普通高等教育"十一五"规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUIHUA JIAOCAI

SHUZI DIANZI JISHU JICHU

数字电子技术基础

李月乔 主编

中国电力出版社
http://jc.cepp.com.cn

普通高等教育“十一五”规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUIHUA JIAOCAI

SHUZI DIANZI JISHU JICHU

数字电子技术基础

主　编　李月乔
编　写　刘向军　刘春颖　王　赟
　　文亚凤　陈晓梅　赵莲清
主　审　朱承高

中国电力出版社
http://jc.cepp.com.cn

内 容 提 要

本书为普通高等教育“十一五”规划教材。

全书共分十章，主要内容包括数字逻辑基础、逻辑门电路基础、组合逻辑电路、触发器、时序逻辑电路、脉冲波形的产生与整形、半导体存储器、可编程逻辑器件与VHDL语言、数模与模数转换电路、数字系统设计。本书基本概念清晰，例题讲解步骤详细，习题丰富且综合性强；注重培养学生的综合能力，将理论知识与实际数字系统设计充分结合；配有电子教案，易于教学安排。

本书主要作为普通高等学校电气信息类专业的本科生教材，也可作为从事相关工作工程技术人员的参考用书。

图书在版编目（CIP）数据

数字电子技术基础/李月乔主编. —北京：中国电力出版社，2008.2（2014.1重印）

普通高等教育“十一五”规划教材

ISBN 978-7-5083-6488-9

Ⅰ.数… Ⅱ.李… Ⅲ.数字电路—电子技术—高等学校—教材 Ⅳ.TN79

中国版本图书馆CIP数据核字（2007）第195993号

中国电力出版社出版、发行

（北京市东城区北京站西街19号 100005 http：//jc.cepp.com.cn）

北京丰源印刷厂印刷

各地新华书店经售

*

2008年2月第一版 2014年1月北京第四次印刷

787毫米×1092毫米 16开本 26.75印张 655千字

定价 **38.00** 元

敬 告 读 者

本书封底贴有防伪标签，刮开涂层可查询真伪

本书如有印装质量问题，我社发行部负责退换

版 权 专 有 翻 印 必 究

前　言

为贯彻落实教育部《关于进一步加强高等学校本科教学工作的若干意见》和《教育部关于以就业为导向深化高等职业教育改革的若干意见》的精神，加强教材建设，确保教材质量，中国电力教育协会组织制订了普通高等教育“十一五”教材规划。该规划强调适应不同层次、不同类型院校，满足学科发展和人才培养的需求，坚持专业基础课教材与教学急需的专业教材并重、新编与修订相结合。本书为新编教材。

数字电子技术基础是电类各专业的重要技术基础课，处于各专业教学的中间环节，是学生基本素质形成的关键课程。本书是为电类各专业的本科生学习数字电路基础知识而写的，满足数字电路教学的基本要求。

本书的编写力求突出重点，基本概念明确清晰，例题讲解步骤详细。每章后都附有一定数量的习题，帮助学生加深对课程内容的理解，其中部分习题有一定的深度，可使学生在深入掌握课程内容的基础上扩展知识；部分习题综合了多个章节的内容，以锻炼学生综合运用知识的能力。最后一章结合数据移位计算乘法器、彩灯控制器、抢答器等实际的数字系统设计来综合数字电路各部分的知识，并介绍数字系统设计流程，有利于培养学生的综合应用能力，这种讲解方法是本书的特色。

本书采用了国家标准逻辑符号，使读者对中规模集成电路的理解和灵活运用更加容易，弥补了框图符号的不足。本书对可编程逻辑器件和 VHDL 等内容加大了介绍篇幅，可以使学生很快掌握现代数字系统设计方法的精髓。

本书共分十章，第一章重点论述了数字逻辑的基础知识，概念解释详尽，对于学生学习过程中容易出现问题的地方进行了重点强调；第二章讲述了逻辑门电路的基础知识；第三章讲述了组合逻辑电路的分析和设计。第四章内容是触发器的基本知识；第五章讲述了时序逻辑电路的分析和设计；第六章讲述了脉冲波形的产生和变换；第七章讲述了半导体存储器；第八章讲述了可编程逻辑器件和 VHDL 语言，对 VHDL 语言的使用进行了总结；第九章是数模与模数转换电路；第十章是数字系统设计，用四个具体设计实例详细解释了数字系统的设计方法。中规模组合和时序逻辑电路芯片的逻辑符号采用了国家标准符号，非常便于芯片的使用和逻辑功能的记忆。

本教材配备教师授课使用的电子教案，授课教师可以对授课内容任意组织，以适应学生对知识的理解。

编写本书的教师多年从事电子电路课程的教学和改革，书中汇集了教师多年的教学经验和体会。李月乔任本书主编，其中刘向军编写第四章、第九章；刘春颖编写第六章；王赟、赵莲清、陈晓梅、文亚凤、李月乔编写第一章、第二章、第三章、第五章；李月乔编写第七章、第八章、第十章，并对全书进行了整理和统稿。

上海交通大学朱承高教授审阅了全部书稿，并提出了许多宝贵意见，在此表示诚挚的谢意。

本书在编写过程中得到了张根保教授和谢志远教授的大力支持，在此表示诚挚的

谢意。

由于编者水平有限，书中难免存在不妥之处，希望读者批评指正，并将意见和建议反馈给我们，邮箱地址为 lyqiao@ncepu.edu.cn。

编者

2008 年 1 月

目　　录

前言

第一章　数字逻辑基础 …… 1
第一节　概述 …… 1
第二节　数制 …… 2
第三节　各种数制之间的转换 …… 5
第四节　码制 …… 9
第五节　逻辑问题的描述 …… 12
第六节　逻辑代数基础 …… 18
第七节　逻辑函数的五种描述方法 …… 21
第八节　逻辑函数的化简 …… 28
小结 …… 37
习题 …… 38
第二章　逻辑门电路基础 …… 42
第一节　二极管、三极管的开关特性 …… 42
第二节　二极管逻辑门电路 …… 49
第三节　TTL 逻辑门电路 …… 53
第四节　射极耦合逻辑门电路 …… 69
第五节　CMOS 逻辑门电路 …… 71
第六节　各种工艺的逻辑门之间的接口问题 …… 77
小结 …… 80
习题 …… 81
第三章　组合逻辑电路 …… 88
第一节　组合逻辑电路的分析 …… 88
第二节　用小规模集成电路（SSI）实现组合逻辑电路的设计 …… 89
第三节　组合逻辑电路中的竞争—冒险 …… 94
第四节　编码器 …… 99
第五节　译码器 …… 109
第六节　数据选择器 …… 124
第七节　算术运算电路 …… 130
第八节　数值比较器 …… 139
小结 …… 146
习题 …… 147

第四章　触发器 …… 152
第一节　触发器的电路结构及工作特点 …… 152
第二节　触发器的逻辑功能及功能转换 …… 170
第三节　集成触发器的主要参数 …… 174
小结 …… 175
习题 …… 175
第五章　时序逻辑电路 …… 180
第一节　概述 …… 180
第二节　时序逻辑电路的分析 …… 183
第三节　同步时序逻辑电路的设计 …… 190
第四节　计数器 …… 208
第五节　常用中规模计数器芯片及应用 …… 219
第六节　数码寄存器与移位寄存器 …… 251
小结 …… 260
习题 …… 261
第六章　脉冲波形的产生与整形 …… 269
第一节　概述 …… 269
第二节　555 定时器芯片 …… 269
第三节　脉冲波形产生电路 …… 272
第四节　脉冲波形整形电路 …… 280
小结 …… 288
习题 …… 288
第七章　半导体存储器 …… 294
第一节　存储器的基本概念和分类 …… 294
第二节　半导体存储器 …… 294
第三节　只读存储器（ROM） …… 295
第四节　随机存取存储器（RAM） …… 315
小结 …… 321
习题 …… 322
第八章　可编程逻辑器件与 VHDL 语言 …… 327
第一节　概述 …… 327
第二节　可编程逻辑器件 …… 330
第三节　硬件描述语言 VHDL 的基本语法 …… 338
第四节　基本的 VHDL 的并行语句和串行语句 …… 351
第五节　基本硬件电路模块的 VHDL 模型 …… 356
小结 …… 359
习题 …… 359
第九章　数模与模数转换电路 …… 362
第一节　概述 …… 362

第二节　D/A 转换器 …… 363
第三节　A/D 转换器 …… 374
小结 …… 385
习题 …… 386
第十章　数字系统设计 …… 388
第一节　概述 …… 388
第二节　算法状态机 …… 389
第三节　数字系统设计举例之一 …… 390
第四节　数字系统设计举例之二 …… 394
第五节　数字系统设计举例之三 …… 398
第六节　数字系统设计举例之四 …… 411
小结 …… 416
习题 …… 416
参考文献 …… 419

第一章 数字逻辑基础

本章提要

本章首先介绍模拟信号与数字信号、数字电路的一些基本概念，然后讨论数字电路中常用的数制与码制，接着介绍基本逻辑运算、逻辑函数及其表示方法，最后讨论逻辑函数的代数化简法和卡诺图化简法。

第一节 概 述

一、模拟信号与数字信号

在自然界中有许许多多的物理量，尽管它们的性质各不相同，但就其变化规律的特点而言，不外乎分两大类。一类物理量在时间上和数值上都是连续的，这一类物理量称为模拟量。表示模拟量的信号叫做模拟信号（Analog Signal）。例如，用热电偶测量温度，将被测温度以电信号形式输出，因为在任何情况下温度都不可能发生突变，所以该信号无论在时间上还是数值上都是连续的，属于模拟信号。而传输、处理模拟信号的电路称为模拟电路。另一类物理量在时间上和数值上都是离散的，也就是说，它们的变化在时间上是不连续的，总是发生在一系列离散的瞬间，在数值上也只能取某几个固定的值，这一类物理量叫数字量。我们把表示数字量的信号叫做数字信号（Digital Signal）。例如，生产中自动记录零件个数的计数信号，由计算机键盘输入计算机的信号等。

二、数字电路

传输、处理数字信号的电路被称为数字电路（Digital Circuit）。

与模拟电路不同，在数字电路中的信号是数字信号，其取值只有高电平和低电平两种。电路中的三极管工作在开关状态，即饱和状态和截止状态。数字电路研究的主要问题是电路的输入信号和输出信号之间的逻辑关系。分析数字电路的数学工具是逻辑代数，分析数字电路逻辑功能的主要方法是真值表、逻辑表达式、逻辑图和波形图等。

数字电路的特点主要表现在以下几个方面。

1. 逻辑运算能力

数字电路不仅能够完成算术运算，而且能够完成逻辑运算，且具有逻辑推理和逻辑判断的能力，因此被称为数字逻辑电路或逻辑电路。计算机也因为这种逻辑思维能力而被称为电脑。

2. 抗干扰能力强

模拟电路主要研究的是输出、输入之间的数量关系，而在数字电路中，是根据数字信号的脉冲宽度、脉冲频率、脉冲个数等条件研究输出、输入之间的逻辑关系。数字信号在传输和处理过程中，来自不同方面的干扰往往只会影响到脉冲的幅度，而对数字脉冲信号的个数、频率、脉宽等表征逻辑关系的参数没有影响，即使出现大的干扰导致脉冲信号发生改变，也能通过纠错的方法进行校正，所以数字电路的抗干扰能力较强，便于使用、维护和进

行故障诊断。

3. 功耗低

在模拟电路中，由于三极管等半导体元件基本工作在放大状态，整体上功耗较大。而在数字电路中，三极管等半导体元件一般工作在开关状态，即交替工作在饱和状态和截止状态，数字电路的功耗较低，所以目前数字电路的集成度可以做得很高。

4. 电路结构简单，通用性强

数字电路传输、处理的数字信号实质上就是二值数据，即高电平和低电平。具有稳定的高、低电平输出的电路都可作为数字电路的基本单元，数字电路的很多功能都能模块化、单元化，复杂的数字电路都可以由这些基本模块组成。因此，数字电路便于集成和系列化生产，通用性强，成本低，使用方便。除此之外，数字电路对元器件参数精度的要求不高，因为高、低电平的取值有一定的容限范围。

5. 保密性好

对数字信号可以采取很多方法进行加密处理，所以数字电路对数字化的信息有很强的保密性。

由于数字电路具有这些优点，所以其发展非常迅速。数字电路研究的对象是很广泛的，包括脉冲信号的产生、放大、整形、控制、记忆、计数、显示和电路输入、输出间的逻辑关系等问题。数字电路现已广泛应用于计算机、数字通信、数字仪表、自动控制、遥测技术等各个领域。

第二节 数　　制

人们在日常生活中经常遇到计数问题，并且习惯于用十进制数。而在数字系统中，通常采用二进制数，有时也采用八进制数或十六进制数。

计数的体制和方法就是数制。在某种数制中所使用的符号称为数码，数码的个数称为基(Base，Radix)。不同的数制有不同的计数规律。下面分别介绍十进制、二进制、八进制和十六进制这四种数制。

每种数制中的数有位置计数法和多项式计数法两种表示形式。

一、十进制

十进制是日常生活中最常使用的一种计数体制。十进制有 0、1、2、3、4、5、6、7、8、9 这 10 个数码，数码的个数称为基，所以十进制的基是 10，这个 10 本身是十进制数，用 $(10)_D$ 表示。因为在本书中要同时介绍四种数制，所以我们用下标来区分这些不同的数制。十进制的计数规律是“逢十进一”。

因为人类有 10 个手指和 10 个脚趾，在人类祖先认知的过程中，一开始用手指或脚趾来计数，所以产生了十进制的计数方法。假如我们人类有 11 个手指和 11 个脚趾，那么现在使用的可能就是十一进制了。

(1) 位置计数法

$$N_D = (k_{n-1}k_{n-2}\cdots k_1k_0.k_{-1}\cdots k_{-m})_D \tag{1-1}$$

式中：N_D 中的下标 D 表示十进制数 (Decimal)，十进制数的下标也可以用 10 来表示；小数点左边的部分是整数部分，有 n 个整数位；小数点右边的部分是小数部分，有 m 个小数

位；k_{n-1}是最高有效位；k_{-m}是最低有效位。

处于不同位置上的数码所代表的数值大小是不同的，即位权不同。十进制数中每个位置上的位权为10^i，i称做位号，表示第i位，以k_i表示第i位的数码，可以是0、1、2、3、4、5、6、7、8、9中的任意一个。

（2）多项式计数法

$$N_D = k_{n-1} \times 10^{n-1} + k_{n-2} \times 10^{n-2} + \cdots + k_1 \times 10^1 + k_0 \times 10^0 + k_{-1} \times 10^{-1} + \cdots + k_{-m} \times 10^{-m}$$
$$= \sum_{i=-m}^{n-1} k_i \times 10^i \tag{1-2}$$

对式（1-2）进行十进制的四则运算，可以得到这个数的十进制形式的数。

【例 1-1】 将位置计数法形式的十进制数$(5555)_D$转换成多项式计数法形式。

解 将每一位数码乘以该位的权值，然后相加，可得该十进制数的多项式计数法形式为

$$(5555)_D = 5 \times 10^3 + 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0$$

十进制虽然是人们习惯使用的计数体制，但是，要让一个器件或一个电路具有10个不同状态，与十进制的10个不同的数码一一对应起来，实现起来有一定的困难。因此在数字电路中不直接使用十进制的计数体制。

二、二进制

二进制是数字系统中最常使用的一种计数体制。二进制有0、1这两个数码，数码的个数称为基，所以二进制的基是2，这个2本身是十进制数。二进制的计数规律是“逢二进一”。

因为现在实际存在的器件都有两个稳定的状态，比如以前的电子管、半导体二极管、双极型三极管、场效应管等，所以有二进制计数的物理基础。因此在数字系统中使用二进制。

（1）位置计数法

$$N_B = (k_{n-1}k_{n-2}\cdots k_1k_0.k_{-1}\cdots k_{-m})_B \tag{1-3}$$

式中：下标B表示二进制数（Binary），二进制数的下标也可以用2来表示；小数点左边的部分是整数部分，有n个整数位；小数点右边的部分是小数部分，有m个小数位；k_{n-1}是最高有效位；k_{-m}是最低有效位。

二进制数中每位上的位权为2^i，i称做位号，表示第i位，以k_i表示第i位的数码，可以是0、1中的任意一个。

（2）多项式计数法

$$N_B = k_{n-1} \times 2^{n-1} + k_{n-2} \times 2^{n-2} + \cdots + k_1 \times 2^1 + k_0 \times 2^0 + k_{-1} \times 2^{-1} + \cdots + k_{-m} \times 2^{-m}$$
$$= \sum_{i=-m}^{n-1} k_i \times 2^i \tag{1-4}$$

对式（1-4）进行十进制的四则运算，可以得到这个数的十进制形式的数。

【例 1-2】 将位置计数法形式的二进制数$(11011.101)_B$转换成多项式计数法形式。

解 将每一位的二进制数码乘以该位的权值，然后相加，可得

$$(11011.101)_B = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

三、八进制

当一个数比较大时，采用二进制时，就会遇到二进制码的个数比较多的问题。这时可以

采用八进制来表示，这样数码的个数可以少一些，书写起来比较简便，不易出错。八进制只是二进制的一种简便表示方法而已。

八进制有0、1、2、3、4、5、6、7这8个数码，所以八进制的基是8，这个8本身是十进制数。八进制的计数规律是“逢八进一”。

（1）位置计数法

$$N_{\mathrm{O}}=(k_{n-1}k_{n-2}\cdots k_1k_0.k_{-1}\cdots k_{-m})_{\mathrm{O}} \tag{1-5}$$

式中：下标O表示八进制数（Octal），八进制数的下标也可以用8来表示；小数点左边的部分是整数部分，有n个整数位；小数点右边的部分是小数部分，有m个小数位；k_{n-1}是最高有效位；k_{-m}是最低有效位。

八进制数中每位上的位权为8^i，i称做位号，表示第i位，以k_i表示第i位的数码，可以是0、1、2、3、4、5、6、7中的任意一个。

（2）多项式计数法

$$\begin{aligned}N_{\mathrm{O}}&=k_{n-1}\times 8^{n-1}+k_{n-2}\times 8^{n-2}+\cdots+k_1\times 8^1+k_0\times 8^0+k_{-1}\times 8^{-1}+\cdots+k_{-m}\times 8^{-m}\\&=\sum_{i=-m}^{n-1}k_i\times 8^i\end{aligned} \tag{1-6}$$

对式（1-6）进行十进制的四则运算，可以得到这个数的十进制形式的数。

【例1-3】 将位置计数法形式的八进制数$(207.04)_{\mathrm{O}}$转换成多项式计数法形式。

解 将八进制数按权展开，可得

$$(207.04)_{\mathrm{O}}=2\times 8^2+0\times 8^1+7\times 8^0+0\times 8^{-1}+4\times 8^{-2}$$

四、十六进制

当一个数比较大时，为了避免数码个数太多，除了可以采用上面介绍的八进制来表示外，还可以用十六进制表示。十六进制码比八进制码的个数更少一些，书写起来更简便，不易出错。十六进制也是二进制的一种简便表示方法。

十六进制有0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F这16个数码，所以十六进制的基是16，这个16本身是十进制数。十六进制的计数规律是“逢十六进一”。

（1）位置计数法

$$N_{\mathrm{H}}=(k_{n-1}k_{n-2}\cdots k_1k_0.k_{-1}\cdots k_{-m})_{\mathrm{H}} \tag{1-7}$$

式中：下标H表示十六进制数（Hexadecimal），十六进制数的下标也可以用16来表示；小数点左边的部分是整数部分，有n个整数位；小数点右边的部分是小数部分，有m个小数位；k_{n-1}是最高有效位；k_{-m}是最低有效位。

十六进制数中每位上的位权为16^i，i称做位号，表示第i位，以k_i表示第i位的数码，可以是0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F中的任意一个。

（2）多项式计数法

$$\begin{aligned}N_{\mathrm{H}}&=k_{n-1}\times 16^{n-1}+k_{n-2}\times 16^{n-2}+\cdots+k_1\times 16^1+k_0\times 16^0+k_{-1}\times 16^{-1}+\cdots+k_{-m}\times 16^{-m}\\&=\sum_{i=-m}^{n-1}k_i\times 16^i\end{aligned} \tag{1-8}$$

对式（1-8）进行十进制的四则运算，可以得到这个数的十进制形式的数。

【例1-4】 将位置计数法形式的十六进制数$(\mathrm{D8.A})_{\mathrm{H}}$转换成其多项式计数法形式。

解　将十六进制数按权展开，得到

$$(D8.A)_H = 13\times16^1 + 8\times16^0 + 10\times16^{-1}$$

五、任意进制的一般规律

按照上面二进制、十进制、八进制和十六进制的表示方法类推，任意 N 进制数 Y 的位置计数法和多项式计数法分别如式（1-9a）、式（1-9b）所示。

$$Y_N = (k_{n-1}k_{n-2}\cdots k_1k_0.k_{-1}\cdots k_{-m})_N \tag{1-9a}$$

$$Y_N = \sum_{i=-m}^{n-1} k_i \times N^i \tag{1-9b}$$

式中：Y 是 N 进制数；k_i 为第 i 位的数码；N^i 为第 i 位的权；N 是任意进制数的基；n、m 为正整数，分别代表 N 进制数的整数部分的位数和小数部分的位数。

第三节　各种数制之间的转换

既然同一个数可以用十进制、二进制、八进制、十六进制等各种数制的计数方法表示，那么它们之间必然存在着一定的转换关系。下面分别介绍它们之间的转换方法。

一、十进制和二进制之间的相互转换

人们在日常生活中习惯使用十进制数，而计算机等数字系统中使用二进制数。因此，需要把一个十进制数转换成二进制数以后，再利用计算机进行各种计算，运算结束后，将计算机结果输出时，还需把二进制数再转换为十进制数。

（一）二进制数转换成十进制数

任意一个二进制数都可以表示成多项式计数法的形式，只要对多项式进行十进制的四则运算，就能把一个二进制数转换成对应的十进制数。

【例 1-5】　将二进制数 $(1011001.001)_B$ 转换为对应的十进制数。

解　将每一位二进制数乘以该位的权值，然后相加，运算法则采用十进制的运算法则，可得

$$\begin{aligned}(1011001.001)_B &= 1\times2^6+0\times2^5+1\times2^4+1\times2^3+0\times2^2\\ &\quad +0\times2^1+1\times2^0+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}\\ &= 64+0+16+8+0+0+1+0+0+0.125\\ &= (89.125)_D\end{aligned}$$

（二）十进制数转换成二进制数

十进制数转换为二进制数的过程相对复杂一些，对整数部分与小数部分要分别采用不同的转换方法。

1. 整数部分

设十进制整数 $(A)_D$ 等于二进制整数 $(k_nk_{n-1}\cdots k_0)_B$，则有

$$\begin{aligned}(A)_D &= k_n\times2^n + k_{n-1}\times2^{n-1} + \cdots + k_1\times2^1 + k_0\times2^0\\ &= 2(k_n\times2^{n-1} + k_{n-1}\times2^{n-2} + \cdots + k_1) + k_0\end{aligned} \tag{1-10}$$

式（1-10）表明，若将十进制整数 $(A)_D$ 除以 2（2 是二进制数的基），并进行十进制的除法运算，则得到的余数为 k_0，余数只能是 1、0。依次类推，直到商为 0 为止，则可分别得

到 k_1、k_2、…、k_n。

由于这种把十进制整数转换成二进制整数的方法是通过除 2 取余数的方法来完成的，故简称为“除 2 取余”法。其中先得到的余数是等值二进制数的最低有效位（LSB，Least Significant Bit），最后得到的余数是等值二进制数的最高有效位（MSB，Most Significant Bit）。

2. 小数部分

设十进制小数 $(A)_D$ 等于二进制小数 $(0.k_{-1}k_{-2}\cdots k_{-m})_B$，则有

$$(A)_D = k_{-1}\times 2^{-1} + k_{-2}\times 2^{-2} + \cdots + k_{-n}\times 2^{-m} \tag{1-11}$$

式（1-11）两边同乘以 2，并进行十进制的四则运算，得到

$$2\times(A)_D = k_{-1} + (k_{-2}\times 2^{-1} + \cdots + k_{-n}\times 2^{-m+1}) \tag{1-12}$$

式（1-12）表明，若将十进制小数 $(A)_D$ 乘以 2（2 是二进制数的基），并进行十进制的乘法运算，则所得乘积的整数部分为 k_{-1}，把整数部分拿走，剩下小数部分。剩下的小数部分再乘以 2，进行十进制的乘法运算，直到小数部分为 0 为止。依次类推，则可分别得到 k_{-2}、k_{-3}、…、k_{-m}。

十进制的小数部分转换成二进制小数是通过乘 2 取整数的方法完成的，所以简称“乘 2 取整”法，用它先得到的整数是二进制小数的高位，后得到的整数是二进制小数的低位。

小数部分“乘 2 取整”的过程，最后不一定能使小数部分为 0，因此转换值存在误差。通常在二进制小数的精度已达到预定的要求时，运算便可结束。

【例 1-6】 将十进制小数 $(0.1875)_D$ 转换为二进制小数。

解 运算法则采用十进制的运算法则，用“乘 2 取整”法，按下列步骤转换。

0.1875×2=0.375 ……整数部分为 0

0.3750×2=0.75 …… 整数部分为 0

0.7500×2=1.5 …… 整数部分为 1

0.5000×2=1.0 …… 整数部分为 1

小数部分已经是 0，运算结束。

因此，$(0.1875)_D=(0.0011)_B$。

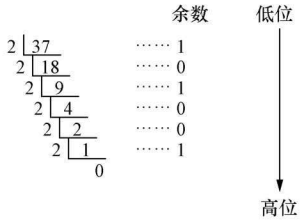


图 1-1 [例 1-7] 整数部分的“除 2 取余”运算过程

【例 1-7】 将 $(37.41)_D$ 转化为二进制数，要求其误差不大于 2^{-5}。

解 由于整数和小数的转换方法不同，分别对其整数部分和小数部分进行转换，然后再将两部分转换结果合并，可得对应的二进制数。运算法则采用十进制的运算法则。

（1）对整数部分处理，其运算过程如图 1-1 所示。因此，整数部分对应的二进制数为 $(100101)_B$。

（2）对小数部分处理。题目中要求误差不大于 2^{-5}，即小数部分保留到－5 位号，运算过程为

0.41×2=0.82 …… 整数部分为 0

0.82×2=1.64 …… 整数部分为 1

0.64×2=1.28 …… 整数部分为 1

$0.28\times2=0.56$　……　整数部分为 0

$0.56\times2=1.12$　……　整数部分为 1

所以，$(37.41)_D=(100101.01101)_B$。

同理，若将十进制数转换成任意 N 进制数，则整数部分转换采用“除 N 取余”法，小数部分转换采用“乘 N 取整”法。

二、十进制和八进制之间的相互转换

（一）八进制数转换成十进制数

同二进制数转换成十进制数的方法类似，任意 1 个八进制数都可以表示成多项式计数法的形式，只要对多项式进行十进制的四则运算，就能把 1 个八进制数转换成十进制数。

【例 1-8】 将 $(706.001)_O$转化为十进制数。

解　运算法则采用十进制的运算法则，得

$$
\begin{aligned}
(706.001)_O &= 7\times8^2+0\times8^1+6\times8^0+0\times8^{-1}+0\times8^{-2}+1\times8^{-3}\\
&=448+0+6+0+0+\frac{1}{512}\\
&=\left(454\frac{1}{512}\right)_D
\end{aligned}
$$

（二）十进制数转换成八进制数

同十进制数转换成二进制数的方法类似，把“除 2 取余”法和“乘 2 取整”法中的 2（2 是二进制的基）换成 8（8 是八进制的基），其余步骤按照十进制数转换成二进制数的方法即可。

【例 1-9】 将 $(44.375)_D$转化为八进制数。

解　对整数部分采用“除 8 取余”法，运算法则采用十进制的运算法则。小数部分采用“乘 8 取整”法，运算法则采用十进制的运算法则。两部分结合在一起然后得到完整结果。

（1）对整数部分处理。整数部分“除 8 取余”的运算过程如图 1-2 所示。

（2）对小数部分处理

$0.375\times8=3.0$　……　整数部分为 3

小数部分已经是 0，运算结束。

所以，$(44.375)_D=(54.3)_O$。

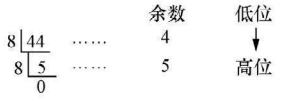


图 1-2　[例 1-9] 整数部分“除 8 取余”运算过程

三、十进制和十六进制之间的相互转换

（一）十六进制数转换成十进制数

同二进制数转换成十进制数的方法类似，任意 1 个十六进制都可以表示成多项式计数法的形式，只要对多项式进行十进制的四则运算，就能把 1 个十六进制数转换成十进制数。

【例 1-10】 将 $(7BA.E01)_H$转化为十进制数。

解　采用十进制的运算法则，有

$$
\begin{aligned}
(7BA.E01)_H &= 7\times16^2+11\times16^1+10\times16^0+14\times16^{-1}+0\times16^{-2}+1\times16^{-3}\\
&=1792+176+10+\frac{3584}{4096}+0+\frac{1}{4096}\\
&=\left(1978\frac{3585}{4096}\right)_D
\end{aligned}
$$

（二）十进制数转换成十六进制数

同十进制数转换成二进制数的方法类似，把“除 2 取余”法和“乘 2 取整”法中的 2（2 是二进制的基）换成 16（16 是十六进制的基），其余步骤按照十进制数转换成二进制数的方法即可。

【例 1-11】 将 $(154.375)_D$转化为十六进制数。

解 对整数部分采用“除 16 取余”法，运算法则采用十进制的运算法则。小数部分采用“乘 16 取整”法，运算法则采用十进制的运算法则。两部分结合在一起然后得到完整结果。

