum_{i=0}^3 D_i \cdot 2^i \end{aligned}$$

采用了恒流源电路后,各支路权电流的大小均不受开关导通电阻和压降的影响,这就降低了对开关电路的要求,提高了转换精度。

如将图 10.1.3 中所示恒流源采用具有电流负反馈的 BJT 恒流源电路,即可得如图 10.1.4 所示的实际的权电流 D/A 转换器电路。

可推得 n 位倒 T 形权电流 D/A 转换器的输出电压

$$v_O = \frac{V_{REF}}{R_1} \cdot \frac{R_f}{2^n} \sum_{i=0}^{n-1} D_i \cdot 2^i \quad (10.1.7)$$

式(10.1.6)表明,基准电流仅与基准电压 V_{REF} 和电阻 R_1 有关,而与 BJT、 R 、 $2R$ 电阻无关。这样,电路降低了对 BJT 参数及 R 、 $2R$ 取值的要求,对于集成化十分有利。

由于在这种权电流 D/A 转换器中采用了高速电子开关,电路还具有较高的转换速度。通常采用的单片集成权电流 D/A 转换器有 AD1408、DAC0806、DAC0808 等。

10.1.3 D/A 转换器的输出方式

常用的 D/A 转换器绝大部分是数字电流转换器,输出量是电流。如要实现电压输出,在实际应用时还需增加输出电路将电流转换成电压。使用 D/A 转换器,正确选择和设计输出电路是非常重要的,下面来讨论这方面的内容。

在前面介绍的 D/A 转换器中,输入的数字均视为正数,即二进制数的所有位都为数值位。根据电路形式或参考电压的极性不同,输出电压或为 0 V 到正满度值,或为 0 V 到负满度值,D/A 转换器处于单极性输出方式。采用单极性输出方式时,数字输入量采用自然二进制码,8 位 D/A 转换器单极性输出时,输入数字量与输出模拟量之间的关系如表 10.1.1 所示。

表 10.1.1 8 位 D/A 转换器在单极性输出时的输入/输出关系

数字量								模拟量
MSB							LSB	
1	1	1	1	1	1	1	1	$\pm V_{REF} \left(\frac{255}{256} \right)$
				⋮				⋮
1	0	0	0	0	0	0	1	$\pm V_{REF} \left(\frac{129}{256} \right)$
1	0	0	0	0	0	0	0	$\pm V_{REF} \left(\frac{128}{256} \right)$
0	1	1	1	1	1	1	1	$\pm V_{REF} \left(\frac{127}{256} \right)$
				⋮				⋮
0	0	0	0	0	0	0	1	$\pm V_{REF} \left(\frac{1}{256} \right)$
0	0	0	0	0	0	0	0	$\pm V_{REF} \left(\frac{0}{256} \right)$

倒 T 形电阻网络 D/A 转换器单极性电压输出的电路分别如图 10.1.5a、b 所示。其中图 a 为单极性反相电压输出电路，

$$v_O = -i_{\Sigma} R_f \quad (10.1.8)$$

图 b 为同相电压输出电路，此时

$$v_O = i_{\Sigma} R (1 + R_2/R_1) \quad (10.1.9)$$

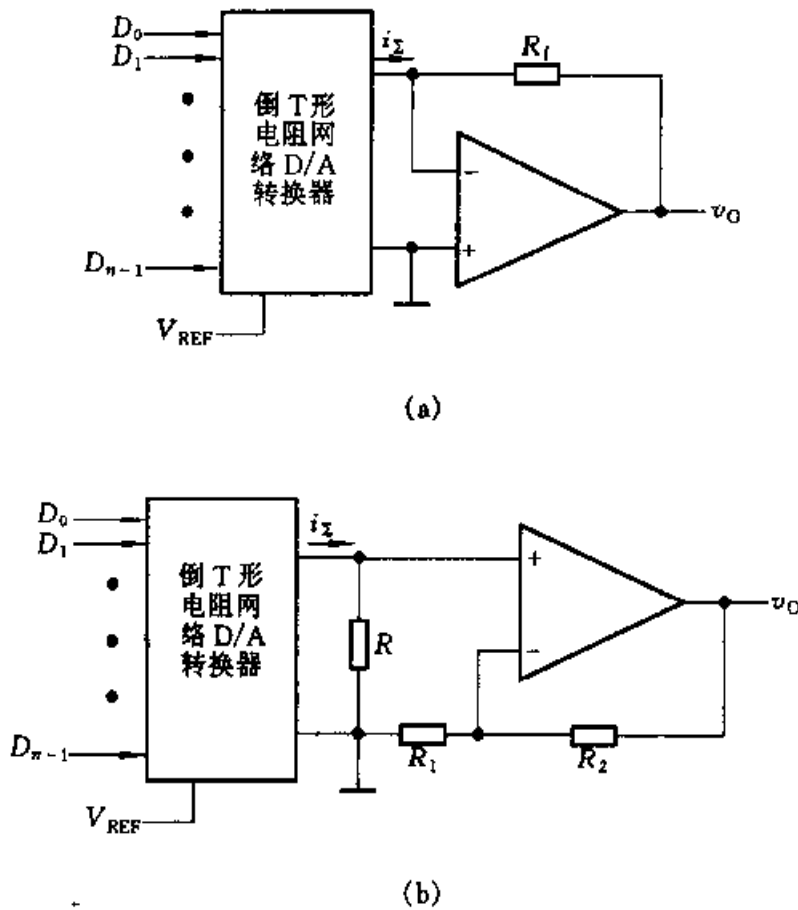


图 10.1.5 D/A 转换器的单极性电压输出

(a) 反相输出 (b) 同相输出

在实际应用中, D/A 转换器输入的数字量有正极性也有负极性。这就要求 D/A 转换器能将不同极性的数字量对应转换为正、负极性的模拟电压, 工作于双极性方式。双极性 D/A 转换常用的编码有: 2 的补码、偏移二进制码及符号-数值码(符号位加数值码)等。表 10.1.2 列出了 8 位 2 的补码、偏移二进制码及模拟量之间的对应关系。

由表 10.1.2 可见, 偏移二进制码与无符号二进制码形式相同, 它实际上是将二进制码对应的模拟量的零值偏移至 80H, 使偏移后的数中, 只有大于 128 的才是正数, 而小于 128 的则为负数。所以, 若将单极性 8 位 D/A 转换器的输出电压减去 $\frac{V_{REF}}{2}$ (80H 所对应的模拟量), 就可得到极性正确的偏移二进制码输出

电压。

表 10.1.2 常用双极性输出模拟量

十进制数	2 的补码								偏移二进制码								模拟量 v_O/V_{1LSB}
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
127	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	127
126	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	126
				⋮								⋮					⋮
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
-1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	-1
				⋮								⋮					⋮
-127	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	-127
-128	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-128

* 表中 $V_{1LSB} = V_{REF}/256$ 。

若 D/A 转换器输入数字量是 2 的补码,那么,需先将它转换为偏移二进制码,然后输入到上述 D/A 转换电路中就可实现其双极性输出。比较表 10.1.2 中 2 的补码与偏移二进制码可以发现,若将 8 位 2 的补码加 80H,并舍弃进位就可得偏移二进制码。实现 2 的补码加 80H 很简单,只需将高位求反即可。这样,可得到采用 2 的补码输入的 8 位双极性输出 D/A 转换电路,如图 10.1.6 所示。

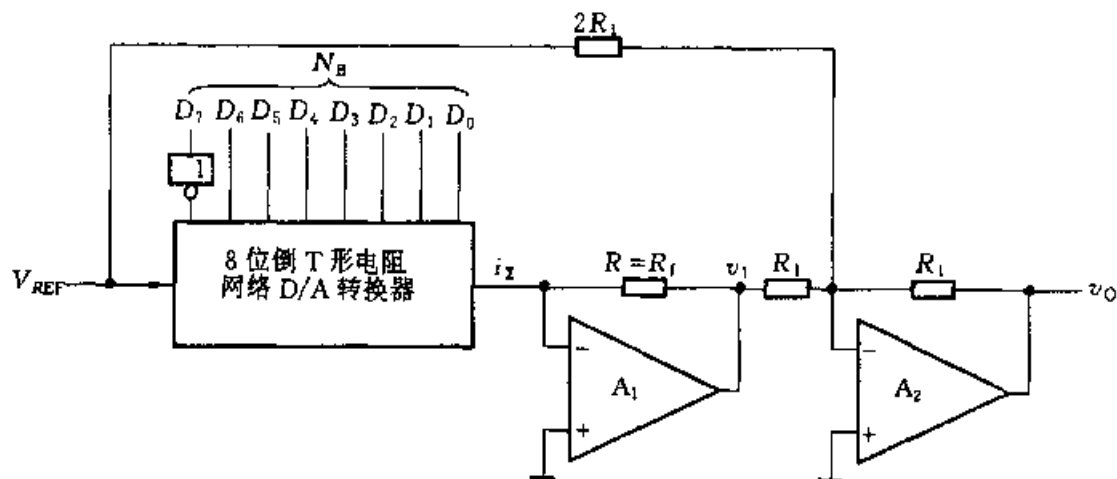


图 10.1.6 双极性输出 D/A 转换器

图中,输入 N_B 是原码的 2 的补码,最高位取反(加 80H)变为偏移二进制码后送入 D/A 转换器,由 D/A 转换器输出的模拟量 v_1 经 A_2 组成的第二个求和放大器减去 $V_{REF}/2$ 后,得到极性正确的输出电压 v_O ,即

$$v_O = -v_1 - \frac{1}{2} V_{REF} = - \left(- \frac{N_B V_{REF}}{2^8} - \frac{V_{REF}}{2} \right) - \frac{V_{REF}}{2}$$

$$= V_{\text{REF}} \cdot \frac{N_B}{256} \quad (10.1.10)$$

电路输入 2 的补码 N_B 与 v_O 满足表 10.1.2 所示的对应关系。

10.1.4 D/A 转换器的主要技术指标

D/A 转换器的主要技术指标有转换精度、转换速度和温度特性等。

1. 转换精度

D/A 转换器的转换精度通常用分辨率和转换误差来描述。

分辨率用于表征 D/A 转换器对输入微小量变化敏感程度的。其定义为 D/A 转换器模拟输出电压可能被分离的等级数。输入数字量位数愈多,输出电压可分离的等级愈多,即分辨率愈高。所以在实际应用中,往往用输入数字量的位数表示 D/A 转换器的分辨率。此外,D/A 转换器也可以用能分辨最小输出电压与最大输出电压之比给出。 n 位 D/A 转换器的分辨率可表示为 $\frac{1}{2^n - 1}$ 。它表示 D/A 转换器在理论上可以达到的精度。

由于 D/A 转换器中各元件参数值存在误差,基准电压不够稳定和运算放大器的零漂等各种因素的影响,使得 D/A 转换器实际精度还与一些转换误差有关,如比例系数误差、失调误差和非线性误差等。

比例系数误差是指实际转换特性曲线的斜率与理想特性曲线斜率的偏差。如在 n 位倒 T 形电阻网络 D/A 转换器中,当 V_{REF} 偏离标准值 ΔV_{REF} 时,就会在输出端产生误差电压 Δv_O 。由式(10.1.3)可知

$$\Delta v_O = \frac{\Delta V_{\text{REF}}}{2^n} \cdot \frac{R_f}{R} \sum_{i=0}^{n-1} D_i \cdot 2^i \quad (10.1.11)$$

由 ΔV_{REF} 引起的误差属于比例系数误差。3 位 D/A 转换器的比例系数误差如图 10.1.7 所示。

失调误差由运算放大器的零点漂移引起,其大小与输入数字量无关,该误差使输出电压的转移特性曲线发生平移,3 位 D/A 转换器的失调误差如图 10.1.8 所示。

非线性误差是一种没有一定变化规律的误差,一般用在满刻度范围内,偏离理想的转移特性的最大值来表示。引起非线性误差的原因较多,如电路中的各模拟开关不仅存在不同的导通电压和导通电阻,而且每个开关处于不同位置(接地或接 V_{REF})时,其开关压降和电阻也不一定相等。又如,在电阻网络中,每个支路上电阻误差不相同,不同位置上的电阻的误差对输出电压的影响也不相同,这些都会导致非线性误差。

综上所述,为获得高精度的 D/A 转换器,不仅应选择位数较多的高分辨率

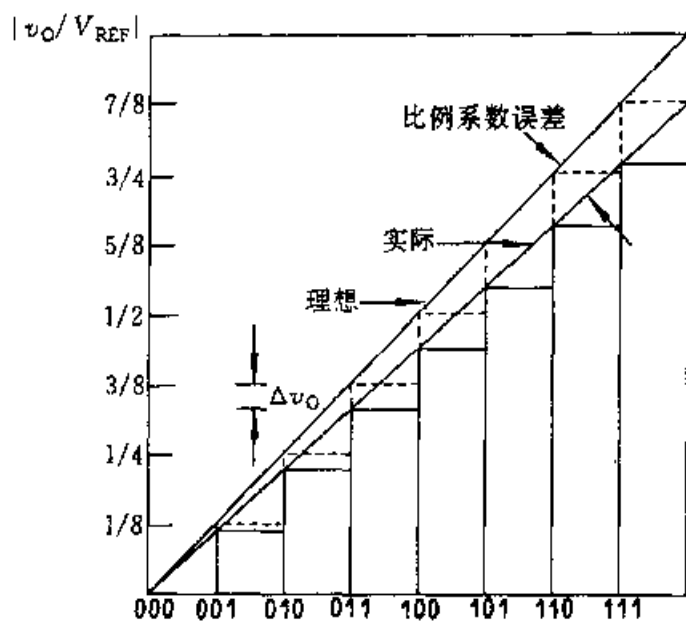


图 10.1.7 3 位 D/A 转换器的比例系数误差

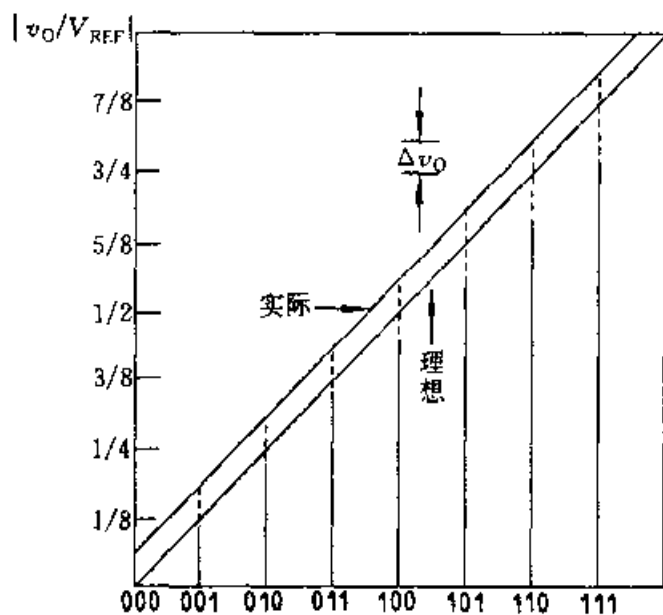


图 10.1.8 3 位 D/A 转换器的失调误差

的 D/A 转换器,而且还需要选用高稳定度的 V_{REF} 和低零漂的运算放大器等器件与之配合才能达到要求。

2. 转换速度

当 D/A 转换器输入的数字量发生变化时,输出的模拟量并不能立即达到所对应的量值,它需要一段时间。通常用建立时间和转换速率两个参数来描述 D/A 转换器的转换速度。

建立时间(t_{set})指输入数字量变化时,输出电压变化到相应稳定电压值所需时间。一般用 D/A 转换器输入的数字量 N_B 从全 0 变为全 1 时,输出电压达到规定的误差范围($\pm \text{LSB}/2$)时所需时间表示。D/A 转换器的建立时间较快,单片集成 D/A 转换器建立时间最短可达 $0.1 \mu\text{s}$ 以内。

转换速率(SR)用大信号工作状态下,模拟电压的变化率表示。一般集成 D/A 转换器在不包含外接参考电压源和运算放大器时,转换速率比较高。实际应用中,要实现快速 D/A 转换不仅要求 D/A 转换器有较高的转换速率,而且还应选用转换速率较高的集成运算放大器与之配合使用才行。

3. 温度系数

是指在输入不变的情况下,输出模拟电压随温度变化产生的变化量。一般用满刻度输出条件下温度每升高 1°C ,输出电压变化的百分数作为温度系数。

10.1.5 集成 D/A 转换器及其应用

单片集成 D/A 转换器产品的种类繁多,性能指标各异,按其内部电路结构不同一般分为两类:一类集成芯片内部只集成了电阻网络(或恒流源网络)和模拟电子开关,另一类则集成了组成 D/A 转换器的全部电路。集成 D/A 转换器 AD7520 属于前一类,下面以它为例介绍集成 D/A 转换器结构及其应用。

1. AD7520 D/A 转换器

AD7520 是 10 位 CMOS 电流开关型 D/A 转换器,其结构简单,通用性好。AD7520 芯片内只含倒 T 形电阻网络、CMOS 电流开关和反馈电阻($R = 10 \text{ k}\Omega$),该集成 D/A 转换器在应用时必须外接参考电压源和运算放大器。由 AD7520 采用内部反馈电阻组成的 D/A 转换电路如图 10.1.9 所示,图中虚线内部分为 AD7520 内部电路。AD7520 芯片引线排列图如图 10.1.10 所示。

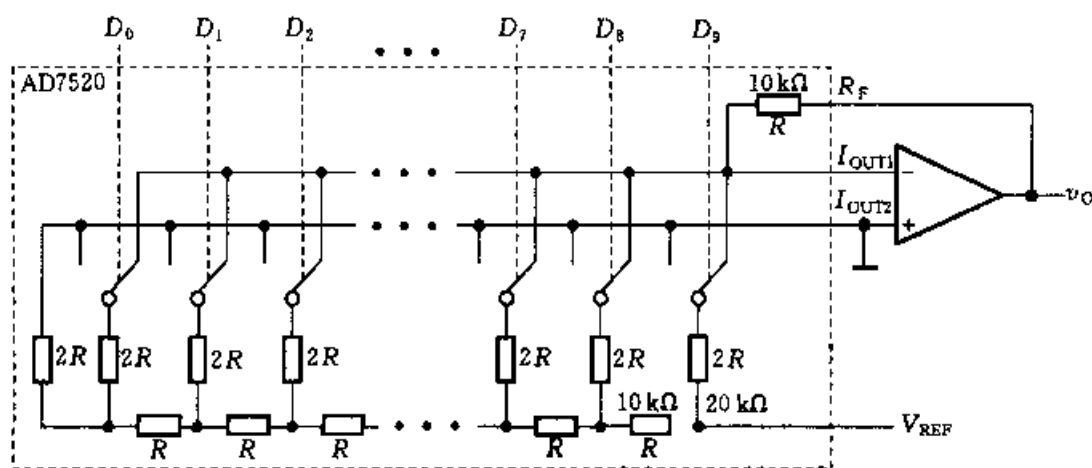


图 10.1.9 AD7520 内部电路

图 10.1.9 中每个电子开关的实际电路如图 10.1.11 所示。它是由 9 个 MOS 管组成的 CMOS 模拟开关电路。图中 $T_1 \sim T_3$ 组成电平转移电路,使输入信号能与 TTL 电平兼容。 T_4 、 T_5 及 T_6 、 T_7 组成两个反相器,分别作为模拟开关管 T_8 、 T_9 的驱动电路, T_8 、 T_9 构成单刀双掷开关。

当 $D_i = 1$ 时, T_1 输出低电平, T_4 、 T_5 反相器输出高电平,而 T_6 、 T_7 反相器输出低电平,从而使 T_8 截止、 T_9 导通, $2R$ 电阻经 T_9 接至运算放大器的反相输入端, 电流流入运算放大器。

当 $D_i = 0$ 时, T_1 输出高电平, T_4 、 T_5 反相器输出的

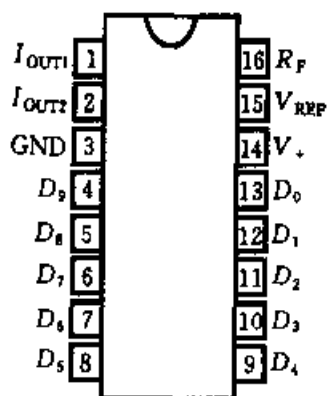


图 10.1.10 AD7520
引脚图

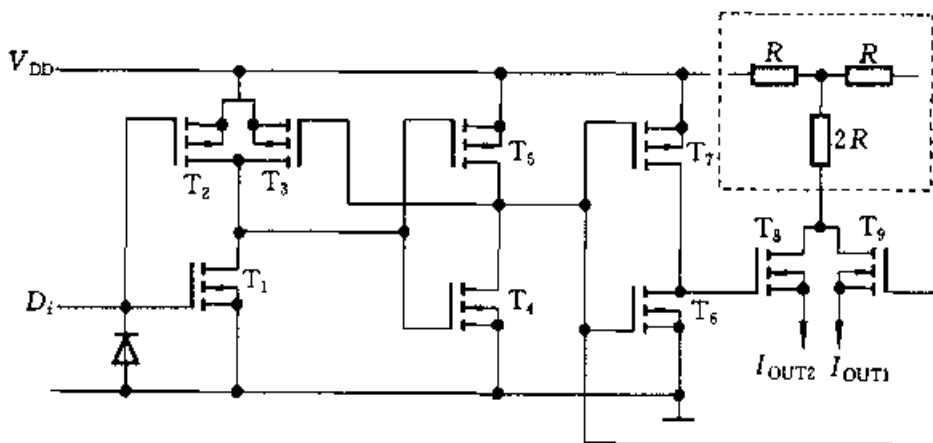


图 10.1.11 CMOS 模拟开关电路

低电平使 T_9 截止, T_6 、 T_7 反相器输出的高电平使 T_8 导通, 这样 $2R$ 电阻经 T_8 接地。CMOS 模拟开关导通电阻较大, 通过工艺设计可控制其大小并计入电阻网络。该电路具有使用简便, 功耗低, 转换速度较快, 温度系数小, 通用性强等优点。

2. 集成 D/A 转换器应用举例

D/A 转换器在实际电路中应用很广, 它不仅常作为接口电路用于微机系统, 而且还可利用其电路结构特征和输入、输出电量之间的关系构成数控电流源、电压源, 数字式可编程增益控制电路和波形产生电路等。下面以数字式可编程增益控制电路和波形产生电路为例说明它的应用。

(1) 数字式可编程增益控制电路

数字式可编程增益控制电路如图 10.1.12 所示。电路中运算放大器接成普通的反比例放大形式, AD7520 内部的反馈电阻 R 为运算放大器的输入电阻,

而由数字量控制的倒 T 形电阻网络为其反馈电阻。当输入数字量变化时,倒 T 形电阻网络的等效电阻便随之改变。这样,反比例放大器在其输入电阻一定的情况便可得到不同的增益。

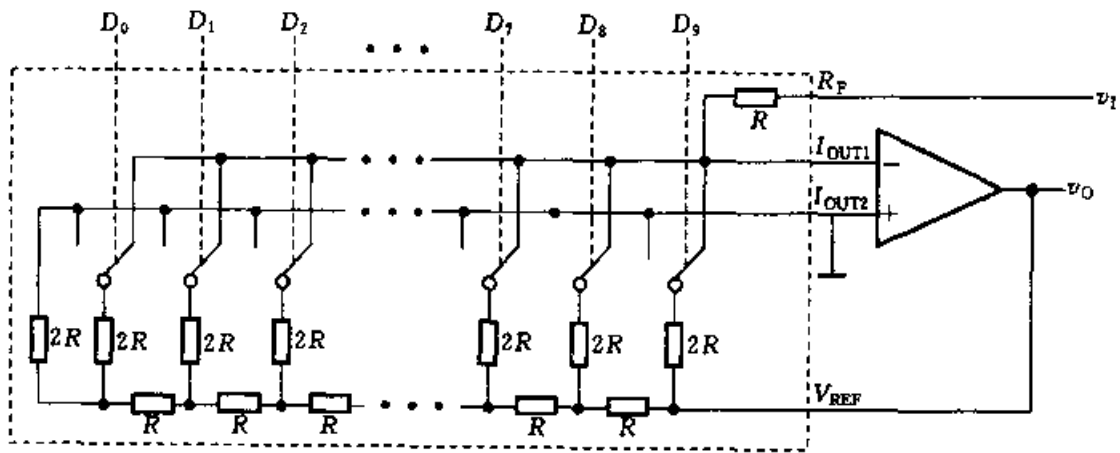


图 10.1.12 数字式可编程增益控制电路

根据运算放大器虚地原理,可以得到

$$\frac{v_1}{R} = \frac{-v_0}{2^{10} \cdot R} (D_0 \cdot 2^0 + D_1 \cdot 2^1 + \dots + D_9 \cdot 2^9)$$

所以

$$A_v = \frac{v_0}{v_1} = \frac{-2^{10}}{D_0 \cdot 2^0 + D_1 \cdot 2^1 + \dots + D_9 \cdot 2^9}$$

如将 AD7520 芯片中的反馈电阻 R 作为反相运算放大器的反馈电阻,数控 AD7520 的倒 T 形电阻网络连接成运算放大器的输入电阻,读者不难推断出电路为数字式可编程衰减器。

(2) 脉冲波产生电路

由 D/A 转换器 AD7520、10 位可逆计数器及加减控制电路组成的波形产生电路如图 10.1.13 所示。加/减控制电路与 10 位二进制可逆计数器配合工作,当计数器加到全“1”时,加/减控制电路复位使计数器进入减法计数状态,而当减到全“0”时,加/减控制电路置位,使计数器再次处于加法计数状态,如此周而复始。根据式(10.1.3),可得 D/A 转换器(I)的输出电压为

$$v_{O1} = -\frac{V_{REF}}{2^{10}} \cdot \sum_{i=0}^9 D_i \cdot 2^i$$

可以看出, v_{O1} 是一个近似的三角波。

将这个三角波作为 D/A 转换器(II)的参考电压,由于两个 D/A 转换器数字量相同,于是可得第二级 D/A 转换器输出的模拟电压为

$$v_{(n)} = V_{REF} \left(\frac{\sum_{i=1}^9 D_i \cdot 2^i}{2^{10}} \right)^2$$

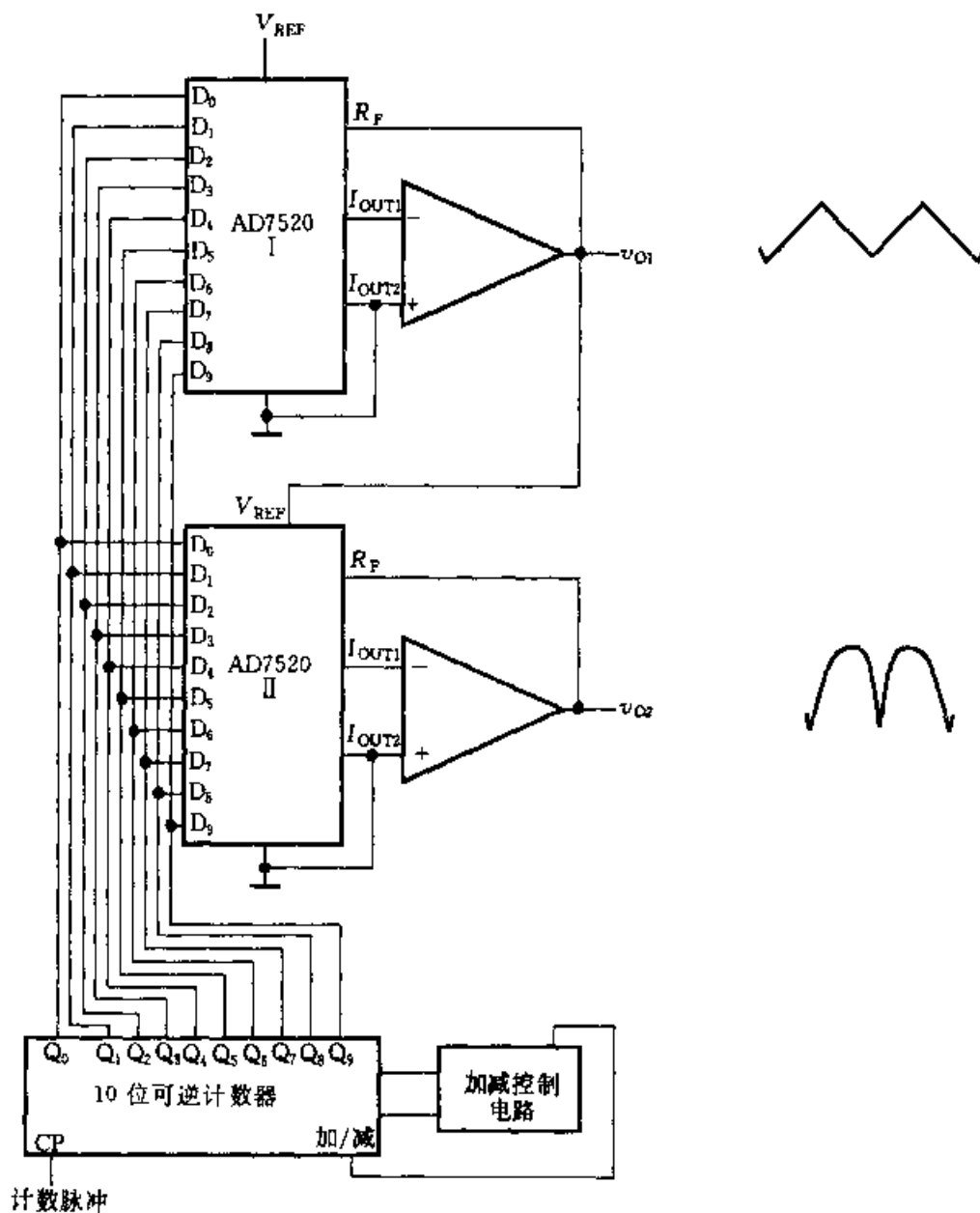


图 10.1.13 AD7520 组成的波形产生电路

显然,这是一个抛物波。

复习思考题

10.1.1 D/A转换器有哪几种基本类型? 倒T形电阻网络D/A转换器和权电流型D/A转换器各有什么特点?

10.1.2 如要求电路的功耗小,你应选择哪种 D/A 转换器?如要求转换速度快呢?

10.1.3 为使倒 T 形电阻网络 D/A 转换器有足够的转换精度,在电路器件及参数选择上应有什么基本要求?

10.1.4 如 D/A 转换器输出电压的误差电压 Δv_o 与输入数字量无关,在温度一定时为恒定值,这种误差属于什么误差?引起此误差的原因是什么?

10.1.5 已知某 D/A 转换器满刻度输出电压为 10 V,试问要求 1 mV 的分辨率,其输入数字量的位数 n 至少是多少?

10.2 A/D 转换器

为将时间连续、幅值也连续的模拟量转换为时间离散、幅值也离散的数字信号,A/D 转换一般要经过取样、保持、量化及编码 4 个过程。在实际电路中,这些过程有的是合并进行的,例如,取样和保持,量化和编码往往都是在转换过程中同时实现。

10.2.1 A/D 转换的一般工作过程

1. 取样与保持

取样是将随时间连续变化的模拟量转换为时间离散的模拟量。取样过程示意图如图 10.2.1 所示。图 a 中,传输门受取样信号 $S(t)$ 控制,在 $S(t)$ 的脉宽 τ 期间,传输门导通,输出信号 $v_o(t)$ 为输入信号 $v_i(t)$,而在 $(T_s - \tau)$ 期间,传输门关闭,输出信号 $v_o(t) = 0$ 。电路中各信号波形如图 b 所示。

通过分析可以看出,取样信号 $S(t)$ 的频率愈高,所取得信号经低通滤波器后愈能真实地复现输入信号。合理的取样频率由取样定理确定。

取样定理:设取样信号 $S(t)$ 的频率为 f_s ,输入模拟信号 $v_i(t)$ 的最高频率分量的频率为 f_{imax} ,则 f_s 与 f_{imax} 必须满足下面的关系

$$f_s \geq 2f_{\text{imax}} \quad (10.2.1)$$

一般取 $f_s > 2f_{\text{imax}}$ 。

将取样电路每次取得的模拟信号转换为数字信号都需要一定时间,为了给后续的量化编码过程提供一个稳定值,每次取得的模拟信号必须通过保持电路保持一段时间。

取样与保持过程往往是通过取样-保持电路同时完成的。取样-保持电路的原理图及输出波形如图 10.2.2 所示。

电路由输入放大器 A_1 、输出放大器 A_2 、保持电容 C_H 和开关驱动电路组成。电路中要求 A_1 具有很高的输入阻抗,以减小对输入信号源的影响。为使保持阶段 C_H 上所存电荷不易泄放, A_2 也应具有较高输入阻抗, A_2 还应具有低的输

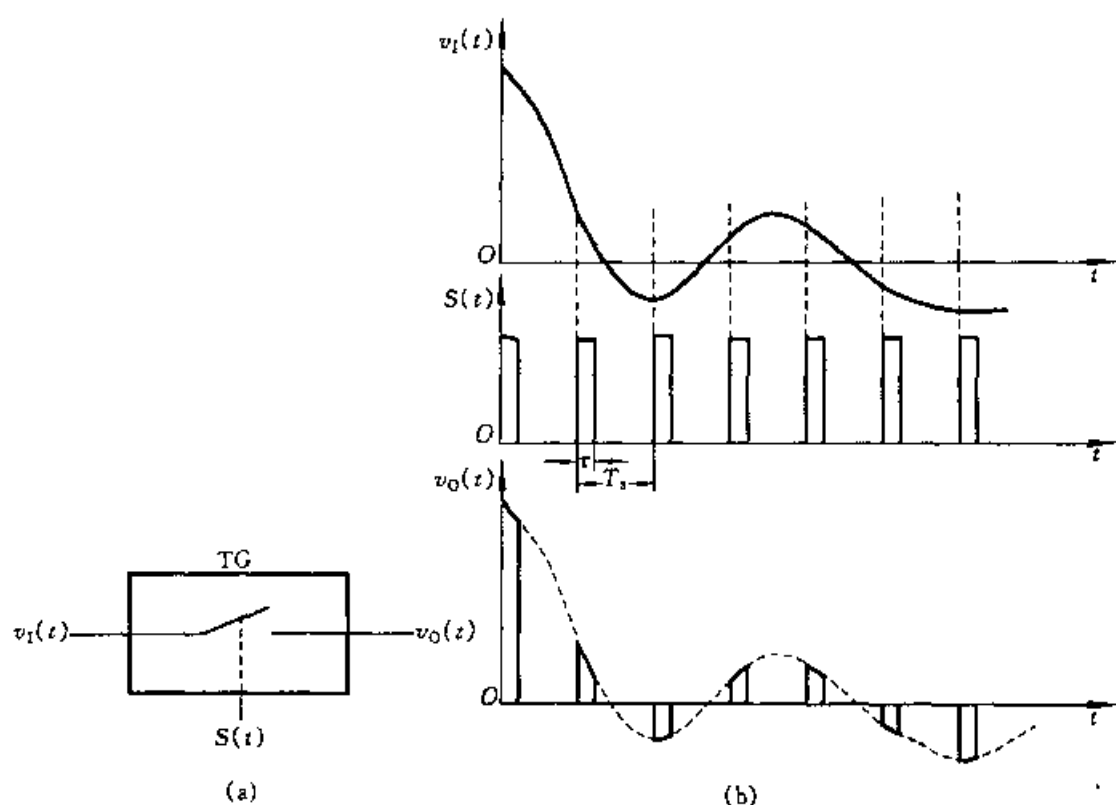


图 10.2.1 取样过程

出阻抗,这样可以提高电路的带负载能力。一般还要求电路中 $A_{V1} \cdot A_{V2} = 1$ 。

现结合图 10.2.2 来分析取样-保持电路的工作原理。在 $t = t_0$ 时,开关 S 闭合,电容被迅速充电,由于 $A_{V1} \cdot A_{V2} = 1$,因此 $v_0 = v_1$,在 $t_0 \sim t_1$ 时间间隔内是取样阶段。当 $t = t_1$ 时刻 S 断开。若 A_2 的输入阻抗为无穷大、S 为理想开关,这样就可认为电容 C_H 没有放电回路,其两端电压保持为 v_0 不变,图 10.2.2b 中 t_1 到 t_2 的平坦段,就是保持阶段。

取样-保持电路已有多种型号的单片集成电路产品。如双极型工艺的有 AD585、AD684;混合型工艺的有 AD1154、SHC76 等。

2. 量化与编码

数字信号不仅在时间上是离散的,而且在幅值上也是不连续的。任何一个数字量的大小只能是某个规定的最小数量单位的整数倍。为将模拟信号转换为数字量,在 A/D 转换过程中,还必须将取样-保持电路的输出电压,按某种近似方式归化到与之相应的离散电平上。这一转化过程称为数值量化,简称量化。量化后的数值最后还须通过编码过程用一个代码表示出来。经编码后得到的代码就是 A/D 转换器输出的数字量。