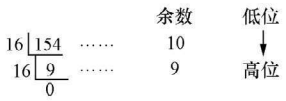


图 1-3 ［例 1-11］整数部分“除 16 取余”运算过程

（1）对整数部分处理。整数部分“除 16 取余”运算过程如图 1-3 所示。

（2）对小数部分处理

0.375×16＝6.0 …… 整数部分为 6

因为小数部分已经是 0，运算结束。所以，$(154.375)_D=(9A.6)_H$。

四、二进制数和八进制数之间的相互转换

二进制数和八进制数之间有很简单的对应关系，即 3 位二进制数对应 1 位八进制数，对应关系如表 1-1 所示。

表 1-1　　3 位二进制数与 1 位八进制数的对应关系

3 位二进制数	1 位八进制数	3 位二进制数	1 位八进制数
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

若将二进制数转换为八进制数，对二进制数的整数部分，从小数点左边的第一位起，往左数，每 3 位二进制数为一组变成对应的一个八进制数，若不够 3 位，在左边补 0；对二进制数的小数部分，从小数点右边的第一位起，往右数，每 3 位二进制数为一组变成对应的一个八进制数，若不够 3 位，在右边补 0。

若将八进制数转换为二进制数，只要将 1 位八进制数写成对应的 3 位二进制数即可。

【例 1-12】 将二进制数 $(01101111010.1011)_B$ 转换为对应的八进制数。

解 二进制数：001　101　111　010　.　101　100

八进制数：1　5　7　2　.　5　4

所以，$(01101111010.1011)_B=(1572.54)_O$

【例 1-13】 将 $(374.26)_O$ 转化为对应的二进制数。

解 八进制数：3　7　4　.　2　6

二进制数：011　111　100　.　010　110

所以，$(374.26)_O=(011111100.010110)_B=(11111100.01011)_B$

五、二进制数和十六进制数之间的相互转换

二进制数和十六进制数之间有很简单的对应关系，即 4 位二进制数对应 1 位十六进制数，对应关系如表 1 - 2 所示。

表 1 - 2　　4 位二进制数与 1 位十六进制数的对应关系

4 位二进制数	1 位十六进制数	4 位二进制数	1 位十六进制数
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

若将二进制数转换为十六进制数，对二进制数的整数部分，从小数点左边的第一位起，往左数，每 4 位二进制数为一组变成对应的 1 位十六进制数，若不够四位，在左边补 0；对二进制数的小数部分，从小数点右边的第一位起，往右数，每 4 位二进制数为一组变成对应的 1 位十六进制数，若不够四位，在右边补 0。

若将十六进制数转换为二进制数，只要将 1 位十六进制数写成对应的 4 位二进制数即可。

【例 1 - 14】 将 $(10101100)_B$ 和 $(0.10111101)_B$ 转化为对应的十六进制数。

解　由表1 - 2 可得

$$(10101100)_B=(AC)_H$$

$$(0.10111101)_B=(0.BD)_H$$

【例 1 - 15】 将 $(AF4.76)_H$ 转化为对应的二进制数。

解　由表 1 - 2 可得

$$(AF4.76)_H=(101011110100.01110110)_B$$

第四节　码　　制

计算机技术最初使用的目的纯粹是为了计算，后来 ASCII 码的引入使得文本成为计算机新的处理对象。所以，数字系统中的信息共有两类：一类是数值信息，是进行科学计算用的，这就是数制；另一类是文字符号信息，给文字符号信息编码的方法，这就是码制。

所谓码制就是编码的方法。编码，通俗地讲，就是起名字。在现实生活中，可以用汉字的组合给每人一个名字，还可以用十进制的数码组合给每个人赋予一组数字，也就是身份证代号。但是在数字系统中，我们只能用具有一定位数的二进制数码的组合来给各个信息起名字。

我们用 N 表示信息的个数，用 n 表示二进制码的位数，则它们之间的关系为

$$2^{n-1}<N\leqslant 2^n \tag{1-13}$$

一、BCD 码

用 4 位二进制数码的组合为十进制数的 10 个数码 0、1、2、3、4、5、6、7、8、9 起名字，这就是 BCD 码（Binary-Coded Decimal）。4 位二进制数码的组合共有 16 种，从其中任意地拿出 10 种，并任意地分配给十进制数的 10 个数码，那么将有很多种组合，所以这种起名字的方法有很多种。其中有几种方法比较有规律，便于记忆和使用，比较常用。表 1-3 中列出了几种常见的 BCD 码。

1. 8421 码

观察表 1-3 中的 8421 码这一列，如果规定各位的位权从左至右依次为 8、4、2、1，把 8421 码看成二进制数，这个二进制数所对应的十进制数，正好就是 8421 码所对应的十进制数码。因为有这样简单的对应关系，所以 8421 码是最容易记忆的一种 BCD 代码。8421 码不是二进制数，只是十进制数的 10 个数码 0、1、2、3、4、5、6、7、8、9 的一种名字而已。

2. 5421 码

表 1-3 中的 5421 码这一列，如果规定各位的位权从左至右依次为 5、4、2、1，将各位的值相加，正好就是所表示的十进制数码。

3. 2421 码

表 1-3 中的 2421 码这一列，如果规定各位的位权从左至右依次为 2、4、2、1，将各位的值相加，正好就是所表示的十进制数码。

4. 余 3 码

表 1-3 中的余 3BCD 码这一列，如果规定各位的位权从左至右依次为 8、4、2、1，把它看成二进制数，这个二进制数所对应的十进制数，比这个余 3 码所表示的十进制数，正好多 3，余 3 码就是从这个角度来命名的。

表 1-3　　几种常用的 BCD 码

十进制数	8421 码	5421 码	2421 码	余 3 码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

【例 1-16】 分别用 8421BCD 码和余 3 码表示十进制数 $(258.369)_D$。

解　由表 1-3 可得

$$(258.369)_D = (0010\ 0101\ 1000\ .\ 0011\ 0110\ 1001)_{8421码}$$
$$= (0101\ 1000\ 1011\ .\ 0110\ 1001\ 1100)_{余3码}$$

二、格雷码

格雷码是一种典型的循环码（Cyclic Code）。循环码有两个特点：

（1）相邻性。相邻性是指任意两个相邻的代码中仅有 1 位取值不同。

（2）循环性。循环性是指首尾的两个代码也具有相邻性。

凡是满足这两个特性的编码都称为循环码。在第五章介绍的时序逻辑电路中采用循环码编码时，不仅可以有效地防止波形出现毛刺（Glitch），而且可以提高电路的工作速度。

十进制数 0～15 的 4 位二进制格雷码（Gray Code）如表 1 - 4 所示。

表 1 - 4　　格　雷　码

二进制数				格雷码				二进制数				格雷码			
b_3	b_2	b_1	b_0	G_3	G_2	G_1	G_0	b_3	b_2	b_1	b_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0

显然，格雷码符合循环码的两个特性。例如，5 和 6 的两个代码分别为 0111 和 0101，只有次低位取值不同；首尾的 0 和 15 的两个代码分别为 0000 和 1000，也只有最高位取值不同。

三、ASCII 码

ASCII 码是美国信息交换标准代码（American Standard Code for Information Interchange）的简称，是目前国际上最通用的一种字符码。ASCII 码采用 7 位二进制编码表示十进制的 10 个数码、英文大小写字母、运算符、控制符以及特殊符号。计算机输出到打印机的字符码就采用 ASCII 码。表 1 - 5 是 ASCII 码表。

表 1 - 5　　ASCII　码　表

$b_6b_5b_4$ / $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENG	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	′	7	G	W	g	w

续表

$b_3b_2b_1b_0$ \ $b_6b_5b_4$	000	001	010	011	100	101	110	111
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	(
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	←	o	DEL

第五节 逻辑问题的描述

逻辑是指事物因果之间所遵循的规律。现实世界中存在的事物之间有很复杂的逻辑关系，但是无论多复杂的逻辑关系都是由最基本的三种逻辑关系组成。这三种最基本的逻辑关系就是与逻辑、或逻辑和非逻辑。

一、自然界中三种基本逻辑关系

1. 与逻辑关系

决定某一事物结果的所有条件同时具备，结果才会发生。这一因果关系称与逻辑关系。

2. 或逻辑关系

决定某一事物结果的诸条件只要有一个条件具备，结果就会发生。这一因果关系称或逻辑关系。

3. 非逻辑关系

决定某一事物结果的条件如果具备的话，结果就不发生。这一因果关系称非逻辑关系。

二、逻辑代数的由来

19 世纪英国数学家乔治·布尔（George Boole）于 1854 年首先提出了描述客观事物逻辑关系的数学方法——布尔代数。由于布尔代数被广泛地用于描述逻辑关系，所以也称布尔代数为逻辑代数。

逻辑代数是用于逻辑分析的数学工具，是在逻辑学的基础上发展起来的一门学科。它采用“一套符号”来描述逻辑思维，并将复杂的逻辑问题抽象为一种简单的符号演算，摆脱了冗繁的文字描述。“一套符号”指的是用字母表示条件、结果，称做逻辑变量（自变量、因变量），其取值只有两种可能并用符号 0、1 表示。0、1 本身没有数量的概念，只是用来表示条件具备还是不具备，结果发生还是不发生，在逻辑代数中，规定用符号 1 表示条件具备，用符号 0 表示条件不具备；用符号 1 表示结果发生，用符号 0 表示结果不发生。

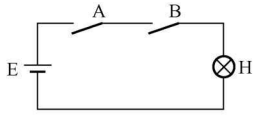


图 1-4 开关串联的控制灯泡亮灭的电路

下面分析图 1-4 所示开关串联的控制灯泡亮与灭的电路中的逻辑问题。

当该电路中开关 A 和 B 同时闭合时，灯 H 亮；当开关 A、

开关B中有任一个打开时，灯H灭。我们的目的可能有两个，一个是让灯如何亮；另一个是让灯如何灭。

（1）下面先分析“让灯如何亮”的问题，分析的目的是找到“让灯亮”的控制方法。

根据电路的基本知识，我们知道，当开关A和B同时闭合时，电路形成了闭合回路，灯H就能够亮，只要有一个开关不闭合，灯H就不亮，其关系如表1-6所示。可见，控制灯H亮还是不亮这件事的条件有两个：条件一是开关A闭合还是不闭合，条件二是开关B闭合还是不闭合。结果是灯亮还是不亮。

下面进行逻辑抽象，用符号M表示条件一（开关A闭合还是不闭合），用符号N表示条件二（开关B闭合还是不闭合），用符号P表示结果（灯H亮还是不亮）。如果开关A闭合用符号1表示，开关A不闭合用符号0表示，开关B闭合用符号1表示，开关B不闭合用符号0表示，灯H亮用符号1表示，灯H不亮用符号0表示，那么列出表格如表1-7所示，这就是真值表。真值表已经用符号的演算代替了自然语言对逻辑问题的描述和推理。

分析表1-7已经知道，在逻辑代数中，符号1表示的是条件具备，符号0表示的是条件不具备，符号1表示的是结果发生，符号0表示的是结果不发生。表1-7的最后一行表明当两个条件都具备时，结果才发生，前三行表明，只要有一个条件不具备，结果就不会发生，这正好符合“与”逻辑关系的定义，即决定某一事物结果的所有条件同时具备时，结果才会发生。

表1-6　灯H亮与不亮和开关A、B之间的关系

开关A	开关B	灯H
不闭合	不闭合	不亮
不闭合	闭合	不亮
闭合	不闭合	不亮
闭合	闭合	亮

表1-7　表1-6所对应的真值表

M	N	P
0	0	0
0	1	0
1	0	0
1	1	1

表1-7可以用下面的逻辑表达式来描述（符号“·”表示“与”逻辑运算，也可以省略）。

$$P = M \cdot N = MN \tag{1-14}$$

展开来，是如下四个表达式：

$$0 = 0 \cdot 0 \tag{1-15}$$

$$0 = 0 \cdot 1 \tag{1-16}$$

$$0 = 1 \cdot 0 \tag{1-17}$$

$$1 = 1 \cdot 1 \tag{1-18}$$

通过观察式（1-15）、式（1-16）、式（1-17）、式（1-18）可以看出，这种逻辑运算的表面看来与普通代数中的乘法运算完全一致，所以有时也称“与”逻辑运算为“逻辑乘”，有时也直接称为“乘”。要特别注意，之所以能这样来称呼，是因为在本书中研究的是逻辑问题，与普通代数没有关系。

式（1-15）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“与”的逻辑关系，现在这两个条件都不具备，所以结果不发生。

式（1-16）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“与”的逻辑关系，现在这两个条件一个不具备、另一个具备，所以结果不发生。

式（1-17）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是

“与”的逻辑关系，现在这两个条件一个具备、另一个不具备，所以结果不发生。

式（1-18）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“与”的逻辑关系，现在这两个条件都具备，所以结果发生。

（2）下面再来分析“让灯如何灭”的问题。分析的目的是找到“让灯灭”的控制方法。

根据电路的基本知识，我们知道，当开关A、B只要有一个打开，电路不能形成闭合回路，灯H就能够灭，关系如表1-8所示。控制灯H灭还是不灭这件事的条件有两个：条件一是开关A打开还是不打开，条件二是开关B打开还是不打开。我们称“开关A打开”为条件一具备，“开关A不打开”为条件一不具备，“开关B打开”为条件二具备，“开关B不打开”为条件二不具备，“灯H灭”为结果发生，“灯H不灭”为结果不发生。可以看出条件中只要有一个具备时，结果就能发生，所以这是“或”逻辑关系。

在“让灯如何灭”的逻辑问题中，这样来进行逻辑抽象，用符号M表示条件一（开关A打开还是不打开），用符号N表示条件二（开关B打开还是不打开），用符号P表示结果（灯H灭还是不灭）。开关A打开用符号1表示，开关A不打开用符号0表示；开关B打开用符号1表示，开关B不打开用符号0表示；灯H灭用符号1表示，灯H不灭用符号0表示。列出表格如表1-9所示，这就是真值表。

表1-8　灯H灭与不灭和开关A、B之间的关系

开关A	开关B	灯H
不打开	不打开	不灭
不打开	打开	灭
打开	不打开	灭
打开	打开	灭

表1-9　表1-8所对应的真值表

M	N	P
0	0	0
0	1	1
1	0	1
1	1	1

表1-9可以用逻辑表达式来描述（符号“+”表示“或”逻辑运算）。

$$P = M + N \tag{1-19}$$

展开来，是如下四个表达式：

$$0 = 0 + 0 \tag{1-20}$$

$$1 = 0 + 1 \tag{1-21}$$

$$1 = 1 + 0 \tag{1-22}$$

$$1 = 1 + 1 \tag{1-23}$$

通过观察式（1-20）、式（1-21）、式（1-22），可以看出，这种逻辑运算的表面看来与普通代数中的加法运算有些类似，而式（1-23）的表面看来与普通代数中的加法运算则完全不同。通常也称“或”逻辑运算为“逻辑加”，有时也直接称为“加”。要特别注意，之所以能这样来称呼，是因为在本书中我们研究的是逻辑问题，与普通代数没有关系。

式（1-20）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“或”的逻辑关系，现在这两个条件都不具备，所以结果不发生。

式（1-21）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“或”的逻辑关系，现在这两个条件一个不具备、另一个具备，所以结果发生。

式（1-22）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“或”的逻辑关系，现在这两个条件一个具备、另一个不具备，所以结果发生。

式（1-23）的含义为，有两个条件来决定这个逻辑问题的结果，这两个条件之间是“或”的逻辑关系，现在这两个条件都具备，所以结果发生。

通过上述分析，可以看出，对于同一件事，如果我们看待它的角度不同，其中蕴涵着的逻辑关系也不同，也可以这样说，对于图 1-4 所示电路，既可以实现“与”逻辑运算，又可以实现“或”逻辑运算。

用与分析图 1-4 相同的方法，对图 1-5 所示的开关并联的电路包含的逻辑问题进行分析，请读者自行完成。通过分析可以得知，图 1-5 所示的电路从不同的角度看，可以是“或”逻辑关系，也可以是“与”逻辑关系，既可以实现“或”逻辑运算，又可以实现“与”逻辑运算。

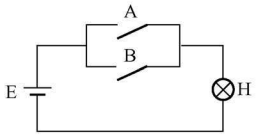


图 1-5　开关并联的控制灯泡亮灭的电路

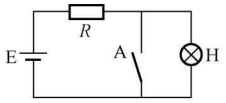


图 1-6　单开关控制灯泡亮灭的电路

【例 1-17】 分析图 1-6 所示的单开关控制灯泡亮灭的电路中的逻辑问题。

解　根据电路的基本知识可知，只要开关 A 闭合，将灯 H 短路，灯 H 就不亮。开关 A 不闭合，灯 H 两端有电压，灯 H 就亮。灯 H 亮与不亮和开关 A 之间的关系如表 1-10 所示。

约定：开关 A 用符号 M 表示，灯 H 用符号 P 表示；开关 A 不闭合用符号 0 表示，开关 A 闭合用符号 1 表示；灯 H 亮用符号 1 表示，灯 H 不亮用符号 0 表示。由此得到真值表如表 1-11 所示。

表 1-10　灯 H 亮与不亮和开关 A 之间的关系

开关 A	灯 H
不闭合	亮
闭合	不亮

表 1-11　　表 1-10 对应的真值表

M	P
0	1
1	0

逻辑表达式为

$$P = \overline{M} \tag{1-24}$$

三、逻辑代数中的三种基本逻辑运算以及一些复合逻辑运算

与自然界中的三种基本逻辑关系相对应，在逻辑代数中有三种基本的逻辑运算，即与逻辑运算、或逻辑运算、非逻辑运算。通过前面的例题，我们对这三种基本的逻辑运算已经有了一定的了解。

利用“与”、“或”、“非”三种基本运算，可构成各种复合逻辑运算，常见的有与非、或非、与或非、同或、异或等。实现这些逻辑运算的电路称为门电路。

1. 与逻辑运算

实现与逻辑运算的电路称为与门电路。其逻辑符号如图 1-7 所示。国标逻辑符号，即国家标准逻辑符号，后文中除特殊说明，所述逻辑符号均为国标逻辑符号。

2. 或逻辑运算

实现或逻辑运算的电路称为或门电路。其逻辑符号如图 1-8 所示。

图 1-7 与门的逻辑符号

(a) 与门的国标逻辑符号；

(b) 与门的国际流行逻辑符号

图 1-8 或门的逻辑符号

(a) 或门的国标逻辑符号；

(b) 或门的国际流行逻辑符号

图 1-9 非门的逻辑符号

(a) 非门的国标逻辑符号；

(b) 非门的国际流行逻辑符号

3. 非逻辑运算

实现非逻辑运算的电路称为非门电路。其逻辑符号如图 1-9 所示。

在现实生活中，最基本的与逻辑、或逻辑、非逻辑是不能完全描述事件之间的复杂的逻辑关系的，而必须用它们的组合才能进行完全的描述。所以我们还要研究一些常用的复合逻辑运算。

4. 与非逻辑运算

与非逻辑运算的表达式为

$$L = \overline{AB}$$

从与非逻辑运算的表达式可知，与非逻辑运算的结果是“有 0 出 1，全 1 出 0”。与非逻辑运算的真值表见表 1-12。实现与非逻辑运算的电路称为与非门，其逻辑符号如图 1-10 所示，图中小圆圈表示非运算。

表 1-12 与非逻辑运算的真值表

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

图 1-10 与非门的逻辑符号

(a) 与非门的国标逻辑符号；

(b) 与非门的国际流行逻辑符号

5. 或非逻辑运算

或非逻辑运算的表达式为

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

或非逻辑运算的结果是“有 1 出 0，全 0 出 1”。或非逻辑运算的真值表见表 1-13。实现或非逻辑运算的门电路的逻辑符号如图 1-11 所示。

表 1-13 或非逻辑运算的真值表

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

图 1-11 或非门的逻辑符号

(a) 或非门的国标逻辑符号；

(b) 或非门的国际流行逻辑符号

6. 异或逻辑运算

异或逻辑运算的表达式为

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

异或逻辑运算表示若两个输入变量 A、B 的取值不同，则输出变量 L 为 1；若 A、B 的取值相同，则 L 为 0。用一句话表示就是“相同为 0，不同为 1”。

异或逻辑运算的真值表见表 1-14。实现异或逻辑运算的电路叫异或门，其逻辑符号如图 1-12 所示。

表 1-14　　异或逻辑运算的真值表

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

图 1-12　异或门的逻辑符号

(a) 异或门的国标逻辑符号；

(b) 异或门的国际流行逻辑符号

当异或门的一个输入端为 0，比如 B=0，则 $L=A \oplus 0=A\overline{0}+\overline{A}0=A$，即输出信号 L 与输入信号 A 相等；当异或门的一个输入端为 1，比如 B=1，则 $L=A \oplus 1=A\overline{1}+\overline{A}1=\overline{A}$，即输出信号 L 与输入信号 A 反相。

7. 同或逻辑运算

同或逻辑运算的表达式为

$$L = A \odot B = AB + \overline{A}\,\overline{B} = \overline{A \oplus B}$$

同或逻辑运算表示若两个输入变量 A、B 的取值相同，则输出变量 L 为 1；若 A、B 取值相异，则 L 为 0。用一句话表示就是“相同为 1，不同为 0”。

同或逻辑运算的真值表见表 1-15。实现同或运算的电路叫同或门，其逻辑符号如图 1-13 所示。

表 1-15　　同或逻辑运算的真值表

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

图 1-13　同或门的逻辑符号

(a) 同或门的国标逻辑符号；

(b) 同或门的国际流行逻辑符号

8. 与或非逻辑运算

与或非逻辑运算的表达式为

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

与或非逻辑运算是与、或、非三种运算的组合。与或非逻辑运算的真值表见表 1-16。实现与或非逻辑运算的电路叫与或非门，其国标逻辑符号如图 1-14 所示。

表 1-16 与或非逻辑运算的真值表

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

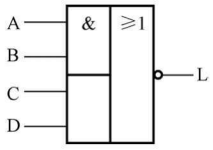


图 1-14 与或非门的国标逻辑符号

【例 1-18】 三个人表决一件事情，结果按“少数服从多数”的原则决定，试建立该逻辑函数。

解 (1) 分析文字描述，找出逻辑问题的条件和结果，条件为自变量，结果为因变量。三个人的意见为条件，约定分别用 A、B、C 表示；结果为能否通过，用 L 表示。每个人同意用 1 表示，不同意用 0 表示；结果能通过用 1 表示，不通过用 0 表示。

(2) 根据“少数服从多数”的原则，将输入变量不同取值组合与函数值间的对应关系列成表格，就建立了该逻辑函数的真值表。该真值表完全描述了该逻辑函数的功能。如表 1-17 所示（双竖线将真值表分为左、右两部分），真值表的第 1 行是输入变量和输出变量；左侧第 2 行表示 A、B、C 三个人都不同意这件事情，结果不能通过；左侧第 3 行表示当 A、B、C 三个人中只有 C 一个人同意这件事情，结果不能通过；左侧第 4 行表示当 A、B、C 三个人中只有 B 一个人同意这件事情，结果不能通过；左侧第 5 行表示当 A、B、C 三个人中有 B 和 C 两个人同意这件事情，结果能通过；右侧第 2 行表示当 A、B、C 三个人中只有 A 一个人同意这件事情，结果不能通过；右侧第 3 行表示当 A、B、C 三个人中有 A 和 C 两个人同意这件事情，结果能通过；右侧第 4 行表示当 A、B、C 三个人中有 A 和 B 两个人同意这件事情，结果能通过；右侧第 5 行表示当 A、B、C 三个人都同意这件事情，结果能通过。可以看出真值表把所有可能的组合情况全部列出来了，非常直观。

表 1-17 [例 1-18] 的真值表

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

第六节 逻辑代数基础

如前所述，逻辑代数用一套符号来描述逻辑思维指的是用字母表示逻辑问题中的条件、结果，称做逻辑变量（自变量、因变量），其取值只有两种可能，用符号 0、1 表示。普通代数在逻辑代数之前已经产生，人们对普通代数已经有一定的研究，逻辑代数借用了普通代数中的一

些运算法则，以便于人们快速记忆逻辑代数的一些运算规则。在逻辑代数中我们完全可以用符号3、4等其他任意的两个符号表示条件具备、不具备，但是习惯用1、0表示，因为这样逻辑代数与普通代数的某些运算规则相一致，便于我们对逻辑代数的记忆，而且约定条件具备用符号1表示、条件不具备用符号0表示，结果发生用符号1表示，结果不发生用符号0表示。

与普通代数不同的是，在逻辑代数中，不管是变量还是函数，其取值都只能是0或1，并且这里的0和1只表示两种不同的状态，没有数量的含义。

一、逻辑代数的基本定律

逻辑代数有一定的运算规律与规则，用这些规律与规则对逻辑表达式进行处理，可以完成对电路的简化、变换、分析和设计。

逻辑表达式是由逻辑变量和与、或、非三种基本运算符连接起来所构成的式子。在逻辑表达式中，等式右边的字母称为输入逻辑变量，等式左边的字母称为输出逻辑变量，字母上面没有非运算符的叫做原变量，有非运算符的叫做反变量。

设有两个逻辑函数

$$L_1 = f(A,B,C,D,\cdots),L_2 = g(A,B,C,D,\cdots)$$

它们的自变量都是A、B、C、D、…。

如果对应于自变量A、B、C、D、…的任何一组变量取值组合，L_1和L_2的值都相同，则称L_1和L_2是相等的，记为$L_1=L_2$。若两个逻辑函数相等，则它们的真值表一定相同，因此，要证明两个逻辑函数是否相等，只要看看它们的真值表是否相同即可。

逻辑代数的基本定律有0-1律、幂等律、互补律、结合律、交换律、分配律、德摩根定理等，如表1-18所示。要熟记表1-18给出的逻辑代数的基本定律。除了表1-18给出的逻辑代数的基本定律外，还可以推导出如下一些常用的公式：

$$A + AB = A \tag{1-25}$$

$$A(A + B) = A \tag{1-26}$$

$$A + \overline{A}B = A + B \tag{1-27}$$

冗余定理：

$$AB + \overline{A}C + BC = AB + \overline{A}C \tag{1-28}$$

$$AB + \overline{A}C + BCD = AB + \overline{A}C \tag{1-29}$$

上述公式不用死记硬背，可以在使用的过程中逐渐熟悉它们。下面将证明其中几个定律。

表1-18　逻辑代数的基本定律

	公式一	公式二	公式三
0-1律	$A+0=A$ $A+1=1$	$A\cdot 0=0$ $A\cdot 1=A$	$\overline{0}=1$ $\overline{1}=0$
幂等律	$A+A=A$	$A\cdot A=A$	
互补律	$A+\overline{A}=1$	$A\cdot\overline{A}=0$	$\overline{\overline{A}}=A$
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	$A(B+C)=AB+AC$	$A+BC=(A+B)(A+C)$	
德摩根定理	$\overline{A\cdot B\cdot C\cdots}=\overline{A}+\overline{B}+\overline{C}+\cdots$	$\overline{A+B+C+\cdots}=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdots$	

二、逻辑代数的三个基本规则

（一）代入规则

对于任何一个成立的逻辑等式，如果将等式两边出现的某变量 A，全部用另一个变量或变量的组合来代替，则等式仍然成立，这个规则称为代入规则。因为变量 A 的取值只能是 0 或 1，而其他任何变量或变量的组合的取值也只能是 0 或 1 两种情况，所以取代式中的 A 自然等式也成立。

例如，用代入规则很容易证明多变量德摩根定理（反演律）。

2 变量德摩根定理为$\overline{A+B}=\overline{A}\cdot\overline{B}$和$\overline{AB}=\overline{A}+\overline{B}$，以（B＋C）代替前一等式左边项的 B，以 BC 代替后一等式左边项的 B，可以得到

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

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

（二）反演规则

研究反演规则的目的是为了求反函数。

1. 反函数（补函数）的定义

设有两个 2 变量逻辑函数 L_1 和 L_2，其真值表如表 1 - 19 所示。在输入变量的每一种组合下，它们的值均相反，则称 L_1 和 L_2 互为反函数。

表 1 - 19　　L_1 和 L_2 的真值表

A	B	L_1	L_2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

2. 求一个函数反函数的方法

（1）从真值表直接写。

（2）应用德摩根定理。

（3）应用反演规则直接写。

反演规则的内容：对于任何一个逻辑函数表达式 L 进行变换，将表达式中的运算符“·”换成“＋”，“＋”换成“·”；常量“0”换成“1”，“1”换成“0”；原变量换成反变量，反变量换成原变量。那么经变换后所得到新逻辑函数表达式就是函数 L 的反函数$\overline{L}$。

在应用反演规则求反函数时要注意以下两点：一是维持原来的运算优先级不变，运算优先级要遵循“先括号，然后与运算，最后或运算”的运算次序，必要时加入括号；二是将非号下整体当做一个变量（即保留大非号不变，不属于单个变量上的非号保持不变）。利用反演规则，可以非常方便地求得一个函数的反函数。

【例 1 - 19】 求函数 $L=\overline{A}C+B\overline{D}$的反函数。

解 由反演规则可直接写出

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

【例 1 - 20】 求函数 $L=A\cdot\overline{B+C+\overline{D}}$的反函数。

解 由反演规则可直接写出

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

（三）对偶规则

1. 对偶式的定义

L 是一个逻辑表达式，对 L 进行变换，将表达式中的运算符“·”换成“+”，“+”换成“·”；常量“0”换成“1”，“1”换成“0”。经变换后所得新表达式叫做 L 的对偶式，用 L' 表示。求对偶式时要注意维持原来的运算优先级不变，运算优先级要遵循“先括号，然后与运算，最后或运算”的运算次序，必要时加入括号。与反演规则的内容相比，求对偶式时没有对变量的变换。

2. 对偶规则

若某个等式成立，则等号两边表达式的各自的对偶式也是相等的，该性质称为对偶原则。

3. 对偶规则的应用

对偶规则的意义在于利用对偶规则，可以使要证明及要记忆的公式数目减少一半。

【例 1-21】 求逻辑表达式 $L_1=A\overline{B}+C\overline{D}E$ 和 $L_2=A+\overline{B+\overline{C}+\overline{D+\overline{E}}}$ 的对偶式。

解　L_1 的对偶式为

$$L_1'=(A+\overline{B})(C+\overline{D}+E)$$

L_2 的对偶式为

$$L_2'=A\,\overline{B\overline{C}\,\overline{D\overline{E}}}$$

【例 1-22】 已知 $A+\overline{A}B=A+B$ 成立，试利用对偶规则证明 $A(\overline{A}+B)=AB$ 成立。

证明：$A+\overline{A}B$ 的对偶式为 $A(\overline{A}+B)$，$A+B$ 的对偶式为 AB，因为 $A+\overline{A}B=A+B$，所以 $A(\overline{A}+B)=AB$ 成立。

第七节　逻辑函数的五种描述方法

从前面讲过的各种逻辑关系中可以看到，如果以逻辑变量作为输入，以运算结果作为输出，那么对应于输入逻辑变量 A、B、C、…的每一组确定值，输出逻辑变量 L 就有唯一确定的值，则称 L 是 A、B、C、…的逻辑函数，记为

$$L=f(A,B,C,\cdots)$$

逻辑函数的五种描述方法分别是真值表、逻辑表达式、逻辑电路图、波形图、卡诺图。另外，还有适于计算机辅助设计的立方体表示法和二叉判决图表示法。本书对后两种表示方法不做介绍。真值表、逻辑表达式这两种表示方法在前面已经提及，本节中将再次系统地进行介绍。

下面以［例 1-18］三人表决的逻辑问题为例，说明逻辑函数的五种描述方法。在［例 1-18］中已经做了如下约定：三个人的意见为条件，分别用 A、B、C 表示，同意用 1 表示，不同意用 0 表示；结果为能否通过，用 L 表示。结果能通过用 1 表示，不通过用 0 表示。

一、真值表

将输入变量的各种可能取值组合及其对应的输出变量值列在一张表上，每一个变量均有 0、1 两种取值，n 个变量共有 2^n 种不同的取值，将这 2^n 种不同的取值组合排列起来，同时

在相应位置上填入输出对应的值，便可得到逻辑函数的真值表（Truth Table）。变量取值组合排列的顺序是任意的，但是为了不丢失某些组合，常常按所谓的“二进制递增”的规律来排列。另外，还要特别注意这些 0、1 只是表示条件具备还是不具备而已，没有数的概念。但是有时为了描述起来方便，经常用“二进制”这样的字眼。

根据“少数服从多数”的原则，将输入变量的不同取值组合与函数值间的对应关系列成表格，就建立了该逻辑函数的真值表，见表 1-17。

从文字描述首先抽象出来的是真值表，其特点是直观明了。输入变量取值一旦确定后，即可在真值表中查出相应的函数值。把一个实际的逻辑问题抽象成一个逻辑函数时，使用真值表是最方便的。所以，在设计逻辑电路时，总是先根据设计要求列出真值表。真值表的缺点是，当变量数比较多时，表比较大，显得过于繁琐。

二、逻辑表达式

在真值表 1-17 中，将函数值为 1 的变量取值组合挑出来，共有 4 个这样的组合，按 A、B、C 的顺序排列为 011、101、110、111。将 1 对应原变量，0 对应反变量，写出四个与项（借用普通代数的概念，又称为乘积项），分别是$\overline{A}BC$、$A\overline{B}C$、$AB\overline{C}$、ABC。

将这四个与项做或运算（借用普通代数的概念，又称为“加”运算），得到逻辑函数的逻辑表达式为

$$L = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \tag{1-30}$$

在详细讲述逻辑函数的逻辑表达式之前，先介绍一下最小项和最大项的概念，这两个概念非常重要。

（一）最小项和最小项表达式

1. 最小项的定义

在 n 个变量的逻辑函数中，若某个乘积项为 n 个变量的“与”，而且这 n 个变量均以原变量或反变量的形式出现一次，则称这个乘积项为该函数的一个最小项（minterm）。n 变量逻辑函数的全部最小项共有 2^n 个。例如，3 个变量 A、B、C 的逻辑函数，共有 8 个最小项，分别是 $\overline{A}\overline{B}\overline{C}$、$\overline{A}\overline{B}C$、$\overline{A}B\overline{C}$、$\overline{A}BC$、$A\overline{B}\overline{C}$、$A\overline{B}C$、$AB\overline{C}$、ABC。

2. 最小项的编号

把与某个最小项对应的那一组变量取值组合，对应关系为，原变量对应 1，反变量对应 0。把这样的一组变量取值组合人为地看作二进制数（位权任意规定），与其对应的十进制数，就是该最小项的编号。

为什么要对最小项进行编号？当自变量的个数较多时，逻辑表达式写起来会很麻烦，用最小项编号的形式则较简单，这是一种人为想出来的办法，此概念的理解非常重要。通常用符号 m_i 来表示最小项，下标 i 就是最小项的编号。

3. 最小项的基本性质

以三变量函数为例说明最小项的性质，列出三变量函数的全部最小项的真值表，如表 1-20 所示。仔细观察，从表 1-20 中可以看出最小项具有以下几个性质。

（1）每一个最小项对应了一组变量取值组合。对于任意一个最小项，只有对应的那一组取值组合使其值为 1，而其余各种变量取值组合均使它的值为 0。

（2）任意两个最小项之积恒为 0。

（3）全体最小项之和恒为 1。

表 1 - 20　　三变量全部最小项的真值表

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

4. 最小项表达式（标准与或式）

全部由最小项组成的“与或式”称为最小项表达式，也称为逻辑函数的标准与或式。任何一个逻辑函数的最小项表达式都是唯一的。用编号形式的最小项来写逻辑表达式时一定要注明最小项是如何被编号的，即在括号中注明位权任意规定的方法。

对于不是最小项表达式的与或表达式，可利用公式 $A+\overline{A}=1$ 来配项展开成最小项表达式。

【例 1 - 23】 将函数 $L=AB+\overline{A}C$ 展开成最小项表达式，并用编号形式的最小项来写逻辑表达式。

解　$L(A,B,C)=AB+\overline{A}C=AB(C+\overline{C})+\overline{A}C(B+\overline{B})$

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

$$=m_7+m_6+m_3+m_1$$

也可写作

$$L(C,B,A)=m_7+m_3+m_6+m_4$$

如果按 L(A，C，B)、L(B，C，A)、L(B，A，C)、L(C，A，B) 的方式来对最小项进行编号，又该如何呢？请读者自行思考。

如果给出了函数的真值表，则只要将函数值为 1 的那些最小项相或，便是函数的最小项表达式。对于表 1 - 17 所示的三人表决逻辑问题的真值表，其最小项表达式为

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

写成最小项编号的形式为

$$L(A,B,C)=m_3+m_5+m_6+m_7=\sum_m(3,5,6,7)$$

如果将表 1 - 17 中函数值为 0 的那些最小项相或，便可得到反函数的最小项表达式。表 1 - 17 所示的三人表决逻辑函数的反函数的最小项表达式为

$$\overline{L}=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}\tag{1 - 31}$$

（二）最大项和最大项表达式

1. 最大项的定义

在 n 个变量的逻辑函数中，若或项 M 为 n 个变量的“或”，而且这 n 个变量均以原变量或反变量的形式出现一次，则称或项 M 为该组变量的最大项。n 个变量的逻辑函数共有 2^n 个最大项。例如，3 个变量 A、B、C 的逻辑函数，共有 8 个最大项，分别是 $\overline{A}+\overline{B}+\overline{C}$、$\overline{A}+\overline{B}+C$、$\overline{A}+B+\overline{C}$、$\overline{A}+B+C$、$A+\overline{B}+\overline{C}$、$A+\overline{B}+C$、$A+B+\overline{C}$、$A+B+C$。三变量逻辑函数全部最大项的真值表如表 1 - 21 所示。

表 1-21　　三变量逻辑函数全部最大项的真值表

A	B	C	$\overline{A}+\overline{B}+\overline{C}$	$\overline{A}+\overline{B}+C$	$\overline{A}+B+\overline{C}$	$\overline{A}+B+C$	$A+\overline{B}+\overline{C}$	$A+\overline{B}+C$	$A+B+\overline{C}$	$A+B+C$
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

2. 最大项的编号

把与最大项对应的那一组变量取值组合，对应关系为，原变量对应 0，反变量对应 1，把这样的一组变量取值组合人为地看作二进制数（位权任意规定），与其对应的十进制数，就是该最大项的编号。最大项用符号 M_i 表示，下标 i 就是最大项的编号。

例如，最大项$\overline{A}+B+\overline{C}$所对应的变量取值组合为 101。位权不妨这样来规定：A 的位权为 4，B 的位权为 2，C 的位权为 1，即按 A、B、C 的顺序，A 为高位来规定位权。那么，与 101 对应的十进制数为 5，故$\overline{A}+B+\overline{C}$可以用 M_5 表示。

3. 最大项的基本性质

与最小项恰好相反，对于任何一个最大项，只有一组变量取值组合使它为 0，而变量的其余取值组合均使它为 1。

最大项的基本性质如下：

（1）n 个变量的全部最大项的“与”恒为 0，即 $\prod_{i=0}^{2^n-1} M_i = 0$；

（2）n 个变量的任意两个不同的最大项的“或”必等于 1，即 $M_i + M_j = 1(i \neq j)$。

4. 最大项表达式（标准或与式）

在一个“或与式”中，如果所有的或项均为最大项，则称这种表达式为最大项表达式，或称为标准或与式、标准和之积表达式。

如果一个逻辑函数的真值表已给出，要写出该函数的最大项表达式，把函数值为 0 对应的变量取值组合挑出来。对变量取值组合做如下的对应：0 对应原变量，1 对应反变量。写出对应的最大项，然后将这些最大项相“与”，便得到最大项表达式。

任何一个逻辑函数既可以用最小项表达式表示，也可以用最大项表达式表示。如果将一个 n 变量函数的最小项表达式改为最大项表达式时，其最大项的编号必定都不是最小项的编号，而且这些最小项的个数和最大项的个数之和为 2^n。

对于表 1-17 所示的三人表决问题的真值表，其最大项表达式为

$$\begin{aligned} L(A,B,C) &= (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C) \\ &= M_0 \cdot M_1 \cdot M_2 \cdot M_4 = \prod M(0,1,2,4) \end{aligned}$$

（三）最小项与最大项之间的关系

变量个数相同、编号方式相同、下标也相同的最小项和最大项之间存在互补关系，即 $\overline{m_i}=M_i$，$\overline{M_i}=m_i$。

例如，按（A，B，C）的顺序对最小项编号，即按 A、B、C 的顺序，A 为高位来规定

位权，则有

$$m_7 = A \cdot B \cdot C = \overline{\overline{A \cdot B \cdot C}} = \overline{\overline{A} + \overline{B} + \overline{C}} = \overline{M_7}$$

$$M_7 = \overline{A} + \overline{B} + \overline{C} = \overline{A \cdot B \cdot C} = \overline{m_7}$$

（四）两个最小项的逻辑相邻

如果两个最小项中只有一个变量不同，其余的完全相同，则称这两个最小项为逻辑相邻的最小项。

对两个逻辑相邻的最小项做“或”运算，可以消去那个不同的变量。例如，对于一个三变量的逻辑函数，$\overline{A}BC$ 和 ABC 为两个逻辑相邻的最小项，对这两个逻辑相邻的最小项做“或”运算，即$\overline{A}BC+ABC=BC$，则可以消去那个不同的变量 A。

（五）两个与项（乘积项）的逻辑相邻

如果两个与项中只有一个变量不同，而且这个不同的变量分别是原变量和其对应的反变量，其余的完全相同，则称这两个与项为逻辑相邻的与项。

对两个逻辑相邻的与项做“或”运算，可以消去那个不同的变量。例如，对于一个三变量或三变量以上的逻辑函数，$\overline{A}B$ 和 AB 为两个逻辑相邻的与项，对这两个逻辑相邻的与项做“或”运算，即$\overline{A}B+AB=B$，则可以消去那个不同的变量 A。

三、卡诺图

逻辑函数的卡诺图是美国工程师卡诺发明的一种逻辑函数的图形描述方法。研究逻辑函数的卡诺图描述方法的唯一目的就是为了化简逻辑函数。

（一）卡诺图的结构

将逻辑函数的真值表重新排列成矩阵形式，并且使矩阵的横方向和纵方向的逻辑变量的取值按照格雷码的顺序排列，这样构成的图形就是卡诺图。卡诺图实际上是真值表的一种变形，一个逻辑函数的真值表有多少行，卡诺图就有多少个小方格。n 变量函数的每个最小项各用一个小方格表示。

1. 两变量逻辑函数的卡诺图

两变量逻辑函数的卡诺图有很多种画法，不管在哪种画法中，变量的取值组合必须按格雷码的顺序列出。图 1-15 给出了两变量逻辑函数卡诺图的四种不同画法。在图 1-15（a）中，对最小项的编号采用了（A，B）的顺序，即按 A、B 的顺序，A 为高位来规定位权。在图的左上角标有本逻辑函数的自变量，逻辑函数的自变量的位置可以任意规定，但是一定要标示清楚，不要遗漏，在每个小方格中写有对应的最小项的编号。在图 1-15（b）中，画出了原变量 A 和 B 的位置，那么没有标注的位置就是反变量的位置，在每个小方格中，写有对应的最小项，读者仔细观察就可以看出其中的规律，图 1-15（b）和图 1-15（a）的变量放置的位置是一样的。图 1-15（c）给出了两变量逻辑函数的卡诺图的自变量位置的第二种标示方法，与图 1-15（a）、图 1-15（b）相比，A、B 的位置对调了，但是对最小项的编号仍然采用了（A，B）的顺序。不管在哪种画法中，变量的取值组合必须按格雷码的顺序列出。对比图 1-15（a）和图 1-15（c）可以看出，当改变了自变量位置的标示方法后，相同位置的小方格中最小项的编号有所不同。所以在以后用卡诺图来描述逻辑函数时，一定要将相关的前提条件写清楚，否则会引起误解。图 1-15（d）给出了两变量逻辑函数的卡诺图的第四种画法，两个变量 A、B 分列在两边。此处对最小项的编号采用了（B，A）的顺序，那么最小项的编号就不一样了。

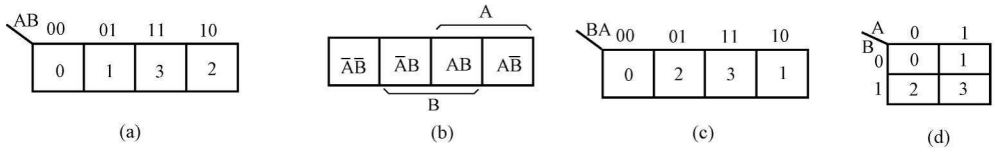


图 1-15 二变量逻辑函数的卡诺图

（a）第一种画法；（b）第二种画法；（c）第三种画法；（d）第四种画法

2. 三变量逻辑函数的卡诺图

三变量逻辑函数的卡诺图如图 1-16 所示。对最小项的编号采用了（A，B，C）的顺序。图 1-16（a）中每个小方格中写有对应的最小项的编号。图 1-16（b）中每个小方格中写有对应的最小项。图 1-16（c）给出了三变量逻辑函数的卡诺图的第三种画法，其中每个小方格中写有对应的最小项的编号。根据变量位置的标示方法的不同，还可以有其他的画法，不管哪种画法，表示的逻辑函数都是一样的。

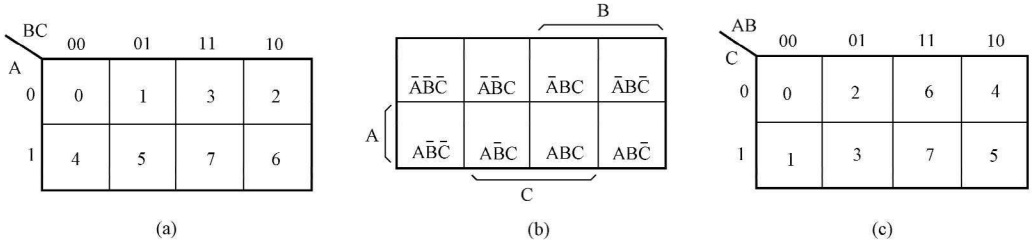


图 1-16 三变量逻辑函数的卡诺图

（a）第一种画法；（b）第二种画法；（c）第三种画法

3. 四变量逻辑函数的卡诺图

四变量逻辑函数的卡诺图如图 1-17 所示。对最小项的编号采用了（A，B，C，D）的顺序。图 1-17（a）中每个小方格中写有对应的最小项的编号。图 1-17（b）中每个小方格中写有对应的最小项。图 1-17（c）给出了四变量逻辑函数的卡诺图的第三种画法，其中每个小方格中写有对应的最小项的编号。根据变量位置的标示方法的不同，还可以有其他的画法，不管哪种画法，表示的逻辑函数都是一样的。

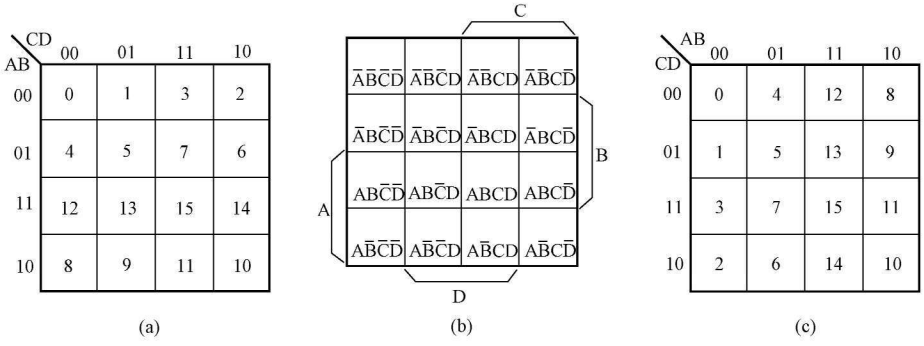


图 1-17 四变量逻辑函数的卡诺图

（a）第一种画法；（b）第二种画法；（c）第三种画法

4. 五变量逻辑函数的卡诺图

五变量逻辑函数的卡诺图如图 1-18 所示，将一个五变量的逻辑函数的卡诺图表示成A＝0 和 A＝1 的四变量逻辑函数的卡诺图，然后再将这两个平面图形合成一个立体图形。在图 1-18 中是以 A、B、C、D、E 的顺序，A 为高位对最小项进行编号的，每个最小项的编号写进了对应的小方格中。注意在卡诺图中 A、B、C、D、E 放置的位置可以是任意的，但是必须保证按格雷码的顺序来给出变量的取值组合。

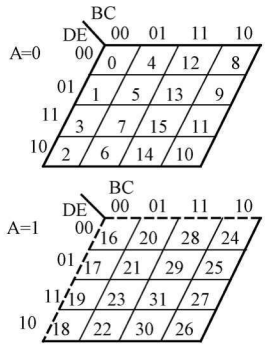


图 1-18　五变量逻辑函数的卡诺图

5. 六变量逻辑函数的卡诺图

六变量逻辑函数的卡诺图是一个立体图形，如图 1-19 所示，将一个六变量的逻辑函数的卡诺图表示成 AB＝00、AB＝01、AB＝11、AB＝10 的四变量逻辑函数的卡诺图。在图 1-19 中以 A、B、C、D、E、F 的顺序，A 为高位对最小项进行编号。每个最小项的编号写进了对应的小方格中。注意在卡诺图中 A、B、C、D、E、F 放置的位置可以是任意的，但是必须保证按格雷码的顺序来给出变量的取值组合。

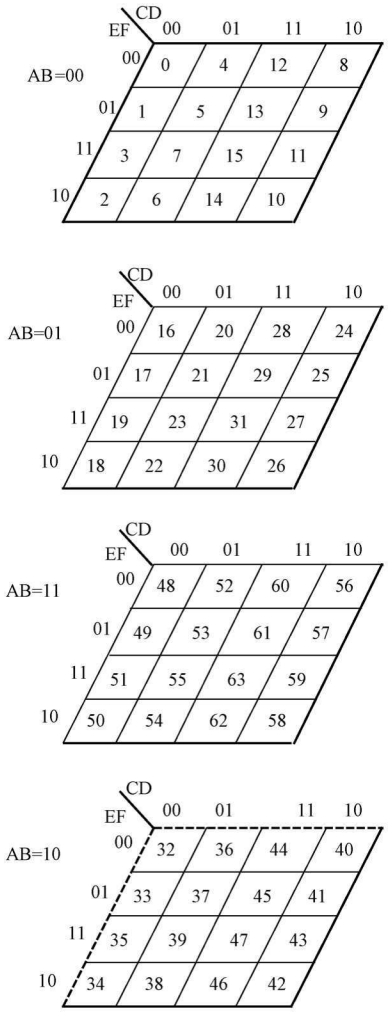


图 1-19　六变量逻辑函数的卡诺图

卡诺图的缺点是随着输入变量个数的增加图形变得越来越复杂，几何相邻项的观察也变得不那么直观。因此它只适于用来表示 6 个及 6 个以下变量的逻辑函数。

（二）逻辑函数在卡诺图中的表示

如果逻辑函数是以真值表或者以最小项表达式给出，在卡诺图上将给定的最小项相对应的方格内填入 1，其余的方格内填入 0。

如果逻辑函数以一般的逻辑表达式给出，需先将函数变换为最小项表达式，再填入卡诺图。也可以不必变换为最小项之和的形式，只要化成一般与或式，然后在卡诺图上找出每一个乘积项所包含的那些最小项（该乘积项就是这些最小项的公因子）对应的方格，在这几个方格内均填入 1，其余的方格内均填入 0，就完成了函数的卡诺图的建立。［例 1-18］三人表决问题逻辑函数的卡诺图如图1-20 所示。

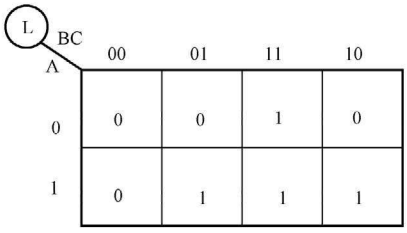


图 1-20　［例 1-18］的卡诺图

四、逻辑电路图

将逻辑表达式中的各变量之间的逻辑运算关系用逻辑符号表示出来，则可以得到逻辑电路图。[例 1-18] 三人表决逻辑问题的最小项表达式为 $L=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$，对应的逻辑电路图如图 1-21 所示。

五、时序图

时序图又称波形图，是反映输入信号和输出信号波形变化规律的图形，是在同一时间坐标前提下，由输入变量的所有可能取值组合的高、低电平及其对应的输出函数值的高、低电平所构成的图形。在画波形图时，通常将横坐标轴和纵坐标轴省略不画，横坐标轴表示时间，纵坐标轴表示电压的大小。

图 1-22 是 [例 1-18] 三人表决逻辑问题在给定 A、B、C 波形后，逻辑结果 L 的波形图。图中，高电平用 1 表示，低电平用 0 表示，从左到右依次给出了 ABC 在 000、001、010、011、100、101、110、111 输入情况下的输出值，输出值分别为 0、0、0、1、0、1、1、1，与真值表 1-17 完全相同。

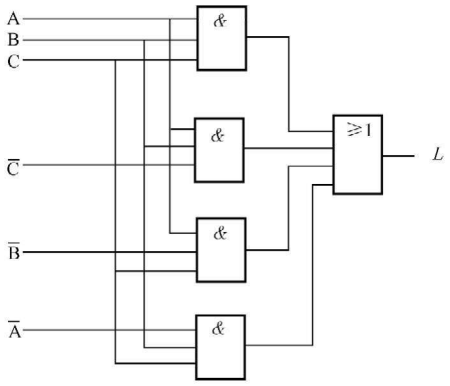


图 1-21 用与门和或门实现的三人表决问题的逻辑电路图

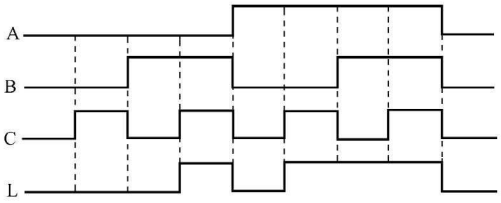


图 1-22 三人表决逻辑电路的波形图

注意，在波形图中一定要将变量的所有取值组合全部给出，这样才能完整地描述逻辑函数。

波形图能直接反映出变量之间的时间关系和函数关系，所以常在数字电路分析和调试时用到。

第八节 逻辑函数的化简

从前面的讲解可以看出，同一个逻辑函数可以有繁简程度相差甚远的不同形式，实现它的电路也不相同。逻辑式越简单，电路使用的器件就越少，器件之间的连线就越少，电路的工作就越可靠。本节主要介绍将逻辑函数化简成最简的与或表达式的方法，即用最简的二级与或电路来实现逻辑函数。本节将学习逻辑函数的两种化简方法：代数化简法和卡诺图化简法。代数化简法和卡诺图化简法适于用手工实现。除此之外，列表化简法是一种系统化的方法，有比较固定的规律可循，适于用计算机编程来实现逻辑函数的化简。组合逻辑函数还有立方体表示法和二叉判决图表示法，这两种表示方法也非常适于用计算机编程来实现逻辑函

数的化简，在目前流行的电子设计自动化软件中有非常广泛的应用。不管是手工方法化简还是计算机编程化简，它们的理论基础仍然是逻辑代数的基本定理。

一、逻辑函数最简的标准

本书采用的逻辑函数最简的标准是针对二级与或电路而言的。化简的目的就是为了得到最简与或表达式。逻辑函数的最简与或表达式的标准是：首先乘积项的个数最少（与门的个数最少，即或门的输入端数最少）；然后，每个乘积项中的变量数目最少（与门的输入端个数最少）。特别提醒读者注意的是，该标准要首先满足前者，在满足前者的前提下，再满足后者。

在电子系统的设计过程中，只有最简与或表达式可能并不能满足要求，需要将最简与或表达式变换成其他形式的最简表达式。在得到逻辑函数的最简与或表达式之后，经过变换很容易得到其他形式的最简表达式，比如最简或与表达式、最简与非—与非表达式、最简或非—或非表达式、最简与—或—非表达式。

如式（1-32）中，从上往下分别是逻辑函数 L 的与或表达式、或与表达式、与非—与非表达式、或非—或非表达式、与—或—非表达式。

$$
\begin{aligned}
L &= AB + \overline{A}C \\
&= (A + C)(\overline{A} + B) \\
&= \overline{\overline{AB}\,\overline{\overline{A}C}} \\
&= \overline{\overline{A + C} + \overline{\overline{A} + B}} \\
&= \overline{\overline{A}\,\overline{C} + A\,\overline{B}}
\end{aligned}
\tag{1-32}
$$

以上五个表达式是同一函数的不同形式，其变换方法如下：

（1）通过化简得到函数的最简与或表达式；

（2）通过化简得到函数的反函数的最简与或表达式（可采用后面介绍的卡诺图圈 0 的方法），再求反，就得到最简或与表达式；

（3）在最简与或表达式的基础上，应用两次德摩根定理，就可以得到最简与非—与非表达式；

（4）在最简或与表达式的基础上，应用两次德摩根定理，就可以得到最简或非—或非表达式；

（5）对函数的反函数的最简与或表达式直接加上非号就得到最简与—或—非表达式；

（6）在最简与—或—非表达式的基础上，只对其中的与项两次求反，就可以得到最简或非—或非表达式。

二、代数化简法

逻辑函数的代数化简法就是利用逻辑代数的基本定律，消去多余的乘积项和每个乘积项中的多余变量，得出最简与或表达式。在应用德摩根定理去掉函数中的大非号，得到表达式的一般与或式后，关键问题是观察逻辑表达式中逻辑相邻的最小项或乘积项（与项），然后将这两个逻辑相邻的最小项或乘积项结合在一起做或运算，可以消去一个变量。当逻辑表达式比较复杂时，可能有很多逻辑相邻的最小项或乘积项（与项），这时不太容易确定哪些乘积项结合在一起更合适，从而可以消去更多的变量。

【例 1-24】 化简逻辑函数 $L = ABC + \overline{A}BC + B\overline{C}$。

解 由并项法，可得

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

【例 1-25】 化简逻辑函数 $L = AB + AB\overline{C} + ABD + AB(\overline{C} + \overline{D})$。

解 利用吸收法，可得

$$L = AB + AB\overline{C} + ABD + AB(\overline{C} + \overline{D})$$
$$= AB + AB\overline{C} + ABD + AB\overline{C} + AB\overline{D}$$
$$= AB(1 + \overline{C} + D + \overline{C} + \overline{D})$$
$$= AB$$

【例 1-26】 化简下列逻辑函数：

(1) $L_1 = A\overline{B} + AC + ADE + \overline{C}D$；

(2) $L_2 = AB + \overline{B}C + AC(DE + FG)$。

解 (1) $L_1 = A\overline{B} + AC + ADE + \overline{C}D$

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

(2) $L_2 = AB + \overline{B}C + AC(DE + FG)$

$$= AB + \overline{B}C$$

可见，在本例中利用了冗余定理来进行化简。

【例 1-27】 化简逻辑函数 $L = \overline{B} + ABC$。

解 $L = \overline{B} + ABC = (\overline{B} + B)(\overline{B} + AC) = \overline{B} + AC$

【例 1-28】 用代数方法对［例 1-18］的逻辑函数进行化简，并画出化简后的逻辑电路图。

解 从［例 1-18］的真值表直接写出函数最小项表达式为

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

用代数法化简该逻辑函数，根据幂等率 $A = A + A + A$，有 $ABC = ABC + ABC + ABC$，然后找逻辑相邻的最小项，合并消去一个变量，得到最简与或表达式。化简过程为