量化过程中所取最小数量单位称为量化单位,用 Δ 表示。它是数字信号最低位为 1 时所对应的模拟量,即 1LSB。

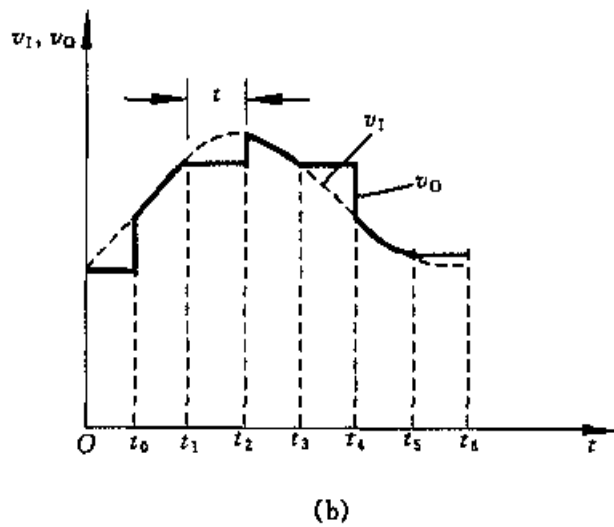
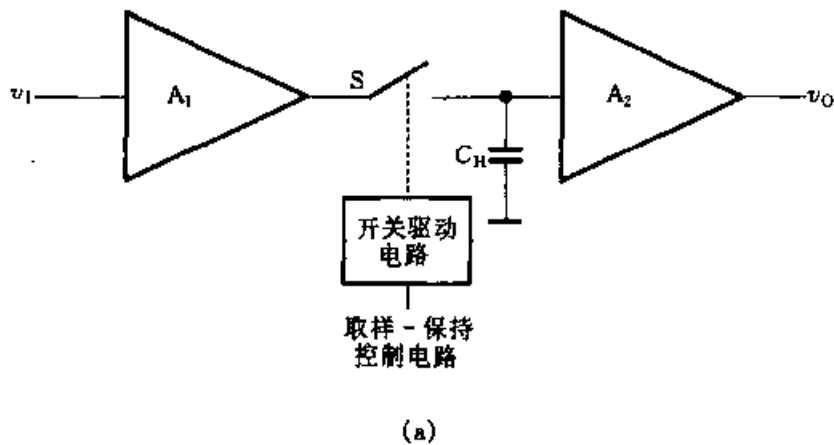


图 10.2.2 取样-保持电路

(a) 原理图 (b) 波形图

在量化过程中,由于取样电压不一定能被 Δ 整除,所以量化前后不可避免地存在误差,此误差称之为**量化误差**,用 ϵ 表示。量化误差属原理误差,它是无法消除的。A/D 转换器的位数越多,各离散电平之间的差值越小,量化误差越小。

量化过程常采用两种近似量化方式:只舍不入量化方式和四舍五入的量化方式。以 3 位 A/D 转换器为例,设输入信号 v_1 的变化范围为 $0 \sim 8 \text{ V}$,采用只舍不入量化方式时,取 $\Delta = 1 \text{ V}$,量化中把不足量化单位部分舍弃,如数值在 $0 \sim 1 \text{ V}$ 之间的模拟电压都当作 0Δ ,用二进制数 **000** 表示,而数值在 $1 \sim 2 \text{ V}$ 之间的模拟电压都当作 1Δ ,用二进制数 **001** 表示……。这种量化方式的最大量化误差为 Δ ;如采用四舍五入量化方式,则取量化单位 $\Delta = 8 \text{ V}/15$,量化过程将不足半个量化单位部分舍弃,对于等于或大于半个量化单位部分按一个量化单位处理。它将数值在 $0 \sim 4 \text{ V}/15$ 之间的模拟电压都当作 0Δ 对待,用二进制数 **000** 表示,而数值在 $4 \text{ V}/15 \sim 12 \text{ V}/15$ 之间的模拟电压均当作 1Δ ,用二进制数 **001** 表示等。不难看出,采用前一种只舍不入量化方式最大量化误差 $|\epsilon_{\max}| \approx 1\text{LSB}$,而采用后一种有

舍有入量化方式 $\epsilon_{\max} = \text{LSB}/2$, 后者量化误差比前者小, 故为大多数 A/D 转换器所采用。

A/D 转换器的种类很多, 按其工作原理不同分为直接 A/D 转换器和间接 A/D 转换器两类。直接 A/D 转换器可将模拟信号直接转换为数字信号, 这类 A/D 转换器具有较快的转换速度, 其典型电路有并行比较型 A/D 转换器、逐次比较型 A/D 转换器。而间接 A/D 转换器则是先将模拟信号转换成某一中间量(时间或频率), 然后再将中间量转换为数字量输出。此类 A/D 转换器的速度较慢, 典型电路是双积分型 A/D 转换器、电压频率转换型 A/D 转换器。下面将详细介绍这几种 A/D 转换器的电路结构及工作原理。

10.2.2 并行比较型 A/D 转换器

3 位并行比较型 A/D 转换器原理电路如图 10.2.3 所示。它由电阻分压器、电压比较器、寄存器及编码器组成。图中的 8 个电阻将参考电压 V_{REF} 分成 8 个等级, 其中 7 个等级的电压分别作为 7 个比较器 $C_1 \sim C_7$ 的参考电压, 其数值分别为 $V_{\text{REF}}/15, 3V_{\text{REF}}/15, \dots, 13V_{\text{REF}}/15$ 。输入电压为 v_i , 它的大小决定各比较器的输出状态, 例如, 当 $0 \leq v_i < V_{\text{REF}}/15$ 时, $C_7 \sim C_1$ 的输出状态都为 0; 当 $3V_{\text{REF}}/15 \leq v_i < 5V_{\text{REF}}/15$ 时, 比较器 C_6 和 C_7 的输出 $C_{06} = C_{07} = 1$, 其余各比较器的状态均为 0。根据各比较器的参考电压值, 可以确定输入模拟电压值与各比较器输出状态的关系。比较器的输出状态由 D 触发器存储, 经优先编码器编码, 得到数字量输出。优先编码器优先级别最高是 I_7 , 最低的是 I_1 。

设 v_i 变化范围是 $0 \sim V_{\text{REF}}$, 输出 3 位数字量为 $D_2 D_1 D_0$, 3 位并行比较型 A/D 转换器的输入、输出关系如表 10.2.1 所示。

表 10.2.1 3 位并行 A/D 转换器输入与输出关系对照表

模拟输入	比较器输出状态							数字输出		
	C_{01}	C_{02}	C_{03}	C_{04}	C_{05}	C_{06}	C_{07}	D_2	D_1	D_0
$0 \leq v_i < V_{\text{REF}}/15$	0	0	0	0	0	0	0	0	0	0
$V_{\text{REF}}/15 \leq v_i < 3V_{\text{REF}}/15$	0	0	0	0	0	0	1	0	0	1
$3V_{\text{REF}}/15 \leq v_i < 5V_{\text{REF}}/15$	0	0	0	0	0	1	1	0	1	0
$5V_{\text{REF}}/15 \leq v_i < 7V_{\text{REF}}/15$	0	0	0	0	1	1	1	0	1	1
$7V_{\text{REF}}/15 \leq v_i < 9V_{\text{REF}}/15$	0	0	0	1	1	1	1	1	0	0
$9V_{\text{REF}}/15 \leq v_i < 11V_{\text{REF}}/15$	0	0	1	1	1	1	1	1	0	1
$11V_{\text{REF}}/15 \leq v_i < 13V_{\text{REF}}/15$	0	1	1	1	1	1	1	1	1	0
$13V_{\text{REF}}/15 \leq v_i < V_{\text{REF}}$	1	1	1	1	1	1	1	1	1	1

在并行 A/D 转换器中, 输入电压 v_i 同时加到所有比较器的输入端, 从 v_i 加入到 3 位数字量稳定输出所经历的时间为比较器、D 触发器和编码器延迟时

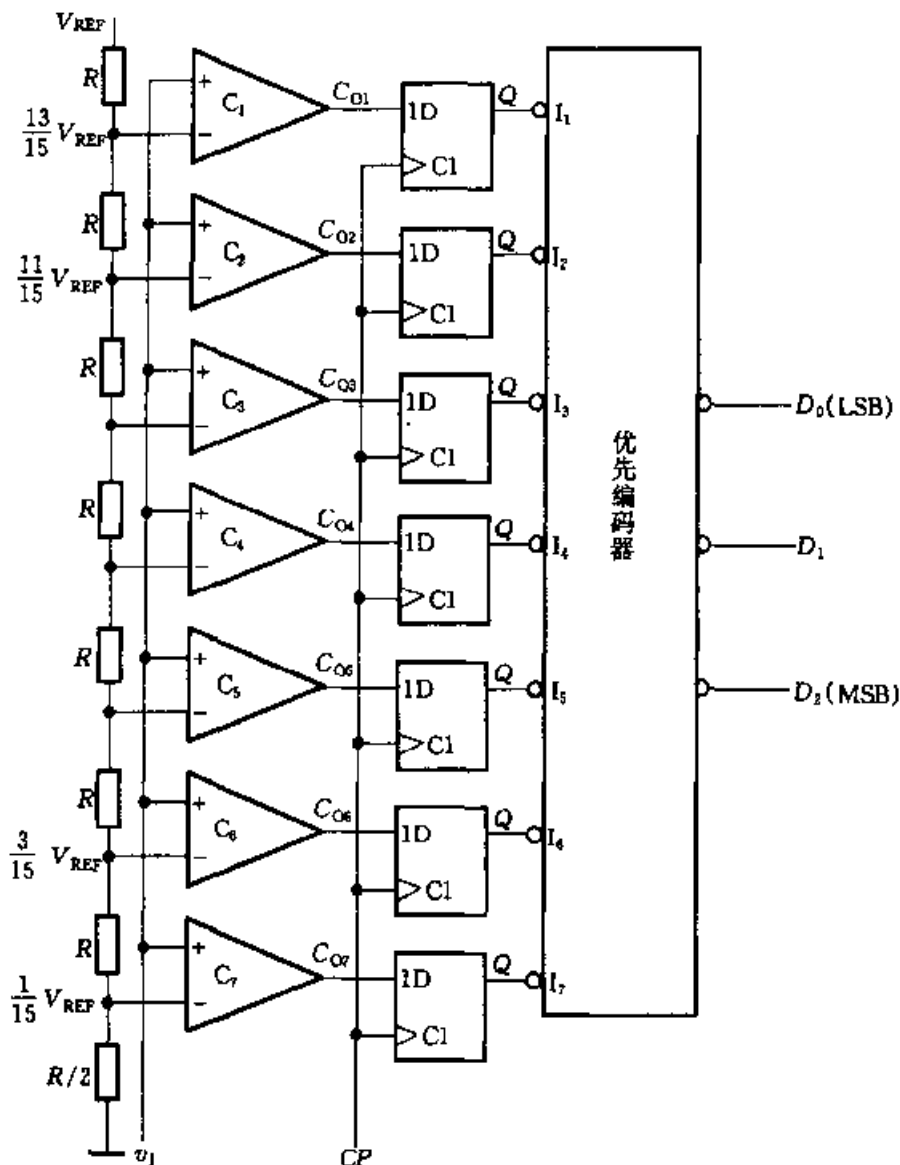


图 10.2.3 3 位并行 A/D 转换器

间之和。如不考虑上述器件的延迟,可认为 3 位数字量是与 v_i 输入时刻同时获得的。所以它具有最短的转换时间。

单片集成并行比较型 A/D 转换器的产品很多,如 AD 公司的 AD9012(TTL 工艺,8 位)、AD9002(ECL 工艺,8 位)、AD9020(TTL 工艺,10 位)等。

并行 A/D 转换器具有如下的特点:

(1) 由于转换是并行的,其转换时间只受比较器、触发器和编码电路延迟时间的限制,因此转换速度最快。

(2) 随着分辨率的提高,元件数目要按几何级数增加。一个 n 位转换器,所用比较器的个数为 $2^n - 1$,如 8 位的并行 A/D 转换器就需要 $2^8 - 1 = 255$ 个比较器。由于位数愈多,电路愈复杂,因此制成分辨率较高的集成并行 A/D 转换器

是比较困难的。

(3) 为了解决提高分辨率和增加元件数的矛盾,可以采取分级并行转换的方法。10位分级并行A/D转换原理图如图10.2.4所示。图中输入模拟信号 v_i ,经取样-保持电路后分两路,一路先经第一级5位并行A/D转换进行粗转换得到输出数字量的高5位,另一路送至减法器,与高5位D/A转换得到的模拟电压相减。由于相减所得差值电压小于 $1V_{LSB}$,为保证第二级A/D转换器的转换精度,将差值放大 $2^5 = 32$ 倍,送第二级5位并行比较A/D转换器,得到低5位输出。这种方法虽然在速度上作了牺牲,却使元件数大为减少,在需要兼顾分辨率和速度的情况下常被采用。

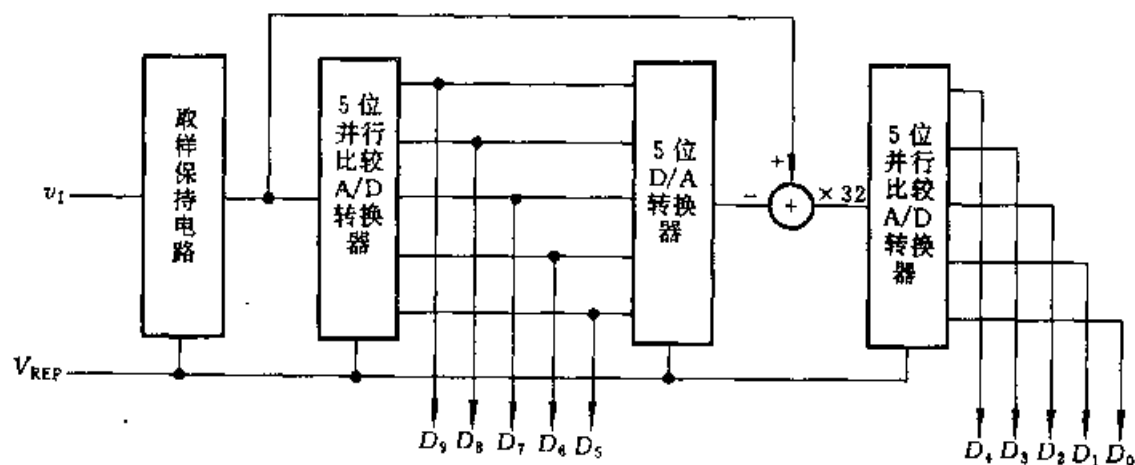


图 10.2.4 分级并行转换 10 位 A/D 转换器

10.2.3 逐次比较型 A/D 转换器

1. 转换原理

在直接 A/D 转换器中,逐次比较型 A/D 转换器是目前采用最多的一种。逐次逼近转换过程与用天平称物重非常相似。天平称重过程是,从最重的砝码开始试放,与被称物体行进比较,若物体重于砝码,则该砝码保留,否则移去。再加上第二个次重砝码,由物体的重量是否大于砝码的重量决定第二个砝码是留下还是移去。照此一直加到最小一个砝码为止。将所有留下的砝码重量相加,就得物体重量。仿照这一思路,逐次比较型 A/D 转换器,就是将输入模拟信号与不同的参考电压做多次比较,使转换所得的数字量在数值上逐次逼近输入模拟量对应值。

n 位逐次比较型 A/D 转换器框图如图 10.2.5 所示。它由控制逻辑电路、数据寄存器、移位寄存器、D/A 转换器及电压比较器组成,其工作原理如下:电路由启动脉冲启动后,在第一个时钟脉冲作用下,控制电路使移位寄存器的最高位

置1,其他位置0,其输出经数据寄存器将1000...0,送入D/A转换器。输入电压首先与D/A转换器输出电压($V_{REF}/2$)相比较,如 $v_i \geq V_{REF}/2$,比较器输出为1,若 $v_i < V_{REF}/2$,则为0。比较结果存于数据寄存器的 D_{n-1} 位。然后在第二个CP作用下,移位寄存器的次高位置1,其他低位置0。如最高位已存1,则此时 $v_o' = (3/4)V_{REF}$ 。于是 v_i 再与 $(3/4)V_{REF}$ 相比较,如 $v_i \geq (3/4)V_{REF}$,则次高位 D_{n-2} 存1,否则 $D_{n-2} = 0$;如最高位为0,则 $v_o' = V_{REF}/4$, v_i 与 v_o' 比较,如 $v_i \geq V_{REF}/4$,则 D_{n-2} 位存1,否则存0……。依此类推,逐次比较得到输出数字量。

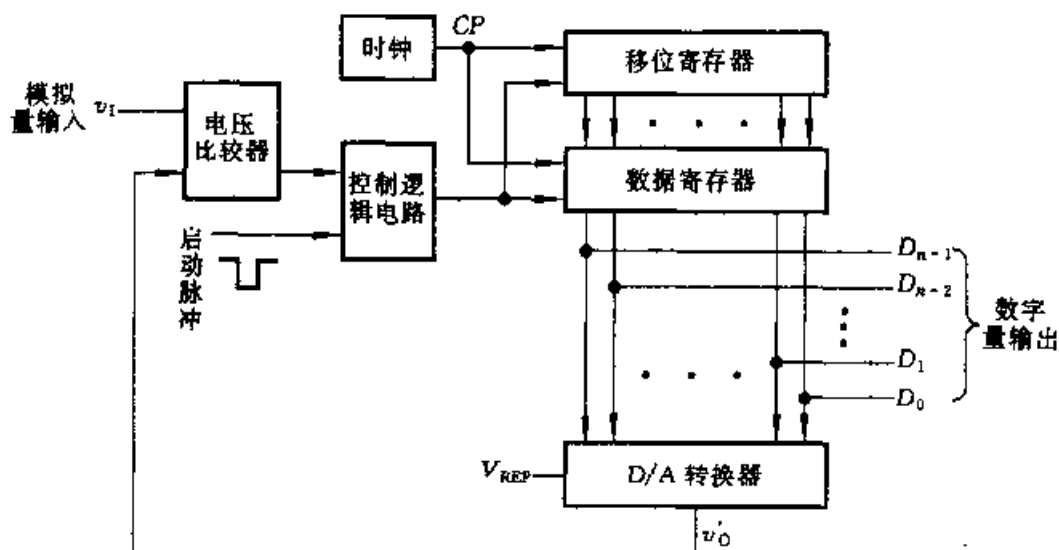


图 10.2.5 逐次比较型 A/D 转换器框图

为进一步理解逐次比较 A/D 转换器的工作原理及转换过程,下面用实例加以说明。

设图 10.2.5 电路为 8 位 A/D 转换器,输入模拟量 $v_i = v_A = 6.84 \text{ V}$,D/A 转换器基准电压 $V_{REF} = -10 \text{ V}$ 。

根据逐次比较 D/A 转换器的工作原理,可画出在转换过程中 CP、启动脉冲、 $D_7 \sim D_0$ 及 D/A 转换器输出电压 v_o' 的波形,如图 10.2.6 所示。

由图 10.2.6 可见,当启动脉冲低电平到来后转换开始。在第一个 CP 作用下,数据寄存器将 $D_7 \sim D_0 = 10000000$ 送入 D/A 转换器,其输出电压 $v_o' = 5 \text{ V}$, v_A 与 v_o' 比较, $v_A > v_o'$, D_7 存 1;第二个 CP 到来时,寄存器输出 $D_7 \sim D_0 = 11000000$, v_o' 为 7.5 V , v_A 再与 7.5 V 比较,因为 $v_A < 7.5 \text{ V}$,所以 D_6 存 0;输入第三个 CP 时, $D_7 \sim D_0 = 10100000$, $v_o' = 6.25 \text{ V}$; v_A 再与 v_o' 比较,……如此重复比较下去,经 8 个时钟周期,转换结束。由图中 v_o' 的波形可见,在逐次比较过程中,与输出数字量对应的模拟电压 v_o' 逐渐逼近 v_A 值,最后得到 A/D 转换器转换结果 $D_7 \sim D_0$ 为 10101111。该数字量所对应的模拟电压为

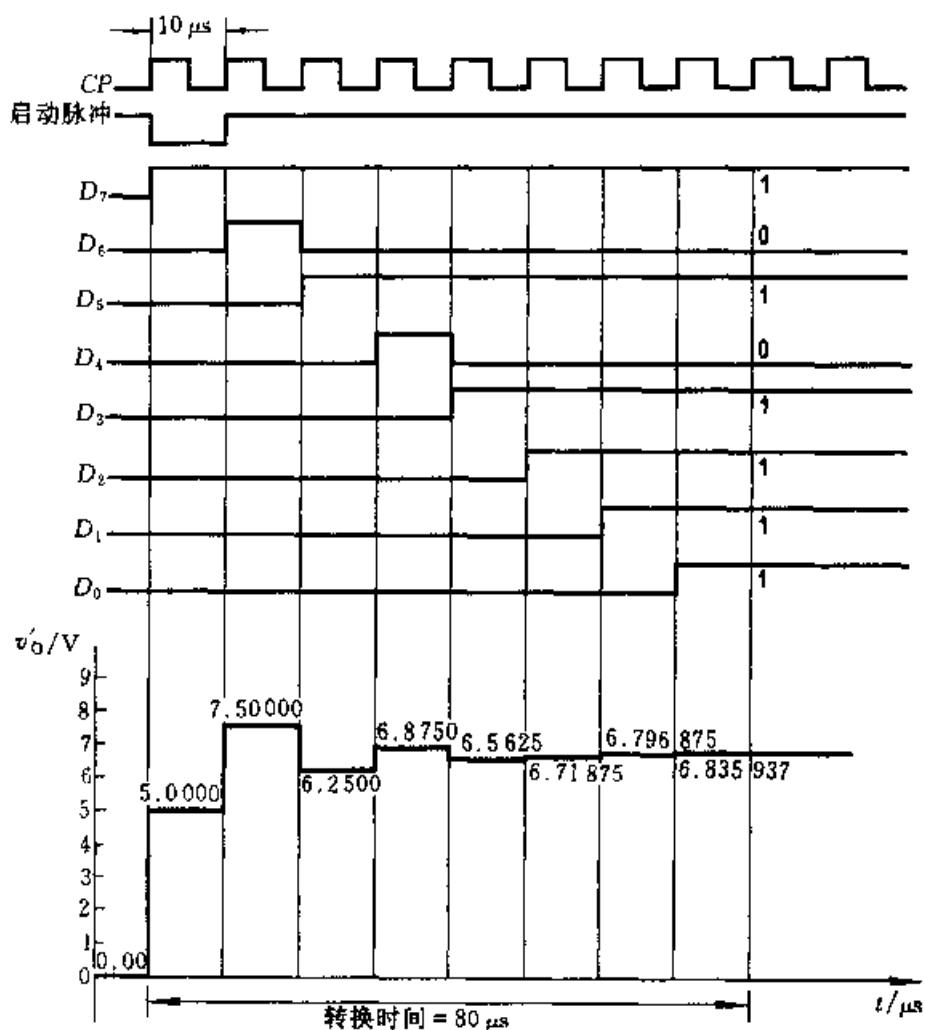


图 10.2.6 8 位逐次比较型 A/D 转换器波形图

6.8359375 V, 与实际输入的模拟电压 6.84 V 的相对误差仅为 0.06%。

例 10.2.1 4 位逐次比较型 A/D 转换器的逻辑电路如图 10.2.7 所示。图中 5 位移位寄存器可进行并入/并出或串入/并出操作, 其 F 为并行置数端, 高电平有效, S 为高位串行输入。数据寄存器由 D 边沿触发器组成, 数字量从 $Q_4 \sim Q_1$ 输出, 试分析电路的工作原理。

解: 电路工作过程如下: 当启动脉冲上升沿到来后, $FF_0 \sim FF_3$ 被清零, Q_5 置 1, Q_5 的高电平开启 G_2 门, 时钟 CP 脉冲进入移位寄存器。在第一个 CP 脉冲作用下, 由于移位寄存器的置数使能端 F 已由 0 变 1, 并行输入数据 $ABCDE$ 置入, $Q_A Q_B Q_C Q_D Q_E = 01111$ 。 Q_A 的低电平使数据寄存器的最高位置 1, 即 $Q_4 Q_3 Q_2 Q_1 = 1000$ 。 D/A 转换器将数字量 1000 转换为模拟电压 v'_0 , 送入比较器 C 与输入模拟电压 v_1 比较, 若输入电压 $v_1 > v'_0$, 则比较器 C 输出 v_c 为 1, 否则为 0。比较结果送 $D_4 \sim D_1$ 。

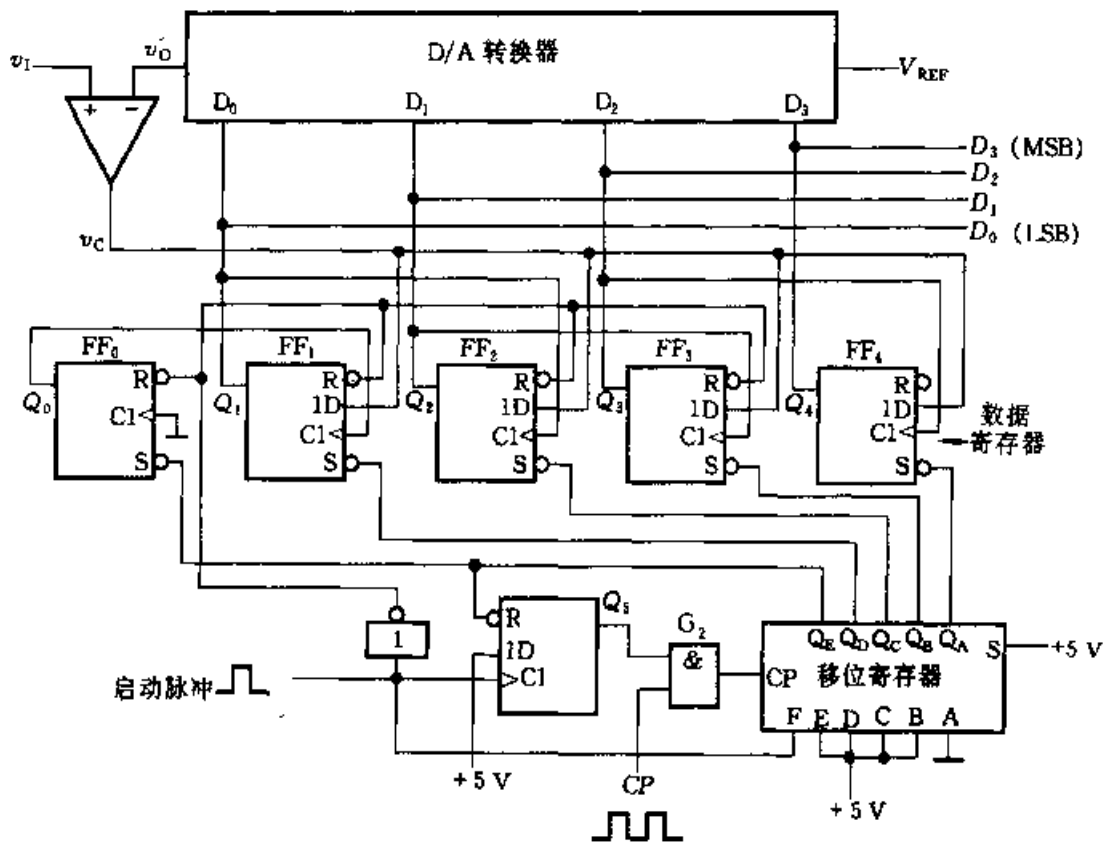


图 10.2.7 4 位逐次比较型 A/D 转换器的逻辑电路

第二个 CP 脉冲到来后,移位寄存器的串行输入端 S 为高电平, Q_A 由 0 变 1,同时最高位 Q_A 的 0 移至次高位 Q_B 。于是数据寄存器的 Q_3 由 0 变 1,这个正跳变作为有效触发信号加到 FF_4 的 CP 端使 v_C 的电平得以在 Q_4 保存下来。此时,由于其他触发器无正跳变脉冲, v_C 的信号对它们不起作用。 Q_3 变 1 后建立了新的 D/A 转换器的数据,输入电压再与其输出电压 v_O' 相比较,比较结果在第三个时钟脉冲作用下存于 Q_3 ……如此进行,直到 Q_E 由 1 变 0,使 Q_5 由 1 变 0 后将 G_2 封锁,转换完毕。于是电路的输出端 $D_3D_2D_1D_0$ 得到与输入电压 v_I 成正比的数字量。

由以上分析可见,逐次比较型 A/D 转换器完成一次转换所需时间与其位数和时钟脉冲频率有关,位数愈少,时钟频率越高,转换所需时间越短。这种 A/D 转换器具有转换速度快,精度高的特点。

常用集成逐次比较型 A/D 转换器有 ADC0808/0809 系列(8 位)、AD575(10 位)、AD574A(12 位)等。

10.2.4 双积分式 A/D 转换器

双积分 A/D 转换器是一种间接 A/D 转换器。它的基本原理是,对输入模拟

电压和参考电压分别进行两次积分,将输入电压平均值变换成与之成正比的时间间隔,然后利用时钟脉冲和计数器测出此时间间隔,进而得到相应的数字量输出。由于该转换电路是对输入电压的平均值进行变换,所以它具有很强的抗工频干扰能力,在数字测量中得到广泛应用。

图 10.2.8 是这种转换器的原理电路,它由积分器(由集成运放 A 组成)、过零比较器(C)、时钟脉冲控制门(G)和定时/计数器($FF_0 \sim FF_n$)等几部分组成。

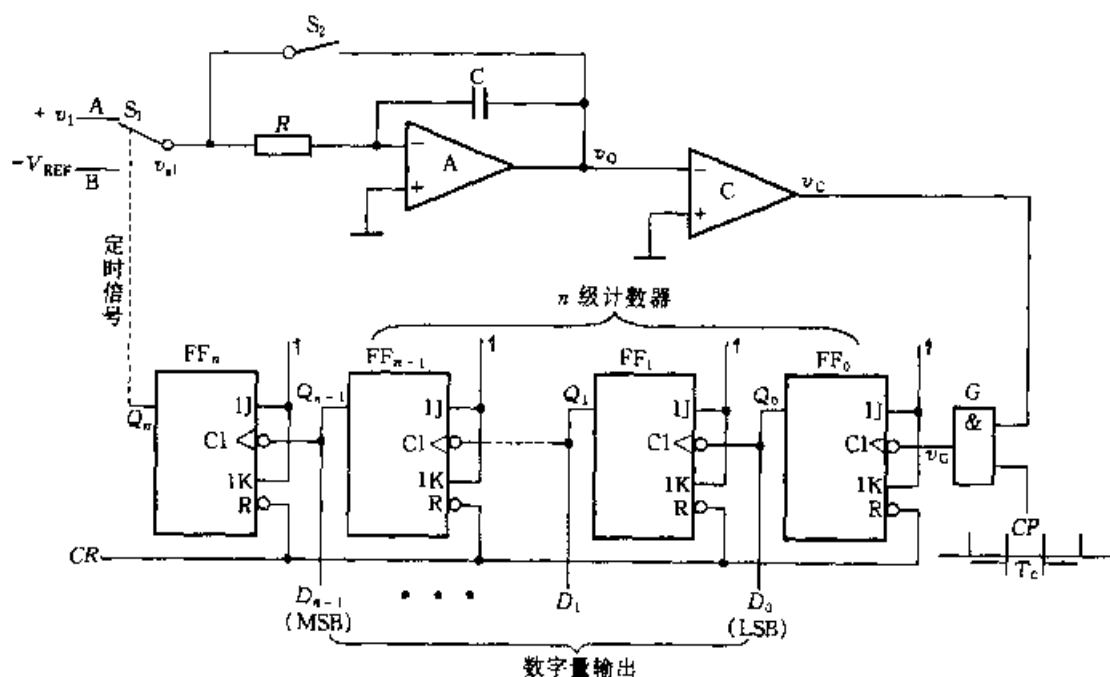


图 10.2.8 双积分 A/D 转换器

积分器 积分器是转换器的核心部分,它的输入端所接开关 S_1 由定时信号 Q_n 控制。当 Q_n 为不同电平时,极性相反的输入电压 v_i 和参考电压 V_{REF} 将分别加到积分器的输入端,进行两次方向相反的积分,积分时间常数 $\tau = RC$ 。

过零比较器 过零比较器用来确定积分器输出电压 v_o 过零的时刻。当 $v_o \geq 0$ 时,比较器输出 v_c 为低电平;当 $v_o < 0$ 时, v_c 为高电平。比较器的输出信号接至时钟控制门(G)作为关门和开门信号。

计数器和定时器 它由 $n+1$ 个接成计数型的触发器 $FF_0 \sim FF_n$ 串联组成。触发器 $FF_0 \sim FF_{n-1}$ 组成 n 级计数器,对输入时钟脉冲 CP 计数,以便把与输入电压平均值成正比的时间间隔转变成数字信号输出。当计数到 2^n 个时钟脉冲时, $FF_0 \sim FF_{n-1}$ 均回到 0 态,而 FF_n 翻转为 1 态, $Q_n = 1$ 后开关 S_1 从位置 A 转接到 B。

时钟脉冲控制门 时钟脉冲源标准周期 T_c ,作为测量时间间隔的标准时间。当 $v_c = 1$ 时,门打开,时钟脉冲通过门加到触发器 FF_0 的输入端。

下面以输入正极性的直流电压 v_i 为例,说明电路将模拟电压转换为数字量的基本原理。电路工作过程分为以下几个阶段进行,图中各处的工作波形如图 10.2.9 所示。

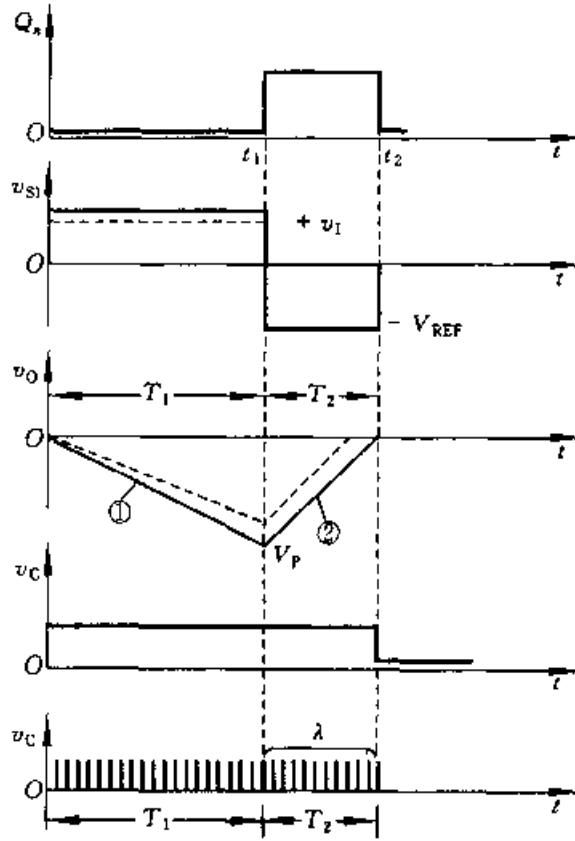


图 10.2.9 双积分 A/D 转换器各处工作波形

(1) 准备阶段

首先控制电路提供 CR 信号使计数器清零,同时使开关 S_2 闭合,待积分电容放电完毕后,再使 S_2 断开。

(2) 第一次积分阶段

在转换过程开始时($t=0$),开关 S_1 与 A 端接通,正的输入电压 v_i 加到积分器的输入端。积分器从 0 V 开始对 v_i 积分,其波形如图 10.2.10c 斜线 $O-V_p$ 段所示。根据积分器的原理可得

$$v_o = -\frac{1}{\tau} \int_0^t v_i dt \quad (10.2.2)$$

由于 $v_o < 0$, 过零比较器输出为高电平,时钟控制门 G 被打开。于是,计数器在 CP 作用下从 0 开始计数。经 2^n 个时钟脉冲后,触发器 $FF_0 \sim FF_{n-1}$ 都翻转到 0 态,而 $Q_n = 1$, 开关 S_1 由 A 点转接到 B 点,第一次积分结束。第一次积分时间为

$$t = T_1 = 2^n T_c \quad (10.2.3)$$

令 V_1 为输入电压在 T_1 时间间隔内的平均值,则由式(10.2.2)可得第一次积分结束时积分器的输出电压为 V_p

$$V_p = -\frac{T_1}{\tau} V_1 = -\frac{2^n T_c}{\tau} V_1 \quad (10.2.4)$$

(3) 第二次积分阶段

当 $t = t_1$ 时, S_1 转接到 B 点,具有与 v_1 相反极性的基准电压 $-V_{REF}$ 加到积分器的输入端;积分器开始向相反方向进行第二次积分;当 $t = t_2$ 时,积分器输出电压 $v_o \geq 0$,比较器输出 $v_c = 0$,时钟脉冲控制门 G 被关闭,计数停止。在此阶段结束时 v_o 的表达式可写为

$$v_o(t_2) = V_p - \frac{1}{\tau} \int_{t_1}^{t_2} (-V_{REF}) dt = 0 \quad (10.2.5)$$

设 $T_2 = t_2 - t_1$,于是有

$$\frac{V_{REF} T_2}{\tau} = \frac{2^n T_c}{\tau} V_1$$

设在此期间计数器所累计的时钟脉冲个数为 λ ,则

$$T_2 = \lambda T_c \quad (10.2.6)$$

$$T_2 = \frac{2^n T_c}{V_{REF}} V_1 \quad (10.2.7)$$

可见, T_2 与 V_1 成正比, T_2 就是双积分 A/D 转换过程中的中间变量。

$$\lambda = \frac{T_2}{T_c} = \frac{2^n}{V_{REF}} V_1 \quad (10.2.8)$$

式(10.2.8)表明,在计数器中所计得的数 λ ($\lambda = Q_{n-1} \cdots Q_1 Q_0$),与在取样时间 T_1 内输入电压的平均值 V_1 成正比的。只要 $V_1 < V_{REF}$,转换器就能正常地将输入模拟电压转换为数字量,并能从计数器读取转换的结果。如果取 $V_{REF} = 2^n V$,则 $\lambda = V_1$,计数器所计的数在数值上就等于被测电压。

由于双积分 A/D 转换器在 T_1 时间内采的是输入电压的平均值,因此具有很强的抗工频干扰的能力。尤其对周期等于 T_1 或几分之一 T_1 的对称干扰(所谓对称干扰是指整个周期内平均值为零的干扰),从理论上来说,有无穷大的抑制能力。即使当工频干扰幅度大于被测直流信号,使得输入信号正负变化时,仍有良好的抑制能力。由于在工业系统中经常碰到的是工频(50 Hz)或工频的倍频干扰,故通常选定采样时间 T_1 总是等于工频电源周期的倍数,如 20 ms 或 40 ms 等。另一方面,由于在转换过程中,前后两次积分所采用的同一积分器。因此,在两次积分期间(一般在几十至数百毫秒之间),R、C 和脉冲源等元器件

参数的变化对转换精度的影响均可以忽略。

最后必须指出,在第二次积分阶段结束后,控制电路又使开关 S_2 闭合,电容 C 放电,积分器回零。电路再次进入准备阶段,等待下一次转换开始。

单片集成双积分式 A/D 转换器有 ADC - EK8B(8 位,二进制码)、ADC - EK10B(10 位,二进制码)、MC14433($3\frac{1}{2}$ 位,BCD 码)等。

10.2.5 A/D 转换器的主要技术指标

A/D 转换器的主要技术指标有转换精度、转换速度等。选择 A/D 转换器时,除考虑这两项技术指标外,还应注意满足其输入电压的范围、输出数字的编码、工作温度范围和电压稳定度等方面的要求。

1. 转换精度

单片集成 A/D 转换器的转换精度是用分辨率和转换误差来描述的。

(1) 分辨率

A/D 转换器的分辨率以输出二进制(或十进制)数的位数表示。它说明 A/D 转换器对输入信号的分辨能力。从理论上讲, n 位输出的 A/D 转换器能区分 2^n 个不同等级的输入模拟电压,能区分输入电压的最小值为满量程输入的 $1/2^n$ 。在最大输入电压一定时,输出位数愈多,量化单位愈小,分辨率愈高。例如 A/D 转换器输出为 8 位二进制数,输入信号最大值为 5 V,那么这个转换器应能区分出输入信号的最小电压为 19.53 mV。

(2) 转换误差

转换误差通常是以输出误差的最大值形式给出。它表示 A/D 转换器实际输出的数字量和理论上的输出数字量之间的差别。常用最低有效位的倍数表示。例如给出相对误差 $\leq \pm \text{LSB}/2$,这就表明实际输出的数字量和理论上应得到的输出数字量之间的误差小于最低位的半个字。

2. 转换时间

转换时间是指 A/D 转换器从转换控制信号到来开始,到输出端得到稳定的数字信号所经过的时间。A/D 转换器的转换时间与转换电路的类型有关。不同类型的转换器转换速度相差甚远。其中并行比较 A/D 转换器的转换速度最高,8 位二进制输出的单片集成 A/D 转换器转换时间可达到 50 ns 以内,逐次比较型 A/D 转换器次之,它们多数转换时间在 10~50 μs 之间,也有达几百纳秒的。间接 A/D 转换器的速度最慢,如双积分 A/D 转换器的转换时间大都在几十毫秒至几百毫秒之间。在实际应用中,应从系统数据总的位数、精度要求、输入模拟信号的范围及输入信号极性等方面综合考虑 A/D 转换器的选用。

例 10.2.2 某信号采集系统要求用一片 A/D 转换集成芯片在 1s(秒)内对

16个热电偶的输出电压分时进行A/D转换。已知热电偶输出电压范围为0~0.025 V(对应于0~450℃温度范围),需要分辨的温度为0.1℃,试问应选择多少位的A/D转换器,其转换时间为多少?