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

由函数的最简与或表达式画出逻辑电路图，如图 1-23 所示。

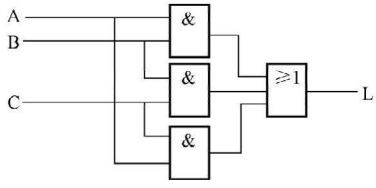


图 1-23 ［例 1-28］的逻辑电路图

对于初学者，有时很难想到上述方法，特别是当表达式很复杂时，就更难把握，需要经验和技巧。在［例 1-29］中用卡诺图法对［例 1-18］的逻辑函数进行了化简，读者可以做个对比。

三、卡诺图化简法

代数化简法没有固定的步骤可循，需要熟练运用

各种公式和定理，在化简一些较为复杂的逻辑函数时还需要一定的技巧和经验，有时很难判定化简结果是否最简。为了解决上述问题，美国工程师卡诺提出了逻辑函数的图形化简法，从而解决了代数化简法的不足。逻辑函数的图形化简法是将逻辑函数用卡诺图来表示，利用卡诺图来化简逻辑函数。该法直观、易掌握。

逻辑函数的与或表达式有三种：标准与或式、一般与或式、最简与或式。卡诺图化简得到逻辑函数的最简与或式。

（一）卡诺图的几何位置相邻

在卡诺图中，观察任意两个表示最小项的方格，如果有相接（紧挨着），或相对（任意一行或一列的两头），或相重（对折起来重合），则称这两个最小项为几何位置相邻。

研究卡诺图的几何位置相邻的目的是：卡诺图中几何位置相邻的最小项在逻辑上也具有相邻性。逻辑相邻不容易观察，尤其是在较复杂的逻辑表达式中。但是卡诺图的几何位置相邻特别容易观察。

（二）卡诺图化简逻辑函数的依据

根据前面的知识已经知道，在代数化简法中，对两个逻辑相邻的最小项做或运算，可以消去那个不同的变量。但是在较复杂的逻辑表达式中，不容易观察出哪些是逻辑相邻的最小项，而卡诺图的几何位置相邻就特别容易观察。卡诺图中几何位置相邻的最小项在逻辑上也具有相邻性，这就是卡诺图化简逻辑函数的依据。卡诺图化简逻辑函数的数学基础仍然是逻辑代数的基本定理。卡诺图是逻辑代数的基本定理的图形表示方法，读者在学习卡诺图时不要死记硬背卡诺图化简的规则，要将之与代数化简法对比，这样对卡诺图才可以有一个更好的理解。

【例 1-29】 用卡诺图方法对［例 1-18］的三个人表决逻辑函数进行化简。

解　画出［例 1-18］的卡诺图如图 1-24 所示，观察得知，最小项 ABC 分别和最小项 $\overline{A}BC$、$A\overline{B}C$、$AB\overline{C}$是几何相邻的。所以，最小项 ABC 分别和最小项$\overline{A}BC$、$A\overline{B}C$、$AB\overline{C}$是逻辑相邻的。将几何相邻的那些最小项用圆圈圈在一起，表示它们可以进行或运算并可以消去相关变量。与［例 1-28］的代数方法化简过程做个对比，最小项 ABC 在画圈过程中重复使用相当于将 ABC 展开成 ABC＋ABC＋ABC。可以看出，卡诺图法更易于掌握。在图 1-24 中，画出了三个圈，将这三个圈中的最小项分别做或运算，得到三个与项，再将这三个与项相或，就得到最简的与或表达式 L＝BC＋AC＋AB。

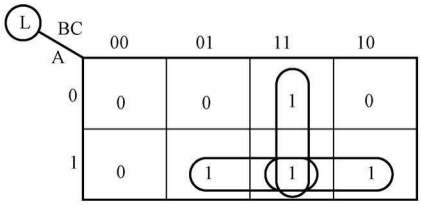


图 1-24　［例 1-29］的卡诺图

与［例 1-28］中用代数方法对［例 1-18］的逻辑函数的化简做对比，可以看出，在代数方法中很难想到如何去配项、合并，在卡诺图中可以非常直观地观察出来。特别提醒读者的注意的是，从［例 1-28］来体会［例 1-29］中卡诺图化简的数学基础是逻辑代数，认真体会卡诺图化简的依据。

（三）用卡诺图化简逻辑函数的几个关键点

(1) 圈的总个数尽量少。在圈的总个数最少的前提下，再保证每个圈是最大的圈。这与前面定义的逻辑函数的最简与或表达式的标准相一致，即乘积项的个数最少（圈的总数最少），每个乘积项中的变量数目最少（圈最大）。

（2）尽量画大圈，要特别注意对边相邻性和四角相邻性。

（3）每个圈内只能含有 2^n（$n=0$，1，2，3，4，5，6）个最小项，即只能将 1 个、或者 2 个、或者 4 个、或者 8 个、或者 16 个、或者 32 个、或者 64 个最小项圈入一个圈中。3 个、5 个、6 个……总数不是 2^n 个的最小项不能圈入一个圈中。

（4）依据幂等率 A＋A＋A＝A，方格中的 1 可以被重复使用，即最小项可以被重复使用。但是，在简化过程中，尽量减小重复使用的次数，这样可以减少圈的总个数。

（5）卡诺图中所有取值为 1 的方格均要被圈过，即不能漏下任何一个取值为 1 的最小项。

（6）最后一定要检查是否有多余的圈。如果某个圈中没有一个自己独有的 1，则这个圈就是多余的。

2 个相邻的最小项结合，可以消去 1 个取值不同的变量而合并为 1 项；4 个相邻的最小项结合，可以消去 2 个取值不同的变量而合并为 1 项；8 个相邻的最小项结合，可以消去 3 个取值不同的变量而合并为 1 项。总之，2^n 个相邻的最小项结合，可以消去 n 个取值不同的变量而合并为 1 项。

【例 1-30】 使用卡诺图法化简逻辑函数 $L=A\overline{C}+\overline{A}C+B\overline{C}+\overline{B}C$ 为最简与或式。

解 （1）函数以一般与或式的形式给出，可以直接填入卡诺图。画出函数 L 的卡诺图如图 1-25（a）所示。

（2）找出能够合并的最小项，进行合并。有两种画圈的方法，如图 1-25（b）、（c）所示。

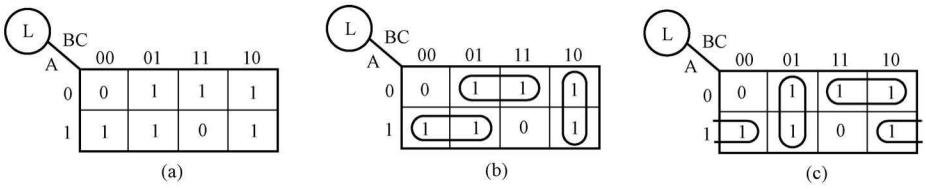


图 1-25 ［例 1-30］的卡诺图化简

（a）［例 1-30］的卡诺图；（b）第一种圈法；（c）第二种圈法

（3）写出最简的逻辑函数表达式。

由图 1-25（b）可得

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

由图 1-25（c）可得

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

由此可见，有时同一个逻辑函数化简得到的结果并不是唯一的。

【例 1-31】 用卡诺图化简逻辑函数 $L(A, B, C, D)=\sum_m(0, 4, 5, 8, 9, 11, 13, 15)$。

解 函数以最小项表达式的形式给出，可以直接填入卡诺图。

读者可能会想到两种圈法，分别如图 1-26（a）、（b）所示。图 1-26（a）化简结果是 $L=\overline{A}\,\overline{C}\,\overline{D}+B\overline{C}D+A\overline{B}\,\overline{C}+AD$，共有四个与项。而图 1-26（b）化简结果是 $L=\overline{B}\,\overline{C}\,\overline{D}+\overline{A}B\overline{C}+AD$，共有三个与项。在本教材中，定义逻辑函数的最简与或表达式的标准为：首先乘

积项的个数最少（与门的个数最少，即或门的输入端数最少），也就是圈的总个数最少。然后才是每个乘积项中的变量数目最少（与门的输入端数最少），也就是圈最大。所以，比较而言，图 1-26（b）的圈法是最简的，而图 1-26（a）还不是最简的。通过［例 1-31］，可以总结出这样一个规律，在有些情况下，最小项的圈法不止一种，得到的各个乘积项组成的与或表达式各不相同，哪个是最简的，还要经过比较、检查才能确定。

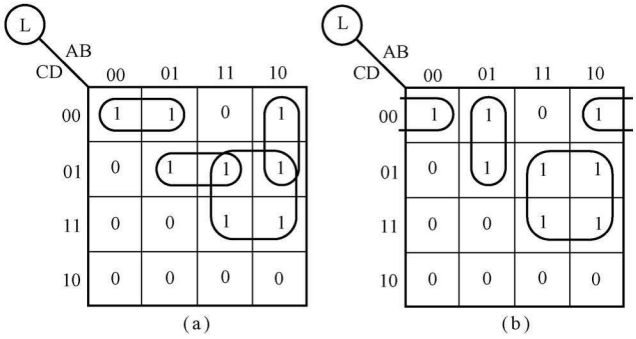


图 1-26　［例 1-31］的卡诺图

（a）第一种圈法；（b）第二种圈法

【例 1-32】 用卡诺图化简逻辑函数 $L(A, B, C, D)=\sum_m(0, 1, 2, 5, 6, 7, 8, 9, 13, 14)$。

解　函数以最小项表达式的形式给出，可以直接填入卡诺图。可以有两种圈法，分别如图 1-27（a）和图 1-27（b）所示。

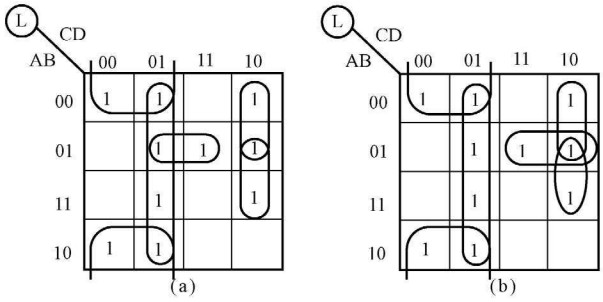


图 1-27　［例 1-32］的卡诺图

（a）第一种圈法；（b）第二种圈法

由图 1-27（a）写出表达式

$$L=\overline{C}D+\overline{B}\,\overline{C}+\overline{A}BD+\overline{A}C\,\overline{D}+BC\,\overline{D}$$

由图 1-27（b）写出表达式

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

通过这个例子也可以看出，一个逻辑函数的真值表是唯一的，卡诺图也是唯一的，但化简结果有时不是唯一的。

在前面的几个例题中，采取的是对 1 进行处理的办法，直接得到的是原函数的最简与或表达式。除了上述方法以外，对于卡诺图中 0 比较少的逻辑函数，还可以采用另外一种方

法，即“圈 0 法”，得到的是反函数的最简与或表达式，然后再求取反函数，就可得出原函数的最简与或表达式。

【例 1-33】 已知逻辑函数，分别用“圈 1 法”和“圈 0 法”得出其最简与或式。

解 (1) 如图 1-28 (a) 所示，用“圈 1 法”画包围圈，得

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

(2) 如图 1-28 (b) 所示，用“圈 0 法”画包围圈，得

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

对 $\overline{L}$ 取非得

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

【例 1-34】 用卡诺图化简逻辑函数

$$L(A,B,C,D,E) = \sum_m(2,5,7,8,10,13,15,17,19,21,23,24,29\ 31)$$

解 画出逻辑函数的卡诺图如图 1-29 所示。图中，分别画出了 A=0、A=1 两个平面图形，这两个平面图形组合在一起形成了一个立体图形。每个平面图形是一个四个变量的卡诺图，其最小项的几何相邻的关系如前所述。为了清晰起见，方格中的 0 省略而没有标出，只标出了 1，即没有任何标注的方格中为 0。要特别注意的是，在这两个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。根据图 1-29 可得到函数最简与或表达式为

$$L = CE + A\overline{B}E + B\overline{C}\,\overline{D}\,\overline{E} + \overline{A}\,\overline{C}D\,\overline{E}$$

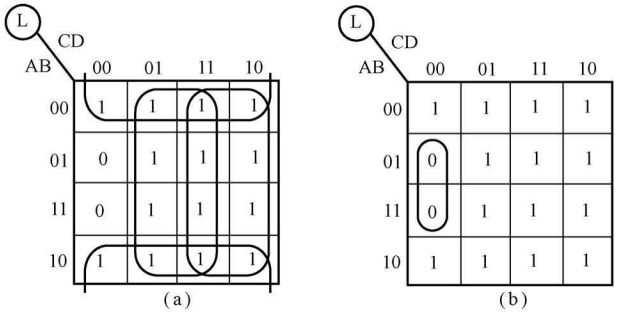


图 1-28 [例 1-33] 的卡诺图

(a) 圈 1 法；(b) 圈 0 法

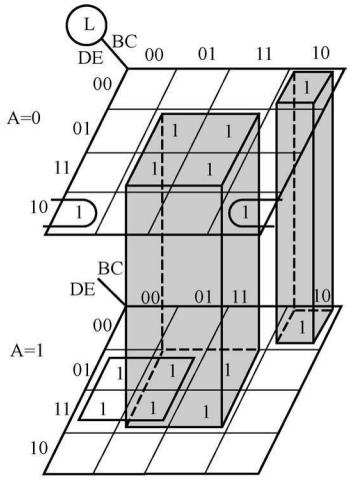


图 1-29 [例 1-34] 的卡诺图

【例 1-35】 用卡诺图化简逻辑函数

$$L(A,B,C,D,E,F) = \sum_m(2,8,10,18,24,26,34,37,42,45,50,53,58,61)$$

解 画出逻辑函数的卡诺图如图 1-30 所示。在图 1-30 中，分别画出了 AB=00、AB=01、AB=11、AB=10 四个平面图形，这四个平面图形组合在一起形成了一个立体图形。每个平面图形是一个四个变量的卡诺图，其最小项的几何相邻的关系如前所述。为了清晰起见，方格中的 0 省略而没有标出，只标出了 1。

要特别注意的是，在这四个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。在AB=00、AB=01两个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。在AB=00、AB=10两个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。在AB=01、AB=11两个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。在AB=11、AB=10两个平面中在对应的位置如果都有相同的最小项，这些最小项也是几何相邻的。

由图1-30可得到函数的最简与或表达式为

$$L = \overline{D}E\overline{F} + AD\overline{E}F + \overline{A}C\overline{D}\overline{F}$$

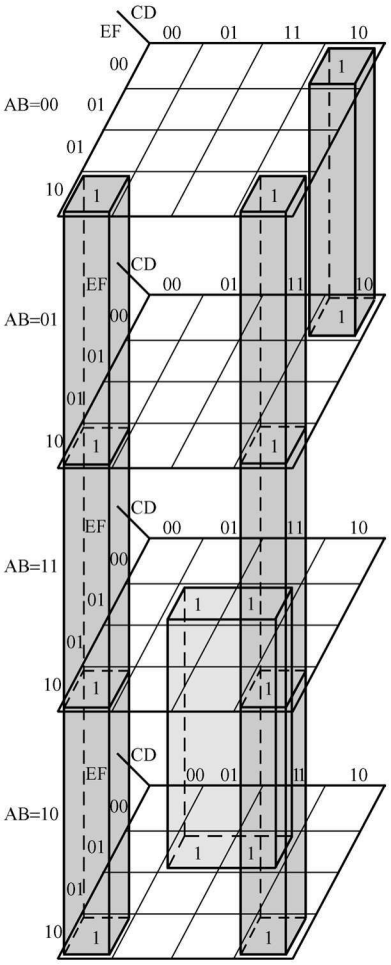


图1-30　[例1-35]的卡诺图

（四）逻辑问题中的无关项及其在逻辑函数化简中的应用

1. 逻辑问题中的无关项

在有些逻辑问题中，输入变量的某些取值组合不会出现，或者一旦出现，逻辑值也可以是任意的，对后续控制过程没有意义。不会出现的变量取值组合称为约束项，可以出现，但对后续控制过程没有意义的变量取值组合称为任意项。约束项、任意项统称为无关项（don't cares）。

研究一个逻辑问题的无关项的唯一目的就是为了化简这个逻辑问题对应的逻辑函数，使实现这个逻辑函数的电路更简单。如果不在乎电路复杂与否，完全可以不研究这个逻辑函数是否有无关项。

【例1-36】　在十字路口有红绿黄三色交通信号灯，规定红灯亮表示禁止通行，绿灯亮表示放行，黄灯亮表示等一等，试分析车辆行驶状态与三色信号灯之间逻辑关系。

解　(1) 约定红、绿、黄灯分别用A、B、C表示，且灯亮为1，灯灭为0；车用L表示，车行L=1，车停L=0。

(2) 列出该逻辑问题的真值表。在这个函数中，根据题目的要求，在系统正常工作的情况下，可以得知有5种变量取值组合不会出现，一是红灯、绿灯和黄灯都灭的情况；二是红灯灭、绿灯和黄灯都亮的情况；三是红灯和黄灯都亮、绿灯灭的情况；四是红灯和绿灯都亮、黄灯灭的情况；五是红灯、绿灯和黄灯都亮的情况。所以该逻辑函数共有5个最小项为无关项。车辆行驶状态与信号的真值表如表1-22所示。

表1-22　　[例1-36]的真值表

A	B	C	L	A	B	C	L
0	0	0	×	1	0	0	0
0	0	1	0	1	0	1	×
0	1	0	1	1	1	0	×
0	1	1	×	1	1	1	×

2. 带有无关项的逻辑函数的化简

化简具有无关项的逻辑函数时，要充分利用无关项对应的逻辑函数值既可以当0也可以当1的特点，尽量扩大卡诺圈，使逻辑函数更简。为什么无关项既可以当0，也可以当1？因为无关项对应的变量取值组合不会出现，或者一旦出现，逻辑值也可以是任意的。所以当作什么看待都没关系。利用了无关项化简得到的电路工作时，一定要遵循无关项的约定，否则电路工作会出现问题。带有无关项的逻辑函数的最小项表达式为 $L(A, B, C)=\sum_m()+\sum_d()$，其中 $\sum_m()$ 是函数值为1的最小项，括号中是最小项的编号；$\sum_d(\ )$ 是无关项，括号中是作为无关项的最小项的编号，d是 don't care 的第一个字母。另外，无关项还可以用一个等于0的表达式来表示，这个表达式所包含的全部最小项都是无关项。用一个等于0的表达式，而不是用一个等于1的表达式来表示无关项的原因是，原表达式与一个等于0的表达式做或运算对结果没有任何影响。

在卡诺图中，无关项的位置既可以写入1，也可以写入0，因此，我们统一用符号“×”表示无关项。无关项可以帮助1去圈大圈，却不一定都圈到圈里去，将有用的“×”圈进去即可，但是1必须圈完，否则函数会发生错误。

在逻辑函数的化简中，充分利用无关项可以当作0也可以当作1的特点，尽量扩大卡诺圈，使逻辑函数更简，可以得到更加简单的逻辑表达式，其相应的逻辑电路也更简单。在化简过程中，无关项的取值可视具体情况取0或取1。具体地讲，如果无关项对化简有利，则取1；如果无关项对化简不利，则取0。在考虑无关项时，哪些无关项当作1，哪些无关项当作0，要以尽量扩大卡诺圈、减少圈的个数，使逻辑函数更简为原则。

【例1-37】 按要求化简逻辑函数

$$L(A,B,C,D)=\sum_m(0,2,4,6,8)+\sum_d(10,11,12,13,14,15)$$

要求：(1) 不利用无关项化简；

(2) 利用无关项化简。

解 (1) 图1-31是逻辑函数不利用无关项化简的卡诺图，只圈1，化简结果为 $L=\overline{A}\,\overline{D}+\overline{B}\,\overline{C}\,\overline{D}$。

(2) 图1-32是逻辑函数利用无关项化简的卡诺图，利用无关项后，圈可以画得更大些，使逻辑函数得到进一步的化简，结果为 $L=\overline{D}$。注意，在此题中，有3个“×”没有圈进去。

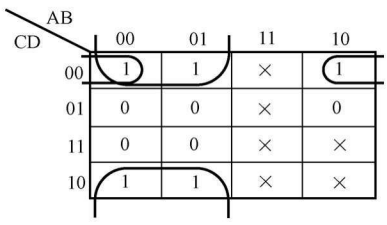


图1-31 不利用无关项化简

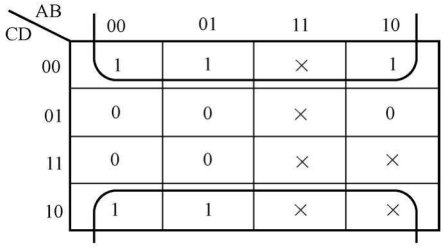


图1-32 利用无关项化简

利用无关项化简后的函数的表达式（表达式中要包括对无关项的描述）为

$$\begin{cases} L(A,B,C,D) = \overline{D} \\ \sum_d(10,11,12,13,14,15) = 0 \end{cases}$$

【例 1-38】 化简具有无关项的逻辑函数 $L(A, B, C, D)=\sum_m(2, 4, 6, 8)$，约束条件为 $AB+AC=0$。

解 约束条件 $AB+AC=0$ 的含义是，在（A，B，C，D）的编号方式下 m_{10}、m_{11}、m_{12}、m_{13}、m_{14}、m_{15}为无关项。在卡诺图中用×表示。画出带无关项的卡诺图，如图 1-33 所示。

将对化简有利的无关项看作 1，对化简无利的无关项视为 0。化简的结果为 $L(A,B,C,D) = B\overline{D} + A\overline{D} + C\overline{D}$

【例 1-39】 用门电路来实现［例 1-36］要求的逻辑关系，要求利用无关项来帮助化简。

解 利用无关项化简的逻辑函数的卡诺图如图 1-34 所示。

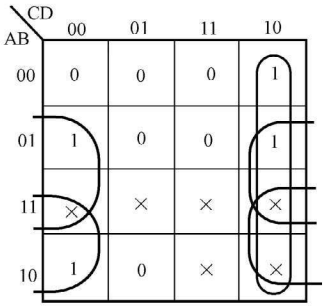


图 1-33 ［例 1-38］的卡诺图

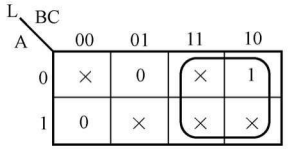


图 1-34 ［例 1-39］的卡诺图

根据图 1-34 的卡诺图写出该逻辑关系的表达式为 L=B。因为利用了无关项来进行化简，所以要遵守无关项的约定，即在人们遵守交通规则的前提下，只要看到绿灯亮了，就可以通行。图 1-35 是实现该逻辑功能的逻辑电路图，很简单，只有一根导线。

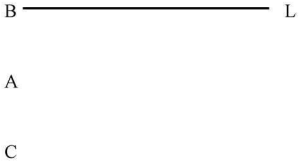


图 1-35 ［例 1-39］的逻辑电路图

小　　结

在时间上和数值上均是连续的信号称为模拟信号，在时间上和数值上均是离散的信号称为数字信号，数字信号有许多优点。数字电路处理的信号是数字信号。

常用的数制有二进制、八进制、十进制、十六进制，数字电路中使用的数制是二进制。码制就是编码的方法。

现实世界中存在的事物之间有很复杂的逻辑关系，但是无论多复杂的逻辑关系都是由最基本的三种逻辑关系组成。这三种最基本的逻辑关系就是与逻辑、或逻辑和非逻辑。

逻辑代数是用于逻辑分析的数学工具。逻辑代数在逻辑学的基础上发展的一门学科，它

采用一套符号来描述逻辑思维，并将复杂的逻辑问题抽象为一种简单的符号演算，摆脱了冗繁的文字描述。“一套符号”指的是用字母表示条件、结果，称作逻辑变量（自变量、因变量），其取值只有两种可能，即条件具备还是不具备，结果发生还是不发生，并用符号 0、1 表示。0、1 本身没有数量的概念，只是表示条件具备还是不具备，结果发生还是不发生，两者的对应关系可以任意指定。

逻辑函数的五种描述方法分别是真值表、逻辑表达式、逻辑电路图、波形图、卡诺图；另外还有适于计算机辅助设计的、立方体表示法和二叉判决图表示法，本书不做介绍。

逻辑函数的最简与或表达式的标准是：首先乘积项的个数最少（与门的个数最少，即或门的输入端数最少），然后，每个乘积项中的变量数目最少（与门的输入端数最少）。特别提醒读者注意的是，要首先满足前者，在满足前者的前提下，再满足后者。

逻辑函数的代数化简法就是利用逻辑代数的基本定律，消去多余的乘积项和每个乘积项中的多余变量，得出最简的与或表达式。代数法化简没有固定的步骤。

卡诺图中几何位置相邻的最小项在逻辑上也具有相邻性，这就是卡诺图化简逻辑函数的依据。

在有些逻辑函数中，输入变量的某些取值组合不会出现，或者一旦出现，逻辑值也可以是任意的。这样的取值组合所对应的最小项分别称为任意项、约束项，统称为无关项。我们研究一个逻辑函数的无关项的唯一目的就是为了化简这个函数，使实现这个逻辑函数的电路更简单。如果不怕电路复杂、完全可以不研究这个函数是否有无关项。化简具有无关项的逻辑函数时，要充分利用无关项对应的逻辑函数值既可以当 0 也可以当 1 的特点，尽量扩大卡诺圈，使逻辑函数更简。

习　　题

1-1　写出下列位置计数法表示的各数的多项式计数法形式：

(1) $(3026)_D$；(2) $(826)_D$；(3) $(101000)_B$；

(4) $(11110)_B$；(5) $(256)_O$；(6) $(4B6)_H$；

(7) $(256.076)_O$；(8) $(4B6.AEF)_H$。

1-2　将下列二进制数转换为十六进制数和十进制数：

(1) $(10010111)_B$；(2) $(1101101)_B$；

(3) $(0.01011111)_B$；(4) $(11.001)_B$。

1-3　将下列数转换为十进制数：

(1) $(1101110)_B$；(2) $(36.7)_O$；(3) $(4E6.2)_H$。

1-4　将下列十进制数转换为 8421BCD 码、5421BCD 码和余三码：

(1) $(48)_D$；(2) $(34.15)_D$；(3) $(121.08)_D$。

1-5　将下列十进制数转换成二进制数，要求误差不大于 2^{-8}。

(1) $(748)_D$；(2) $(75.75)_D$；(3) $(108.85)_D$。

1-6　将下列十进制数转换为十六进制数，要求误差不大于 2^{-9}。

(1) $(748)_D$；(2) $(75.75)_D$；(3) $(108.85)_D$。

1-7　写出下列 BCD 码对应的十进制数：

(1) $(10000111011001010100)_{8421BCD}$；(2) $(10010011.10000011)_{8421BCD}$；

(3) $(100001110101.10010011)_{余3码}$；(4) $(00101001.0100)_{8421BCD}$。

1-8　用真值表证明下列逻辑等式：

(1) $A+BC=(A+B)(A+C)$；

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

1-9　证明下列逻辑等式成立：

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

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

(3) $\overline{\overline{A+B+\overline{C}}\,\overline{CD}}+(B+\overline{C})(A\overline{B}D+\overline{B}\,\overline{C})=1$

(4) $\overline{A}(C\oplus D)+B\overline{C}D+AC\overline{D}+A\overline{B}\overline{C}D=C\oplus D$

1-10　用德摩根定理直接写出下列函数的反函数：

(1) $L_1=A+\overline{BC+\overline{D+E}}$；

(2) $L_2=A+\overline{B}(AD+\overline{E})$。

1-11　求下列表达式的对偶式：

(1) $L_1=A(B+C+\overline{D})+A\overline{C}$；

(2) $L_2=\overline{A\overline{B}+CD}(B+\overline{C})+AB\overline{C}$。

1-12　用代数化简法化简下列逻辑函数为最简与或式：

(1) $L=A\overline{B}+A\overline{C}+B\overline{C}+A\overline{B}\overline{C}+AB\overline{C}D$；

(2) $L=B+\overline{A}B+A\overline{B}$；

(3) $L=AB+\overline{A}C+ABD+BCD$；

(4) $L=AC(\overline{C}D+\overline{A}B)+BC(\overline{\overline{B}+\overline{AD}+CE})$；

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

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

(7) $L=\overline{\overline{\overline{A}+B}+\overline{A+\overline{B}}}+\overline{\overline{\overline{AB}}\,\overline{A\overline{B}}}$。

1-13　用代数化简法将下列逻辑函数化简成为最简与或式：

(1) $L_1=AC+\overline{B}C+B\overline{D}+A(B+\overline{C})+\overline{A}BC\overline{D}+A\overline{B}DE$；

(2) $L_2=(A+B+\overline{C})(A+B+C)$；

(3) $L_3=A\overline{B}+\overline{A}C+\overline{A}B+B\overline{C}$；

(4) $L_4=ABC+\overline{A}BC+\overline{A}B\overline{C}+\overline{ABC}\cdot\overline{AB}$；

(5) $L_5=(\overline{A\overline{B}+\overline{A}B}C+A\overline{B}C)(AD+BC)$；

(6) $L_6=\overline{\overline{A\overline{B}+ABC}+A(B+A\overline{B})}$；

(7) $L_7=A\overline{B}(C+D)+B\overline{C}+\overline{A}\,\overline{B}+\overline{A}C+BC+\overline{B}\,\overline{C}D$；

(8) $L_8=A\overline{B}+AC+BCD+\overline{D}$；

(9) $L_9=\overline{A}+\overline{A}\,\overline{B}+BC\overline{D}+B\overline{D}$；

(10) $L_{10}=A\overline{B}C+(\overline{B}+\overline{C})(\overline{B}+\overline{D})+(\overline{A+C+D})$；

(11) $L_{11}=A+\overline{A}B+(\overline{A+B})C+(\overline{A+B+C})D$。

1-14　用与非门为组件设计一多数表决电路。要求 A、B、C 三人中只要有半数以上人

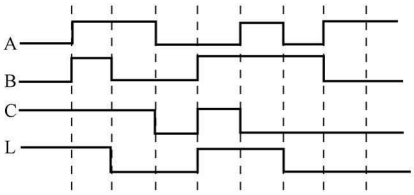


图 1-36 题 1-15 图

同意，则决议就能通过；A 还具有否决权，即只要 A 不同意，既使多数人同意也不能通过。

1-15 根据图 1-36 所示的波形图，要求：

(1) 写出逻辑关系表达式 $L=f(A, B, C)$；

(2) 把上述表达式简化成最简与或非表达式；

(3) 把上述表达式简化成最简或非—或非表达式。

1-16 用卡诺图化简法化简下列逻辑函数：

(1) $L=ABC+ABD+\overline{C}\overline{D}+A\overline{B}C+\overline{A}C\overline{D}+A\overline{C}D$；

(2) $L=A\overline{B}+\overline{A}C+BC+\overline{C}D$；

(3) $L=\overline{A}\overline{B}+B\overline{C}+\overline{A}+\overline{B}+ABC$；

(4) $L=\overline{A}\overline{B}+AC+\overline{B}C$；

(5) $L=A\overline{BC}+\overline{A}\overline{B}+\overline{A}D+C+BD$；

(6) $L(A, B, C)=\sum_m(0, 1, 3, 4, 5, 7)$；

(7) $L(A, B, C, D)=\sum_m(0, 2, 8, 10)$；

(8) $L(A, B, C, D)=\sum_m(0, 2, 3, 5, 7, 8, 10, 11, 13, 15)$；

(9) $L(A, B, C, D)=\sum_m(0, 1, 2, 5, 8, 9, 10, 12, 14)$；

(10) $L(A, B, C, D)=\sum_m(1, 2, 3, 4, 5, 7, 9, 15)$。

1-17 用卡诺图化简法将下列函数化简成为最简与或式：

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

(2) $L(A, B, C, D)=\sum_m(0, 1, 2, 3, 4, 8, 10, 11, 14, 15)$；

(3) $L(A, B, C, D)=\sum_m(0, 1, 2, 5, 6, 7, 8, 9, 13, 14)$；

(4) $L(A, B, C, D)=\sum_m(1, 3, 5, 7, 9)+\sum_d(10, 11, 12, 13, 14, 15)$；

(5) $L(A, B, C, D)=\sum_m(0, 5, 7, 8, 9, 11, 15)+\sum_d(2, 6, 10, 14)$；

(6) $L(A, B, C, D)=\sum_m(0, 1, 2, 4, 5, 6, 12)+\sum_d(3, 8, 10, 11, 14)$；

(7) $L=AB\overline{C}+AB\overline{D}+\overline{A}BC+AC\overline{D}$，约束条件为 $\overline{B}\overline{C}+\overline{B}CD=0$；

(8) $L(A, B, C, D)=\sum_m(0, 1, 3, 4, 5, 6, 7, 8, 9)$，约束条件为 $AB+AC=0$；

(9) $L(A, B, C, D, E)=\sum_m(2, 8, 10, 18, 24, 26, 31)$；

(10) $L(A, B, C, D, E, F)=\sum_m(2, 8, 10, 18, 24, 26, 34, 37, 42, 45, 50, 53, 58, 60, 62)$。

1-18 用卡诺图化简下列具有无关项的逻辑函数：

(1) $L=\overline{A}\overline{B}\overline{C}D+AB\overline{C}\overline{D}+\overline{A}B\overline{C}\overline{D}+\overline{A}BCD+A\overline{B}\overline{C}\overline{D}+A\overline{B}CD+ABC\overline{D}$，约束条件为 $A\overline{B}\overline{C}D+ABCD=0$；

(2) $L=C\overline{D}(A\oplus B)+\overline{A}B\overline{C}+\overline{A}\overline{C}D$，约束条件为 $AB+CD=0$；

(3) $L=(A\overline{B}+B)C\overline{D}+\overline{(A+B)(\overline{B}+C)}$，约束条件为 $ABCD+ACD+BCD=0$；

(4) $L(A, B, C, D)=\sum_m(3, 5, 6, 7, 10)$，约束条件为 $m_0+m_1+m_2+m_4+m_8=0$；

(5) $L(A, B, C, D)=\sum_m(2, 3, 7, 8, 11, 14)$，约束条件为 $m_0+m_5+m_{10}+m_{15}=0$。

1-19　在举重比赛中，有三名裁判，其中包括一名主裁判和两名副裁判。比赛时，只有主裁判判定运动员成绩有效，加上至少一名副裁判判定运动员成绩有效时，该运动员的成绩才有效。试列出该逻辑关系的真值表，并用逻辑电路实现该逻辑功能。

1-20　设计一个将一位十进制数对应的 8421BCD 码转换为格雷码的电路，用与非门实现。

第二章　逻辑门电路基础

本　章　提　要

本章首先讲解二极管、双极型三极管、单极型三极管的静态开关特性和动态开关特性。然后讨论各种工艺的逻辑门电路，包括二极管逻辑门电路、TTL逻辑门电路、ECL逻辑门电路、MOS逻辑门电路。不同工艺的逻辑门电路可以实现同样的逻辑功能，因为工艺不同，所以性能指标有好有差。最后讲解不同工艺的逻辑门电路之间的接口方法。

第一节　二极管、三极管的开关特性

一、二极管的开关特性

在模拟电子技术课程中，对二极管的特性已经进行了详细的分析，总结了二极管的各种应用，主要包括如下几个方面：一是当作非线性电阻来使用，所有时间内二极管全部工作在正向导通区；二是当作开关来使用，二极管某段时间内导通，某段时间内截止；三是当作小电压稳压器件来使用，所有时间内二极管全部工作在正向导通区；四是当作大电压稳压器件来使用，所有时间内二极管全部工作在反向击穿区。

在数字电路中，二极管工作在开关状态。二极管的开关特性包括静态开关特性和动态开关特性。二极管的静态开关特性是指二极管稳定地处于导通（开关处在闭合状态）和二极管稳定地处于截止（开关处在断开状态）时的特性。二极管的动态开关特性是指二极管从一个状态到另一个状态的过渡过程中的特性。

下面在回忆二极管的静态开关特性的基础上，重点讨论二极管的动态开关特性。

（一）二极管的静态开关特性

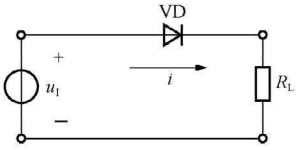


图 2-1　二极管电路图

二极管的静态开关特性研究的主要内容是，当二极管稳定地处于导通或截止状态时，所呈现出的性质特点。

在图 2-1 所示的二极管电路中，$R_L=10k\Omega$，判断当 $u_I=5V$ 和 $u_I=-5V$ 时二极管的状态，并求出电路中的电流 i。假设二极管为理想二极管。

在这个电路中，有一个二极管，首先要判断二极管的状态，然后才能求电路中的电流。为了求解方便，可以取一个电位参考点，如图 2-2 中 O 点所示。判断二极管状态的方法是将二极管断开，求阳极和阴极的电位差。若电位差大于零，则二极管正向导通；若电位差小于零，则二极管反向截止。因为假设是理想二极管，所以电位差与零做比较即可。

（1）$u_I=5V$，由于二极管阳极和阴极的电位差为 5V，所以二极管导通，用理想的导线代替二极管，等效电路如图 2-2 所示。此时，$i=5/10=0.5$（mA），二极管所处的状态相当于开关闭合。

（2）$u_I=-5V$，由于二极管阳极和阴极的电位差为 −5V，所以二极管截止，电路中没有回路，等效电路如图 2-3 所示。图中 $i=0mA$，二极管所处的状态相当于开关打开。

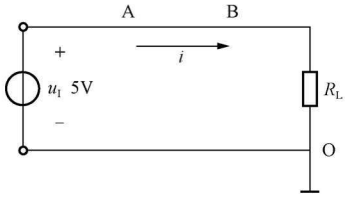


图 2-2　$u_I=5V$ 时的等效电路图

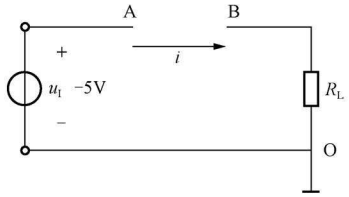


图 2-3　$u_I=-5V$ 时的等效电路图

二极管正偏时导通，管压降为 0V，流过二极管的电流大小决定于外电路，相当于开关闭合；二极管反偏时截止，流过二极管的电流为 0，相当于开关打开，二极管两端电压的大小决定于外电路。这就是二极管的静态开关特性。

（二）二极管的动态开关特性

二极管的动态开关特性是指二极管在导通和截止两种状态转换过程中的行为特点。

若给图 2-1 所示的二极管电路加入一个如图 2-4 所示的方波信号，电路中电流的波形怎样呢？

在 t_1 时刻以前，二极管已经处于导通状态，电路中的电流为 0.5mA。在 t_1 时刻，输入信号从 5V 瞬间变为−5V，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，t_1 时刻二极管并不立即截止，而是电路中的电流由正向的 0.5mA 变为 $i=-5/10=-0.5$（mA），等效电路如图 2-5 所示。在 t_1 时刻二极管两端的电压为 0，可以这样来理解，t_1 时刻二极管的状态还没来得及改变，维持正向导通时的状态不变。

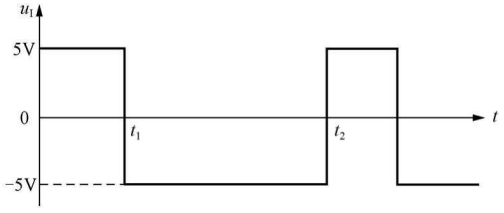


图 2-4　输入信号 u_I 的波形图

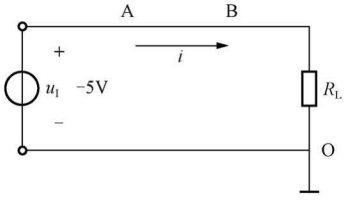


图 2-5　t_1 时刻图 2-1 二极管电路的等效电路

从 t_1 时刻开始，0.5mA 的电流维持一段时间 t_s 后才开始逐渐下降，再经过 t_t 后，下降到一个很小的数值 I_S（二极管反向饱和电流，由少子的漂移形成），这时二极管才进入截止状态。图 2-6（a）是输入信号的电压波形。图 2-6（b）是理想情况下的电流波形，即状态的转换瞬间完成。图 2-6（c）是实际情况下的电流波形。

1. 反向恢复过程

通常把二极管从正向导通转为反向截止所经历的转换过程称为反向恢复过程。在图 2-6（c）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，二极管的开关速度受到限制。产生反向恢复时间的原因是电荷存储效应。当二极管外加正向电压，即二极管处于正偏状态时，P 区的多子（空穴）向 N 区扩散，N 区的多子（自由电子）向 P 区扩散，这样空间电荷区变窄，而且载流子有相当数量的存储，即在 P 区内存储了自由电子，在 N 区内存储了空穴。当输入电压在 t_1 时刻突然从+5V 变为

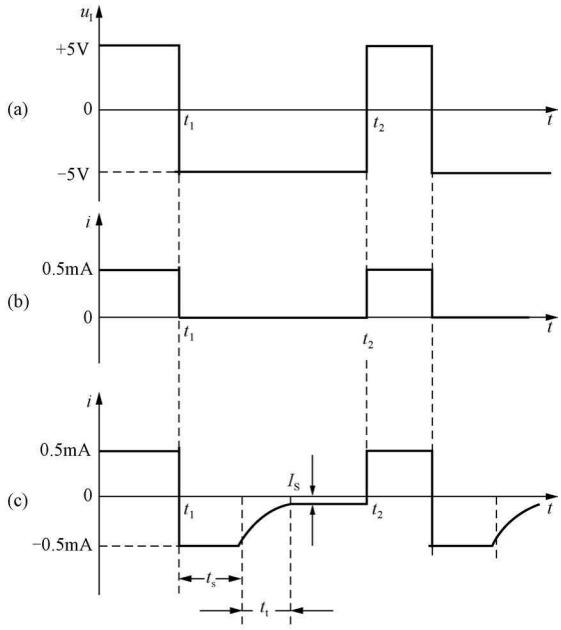


图 2-6 二极管电流的理想波形和实际波形图

−5V 时，P 区内存储的自由电子和 N 区内存储的空穴不会马上消失，而是通过两个途径逐渐消失，一是在反向电场的作用下，P 区的自由电子被拉回 N 区，N 区的空穴被拉回 P 区；二是与多数载流子复合。经过 t_s 存储时间后，P 区内存储的自由电子和 N 区内存储的空穴已经显著减少，空间电荷区逐渐变宽，再经过 t_t 渡越时间后，电路中的电流变为反向饱和电流。

二极管从截止状态到导通状态的时间很短，可以近似认为瞬间完成。

2. 对输入信号 u_I 的要求

由于反向恢复过程的存在，所以对输入信号有一定的要求。u_I 的负半周的宽度应大于 t_{re}，这样二极管才具有单向导电性；若小于 t_{re}，二极管还没有到达截止状态，就又必须随输入正脉冲而导通，从而失去单向导电性。由于二极管从截止状态到导通状态的时间很短，所以对输入信号 u_I 的正半周的宽度要求比较低。

可见，输入信号 u_I 的频率不可太高，以保证二极管能够可靠进入导通状态和截止状态。

二、双极型三极管的开关特性

（一）双极型三极管的静态开关特性

双极型三极管（简称三极管）的静态开关特性说明当三极管稳定地处于饱和或截止状态时，三极管所呈现出的性质特点。

在模拟电子技术课程中，已经知道三极管有两个方面的应用：一是在模拟电路中作为放大元件，二是在数字电路中作为开关元件。三极管有三个状态，即放大状态、饱和状态和截止状态。在放大电路中的三极管工作在放大状态，在数字电路中三极管工作在饱和状态和截止状态。下面我们更深入地讨论三极管的工作状态。

判断三极管工作状态的解题思路如下：

（1）把三极管从电路中拿走，在此电路拓扑结构下求三极管的发射结电压，若发射结反偏、零偏或小于死区电压值，则三极管截止；若发射结正偏，则三极管可能处于放大状态或处于饱和状态，需要进一步判断。

（2）把三极管放入电路中，电路的拓扑结构回到原始状态。假设三极管处于临界饱和状态（三极管既可以认为是处于饱和状态也可以认为是处于放大状态，在放大区和饱和区的交界区域，此时三极管既有饱和状态时的特征 $U_{CES}=0.3V$，又有放大状态时的特征 $I_C=\beta I_B$），求此时三极管的集电极临界饱和电流 I_{CS}，进而求出基极临界饱和电流 I_{BS}。集电极临界饱和电流 I_{CS}是三极管的集电极可能流过的最大电流。

（3）在原始电路拓扑结构基础上，求出三极管的基极支路中实际流动的电流 i_B。

（4）比较 i_B 和 I_{BS}的大小：若 $i_B>I_{BS}$（或者 $\beta i_B>I_{CS}$），则三极管处于饱和状态；若 $i_B<I_{BS}$（或者 $\beta i_B<I_{CS}$），则三极管处于放大状态。

【例 2-1】 试判断图 2-7 电路中三极管的状态，其中 $R_b=2k\Omega$，$R_c=2k\Omega$，$V_{CC}=12V$，$\beta=50$。

解 把三极管从电路中拿走，剩下的电路如图 2-8 所示。在此电路拓扑结构下求三极管的发射结电压。可以看出，发射结零偏，所以三极管截止。

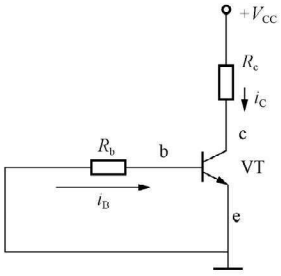


图 2-7　[例 2-1] 的电路图

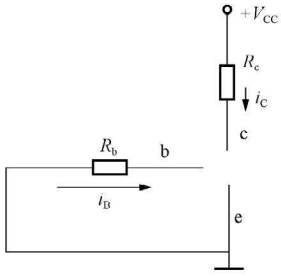


图 2-8　图2-7 三极管拿走后的电路图

【例 2-2】 试判断图 2-9 电路中三极管的状态，其中 $R_b=20k\Omega$，$R_c=2k\Omega$，$V_{CC}=12V$，$\beta=50$。

解 （1）把三极管从电路中拿走，剩下的电路如图 2-10 所示。在此电路拓扑结构下求三极管的发射结电压 $U_{be}=U_b-U_e=12-0=12$（V），因为 $U_{be}>0$（V），所以发射结正偏，则三极管可能处于放大状态或处于饱和状态，需要进一步判断。

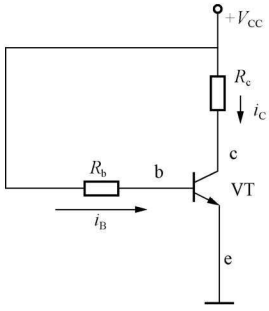


图 2-9　[例 2-2] 的电路图

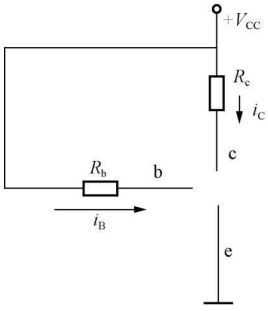


图 2-10　图 2-9 三极管拿走后的电路图

（2）把三极管放入电路中，电路的拓扑结构回到原始状态，如图 2-9 所示。假设三极管处于临界饱和状态，求此时三极管的集电极临界饱和电流 I_{CS}，进而求出基极临界饱和电流 I_{BS}。它们分别为

$$I_{CS}=\frac{V_{CC}}{R_c}=\frac{12}{2}=6(\text{mA})$$

$$I_{BS}=\frac{I_{CS}}{\beta}=\frac{6}{50}=0.12(\text{mA})$$

集电极临界饱和电流 I_{CS}是三极管的集电极可能流过的最大电流。

（3）在图 2-9 所示的原始电路拓扑结构基础上，求出三极管的基极支路中实际流动的电流 i_B。

$$i_B=\frac{12-0.7}{20}\approx 0.565(\text{mA})$$

（4）比较 i_B 和 I_{BS}的大小。因为 $i_B>I_{BS}$，所以三极管处于饱和状态。

【例 2-3】 在［例 2-2］中，如果将 R_b 改为 200kΩ，再判断三极管的状态。

解 与［例 2-2］相比，本题中将 R_b 改为 200kΩ，其他的量都不变，则三极管的基极支路中实际流动的电流 i_B 变为

$$i_B=\frac{12-0.7}{200}\approx 0.0565(\text{mA})$$

因为 $i_B<I_{BS}$，所以三极管处于放大状态。

通过［例 2-2］和［例 2-3］可以总结出如下的规律：要想改变三极管所处的状态，只要改变 i_B 和 I_{BS}的相对关系即可。若保持 I_{BS}不变，则可以通过改变 R_b 的大小来改变 i_B。若保持 i_B不变，则可以通过改变 R_c 的大小来改变 I_{BS}。

三极管在三个状态下的特点总结如下：

截止区：三极管的三个电极所在的支路中的电流为 0，任意两个电极之间的电压的多少，决定于外电路，因为电路要满足电路定律方程；集电结和发射结均处于反偏状态。

饱和区：NPN 管的 $u_{BE}=0.7\text{V}$，PNP 管的 $u_{BE}=-0.2\text{V}$，三极管的集电极电流和基极电流之间没有 β 的关系存在，三极管的三个电极所在的支路中的电流决定于外电路，满足电路定律方程；集电结和发射结均正偏。

放大区：NPN 管的 $u_{BE}=0.7\text{V}$，PNP 管的 $u_{BE}=-0.2\text{V}$，三极管的集电极电流和基极电流之间有 β 的关系存在，三极管的三个电极所在的支路中的电流决定于外电路，满足电路定律方程；发射结正偏，集电结反偏。

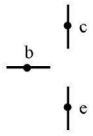
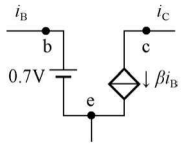
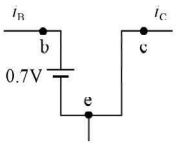
表 2-1 总结了三极管在三个状态下的特点。

【例 2-4】 电路及参数如图 2-11 所示，三极管的 $U_{BE}=0.7\text{V}$，$\beta=60$，输入电压 u_I 取值 3V 和 −2V。

（1）当 $u_I=3\text{V}$ 时判断三极管的状态，并求出 i_C 和 u_O 的值。

（2）当 $u_I=-2\text{V}$ 时判断三极管的状态，并求出 i_C 和 u_O 的值。

表 2-1　　NPN 型三极管三种工作状态的特点

工作状态		截　止	放　大	饱　和
条　件		$i_B \approx 0$	$0 < i_B < I_{BS}$	$i_B > I_{BS}$
工作特点	偏置情况	发射结电压<0.5V 集电结反偏	发射结正偏且>0.5V 集电结反偏	发射结正偏且>0.5V 集电结正偏
	集电极电流	$i_C \approx 0$	$i_C = \beta i_B$	$i_C = I_{CS} \approx V_{CC}/R_c$
	管压降	$U_{CE} \approx V_{CC}$	$U_{CE} = V_{CC} - I_C R_c$	$U_{CE} = U_{CES} \approx 0.3V$
	近似的等效电路			
	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

解　(1) $u_I = 3V$ 时，有

$$i_B = \frac{3-0.7}{10} \approx 0.23(\text{mA})$$

$$I_{BS} = \frac{V_{CC}}{\beta R_c} = \frac{5}{60 \times 10} \approx 0.0083(\text{mA})$$

因为 $i_B > I_{BS}$，所以三极管处于饱和状态。如图 2-12 中的 F 点所示。$u_O = U_{CES} = 0.3V$。

三极管处于饱和状态相当于一个闭合的开关，开关的两个触点是集电极和发射极，示意图如图 2-13 (a) 所示。

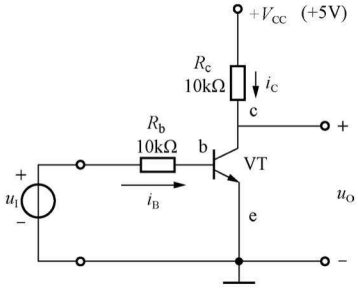


图 2-11　[例 2-4] 的电路图

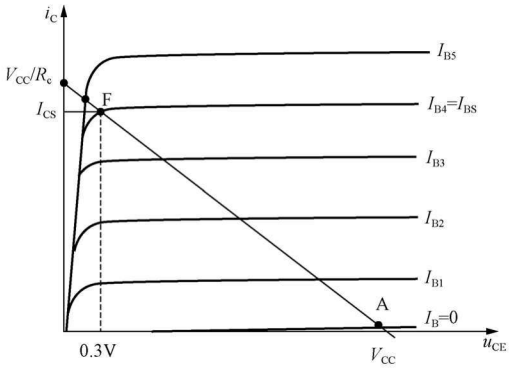


图 2-12　[例 2-4] 电路中三极管状态在输出特性曲线上的点

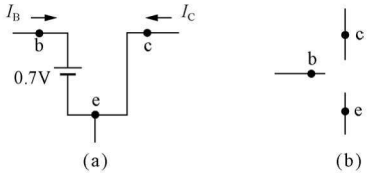


图 2-13　三极管处于开关状态时的等效图

(a) 三极管处于饱和状态时的等效图；

(b) 三极管处于截止状态时的等效图

(2) $u_I = -2V$ 时，三极管的发射结反偏，三极管处于截止状态，如图 2-12 中的 A 点

所示。此时，$u_O=5V$，三极管处于截止状态，相当于一个打开的开关，开关的两个触点是集电极和发射极，示意图如图 2-13（b）所示。

（二）双极型三极管的动态开关特性

若给图 2-11 所示的三极管电路加入一个如图 2-14（a）所示的矩形波输入信号，电路中集电极电流的波形怎样呢？在图 2-14（a）中，0 到 t_1 时刻 u_I 的值为－2V，在 t_1 时刻 u_I 的值变为＋3V，u_I 在＋3V 维持一段时间后，在 t_2 时刻变为－2V。从［例 2-4］可以知道，当 $u_I=3V$ 时，三极管处在饱和状态，输出 $u_O=0.3V$；当 $u_I=-2V$ 时三极管处在截止状态，输出 $u_O=5V$。在图 2-14（a）所示输入信号的作用下，三极管将在饱和状态和截止状态之间转换。在理想情况下，三极管集电极电流的波形如图 2-14（b）所示，输入电压变化的瞬间，电流的变化立即完成。图 2-14（c）是实际情况下三极管集电极电流的波形。

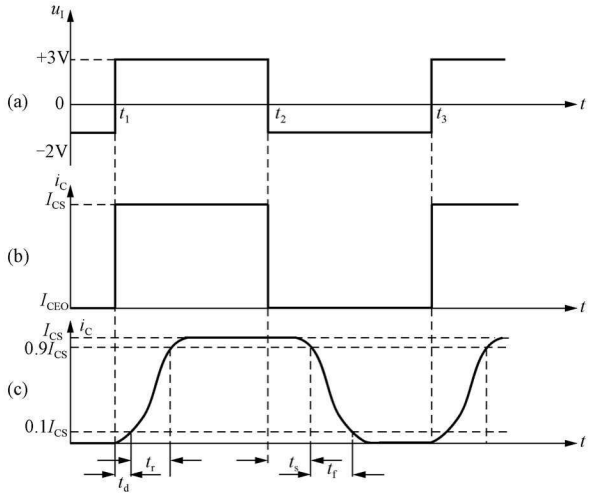


图 2-14　三极管集电极电流的理想波形和实际波形

（a）矩形波输入信号；（b）三极管集电极电流的理想波形；

（c）三极管集电极电流的实际波形

在图 2-14（c）中，0 到 t_1 时刻 u_I 的值为－2V，三极管已经稳定地处于截止状态，三极管集电极电流为 0。在 t_1 时刻 u_I 的值变为＋3V，三极管不能立即进入饱和状态，而是要经过延迟时间 t_d 和上升时间 t_r 后三极管才能稳定地处于饱和状态，三极管集电极的电流才能稳定地停留在集电极临界饱和电流 I_{CS}。u_I 在＋3V 维持一段时间后，在 t_2 时刻 u_I 的值变为－2V，此时三极管不能立即进入截止状态，而是要经过存储时间 t_s 和下降时间 t_f 后，才能稳定地处于截止状态，三极管集电极的电流才能稳定地停留在 0。

三极管动态特性中各时间的含义如下：

（1）延迟时间 t_d——从输入信号 u_I 正跳变的瞬间开始，到集电极电流 i_C 上升到 $0.1I_{CS}$ 所需的时间；

（2）上升时间 t_r——集电极电流从 $0.1I_{CS}$上升到 $0.9I_{CS}$所需的时间；

（3）存储时间 t_s——从输入信号 u_I 下跳变的瞬间开始，到集电极电流 i_C 下降到 $0.9I_{CS}$

所需的时间；

（4）下降时间 t_f——集电极电流从 $0.9I_{CS}$ 下降到 $0.1I_{CS}$ 所需的时间；

（5）开通时间 $t_{on}=t_d+t_r$；

（6）关闭时间 $t_{off}=t_s+t_f$。

由于三极管的状态转换不能瞬间完成，所以对输入信号有一定的要求，输入信号 u_I 的正半周维持的时间必须大于 t_{on}，以保证三极管能可靠进入饱和状态。输入信号 u_I 的负半周维持的时间必须大于 t_{off}，以保证三极管能可靠进入截止状态。可见，输入信号 u_I 的频率不可太高，要保证三极管有足够的时间能够可靠进入饱和状态和截止状态。

三、MOS 管的开关特性

在模拟电子技术课程中，我们已经知道 MOS 管有三个方面的应用：一是在模拟电路中当作电压控制器件用来组成放大电路；二是在数字电路中作为开关元件；三是当作压控可变电阻，即非线性电阻来使用。MOS 三极管有三个状态，即放大状态、压控可变电阻状态和截止状态。在放大电路中的 MOS 三极管处在放大状态，在数字电路中 MOS 三极管工作在压控可变电阻状态和截止状态，即工作在开关状态。下面我们针对 N 沟道增强型 MOS 管讨论三极管的静态开关特性和动态开关特性。

（一）MOS 管的静态开关特性

图 2-15 是 MOS 三极管电路，U_{VT} 为 N 沟道增强型 MOS 管的开启电压。当 u_I 为低电平，且 $u_I<U_{VT}$ 时，MOS 管 VT 截止，没有电流流过电阻 R_D，所以 $u_O=U_{OH}=V_{DD}$，u_O 输出为高电平。当 u_I 为高电平，且 $u_I>U_{VT}$ 时，MOS 管 VT 处于线性电阻区，有电流流过电阻 R_D，由于处于线性电阻区的 MOS 管等效电阻较小，所以 $u_O=U_{OL}\approx 0$，u_O 输出为低电平。可见，其输出与输入为非逻辑关系。

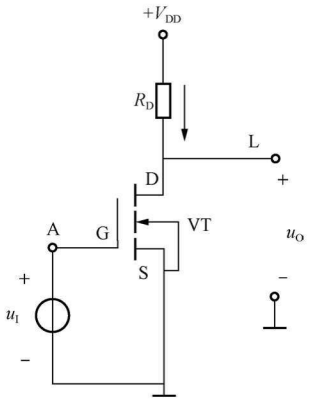


图 2-15　MOS 三极管电路

（二）MOS 管的动态开关特性

在 MOS 集成电路中，由于 MOS 管的输出总要接至若干其他 MOS 管的输入端，所以相当于带有电容性负载。开关电路的开关时间主要取决于输入和输出回路中电容的充、放电时间。