解: 对于从0~450℃温度范围,信号电压为0~0.025 V,分辨温度为0.1℃,这相当于 $\frac{0.1}{450} = \frac{1}{4500}$ 的分辨率。12位A/D转换器的分辨率为 $\frac{1}{2^{12}} = \frac{1}{4096}$,所以必须选用13位的A/D转换器。

系统的取样速率为每秒16次,取样时间为62.5 ms。对于这样慢速的取样,任何一个A/D转换器都可达到。可选用带有取样-保持(S/H)的逐次比较A/D转换器或不带S/H的双积分式A/D转换器均可。

10.2.6 集成A/D转换器及其应用

在单片集成A/D转换器中,逐次比较型使用较多,下面我们以ADC0804介绍集成A/D转换器及其应用。

1. ADC0804 引脚及使用说明

ADC0804是用CMOS集成工艺制成的逐次比较型模数转换芯片。分辨率8位,转换时间100 μs,输入电压范围为0~5 V,增加某些外部电路后,输入模拟电压可为±5 V。该芯片内有输出数据锁存器,当与计算机连接时,转换电路的输出可以直接连接在CPU^①数据总线上,无需附加逻辑接口电路。ADC0804芯片管脚图如图10.2.10所示。引脚名称及意义如下:

V_{IN+} 、 V_{IN-} :ADC0804的两模拟信号输入端,用以接收单极性、双极性和差模输入信号。

$D_7 \sim D_0$:A/D转换器数据输出端,该输出端具有三态特性,能与微机总线相接。

AGND:模拟信号地。

DGND:数字信号地。

CLKIN:外电路提供时钟脉冲输入端。

CLKR:内部时钟发生器外接电阻端,与CLKIN端配合可由芯片自身产生时钟脉冲,其频率为 $1/1.1RC$ 。

CS:片选信号输入端,低电平有效,一旦CS有效,表明A/D转换器被选中,可启动工作。

WR:写信号输入,接受微机系统或其他数字系统控制芯片的启动输入端,低电平有效,当CS、WR同时为低电平时,启动转换。

RD:读信号输入,低电平有效,当CS、RD同时为低电平时,可读取转换输出数据。

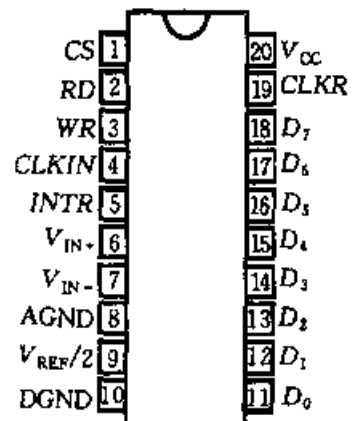


图 10.2.10 ADC0804 引脚图

① CPU系 Central Processing Unit 的缩写。

INTR:转换结束输出信号,低电平有效。输出低电平表示本次转换已完成。该信号常作为向微机系统发出的中断请求信号。

在使用时应注意以下几点:

(1) 转换时序

ADC0804 控制信号的时序图如图 10.2.11 所示,由图可见,各控制信号时

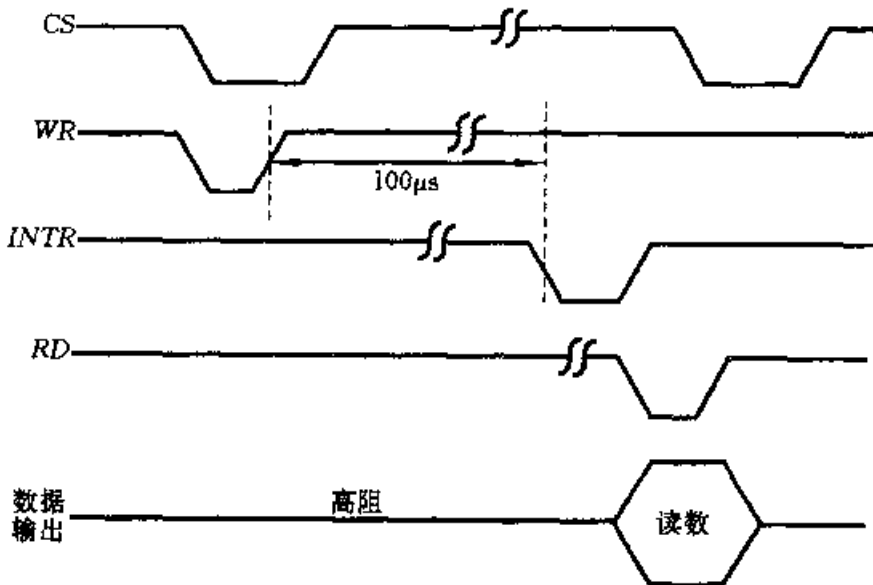


图 10.2.11 ADC0804 控制信号的时序图

序关系为:当 CS 与 WR 同为低电平时,A/D 转换被启动且在 WR 上升沿后 $100\mu\text{s}$ 模数完成转换,转换结果存入数据锁存器,同时 INTR 自动变为低电平,表示本次转换已结束。如 CS、RD 同时来低电平,则数据锁存器三态门打开,数字信号送出,而在 RD 高电平到来后三态门处于高阻状态。

(2) 零点和满刻度调节。

ADC0804 的零点无需调整。满刻度调整时,先给输入端加入电压 $V_{\text{IN}+}$,使满刻度所对应的电压值是 $V_{\text{IN}+} = V_{\text{max}} - 1.5 \left[\frac{V_{\text{max}} - V_{\text{min}}}{256} \right]$,其中 V_{max} 是输入电压的最大值, V_{min} 是输入电压的最小值。当输入电压与 $V_{\text{IN}+}$ 值相当时,调整 $V_{\text{REF}/2}$ 端电压值使输出码为 FEH 或 FFH。

(3) 参考电压的调节

在使用 A/D 转换器时,为保证其转换精度,要求输入电压满量程使用。如输入电压动态范围较小,则可调节参考电压 V_{REF} ,以保证小信号输入时 ADC0804 芯片 8 位的转换精度。

(4) 接地

模数、数模转换电路中要特别注意到地线的正确连接,否则干扰很严重,以

致影响转换结果的准确性。A/D、D/A 及取样-保持芯片上都提供了独立的模拟地(AGND)和数字地(DGND)的引脚。在线路设计中,必须将所有器件的模拟地和数字地分别相连,然后将模拟地与数字地仅在一点上相连接。地线的正确连接方法如图 10.2.12 所示。

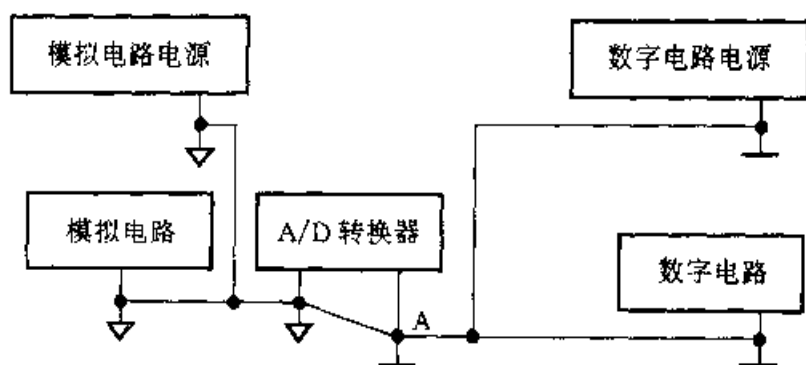


图 10.2.12 正确的地线连接

2. ADC0804 的典型应用

下面以数据采集系统为例介绍 ADC0804 的典型应用。

在现代过程控制及各种智能仪器和仪表中,为采集被控(被测)对象数据以达到由计算机进行实时控制、检测的目的,常用微处理器和 A/D 转换器组成数据采集系统。单通道微机化数据采集系统的示意图如图 10.2.13 所示。

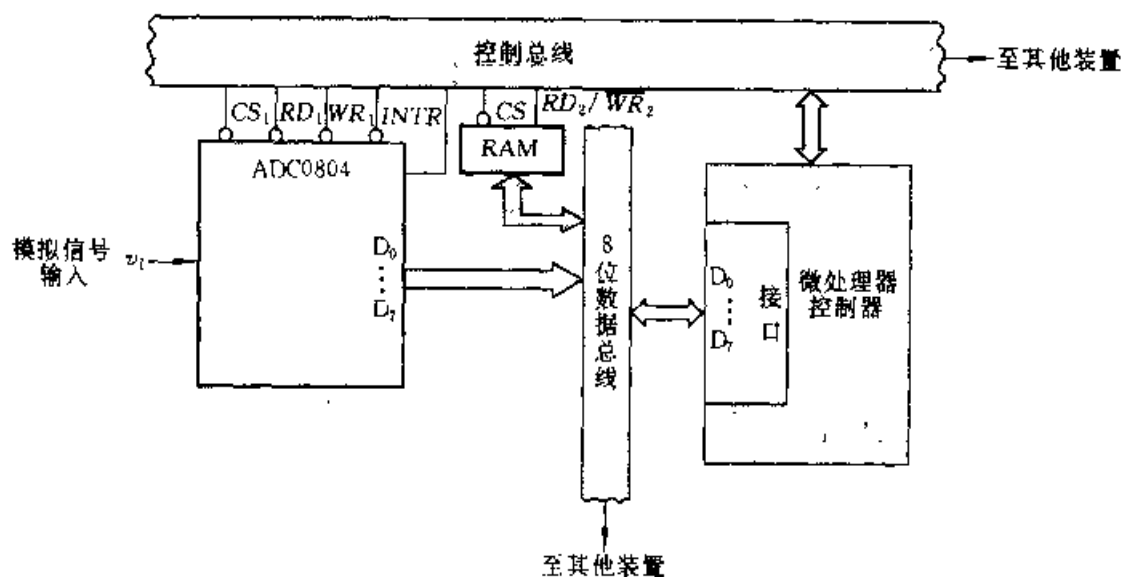


图 10.2.13 单通道微机化数据采集系统示意图

系统由微处理器、存储器和 A/D 转换器组成,它们之间通过数据总线(DBUS)和控制总线(CBUS)连接,系统信号采用总线传送方式。

现以程序查询方式为例,说明 ADC0804 在数据采集系统中的应用。采集数据时,首先微处理器执行一条传送指令,在该指令执行过程中,微处理器在控制总线的同时产生 CS_1 、 WR_1 低电平信号,启动 A/D 转换器工作,ADC0804 经 $100 \mu\text{s}$ 后将输入模拟信号转换为数字信号存于输出锁存器,并在 $INTR$ 端产生低电平表示转换结束,并通知微处理器可来取数。当微处理器通过总线查询到 $INTR$ 为低电平时,立即执行输入指令,以产生 CS 、 RD_2 低电平信号到 ADC0804 相应引脚,将数据取出并存入存储器中。整个数据采集过程中,由微处理器有序地执行若干指令完成。

复习思考题

10.2.1 实现模数转换一般要经过哪 4 个过程?按工作原理不同分类,A/D 转换器可分为哪两种?

10.2.2 在图 10.2.4 所示并行比较 A/D 转换电路中,若输入电压 v_i 为负电压,试问电路能否正常进行 A/D 转换?为什么?如不能正常工作,需要如何改进电路?

10.2.3 已知在图 10.2.4 所示并行 A/D 转换器中, $V_{REF} = 10 \text{ V}$, $V_1 = 9 \text{ V}$,试问输出数字量 $D_2 D_1 D_0 = ?$

10.2.4 在图 10.2.7 所示逐次比较型 A/D 转换器中,完成一次 A/D 转换所需时间为多少?转换时间与哪些因素有关?

10.2.5 试问双积分 A/D 转换器输出数字量 N_B 与下述哪些参数有关?关系如何?(a)积分时间常数;(b)时钟脉冲频率;(c)输入信号电压;(d)计数器位数;(e)运算放大器的零漂;(f) $|V_{REF}|$ 。

10.2.6 比较并行比较型 A/D 转换器、逐次比较型 A/D 转换器、双积分式 A/D 转换器的优、缺点,试问应如何根据实际系统要求合理选用?

* 10.3 CAD 例题

例 CE10.3.1 倒梯形电阻网络 D/A 转换器和计数器构成的波形产生电路如图 CE10.3.1a 所示。其中 $R_i - R = 1 \text{ k}\Omega$ 。试用 PSPICE 程序分析:(1)当 CP 频率为 100 kHz 时, $D_0 \sim D_3$ 和 v_O 的波形;(2)当倒梯形电阻网络中的电阻有 1% 误差时,利用 Worst Case 分析功能,求输出产生的最大误差。

解: (1) 图 CE10.3.1a 中的开关用图 b 的电路模拟,运放采用 LF411。设置瞬态分析,且触发器的初态设为 0。仿真后的输出波形如图 CE10.3.1c 所示。

(2) 将电阻换成 Rbreak 模型,设置瞬态分析和 Worst Case 分析,仿真后由输出文件得到,在 $t = 305.27 \mu\text{s}$ 处有 0.0977 V 的最大误差,则满度值的相对误差为 2.08%。

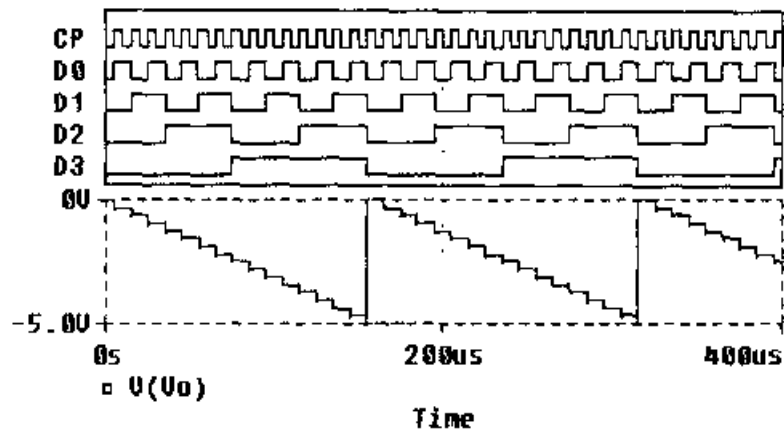
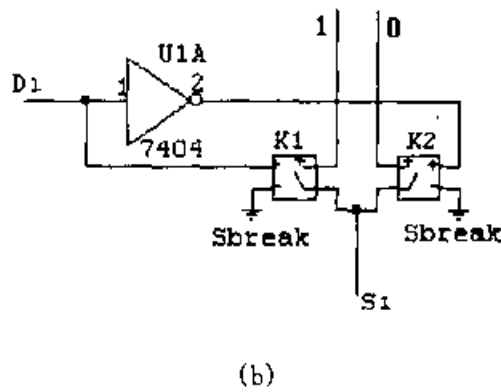
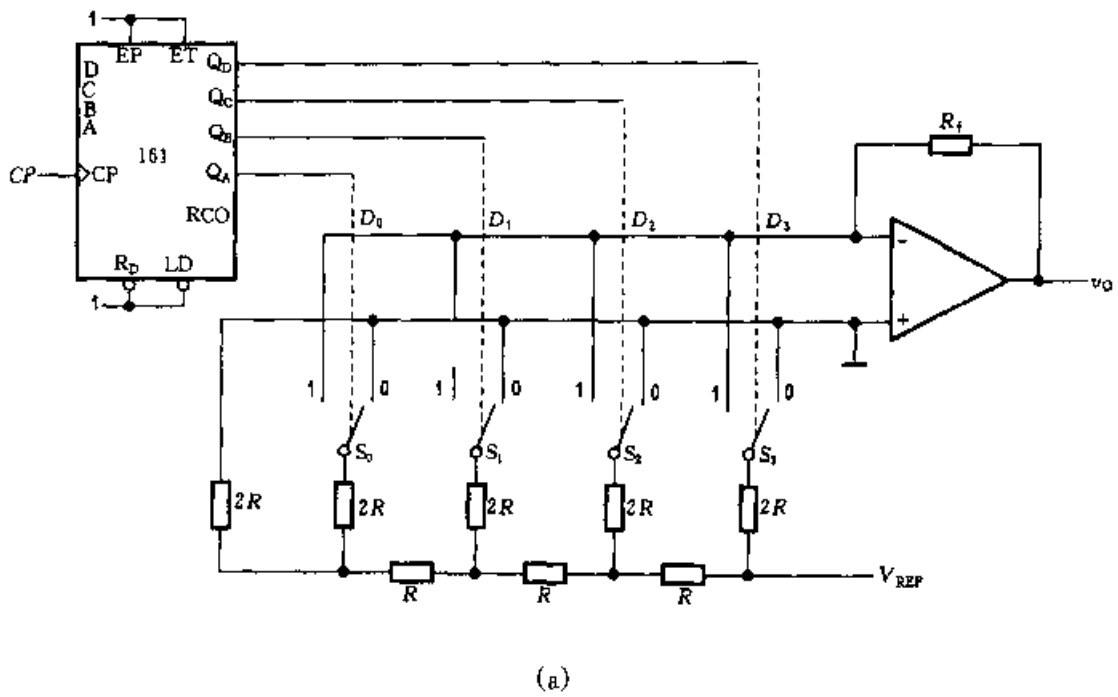
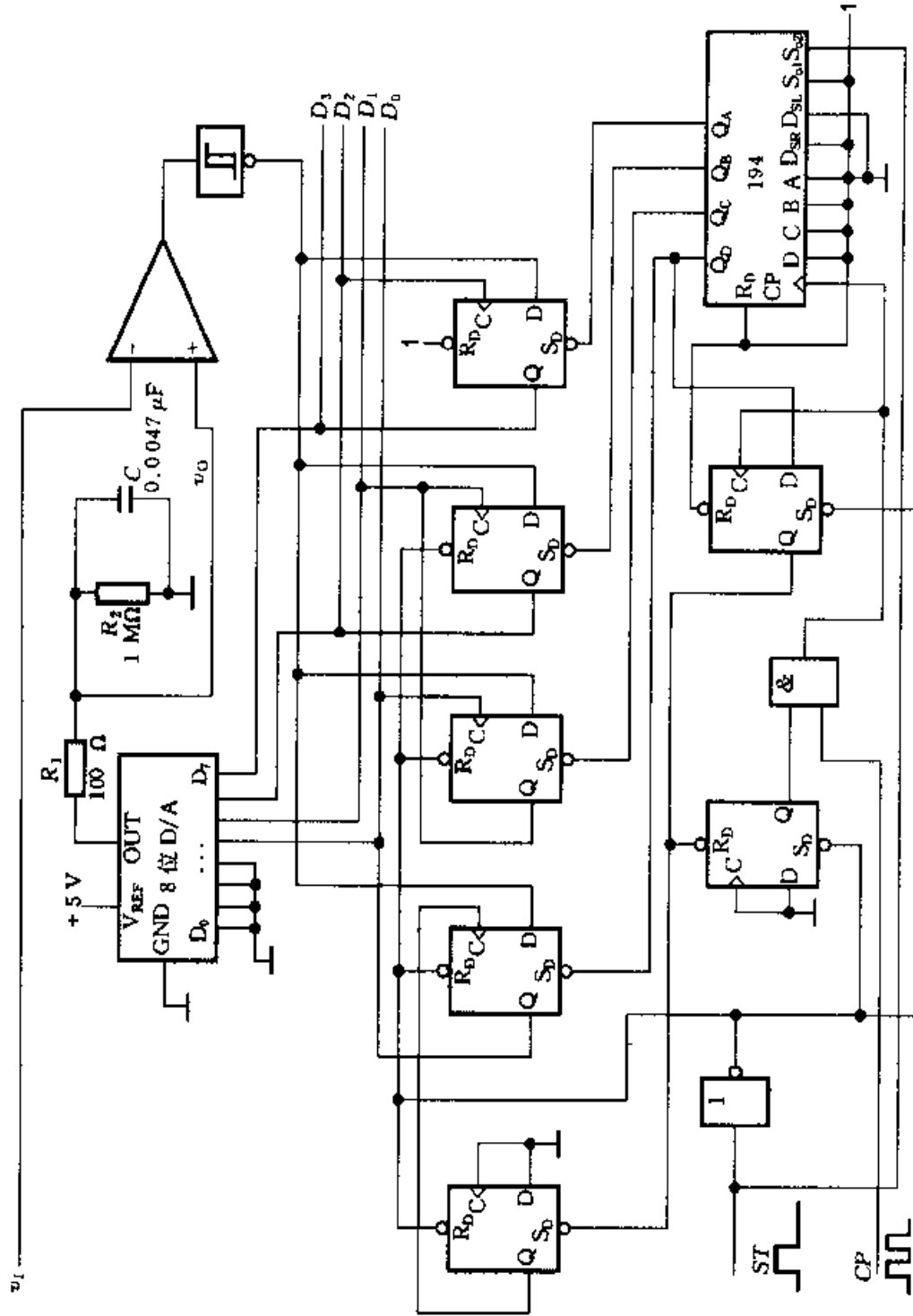


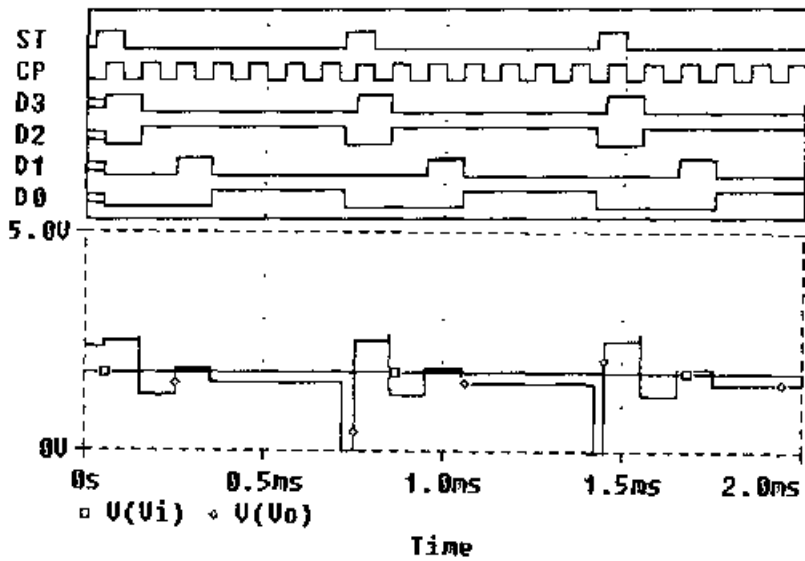
图 CE10.3.1

(a) 倒梯形电阻网络 D/A 转换器和计数器构成的波形产生电路原理图

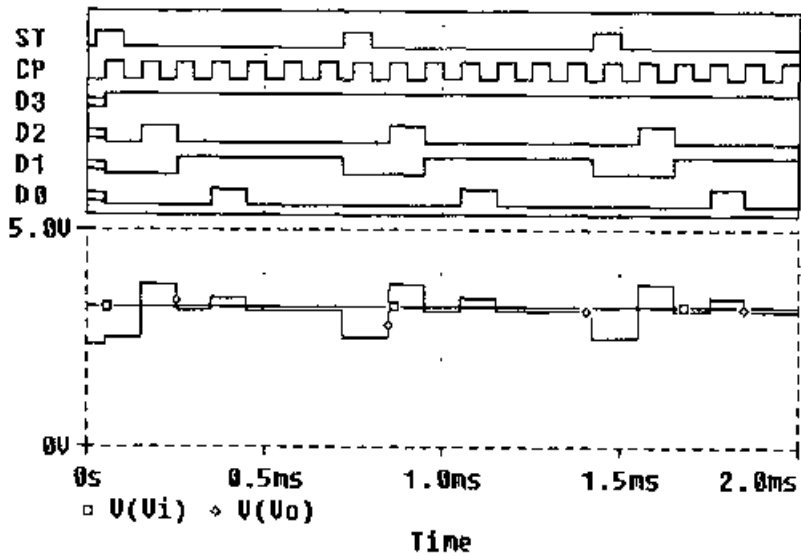
(b) 图 a 中的开关模拟电路 (c) $D_0 \sim D_3$ 和 v_O 的波形



(a)
图 CE10.3.2(-)



(b)



(c)

图 CE10.3.2(二)

(a) 4位逐次比较型 A/D 转换器电路

(b) v_i 为 1.8 V 时的波形 (c) v_i 为 3.2 V 时的波形

例 CE10.3.2 4位逐次比较型 A/D 转换器电路如图 CE10.3.2a 所示。已知 CP 频率为 10 kHz, 试用 PSPICE 程序分析, 当 v_i 分别为 1.8 V 和 3.2 V 时, v_i 、 v_o 、 $D_3 \sim D_0$ 的波形和转换后的数字量。

解: 图 CE10.3.2a 中的 8 位 D/A 转换器采用模型 DAC8break, 比较器采用 LF411。设置瞬态分析, 仿真后得到 v_i 为 1.8 V 和 3.2 V 时的波形如图 b、c 所示。由图中看出, 转换后的数字量分别为 0101 和 1010。

小 结

- A/D 和 D/A 转换器是现代数字系统的重要部件,应用日益广泛。
- 倒 T 形电阻网络 D/A 转换器具有如下特点:电阻网络阻值仅有两种,即 R 和 $2R$;各 $2R$ 支路电流 I_i 与相应的 D_i 数码状态无关,是一定值;由于支路电流流向运放反相端时不存在传输时间,因而具有较高的转换速度。
- 在权电流型 D/A 转换器中,由于恒流源电路和高速模拟开关的运用使其具有精度高、转换快的优点,双极型单片集成 D/A 转换器多采用此种类型电路。
- D/A 转换器有两种输出方式。双极型输出电路与输入编码有关。无论哪种输出方式,在使用时应注意进行零点和满量程调节。
- 不同的 A/D 转换方式具有各自的特点,在要求转换速度高的场合,选用并行 A/D 转换器;在要求精度高的情况,可以采用双积分 A/D 转换器,当然也可选用高分辨率的其他形式 A/D 转换器,但会增加成本。由于逐次比较型 A/D 转换器在一定程度上兼顾了以上两种转换器的优点,因此得到普遍应用。
- A/D 转换器和 D/A 转换器的主要技术参数是转换精度和转换速度,在与系统连接后,转换器的这两项指标决定了系统的精度与速度。目前,A/D 与 D/A 转换器的发展趋势是高速、高分辨率及易于与微型计算机接口,用以满足各个领域对信号处理的要求。

习 题

10.1.1 10 位倒 T 形电阻网络 D/A 转换器如图题 10.1.1 所示,当 $R = R_i$ 时:(1)试求输出电压的取值范围;(2)若要求电路输入数字量为 200H 时输出电压 $V_O = 5\text{ V}$,试问 V_{REF} 应取何值?

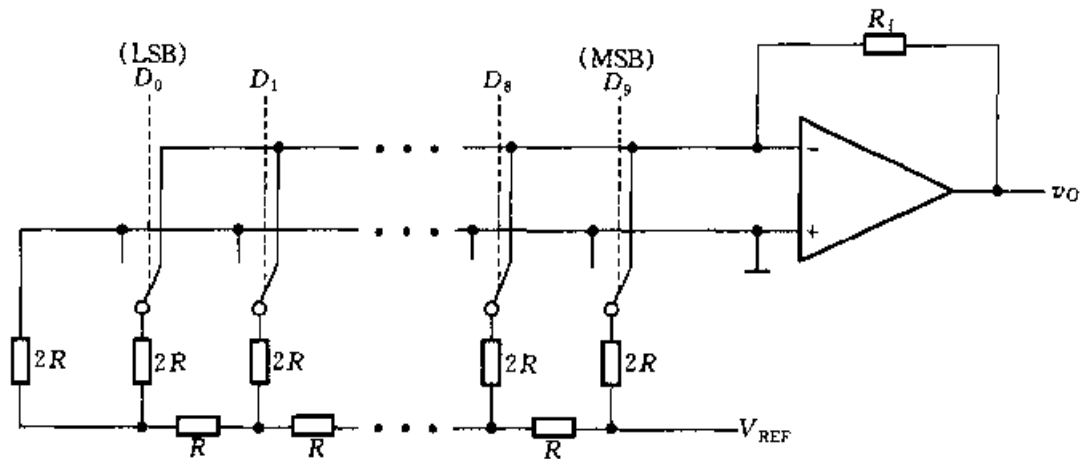
10.1.2 在图 10.1.4 中所示 4 位权电流 D/A 转换器中,已知 $V_{\text{REF}} = -6\text{ V}$, $R_1 = 48\text{ k}\Omega$,当输入 $D_3 D_2 D_1 D_0 = 1100$ 时, $v_O = 1.5\text{ V}$,试确定 R_i 的值。

10.1.3 n 位权电阻 D/A 转换器如图题 10.1.3 所示。(1)试推导输出电压 v_O 与输入数字量的关系式;(2)如 $n = 8$, $V_{\text{REF}} = -10\text{ V}$,当 $R_i = \frac{1}{18}R$ 时,如输入数码为 20 H,试求输出电压值。

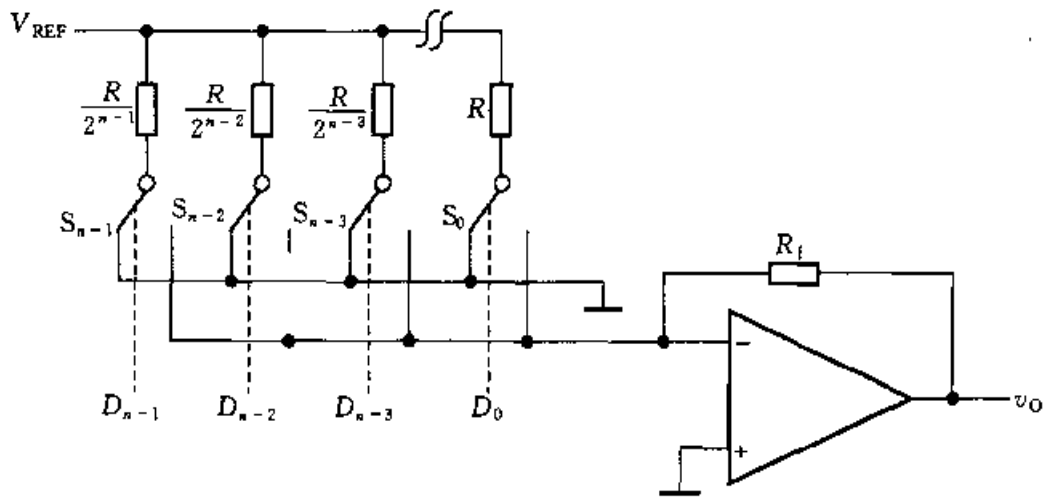
10.1.4 图题 10.1.4 为一权电阻和梯形网络相结合的 D/A 转换电路。

- (1) 试证明:当 $r = 8R$ 时,电路为 8 位的二进制码 D/A 转换器;
- (2) 试证明:当 $r = 4.8R$ 时,该电路为 2 位 BCD 码 D/A 转换器。

10.1.5 在图 10.1.2 所示的倒 T 形电阻网络 D/A 转换器中,设 $R_i = R$,外接参考电压 $V_{\text{REF}} = -10\text{ V}$,为保证 V_{REF} 偏离标准值所引起的误差小于 $\text{LSB}/2$,试计算 V_{REF} 的相对稳定度应取多少?