当 u_I 由低电平跳变到高电平时，MOS 管 VT 立即处于线性电阻区，负载电容通过 MOS 管 VT 放电，随着放电过程的进行，u_O 逐渐下降到 U_{OL}。由于 MOS 管处于线性电阻区时的等效电阻较小，所以放电过程比较快。

当 u_I 由高电平跳变到低电平时，MOS 管 VT 立即处于截止状态，电源通过电阻 R_D 对电容充电，随着充电过程的进行，u_O 逐渐上升到 U_{OH}。

第二节　二极管逻辑门电路

一、正与门电路

图 2-16 是两个二极管组成的与门电路，假设二极管是理想二极管。图 2-17 为图 2-16 所示电路的简化画法，给电路取一个电位参考点，三个电压源的符号省略不画，而是用相应

的电位值来表示对应的电压源。下面针对图 2-16 来分析电路在 U_A、U_B 为四种不同取值组合情况下的输出 U_L 的值。

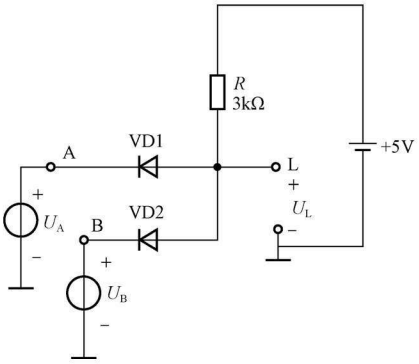


图 2-16　二极管组成的与门电路

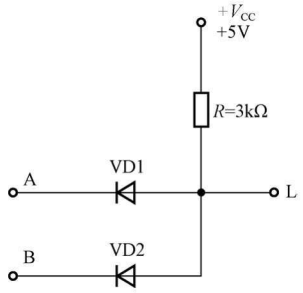


图 2-17　图 2-16 电路的简化画法

分析这种由两个二极管组成的电路的步骤如下：

(1) 将二极管从电路中拿走，在此电路的基础上求两个二极管的阳极和阴极之间的电位差。

(2) 两个二极管的阳极和阴极之间的电位差共有三种情况：均小于 0；均大于 0；一个为正，另一个为负。

(3) 根据不同的情况做出判断：

1) 均小于 0 时，可立即得出结论，两个二极管均处于截止状态。

2) 均大于 0 时，这其中会有一大一小，可以得出结论，大的那个二极管一定处于导通状态，小的那个状态不定，需要做进一步的判断。大的那个二极管导通后用理想的导线代替，这时整个电路就转化成了只有一个二极管的电路，按照一个二极管电路的求解方法继续判断，从而得出最后的结论。如果电位差均大于 0，而且相等，则认为两个二极管同时导通。

3) 一个为正，另一个为负时，正的那个二极管一定导通，负的那个状态不定，需要做进一步的判断。正的那个二极管导通后用理想的导线代替，这时整个电路就转化成了只有一个二极管的电路，按照一个二极管电路的求解方法继续判断，从而得出最后的结论。

根据前面的计算方法，可得出 $U_A=0V$、$U_B=0V$，$U_A=0V$、$U_B=5V$，$U_A=5V$、$U_B=0V$，$U_A=5V$、$U_B=5V$ 四种情况下的输出电压值，如表 2-2 所示。

1. 正逻辑体制

在表 2-2 输出电压列表中，这样来约定，高电平表示 1，低电平表示 0，这就是正逻辑体制。按该体制可以得到如表 2-3 所示的正逻辑体制下的真值表。观察表 2-3，可以看出输出信号与输入信号之间是与逻辑关系，实现的是与逻辑运算。所以称图 2-16、图 2-17 是正逻辑体制下实现与逻辑运算的门电路，简称为正与门。与门的逻辑符号见图 1-7。

表 2-2　图 2-16 电路的输出电压列表

输入		输出
U_A	U_B	U_L
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

表 2-3　正逻辑体制下的真值表

输入		输出
A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

2. 负逻辑体制

在表 2-2 输出电压列表中，如果约定，高电平表示 0，低电平表示 1，这就是负逻辑体制，由此可以得到如表 2-4 所示的负逻辑体制下的真值表。观察表 2-4，可以看出输出信号与输入信号之间是或逻辑关系，实现的是或逻辑运算。所以称图 2-16、图 2-17 是负逻辑体制下实现或逻辑运算的门电路，简称为负或门。或门的国标符号见图 1-8。

表 2-4　负逻辑体制下的真值表

输入		输出	输入		输出
A	B	L	A	B	L
1	1	1	0	1	1
1	0	1	0	0	0

由上面的分析可以看出，同样的一个电路，既可以实现与逻辑运算，又可以实现或逻辑运算，这决定于如何对高电平和低电平进行逻辑赋值。有两种逻辑赋值方法可以选择，即正逻辑体制和负逻辑体制。高电平表示 1，低电平表示 0，这就是正逻辑体制；高电平表示 0，低电平表示 1，这就是负逻辑体制。在后面的叙述中，本书中如果不做明确说明，均使用正逻辑体制。

二、正或门电路

图 2-18 是两个二极管组成的电路，假设二极管是理想二极管。图 2-19 是电路的简化画法，给电路取一个电位参考点，两个电压源的符号省略不画，而是用相应的电位值来表示对应的电压源。下面针对图 2-18 来分析电路在 U_A、U_B 为四种不同取值组合情况下的输出 U_L 的值。根据前面介绍的分析的方法，得出如表 2-5 所示的输出电压列表。

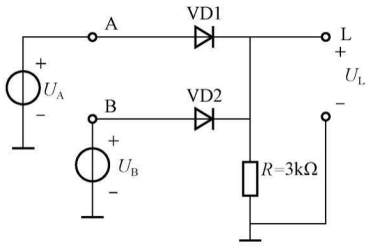


图 2-18　正或门电路的完整画法

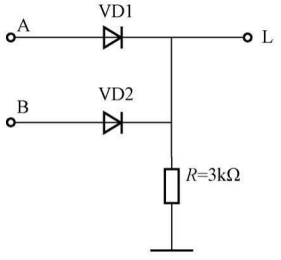


图 2-19　图 2-18 电路的简化画法

1. 正逻辑体制

在表 2-5 输出电压列表中，这样来约定，高电平表示 1，低电平表示 0，即应用正逻辑体制，因此可以得到如表 2-6 所示的正逻辑体制下的真值表。观察表 2-6，可以看出输出信号与输入信号之间是或逻辑关系，实现的是或逻辑运算。所以称图 2-18 是正逻辑体制下实现或逻辑运算的门电路，简称为正或门。

表 2-5　图 2-18 所示电路的输出电压列表

输　入		输　出
U_A	U_B	U_L
0V	0V	0V
0V	5V	5V
5V	0V	5V
5V	5V	5V

表 2-6　正逻辑体制下的真值表

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

2. 负逻辑体制

在表 2-5 输出电压列表中，这样来约定，高电平表示 0，低电平表示 1，即应用负逻辑体制，由此可以得到如表 2-7 所示的负逻辑体制下的真值表。观察表 2-7，可以看出输出信号与输入信号之间是与逻辑关系，实现的是与逻辑运算。所以称图 2-18 是负逻辑体制下实现与逻辑运算的门电路，简称为负与门。

表 2-7　负逻辑体制下的真值表

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

三、非门电路

分析图 2-11 所示电路在 U_A 为 3V、0V 情况下的输出 U_L 的值。根据前面的分析，可以列出如表 2-8 所示的输出电压列表。无论使用正逻辑体制还是负逻辑体制，都得到如表 2-9 所示的真值表。非门的逻辑符号见图 1-9。

表 2-8　图 2-11 所示电路的输出电压列表

输　入	输　出
U_A	U_L
0V	5V
3V	0.3V

表 2-9　图 2-11 所示电路的真值表

输　入	输　出
A	L
0	1
1	0

以上所讨论的是基本的与门、或门和非门，利用它们可以实现与、或、非三种基本逻辑运算。如果将它们串联连接，还可以实现更复杂的逻辑运算，但是由于这些电路的输出电阻较大，带负载能力差，开关性能不是特别理想，因而下面讨论性能更好的逻辑门电路。

第三节　TTL 逻辑门电路

TTL 是 Transistor Transistor Logic（晶体管—晶体管逻辑电路）的缩写，是应用最为广泛的一类集成电路。TTL 集成电路分为两类：74 系列和 54 系列。74 系列 TTL 集成电路的编号为 74XX，54 系列 TTL 集成电路的编号为 54XX，其中“XX”表示不同逻辑功能电路的编号。74XX 系列集成电路主要应用于民用电子产品的设计和生产，该类型集成电路可以在 0～70℃环境温度下工作。54 XX 系列集成电路主要应用于军用电子产品的设计和生产，该类型集成电路可以在－55～＋125℃恶劣环境下工作。为了便于设计和使用，具有相同编号的 54 和 74 系列集成电路的引脚排列完全相同。在集成电路编号中，位于 54 和 74 字符后面的字符串指明了集成电路生产所采用的工艺和技术。

TTL 逻辑门电路由若干双极型三极管（BJT）和电阻组成。这种门电路于 20 世纪 60 年代出现，随后经过电路结构和工艺方面的改进，性能有了较大地提高，至今仍有广泛的应用。目前 MOS 工艺的集成电路已经是电子系统设计的主流电路，但是本书的安排还是先从 TTL 逻辑门电路开始学习一些基本的概念，这些概念大多数可以应用于 MOS 电路中。TTL 门电路除反相（非）的逻辑功能外，还有与非、或非、与或非、异或等多种逻辑功能类型。TTL 门电路除标准的生产工艺外，还有低功耗（L）、低功耗肖特基（LS）、先进的低功耗肖特基（ALS）、先进的肖特基（AS）、高速（H）等生产工艺的门电路。TTL 门电路有集电极开路门、三态门等电路结构以便于实现线与、总线等功能。本节以标准生产工艺的非门为例解释 TTL 门电路的一些基本参数，这些参数的定义对其他逻辑功能类型的标准生产工艺的 TTL 门电路完全适用。

一、标准生产工艺的 TTL 非门的工作原理

第一节中的图 2-11 是由一个三极管组成的普通非门（反相器）的电路，这种非门电路的开关速度不高，而 TTL 非门的开关速度有了明显的提高。图 2-20 是美国德州仪器公司生产的标准生产工艺的 TTL 非门芯片 SN7404 的内部电路图。从图可以观察出，输入级有一个三极管，输出级有两个三极管，TTL 就是从电路的这种结构来命名的。下面分析标准生产工艺的 TTL 非门的工作原理。

（一）输入 u_I 为高电平 3.6V

在图 2-20 中，当 u_I 为高电平 3.6V 时，可以通过如下的方法来大致估算以下电路中各点的电位，如果要得出电路各点准确的电位值，则可以用专门的计算机软件来计算。本处只是大致了解电路的工作原理，不需要特别准确的值，所以此处的估算方法可以满足要求。将三极管 VT1 从电路中拿走，电路从三极管 VT1 的三个极断开，电阻 R_{b1} 所在的支路断开，没有电流流过电阻 R_{b1}，B_1 点电位为 5V，B_1 点和 A 点的电位差为 5－3.6＝1.4(V)，使三极管 VT1 的发射结导通，如图 2-21 所示，发射结导通后两端电压为 0.7V，从而使 B_1 点的电位钳位在 3.6＋0.7＝4.3（V），4.3V 的电压降落在三极管 VT1 的集电结、三极管 VT2 的发射结、三极管 VT3 的发射结上，使这三个 PN 结导通，导通后的 PN 结两端电压为 0.7V，所以最后使 B_1 点的电位钳制在 0.7＋0.7＋0.7＝2.1（V）。这时三极管 VT1 的发射结反偏、集电结正偏，称处于倒置状态。电路参数的设计使 VT2、VT3 一定能进入饱和状态。由于 VT3 饱和，输出电压为 $u_O = U_{CES3} \approx 0.3$V，因为 VT2 饱和，故有 $U_{C2} = U_{E2} +$

$U_{CES2}=0.7+0.3=1$（V）。$U_{C2}=1$V 使 VT4 和二极管 VD 都截止。实现了输入高电平时，输出为低电平。

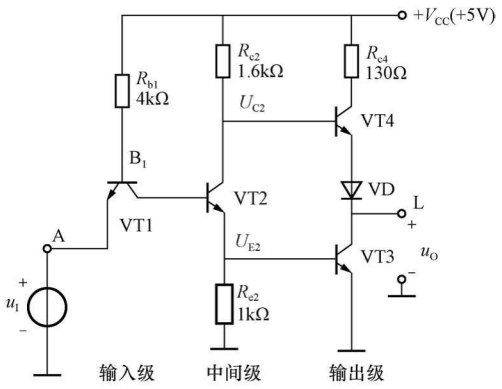


图 2-20 标准生产工艺的 TTL 非门的电路图

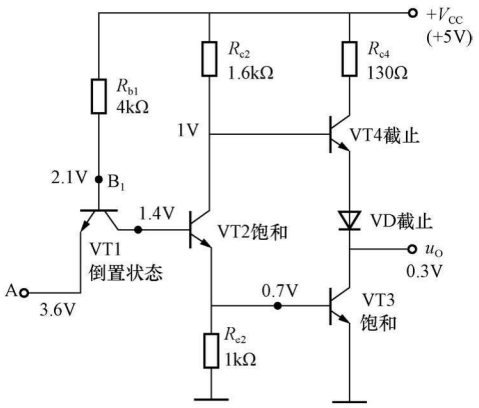


图 2-21 当输入为高电平 3.6V 时，TTL 非门电路各点的电位示意图

此处给出“开门”的含义，将三极管 VT3 处于饱和状态叫做开门。三极管处于饱和状态相当于三极管的 C、E 之间的开关闭合，将门打开了。

（二）输入 u_I 为低电平 0.3V

在图 2-20 中，当 u_I 为低电平 0.3V 时，可以通过如下的方法来大致估算电路中各点的电位。将三极管 VT1 从电路中拿走，电路从三极管 VT1 的三个极断开，电阻 R_{b1} 所在的支路断开，没有电流流过电阻 R_{b1}。此时，B_1 点电位为 5V，B_1 点和 A 点的电位差为 $5-0.3=4.7$（V），使三极管 VT1 的发射结导通。发射结导通后两端电压为 0.7V，从而使 B_1 点的电位钳位在 $0.3+0.7=1$（V）。1V 的电压降落在三极管 VT1 的集电结、三极管 VT2 的发射结、三极管 VT3 的发射结上，使这三个 PN 结都处于截止状态，所以最后使 B_1 点的电位钳位在 1V，各点电位示意图如图 2-22 所示。VT2 截止，其集电极支路没有电流流过，相当于断开。流过 R_{c2} 的电流就是 VT4 的基极电流，该电流较小可以忽略，所以 $U_{B4}\approx V_{CC}=5$V，使 VT4 和 VD 导通，则有 $u_O\approx V_{CC}-U_{BE4}-U_D=5-0.7-0.7=3.6$（V）。此电路实现了输入为低电平时，输出为高电平。

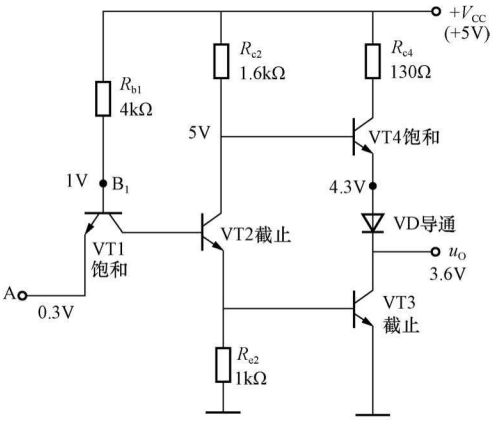


图 2-22 当输入为低电平 0.3V 时 TTL 非门电路各点的电位示意图

此处给出“关门”的定义，将三极管 VT3 处于截止状态叫做关门。

读者要特别注意“开门”和“关门”的定义，三极管 VT3 处于饱和状态叫做开门，三极管 VT3 处于截止状态叫做关门。

标准生产工艺的 TTL 非门电路的特点是输入级采用三极管以提高工作速度，采用推拉式输出级以提高开关速度和带负载能力。推拉式输出级的输出阻抗比较小，可迅速给负载电容充放电，提高了开关速度和带负载能力。

二、TTL 非门的电压传输特性曲线和电路参数

（一）电压传输特性曲线

电压传输特性曲线是指 TTL 非门（反相器）的输出电压与输入电压之间的关系曲线。图 2-20 所示的标准生产工艺的 TTL 反相器电路的电压传输特性曲线如图 2-23 所示，下面将对其进行分析。

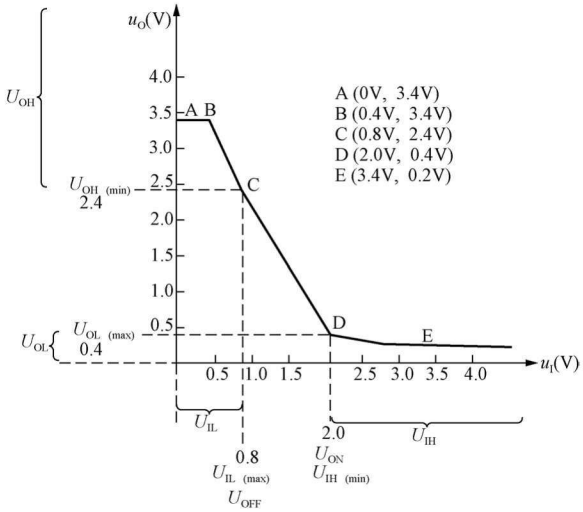


图 2-23　标准生产工艺的 TTL 非门的电压传输特性曲线

1. ABC 段

在 ABC 段时，$u_I \approx 0 \sim 0.8$V，输出电压 u_O 大于 2.4V。三极管 VT1 饱和，三极管 VT2、VT3 截止，三极管 VT4 饱和。在 ABC 段，由于输出级的三极管 VT3 是截止的，所以称 ABC 段为截止区，也叫做关门。

2. CD 段

CD 段时，$u_I \approx 0.8 \sim 2.0$V，输入电压 u_I 增加时，输出电压 u_O 下降速度很快，CD 段又叫做传输特性的过渡区。

3. DE 段

DE 段时，$u_I > 2.0$V 时，三极管 VT1 的发射结反偏、集电结正偏，称处于倒置状态；VT2、VT3 进入饱和状态；VT4 截止，u_O 小于 0.4V。由于输出级的三极管 VT3 是饱和的，所以称 DE 段为饱和区，也叫做开门。

（二）从电压传输特性曲线上可以得出的电路参数

1. 输出高电平 U_{OH}

输出高电平 U_{OH} 是一个电压范围，芯片 SN7404 的 U_{OH} 在 2.4V 和 5V 之间，其典型值为 3.4V。

2. 标准输出高电平 U_{SH}

标准输出高电平 U_{SH} 是输出高电平的下限，也记做 $U_{OH(min)}$，芯片 SN7404 的 $U_{OH(min)}$ 为 2.4V。

3. 输出低电平 U_{OL}

输出低电平 U_{OL} 是一个电压范围，芯片 SN7404 的 U_{OL} 在 0V 和 0.4V 之间，其典型值为

0.2V。

4. 标准输出低电平 U_{SL}

标准输出低电平 U_{SL} 是输出低电平的上限，也记做 $U_{OL(max)}$，芯片 SN7404 的 $U_{OL(max)}$ 为 0.4V。

5. 输入高电平 U_{IH}

输入高电平 U_{IH} 是一个电压范围，芯片 SN7404 的 U_{IH} 在 2V 和 5V 之间。

6. 输入高电平的下限 $U_{IH(min)}$（开门电平 U_{ON}）

输入高电平的下限 $U_{IH(min)}$ 是输出电压为 $U_{OL(max)}$ 时对应的输入电压，即输入高电平的最小值，如果输入的电压值比这个值还要小，那么输出的电压值将要比 $U_{OL(max)}$ 大，输出不能再被认为是低电平，非逻辑关系被破坏。芯片 SN7404 的 $U_{IH(min)}$ 值为 2V。输入高电平的下限 $U_{IH(min)}$ 还有一个名字叫开门电平 U_{ON}，因为当输入为高电平的下限 $U_{IH(min)}$ 时，图2-20中输出级的三极管 VT3 处于饱和状态，称为开门，所以 $U_{IH(min)}$ 又称为开门电平 U_{ON}。

7. 输入低电平 U_{IL}

输入低电平 U_{IL} 是一个电压范围，芯片 SN7404 的 U_{IL} 在 0V 和 0.8V 之间。

8. 输入低电平的上限 $U_{IL(max)}$（关门电平 U_{OFF}）

输入低电平的上限 $U_{IL(max)}$ 是输出电压为 $U_{OH(min)}$ 时对应的输入电压，即输入低电平的最大值，如果输入的电压值比这个值还要大，那么输出的电压值将要比 $U_{OH(min)}$ 小，输出不能再被认为是高电平，非逻辑关系被破坏。芯片 SN7404 $U_{IL(max)}$ 的值为 0.8V。输入低电平的上限 $U_{IL(max)}$ 还有一个名字叫关门电平 U_{OFF}，因为当输入为低电平的上限 $U_{IL(max)}$ 时，图 2-20 中输出级的三极管 VT3 处于截止状态，称为关门，所以 $U_{IL(max)}$ 又称为关门电平 U_{OFF}。

9. 噪声容限电压

在逻辑电路中，一个门的输出信号要送给其他的门作为输入信号。通过前面的分析可以看出，门输入的高、低电平范围要比输出的高、低电平范围宽。这样就有一个输入的抗干扰能力问题。以下分两种情况讨论。

（1）输入高电平噪声容限电压（最大允许负向干扰电压）。输入高电平噪声容限电压中的“输入”是指前一级门电路的输出作为后面门电路的输入，这里特别提醒读者注意理解。

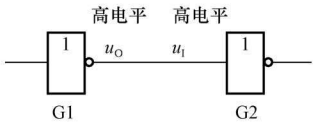


图 2-24 解释输入高电平噪声容限电压含义的电路图

输入高电平噪声容限电压是衡量门电路抗干扰能力的一个参数。在图 2-24 中，非门 G1 的输出信号送给非门 G2 的输入端。当 G1 输出高电平时，非门 G2 的输入端将送入高电平。G1 输出的高电平在 2.4V 和 5V 之间，不会低于 2.4V。而对于非门 G2 来说，在 2V 和 5V 之间的电压都算是输入了高电平。

假设非门 G1 输出的高电平在 2.4V，处于高电平范围的最小值。有许多因素在连接非门 G1 的输出和非门 G2 的输入端这根连接导线上串入干扰，如果在这根连线上串入了一个正向干扰电压，那么送入非门 G2 输入端的电压将会比 2.4V 还要高，对于非门 G2 来说，这个电压是高电平，所以串入正向干扰电压不会出现逻辑错误。如果在这根线上串入了一个负向干扰电压，那么送入非门 G2 输入端的电压将会比 2.4V 低，对于非门 G2 来说，只要这个电压不低于 2V，这个电压还是高电平，所以在这根连线上串入的负向干扰电压不能大于 2.4－2.0＝0.4（V）。

如果非门 G1 输出的高电平没有处于最小值，大于 2.4V，比如为 3.6V，那么在这根连线上串入的负向干扰电压最大就可以达到 3.6－2.0＝1.6（V），从而不会引起逻辑错误。

当一个电路在正常工作时，非门 G1 输出的高电平可能是 2.4V 和 5V 之间的任意值，这决定于不同的时刻电路的带负载情况。输入高电平噪声容限又称为最大允许负向干扰电压，由 G1 输出 2.4V 时的值决定。可由下式得到

$$\text{输入高电平噪声容限电压 } U_{NH} = U_{OH(min)} - U_{ON} = U_{OH(min)} - U_{IH(min)}$$
$$= 2.4 - 2.0 = 0.4(V)$$

（2）输入低电平噪声容限电压（最大允许正向干扰电压）。

输入低电平噪声容限电压也是衡量门电路抗干扰能力的一个参数。在图 2-25 中，非门 G1 的输出信号送给非门 G2 的输入端。当 G1 输出低电平时，非门 G2 的输入端将送入低电平。G1 输出的低电平在 0V 和 0.4V 之间，不会高于 0.4V。而对于非门 G2 来说，在 0V 和 0.8V 之间的电压都算是输入了低电平。

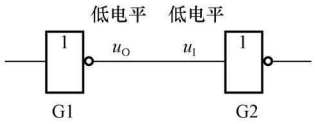


图 2-25　解释输入低电平噪声容限电压含义的电路图

假设非门 G1 输出的低电平在 0.4V，处于低电平范围的最大值。有许多因素在连接非门 G1 的输出和非门 G2 的输入端这根连接导线上串入干扰，如果在这根连线上串入了一个负向干扰电压，那么送入非门 G2 输入端的电压将会比 0.4V 还要低，对于非门 G2 来说，这个电压是低电平，所以串入负向干扰电压不会出现逻辑错误。如果在这根线上串入了一个正向干扰电压，那么送入非门 G2 输入端的电压将会比 0.4V 高，对于非门 G2 来说，只要这个电压不高于 0.8V，对于非门 G2 来说，这个电压还能算是低电平，所以在这根连线上串入的正向干扰电压不能大于 0.8－0.4＝0.4（V）。

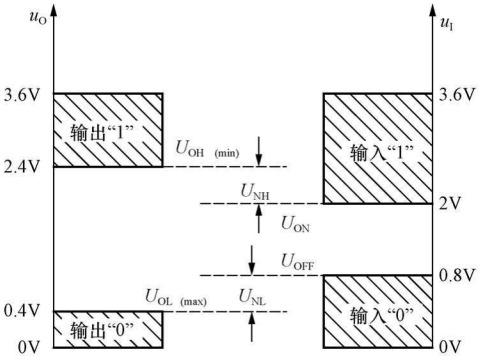


图 2-26　输入低电平噪声容限电压和输入高电平噪声容限电压

如果非门 G1 输出的低电平没有处于最大值，比 0.4V 小，比如为 0.2V，那么在这根连线上串入的正向干扰电压最大就可以达到 0.8－0.2＝0.6（V），而不会引起逻辑错误。

当一个电路在正常工作时，非门 G1 输出的低电平可能是 0V 和 0.4V 之间的任意值，这决定于不同的时刻电路的带负载情况。输入低电平噪声容限又称为最大允许正向干扰电压，由 G1 输出 0.4V 时的值决定。可由下式得到

$$\text{输入低电平噪声容限 } U_{NL} = U_{OFF} - U_{OL(max)} = U_{IL(max)} - U_{OL(max)}$$
$$= 0.8 - 0.4 = 0.4(V)$$

图 2-26 形象地给出了输入高电平噪声容限电压和输入低电平噪声容限电压的求法。

三、TTL 非门输入特性和从其上可以得出的参数

（一）标准生产工艺的 TTL 非门的输入特性曲线

输入特性曲线是研究输入电压与输入电流之间关系的曲线。图 2-27 是解释标准生产工

艺的 TTL 非门输入特性的电路图。输入电压 u_I 和输入电流 i_I 的参考方向如图 2-27 所示。图 2-28 是标准生产工艺的 TTL 非门输入特性曲线。

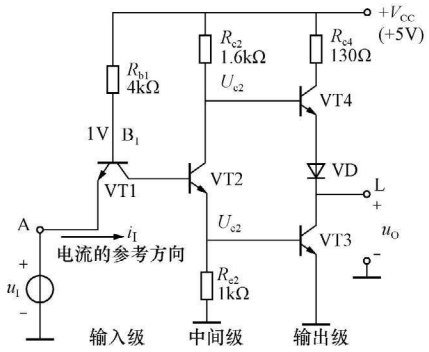


图 2-27 解释标准生产工艺的 TTL 非门输入特性的电路图

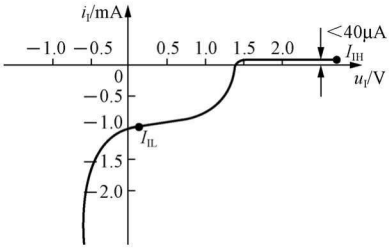


图 2-28 标准生产工艺的 TTL 非门输入特性曲线

（二）从标准生产工艺的 TTL 非门输入特性曲线上可以得出的参数

1. 输入低电平电流 I_{IL}

在图 2-29 中，将三极管 VT1 从电路中拿走，电路从三极管 VT1 的三个极断开，电阻 R_{b1} 所在的支路断开，没有电流流过电阻 R_{b1}，B_1 点电位为 5V，B_1 点和 A 点的电位差为 5−0.3=4.7（V），使三极管 VT1 的发射结导通，发射结导通后两端电压为 0.7V，从而使 B_1 点的电位箝位在 0.3+0.7=1（V）。1V 的电压降落在三极管 VT1 的集电结、三极管 VT2 的发射结、三极管 VT3 的发射结上，使这三个 PN 结都截止，所以最后使 B_1 点的电位钳位在 1V。

输入低电平电流 I_{IL} 是指当门电路的输入端接低电平时，从门电路输入端流出的电流。I_{IL} 的大小是 $\frac{V_{CC}-U_{B1}}{R_{b1}}=\frac{5-1}{4}=1$（mA），而输入低电平电流 I_{IL} 的参考方向如图 2-29 所示。所以在图 2-28 中 I_{IL} 为负值，在纵轴的负半轴上。

2. 输入高电平电流 I_{IH}

将三极管 VT1 从电路中拿走，电路从三极管 VT1 的三个极断开，电阻 R_{b1} 所在的支路断开，没有电流流过电阻 R_{b1}，B_1 点电位为 5V，B_1 点和 A 点的电位差为 5−3.6=1.4（V），使三极管 VT1 的发射结导通，如图 2-30 所示，发射结导通后两端电压为 0.7V，从而使 B_1 点的电位钳位在 3.6+0.7=4.3（V），4.3V 的电压降落在三极管 VT1 的集电结、三极管 VT2 的发射结、三极管 VT3 的发射结上，使这三个 PN 结导通，导通后的 PN 结两端电压为 0.7V，所以最后使 B_1 点的电位钳位在 0.7+0.7+0.7=2.1（V）。这时三极管 VT1 的发射结反偏、集电结正偏，称处于倒置状态。

输入高电平电流 I_{IH} 是指当门电路的输入端接高电平时，流入输入端的电流。I_{IH} 是三极管 VT1 的发射结反向饱和电流，它的值非常小，几乎为 0。输入高电平电流 I_{IH} 的参考方向如图 2-30 所示。所以在图 2-28 中 I_{IH} 为正值，在纵轴的正半轴上。

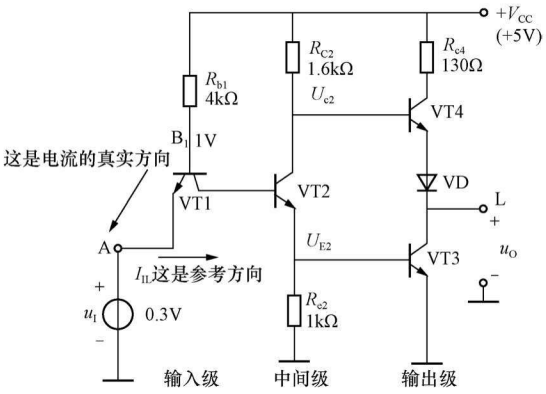


图 2-29　当输入为低电平时解释 TTL 非门输入特性的电路图

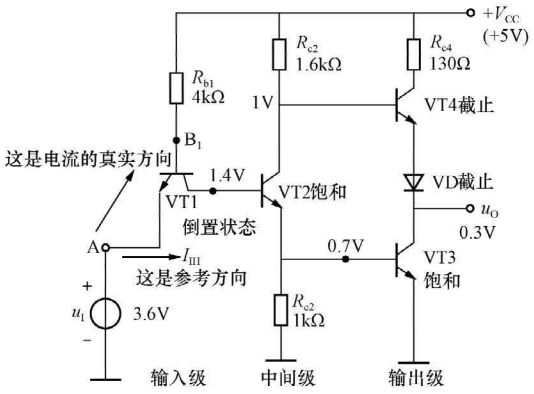


图 2-30　当输入为高电平时解释 TTL 非门输入特性的电路图

四、TTL 非门的输入负载特性和从其上可以得出的参数

（一）TTL 非门的输入负载特性

在实际使用中，经常会遇到在 TTL 非门的输入端经过电阻接地的情况，如图 2-31 所示。有电流流过电阻 R_i，在其两端产生电压 u_I，电压 u_I 在一定范围内随着电阻 R_i 的增大而增大。将反映两者关系的曲线，称做输入负载特性曲线，如图 2-32 所示。

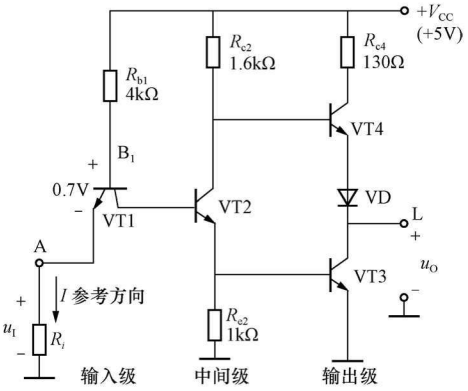


图 2-31　解释 TTL 非门的输入负载特性的电路图

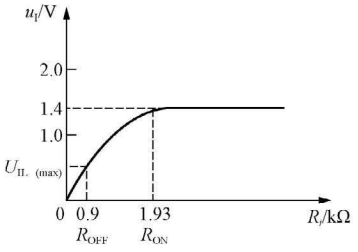


图 2-32　TTL 非门的输入负载特性曲线

当电阻 R_i 较小，在三极管 VT3 饱和之前，可以近似认为 u_I 和 R_i 的关系为

$$u_I = \frac{R_i}{R_i + R_{b1}}(V_{CC} - 0.7) \tag{2-1}$$

当 R_i 很小时 u_I 很小，相当于输入是低电平，此时三极管 VT3 截止，输出为高电平。当 R_i 很大时，比如等于 3kΩ，用上式计算可得 $u_I = 2.5$V，显然这是不可能的。因为当 R_i 增大，使 $u_I = 1.4$V 时，U_{B1} 就会升高到 2.1V，使 VT3 导通。此后即使再增加 R_i 的值，由于导通了 VT1 的集电结、VT2 的发射结、VT3 的发射结的箝位作用，U_{B1} 将保持在 2.1V，所以 u_I 的极限值为 1.4V，如图 2-32 所示。

（二）从 TTL 非门的输入负载特性上可以得出的参数

1. 关门电阻 R_{OFF}

当 R_i 的值使 $u_I=U_{IL(max)}$，即输入电压为输入低电平的上限时，R_i 定义为关门电阻 R_{OFF}。之所以叫关门电阻，是因为输出级的三极管 VT3 截止，处于关门状态。

对于 SN7404 芯片，从其内部电路可以看出 $R_{1b}=4k\Omega$，SN7404 芯片的输入电平的上限值为 0.8V，由式（2-1）可得

$$0.8=\frac{R_i}{R_i+R_{b1}}\times(5-0.7)$$

得出 $R_i=914\Omega$，所以 $R_{OFF}\approx900\Omega\approx0.9k\Omega$。

2. 开门电阻 R_{ON}

当 R_i 的值使 $u_I=1.4V$，U_{B1} 就会升高到 2.1V，使 VT3 饱和，这时的 R_i 定义为开门电阻。之所以叫开门电阻，是因为输出级的三极管 VT3 饱和，处于开门状态。

令 $u_I=1.4V$，由式（2-1）可得

$$1.4=\frac{R_i}{R_i+R_{b1}}\times(5-0.7)$$

由上式得出 $R_i=1.93k\Omega$。所以，为了保险起见，R_i 应该大于 $1.93k\Omega$，才能使输出级的三极管 VT3 可靠地进入饱和状态，输出低电平。

五、TTL 非门的输出特性和从其上可以得出的参数

（一）TTL 非门的低电平输出特性

1. TTL 非门的低电平输出特性曲线

在图 2-33 所示的电路中，VT3、VT4 是前级驱动门的输出级中的两个三极管，驱动门的输入级和中间级图中没有给出，两个 VT1 管分别是作为负载门的两个 TTL 非门的输入级中的三极管，即前级驱动门有两个同类门作为负载。当前级驱动门的输出为低电平时，送入负载门的输入信号也为低电平，因为 VT4 和 VD 处于截止状态，这时的图 2-33 电路可以简化为如图 2-34 所示电路。电流 I_{OL} 的实际方向就如图 2-34 所示的参考方向一样，这种将电流灌入前级门的负载称为灌电流负载。图 2-35 是 TTL 非门的低电平输出负载特性曲线。

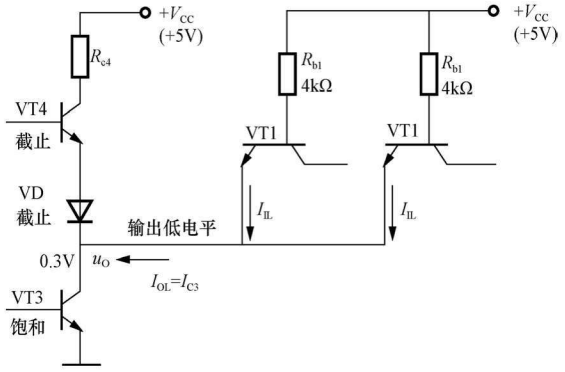


图 2-33 解释灌电流负载的电路图

2. 最大输出低电平电流 $I_{OL(max)}$

负载门输入端的电流加起来进入驱动门的三极管 VT3 的集电极，当负载门的个数增加，灌电流增大，会使三极管 VT3 脱离饱和。因为对于三极管 VT3 来说，相当于 I_{CS} 增加，而

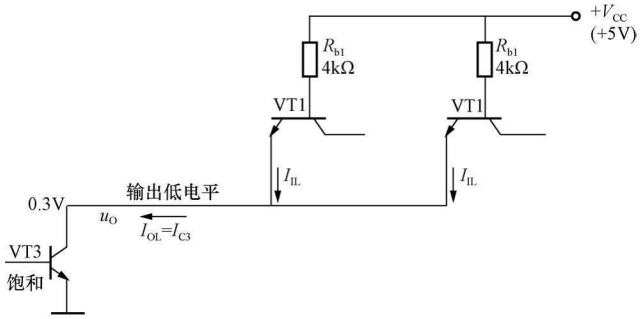


图 2-34　图 2-33 的简化电路

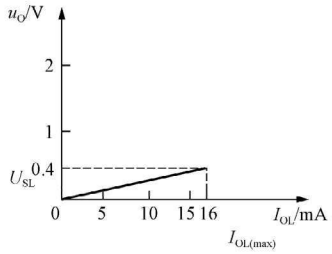


图 2-35　TTL 非门的低电平输出负载特性曲线

实际基极电流 i_B不变，这样当负载门比较少时，$i_B > I_{BS}$，满足处于饱和状态的条件。当负载门比较少时，集电极电流小，不能使三极管的集电极电流和基极电流满足 β 的关系，所以三极管处于饱和状态。当负载门比较多时，I_{CS}增加，$i_B < I_{BS}$，不满足处于饱和状态的条件，三极管 VT3 进入放大状态，其输出的低电平升高。因此，把驱动门对应标准输出低电平 U_{SL}时，灌入其输出端的电流定义为最大输出低电平电流 $I_{OL(max)}$（也叫做最大灌电流）。SN7404 芯片的最大灌电流为 16mA。

3. 输出低电平时的扇出系数

最大输出低电平电流与输入低电平电流 I_{IL}之比定义为输出低电平时的扇出系数，用 N_{OL}表示，即能带动的同类负载门的个数（要舍掉小数部分取整数）。其表达式为

$$N_{OL} = \frac{I_{OL(max)}}{I_{IL}}$$

（二）TTL 非门的高电平输出负载特性

1. TTL 非门的高电平输出负载特性曲线

在图 2-36 所示的电路中，前级驱动门有两个门作为负载。当前级驱动门的输出为高电平时，送入负载门的输入信号也为高电平。因为 VT3 处于截止状态，这时图 2-36 电路可以简化成图 2-37，电路中的电流的实际方向就如图 2-37 所示的参考方向一样，这种将电流从前级驱动门拉出的负载被称为拉电流负载。图 2-38 是 TTL 非门的高电平输出负载特性曲线。

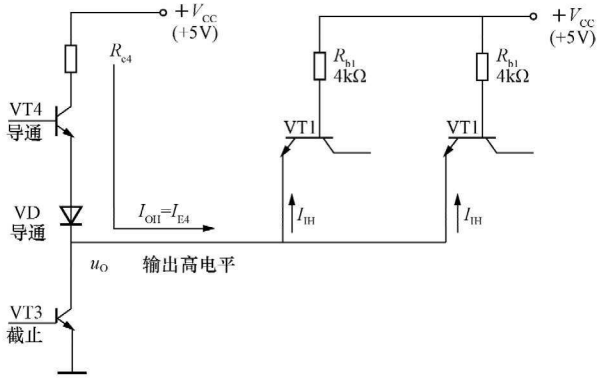


图 2-36　解释拉电流负载的电路图

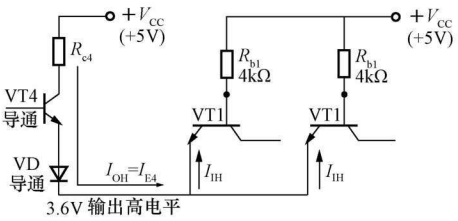


图 2-37 图 2-36 的简化电路

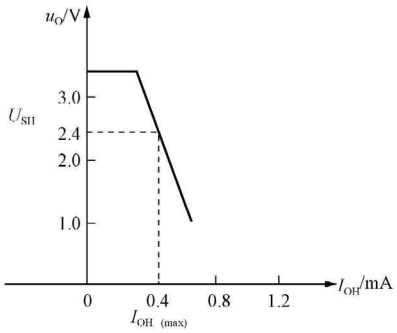


图 2-38 TTL 非门的高电平输出负载特性曲线

2. 最大输出高电平电流 $I_{OH(max)}$

当负载门的个数增加，拉电流增大时，R_{c4}上的压降增大，会使输出高电平降低。因此，把驱动门对应标准输出高电平 U_{SH}时拉出其输出端的电流定义为最大输出高电平电流 $I_{OH(max)}$（也叫做最大拉电流）。SN7404 芯片的 $I_{OH(max)}$ 为 0.4mA。

3. 输出高电平时的扇出系数

最大输出高电平电流与输入高电平电流 I_{IH}之比定义为输出高电平时的扇出系数，用 N_{OH}表示，即能带动的同类负载门的个数（要舍掉小数部分取整数）。其表达式为

$$N_{OH}=\frac{I_{OH(max)}}{I_{IH}}$$

（三）扇出系数

一般情况下，$N_{OL}\neq N_{OH}$，常取两者中的较小值作为门电路的扇出系数，用 N_O 表示。

六、传输延迟时间

如果把一个理想的矩形波加到 TTL 非门的输入端，在输出端得到的脉冲不但要比输入脉冲滞后，而且波形的边沿也要变坏，如图2-39所示。

传输延迟时间是指门电路在输入信号 u_I 波形的作用下，输出信号 u_O 波形相对于输入信号 u_I 波形在时间上的滞后和延迟。在图 2-39 中，输出信号 u_O 由高电平转为低电平的时间称为截止延迟时间，用 t_{PHL}表示；输出信号 u_O 由低电平转为高电平的时间称为导通延迟时间，用 t_{PLH}表示。通常 $t_{PLH}>t_{PHL}$。非门 SN7404 的 t_{PLH}的最大值为 22ns，典型值为 12ns；t_{PHL}的最大值为 15ns，典型值为 8ns。t_{PLH}和 t_{PHL}两者的平均值定义为平均传输延迟时间 t_{PD}，即

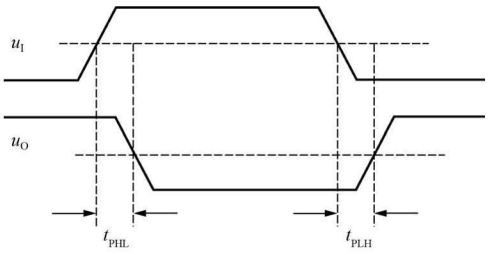


图 2-39 TTL 非门传输延迟时间

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

SN77404 的平均传输延迟时间 t_{PD}的最大值为 18.5ns，典型值为 10ns。t_{PD}越小，TTL 器件的速度越高。

七、功率损耗（P_D）和功耗—延时积（DP）

1. 功率损耗

功率损耗（功耗）是门电路的重要参数之一，功耗有静态功耗和动态功耗之分。静态功

耗指的是电路没有发生状态转换时的功耗，即门电路空载时电源总电流 I_{CC} 乘以电源电压 V_{CC}。静态功耗有两个参数，分别是空载导通功耗 P_{ON} 和空载截止功耗 P_{OFF}。

空载导通功耗 P_{ON}，指的是输出端开路、输入端全部悬空、门电路导通时，电源电流 I_{CCL} 与电源电压 V_{CC} 的乘积。

空载截止功耗 P_{OFF}，指的是输出端开路、输入端接地、门电路截止时，电源电流 I_{CCH} 与电源电压 V_{CC} 的乘积。

动态功耗指的是电路发生转换时的功耗，只发生在状态转换的瞬间或电路中有容性负载时。对于 TTL 门电路而言，静态功耗是主要的。

2. 功耗—延时积 DP

理想的数字电路要求高速度、低功耗，但要同时做到这两点却并不容易，高速数字电路往往要付出较大的功耗代价。为此采用功耗和传输延迟时间的乘积 DP 来全面评价门电路的性能，DP 越小，电路的综合性能越好。其表达式为

$$DP = t_{PD} \cdot P_D(\mathrm{J})$$

八、其他逻辑功能的 TTL 门电路

TTL 门电路除非门外，还有与非门、或非门、与或非门、集电极开路门、三态门、异或门等多种类型。

（一）TTL 正与非门

SN7400 是一种标准的 TTL 与非门器件，内部含有 4 个 2 输入端与非门，共有 14 个引脚。引脚排列图如图 2 - 40 所示。一片 SN7400 中有 4 个完全相同的与非门。图 2 - 41 是 SN7400 的内部电路图，可以看出它与图 2 - 20 所示的标准生产工艺的 TTL 反相器 SN7404 的电路图的差别只是在输入级的三极管 VT1 又增加了一个发射极，从而使这个电路有两个输入端。当这两个输入端中只要有一个为低电平时，三极管 VT1 的基极电位就被钳位在 1V，输出级的三极管 VT3 截止，输出高电平；当这两个输入端全部为高电平时，三极管 VT1 的基极电位就被箝位在 2.1V，输出级的三极管 VT3 饱和，输出低电平。如果使用正逻辑体制，输出与输入之间是与非的逻辑关系，三极管 VT1 的两个发射极形成了一个与的逻辑关系。

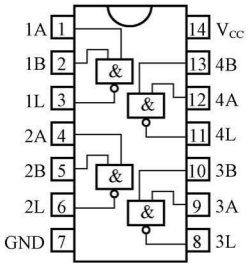


图 2 - 40　SN7400 的引脚排列图

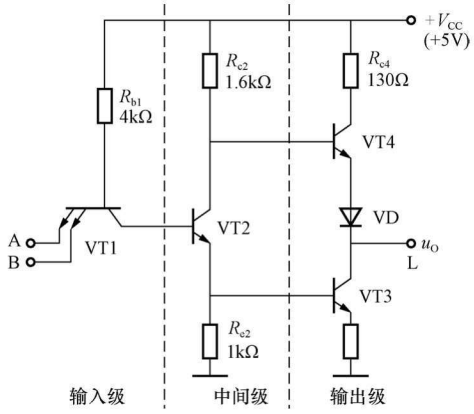


图 2 - 41　SN7400 的内部电路图

（二）TTL 正或非门

SN7402 是一种标准的 TTL 或非门器件，内部含有 4 个 2 输入端或非门，共有 14 个引

脚。一片 SN7402 中有 4 个完全相同的或非门。

除了上述两种 TTL 门电路外，还有 TTL 正与或非门和 TTL 异或门。

SN7451 是一种标准的 TTL 与或非门器件，内部含有 2 个 2—2 输入与或非门，共有 14 个引脚。

SN7486 是一种标准的 TTL 异或门器件，内部含有 4 个异或门，共有 14 个引脚。

九、集电极开路（OC）的 TTL 门电路

根据前面学到的知识，如果想实现两个信号 A、B 的与运算，就要将这两个信号加入一个 TTL 与门的输入端，在这个 TTL 与门的输出就可以得到这两个信号做与运算后的结果。也就是说，必须使用 TTL 与门电路才能实现与运算。

实际上，可以不使用 TTL 与门，将两个要做与运算的信号 A、B 直接相连，即使用一根导线就实现了与运算，这就是线与。图 2-42 为其示意图。实现线与运算的两个信号的出处是有讲究的，前面所讲的那些 TTL 门电路的输出是不能进行线与的，而只能使用特殊的 TTL 门电路，这个特殊的门电路就是集电极开路的 TTL 门电路。在此将以前所讲的 TTL 门电路称为普通的门电路，以和特殊的 TTL 门电路（集电极开路的 TTL 门电路）相区别。

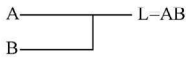


图 2-42 线与的逻辑电路图

（一）普通的 TTL 门电路输出端直接相连的后果

普通的 TTL 非门输出端直接并联逻辑电路图如图 2-43 所示，当两个门同时输出高电平时电路中没有从电源到地的低阻通路。当两个门同时输出低电平时电路中也没有从电源到地的低阻通路。当门 G1 输出高电平、门 G2 输出低电平时，图 2-43对应的内部电路图如图 2-44 所示。为了简化起见，在图 2-44 中，门 G1 的内部电路和门 G2 的内部电路只画出了输出级。在 V_{CC}与地之间形成一个低阻通路，从而产生超出正常工作电流的巨大电流，这一电流不仅抬高了输出的低电平，而且会因功耗过大损坏器件。因此普通 TTL 电路绝不允许将输出端直接并联使用。当门 G1 输出低电平、门 G2 输出高电平时的情况与上述情况类似，不再深入分析。

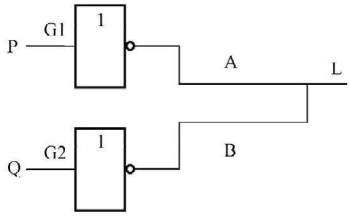


图 2-43 普通的 TTL 非门输出端直接并联逻辑电路图

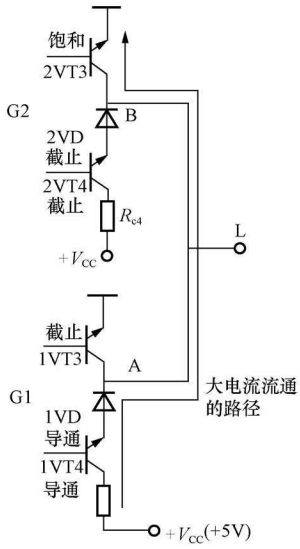


图 2-44 输出端直接并联的普通的 TTL 非门内部电路图

（二）集电极开路的 TTL 门电路

集电极开路的 TTL 与非门电路如图 2-45 所示。图中，与普通 TTL 与非门相比，去掉了输出级的三极管 VT4 和二极管 VD，三极管 VT3 的集电极开路（Open Collector，OC），集电极开路的 TTL 门电路的名字也就是由此而来。集电极开路的 TTL 与非门的国标逻辑

符号如图 2-46 所示，“◇”表示集电极开路的输出端。

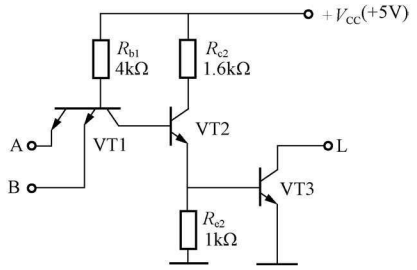


图 2-45　集电极开路的 TTL 与非门电路图

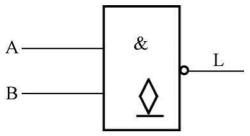


图 2-46　集电极开路的 TTL 与非门的国标逻辑符号

（三）集电极开路的 TTL 门电路可以实现线与运算

集电极开路的 TTL 门电路（简称 OC 门）要正常工作，必须给 VT3 外的集电极接电阻和电源。只要外接电阻值和电源电压值选择得当，就能实现多个 OC 门的输出端的与运算。两个 OC 门的输出端实现与运算的电路图如图 2-47 所示。其中，G1 实现 M、N 之间的与非逻辑运算，结果为 A；G2 实现 O、P 之间的与非逻辑运算，结果为 B；G1 和 G2 输出端连在一起，实现与运算，L＝AB。只要 G1 的输出或 G2 的输出中有一个是低电平，输出 L 就是低电平；只有 G1 的输出和 G2 的输出都是高电平，输出 L 才是高电平。在使用正逻辑体制的前提下，其逻辑表达式为

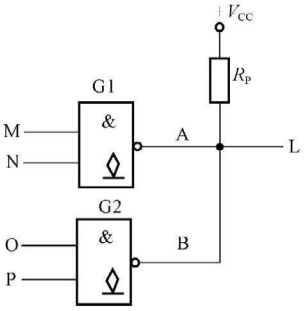


图 2-47　两个 OC 门的输出端实现与运算的电路图

$$L = AB = \overline{MN} \cdot \overline{OP} = \overline{MN + OP}$$

从逻辑表达式可以看出，两个 OC 门的线与连接可以实现与或非的逻辑运算。

（四）外接上拉电阻值的计算方法

假设要将 n 个 OC 门的输出端做线与连接，即直接相连，负载是 m 个普通或特殊的 TTL 与非门的输入端，如图 2-48 和图 2-49 所示。

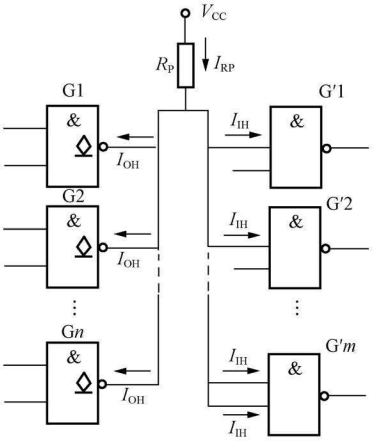


图 2-48　OC 门上拉电阻最大值的计算

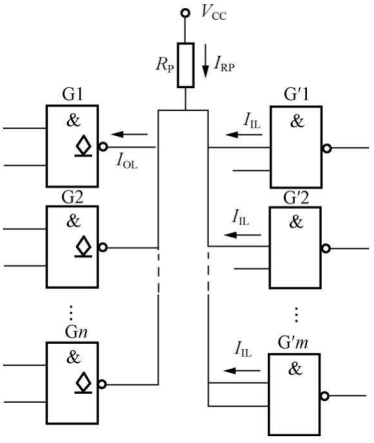


图 2-49　OC 门上拉电阻最小值的计算

在图 2-48 中，当所有 n 个 OC 门都截止，输出均为高电平，线与后仍然为高电平，送给后面的 m 个普通或特殊的 TTL 与非门的输入端，观察 m 个 TTL 与非门的输入端，共有 $m+1$ 个输入端。图中所标的电流的方向是电流的实际方向，I_{OH} 是 OC 门输出三极管 VT3 截止时的漏电流，I_{IH} 是负载门每个输入端的输入高电平电流。此处要特别注意，图中右下角的门 $G'm$ 的两个输入端同时和线与后的信号相连，这时每个输入端都有输入高电平电流。流过上拉电阻 R_p 的电流的和为 $nI_{OH}+(m+1)I_{IH}$，为保证 OC 门输出的高电平不低于高电平的下限 $U_{OH(min)}$，R_p 的值不能选得太大，即要保证 $V_{CC}-I_{RP}R_P \geqslant U_{OH(min)}$，所以 R_p 的最大值为

$$R_{p(max)}=\frac{V_{CC}-U_{OH(min)}}{nI_{OH}+(m+1)I_{IH}}$$

图 2-49 中，当任何一个 OC 门的输出级三极管 VT3 处于饱和状态时，输出低电平，线与后的信号为低电平，送给后面的负载门，而且，应当确保在最不利的情况下，即只有一个 OC 门的输出级三极管 VT3 处于饱和状态。这时所有负载电流全部流入唯一的那个处于饱和状态的输出级三极管 VT3 的集电极，输出的低电平要保证低于输出低电平的上限 $U_{OL(max)}$。流过上拉电阻 R_p 的电流的和为 $I_{OL(max)}-mI_{IL}$，因而 R_p 值不可太小，即要保证 $V_{CC}-I_{RP}R_P \leqslant U_{OL(max)}$，由此得出计算 R_p 最小值的公式为

$$R_{p(min)}=\frac{V_{CC}-U_{OL(max)}}{I_{OL(max)}-mI_{IL}}$$

式中：$I_{OL(max)}$ 为 OC 门的最大灌电流；I_{IL} 为负载门输入短路电流。

上拉电阻 R_p 的取值应在 $R_{p(min)}$ 和 $R_{p(max)}$ 两者之间。

OC 结构可以做驱动器使用，如驱动高电压、大电流负载的门电路。除与非门外，或非门、与或非门、异或门等都可以做成集电极开路的结构，所接上拉电阻的计算方法也相同。

特别提醒读者注意的是，在计算上拉电阻最小值和最大值时，应使驱动门的输出高、低电平满足其要求，而不是以输入门的输入高、低电平满足其要求为标准。因为，还要考虑连接输入和输出信号的导线上应该有一定的抗干扰能力。

十、TTL 三态输出门电路

利用普通 OC 门虽然可以实现线与功能，但外接上拉电阻的选择要受到一定的限制而不能取得太小，外接上拉电阻越大工作速度就越慢；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并且能做线与连接，人们又开发了一种三态 TTL 门，它的输出除了具有一般 TTL 门的两种状态，即输出电阻小的高低电平外，还具有高输出电阻的第三态，称为高阻态。

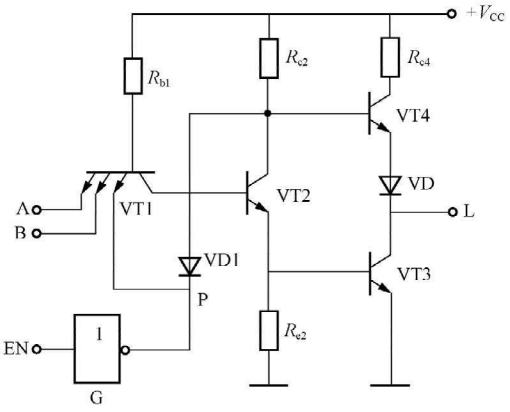


图 2-50 TTL 三态输出的与非门电路（控制端低电平有效）

在普通 TTL 门电路的基础上增加控制端和控制电路就可以构成 TTL 三态输出门。

（一）TTL 三态输出的与非门电路的工作原理

图 2-50 是控制端为低电平有效的 TTL 三态输出的与非门电路，图 2-51 是其逻辑符号。图中，当控制端 EN 为低电平时，P 点为高电

平，二极管 VD1 截止，其余电路处于正常工作状态，与一般的二输入 TTL 与非门一样，输出 $L=\overline{A \cdot B}$，由 A、B 的状态决定输出是高电平还是低电平。当 EN 为高电平时，P 点为低电平，使三极管 VT1 的基极电位箝位在 1V，使三极管 VT3 处于截止状态。P 点为低电平还使二极管 VD1 导通，使三极管 VT2 的集电极电位箝位在 1V，使三极管 VT4 和二极管 VD 均处于截止状态。总之，三极管 VT3、三极管 VT4 和二极管 VD 均处于截止状态，这样的状态称为高阻状态。高阻状态可以认为是与其他电路没有电的联系的一种状态，相当于断路；从输出电阻的角度看，输出电阻非常大，称为高阻。