图题 10.1.1



图题 10.1.3

10.1.6 由 AD7520 组成双极性输出 D/A 转换器如图题 10.1.6 所示。

(1) 根据电路写出输出电压 v_O 的表达式；

(2) 试问为实现 2 的补码, 双极性输出电路应如何连接, 电路中 V_B 、 R_B 、 V_{REF} 和片内的 R 应满足什么关系。

10.1.7 可编程放大器(数控可变增益放大器)电路如图题 10.1.7 所示。

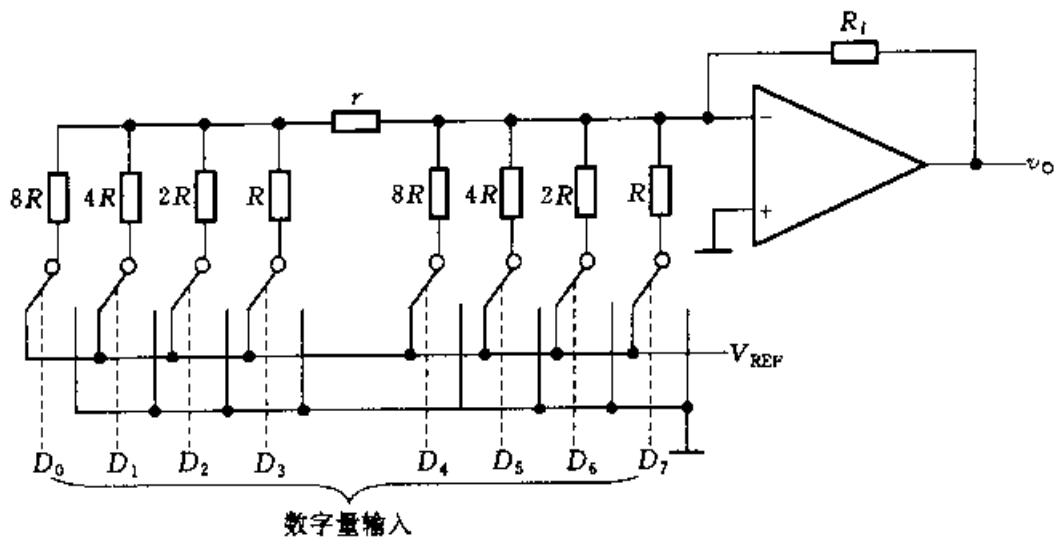
(1) 推导电路电压放大倍数 $A_V = v_O/v_I$ 的表达式；

(2) 当输入编码为(001H)和(3FF)时, 电压放大倍数 A_V 分别为多少？

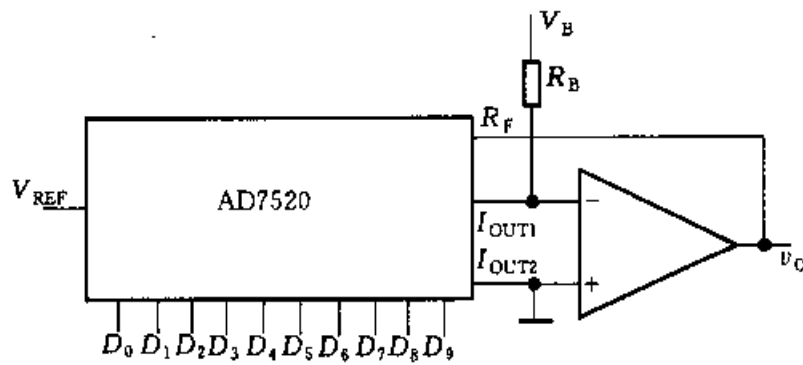
(3) 试问当输入编码为(000 H)时, 运放 A_1 处于什么状态？

10.1.8 试用 D/A 转换器 AD7520 和计数器 74161 组成如图题 10.1.8 所示的 10 阶梯形波发生器, 要求画出完整的逻辑图。

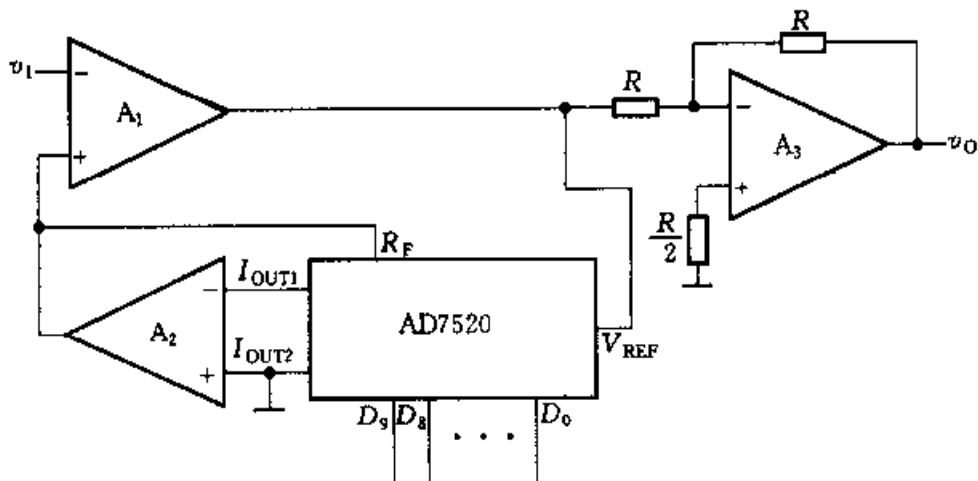
10.2.1 在图 10.2.3 所示并行比较型 A/D 转换器中, $V_{REF} = 7\text{ V}$, 试问电路的最小量化单位 Δ 等于多少？当 $v_I = 2.4\text{ V}$ 时输出数字量 $D_2 D_1 D_0 = ?$ 此时的量化误差 ϵ 为多少？



图题 10.1.4



图题 10.1.6

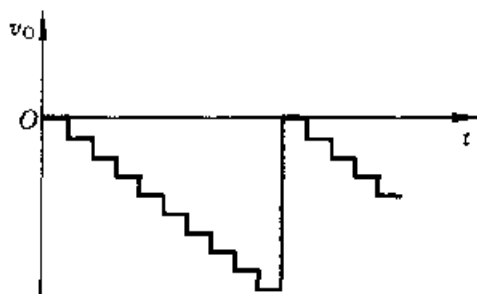


图题 10.1.7

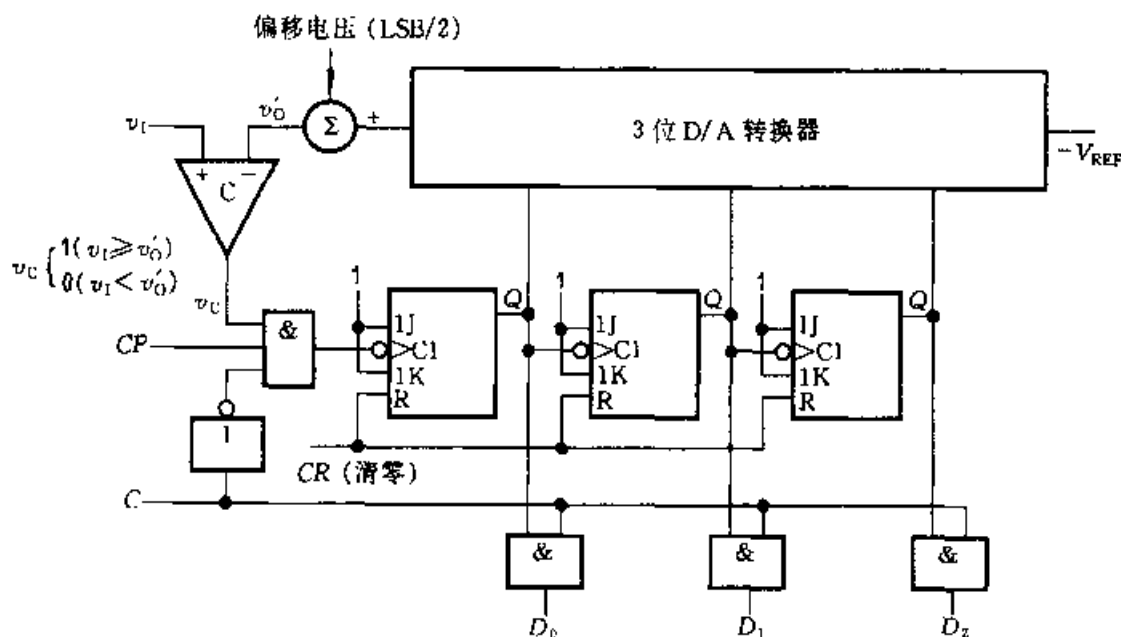
10.2.2 在图 10.2.5 逐次比较 A/D 转换器中,若 $n = 10$,已知时钟频率为 1 MHz,则完成一次转换所需时间是多少?如要求完成一次转换的时间小于 $100 \mu\text{s}$,问时钟频率应选多大?

10.2.3 在图 10.2.7 所示逐次比较 A/D 转换器中,设 $V_{\text{REF}} = 10 \text{ V}$, $v_i = 8.26 \text{ V}$,试画出在时钟脉冲作用下 v'_o 的波形并写出转换结果。

10.2.4 一计数型 A/D 转换器如图题 10.2.4 所示。试分析其工作原理



图题 10.1.8



图题 10.2.4

10.2.5 某双积分 A/D 转换器中,计数器为十进制计数器,其最大计数容量为 $(3000)_{10}$ 。已知计数时钟频率 $f_{\text{CP}} = 30 \text{ kHz}$,积分器中 $R = 100 \text{ k}\Omega$, $C = 1 \mu\text{F}$,输入电压 v_i 的变化范围为 $0 \sim 5 \text{ V}$ 。试求:(1)第一次积分时间 T_1 ;(2)求积分器的最大输出电压 $|V_{\text{Omax}}|$;(3)当 $V_{\text{REF}} = 10 \text{ V}$,第二次积分计数器计数值 $\lambda = (1500)_{10}$ 时,输入电压的平均值 V_i 为多少?

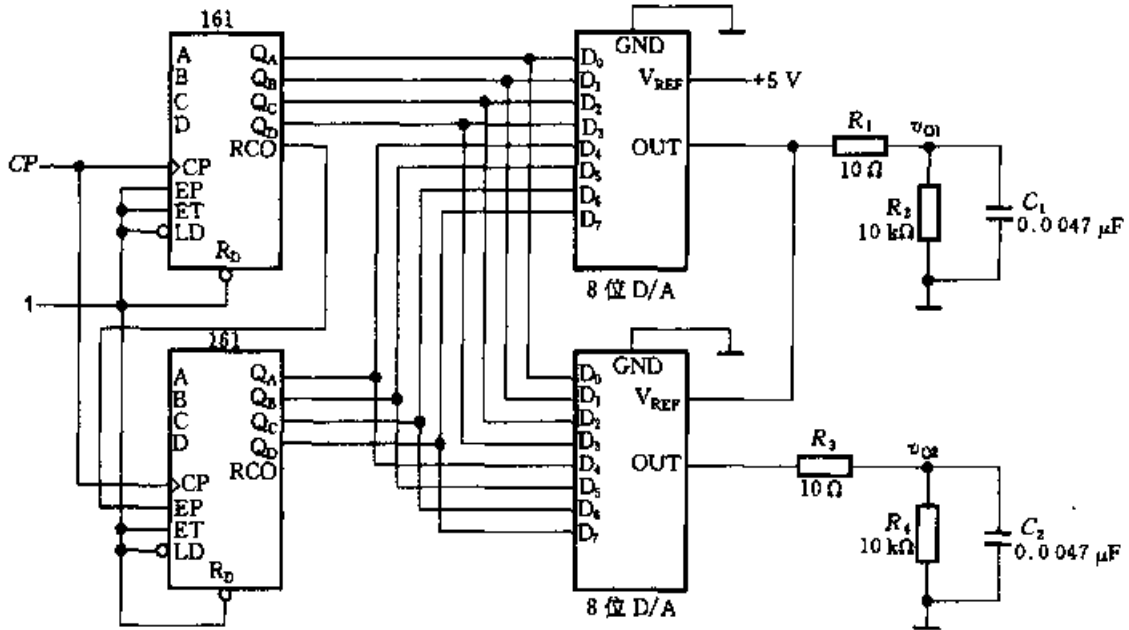
10.2.6 在图 10.2.8 中所示双积分 A/D 转换中,设时钟脉冲频率为 f_{CP} ,其分辨率为 n 位,写出最高的转换频率表达式。

10.2.7 在双积分 A/D 转换器中,输入电压 v_i 和参考电压 V_{REF} 在极性和数值上应满足什么关系?如 $v_i > |V_{\text{REF}}|$,电路能完成模数转换吗?为什么?

10.2.8 在应用 A/D 转换过程中应注意哪些主要问题,如某人用满度值为 10 V 的 8 位 A/D 转换器对输入信号幅值为 0.5 V 的电压进行模数转换,你认为这样使用正确吗?为什么?

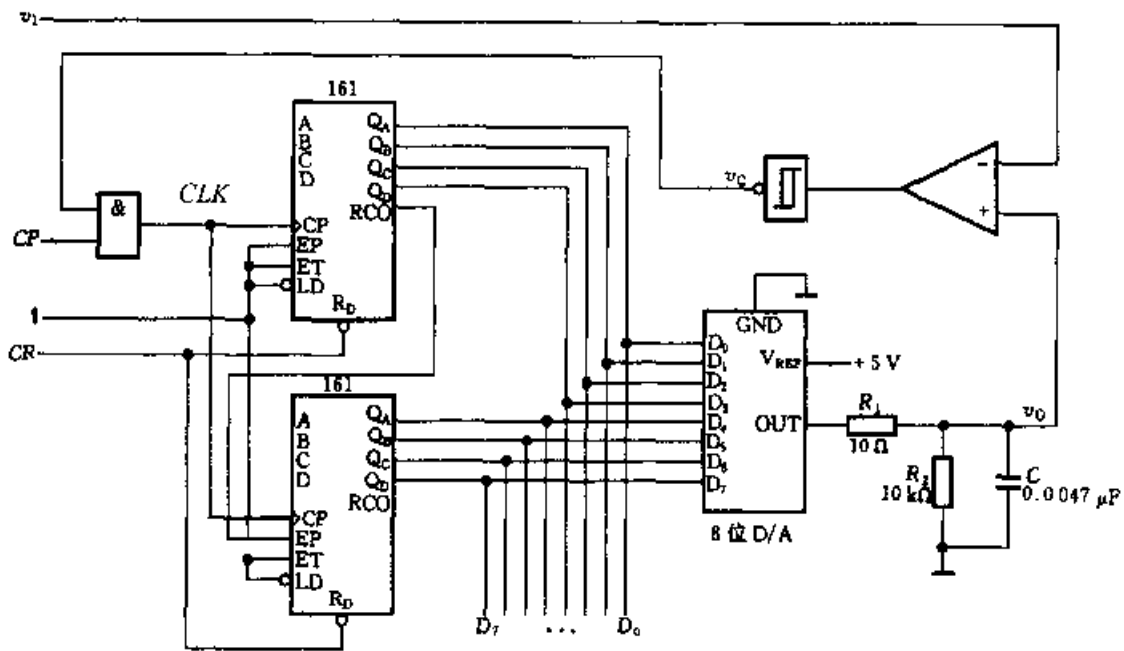
* CAD 习题

C10.3.1 由计数器和 D/A 转换器组成的波形产生电路如图题 C10.3.1 所示。图中



图题 C10.3.1

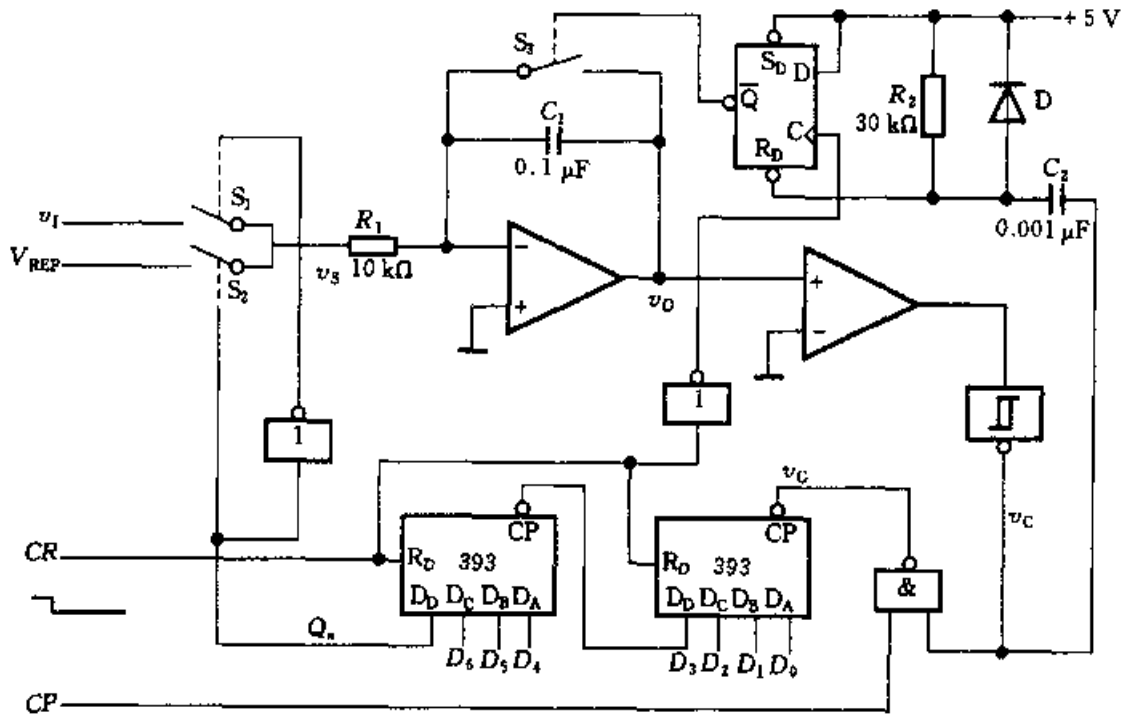
R_1 、 R_2 、 C_1 和 R_3 、 R_4 、 C_2 为滤波电路,用来滤除 D/A 转换器输出波形中的毛刺。已知 CP 的频率为 1 MHz,试用 PSPICE 程序分析,求输出 v_{O1} 和 v_{O2} 的波形。



图题 C10.3.2

C10.3.2 计数型 A/D 转换电路如图题 C10.3.2 所示。已知 CP 的频率为 10 kHz, 应用 PSPICE 程序进行仿真分析, 当 v_1 分别为 2.5 V 和 4 V 时, v_1 、 v_0 、 v_c 、CLK、 $D_7 \sim D_0$ 的波形, 并将转换结果与理论值进行比较。

C10.3.3 双积分 A/D 转换电路如图题 C10.3.3 所示。已知 CP 的频率为 100 kHz, 应用 PSPICE 程序进行仿真分析, 当 v_1 分别为 2 V 和 3.8 V 时, 求 Q_n 、 v_s 、 v_0 、 v_c 和 D_0 的波形, 并将转换结果与理论值进行比较。



图题 C10.3.3

* 11 数字系统设计基础

引言 前面几章讨论了组合逻辑电路和时序逻辑电路的分析和设计方法。这些分析和设计方法是建立在真值表、卡诺图和状态表的基础上,针对基本逻辑单元电路进行的。如果将这些基本逻辑部件组成规模更大,功能更复杂的数字电路时,用经典的逻辑图和逻辑代数方程等进行描述、设计就比较困难,需要采用新的方法来描述和设计。

数字系统的设计方法有两种,即自下而上的设计方法和自上而下的设计方法。本章主要介绍自上而下设计方法中常用的两种工具,即算法状态机 ASM^①图和寄存器传输语言 RTL^②,并介绍数字系统常用的两种实现方法:用中规模集成器件和用可编程逻辑器件实现方法,同时列举简单的实例说明数字系统的设计过程和实现方法。通过讲述这些引导性的内容,以期初学者获得数字系统设计的基础知识和设计技巧。

11.1 数字系统的设计方法

11.1.1 数字系统的组成

数字系统通常由三部分组成:输入接口、输出接口,数据处理器和控制器,如图 11.1.1 所示。

输入、输出接口是用来将模拟量转化为数字量,或将数字量转化为模拟量的模块。处理器的作用是控制系统内各部分模块的工作,使它们按一定顺序进行操作。通常以是否有控制器作为区别功能部件和数字系统的标志,凡是包含控制器且能按顺序进行操作的系统,不论规模大小,一律称为数字系统,否则只能算是一个子系统部件,不能叫做一个独立的数字系统。例如,大容量存储器尽管

① ASM - Algorithmic State Machine

② RTL - Register Transfer Language

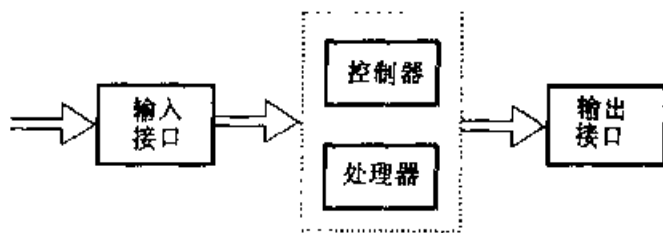


图 11.1.1 数字系统框图

电路规模很大,但也不能称为数字系统。

11.1.2 数字系统的设计方法

数字系统的设计方法可分为两大类,即自下而上的设计方法和自上而下的设计方法。现分别介绍如下:

1. 自下而上的设计方法

数字系统自下而上的设计是一种试探法。设计者根据自己的经验将规模大、功能复杂的数字系统按逻辑功能划分成若干子模块,一直分到这些子模块可以用经典的方法和标准的逻辑功能部件进行设计为止,最后将整个系统安装、调试达到设计要求。

自下而上设计方法具有如下特点:

(1) 这种设计方法没有明显的规律可循,主要依靠设计者的实践经验和熟练的设计技巧,用逐步试探的方法最后设计出一个完整的数字系统。

(2) 系统的各项性能指标只有在系统构成后才能分析测试。如果系统设计存在比较大的问题,也有可能要重新设计,使得设计周期加长、资源浪费也较大。

尽管自下而上的硬件设计方法,在实际运用的一定范围内解决了不少问题,而且目前应用还比较普遍,但随着计算机技术及电子技术的发展,这种设计方法日益显得陈旧,取而代之的是自上而下的设计方法。

2. 自上而下的设计方法

自上而下的设计方法是,将整个系统从逻辑上划分成控制器和处理器两大部分,采用 ASM 图或 RTL 语言来描述控制器和处理器的的工作过程。如果控制器和处理器仍比较复杂,可以在控制器和处理器内部多重地进行逻辑划分,然后选用适当的器件以实现各子系统,最后把它们连接起来,得到所要求的数字系统。

自上而下的设计方法一般要遵循下列几个步骤:

(1) 明确所要设计系统的逻辑功能。

设计题目的叙述通常比较简单,没有细节说明,设计者必须对题目消化、理解,逐步明确系统要完成的逻辑功能。

(2) 确定系统方案与逻辑划分,画出系统方框图。

明确了系统的逻辑要求之后,就要考虑如何来实现设计所规定的逻辑要求。选定实现所需数字系统的原理和方法,进行逻辑划分,确定基本结构,画出系统的方框图。

(3) 采用某种算法描述系统。

(4) 设计控制器和处理器,并组合成所需要的数字系统。

数字系统自上而下的设计有多种方法,本章只重点地介绍算法状态机和寄存器传输语言。

11.1.3 现代数字系统的实现

随着集成电路技术的发展和计算机的应用,数字系统的实现方法也经历了由分立元件、小规模、中规模到大规模、超大规模,直至今天的专用集成电路(ASIC)。现在的 ASIC 芯片规模已经达到几百万个元件。FPGA 或 CPLD 属于 ASIC 电路一类。一个复杂的数字系统只要一片或几片 ASIC 即可实现。制作 ASIC 的方法大体可分为两种,一种是掩膜方法,即由半导体厂家制造;另一种是现场可编程方法,用户可将所设计的电路通过计算机和开发工具,生成关于阵列连接的信息文件,并将信息文件通过编程器“编程”到芯片上。如果采用在系统编程器件,不需要编程器,直接将芯片装在所设计的系统或电路板上,通过编程电缆直接对其编程或修改。

一般可编程逻辑器件的集成软件开发系统,支持两种设计输入方式或两种输入的混合方式:一种是图形设计输入;另一种是硬件描述语言输入,即计算机对输入文件进行编译、综合、优化、适配等操作,最后生成供编程用的 JEDEC^①文件,就可编程到芯片中。

所谓硬件描述语言,就是利用该语言描述硬件电路的功能、信号连接关系及定时关系。它能比电路原理图更有效地表示硬件电路的特性。硬件描述语言在硬件设计领域的作用与 C 或 C++ 在软件设计领域的作用类似。软件语言在某时刻只需执行一条语句,而硬件描述语言可能同时要执行几条语句,因为实际系统中许多操作是并行的,这是它与软件语言的最大区别之一。

硬件描述语言有很多种,现在比较流行的硬件描述语言有 ABEL^② 和 VHDL^③。比较而言,ABEL 是一种用来描述相对简单的数字系统,而 VHDL 则是用来描述更复杂的数字系统。这里主要介绍 ABEL 语言的应用。

① JEDEC——Joint Electron Device Engineering Council

② ABEL——Advanced Boolean Equation Language

③ VHDL——VHSIC Hardware Description Language

11.2 算法状态机

ASM图是描述数字系统控制算法的流程图。应用ASM图设计数字系统,可以很容易将语言描述的设计问题变成时序流程图的描述,只要描述逻辑设计问题的时序流程图一旦形成,状态函数和输出函数就容易获得,从而得出相应的硬件电路。

11.2.1 ASM图符号

ASM图中有三种基本符号,即状态框、判断框和输出框。

1. 状态框

数字系统控制序列中的状态用状态框表示,如图11.2.1a所示。框内标出在此状态下实现的寄存器传输操作和输出,状态的名称置于状态框左上角,分配给状态的二进制代码位于状态框的右上角,图11.2.1b为状态框实例。状态框的名称是 S_1 ,其代码是010,框内规定的寄存器的操作是 $B \leftarrow A$,输出信号是 Z 。图11.2.1中的箭头表示系统状态的流向,在时钟脉冲触发沿的触发下,系统进入状态 S_1 ,在下一个时钟脉冲触发沿的触发下,系统离开状态 S_1 ,因此一个状态框占用一个时钟脉冲周期。

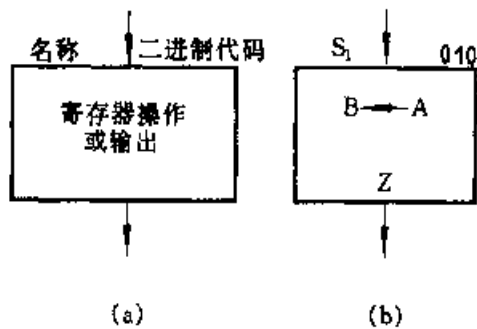


图 11.2.1 状态框与实例
(a)状态框 (b)状态框实例

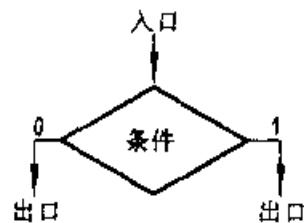


图 11.2.2 判断框

2. 判断框

判断框表示状态变量对控制器工作的影响,如图11.2.2所示。它有一个入口和多个出口,框内填判断条件,如果条件是真,选择一个出口,若条件是假,选择另一个出口。

判断框的入口来自某一个状态框,在该状态占用的一个时钟周期内,根据判断框中的条件,以决定下一个时钟脉冲触发沿来到时,该状态从判断框的哪个出口出去,因此,判断框不占用时间。

3. 条件输出框

条件输出框如图 11.2.3a 所示,条件框的入口必定与判断框的输出相连。列在条件框内的寄存器操作或输出是在给定的状态下,满足判断条件才发生的。在图 11.2.3b 的例子中,当系统处于状态 S_1 时,若条件 $X=1$,则寄存器 R 被清 0,否则 R 保持不变;不论 X 为何值,系统的下一个状态都是 S_2 。

4. 各种逻辑框之间的时间关系

从表面上看,ASM 图与程序流程图很相似,但实际上有很大差异。程序流程图只表示事件发生的先后顺序,没有时间概念,而 ASM 图则不同,它表示事件的精确时间间隔顺序。在 ASM 图中,每一个状态框表示一个时间周期内的系统状态,状态框和与之相连的判断框、条件输出框所规定的操作,都是在一个共同的时钟周期实现的,同时系统的控制器从现在状态(现态)转移到下一个状态(次态)。图 11.2.4 给出了 ASM 图的各种操作及状态转换的时间图。假设系统中所有触发器都是上升沿触发,在第一个时钟脉冲上升沿来到时,系统转换到 S_0 状态,随后根据条件由判断框输出 1(真)或 0(假),以便在下一个时钟脉冲上升沿到达时,系统的状态由 S_0 转换到 S_1 、 S_2 和 S_3 中的一个。

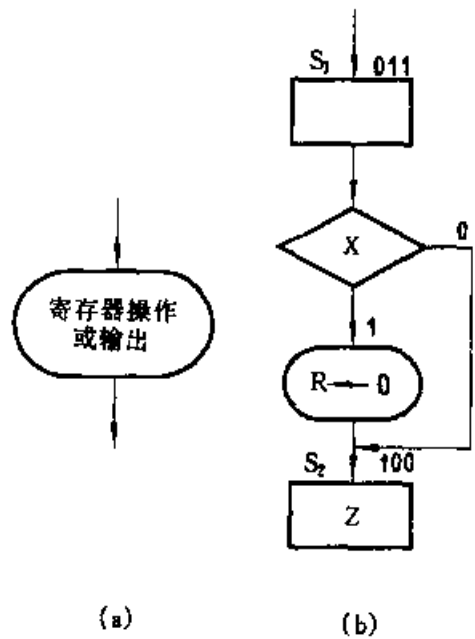
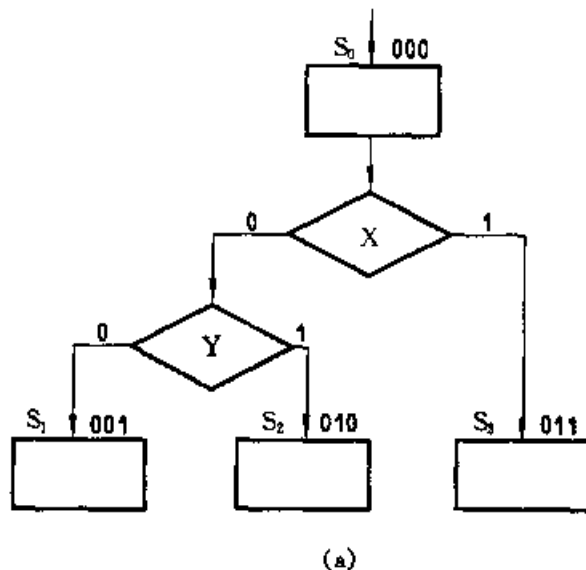


图 11.2.3 条件输出框与实例
(a)条件输出框 (b)实例



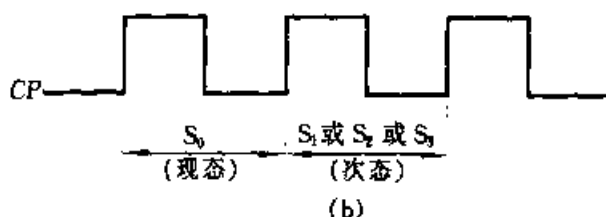


图 11.2.4 ASM 图与状态的时间图

(a) ASM 图 (b) 状态的时间图

11.2.2 数字系统的 ASM 图法设计举例

现以一个交通灯控制系统为例,说明 ASM 图法的设计过程。至于系统的电路实现方法很多,这里只介绍两种方法,一种是用中规模集成电路来实现,另一种是用可编程逻辑器件来实现,将在 11.4 节介绍。

例 11.2.1 设计一个主干道和支干道十字路口的交通灯控制电路,其技术要求如下:

- (1) 一般情况下,保持主干道畅通,主干道绿灯亮、支干道红灯亮,并且主干道绿灯亮的时间不得少于 60s(秒)。
- (2) 主干道无车,支干道有车,则主干道红灯亮、支干道绿灯亮,但支干道绿灯亮的时间不得超过 30s。
- (3) 每次主干道或支干道绿灯变红灯时,黄灯先亮 5s。

解: 设计过程如下:

1. 明确所要设计的系统的逻辑功能

图 11.2.5 表示位于主干道和支干道的十字路口交通灯系统,支干道两边安装传感器 S,要求优先保证主干道的畅通。平时处于主干道绿灯、支干道红灯的状态。当支干道有车时,传感器发出信号 $S=1$,主干道绿灯先转换成黄灯、再变成红灯,支干道由红灯变成绿灯。如果支干道继续有车通过时,则传感器继续有信号,使支干道保持绿灯亮,但支干道绿灯持续亮的时间不得超过 30s,否则支干道绿灯先转换成黄灯再变成红

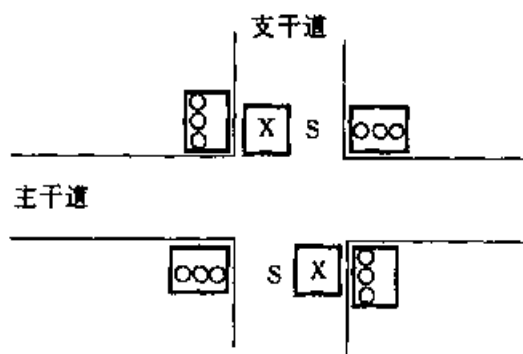


图 11.2.5 交通灯示意图

灯,同时主干道由红灯变成绿灯。主干道每次通行时间不得短于 60s,在此期间,即使支干道有信号 S 输出,也不能中止主干道的绿灯亮。

2. 确定系统方案并画出 ASM 图

系统由控制器和处理器组成,控制器接收外部系统时钟和传感器信号。处理器由定时器和译码显示器组成。定时器能向控制器发出 60s、30s 或 5s 定时信号,译码显示器在控制器的控制下,改变交通灯信号。根据题目要求画出系统框图,如图 11.2.6 所示,其中:

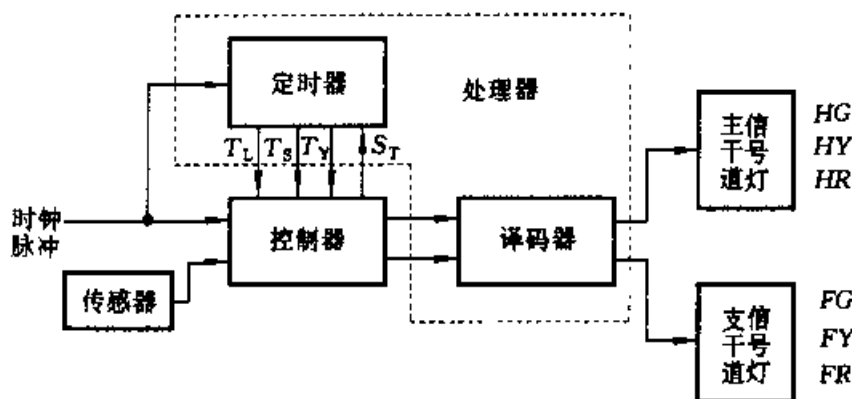


图 11.2.6 交通灯系统框图

HG、HY、HR 分别表示主干道绿、黄、红三色灯。

FG、FY、FR 分别表示支干道绿、黄、红三色灯。

交通灯系统工作主要有三个时间间隔, T_L 、 T_S 和 T_Y , 其中:

T_L : 主干道绿灯亮的最短时间间隔, 不少于 60s。

T_S : 支干道绿灯亮的最长时间间隔, 不多于 30s。

T_Y : 主干道或支干道黄灯亮的时间间隔, 为 5s。

因此, 用定时器分别产生 3 个时间间隔后, 向控制器发出“时间已到”信号, 控制器根据定时器及传感器的信号, 决定是否进行状态转换。如果肯定, 则控制器发出状态转换信号 S_T , 定时器开始清零, 准备重新计时。

交通灯控制器的控制过程分为 4 个阶段, 对应的输出有 4 种状态, 分别用 S_0 、 S_1 、 S_2 和 S_3 表示:

S_0 状态: 主干道绿灯亮, 支干道红灯亮, 此时若支干道有车等待通过, 而且主干道绿灯已亮足规定的时间间隔 T_L , 控制器发出状态转换信号 S_T , 输出从状态 S_0 转换到 S_1 。

S_1 状态: 主干道黄灯亮, 支干道红灯亮, 进入此状态, 黄灯亮足规定的时间间隔 T_Y 时, 控制器发出状态转换信号 S_T , 输出从状态 S_1 转换到 S_2 。

S_2 状态: 支干道绿灯亮, 主干道红灯亮, 若此时支干道继续有车, 则继续保持此状态, 但支干道绿灯亮的时间不得超过 T_S 时间间隔, 否则控制器发出状态转换信号 S_T , 使输出转换到 S_3 状态。

S_3 状态: 支干道黄灯亮, 主干道红灯亮, 此时状态与 S_1 状态持续的时间间

隔相同,均为 T_Y ,时间到时,控制器发出 S_T 信号,输出从状态 S_3 回到 S_0 状态。

对上述 S_0 、 S_1 、 S_2 和 S_3 4 种状态按照格雷码进行编码分别为 00、01、11 和 10,由此得到交通灯控制系统的 ASM 图,如图 11.2.7 所示。设系统的初始状态为主干道绿灯亮、支干道红灯亮,用 S_0 状态框表示。当 S_0 状态持续时间 T_L 大于等于 60 s,并且支干道有车等待通过,传感器 $S=1$ 时,此时满足判断框中的 $T_L \cdot S=1$ 条件,系统控制器发出状态转换信号 S_T ,由条件输出框表示,同时系统从状态 S_0 转到主干道黄灯亮,支干道红灯亮的 S_1 状态。依此类推,得出图 11.2.7 所示的 ASM 的图。

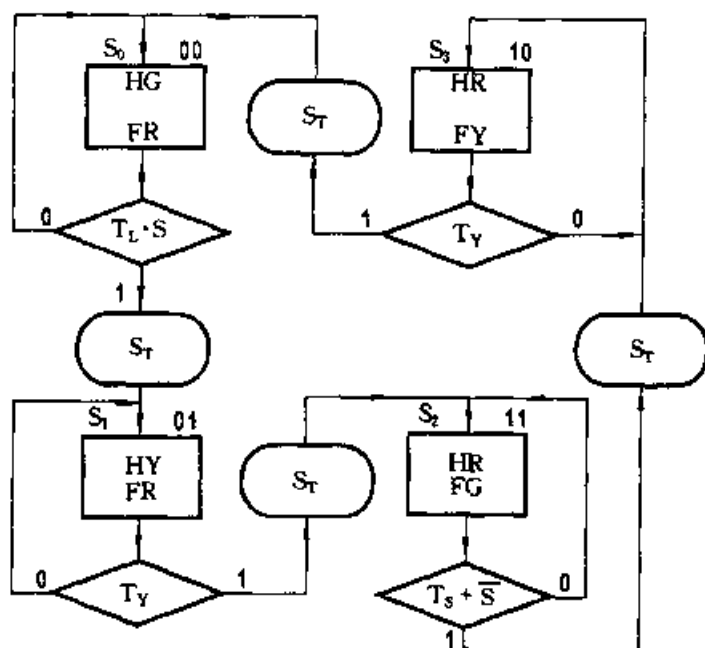


图 11.2.7 交通灯控制器 ASM 图

3. 设计控制器和受控电路

通过分析交通灯控制系统的要求可知,系统主要由传感器、时钟脉冲产生器、定时器、控制器及译码器构成,传感器 S 在有车辆通过时发出一个高电平信号。时钟脉冲产生器由石英晶体组成(见第 9 章)。下面主要介绍用中规模集成电路设计定时器、控制器和译码器。

(1) 定时器

定时器由与系统秒脉冲(由时钟脉冲产生器提供)同步的计数器构成,时钟脉冲上升沿到来时,在控制信号 S_T 作用下,计数器从零开始计数,并向控制器提供模 M_5 、 M_{30} 和 M_{60} 信号,即 T_Y 、 T_S 和 T_L 时间间隔信号。

当系统处于 S_0 状态,为满足主干道绿灯亮,支干道红灯亮的时间间隔 $T_L \geq 60$ s,所以要将 M_{60} 的输出端反馈到计数器的使能端 EN,使它计到 59 时停止

计数,并保持在 $M=60$ 的状态直到支干道有车要通过时,才转换到 S_1 状态。

要求计数器在状态转换信号 S_T 作用下,首先清零,然后开始计数。定时器框图如图 11.2.8 所示。

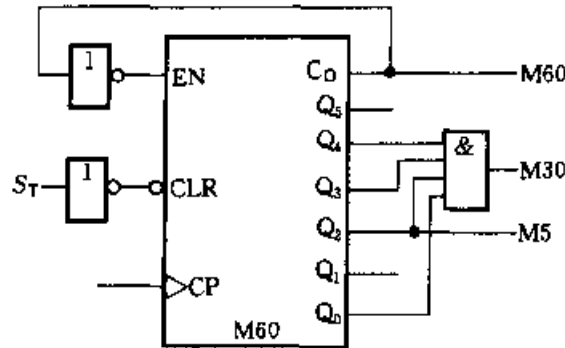


图 11.2.8 定时器框图

计数器具有高电平有效使能端 EN 、低电平有效同步清零端 CLR 和进位输出端 C_0 。当计数到 $Q_5 Q_4 Q_3 Q_2 Q_1 Q_0 = 111011$, 即 $M=60$ 时, $C_0=1$, 将其反相后接入使能端, 就可以保持在 $M=60$ 状态。控制器发出的 S_T 信号是高电平有效, 所以经反相后接至计数器清零端, 具体电路可使用中规模芯片自行设计。

(2) 控制器

从 ASM 图可以列出状态转换表, 如表 11.2.1 所示。选用两个触发器 FF_1 、 FF_0 作为时序寄存器产生 4 种状态, 控制器状态转换的条件为 $T_L \cdot S$ 、 T_Y 和 $T_S + \bar{S}$, 当控制器处于 $Q_1^i Q_0^i = 00$ 状态时, 如果 $T_L \cdot S = 0$, 则控制器保持在 00 状态; 如果 $T_L \cdot S = 1$, 则控制器转换到 $Q_1^{i+1} Q_0^{i+1} = 01$ 状态。这两种情况与条件 T_Y 和 $T_S + \bar{S}$ 无关, 所以用无关项“ \times ”表示。其余情况依此类推, 同时表中列出状态转换信号 S_T 。

表 11.2.1 控制器状态转换表

输入		输出					
现态		状态转换条件			次态		状态转换信号
Q_1^i	Q_0^i	$T_L \cdot S$	T_Y	$T_S + \bar{S}$	Q_1^{i+1}	Q_0^{i+1}	S_T
0	0	0	\times	\times	0	0	0
0	0	1	\times	\times	0	1	1
0	1	\times	0	\times	0	1	0
0	1	\times	1	\times	1	1	1
1	1	\times	\times	0	1	1	0
1	1	\times	\times	1	1	0	1
1	0	\times	0	\times	1	0	0
1	0	\times	1	\times	0	0	1

在表 11.2.1 中,将 Q_1^{n+1} 、 Q_0^{n+1} 和 S_T 为 1 的项所对应的输入或状态转换条件变量相与,其中“1”用原变量表示,“0”用反变量表示,然后将各与项相或,可以推出状态方程和转换信号方程如下:

$$Q_1^{n+1} = \overline{Q_1^n} Q_0^n T_Y + Q_1^n \overline{Q_0^n} \overline{T_Y} + Q_1^n Q_0^n$$

$$Q_0^{n+1} = \overline{Q_1^n} \overline{Q_0^n} T_L \cdot S + \overline{Q_1^n} Q_0^n + Q_1^n Q_0^n \overline{T_S + \overline{S}}$$

$$S_T = \overline{Q_1^n} \overline{Q_0^n} T_L \cdot S + \overline{Q_1^n} Q_0^n T_Y + Q_1^n \overline{Q_0^n} T_Y + Q_1^n Q_0^n (T_S + \overline{S})$$

利用中规模集成电路实现控制器时,控制器有 4 个状态。因此选用两个 D 触发器 FF_1 和 FF_0 ,每个触发器的输入函数用数据选择器选择,所以,数据选择器的数目等于触发器的数目。考虑到控制信号 S_i 与输入变量的关系,也需要用一个数据选择器,所以共选用 3 个数据选择器,并将触发器的现态值加到数据选择器的选择变量端,数据选择器的输入端信号可以根据状态方程和转换信号方程得出。交通灯控制器的逻辑图如图 11.2.9 所示。

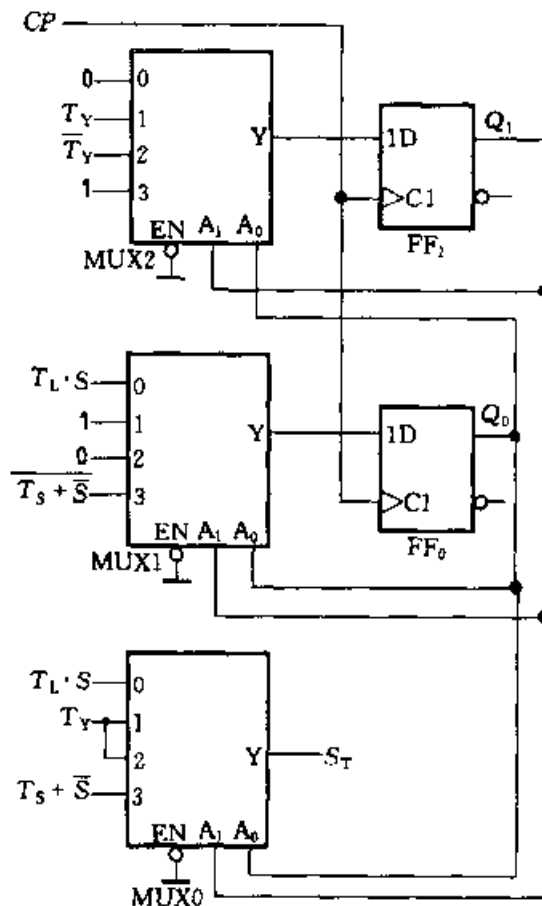


图 11.2.9 交通灯控制器电路

(3) 译码器

系统的输出是在 Q_1 、 Q_0 驱动下的 6 个信号灯,各状态与信号灯的关系由表 11.2.2 给出,因此,得到灯光信号与控制器状态变量的关系为

$$\begin{aligned}
 HG &= \overline{Q_1} \overline{Q_0} & FG &= Q_1 Q_0 \\
 HY &= \overline{Q_1} Q_0 & FY &= Q_1 \overline{Q_0} \\
 HR &= Q_1 & FR &= \overline{Q_1}
 \end{aligned}$$

表 11.2.2 信号灯与控制器状态编码表

状态	HG	HY	HR	FG	FY	FR
S ₀	1	0	0	0	0	1
S ₁	0	1	0	0	0	1
S ₂	0	0	1	1	0	0
S ₃	0	0	1	0	1	0

实现上述关系的译码电路如图 11.2.10 所示。

D 触发器选用 7474, 多路选择器选用 74153, 译码器选用 74155, 外加反相器构成输出高电平有效电路, 计数器选用 74163, 就可构成系统电路。系统逻辑图略。系统工作波形如图 11.2.11 所示。

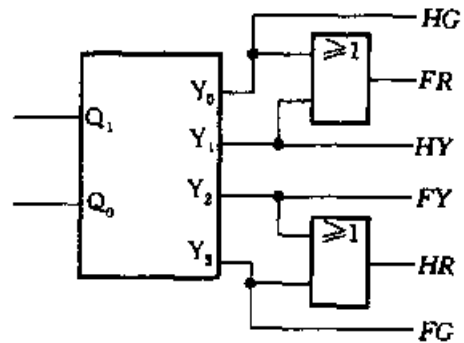


图 11.2.10 灯光信号与状态变量的逻辑电路

例 11.2.2 设计一个 8 位串行数字密码锁, 该锁只有当依次收到的 8 位串行码与规定的二进制数码一致时, 才能被打开。

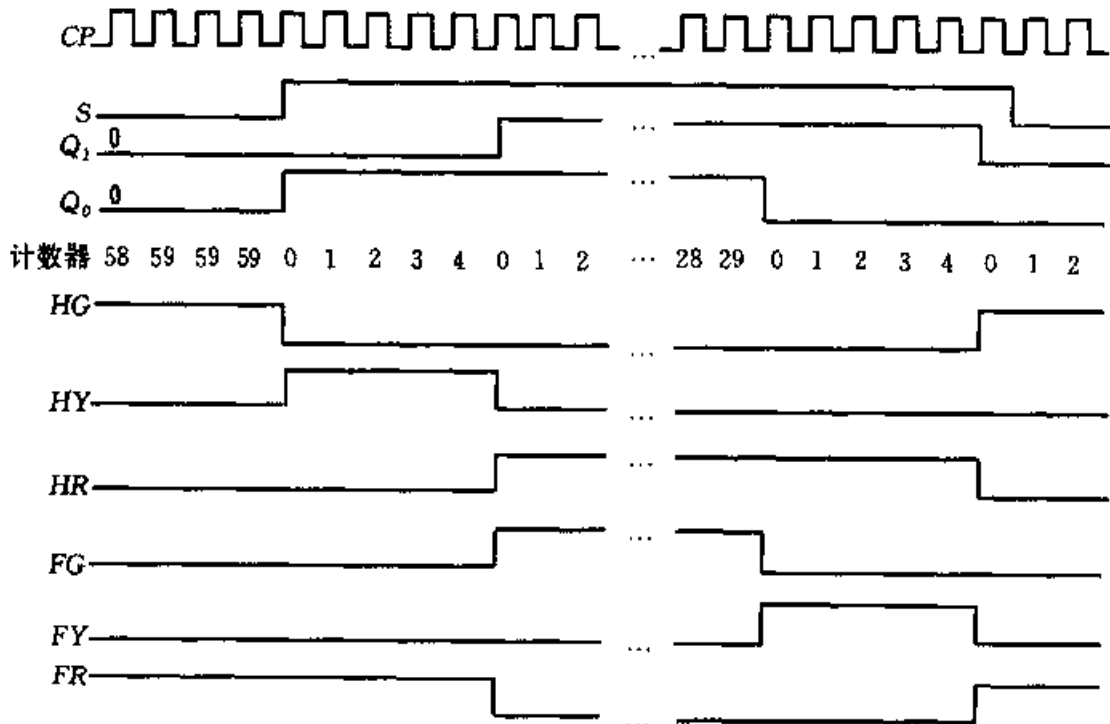
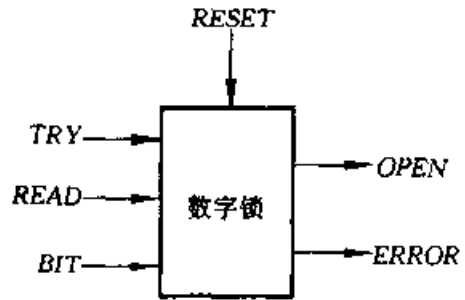


图 11.2.11 交通灯系统工作波形

解：设计过程如下：

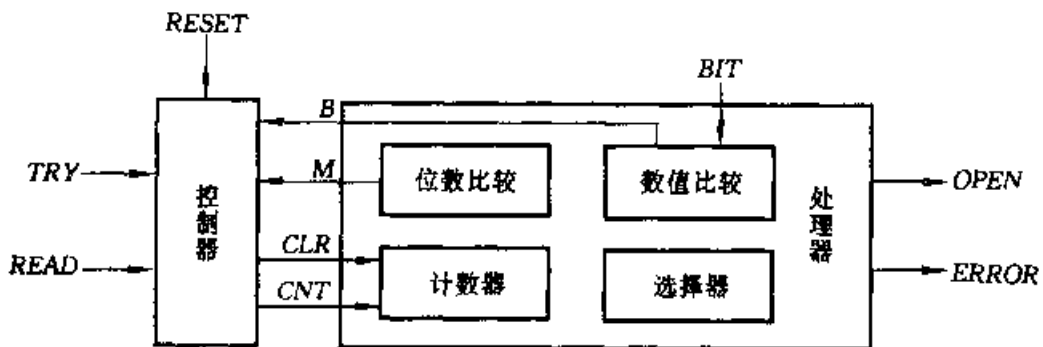
1. 明确系统的设计任务,确定系统的逻辑功能

数字密码锁内部已经设置了8位二进制密码,分别用 D_7 、 D_6 、 D_5 、 D_4 、 D_3 、 D_2 、 D_1 、 D_0 表示。开锁时的串行输入数码由开关 BIT 产生,可以为0或1,如图11.2.12所示。为了使系统能1位1位地依次读取由 BIT 开关送来的串行数码,设置了1个按钮开关 $READ$,送入数码时,首先用 BIT 开关设置1位数码,然后按下 $READ$ 开关,这样就将 BIT 开关产生的当前数码读入系统。为了标识串行数码输入开始和结束,设置了 $RESET$ 和 TRY 按钮开关, $RESET$ 信号使系统进入初始状态,准备接受新的串行数码,当送8位数码与开锁密码一致时,按下 TRY 产生开锁信号,系统便输出 $OPEN$ 信号打开锁,否则数字锁不开,并输出错误信号 $ERROR$ 。



2. 将系统划分为控制器和处理器

将数字密码锁划分为控制器和处理器两部分,如图11.2.13所示。控制器控制处理器接受 BIT 产生的数码,并将其与对应的密码位相比较,比较结果 B 作为状态信息送到控制器。为累计输入数码的位数,需要一个计数器 C ,控制器发出的控制信号 CLR 使计数器清零,并使数码的比较从低位开始,同时计数器开始累计输入数码的次数并与密码的位数相比较,两者相等,则输出一个控制信



号 M 到控制器。图11.2.14给出了处理器逻辑结构图。由8位拨动开关设置的数码作为8选1多路选择器的数码输入,3位二进制计数器的输出作为多路选择器的选择数码输入。多路选择器的输出与 BIT 开关产生的数码相比较,两者相同时输出 B 为1,不同时为0。控制器输出 CNT 和 CLR 控制命令,复位后,控制

器发出 CLR 命令使计数器清零,多路选择器被选择的数码从密码的最低位开始。在控制器 CNT 信号的作用下,从低位到高位逐位被选择出来,控制器根据处理电路反馈回来的 B 状态信息,获得各次比较结果。开锁密码位数的确定由比较器完成,当输入数码的位数为 8 位时,比较器输出 M 为 1,否则为 0。

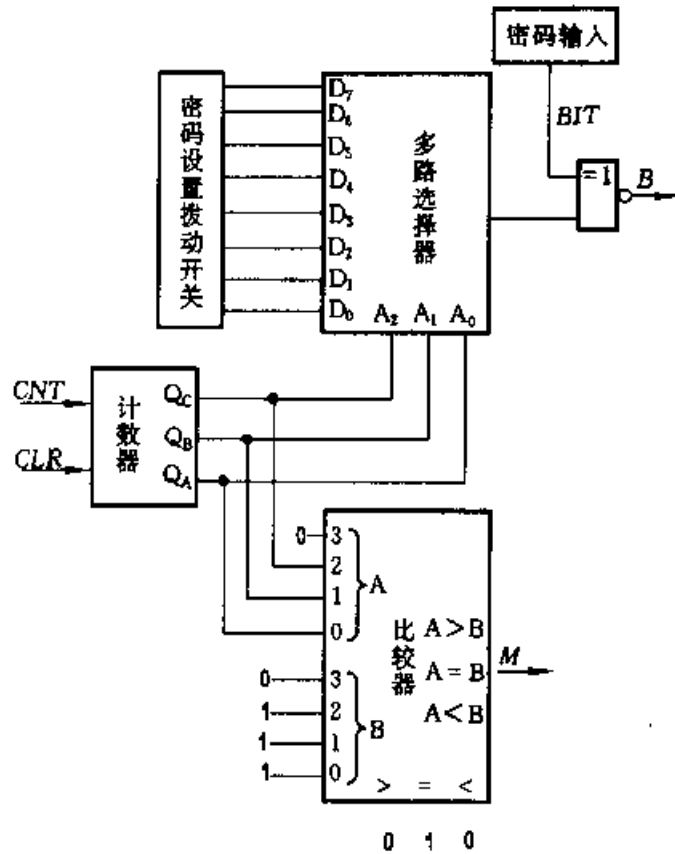


图 11.2.14 处理器逻辑结构图

3. 确定控制器的算法,画出 ASM 图

根据上述分析得出控制器的 ASM 图,如图 11.2.15 所示,其中 S_0 为初始状态, S_1 为接受数码状态, S_2 为准备开锁状态, S_3 状态表示每正确接受一次数码,计数器 C 加 1, S_4 为开锁状态, S_5 为错误状态。异步复位信号 $RESET$ 使得控制器进入初始状态,在该状态时,控制器发出 CLR 命令使计数器 C 清零。下一个时钟到来时,系统无条件地转到接收数码状态 S_1 。在 S_1 状态下,若开锁信号 TRY 为 1 时,控制器进入错误状态 S_5 , TRY 为 0 时,等待接收数码。 $READ$ 为 1 时读取数码,若本次送的数码 BIT 与开锁密码相应位的数值不相等,比较器输出 B 为 0,控制器进入错误状态 S_5 ,若两者数值相等,比较器输出 B 为 1,然后判断位数比较器输出 M 的结果, M 为 0 时控制器进入 S_3 状态。在 S_3 状态,控制器输出 CNT 命令使计数器 C 加 1,然后转换到接收数据状态 S_1 ,这样循环下去,直到正确接收 8 次数码后,位数比较器 M 输出为 1,系统转

入 S_2 状态。

S_2 为准备开锁状态,在 S_2 状态下再读数时,即 $READ$ 为 1 时,控制器进入错误状态 S_5 ,否则等待开锁信号 TRY , TRY 为 1 时,控制器由 S_2 状态进入开锁状态 S_4 ,并输出 $OPEN$ 开锁命令。在上述过程中,任何一次送入的数码与开锁密码的数值不一致,或者 $READ$ 开关和 TRY 开关使用的顺序与规定的不符,都将使控制器进入错误状态 S_5 ,同时控制器发出错误信息 $ERROR$ 。

4. 设计控制器处理器电路

图 11.2.15 所示控制器 ASM 图的硬件实现方法很多,用中规模集成电路实现,可以采用 1 个触发器对应 1 个状态的方法。用 6 个 D 触发器 $FF_0 \sim FF_5$ 输出表示 $S_0 \sim S_5$ 6 种状态,开锁过程中的每一时刻,只能有一个状态为 1,其余状态为 0。根据图 11.2.15 可以列出控制器的状态转换表,如表 11.2.3 所示。以次态是 S_1 状态为例说明列表过程。在图 11.2.15 中有 3 个箭头(S_0 状态、 S_3 状态和 $READ$ 判别框的箭头)最终直接指向 S_1 状态框,说明控制器在 S_0 、 S_1 和 S_3 状态下,根据不同的转换条件作用,可以转换到 S_1 状态。例如,控制器现态为 S_0 ,此时 FF_0 为 1,其余触发器均为 0。在下一个时钟脉冲来到时,不需要其他任何条件,控制器状态将转换到 S_1 状态,所以转换条件为 1,此时 FF_1 为 1,其余触发器为 0;控制器在 S_1 状态时,则 S_1 为 1,当下一个时钟脉冲来到时, TRY 和 $READ$ 均为 0,则控制器保持在 S_1 状态,这样要求 FF_1 的输入为 $S_1 \cdot \overline{TRY} \cdot \overline{READ}$,所以转换条件为 $\overline{TRY} \cdot \overline{REAY}$ 。其余依此类推得出表 11.2.3。表中没有列出 S_0 为次态的情况,因为 S_0 是直接由复位信号建立的,另外为方便起见,没有区别现态和次态的符号。由表 11.2.3 可以写出各触发器

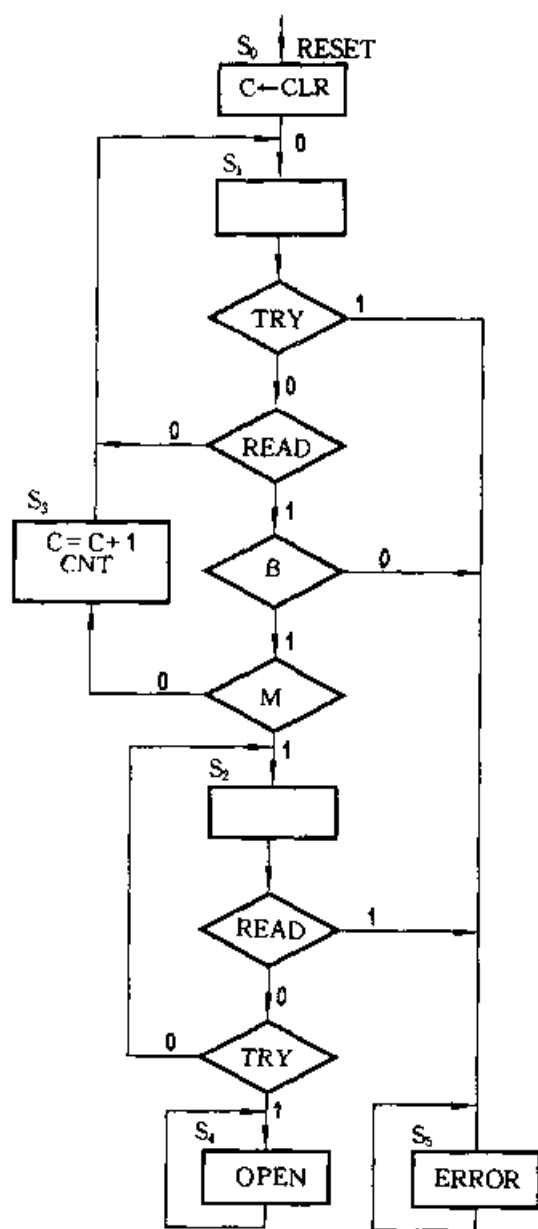


图 11.2.15 控制器 ASM 图

器在 S_0 、 S_1 和 S_3 状态下,根据不同的转换条件作用,可以转换到 S_1 状态。例如,控制器现态为 S_0 ,此时 FF_0 为 1,其余触发器均为 0。在下一个时钟脉冲来到时,不需要其他任何条件,控制器状态将转换到 S_1 状态,所以转换条件为 1,此时 FF_1 为 1,其余触发器为 0;控制器在 S_1 状态时,则 S_1 为 1,当下一个时钟脉冲来到时, TRY 和 $READ$ 均为 0,则控制器保持在 S_1 状态,这样要求 FF_1 的输入为 $S_1 \cdot \overline{TRY} \cdot \overline{READ}$,所以转换条件为 $\overline{TRY} \cdot \overline{REAY}$ 。其余依此类推得出表 11.2.3。表中没有列出 S_0 为次态的情况,因为 S_0 是直接由复位信号建立的,另外为方便起见,没有区别现态和次态的符号。由表 11.2.3 可以写出各触发器

的输入逻辑关系式:

表 11.2.3 控制器状态转换表

输 入		输 出
现态 Q_i^n	转换条件	次态 Q_i^{n+1}
S_0	1	S_1
S_1	$\overline{TRY} \cdot \overline{READ}$	
S_3	1	
S_1	$\overline{TRY} \cdot \overline{READ} \cdot B \cdot M$	S_2
S_2	$\overline{TRY} \cdot \overline{READ}$	
S_1	$\overline{TRY} \cdot \overline{READ} \cdot B \cdot \overline{M}$	S_3
S_2	$\overline{READ} \cdot TRY$	S_4
S_4	1	
S_1	TRY	S_5
S_1	$\overline{TRY} \cdot \overline{READ} \cdot \overline{B}$	
S_2	$READ$	
S_3	1	

$$FF_0: D_0 = 0 \quad (\text{RESET 异步置 } S_0 = 1)$$

$$FF_1: D_1(S_1) = S_0 + S_1 \cdot \overline{TRY} \cdot \overline{READ} + S_3$$

$$FF_2: D_2(S_2) = S_1 \cdot \overline{TRY} \cdot \overline{READ} \cdot B \cdot M + S_2 \cdot \overline{TRY} \cdot \overline{READ}$$

$$FF_3: D_3(S_3) = S_1 \cdot \overline{TRY} \cdot \overline{READ} \cdot B \cdot \overline{M}$$

$$FF_4: D_4(S_4) = S_2 \cdot \overline{READ} \cdot TRY + S_4$$

$$FF_5: D_5(S_5) = S_1 \cdot TRY + S_1 \cdot \overline{TRY} \cdot \overline{READ} \cdot \overline{B} + S_2 \cdot READ + S_3$$

$$= S_1 \cdot TRY + S_1 \cdot \overline{READ} \cdot \overline{B} + S_2 \cdot READ + S_3$$

由图 11.2.15 可以写出控制器输出信号的逻辑表达式为

$$CLR = \overline{S_0} \text{ (低电平有效)}$$

$$CNT = S_3$$

$$OPEN = S_4$$

$$ERROR = S_5$$

根据上述逻辑方程得出数字锁的控制器电路,如图 11.2.16 所示。D 触发器选用 7474,与门选用 7408,或门选用 7432。在图 11.2.14 给出了处理器电路的结构图,其中多路选择器选用 74151,比较器选用 7485,计数器选用 74163,同或门可以用门电路构成。将控制器和处理器组合起来就可以构成数字密码锁系统,数字锁系统电路图略。

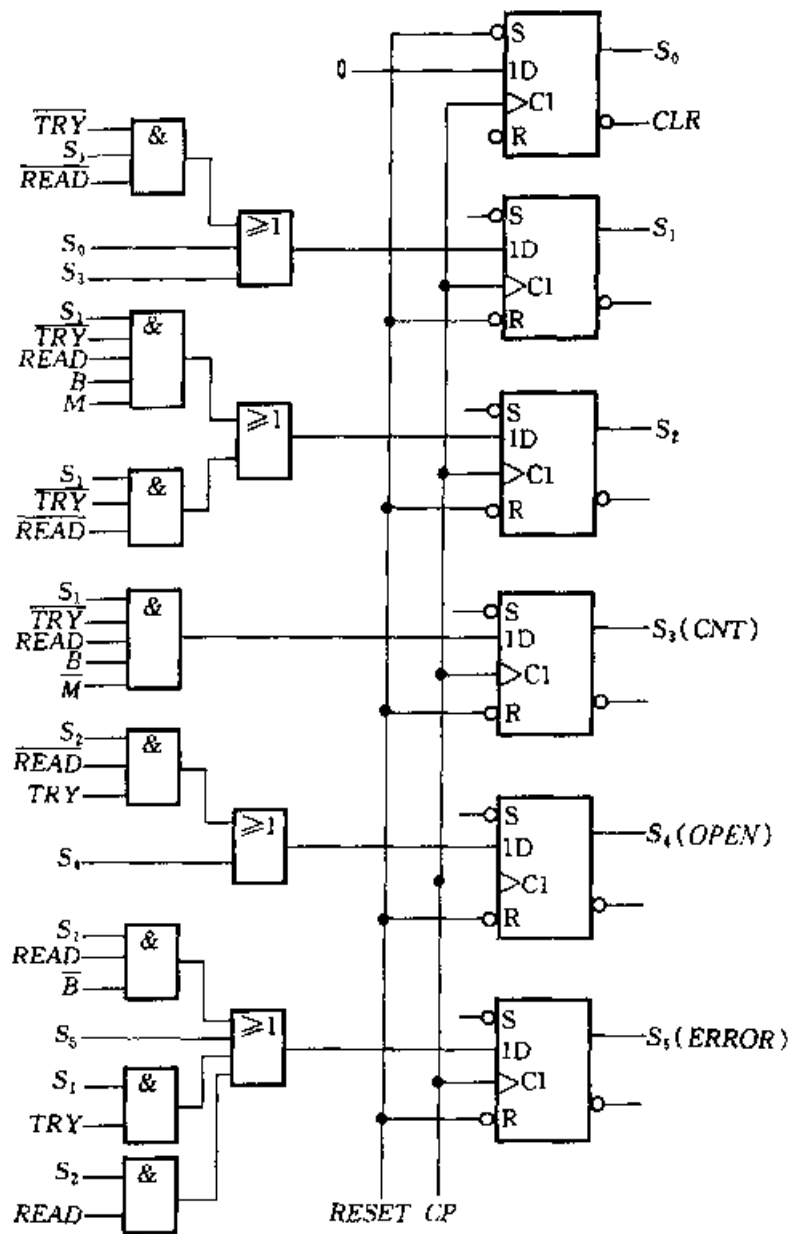


图 11.2.16 数字锁控制器电路

复习思考题

11.2.1 ASM图的条件分支与条件输出有什么区别,条件输出与状态输出有什么区别?

11.3 寄存器传输语言

数字系统各模块之间的信息传递,以及模块内部各子模块之间的信息加工、存储与传输操作,不能用组合电路和时序电路中采用的方法进行描述,必须采用

更高级的描述方法,方法之一是寄存器传输语言 RTL,并且这个表示法同硬件之间有着简单的对应关系。

寄存器传输语言中定义的寄存器是广义的,不仅包括暂存信息的寄存器,而且包括移位寄存器、计数器、存储器以及其他类型的寄存器。计数器是具有递增/递减功能的寄存器,存储器则是寄存器的集合。

寄存器传输语言适于描述功能部件级的数字系统的工作,用这种语言使系统技术要求与硬件电路实现之间建立了一一对应的关系。在寄存器传输语言中,一个语句标号对应于时序流程图中的一个状态框,条件转移语句对应时序流程图中的判断框,用无条件转移语句描述状态之间的无条件转移等等。因此用寄存器传输语言能够简明、精确地描述系统内信息的传送和处理,在数字系统中最常用到的操作有以下4类:

- (1) 寄存器相互传送操作;
- (2) 算术运算操作;
- (3) 逻辑运算操作;
- (4) 移位操作。

11.3.1 寄存器传输语言中的几种操作

1. 寄存器间相互传送操作

在寄存器传输语言中,用大写英文字母表示寄存器,例如 A、B、R1 和 IR 等等,也可以用图形表示一个寄存器,如图 11.3.1a 所示。还可以把寄存器的每一位都表示出来,如图 11.3.1b 所示,其中方括号内是每一位的编号,例如 A[2] 表示寄存 A 的右起第 3 位。

(1) 无条件传送语句

其语句为: $A \leftarrow B$

箭头表示传送方向, B 称为源寄存器, A 称为目的寄存器,在图 11.3.1b 中, A、B 是两个 3 位寄存器,当 A 寄存器加入时钟脉冲 CP 时, B 寄存器中各触发器的内容对应传给 A 寄存器的各个触发器中,其含义是:

$A[2] \leftarrow B[2]; A[1] \leftarrow B[1]; A[0] \leftarrow B[0];$

传送操作是个复制过程,不改变源寄存器的内容,通常不是伴随着每个时钟脉冲都发生这种传输,而是在一定条件下发生。

(2) 条件传送语句

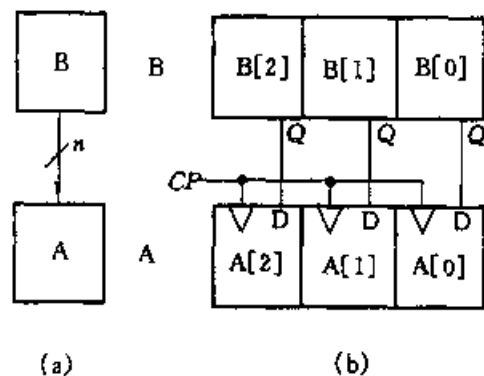


图 11.3.1 寄存器传输操作

(a) $A \leftarrow B$ 简化逻辑图 (b) $A \leftarrow B$ 的逻辑图

传送条件常由控制器给出的逻辑函数规定,例如

$$A \leftarrow (B!C) * (\bar{D}, D)$$

其中“!”和“,”是条件语句专用符,“!”称为隔离符,其意义是隔离符左边和右边的数据之间没有联系。“*”是条件语句的连接符,“*”号右边是传送条件,上述语句表示

若 $\bar{D}=1, D=0$, 则执行 $A \leftarrow B$

若 $\bar{D}=0, D=1$, 则执行 $A \leftarrow C$

又例如: $(B!C) * (F_1, F_2) \leftarrow A$

上式表示:若 $F_1=1, F_2=0$ 则执行 $B \leftarrow A$

若 $F_1=0, F_2=1$ 则执行 $C \leftarrow A$

若 $F_1=1, F_2=1$ 则执行 $B \leftarrow A \quad C \leftarrow A$;

若 $F_1=0, F_2=0$ 则 A 不传送到 B, 也不传送到 C。

2. 算术运算操作

基本的算术操作是加、减、取反和移位,根据基本的算术操作,可获得其他的算术操作,如表 11.3.1 所示。

表 11.3.1 算术操作

符号表示法	说明
$F \leftarrow A + B$	A 与 B 之和传输给 F
$F \leftarrow A - B$	A 与 B 之差传输给 F
$B \leftarrow \bar{B} + 1$	求寄存器 B 存数的补码
$B \leftarrow \bar{B}$	求寄存器 B 存数的反码
$F \leftarrow A + \bar{B} + 1$	A 加 B 的补码传输给 F
$A \leftarrow A + 1$	A 加 1
$A \leftarrow A - 1$	A 减 1

例如 $F \leftarrow A + B$

表示 A 寄存器内容加上 B 寄存器内容,其和传送给 F 寄存器。

3. 逻辑操作

逻辑操作是两个寄存器对应位之间的操作,为了区别于算术运算符号,这里的与、或、非分别用符号 \wedge 、 \vee 和字母上方加一横表示,例如 $F \leftarrow A \vee B$ 表示寄存器 A 和寄存器 B 对应位进行“或”操作,其结果传输给寄存器 F,如图 11.3.2 所示,逻辑操作的符号表示法示于表 11.3.2。

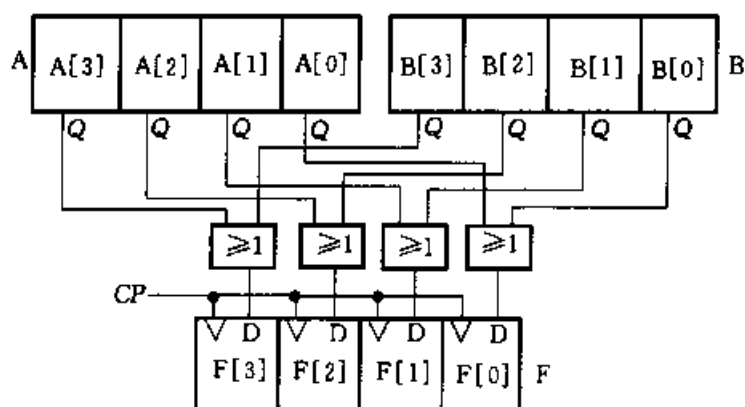


图 11.3.2 “或”操作

表 11.3.2 逻辑操作

符号表示法	说明
$F \leftarrow \bar{A}$	“非”操作
$F \leftarrow A \vee B$	“或”操作
$F \leftarrow A \wedge B$	“与”操作
$F \leftarrow A \oplus B$	“异或”操作

4. 移位操作

移位操作分为左移,右移两种,简要介绍如下:

(1) 左移操作

语句 $A \leftarrow \text{shl}A, A[0] \leftarrow 0$

寄存器 A 各位左移 1 位,丢弃最高位,0 移至 A 的最低位。

(2) 右移操作

语句 $A \leftarrow \text{shr}A, A[3] \leftarrow 0$

寄存器 A 各位右移 1 位,丢弃最低位,0 移至 A 的最高位。

(3) 循环左移操作

语句 $A \leftarrow \text{shl}A, A[0] \leftarrow A[3]$

寄存器 A 的各位左移 1 位,把 A 的最高位移至 A 的最低位。

(4) 循环右移操作

语句 $A \leftarrow \text{shr}A, A[3] \leftarrow A[0]$

寄存器 A 的各位右移 1 位,把 A 的最低位移至 A 的最高位。

5. 输入和输出操作

寄存器传输语言还可以描述系统输入、输出操作,如果将输入线 X 的数据传送到 A 寄存器,则表示为

$$A \leftarrow X$$

把寄存器 A 的各位传送到输出线时,则采用符号“=”表示

$$Z = A$$

该语句意味着寄存器输出与输出线 Z 直接相连。

6. 无条件转移和条件转移

寄存器传输语言中也包含分支语句,分支语句说明下一步要执行多种传送中的哪一种,描述这种操作的语句叫转移语句,通常有无条件转移语句和条件转移语句两种。

(1) 无条件转移语句

→(S)

表示下一步转向编号为 S 的语句继续执行。

(2) 条件转移语句

$(f_1, f_2, f_n)/(S_1, S_2, S_n)$

其中 f 是系统变量的函数,取值为 0 或 1,当 $f_i = 1$ 时,执行 S_i 标号的语句,当所有的 f_i 均为 0 时,顺序执行下一条语句,例如

→(\overline{EN}, EN)/(3, 5)

说明 $EN = 0$ 时,下一步执行语句 3; $EN = 1$ 时,则执行语句 5。

(3) 空操作语句

→NULL

表示不进行任何操作,而是利用它得到一个时钟的延迟时间,然后顺序执行下一条语句。

11.3.2 运用寄存器传输语言设计举例

用寄存器传输语言描述一个数字系统的硬件结构和操作程序叫硬件结构描述过程,简称硬件程序。如果设计的系统比较简单时,可以根据要求直接描写它的硬件程序,如果问题比较复杂,可先画出系统的简单框图或 ASM 图,以帮助编写硬件描述程序。

1. 语言描述

仍以 11.2.2 节的交通灯控制系统为例,说明用寄存器传输语言的设计过程。如图 11.2.7 交通灯控制器的 ASM 图所示,系统共有 4 个状态,每个状态对应寄存器传输语言中的一步,分别用 1、2、3、4 表示,并选用 1 个触发器对应 1 个状态的方法,即 $Q_0 = S_0, Q_1 = S_1, Q_2 = S_2, Q_3 = S_3$,这种 1 个状态使用 1 个触发器的方法,是一种常用方法,虽然使用触发器较多,但触发器的激励函数和输出均变得简单。交通灯控制系统的寄存器传输语言描述如下,ASM 图中的输出用输入、输出语句表示,判断框用条件转移语句表示:

1 $Q_0 = 1; HG = 1; FR = 1$

$\rightarrow (T_L S, \overline{T_L \cdot S}) / (2, 1)$

2 $Q_1 = 1; HY = 1; FR = 1$

$\rightarrow (T_Y, \overline{T_Y}) / (3, 2)$

3 $Q_2 = 1; HR = 1; FG = 1$

$\rightarrow (T_S + \overline{S}, \overline{T_S + \overline{S}}) / (4, 3)$

4 $Q_3 = 1; HR = 1; FY = 1$

$\rightarrow (T_Y, \overline{T_Y}) / (1, 4)$

END

执行语句 1 时, $S_0 = 1, S_1, S_2, S_3$ 均为 0, 其他语句依此类推。故由寄存器传输语言程序写出各输出函数:

$$HG = S_0; HY = S_1; HR = S_2 + S_3;$$

$$FR = S_0 + S_1; FY = S_3; FG = S_2;$$

$$S_T = T_L \cdot S \cdot S_0 + T_Y \cdot S_1 + (T_S + \overline{S}) S_2 + T_Y \cdot S_3$$

当满足状态转换条件时, 控制器发出 S_T 信号。例如在 S_0 状态, 当 $T_L \cdot S = 1$ 时, S_T 为 1, 其他状态依此类推, 得出 S_T 表达式。

以上语句中为方便起见, 均以逻辑代数中的符号“+”及“ \cdot ”代替“ \vee ”和“ \wedge ”符号。

2. 根据寄存器语言的描述来实现电路

(1) 定时器电路

从交通灯控制系统的寄存器传输语言描述中, 可以看到状态的转换主要是由 3 个时间间隔和传感器 S 的条件控制, 如 11.2.2 节所述情况相同, 所以计数器的设计也相同。

(2) 控制器和输出电路

为了使交通灯能按要求的时间顺序进行工作, 必须按严格的次序和时间发送相应的控制信号和时钟脉冲。图 11.3.3 所示为交通灯控制器电路。它由 4 个触发器构成, 它们的时钟脉冲端 C 均直接与系统时钟脉冲信号 CP 相连。每个状态变化由时钟脉冲信号上升沿触发, 它们的输出分别是 S_0, S_1, S_2 和 S_3 , 初始状态由 $Reset$ 端设置, $Reset$ 接至触发器 FF_0 的 S 端和触发器 FF_1, FF_2, FF_3 的 R 端, 以保证初始状态 $S_0 = 1, S_1, S_2, S_3$ 均为 0。

语句 1 的条件转移语句表达式的含义是, 若 $T_L \cdot S = 1$ 时, 则在下一个时钟脉冲上升沿来到时, 转向执行语句 2; 若 $\overline{T_L \cdot S} = 1$ 时, 则在下一个时钟脉冲上升沿来到时仍执行本语句。其他语句类推。当执行到语句 4 后, 若满足条件 $T_Y = 1$, 则在下一时钟脉冲上升沿来到时又回到语句 1, 至此完成了交通灯控制器电

路的设计,将其与定时器电路组合在一起就构成完整的数字系统。

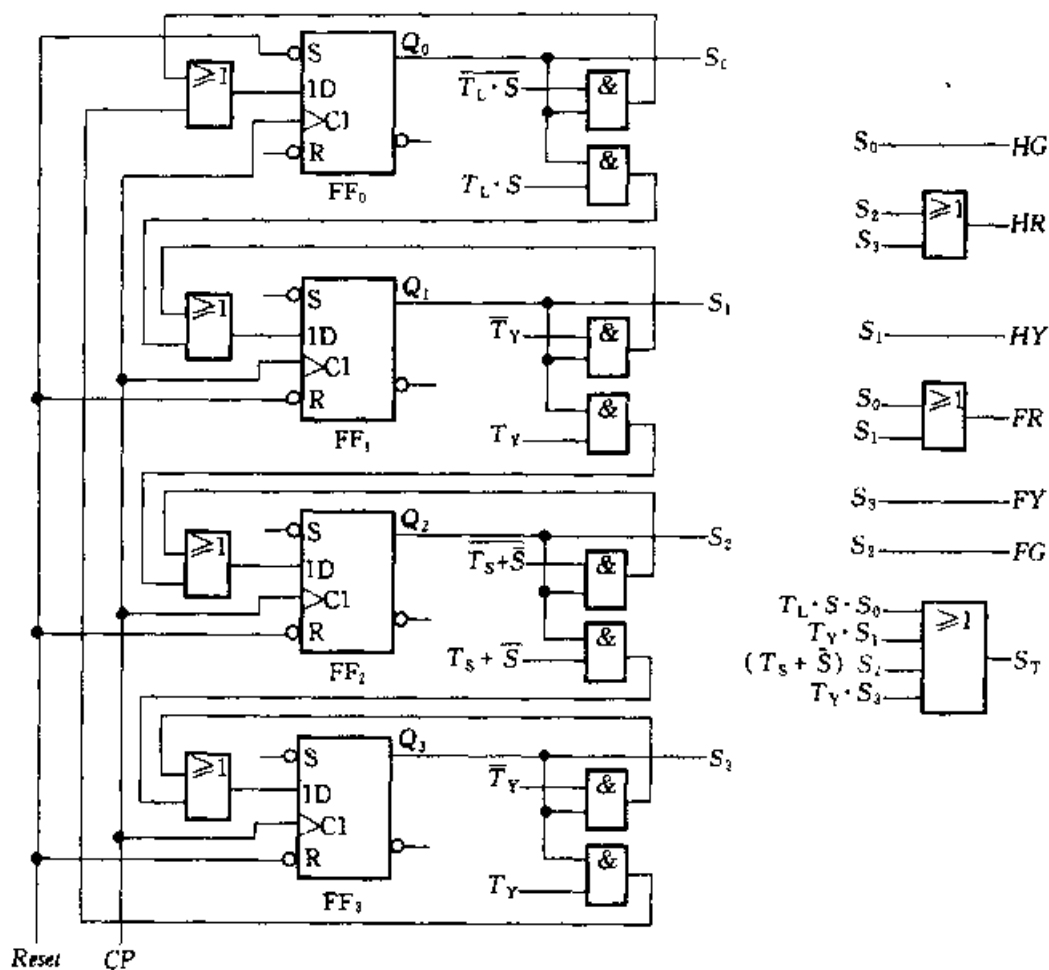


图 11.3.3 交通灯控制器及输出电路

从上述例子可以看到,将数字系统的寄存器传输语言的描述转换成硬件电路是比较简单而直接的,不需要列出状态转换表。硬件程序法比较适合于设计有序传输及处理多位数据的数字系统。如果将用寄存器传输语言设计的电路输入可编程逻辑器件开发系统,同样可得到 JEDEC 文件。

11.4 用可编程逻辑器件实现数字系统

前面两节介绍了数字系统自上而下设计方法的两种常用工具 ASM 图和 RTL。用这两种设计工具可以得到控制器和处理器的逻辑电路。数字系统的实现可以采用中规模集成电路实现,也可以用各种可编程逻辑器件实现。可编程逻辑器件的开发系统支持图形设计输入、硬件描述语言输入或两种输入的混合方式。如果将前面的 ASM 图或 RTL 方法设计的逻辑电路,以图形方式输入可编程器件开发系统,经过编译生成 JEDEC 文件,就可以用可编程逻辑器件实现

系统电路,如果用 ABEL 语言描述计数器、控制器和译码器电路的工作过程,可以简化设计过程,将硬件描述语言直接输入到可编程逻辑器件开发系统进行编译,就得到 JEDEC 文件。

11.4.1 交通灯控制系统的实现

这里采用 Lattice 公司的在系统可编程逻辑器件 ispLSI 1016 芯片,实现交通灯控制系统。根据图 11.2.7 所示的交通灯控制系统的 ASM 图,得出系统状态图,如图 11.4.1 所示。ASM 图中的状态框与状态图中的状态相对应,判断框中的条件是状态转换的输入条件,条件输出框与控制器状态转换的输出相对应。状态图是描述状态之间的转换,例如在 S_0 状态,如果条件 $T_L \cdot S = 1$ 时,系统状态转移到 S_1 ,同时输出状态转换信号 S_T 。如果 $T_L \cdot S = 0$,则系统保持在 S_0 状态。图中未画出 4 种状态下主干道和支干道交通灯的输出情况,它们是 4 种状态的组合输出。 S_T 是控制器的输出信号,用来控制定时器的的工作。

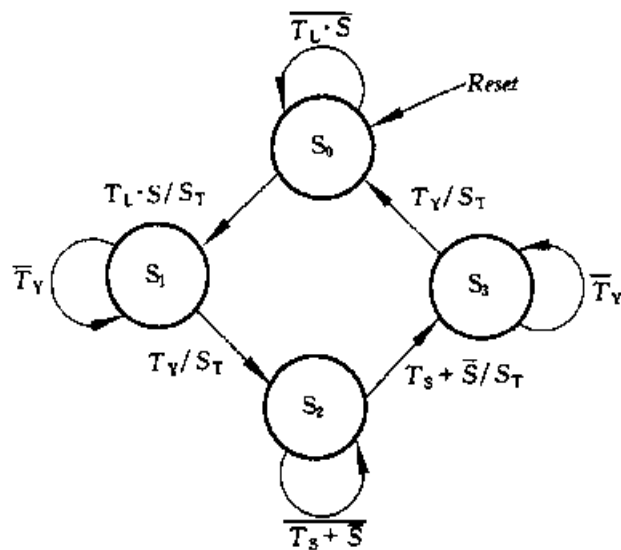


图 11.4.1 交通灯状态图

根据状态图写出交通灯控制系统的 ABEL 语言描述,其中包括控制器、定时器、译码器和测试向量等。测试向量用以测试 ABEL 语言描述的硬件电路是否与实际要求符合。将 ABEL 语言描述的程序输入到可编程逻辑器件开发系统,经过编译就可得到 JEDEC 文件。由此可见,使用硬件描述语言可以大大简化设计过程。

```
MODULE traffic
```

```
TITLE 'traffic lights controlling system'
```

```

"====Declarations===="
"Inputs
  CLK          pin;
  S            pin;
  RESET       pin;
"Outputs
  HG, HY, HR   pin istype 'com';
  FG, FY, FR   pin istype 'com';
"Nodes
  Q1..Q0       node istype 'reg';    "state registers
  Timer5..Timer0 node istype 'reg';  "Module 60 Timer
  Tl, Ts, Ty   node istype 'com';
  St           node istype 'com';
"Set Declarations
  Q = [Q1..Q0];
  Timer = [Timer5..Timer0];
"State Declarations
  S0 = [0,0];   S1 = [0,1];
  S2 = [1,1];   S3 = [1,0];
"Constants Declarations
  X, C = .X..C.;
"====Description of the Controller====
equations
  Q.c = CLK;
  Q.clr = RESET;
state _ diagram Q
  state S0: if (!(S&Tl)) then S0 with St=0;
            else S1 with St=1;
  state S1: if (!Ty) then S1 with St=0;
            else S2 with St=1;
  state S2: if (!(Ts+!S)) then S2 with St=0;
            else S3 with St=1;
  state S3: if (!Ty) then S3 with St=0;
            else S0 with St=1;
"====Description of the timers====

```

equations

```

Timer. c = CLK;
Timer. clr = St # RESET;
when (Timer < 59) then {Timer := Timer. fb + 1; Tl = 0;}
                    else {Timer := Timer. fb; Tl = 1;}
when (Timer < 29) then Ts = 0;
                    else Ts = 1;
when (Timer < 4) then Ty = 0;
                    else Ty = 1;

```

"==== Description of the decoder =====

truth_table (Q -> [HG, HY, HR, FG, FY, FR])

S0 -> [1, 0, 0, 0, 0, 1];

S1 -> [0, 1, 0, 0, 0, 1];

S2 -> [0, 0, 1, 1, 0, 0];

S3 -> [0, 0, 1, 0, 1, 0];

"==== Test Vectors =====

test_vectors

([RESET, CLK, S] -> [Q, Timer, Tl, Ts, Ty, St, HG, HY, HR, FG, FY, FR])

[1, C, 0] -> X;

@repeat 20 {[0, C, 0] -> X;}

@repeat 20 {[0, C, 1] -> X;}

@repeat 10 {[0, C, 0] -> X;}

@repeat 70 {[0, C, 1] -> X;}

END traffic

11.4.2 数字密码锁的实现

以下仍然采用 Lattice 公司的在系统可编程逻辑器件 ispLSI 1016 芯片实现数字密码锁的控制器电路。根据图 11.2.15 所示的数字密码锁控制器的 ASM 图, 得出状态图, 如图 11.4.2 所示, ASM 图中的状态框与状态图中的状态相对应, 判断框中的条件是状态转换的输入条件, 状态框中的输出与状态图中的输出相对应。例如在 S_1 状态接受数码状态, 如果条件 $\overline{TRY} \cdot READ \cdot B \cdot \overline{M}$ 为 1, 系统转到 S_3 状态等待下一个时钟到来时, 再回到 S_1 状态接受下一个数码。

根据状态图写出数字锁系统的 ABEL 语言描述如下, 它由控制器、受控电路和测试向量等几部分组成。测试向量用来测试 ABEL 语言描述的数字锁是否符合实际要求, 因此给出了正确开锁过程和错误开锁过程两种情况, 数字锁系

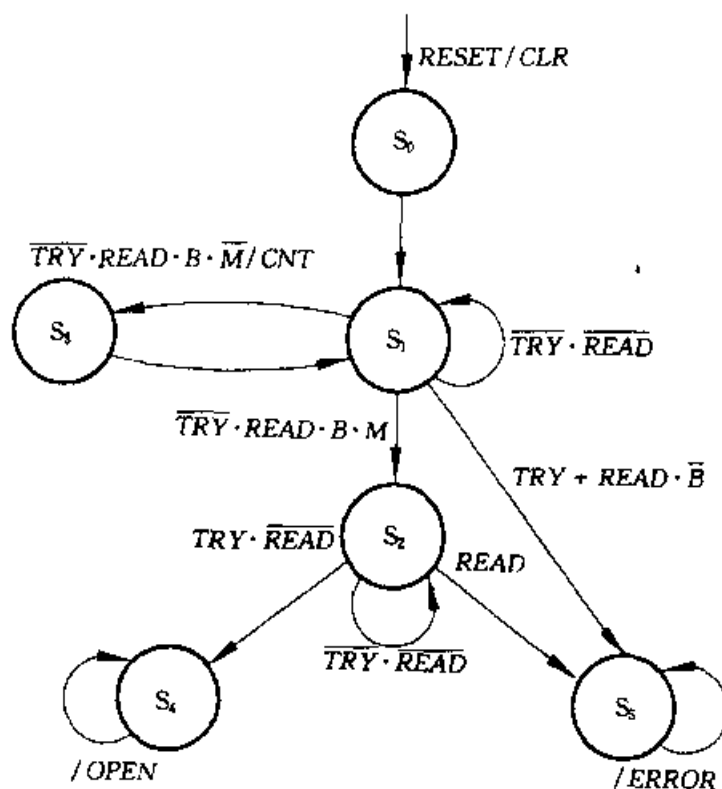


图 11.4.2 数字锁控制器状态图

统仿真波形如图 11.4.3 所示。

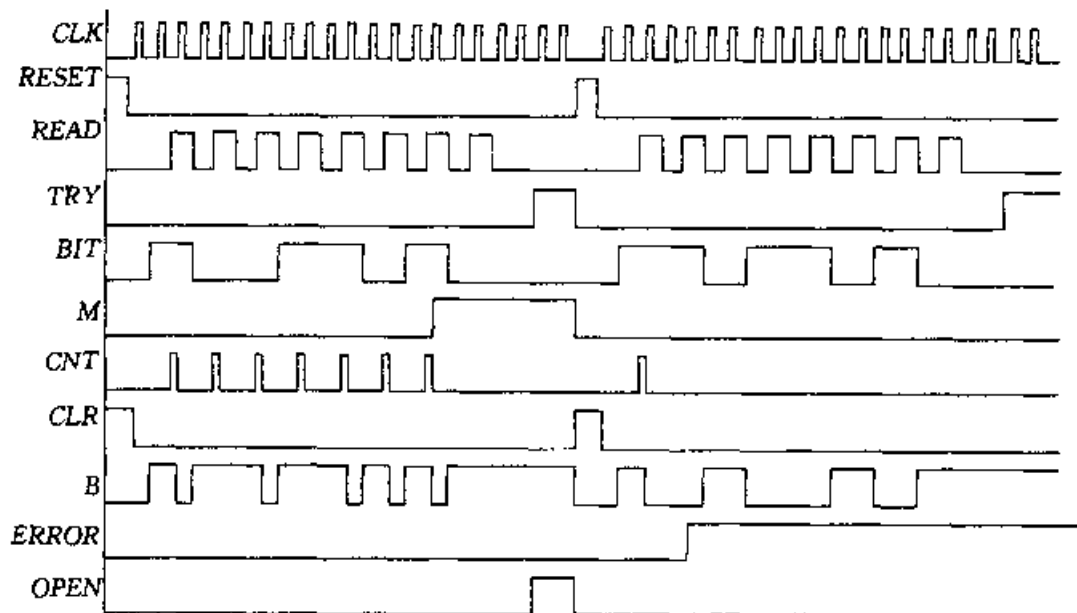


图 11.4.3 数字锁工作波形

MODULE lock

TITLE '8-bits password lock'


```

"====Declarations===="
"Inputs
  RESET          pin;
  TRY            pin;
  READ          pin;
  BIT           pin;
  SET0..SET7    pin;
  CLK           pin;
"Outputs
  OPEN          pin istance 'com';
  ERROR        pin istance 'com';
"Nodes
  B            node istance 'com';
  M            node istance 'com';
  CNT         node istance 'com';
  CLR         node istance 'com';
  STA0..STA2  node istance 'reg';
  Q0..Q2      node istance 'reg';
  BIT_CMP     node istance 'com';
"Set Declarations
  SET_CODE = [SET7..SET0];
  STATES   = [STA2..STA0];
  Q        = [Q2..Q0];
"State Declarations
  S0 = [0,0,0];   S1 = [0,0,1];   S2 = [0,1,0];
  S3 = [0,1,1];   S4 = [1,0,0];   S5 = [1,0,1];
"Constants Declarations
  X,C   = .X.,.C.;
  CODE  = ^b01011001;
"====Description of Controller====
equations
  STATES.c = CLK;
  STATES.ar = RESET;
state _ diagram STATES

```

```

state S0:CLR = 1;goto S1;
state S1:if (!TRY&READ&B&!M = = 1)      then S3 with CNT = 1;
        else if (!TRY&READ&B&M = = 1)    then S2
        else if ((TRY # READ&!B) = = 1)  then S5;
        else if (!TRY&!READ = = 1)      then S1;
state S2:if (TRY&!READ = = 1)           thenS4 with OPEN = 1;
        else if (READ = = 1)             then S5;
        else S2;
state S3:goto S1;
state S4:OPEN = 1;goto S4;
state S5:ERROR = 1;goto S5;
"=====Description of Controlled Circuit =====
equations
  Q.c = CLK;
  Q.ar = CLR;
  when (CNT = = 1)      then Q := Q.fb + 1;
                        else Q := Q.fb;
  when (Q = = 7)       then M = 1;
                        else M = 0;
  BIT_CMP = ((Q = = 0) &SET0) + ((Q = = 1) &SET1) + ((Q = = 2)
&SET2) + ((Q = = 3) &SET3) + ((Q = = 4) &EST4) + ((Q = = 5) &SET5) +
((Q = = 6) &SET6) + ((Q = = 7) &SET7);
  when (BIT_CMP = = BIT) then B = 1;
                        else B = 0;
"=====following test Vector shows a correctly unlocking processing=====
"CODE = 'b01011001;
test_vectors
([RESET, TRY, READ, BIT, SET_CODE, CLK] -> [STATES, Q, OPEN, ER-
ROR, B, M, CNT, CLR, BIT_CMP])
  [ 1, 0, 0 ,X, CODE, X] -> X;
  [ 0, 0, 0 ,0, CODE, C] -> X;

  [ 0, 0, 0 ,1, CODE, C] -> X;
  [ 0, 0, 1 ,1, CODE, C] -> X;

```

[0, 0, 0 ,0, CODE, C] -> X;
 [0, 0, 1 ,0, CODE, C] -> X;

[0, 0, 0 ,0, CODE, C] -> X;
 [0, 0, 1 ,0, CODE, C] -> X;

[0, 0, 0 ,1, CODE, C] -> X;
 [0, 0, 1 ,1, CODE, C] -> X;

[0, 0, 0 ,1, CODE, C] -> X;
 [0, 0, 1 ,1, CODE, C] -> X;

[0, 0, 0 ,0, CODE, C] -> X;
 [0, 0, 1 ,0, CODE, C] -> X;

[0, 0, 0 ,1, CODE, C] -> X;
 [0, 0, 1 ,1, CODE, C] -> X;

[0, 0, 0 ,0, CODE, C] -> X;
 [0, 0, 1 ,0, CODE, C] -> X;

[0, 0, 0 ,0, CODE, C] -> X;
 [0, 0, 0 ,0, CODE, C] -> X;

[0, 1, 0 ,0, CODE, C] -> X;
 [0, 1, 0 ,0, CODE, C] -> X;

"====following test Vector shows a incorrectly unlocking processing====
 test _ vectors

([RESET, TRY, READ, BIT, SET_CODE, CLK] -> [STATES, Q, OPEN, ERROR, B, M, CNT, CLR, BIT_CMP])

[1, 0, 0 ,X, CODE, X] -> X;
 [0, 0, 0 ,0, CODE, C] -> X;

[0, 0, 0 ,1, CODE, C] -> X;
 [0, 0, 1 ,1, CODE, C] -> X;

```

[ 0, 0, 0 ,1, CODE, C] -> X;      "incorrect bit
[ 0, 0, 1 ,1, CODE, C] -> X;

[ 0, 0, 0 ,0, CODE, C] -> X;
[ 0, 0, 1 ,0, CODE, C] -> X;

[ 0, 0, 0 ,1, CODE, C] -> X;
[ 0, 0, 1 ,1, CODE, C] -> X;

[ 0, 0, 0 ,1, CODE, C] -> X;
[ 0, 0, 1 ,1, CODE, C] -> X;

[ 0, 0, 0 ,0, CODE, C] -> X;
[ 0, 0, 1 ,0, CODE, C] -> X;

[ 0, 0, 0 ,1, CODE, C] -> X;
[ 0, 0, 1 ,1, CODE, C] -> X;

[ 0, 0, 0 ,0, CODE, C] -> X;
[ 0, 0, 1 ,0, CODE, C] -> X;

[ 0, 0, 0 ,0, CODE, C] -> X;
[ 0, 0, 0 ,0, CODE, C] -> X;

[ 0, 1, 0 ,0, CODE, C] -> X;
[ 0, 1, 0 ,0, CODE, C] -> X;

```

END lock

小 结

• 数字系统的设计方法有两种,自下而上的设计方法和自上而下的设计方法。现代数字系统的设计常采用自上而下的设计方法。自上而下设计方法的步骤是:明确所要设计系统的逻辑功能→确定系统方案与逻辑划分,画出系统方框图→采用某种算法描述系统→设计控制器和受控电路,并组成数字系统。

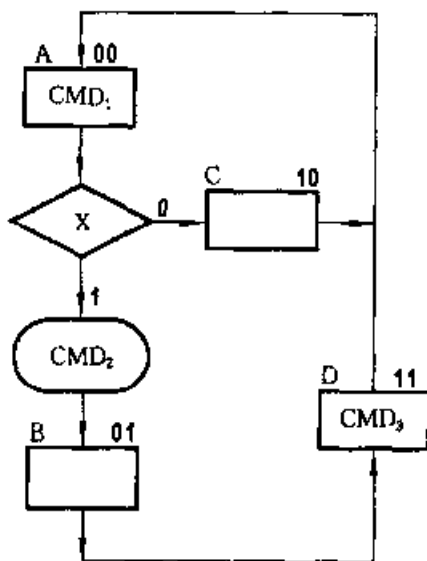
• ASM 图是数字系统自上而下设计方法中经常采用的设计工具,它是指数字系统控制算法的流程图,可以将语言描述的设计问题变成时序流程图的描述,从而获得状态函数或输出函数,得出相应的硬件电路。

• 寄存器传输语言(RTL)也是数字系统自上而下设计方法中采用的设计工具,它可以将系统技术要求与硬件电路实现之间建立一一对应关系,并能完整地描述数字系统内数据的有序流动和处理。

• 数字系统的实现可以采用可编程逻辑器件,用户可以将设计的数字系统,用图形输入方式或硬件描述语言方式或两者的混合方式,输入到开发系统,生成编程用的 JEDEC 文件,编程到芯片中。ABEL 是常用的硬件描述语言,采用 ABEL 语言可以简化数字系统的实现过程。

习 题

11.2.1 用一个触发器一个状态方法实现图题 11.2.1 所示的 ASM 图。



图题 11.2.1

11.3.1 设 $P = 3X - Y$ (X 和 Y 均为不带符号的二进制数)。若 $P = 0$, 将 Y 送到寄存器 C 中。否则将寄存器 C 清零。试用 RTL 表达该系统的工作。

* CAD 习题

11.4.1 设计一个可以控制计数功能的计数器,当 $X = 0$ 时,计数器以二进制数序列 000, 001, 010, 011, 100, 101, 110, 111 计数并重复。当 $X = 1$ 时,计数器以格雷码序列 000, 001, 011, 010, 110, 111, 101, 100 计数并重复,要求:

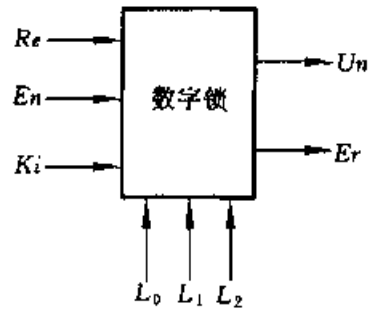
(i) 画出计数器的状态图和 ASM 图;

(2) 用 ABEL 语言描述系统的工作过程。

11.4.2 设计一个简单的 3 位二进制数字密码锁, 控制房门的打开, 当接收到的串行输入数码与原设定的密码相同时, 发出开锁信号, 锁被打开。要求:

- (1) 画出系统的 ASM 图;
- (2) 画出系统的状态转换图;
- (3) 用 ABEL 语言描述其工作过程。

提示: 数字锁系统的示意图如图题 11.4.2 所示, 其中 L_0 、 L_1 、 L_2 为设置密码输入端, 另外 3 个输入端 Re 、 En 、 Ki , Re 是复位端, 当按下 Re 信号 $Re = 1$, 系统复位进入初始状态。 En 是控制开关, Ki 是数据输入端, 当按一下 En 键 $En = 1$ 时, 从 Ki 端输入一个数码, 再按一下 En 键, 再送一个数码, 直到将 3 个数码送完为止。每输入一个数据都要同原设定的相应位密码进行比较。若相等则准备接收下一位数码, 若不相等系统应进入错误状态。输入数码的位数也是开锁的条件。只有每 1 位输入数码的位数和位值与相应密码都相等时, 系统发出开锁信号, 锁被打开。为保密起见, 中间错误状态不显示, 并且不能返回初态, 直到 3 个数码全部送完为止。



图题 11.4.2

附录 A 美国标准信息交换码 (ASCII)

ASCII 采用 7 位 ($b_6b_5b_4b_3b_2b_1b_0$), 可以表示 $2^7 = 128$ 个符号, 如附表 A.1 所示, 任何符号或控制功能都由高 3 位 $b_6b_5b_4$ 和低 4 位 $b_3b_2b_1b_0$ 确定。对所有控制符, 有 $b_6b_5 = 00$, 而对其他符号, 则有 $b_6b_5 = 01$, 或 10 , 或 11 。

附表 A.1

b_3	b_2	b_1	b_0	$b_6b_5 = 00$		$b_6b_5 = 01$		$b_6b_5 = 10$		$b_6b_5 = 11$	
				$b_4 = 0$	$b_4 = 1$	$b_4 = 0$	$b_4 = 1$	$b_4 = 0$	$b_4 = 1$	$b_4 = 0$	$b_4 = 1$
0	0	0	0	控制符		间隔	0	@	P		p
0	0	0	1			!	1	A	Q	a	q
0	0	1	0			"	2	B	R	b	r
0	0	1	1			#	3	C	S	c	s
0	1	0	0			\$	4	D	T	d	t
0	1	0	1			%	5	E	U	e	u
0	1	1	0			&	6	F	V	f	v
0	1	1	1			'	7	G	W	g	w
1	0	0	0			(8	H	X	h	x
1	0	0	1)	9	I	Y	i	y
1	0	1	0			*	:	J	Z	j	z
1	0	1	1			+	;	K	[k	
1	1	0	0			.	<	L	\	l	;
1	1	0	1			-	=	M]	m	'
1	1	1	0			.	>	N	^	n	~
1	1	1	1			/	?	O	_	o	注销

附录 B 二进制数算术运算

二进制数的加、减、乘、除四则运算,在数字系统中是经常遇到的,它们的运算规则与十进制数很相似。加法运算是最基本的一种运算,利用它的运算规则可以实现其他三种运算。例如,减法运算可以借助改变减数的符号再与被减数相加,乘法运算可视为被乘数的连加,而除法则可视为被除数重复地减去除数。

B.1 二进制加法

二进制数加法运算的规则可简单描述如下:

被加数	0	0	1	1	1
加 数	+ 0	+ 1	+ 0	+ 1	+ 1
和	0	1	1	1 0	1 1

B.2 二进制减法

这里先介绍无符号数的减法,其规则如下:

				借入
被减数	0	1	1	1 0
减 数	- 0	- 0	- 1	- 1
差	0	1	0	1

减法运算常用改变减数的符号再相加。通常是利用其补码相加。详见 B.5。

B.3 二进制乘法

二进制乘法与十进制乘法相同,下面列出了 4 条规则:

$$\begin{array}{ll} 0 \times 0 = 0 & 0 \times 1 = 0 \\ 1 \times 0 = 0 & 1 \times 1 = 1 \end{array}$$

B.4 二进制除法

二进制除法与十进制除法相同。

B.5 用带符号位的二进制数以实现减法运算

B.5.1 带符号位的二进制数

一个二进制数既可表示为正数,也可表负数,其方法是在二进制数之前加一符号位。通常用 0 表示正数,而用 1 表示负数,其余数位表示数的大小,例如, $+5 = 0101$, $-5 = 1101$ 。

B.5.2 补码的概念

补码是负数的一种表示方法。现以人们熟悉的十进制数为例来说明补码的概念。

常规减法运算	以 10 为模的减法运算	
$\begin{array}{r} 87 \\ -24 \\ \hline 63 \end{array}$	$\begin{array}{r} 87 \\ -24 \\ \hline \end{array}$	$\begin{array}{r} 87 \\ +76 \\ \hline 163 \end{array}$
		<div style="border: 1px solid black; padding: 2px; display: inline-block;">1 63</div> ↙ 丢弃进位数

可见,将减数 24 变为以 10 为模(称为模 10)的补码 +76,然后相加并丢弃进位数,其结果与常规减法运算相同。模 10 的补码是这样求得的,模数 10 减 1 作为底数 9,然后将减数的每一位数码从底数 9 中减去得到相应位的数码,然后加 1 便得补码。在上例中 $99 - 24 + 1 = 75 + 1 = 76$ 。

B.5.3 二进制数的模 2 补码及减法运算

与模 10 的补码类似,当二进制形成模 2 的补码时,模数 2 减 1 作为底数 1,然后将减数的每一位从底数 1 中减去,得到相应位的数码,然后加 1,便得到补码。例如,011 的模 2 补码为 101。实际上,一种简便的方法是将二进制数码中的 0 变为 1、1 变为 0,再加 1 即可得到模 2 的补码。

常规的减法运算		模 2 补码的减法运算	
被减数	7	0111	0111
减数	-3	-0011	+ 1101
差	4	100	10100
			<div style="border: 1px solid black; padding: 2px; display: inline-block;">1 0100</div> ↙ 丢弃进位

由上可知,模 2 补码的减法运算与常规运算的结果一致。在此例中还需注意到在运算中,符号位参加运算。有关二进制数的四则运算细节,可参阅有关文献[8]、[9]。

附录 C TTL 和 CMOS 逻辑门电路 的技术参数

表 C.1.1

参 数 名 称		类别 (系列)			CMOS	
		74	74LS	74ALS	74HC	74HCT
输入和输出 电 流	$I_{IH(max)}/mA$	0.04	0.02	0.02	0.001	0.001
	$I_{IL(max)}/mA$	1.6	0.4	0.1	0.001	0.001
	$I_{OH(max)}/mA$	0.4	0.4	0.4	4	4
	$I_{OL(max)}/mA$	16	8	8	4	4
输入和输出 电 压	$V_{IH(min)}/V$	2.0	2.0	2.0	3.5	2.0
	$V_{IL(max)}/V$	0.8	0.8	0.8	1.0	0.8
	$V_{OH(min)}/V$	2.4	2.7	2.7	4.9	4.9
	$V_{OL(max)}/V$	0.4	0.5	0.4	0.1	0.1
电源电压	V_{CC} 或 V_{DD}/V	4.75~5.25			2.0~6.0	
平均传输 延迟时间	t_{pd}^*/ns	9.5	8	2.5	10	13
功 耗	P_D^{**}/mW	10	4	2.0	0.8	0.5
扇出数	N_O^{***}	10	20		4000	4000
噪声容限	V_{NL}/V	0.4	0.3	0.4	0.9	0.7
	V_{NH}/V	0.4	0.7	0.7	1.4	2.9

* $t_{pd} = (t_{PLH} + t_{FHL})/2$

** $P_D = [P_{D(静)} + P_{D(动)}]/2$

*** N_O 指带同类门的扇出数。74HC 和 74HCT 的 N_O 均为 4000, 实际上不可能有这么大的数, 因 CMOS 门的输入电容较大, 约为 10pF。本附录的参数引自文献[1], 测量条件为 $V_{CC} = 5V$, $C_L = 15pF$, $R_L = 500\Omega$, $T_a = 25^\circ C$; 对于 74HC 和 74HCT, 测试频率为 1 MHz。更详细的参数, 可查阅有关器件的数据手册。

附录 D 硬件描述语言 ABEL 及其开发软件 ISP Synario

ABEL^① 语言是一种硬件描述语言(HDL^②)。它是开发可编程逻辑器件的一种高级程序设计语言,由美国 DATA I/O 公司推出。该语言支持布尔代数方程、状态图和真值表三种逻辑描述方式,利用其开发软件可以完成逻辑源文件的编辑和一系列的处理工作,并产生编程数据文件(即 JEDEC^③ 文件)。

D.1 硬件描述语言 ABEL

D.1.1 ABEL 语言的基本语法

在用 ABEL 语言进行逻辑设计时,描述逻辑功能的源文件必须是符合 ABEL 语言语法规定的 ASCII 码文件。

ABEL 源文件是由许多语句构成的。语句包括标识符(用户对信号的命名或标号)、字符串、关键字、运算符及表达式等,语句的每一行最长不得超过 150 个字符。

在源文件的语句中,标识符、关键字、数字之间至少要用一个空格将它们隔开;在表达式中,数字与标识符间可以用运算符或括号隔开,空格或圆点不能夹在关键字、数字、运算符及标识符中间。

1. 标识符与关键字

标识符用来表示输入或输出的信号名、引脚名、状态名、集合名、常量、器件名及模块名等等。标识符必须以英文字母或下划线“_”开始,其后可跟数字、英文字母、波浪线“-”和下划线“-”,标识符最长不得超过 31 个字符;标识符与字母的大小写有关,以大写、小写或大小写字母混合方式输入的标识符被看作是不同的标识符。如 En 和 en 是两个不同的标识符。

在 ABEL 语言中,一些具有特殊用途的保留标识符称为关键字。关键字不得用来给器件、引脚、节点、常量、集合、宏或信号命名。关键字可以用大写、小写或大小写混合方式输入,大写、小写或大小写字母混合方式输入的关键字被看作

① ABEL 系 Advanced Boolean Equation Language 的缩写。

② HDL 系 Hardware Description Language 的缩写。

③ JEDEC 系 Joint Electron Device Engineering Council 的缩写。

是同一个关键字。

2. 注释

为便于理解和阅读 ABEL 语言的源文件,用户可用两种方式进行注释:

- (1) 用一对双撇号""括起来或用双撇号"开始,到行尾部结束。
- (2) 用一双斜杠//开始,到行尾部结束。

3. 专用常量

专用常量是 ABEL 语言规定的有特殊含义的常数,用以简化 ABEL 源文件的书写。专用常量的表示法是在英文字母的左下方及右下方各加一个圆点,专用常量字母可以用大写或者小写字母输入,表示的含义相同。常用的专用常量如表 D.1.1 所示。

表 D.1.1 常用的专用常量

常 量	说 明
.C.	时钟脉冲输入(低电平-高电平-低电平转换)
.D.	脉冲下降沿(高电平-低电平转换)
.F.	悬浮的输入或输出信号
.K.	时钟输入(高电平-低电平-高电平转换)
.P.	寄存器预加载
.U.	时钟脉冲上升沿(低电平-高电平转换)
.X.	任意值
.Z.	三态值

4. 数

ABEL 语言的所有数值运算精度为 128 位,即它的数值范围是 $0 \sim 2^{128} - 1$ 。数字可以采用二进制、八进制、十进制、十六进制,分别以符号'b','o','d'及'h(或'B','O','D'及'H)表示,十进制的符号'd'可以省略不写。

在 ABEL 语言中,还可以用英文字母的字符串表示数,方法是在英文字母的前后分别加上单撇号,即以字母的 ASCII 码作为数值,例如:'A' = 'h41', 'AE' = 'h4145'。

5. 运算符和逻辑表达式

ABEL 语言中的运算符和逻辑表达式如表 D.1.2 所示。表达式是标识符和运算符的组合。运算时,按优先级别从高到低依次进行,同一优先级的运算按从左到右的顺序执行。圆括号能改变运算次序,先进行最里层圆括号中的运算。

表 D.1.2 运算符与表达式

类型说明	运算符	表达式举例	说明	运算优先级
算术运算	-	-A	求二进制补码	1(最高)
	-	A-B	减	3
	+	A+B	加	3
	*	A*B	乘	2
	/	A/B	无符号整除	2
	%	A%B	取模:无符号数整除的余数	2
	<<	A<<B	A左移B位	2
	>>	A>>B	A右移B位	2
逻辑运算	!	!A	非	1
	&	A&B	与	2
	#	A#B	或	3
	\$	A\$B	异或	3
	!\$	A!\$B	异或非(同或)	3
关系运算	==	A==B	等于	4(最低)
	!=	A!=B	不等于	4
	<	A<B	小于	4
	<=	A<=B	小于或等于	4
	>	A>B	大于	4
	>=	A>=B	大于或等于	4
赋值运算	=	A=B	组合逻辑赋值(把B的值立即赋给A,没有时延)	
	:=	A:=B	寄存器赋值(在下一个时钟脉冲有效沿到来后,把B的值赋予A)	

6. 字符串

字符串是用单撇号括起来的 ASCII 字符序列,在字符串中允许使用空格。在字符串中若有撇号,则须在其前面加入反斜杠 \,如字符串 'It's easy to use ABEL and Synario' 应写为 'It \ 's easy to use ABEL and Synario'。在字符串中若有反斜杠 \,则要重复两次,即用 \\表示 \。字符串通常在 TITLE、MODULE 和 OPTION 语句以及引脚、节点和属性说明中使用。

7. 块

块是括在大括号 || 内的一段 ASCII 码文本,括号内的文本可以是一行,也可以是多行。块用于方程式、状态图、宏定义和指示字中,块可以嵌套使用,即块中包含块。如果在块的文本字符中包含了大括号,则应在其前加上反斜杠。

8. 集合

集合是作为一个整体参与运算的一组信号或常量。它采用一个标识符,表示方括号内的一组信号或常量,其中的每一个信号或常量称为集合的元素。

定义集合有枚举法、界限符法或它们的组合。例如:

```
Addr = [A7, A6, A5, A4, A3, A2, A1, A0];      为枚举法
Addr = [A7..A0];                             为界限符法
Addr = [A7..A2, A1, A0];                     为两种方法的组合
```

集合可进行表 D.1.2 中除了 * , / , % , << 和 >> 以外的任何一种运算。对集合的运算就是对集合中每个元素单独进行相应的运算,因此如果需要对两个或两个以上的集合进行运算时,这些参与运算的集合元素数目必须相等。

ABEL 语言处理集合时,将它们作为二进制数来对待,二进制数中的每 1 位代表了集合中每一个元素的取值。例如:[A0, A1, A2] = 3 等效于 A0 = 0, A1 = 1, A2 = 1。

D.1.2 ABEL 语言源文件的基本结构

ABEL 源文件由一个或几个相互独立的模块组成,每一个模块包含一个完整的逻辑描述。源文件中所有模块都可以被编译软件同时处理。模块源文件的典型结构是:

```

module 模块名;           "模块开始
[title'标题说明']
Declarations
[器件名 device'器件的工业标号'];
信号名,信号名.....pin[引脚号,引脚号][istype'属性'];
信号名,信号名.....node[istype'属性'];
[常量说明语句;]
[集合的定义;]
[宏定义语句;]
[equations
逻辑方程;
逻辑方程;]
[truth-table(输入变量->输出变量)真值表;]
[state-diagram(状态变量)状态图描述;]
[test-vectors(输入变量->输出变量)测试向量表;]
end 模块名           ;           "模块结束

```

} 说明部分

} 逻辑关系描述部分
(二者取一或其任意组合)

} 测试向量部分

每个模块源文件都以关键字 module 开始,后面跟一个模块名;以关键字 end 后面跟同一个模块名为结束。关键字(以上模块中的英文字都是关键字)可以用大写字母或小写字母,两者的含义相同。用[]括起来的是可选语句,既可以用,

也可以不用。下面对模块中的说明部分、逻辑关系描述部分和测试向量部分分别加以说明。

1. 说明部分

模块的说明部分规定了设计中使用引脚的信号名字及属性、定义常量、宏和状态,模块说明部分的关键字是 Declarations,后面跟定义语句,每个模块都必须有说明部分。

(1) 信号说明

引脚(pin)和节点(node)对设计中使用的信号进行说明,并且引脚或节点的编号同对应的信号联系起来。在将逻辑设计映射到实际的器件之前,引脚和节点的编号可以省略。在引脚或节点的说明中,使用 istype 语句,能将属性赋给信号。常用的信号属性如表 D.1.3 所示。

表 D.1.3 常用的信号属性

属性字符串	意义
'com'	组合型输出
'reg'	寄存器型输出
'invert'	目标器件中的反相器
'buffer'	目标器件中的缓冲器
'neg'	未规定的逻辑为'1'
'pos'	未规定的逻辑为'0'
'keep'	不将此信号从方程式中简化掉
'reg d'	D型触发器
'reg jk'	JK型触发器
'reg sr'	SR型触发器
'reg t'	T型触发器

(2) 常量说明

常量是模块中一直保持常数值的标识符,它用赋值符“=”进行说明。常量说明通常必须放在说明部分。

例如:C,X=.C.,.X.,C是时钟上升沿有效的简化记号,X是任意值的简化记号。

2. 逻辑关系描述部分

逻辑关系描述是每一个模块必须包含的部分,逻辑关系可以用逻辑方程式、真值表和状态图来表示,这三种表示方式既可以单独使用,也可以组合使用。

(1) 方程式语句

关键字 equations 表示逻辑方程开始。一般在 equations 的下一行开始写逻辑方程式,各个逻辑方程必须以分号(;)作为结束符。

方程式中的赋值符为“=”或“:=”，其中“=”为组合逻辑的赋值操作符，“:=”为时序逻辑的赋值操作符。在方程中允许使用 when-then-else 语句。方程的表达形式如下：

```
equations
[when 条件 then][!]变量名 = 表达式;
[else 方程式;]
```

或者

```
[when 条件 then]方程式;
[else 方程式;]
```

例如，用 ABEL 语言中方程的方式描述与非门的逻辑关系如下：

```
equations
X = ! (A&B);“与非门逻辑关系的描述
```

(2) 真值表语句

真值表是 ABEL 语言描述逻辑功能的又一种方法，它用表格的形式描述了输出与输入之间的逻辑关系。真值表以关键字 truth_table 开始，后面跟表头向量和表格，以符号 -> 表示输入、输出关系为组合型，以符号 :> 表示输入、输出关系为寄存器型。

真值表的表达形式为：

```
truth_table (输入 -> 输出)
```

```
    输入 -> 输出;
```

```
    :
```

或 truth_table (输入[:>寄存器型输出] -> 输出;)

```
    :
```

```
    输入[:>寄存器型输出] -> 输出;
```

```
    :
```

例如，采用真值表结构描述一个 2 线 - 4 线译码器，设输入信号为 A, B, 输出信号为 Y0, Y1, Y2, Y3。其方法如下：

```
truth_table ([A, B] -> [Y0, Y1, Y2, Y3])
```

```
    [0, 0] -> [1, 0, 0, 0];
```

```
    [0, 1] -> [0, 1, 0, 0];
```

```
    [1, 0] -> [0, 0, 1, 0];
```

```
    [1, 1] -> [0, 0, 0, 1];
```

(3) 状态图语句

状态图语句专门用于描述时序电路的逻辑功能。ABEL 语言提供 3 种语句来描述状态转移，即无条件转移语句：goto 语句；条件转移语句：if-then-else 语句

和 case-encase。此外,还有条件输出语句:with 语句。在 if-then-else 语句或 goto 语句的后面可以跟 with 语句。注意 if-then-else 语句只能用于状态图的描述,不能用于逻辑方程的描述。

状态图描述以关键字 state _ diagram 为开始,右面是寄存器状态变量的集合,下一行开始写此状态下的无条件输出信号值和状态转移语句,对于每个状态必须有一个转移语句,其表达形式为:

```
state _ diagram 状态寄存器
                [->状态输出]
[state 状态表达式:[方程式]
                [方程式]
                :
状态转移描述]
```

例如,用状态机的结构描述一个模为 4 的同步二进制递增计数器。假设计数器的状态输出用 Q1,Q0 表示,进位输出用 CO 表示,且 $CO=Q1 * Q0$ 。

```
declartions
    Q=[Q1,Q0];           "定义状态寄存器变量
    S0='b00;    S2='b10;   "定义计数器的状态
    S1='b01;    S3='b11;
state _ diagram  Q
    state S0:  goto  S1  with  CO=0;
    state S1:  goto  S2  with  CO=0;
    state S2:  goto  S3  with  CO=0;
    state S3:  goto  S0  with  CO=1;
```

3. 测试向量部分

测试向量用于验证逻辑设计的功能,它通过定义输出为输入的一个函数来指定用户所期望的逻辑运算,以便在仿真时确定逻辑功能是否正确。

测试向量以关键字 test _ vectors 开始,后面跟表头和表格,以符号 -> 连接输入信号与输出信号。若输出为组合逻辑函数,则在输入信号的集合中,不含有时钟输入 CLK,若输出为时序逻辑函数时,则在输入信号的集合中,必须包含时钟输入 CLK。测试向量的表达形式为:

```
test _ vectors[节点]
    (输入信号->输出信号)
    输入值->输出值;
    .....
```

D.2 开发软件 ISP Synario 简介

Synario 是美国 DATA I/O 公司开发的一种运行于 PC 机上的可编程逻辑器件的通用设计与分析软件,它为设计者进行可编程逻辑器件的开发提供了一个集成化的设计环境。它含有图形编辑器、文本编辑器和文件管理系统,还含有编译模块、模拟仿真模块、优化模块等各种处理模块,Synario 的完整工业版还具有 VHDL 和 Verilog 行为描述语言的输入方式,以及功能仿真和定时仿真等功能。

ISP Synario System (简称 ISP Synario)是美国 Lattice 公司和 DATA I/O 公司合作开发的在系统可编程逻辑器件的开发套件。该套件由三部分组成:(1) DATA I/O 公司 Synario 软件中的输入和逻辑功能仿真部分。它包括一个工程项目管理器、原理图编辑器、ABEL-HDL 文本编辑器、编译工具、逻辑功能模拟器和波形显示器。完成设计输入与逻辑功能仿真的任务。(2) Lattice 公司开发的器件适配软件 ispDS+^①Fitter。它主要完成逻辑综合,将逻辑设计适配(布局布线)到设计者选用的具体器件中去的任务,并生成 JEDEC 文件。(3) Lattice 公司开发的菊花链下载编程软件 ispDCD^②。所谓菊花链就是将目标系统板上多个在系统编程器件的编程接口连接成串行的链式结构,编程时能实现一次编程多个器件。通过它,可以将 JEDEC 文件下载到设计者选用的在系统编程器件中去。另外,该软件中还包含可编程开关器件的编译软件等。因此,ISP Synario 是一套完整的电子设计系统,具有设计输入、功能仿真和编程的功能,支持 Lattice 公司生产的(is)pLSI1000 系列、2000 系列的器件和全系列的 GAL 器件,并能将逻辑设计适配到实际的器件中去,能实现在系统编程。

D.2.1 ISP Synario 运行环境

硬件环境:486 以上的 IBM PC 机或兼容机,16M 以上内存,25M 以上的硬盘空间,标准键盘及 SVGA 显示适配器、鼠标、CD-ROM 驱动器。

软件环境:Windows95 或 windowsNT4.0/3.51。

D.2.2 ISP Synario 设计 ISP 器件一般步骤

ISP Synario 软件提供了开发 Lattice 公司在系统可编程逻辑器件所需的全部程序模块。使用它,设计 ISP 器件的流程如图 D.2.1 所示。

1. 输入设计内容

在用开发工具设计可编程逻辑器件时,必须首先分析系统的逻辑功能,将此逻辑功能转化为设计思想,并用合适的形式进行表达,然后输入到计算机中,这

① ispDS+ —— In-system Programmable Development System Plus.

② ispDCD —— In-system Programmable Daisy Chain Download

一过程称为设计输入。设计输入有多种表达方式,常用的有电路图和硬件描述语言两种形式。ISP Synario 软件支持电原理图、ABEL 语言和混合式(原理图+ABEL 语言)输入方式。使用该软件进行设计输入的操作步骤如下:

(1) 启动 ISP Synario

在 Windows 95 桌面上的“开始”菜单中,选择 ISP Synario System|ISP Synario 或选择相应的图标,进入 Synario Project Navigator(工程项目管理器)主程序窗口,如图 D.2.2 所示。

Project Navigator 是该软件的一个前端接口,它把软件的许多程序项集成在一个环境中。窗口顶部第一行为窗口标题信息和当前所设计的工程项目的文

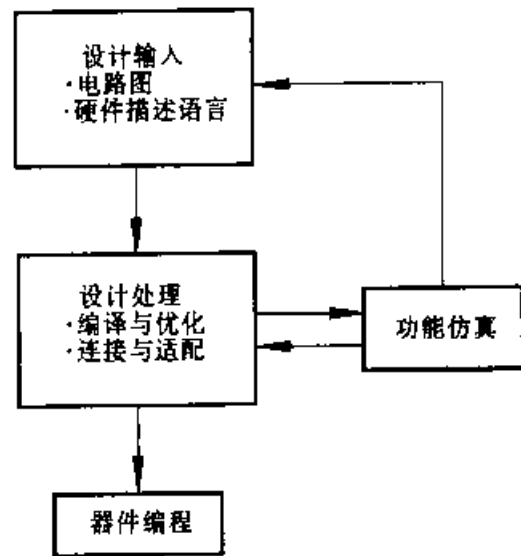


图 D.2.1 ISP 器件设计流程

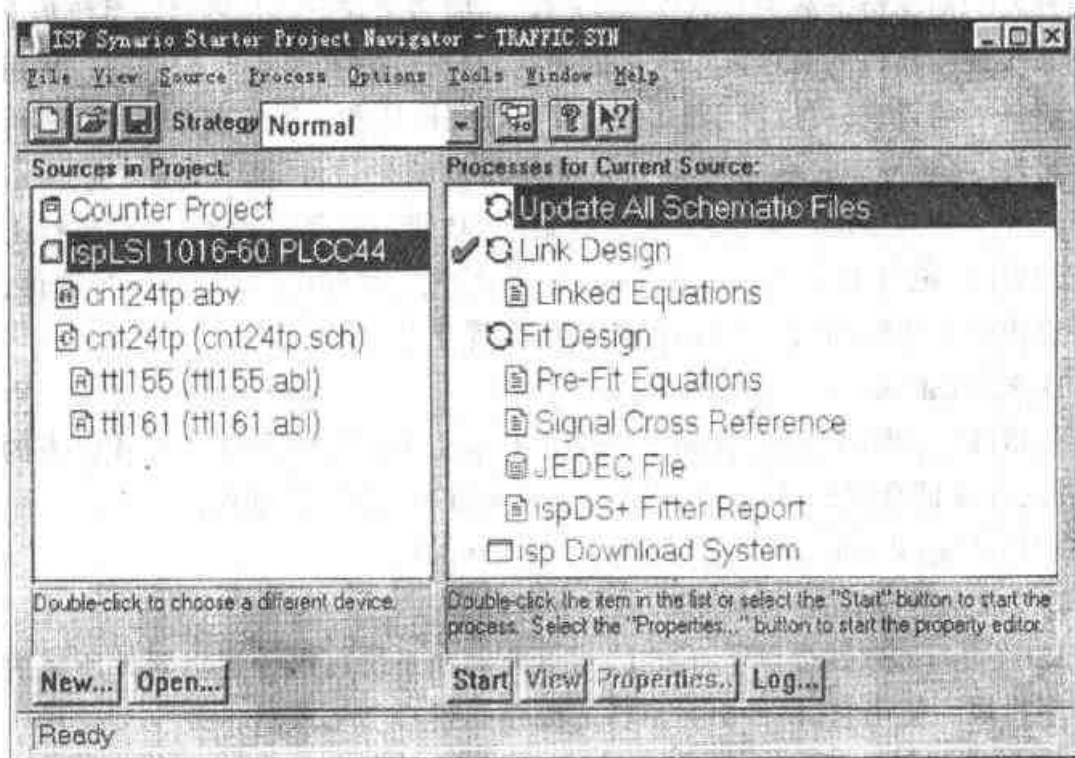


图 D.2.2 工程项目管理器主程序窗口

件名称。Synario 软件引用了工程项目的概念,一个工程项目就是一个设计,每个工程项目最好能对应一个独立的子目录,它包含所有的源文件、中间的数据文件和结束文件。第二行为主菜单栏,当前工程项目中有关源文件的管理及处理

过程的所有操作都可通过选择菜单中相应的命令来完成。

第三行为图标工具栏,从第四行开始,窗口被分成左、右两个窗口,左边是源文件窗口,它显示与当前工程项目相关的所有设计源文件,每一个源文件前都用一个图标表示文件的类型,图 D.2.3 说明了在工程项目管理器中一个项目的典型设计源文件。右边是进程窗口,它显示源文件窗口中被选中的源文件所能进行的所有处理过程。这些处理过程包括:生成网表、编译、逻辑化简、逻辑功能仿真、链接设计、适配设计等操作。每个处理过程前边有两个带箭头的环形标记。左、右窗口的下部都有选中项目的操作提示和快捷工具按钮。整个窗口底部是辅助信息提示栏。由于篇幅所限,这里未能列出各菜单项的功能。








 Counter24 Project	—— 工程标题
 counter.wri	—— 设计说明文档
 ispLSI 1016-60 PLCC44	—— 目标器件
 cnt24top.abv	—— 测试向量源文件
 cnt24top (cnt24top.sch)	—— 顶层电原理图源文件
 cnt10 (cnt10.abl)	—— 底层 ABEL 源文件
 cnt3 (cnt3.abl)	—— 底层 ABEL 源文件

图 D.2.3 典型的设计源文件

(2) 选择 PLD 器件

进行设计输入、逻辑功能仿真时,可以不规定某一特定的器件,但进行适配设计时,要选择目标器件。选择器件时,双击左边窗口源文件中的 Virtual Device(虚拟器件)项目,选择所需的器件,在图 D.2.2 中 ispLSI 1016-60PLCC44 即为所选的器件。

(3) 输入设计内容

选择 File|New Project...建立一个新的工程项目(或者选择 File|Open...,打开一个已存在的工程项目)。

工程项目由电原理图和 ABEL 源文件组成,选择 Source|New 命令,出现源文件输入方式的选择对话框。如果选择原理图输入方式,原理图编辑器立即被打开;调用库中的元件及连线等符号,可以画出电原理图。如果选择 ABEL-HDL Module(模块),则文本编辑器立即被打开,使用 ABEL 语言,可以建立相应的源文件。另外,还有 ABEL Test Vectors(测试向量)和 User Document(用户文档)源文件输入方式,都需要使用文本编辑器。

2. 功能仿真

仿真包括逻辑功能的仿真和定时仿真。该软件只能进行逻辑功能的仿真(该软件称之为方程式仿真),即利用方程式仿真来检验已编译的设计,检查其功能是否正确。方程式仿真的过程如图 D.2.4 所示。

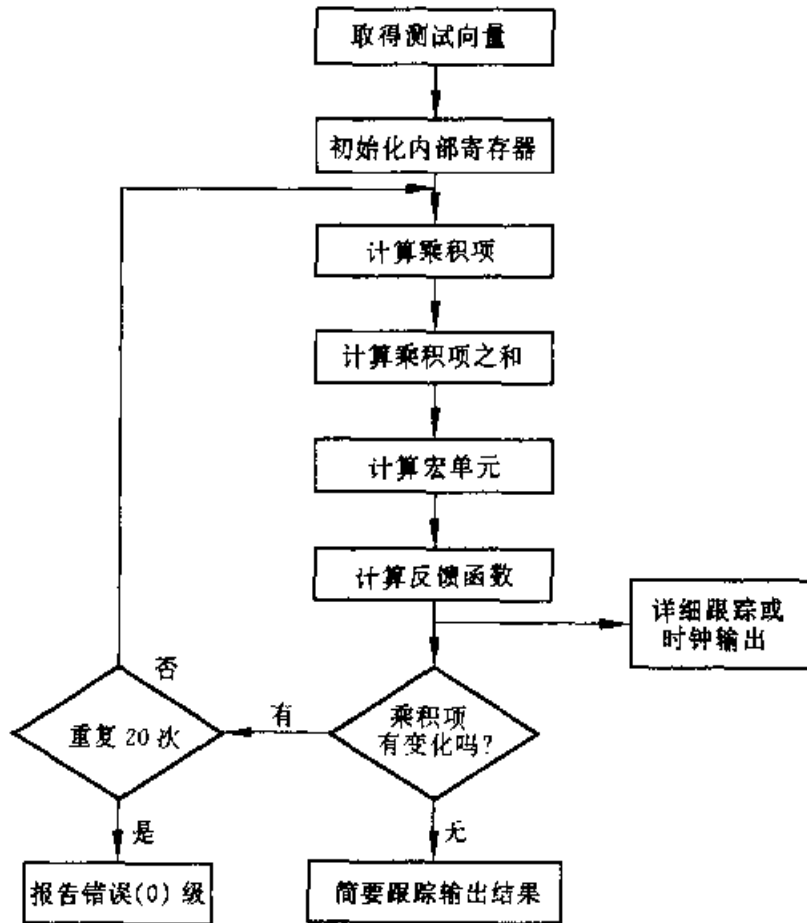










图 D.2.4 仿真的过程

3. 设计处理





输入设计的逻辑后,必须对其进行处理。处理过程包括对单个模块或整个设计所作的语言检查、编译和逻辑优化等步骤。该软件的项目管理器知道每一个源文件应进行哪些处理过程,所以开发人员只需要在项目管理器的左窗 Sources in Project(工程项目源文件)选中将要进行处理的源文件,然后在右窗 Processes for Current Source(处理当前的源文件)中双击相应的处理项目,就能完成各处理过程。如果处理过程通过,处理项目左边会出现绿色“√”标记。如果处理过程失败,处理项目左边将出现红色“×”标记,同时在 Synario Report Viewer(报告观察器)中,将显示出错误所在及错误类型。

该软件中,对于不同的源文件所对应的处理过程如下:




- 对应“器件级源文件(包括所有的源文件)”的处理过程:

-  Update All Schematic Files —— 更新所有的原理图源文件
-  Link Design —— 进行链接设计处理, 将多个源文件合并为一个文件
 -  Linked Equations —— 链接处理后的逻辑方程报告文本文件
-  Fit Design —— 进行适配设计处理
 -  Pre-Fit Equations —— 预适配方程报告文本文件
 -  Signal Cross Reference —— 简化信号名参考
 -  JEDEC File —— 供编程用的数据文件(或称熔丝图文件)
 -  Fitter Report —— 适配报告文本文件







- 对应“测试向量源文件”的处理过程:

-  Compile Test Vectors —— 对测试向量源文件进行编译处理
 -  Compiler Listing —— 编译源文件的列表
-  Simulate Equations —— 逻辑方程式仿真处理
 -  Equation Simulation Report —— 以文本文件的方式显示仿真结果
 - Equation Simulation Waveform —— 以波形图的方式显示仿真的结果

- 对应“顶层电原理图源文件”的处理过程:

- Navigate Hierarchy —— 浏览整个源文件的层次化设计管理器
-  Compile Schematic —— 编译原理图文件
-  Reduce Schematic Logic —— 简化原理图逻辑
 -  Reduced Equations —— 简化的逻辑方程式文本文件

- 对应“ABEL 源文件”的处理过程:

-  Compile Logic —— 编译逻辑方程式
-  Check Syntax —— 语法检查
 -  Compiler Listing —— 编译列表
 -  Compiled Equation —— 编译后的逻辑方程式文本文件
-  Reduce Logic —— 进行简化逻辑方程式的处理
 -  Reduced Equations —— 简化后的逻辑方程式文本文件

其中 Link Design(设计链接处理)将各底层源文件链接成一个较大的设计结构,Fit Design(设计适配)是设计处理的最后一个过程,它将检验所有当前源文件的更新、设计链接和所有的逻辑简化。如果适配成功,就会产生一个 JEDEC 文件。该文件包含了将设计装入到 PLD 器件所需的信息。

4. 器件编程

将设计阶段生成的 JEDEC 文件装入到器件中去的过程称为编程(也称为下载、配置)。对普通的非在系统编程器件(如 GAL16V8 等),需要用专用的编程器对 PLD 器件进行编程。对在系统可编程器件(如 Lattice 公司的 ispLSI1016 等),可使用 ISP 菊花链下载软件(ispDCD),通过计算机的并行口和专用下载电缆直接对器件进行编程。

实际编程时,开发软件在收到设计者发出的编程命令(Download)后,首先检查电缆是否接上,计算机与器件之间的通讯是否正常,然后执行编程操作,并对编程结果进行检验。检验通过后,器件便可在系统中正常使用了。

附录 E 电气图用图形符号二进制逻辑单元(GB4728.12-85)简介

该标准是由国家标准局颁布的用于绘制二进制逻辑^①单元电路的符号标准。

E.1 二进制逻辑单元图形符号的组成

二进制逻辑单元图形符号由方框(或方框的组合)和一个或多个定性符号组成,如图 E.1.1 所示。框图中单星号 * 表示与输入和输出有关的定性符号,双星号 ** 为总定性符号。定性符号在框内,框外为信号。按一般约定,输入线画在方框的左边或上边;输出线画在方框的右边或下边,以保持信息流的方向是从左到右,或从顶到底。

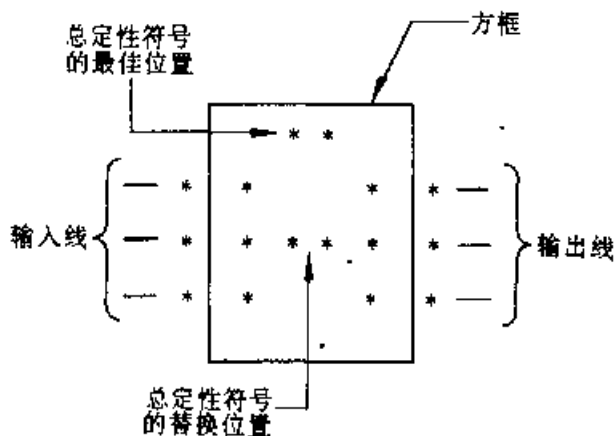


图 E.1.1 符号的组成

E.1.1 方框

如图 E.1.2 所示,方框分为单元框,公共控制框和公共输出单元框。单元框是基本方框,它可与公共控制框或公共输出单元框,或与前二者一起组成逻辑单元框外形轮廓。

方框可用邻接法和相嵌法的方式组合。在有邻接单元或相嵌单元的符号中,

^① 二进制逻辑系指二值数字逻辑。

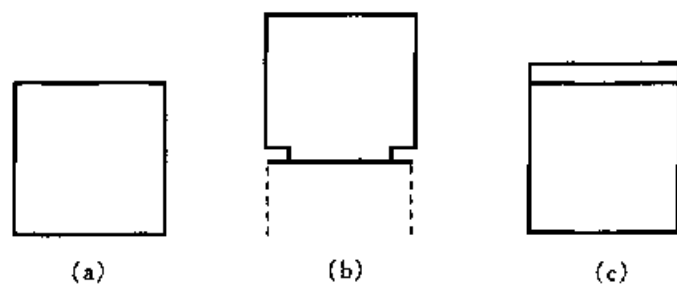


图 E.1.2 三种方框

(a) 单元框 (b) 公共控制框 (c) 公共输出单元框

如果单元框之间的公共线是沿着信息流方向,就表明这些单元之间无逻辑连接;如果单元框之间的公共线是垂直于信息流方向,则表明单元之间至少有一种逻辑连接,如图 E.1.3a、b 所示。

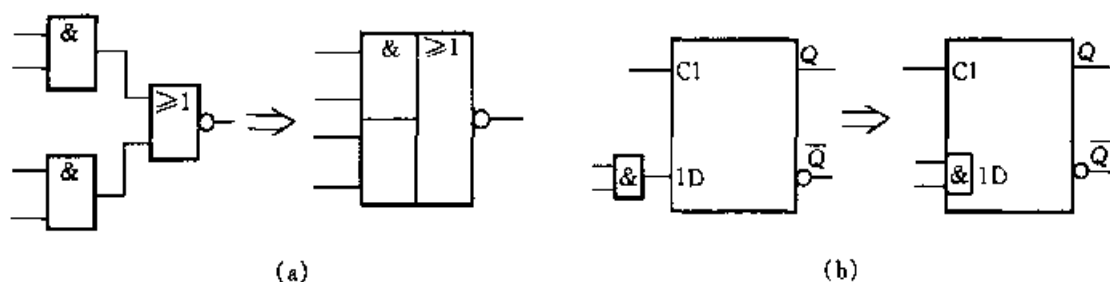


图 E.1.3 邻接法与相嵌法

(a) 邻接法 (b) 相嵌法

E.1.2 定性符号

定性符号分为总定性符号和与输入、输出和其他连接有关的定性符号两种。定性符号画在单元基本框中用以表明该单元输入或输出的物理特性或逻辑特性,或表明该单元全部逻辑特性。

1. 总定性符号

总定性符号用来规定单元的输出与输入之间的逻辑功能,其作用对象是方框内部,即输出、输入内部逻辑状态。常用总定性符号如表 E.1.1 所示。

2. 与输入、输出和其他连接有关的定性符号

与输入、输出和其他连接有关的定性符号用来说明相应输入端或输出端所具有的逻辑功能或物理特性。共有以下 4 种(表 E.1.2~E.1.5):

表 E.1.1 常用总定性符号

符 号	说 明	符 号	说 明
&	与	COMP	数值比较
≥ 1	或	ALU	算术逻辑
$\geq m$	逻辑门框		二进制延迟
$= 1$	异或	$I=0$	初始 0 状态
$= m$	等于 m	$I=1$	初始 1 状态
1	缓冲		单稳,可重复触发
=	恒等		单稳,不可重复触发
$> n/2$	多数		非稳态
$2k$	偶数(偶数校验)		非稳态,同步启动
$2k+1$	奇数(奇数校验)		非稳态,完成最后一个脉冲后停止输出
\triangleright	放大、驱动		
* \diamond	分布连接、点功能、线功能		非稳态,同步启动,完成最后一个脉冲后停止输出
* \square	具有磁滞特性	SRG m	m 位的移位寄存
X/Y	转换	CTR m	循环长度为 2^m 的计数
MUX	多路选择	CTRDIV m	循环长度为 m 的计数
DX 或 DMUX	多路分配	ROM m	只读存储
Σ	加法运算	PROM m	可编程只读存储
P-Q	减法运算	RAM Δ	随机存储
CPG	先行超前进位	CAM Δ	内容可寻址寄存
Π	乘法运算		

注: * 号用表明单元逻辑功能的总定性符号代替; Δ 号用地址和位数的适当符号来代替。

表 E.1.2 逻辑非、逻辑极性和动态输入符号

符 号	说 明	符 号	说 明
	逻辑非,在输入端		逻辑极性 } 在信息流为从右 极性指示符 } 到左的输出端
	逻辑非,在输出端		动态输入
	逻辑极性 } 在输入端 极性指示符 }		带逻辑非的动态输入
	逻辑极性 } 在输出端 极性指示符 }		带极性指示符的动态输入
	逻辑极性 } 在信息流为从右 极性指示符 } 到左的输入端		

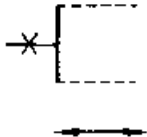
表 E.1.3 内部连接符号

符 号	说 明	符 号	说 明
	内部连接		具有逻辑非和动态特性的内部连接
	具有逻辑非的内部连接		内部输入(虚拟输入)
	具有动态特性的内部连接		内部输出(虚拟输出)

表 E.1.4 方框内符号

符 号	说 明	符 号	说 明
	延迟输出		三态输出
	双向门控输入具有磁滞现象的输入		扩展输入
	开路输出(例如开集电极、开发射极、开漏极、开源极)		扩展输出
	H型开路输出(例如PNP开集电极、NPN开发射极、P沟道开漏极、N沟道开源极)		使能输入
	L型开路输出(例如PNP开发射极、NPN开集电极、P沟道开源极、N沟道开漏极)		D输入
	无源下拉输出		J输入
	无源上拉输出		K输入
	R输入		操作数输入
	S输入		数值比较器的“大于”输入
	T输入		数值比较器的“小于”输入
	移位输入,从左到右或从顶到底		数值比较器的“等于”输入
	移位输入,从右到左或从底到顶		数值比较器的“大于”输出
	正计数输入		数值比较器的“小于”输出
	逆计数输入		数值比较器的“等于”输出
	联想存储器的询问输入		内容输入
	联想存储器的比较输出		内容输出
	联想存储器的匹配输出		在输入边的线组合
	多位输入的位组合		在输出边的线组合
	多位输出的位组合		固定方式输入
			固定状态输出

表 E.1.5 非逻辑连接和信息流指示符

符 号	说 明
	非逻辑连接,在左边示出双向信息流

E.2 关联标注法

运用关联标注法的目的,是为了使二进制逻辑单元的图形符号更紧凑和更确切地表达逻辑单元的内部连接关系,运用这种标注不需具体画出所有单元及所包括的内部连接,就能表明输入之间、输出之间和输入和输出之间的关系,由关联标注所提供的信息补充了单元功能定性符号所提供的信息。

E.2.1 约定

关联标注是人们引入符号语言中的一种概念,需要一些共同的约定。

为便于叙述,引用**影响**和**受影响**两个术语。用一个表达某输入(输出)与其他输入(输出)之间内在关系的特定字母后,跟着标识序号来标记影响其他输入或输出的输入(输出)[称之为“影响输入(输出)”];用与“影响输入(输出)”相同的标识序号来标记“受影响输入(输出)”影响的输入或输出[称之为“受影响输入(输出)”]”。

- 如果以“影响输入(输出)”内部逻辑状态的补状态作为影响条件,则在“受影响输入(输出)”的标识序号上划一横线。

- 如果两个影响输入(输出)有相同的字母和相同的标识序号,则它们之间彼此处在相或关系中。

- 如果需要用一个标记来说明受影响输入(输出)对单元的影响,则应在该标记前面加上“影响输入(输出)”的标识序号作为前缀。

- 如果一个输入(输出)受一个以上“影响输入(输出)”的影响,则“影响输入(输出)”的各个标识序号均应在“受影响输入(输出)”的标记中列出,并以逗号隔开。这些标识序号从左到右的排列次序与影响关系的顺序相同。

E.2.2 关联类型及用途

关联类型共有 10 种,见表 E.2.1。

“与”关联、“或”关联和“非”关联是用来注明输入 和/或 输出之间的布尔关系。

“互连”关联是用来表示一个输入或输出把其逻辑状态强加到另一个或多个输入 和/或 输出上。

“控制”关联是用来标识时序单元的定时输入或时钟输入,并指出受它控制的输入。

“置位”关联和“复位”关联是用来规定当 R 输入和 S 输入均处在它们的内部 1 状态时,基本触发器的内部逻辑状态。

“使能”关联是用来标识使能输入并指出由它控制的输入和/或输出(例如,哪些输出呈现高阻抗状态)。

“方式”关联是用来标识选择单元操作方式的输入,并指出取决于该方式的输入和/或输出。

“地址”关联是用来标识存储器的地址输入。

表 E.2.1 关联类型、符号及作用

关联类型	关联符号	对“受影响输入”或“受影响输出”的影响	
		当“影响输入(出)”=1时	当“影响输入(出)”=0时
控制	C	允许动作	禁止动作
使能	EN	允许动作	禁止“受影响输入”动作、置开路或三态输出于外部高阻抗条件,置其他输出于0状态
方式	M	允许动作(已选方式)	禁止动作(未选方式)
复位	R	“受影响输出”复位	不起作用
置位	S	“受影响输出”置位	不起作用
与	G	允许动作	置0状态
或	V	置1状态	允许动作
非	N	求补状态	不起作用
互连	Z	置1状态	置0状态
地址	A	允许动作(已选地址)	禁止动作(未选地址)

E.3 逻辑状态、逻辑电平与逻辑约定

E.3.1 内、外部逻辑状态与逻辑电平

在本标准中,引入了内、外部逻辑状态的概念,它们分别表示图形符号框内、外输入或输出的逻辑状态。

对输入端而言,指的是在任何限定符号之前的逻辑状态。对输出端而言,指的是在任何限定符号之后的逻辑状态。所有限定符号(除非门外)均表示对内部逻辑状态而言的逻辑功能。

图 E.3.1 为逻辑状态和逻辑电平的概念图解。

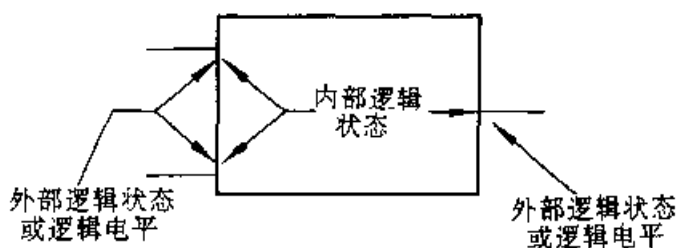


图 E.3.1 逻辑状态和逻辑电平的概念图解

E.3.2 逻辑约定

对逻辑状态与逻辑电平之间关系所作的规定,称之为逻辑约定。逻辑约定有以下两种:

1. 单一逻辑约定

即正逻辑约定或负逻辑约定。这种逻辑约定采用逻辑非符号的图形符号。

在本图标中对正、负逻辑约定的图形符号在画法上无区别。只有用文字或图形符号说明。

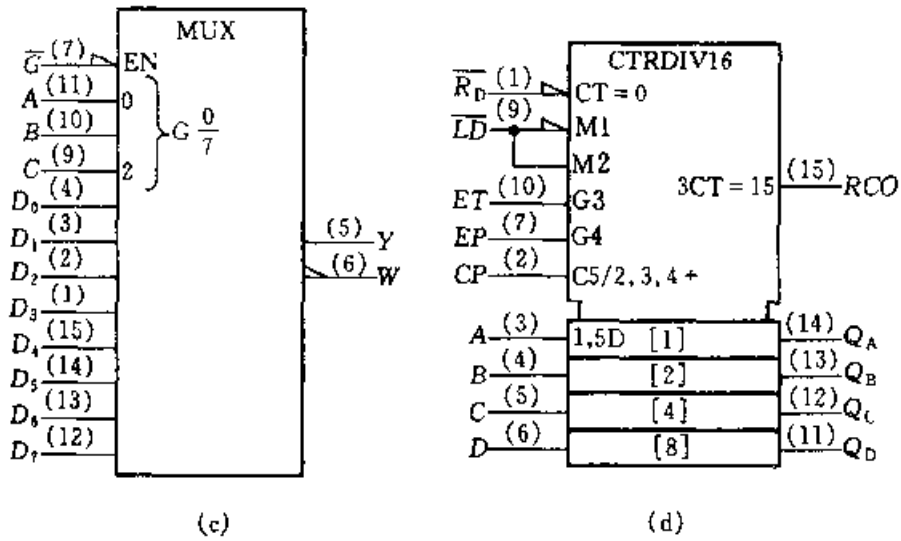
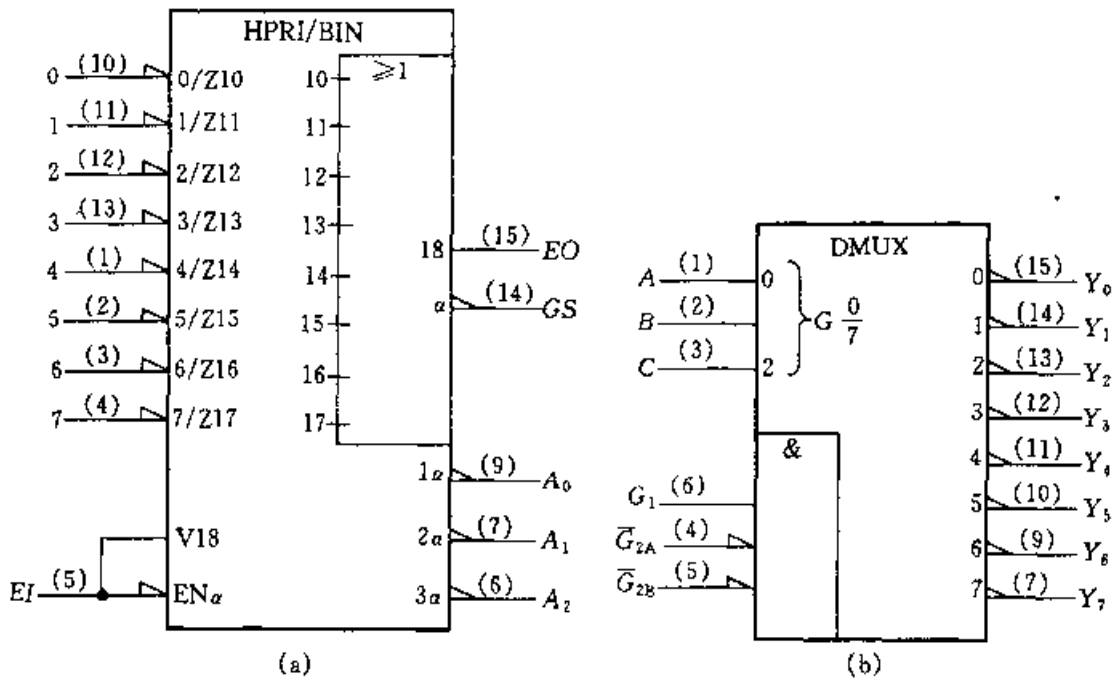
2. 用极性指示符号的逻辑约定

这是一种用极性指示符号(见表 E.1.2)来表示输入(出)端的外部逻辑电平与内部逻辑状态之间关系的逻辑约定。有极性指示符的输入(出)端,其框外低电平对应框内逻辑1状态,框外高电平对应框内逻辑0状态。无极性指示符的输入(出)端,其框外低电平对应框内逻辑0状态,框外高电平对应框内逻辑1状态。

值得强调的是,极性指示符仅表示一种逻辑约定,它与单一逻辑约定中的逻辑非的“ \circ ”运算符性质完全不同。在用极性指示符表示的逻辑电路中,其输入和输出端上不允许采用逻辑非符号,同样,在单一逻辑约定的逻辑电路中,也不允许采用极性指示符。还应当注意,无论采用什么逻辑约定,在符号框内只存在内部逻辑状态,不存在内部逻辑电平。在采用单一逻辑约定的图形符号中,图形框外既存在外部逻辑电平又存在外部逻辑状态;而在采用极性指示符的逻辑约定的图形符号中,图形框外仅存在外部逻辑电平,不存在外部逻辑状态,所以不能将逻辑函数表达式写在采用极性指示符逻辑约定图形符号的输出端。

E.4 典型逻辑器件图形符号示例

根据国标 GB4728.12-85 的规定,选择几种典型逻辑器件的图形符号如图 E.4.1 所示。



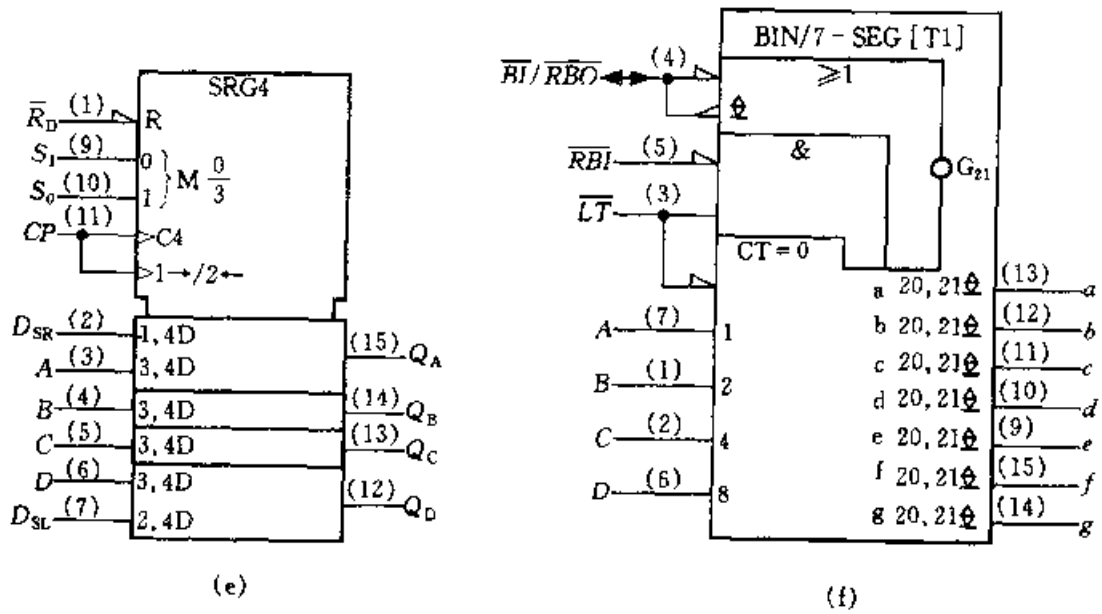
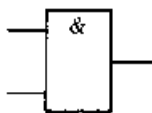
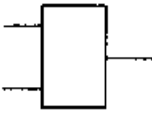

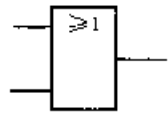
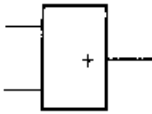

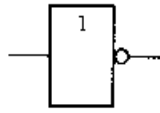
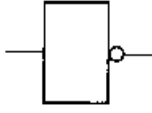
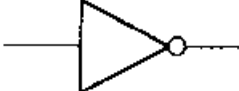
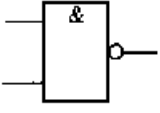
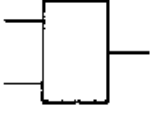

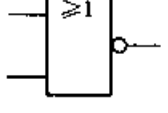
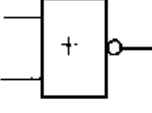

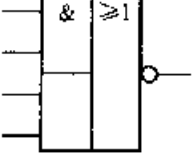
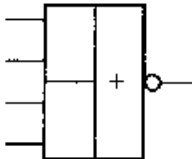
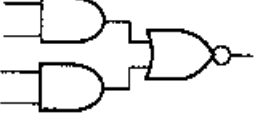
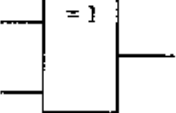
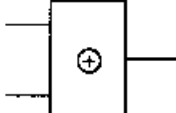

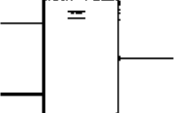
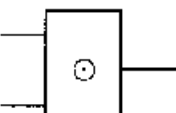

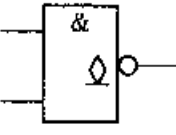
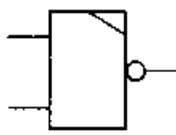

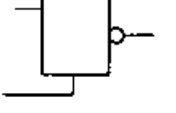


图 E.4.1 几种典型的图形符号

(a) 8 线 - 3 线优先编码器'LS148、'HC148;(b) 3 线 - 8 线优先编码器/多路分配器'LS138、'HC(T)138;(c) 8 选 1 数据选择器/多路转换器'LS151、'HC(T)151;(d) 4 位二进制同步计数器(异步清除)'LS161、'HC(T)161;(e) 4 位双向通用移位寄存器(并行存取)'LS194、'HC(T)194;(f) 4 线 - 七段译码器/驱动器(BCD 输入, 有上拉电阻)'LS48。器件型号前的单撇号“'”表示 54/74 系列号等。

附录 F 常用逻辑符号对照表

符 号 名 称	说 明	本书所用符号	曾用符号	国外所用符号
与 门				
或 门				
非 门				
与非门				
或非门				
与或非门				
异或门				
同或门				
集电极开路 与非门				
三态输出 与非门				

续表

符号名称	说明	本书所用符号	曾用符号	国外所用符号
传输门				
半加器				
全加器				
基本 RS 触发器				
同步 RS 触发器				
上升沿触发 D 触发器				
下降沿触发 JK 触发器				
脉冲触发(主从) JK 触发器				
带施密特触发特性的与门				

* 本书所用符号为国标符号,传输门无国标。

附录 G 国产半导体集成电路 型号命名法 (GB3430 - 82)

本标准适用于半导体集成电路系列和品种的国家标准所生产的半导体集成电路(以下简称器件)。

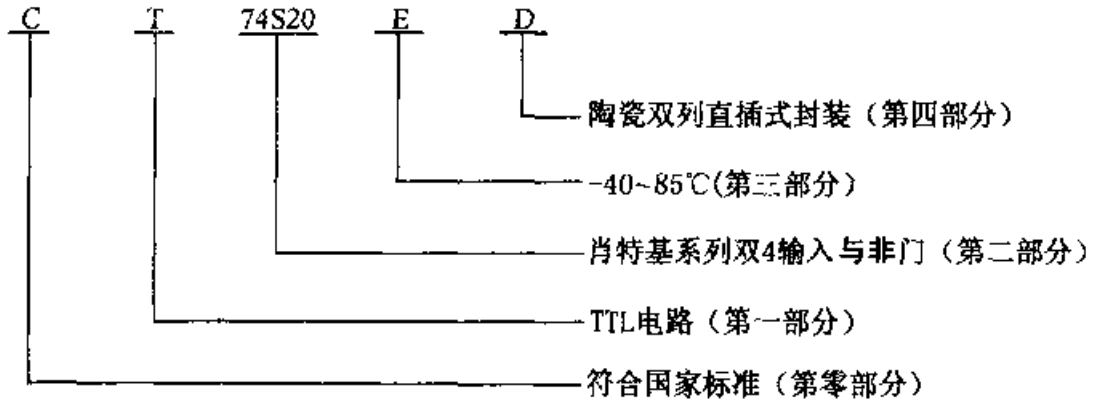
G.1 型号的组成

器件的型号由五部分组成,其五个组成部分的符号及意义如下:

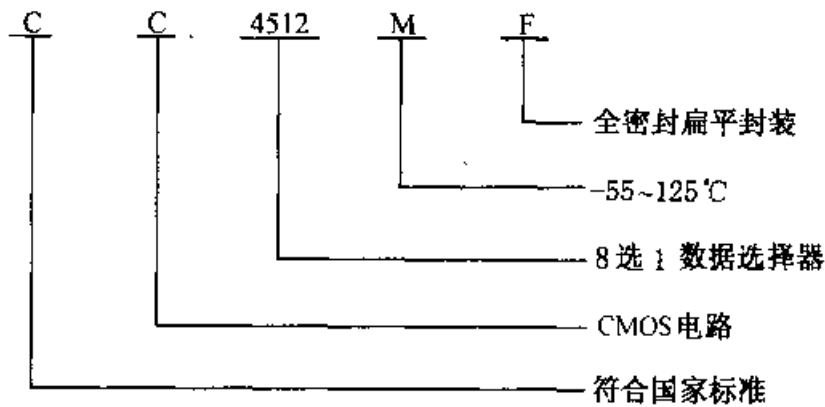
第零部分		第一部分		第二部分	第三部分		第四部分	
用字母表示 器件符合国 家标准		用字母表示器件的类 型		用阿拉伯数字 和字母表示器 件的系列和品 种代号	用字母表示器件 的工作温度范围		用字母表示器件的 封装形式	
符号	意义	符号	意义		符号	意义	符号	意义
C	中国制造	T	TTL		C	0~70℃	W	陶瓷扁平
		H	HTL		E	-40~85℃	B	塑料扁平
		E	ECL		R	-55~85℃	F	全密封扁平
		C	CMOS		M	-55~125℃	D	陶瓷直插
		F	线性放大器		∴	∴	P	塑料直插
		D	音响、电视电路				J	黑陶瓷扁平
		W	稳压器				K	金属菱形
		J	接口电路				T	金属圆形
		B	非线性电路				∴	∴
		M	存储器					
μ	微型机电路							
∴	∴							

G.2 例示

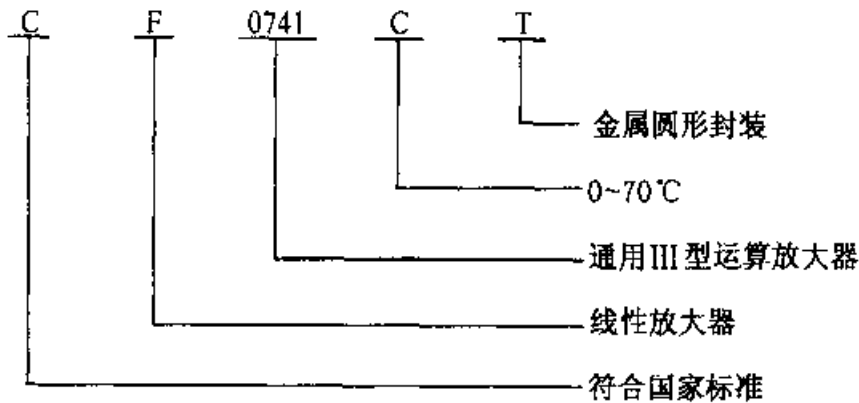
例 1: 肖特基 TTL 双 4 输入与非门



例 2: CMOS 8 选 1 数据选择器 (三态输出)



例 3: 通用运算放大器



参 考 文 献

- 1 Susan A. R. Garrod, Robert J. Borns. Digital logic; Analysis, Application & Design, Purdue University. Saunders College Publishing. Philadelphia, 1991
- 2 David A. Hodges and Horace G. Jackson. Analysis and Design of Digital Integrated Circuits. New York; McGraw-Hill, 1988
- 3 Mark N. Horenstein. Microelectronic Circuits and Devices. Prentice Hall Inc. , 1996
- 4 Adel S. Sedra & Kenneth C. Smith. Microelectronic Circuits. CBC College Oxford University Press, Inc. , 1998
- 5 William Kleitz. Digital Electronics, A Practical Approach. Prentice Hall Inc. , 1996
- 6 M. I. Elmasry. BiCMOS Integrated Circuit Design with Analog, Digital, and Smart Power Applications. IEEE Press, 1994
- 7 Jacob Millman & Arvin Grabel. Microelectronics 2nd ed. McGraw-Hill Book Company, New York, 1987
- 8 Thomas C. Batec. Computer Architecture and Logic Design. McGraw-Hill Book Company, New York, 1987
- 9 Randy H. Katz. Contemporary Logic design. The Benjamin/Cummings Publishing Company, Inc. , Redwood City, California, 1994
- 10 Milos D. Ercegovac, Tomas _Lang, Jaime H. Moreno. Introduction to Digital System. John Wiley Son, 1998
- 11 清华大学电子学教研组编, 阎石主编. 数字电子技术基础. 第四版. 北京: 高等教育出版社, 1998
- 12 西安交通大学电子学教研室编, 沈尚贤主编. 电子技术导论, 上册. 北京: 高等教育出版社, 1985
- 13 李士雄, 丁康源主编. 数字集成电子技术教程. 北京: 高等教育出版社, 1993
- 14 曹汉房, 陈耀奎编著. 数字技术教程. 北京: 电子工业出版社, 1995
- 15 杨晖, 张凤言编著. 大规模可编程逻辑器件与数字系统设计. 北京: 北京航

- 空航天大学出版社,1998
- 16 电子工程手册编委会等编. 中外集成电路简明速查手册——TTL、CMOS.
北京:电子工业出版社,1991

部分习题答案

第 1 章

- 1.3.1 $(43)_D = (101011)_B = (53)_O = (2B)_H$
 $(127)_D = (1111111)_B = (177)_O = (7F)_H$
 $(254.25)_D = (11111110.01)_B = (376.2)_O = (FE.4)_H$
 $(2.718)_D = (10.10110111)_B = (2.56)_O = (2.B7)_H$
- 1.3.2 (1) $(151)_D, (97)_D$; (2) $(2195)_D, (893)_D$; (3) $(329)_D, (149)_D$
- 1.3.3 (1) $(29)_H$; (2) $(3.68)_H$
- 1.3.4 (1) $(1F4)_H$; (2) $(3B)_H$; (3) $(.570A)_H$; (4) $(3EA.7333)_H$
- 1.3.5 (1) $(001000111111.01000101)_B$; (2) $(1010000001000000.01010001)_B$
- 1.3.6 (1) $(259.125)_D$; (2) $(42077.0458)_D$

第 2 章

- 2.2.1 (1) $I_C = 1.08 \text{ mA}, V_{CE} = 0.38 \text{ V}$;
(2) $V'_{CE} = 1.7 \text{ V}$; (3) $V'_{CE} \approx 0.3 \text{ V}$
- 2.4.3 (1) $N_O = 20$; (2) $N_O = 5$
- 2.4.4 (1) 驱动门不超载; (2) 可增加 8 个 LS00 的负载
- 2.4.5 $L = \overline{AB} \cdot \overline{BC} \cdot \overline{D} \cdot E$
- 2.4.6 $R_{p(\min)} \approx 0.29 \text{ k}\Omega, R_{p(\max)} = 130 \text{ k}\Omega$
- 2.6.3 $L = \overline{AB} + A\overline{B} = A \oplus B$
- 2.7.1 $L = \overline{(BC + DE)A} + \overline{(A + G)EF}$
- 2.7.2 $L = \overline{A} \overline{B} + AB = A \odot B$
- 2.9.3 $R_{p(\min)} = 287 \Omega, R_{p(\max)} = 75 \text{ k}\Omega$, 可取 $R_p = 1 \sim 3 \text{ k}\Omega$
- 2.9.10 74LS04, $R \approx 470 \Omega$
- 2.9.11 $LA = A + \overline{B}_1 \overline{B}_2, LB = \overline{LA} = \overline{A} (B_1 + B_2)$
- 2.9.12 $LA = \overline{LB}, LB = B_1 \cdot B_2 \cdot \overline{A}_1 \cdot \overline{A}_2$

CAD 习题

C2.10.1

各节点电压

节点	电压/V
V _i	0.2000
V _{b1}	0.9765
V _{b2}	0.2620
V _{b3}	45.46E-09
V _{b4}	4.9985
V _{e4}	4.2855
V _o	3.7927

各支路电流

	T1	T2	T3	T4
I _B /A	1.01E-03	-4.73E-11	-3.79E-11	9.39E-07
I _C /A	4.73E-11	5.49E-11	4.17E-11	9.39E-05

C2.10.2

各节点电压

节点	电压/V
V _i	3.7000
V _{b1}	2.3099
V _{b2}	1.6055
V _{b3}	0.8001
V _{b4}	0.8998
V _{e4}	0.3139
V _o	0.0620

各支路电流

	T1	T2	T3	T4
I _B	6.73E-04	7.40E-04	2.50E-03	6.86E-09
I _C	7.40E-04	2.56E-03	-8.53E-07	6.90E-07

C2.10.3

传输延迟时间 t_{PHL} 和 t_{PLH} 约为 3.9 ns。

3.3.2

$$L = (A \oplus B) \oplus (C \oplus D) \quad \text{奇偶校验器}$$

3.3.3

$$L_1 = \overline{A+B}, L_2 = \overline{\overline{A+B} + \overline{A+B}}, L_3 = \overline{A+B}$$

3.3.4

$$S = A \oplus B \oplus C, C = \overline{\overline{(A \oplus B)C} \cdot \overline{AB}}$$

奇偶校验电路, $S=1$ 奇数个 1; $S=0$ 偶数个 1; $C=1$, 1 的个数 ≥ 2 ; $C=0$, 1 的个数 ≤ 1 。

3.3.5 奇偶校验电路 $L=0$, 偶数个 1

$$L=1, \text{奇数个 1}$$

3.3.6 多位数全加器

$$S_0 = A_0 \oplus B_0, C_0 = A_0 B_0$$

$$S_1 = A_1 \oplus B_1 \oplus A_0 B_0, C_1 = A_1 B_1 + (A_1 \oplus B_1) A_0 B_0$$

第 4 章

4.3.2

$$\begin{aligned} L &= S_0 S_1 I_0 + S_0 \bar{S}_1 I_1 + \bar{S}_0 S_1 I_2 + S_0 S_1 I_3 \\ &= S_0 S_1 + S_0 \bar{S}_1 + S_0 S_1 = S_1 + \bar{S}_0 S_1 \end{aligned}$$

4.3.3 取 $I_0=0, I_1=I_2=I_3=1$

$$\begin{aligned} F &= S_0 \bar{S}_1 + S_0 S_1 + S_0 S_1 \\ &= S_0 S_1 + \bar{S}_0 S_1 + S_0 S_1 + S_0 S_1 = S_0 + S_1 \end{aligned}$$

第 5 章

5.1.1 功能表

R	S	Q^n	Q^{n+1}
1	0	0	0
1	0	1	0
0	1	0	1
0	1	1	1
1	1	0	1
1	1	1	1
0	0	0	1
0	0	1	1

5.1.4 同步 RS 触发器有时钟信号 CP 控制其翻转。基本 RS 触发器设有 CP 信号。

5.1.6

SR	Q^n	Q^{n+1}	说明
00	0	0	不变
00	1	1	
01	0	0	置 0
01	1	0	
10	0	1	置 1
10	1	1	
11	0	—	不定
11	1	—	

5.1.8 主从结构触发器

SR	Q^n	Q^{n+1}	说明
00	0	0	不变
	1	1	
01	0	0	置 1
	1	1	
10	0	0	置 0
	1	0	
11	0	—	不定
	1	—	

5.1.9

CP	J	K	Q^n	Q^{n+1}	说明
↓	0	0	0	0	保持
			1	1	
↓	1	0	0	1	置 1
			1	1	
↓	0	1	0	0	置 0
			1	0	
↓	1	1	0	1	翻转
			1	0	

5.2.7 相位差： ϕ_1 超前 ϕ_2 一个 CP 时钟周期。

5.3.1 对边沿触发器，CP 信号跳变前瞬间输入信号，CP 跳变后触发翻转；对主从触发器，输入信号在 CP 正跳沿前加入，而 CP 负跳沿后才触发翻转。

第 7 章

7.1.1 $M=6$

7.1.5 $M=5$

7.1.6 $M=7$

7.1.9 $M=10$

7.1.10 $M=11$

7.1.11 $M=11$

7.1.12 $M=11$

7.1.15 $M=4096$ ，采用并行进位

7.1.16 $M=174$

7.1.17 $M=174$

7.2.1 四相时序脉冲产生电路

第 8 章

8.1.2 25.6 ms。

8.1.3 (1) 64K, 16 根, 1 根; (2) 1M, 18 根, 4 根;

(3) 1M, 20 根, 1 根; (4) 1M, 17 根, 8 根

8.1.4 (1) 7FFH; (2) 3FFFH; (3) 3FFFFH

8.1.6 10 根

8.2.2 (1) $2^6 \times 6$ 位; (2) $2^8 \times 10$ 位

8.3.1 $L = \bar{A} \bar{B} C \bar{D} + \bar{A} B \bar{C} \bar{D} + \bar{B} C D + \bar{A} \bar{B} C D + \bar{A} \bar{B} \bar{C} \bar{D}$

8.3.2 当 $OE = 1$ 时

$$X = \overline{A\bar{B} + A\bar{C} + \bar{B}C}$$

$$Y = \overline{\bar{D}EF + \bar{D}\bar{E}F + \bar{D}E\bar{F} + \bar{D}\bar{E}\bar{F}}$$

$$Z = \overline{G\bar{H} + GHJ}$$

当 $OE = 0$ 时, 输出为高阻。

8.3.4 2 位二进制递增计算器

第 9 章

$$9.1.1 (2) T = RC \ln \frac{(V_{DD} + V_{th})(2V_{DD} - V_{th})}{V_{th}(V_{DD} - V_{th})}$$

$$9.2.1 t_w = 0.8 \mu s$$

$$9.2.2 t_w \approx 0.7RC$$

$$9.2.3 t_w = (3.57 \sim 17.57) ms$$

9.3.1 ΔV_T 变动范围为 $0.7 \sim 1.4 V$

$$9.3.2 T = 1.53 ms, t_w = 0.21 ms$$

$$9.3.3 t = \frac{1}{T}, T = R_1 C_1 \ln \frac{V_{DD} - V_{T-}}{V_{DD} - V_{T+}} + R_2 C_2 \ln \frac{V_{T+}}{V_{T-}}$$

$$9.4.1 t \approx \frac{1.43}{(R_1 + 2R_{DS})C}$$

$$9.4.2 t = \left[\frac{2}{3} R_e (R_1 + R_2) C \right] / R_2$$

$$9.4.5 R_1 = 910 M\Omega, R_2 = 0.6 k\Omega$$

$$9.4.6 t_w \approx 1.1 R_3 C_3$$

第 10 章

$$10.1.1 (1) 0 \sim -V_{REF}$$

$$(2) V_{REF} = -10 V$$

$$10.1.2 R_f = 16 k\Omega$$

$$10.1.3 (1) v_O = -\frac{V_{REF} R_f}{R} \sum_{i=0}^{n-1} S_i \cdot 2^i$$

$$(2) -17.8 V$$

$$10.1.5 3.13\%$$

$$10.1.6 (1) v_O = -R_f \left(\frac{V_{REF}}{2^{10} R} \sum_{i=0}^9 D_i 2^i + \frac{V_B}{R_B} \right)$$

$$(2) \frac{|V_B|}{R_B} = \frac{|V_{REF}|}{2R}$$

$$10.2.1 \Delta = V \frac{14}{15}$$

$$D_2 D_1 D_0 = 011$$

$$\epsilon_{\max} = \frac{7}{15} V$$

$$10.2.2 t = 10 \mu\text{s}$$

$$f < 0.1 \text{ MHz}$$

$$10.2.3 1101$$

$$10.2.5 (1) 100 \text{ ms}$$

$$(2) |V_{O\max}| = 5 V$$

$$(3) V_T = -8.3 V$$

$$10.2.6 f_{\max} = f_{CP}/2^{n+1}$$

索引(汉英对照)

二 画

二极管逻辑门(Diode logic gate)	36
二极管 - BJT 逻辑(Diode - transistor logic)	84
二值数字逻辑(Binary digital logic)	3
二进制(Binary)	12
~数(number)	12
~地址(address)	
~权(weights)	12
二进制编码的十进制(Binary - Coded - Decimals, BCD)	19, 104
二 - 十进制转换(Binary to decimal conversion)	12
十进制数(Decimal number)	11
十 - 二进制转换(Decimal to binary conversion)	12, 15
十六进制数(Hexadecimal number)	17
八进制数(Octal number)	18
七段显示器(Seven - Segment display)	138
二的补码(Two' complement)	462, 163

三 画

三极管(Bipolar Junction Transistor, BJT)	28
三极管 - 三极管逻辑[Transistor - Transistor Logic(TTL)]	9, 40 ~ 58
门(Gate)	
~阵列(array)	298

与(AND)~	22, 36, 37
或(OR)~	22, 37 ~ 39
非(NOT)~	39, 40
与或非(AND - OR - INVERT)~	94
异或(Exclusive OR)~	24
异或非(Exclusive NOR)~	65
与非(NAND)~	24, 94
或非(NOR)~	24, 94
三态(three state)~	53 ~ 55, 85
集电极开路(open collector)~	51 ~ 53
上拉电阻(Pull - up resistor)	52
上升沿(Rise edge)	5, 6
上升时间(Rise time)	5, 6
下降沿(Fall edge)	5, 6
下降时间(Fall time)	5, 6

四 画

专用集成电路(Application Specific Integrated Circuit ASIC)	
计数器(Counter)	218, 240 ~ 259
~二进制(binary)~	133
~十进制(decimal)~	
~递增 - 递减(可逆)(up - down)~	245, 246
~异步(纹波)[asynchronous(ripple)]~	214, 221, 222
~同步(synchronous)~	184, 214, 217 ~ 219, 223, 224
无关项(Don't care terms)	
开启电压(Threshold voltage)	

开关特性(Switching characteristics) 32
 开关时间(Switching time) 34,35
 反相器(Inverter) 61
 反演规则(Complementary operation theorem) 91
 反码(One's complement) 163
 反向恢复时间(Reverse recovery time) 30
 双稳态(Bistable)
 双向移位寄存器(Bidirectional shift register) 263
 双列直插式封装(Dual in-line package, DIP) 50
 分辨率(Resolution) 396,414
 分频(Frequency division) 240~242
 互补 MOS 门(Complementary MOS gate, CMOS) 9,61
 双积分(斜)模数转换器(Dual-slope analog to digital converter) 410
 卡诺图(Karnaugh map) 98
 比特率(Bit rate) 6

五 画

布尔代数(Boolean algebra) 21,89
 加法器(Adder) 155
 半(half)~ 156
 全(full)~ 156
 串行进位(Serial carry)~ 158
 正逻辑(Positive logic) 72
 半导体存储器(Semiconductor memory) 275
 只读存储器(Read only memory, ROM) 275,290
 可编程序(Programmable~, PROM) 290
 可擦可编(Erasable Programmable~, EPROM) 290
 电可擦除的可编程只读存储器(Elec-

trically Erasable Programmable~, E²PROM) 290
 可编程逻辑器件(Programmable Logic Device, PLD) 9,295,449~452
 可编程阵列逻辑(Programmable Array Logic, PAL) 298
 可逆计数器(Reversible counter) 245,246
 主-从触发器(Master-Slave flip-flop) 186
 电平触发(Level triggered) 180
 占空比(Pulse duration ration) 5
 边沿触发(Edge triggered) 188
 延时-功耗积(Time delay-power dissipation product) 50

六 画

异或门(见门)
 异步时序逻辑(Asynchronous sequential logic) 214
 异步二进制计数器(Asynchronous binary counter) 240~242
 异步十进制计数器(Asynchronous decimal counter)
 同或门(Exclusive Nor gate) 65,66
 同步时序逻辑(Synchronous sequential logic) 214
 同步触发器(Synchronous flip-flop) 184
 同步二进制计数器(Synchronous binary counter) 242~246
 同步十进制计数器(Synchronous decimal counter) 246
 多谐振荡器(Astable multivibrator) 347~351,363,370,375
 多发射极三极管(Multiemitter transistor) 45
 优先编码器(Priority encoder) 128
 存储器(Memory) 275

只读(read-Only~, ROM) 275
 读/写(read-Write~, RWM) 277, 280
 随机存取(Random Access~, RAM) 275
 静态(static)~ 275
 动态(dynamic)~ 275
 快闪存储器(Flash~) 290, 293
 存储矩阵(Memory array) 278, 279
 存储单元(memory cell) 276
 动态(dynamic)~ 276
 静态(static)~ 276
 存储时间(Storage time) 281
 传输延迟时间(Propagation delay time) 49, 50
 传输特性(Transfer characteristics) 43~45
 传输门(Transmission Gate, TG) 68, 69
 负逻辑(Negative logic) 72
 再生(刷新)(Regenerate, Refresh) 276
 字(Word) 279
 权(Weight) 12, 19
 回差电压(Backlash voltage) 367
 行选择线(Row-Select line) 279
 列选择线(Column-Select line) 279

七 画

纹波计数器(Ripple counter)(见异步二进制计数器)
 肖特基二极管(Schottky diode) 55, 56
 肖特基三极管(Shottky transistor) 56
 译码器(Decoder) 133
 二进制(binary)~ 133
 二-十进制(BCD)~ 137
 时序逻辑电路(Sequential logic circuit) 212
 时钟(Clock)
 ~脉冲(pulse) 7
 位(Bit) 6, 7

~线(line)
 状态(State) 180
 ~表(table)
 ~图(diagram) 191
 次(next)~ 185
 现(present)~ 185
 补码(Complement code) 462, 394, 163
 余3码(Excess three code) 20
 运算电路(Arithmetic circuit) 154
 取样-保持电路(Sample-hold circuit) 402, 403
 驱动方程(Driving, equation) 214, 219

八 画

非门(见门)
 或门(见门)
 或非门(见门)
 线与(Wrie-AND) 51
 单稳态触发器(Monostable multivibrator) 351~364, 373~375
 定时器(Timer) 371~377
 时序图(Timing diagram) 7
 组合逻辑电路(Combinational logic circuit) 89
 建立时间(Setup time) 200
 金属-氧化物-半导体(Metal-Oxide-Semiconductor, MOS) 9
 互补(Complemental~, CMOS) 9
 现场可编程门阵列(Field Programmable Gate Array, FPGA) 9, 323
 奇偶校验(Odd-even check) 106
 拉电流(Draw-off current) 47, 48
 参考电压(Reference voltage) 58
 函数产生器(Function generator) 146

九 画

施密特触发器(Schmitt trigger) 364~371, 377

保持时间(Hold time) 200
 恢复时间(Recovery time) 30
 总线(Bus) 85
 复位(Reset) 180
 复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD) 308
 相邻项(Adjacencies) 102
 荧光数码管(Fluorescent nixie tube) 139

十 画

真值表(Truth table) 21~23
 特性方程(Characteristic equation) 185
 扇出(Fan out) 47,48
 扇入(Fan in) 47,48
 竞争冒险(Race and hazard) 113
 浮置栅雪崩注入式 MOS(Floating Gate Avalanche injection MOS FAMOS) 290
 读/写控制(Read-write control) 280
 射极耦合逻辑(见逻辑)
 逐次逼近 A/D 转换器(见模数转换器) 407~410
 高阻态(High impedance state) 54
 高电平输入电流(High alevel input current) 201
 通用阵列逻辑器件(Generic Array Logic, GAL) 300

十 一 画

逻辑(Logic)
 ~电平(level) 3
 ~函数(function) 91
 ~变量(variables) 91
 ~常量(constant) 92
 ~符号(symbol)
 ~门(gate)
 ~表达式(expression) 90
 二极管-三极管(Diode-Transistor

~,DTL)
 三极管-三极管(Transistor-Transistor~,TTL)
 射极耦合(Emitter-Coupled~,ECL) 58
 寄存器(Register) 240,260,261
 移位寄存器(Shift register) 261~266
 双向(bidirectional)~ 263
 减法器(Subtractor) 163
 基本 RS 触发器(Basic RS flip-flop) 179

寄存器传输语言(Register Transfer Language, RTL) 428,444~449

十 二 画

辉光数码管(Glow discharge nixie tube) 139
 超前进位(Look-ahead carry) 159
 ~产生器(generator) 161
 ~加法器(adder) 159
 量化(Quantification) 403
 编码(Coding) 125
 编码器(Encoder) 125
 优先(priority)~ 128
 锁存器(Latch)
 集电极开路门(见门)
 最小项(Miniterm) 96
 硬件描述语言(Hardware Description Language, HDL) 430
 ABEL (Advanced Boolean Equation Language)~ 78,464

十 三 画

触发器(Flip-Flop) 179
 同步(Synachronous)~ 184
 时钟(Clocked)~ 184
 主从(master-slave)~ 186
 边沿(edge-triggered)~ 188

- | | | | |
|---|---------|--|--------------|
| 数字电路(Digital circuit) | 9 | er, ADC) | 8, 402~418 |
| 数码显示器(Digital display) | 139 | 双积分(dual slope)~ | 410~414 |
| 数字比较器(Digital comparator) | 149 | 逐次逼近(successive approximation) | |
| 数模转换器(Digital to Analog Converter, DAC) | 388~402 | ~ | 407~410 |
| 权电阻(weighted resistor)~ | 389 | 算术逻辑单元(Arithmetic Logic Unit, ALU) | 166 |
| 权电流(weighted current)~ | 391 | 算法状态机(Algorithmic state machine) | |
| 倒置梯型(inverted ladder)~ | 389~391 | | 428, 431~443 |
| 数据选择器(Multiplexer) | 143 | 算法状态机图(Algorithmic State Machine diagram, ASM diagram) | |
| 数据分配器(Demultiplexer) | 141 | | 428, 431~443 |

十四画

- | | |
|----------------------------------|----|
| 模拟开关(Analog switch) | 68 |
| 模数转换器(Analog to Digital Convert- | |

十五画

- | | |
|---------------------------|----|
| 摩根定理(De Morgan's theorem) | 90 |
| 噪声容限(Noise margin) | 47 |