图 2-50 的电路输出端有三种可能的状态：高阻、高电平、低电平，其真值表如表2-10所示。

表 2-10　图 2-50 所示三态门的真值表

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

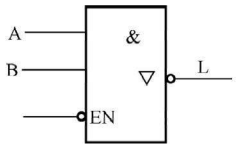


图 2-51　TTL 三态输出的与非门电路（控制端低电平有效）的逻辑符号

图 2-52 是控制端为高电平有效的 TTL 三态输出的与非门电路，图 2-53 是其国标逻辑符号。在图 2-52 的电路中，当控制端 EN 为高电平时，与一般的二输入 TTL 与非门一样，输出 $L=\overline{AB}$。当 EN 为低电平时，P 点为低电平，使三极管 VT1 的基极电位钳位在 1V，使三极管 VT3 处于截止状态，电路处于高阻状态。

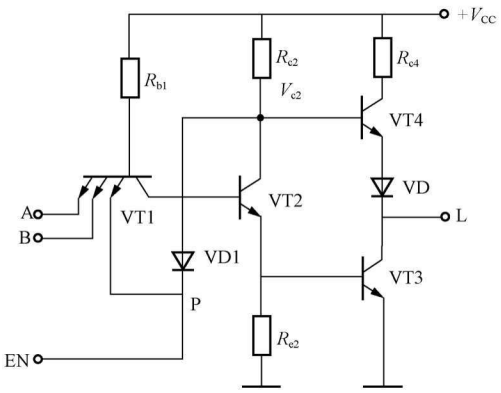


图 2-52　TTL 三态输出的与非门电路（控制端高电平有效）

图 2-52 的电路输出端有三种可能的状态：高阻、高电平、低电平，其真值表如表 2-11 所示。

表 2-11　图 2-52 所示三态门的真值表

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

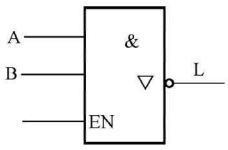


图 2-53　TTL 三态输出的与非门电路（控制端高电平有效）的国标逻辑符号

（二）TTL 三态输出门电路的应用

在某数字系统中，有 A、B、C、D、E、F 六个子部件，各自完成固定的功能，必要的

时候在这六个子部件之间两两都有可能需要交换信息。可以在这六个子部件两两之间连接一根通信的导线，如图 2-54 所示。当数字系统的子部件增多时，两两之间建立连接需要的连接线数目就会非常快地增长，使数字系统的复杂度增加。所以，现代数字系统都使用总线（BUS）技术来建立子部件之间的联系，总线技术大大减少了部件之间连线的数目。各个子部件分时使用总线。

总线是系统中各个子部件共同使用的连接线。图 2-55 是六个子部件通过总线建立连接的示意图。任意两个子部件之间要想交换信息，那么这两个子部件就占有对总线的控制权，当信息交换完毕，就放弃对总线的占有。如何从电路的角度来理解“放弃对总线的占有”这句话的意思呢？就是让这个子部件的输出处于高阻状态，与总线之间断绝电的联系。所以在使用总线技术的数字系统中，各子部件的输出必须具有三态控制功能。

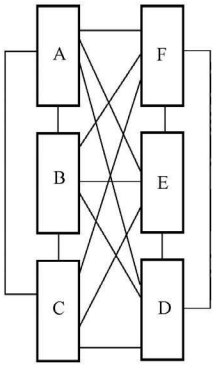


图 2-54　六个子部件两两之间建立连接的示意图

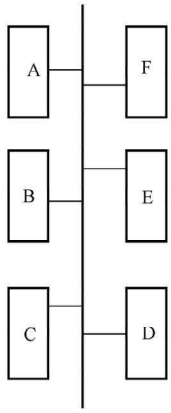


图 2-55　六个子部件通过总线建立连接的示意图

三态门可以实现数据的单向传输。在图 2-56 中，若干个三态门连接在同一根导线（总线）上，任意时刻只有一个三态门处于工作状态，其他所有的三态门均处于高阻状态。处于工作状态的三态门占有对总线的控制权。

三态门还可以实现数据的双向传输。在图 2-57 中，EN＝1 时，G1 工作，G2 为高阻态，数据 IN 经 G1 反相后送到总线；而当 EN＝0 时，G2 工作而 G1 为高阻态，总线上的数据经 G2 反相后从$\overline{OUT}$端送出，实现了数据向总线的送出和从总线的流入。

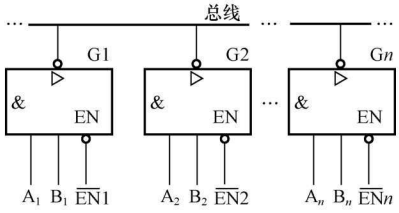


图 2-56　单向总线

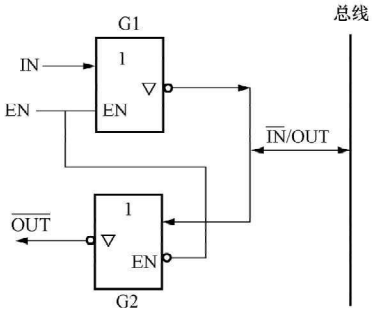


图 2-57　三态门实现数据双向传输

十一、其他生产工艺类型的 TTL 门电路

为了提高门电路的各种性能指标，除了标准生产工艺的 TTL 门电路以外，还有其他各种生产工艺类型的门电路，如表 2-12 所示。

表 2-12　TTL 门电路的工艺类型

符　号	说　　明	符　号	说　　明
	标准 TTL 门电路	ALS	先进的低功耗肖特基 TTL 门电路
LS	低功耗肖特基 TTL 门电路	AS	先进的肖特基 TTL 门电路
S	肖特基 TTL 门电路	H	高速 TTL 门电路
L	低功耗 TTL 门电路		

例如，芯片 SN7404，其中 SN 表示是美国德州仪器公司的产品；74 表示民用产品；04 表示芯片功能编号，为非门电路。

TTL 门电路除 74 系列外，还有 54 系列。74 系列适合工作于 0～70℃温度范围内，供电电压为 4.75～5.25V，为民用系列；54 系列，适合工作于－55～＋125℃温度范围内，供电电压为 4.5～5.5V，为军用系列。无论 54 系列还是 74 系列，编号相同的集成电路引脚排列完全相同。

在实际应用中，TTL 门电路相继发展了在速度、功耗和抗干扰能力等方面的改进型 TTL 门电路，如 H 系列、S 系列、LS 系列、ALS 系列等，其中一些已成为目前 TTL 门电路主要的应用系列。

第四节　射极耦合逻辑门电路

标准 TTL 门电路中部分晶体管工作在饱和区，电路的工作速度受到了限制。为了进一步提高门电路的工作速度，只有将电路中的晶体管工作状态由饱和改为非饱和，使晶体管不进入饱和区，才能从根本上提高电路的工作速度。射极耦合逻辑电路（Emitter Coupled Logic，ECL）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在 2ns 以下，是目前唯一能提供亚毫微秒开关时间的实用电路，主要应用于每秒运算百万次以上的大型高速计算机、数字通信系统、高精度测量设备和频率合成器等方面。

图 2-58 为一简化的 ECL 门电路图。图中，硅三极管 VT1、VT2、VT3 的发射极直接耦合，所以称为射极耦合逻辑电路。ECL 门电路采用了差动放大电路作为输入级。VT3 的基极接固定参考电压（$U_{REF}=1V$）。输入信号 A、B 分别接 VT1、VT2 的基极，输入信号的高电平为 1.5V，低电平为 0.5V，摆幅为 1V。输出信号由 VT1、VT2 的集电极或 VT3 的集电极取得，可分别获得两种不同的逻辑运算。输出信号的高电平为 6V，低电平为 5V，摆幅为 1V。

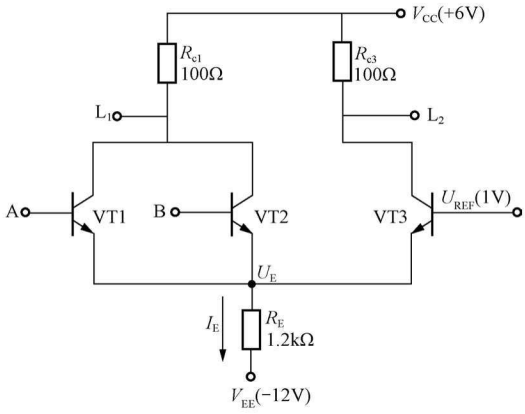


图 2-58　简化的 ECL 门电路

VT3 的基极接固定参考电位 1V，正好是输入信号的高电平 1.5V 和低电平 0.5V 的中间值。输入信号正是相对于基准电压 1V 做＋0.5V 和－0.5V 的电压变化，使 VT1 和 VT3 导通和关闭的。由此可见，基准电压的稳定性很重要。在图 2-58 中，只是标注了基准电位，没有给出具体的电路。

1. 当输入端 A、B 都接低电平时（$U_A=U_B=0.5V$）

由于有 $U_{REF}=1V$ 的存在，比输入端 A、B 的低电平 0.5V 电位高，因此 VT3 首先导通，箝制发射极电位于 $U_E=U_{REF}-U_{BE3}\approx 0.3V$。由于 VT1、VT2 的输入电平为 0.5V，虽然此时基极电位高于发射极电位 0.2V，但 VT1、VT2 作为硅管仍不能导通，处于截止状态。I_E 全部由 VT3 提供，其值为

$$I_E=[U_E-(-U_{EE})]/R_e=\frac{0.3+12}{1.2}\approx 10(\text{mA})$$

L_2 输出端电位为

$$U_{L2}=V_{CC}-I_E R_{c3}=6-10\times 0.1=5(\text{V})(\text{输出低电平})$$

L_1 输出端电位等于 V_{CC}，为 6V（输出高电平）。此时 VT3 虽然导通，但由于 $U_{B3}=U_{REF}=1V$，而 $U_{L2}=5V$，VT3 处于放大状态，未达到饱和。

2. 当输入端 A、B 中有一个接高电平（$U_A=1.5V$，$U_B=0.5V$ 或 $U_A=0.5V$，$U_B=1.5V$）

设高电平从 A 端输入，其值为 1.5V，低电平从 B 端输入，其值为 0.5V。由于 $U_A>U_{REF}$，VT1 首先导通，这时有

$$U_E=U_A-U_{BE1}=1.5-0.7=0.8(\text{V})$$

对 VT3 而言，基极电位比发射极电位只高 0.2V，硅管不能导通，处于截止状态。I_E 由 VT1 提供，其值为

$$I_E=\frac{0.8+12}{1.2}=10.6(\text{mA})$$

L1 输出端电位为

$$U_{L1}=V_{CC}-I_E R_{c1}=6-10.6\times 0.1\approx 5(\text{V})(\text{输出低电平})$$

L_2 输出端电位等于 V_{CC}，为 6V（输出高电平）。此时 VT1 处于放大状态，未达到饱和。

3. 当输入端 A、B 两个均接高电平（$U_A=1.5V$，$U_B=1.5V$）

这种情况与输入端 A、B 中有一个接高电平类似，此处不再重复叙述。

从以上分析可以看出，不论 VT3 导通还是 VT1 导通，流过发射极电阻的电流 I_E 变化不大，可近似地看作为一恒流源。当输入端一侧晶体管（VT1 或 VT2）导通时，电流 I_E 流向左边，当三极管 VT3 导通时，电流 I_E 流向右边，好像发射极电路中有一个开关将恒流源 I_E 接到左边或右边，所以 ECL 电路又称为电流开关型电路或电流模式逻辑电路（Current Mode Logic，CML）。

电压功能表如表 2-13 所示。使用正逻辑体制，规定输入信号高电平 1.5V 表示逻辑 1，输入信号低电平 0.5V 表示 0；输出信号高电平 6V 表示逻辑 1，输出信号低电平 5V 表示 0，得到表 2-14 的真值表。

由真值表得到输出信号与输入信号的逻辑关系式为

$$L_1=\overline{A+B}\qquad(\text{或非逻辑})$$

$$L_2=A+B\qquad(\text{或逻辑})$$

表 2-13　图 2-58 电路的电压功能表

U_A	U_B	U_{L1}	U_{L2}
0.5V	0.5V	6V	5V
0.5V	1.5V	5V	6V
1.5V	0.5V	5V	6V
1.5V	1.5V	5V	6V

表 2-14　正逻辑体制下表 2-13 的真值表

A	B	L_1	L_2
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

图 2-58 所示电路存在的问题是，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）。ECL 电流开关电路，实质是一个差动放大器，它虽然能够完成或非、或逻辑运算，但是两个输出端的高、低电平都不是 ECL 门电路的典型高、低电平，可采用加电压跟随器等移动电平值的措施来解决。

第五节　CMOS 逻辑门电路

一、CMOS 反相器

（一）CMOS 反相器电路结构

图 2-59 是 CMOS 反相器的电路图，其中 VTN 为增强型 NMOS 管，VTP 为增强型 PMOS 管，两管电气特性完全对称。为使衬底与漏源之间的 PN 结始终处于反偏，NMOS 管的衬底总是接到电路的最低电位点，PMOS 管的衬底总是接到电路的最高电位点。两管的栅极相连为输入端 A，漏极相连为输出端 L。VTP 的源极 S2 接 $+V_{DD}$，而 VTN 的源极 S1 接地。NMOS 管的栅源开启电压 $U_{VTN}>0$，PMOS 管的栅源开启电压 $U_{VTP}<0$。为了使 CMOS 电路能正常工作，要求电源电压 $V_{DD}>(U_{VTN}+|U_{VTP}|)$，$V_{DD}$ 取值范围较大，可在 3～18V 之间。

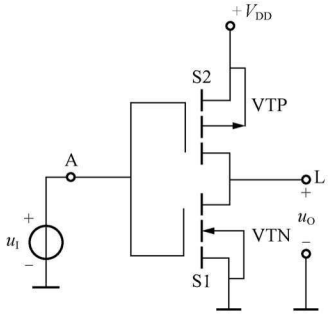


图 2-59　CMOS 反相器的电路图

当输入为低电平，即 $U_A=0V$ 时，$U_{GS1}<U_{VTN}$，VTN 截止，内阻很大；$|U_{GS2}|>|U_{VTP}|$，VTP 导通，内阻很小，电路中电流近似为零（忽略 VTN 的截止漏电流），输出为高电平，$U_L\approx V_{DD}$；同理，当输入为高电平，$U_A=V_{DD}$ 时，VTN 导通，VTP 截止，输出为低电平，$U_L\approx 0V$。可见电路实现非逻辑功能，即 $L=\overline{A}$。

对于 CMOS 反相器，不论输入高电平还是低电平，VTN 和 VTP 总是一个工作在导通状态，另一个工作在截止状态，即两管工作在互补状态。因此，该电路称为互补型 MOS 电路。

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

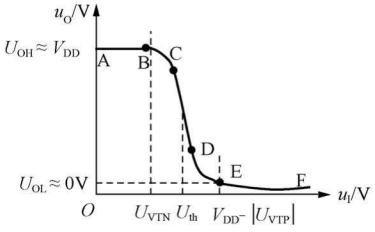


图 2-60　CMOS 反相器的电压传输特性曲线

（二）CMOS 反相器的电压传输特性

CMOS 反相器的电压传输特性曲线如图 2-60 所示，由

五段曲线组成：

（1）AB 段——输入电压 u_I 小于 VTN 开启电压 U_{VTN}，VTN 截止，u_I-V_{DD}小于 U_{VTP}，VTP 导通，$u_O=U_{OH}\approx V_{DD}$。

（2）BC 段——输入电压 u_I 大于 VTN 开启电压 U_{VTN}，VTN 导通，u_I-V_{DD}小于 U_{VTP}，VTP 导通。由于 u_I 不够大，VTN 沟道电阻远大于 VTP 沟道电阻，输出电压下降。

（3）CD 段——输入电压 u_I 大于 VTN 开启电压 U_{VTN}，VTN 导通，u_I-V_{DD}小于 U_{VTP}，VTP 导通，两管栅源电压绝对值近似，VTN 沟道电阻和 VTP 沟道电阻的比值发生显著变化，引起输出电压急剧下降。

（4）DE 段——输入电压 u_I 大于 VTN 开启电压 U_{VTN}，VTN 导通，u_I-V_{DD}小于 U_{VTP}，VTP 导通。由于 u_I 比较大，VTN 沟道电阻远小于 VTP 沟道电阻，输出电压进一步下降。

（5）EF 段——输入电压 u_I 大于 VTN 开启电压 U_{VTN}，VTN 导通，u_I-V_{DD}大于 U_{VTP}，VTP 截止。输出 $u_O=U_{OL}\approx 0V$。

阈值电压 U_{th}是输入低电平和输入高电平的分界点。

由 CMOS 反相器的电压传输特性可知，反相器状态转换时，电压变化率比较大，阈值电压 U_{th}较高（$=V_{DD}/2$）。因而，CMOS 器件有较高的抗干扰容限。

根据电压传输特性分析可以知道，输入电压在 $U_{VTN}<u_I<V_{DD}-|U_{VTP}|$ 范围内，即电压传输特性 BE 段，工作管和负载管同时导通，因此形成漏极电流 i_D。电压传输特性 CD 段非常陡直，且在 $u_I=U_{th}=V_{DD}/2$ 时两管沟道电阻之和最小，所以，此时漏极电流 i_D 达到最大值。

二、其他的 CMOS 门电路

（一）CMOS 传输门

传输门是 CMOS 门电路特有的一种门电路，传输门既可以传输数字信号，也可以传输模拟信号，在信号的传输、选择和分配中有广泛的应用。图 2-61 给出了 CMOS 传输门的组成和逻辑符号。

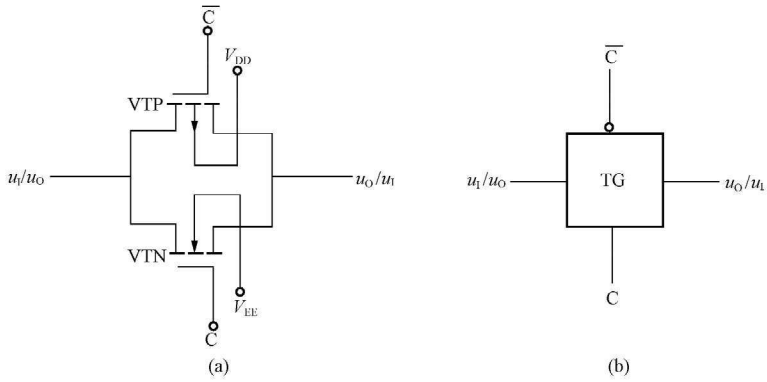


图 2-61 CMOS 传输门

（a）传输门结构；（b）传输门逻辑符号

传输门由两个互补的增强型 MOS 管并联连接构成，两管的栅极分别接互补的控制逻辑信号 C 和$\overline{C}$，两管的衬底接最低电平（V_{EE}）和最高电平（V_{DD}）。因为当场效应管的衬底内部未连接时，漏极（d）和源极（s）可以互换，所以，既可以将漏极作为输入端，也可以将

源极作为输入端，即传输门的信号传输方向是双向的。

为便于分析，设 $V_{DD}=+5V$，$V_{EE}=-5V$，$u_I=-5\sim+5V$，C=0 的电平值为 $-5V$，C=1 的电平值为 $+5V$，VTN 管开启电压 $U_{VTN}=3V$，VTP 管开启电压 $U_{VTP}=-3V$。

在 C=0 时，VTN 栅极接 $-5V$ 电压，VTP 栅极接 $+5V$ 电压。在输入信号范围内，VTN 的栅源电压 $U_{GS}(=-5-u_I)\leqslant 0$，VTN 截止；VTP 的栅源电压 $U_{GS}(=+5-u_I)\geqslant 0$，VTP 也截止。也就是说，C=0 时，传输门关闭，输入和输出之间呈现出高阻抗状态，是两个场效应管截止时的漏源电阻，输入和输出之间不能进行信号传输。

在 C=1 时，VTN 栅极接 $+5V$ 电压，VTP 栅极接 $-5V$ 电压。在输入信号不同范围内，VTN 和 VTP 的工作状态不尽相同。

$u_I=-5\sim-2V$ 时，VTN 管的 $U_{GS}=+5-u_I=10\sim7V$ 可见，$U_{GS}>U_{VTN}$，VTN 导通；VTP 管的 $U_{GS}=-5-u_I=0\sim-3V$，$U_{GS}\geqslant U_{VTP}$，VTP 截止。

$u_I=-2\sim+2V$ 时，VTN 管的 $U_{GS}\geqslant U_{VTN}$，VTN 导通；VTP 管的 $U_{GS}\leqslant U_{VTP}$，VTP 导通。

$u_I=+2\sim+5V$ 时，VTN 管的 $U_{GS}\leqslant U_{VTN}$，VTN 截止；VTP 管的 $U_{GS}<U_{VTP}$，VTP 导通。

由此可见，在整个输入电压范围 $-5\sim+5V$ 内，至少有一个场效应管是导通的。场效应管导通，漏源间的沟道导通电阻（R_{ON}）小于 1kΩ，典型值为 80Ω，漏极和源极之间相当于短路，输出等于输入，即 C=1 时，传输门打开，$u_O=u_I$。

（二）CMOS 双向模拟开关

用传输门和反相器可构成双向模拟开关，其原理图和逻辑符号见图 2-62。

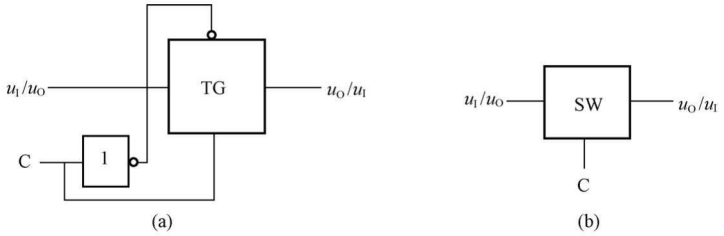


图 2-62　CMOS 双向模拟开关

（a）CMOS 双向模拟开关原理图；（b）CMOS 双向模拟开关的逻辑符号

【例 2-5】 分析图 2-63 所示电路，说明该电路逻辑功能。

解　B=1 时，TG1 导通，TG2 关闭，输入 A 反相后输出；B=0 时，TG1 关闭，TG2 导通，F 反相后由 A 输出。

因此，这是一可控双向数据传输电路。当 B=0 时，$A=\overline{F}$；当 B=1 时，$F=\overline{A}$。

（三）CMOS 与非门

二输入 CMOS 与非门如图 2-64 所示，图中两个 N 沟道工作管 VTN1、VTN2 串联连接，其衬底接电位最低点（地），两个 P 沟道负载管 VTP1、VTP2 并联连接，其衬底接电位最高点（V_{DD}）。工作管 VTN1、VTN2 串联情况下，只有两管都导通，工作管支路才能导通，即 A 和 B 全为 1，工作管支路导通；A 和 B 中有 0 时，工作管支路截止。负载管 VTP1、VTP2 并联连接，A 和 B 中有 0，则负载管支路就导通；A 和 B 全为 1，负载管支路才截止。工作管支路截止负载管支路导通，输出高电平 1，实现“见 0 为 1”逻辑功能。工作管支路导通负载管支路截止，输出低电平 0，实现“全 1 为 0”逻辑功能。可见该电路实现了与非逻辑，即 $L=\overline{AB}$。因此，这是一与非门电路。要实现多输入与非只要相应增加串

联工作管和并联负载管数目，但串联工作管数目越多输出低电平会越高，一般串联工作管数目有限制，最多为 4 个。

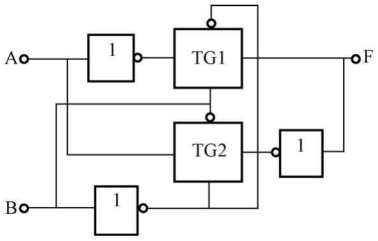


图 2-63 ［例 2-5］的电路图

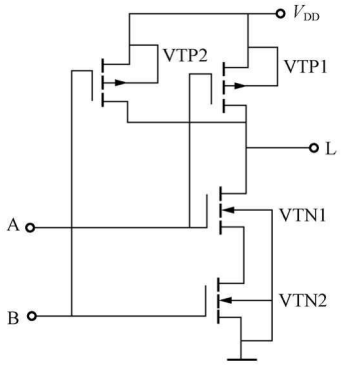


图 2-64 CMOS 与非门

从电路结构上来说，工作管和负载管是成对出现在 CMOS 电路中，且工作管支路的状态与负载管支路的状态相反，因此，工作管支路的连接形式与负载管支路的连接形式也相反（互补），即工作管串联，负载管并联。

（四）CMOS 或非门

图 2-65 为 CMOS 两输入或非门电路。与 CMOS 与非门相反，CMOS 或非门由两个串联的 PMOS 管 VTP1、VTP2 构成负载部分，而两个并联的 NMOS 管 VTN1、VTN2 构成驱动部分。当输入 A 和 B 都为低电平时，VTN1 和 VTN2 截止，VTP1 和 VTP2 导通，输出高电平；当输入 A 和 B 有一个或一个以上为高电平时，与该高电平相连的 NMOS 管导通，输出低电平。所以该电路实现或非逻辑，即 $L=\overline{A+B}$。

（五）CMOS 异或门

CMOS 异或门如图 2-66 所示。它由三个 CMOS 反相器和一个 CMOS 传输门 TG 组成。传输门 TG 受 A 控制，当 A=0 时 TG 断开，当 A=1 时 TG 导通。

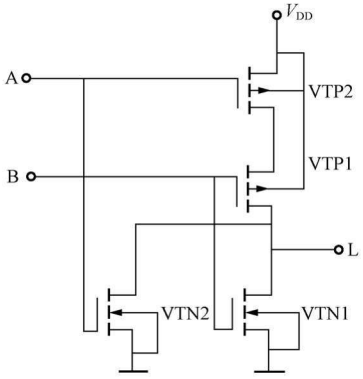


图 2-65 CMOS 或非门

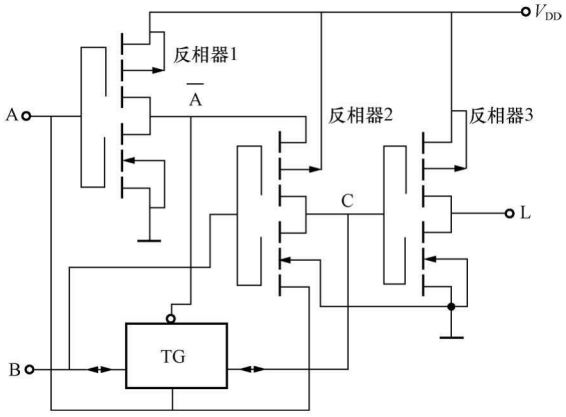


图 2-66 CMOS 异或门

（1）当输入信号 A=B=0 时，因为 A=0，所以 TG 断开，反相器 2 工作，此时 $C=\overline{B}=1$，$L=\overline{C}=0$；

（2）当输入信号 A＝B＝1 时，因为 A＝1，所以 TG 导通，C＝B＝1，在 A 和$\overline{A}$的控制下，反相器 2 的两只 MOS 管都截止，反相器 2 输出处于高阻态，此时 $L=\overline{C}=\overline{B}=0$。

（3）当输入信号 A＝1、B＝0 时，因为 A＝1，所以 TG 导通，C＝B＝0，在 A 和$\overline{A}$的控制下，反相器 2 的两只 MOS 管都截止，反相器 2 输出处于高阻态，$L=\overline{C}=\overline{B}=1$。

（4）当输入信号 A＝0、B＝1 时，因为 A＝0，所以 TG 断开，反相器 2 工作，$C=\overline{B}=0$，$L=\overline{C}=1$。

综上所述，可见此电路为异或逻辑，即 $L=A\oplus B=\overline{A}B+A\overline{B}$。

在实际使用中，常常在上述基本门电路的输入端、输出端增加一级反相器，构成带缓冲级的门电路。加入缓冲级后，输出电阻，输出的高、低电平和电压的传输特性转折区都不受输入端状态的影响，而且其性能也有极大提高。当然，这种缓冲级的增加会改变逻辑电路的功能，如在或非门上加入缓冲级，不是组成带缓冲级的或非门，而是组成带缓冲级的与非门，但采用类似的办法不难得到各种逻辑功能的带缓冲级的门电路。

三、双极型 CMOS 门电路

双极型 CMOS（Bipolar CMOS，Bi-CMOS）电路的特点是用 CMOS 电路实现逻辑门基本功能，用带负载能力强的 TTL 电路做输出级，具有 CMOS 功耗低、TTL 负载能力强、传输延迟时间短、输出电阻低的特性。

（一）Bi-CMOS 非门

Bi-CMOS 非门电路图如图 2－67 所示。图中，VT1 和 VT3 是驱动管，VT5 和 VT6 是双极型输出管，VT2 和 VT4 分别是输出管 VT5 和 VT6 基极的下拉负载管，形成有源负载电路。C_L 为负载电容。当输入高电平时，VT2、VT3 和 VT6 导通，VT1、VT4 和 VT5 截止，输出低电平。当输入低电平时，VT1、VT4 和 VT5 导通，VT2、VT3 和 VT6 截止，输出高电平。在正逻辑体制下输出与输入之间实现非逻辑。由于 VT5 和 VT6 的导通内阻很小，对负载电容的充放电速度很快，故电路的传输延迟时间很短，可以达到 1ns 以下。

（二）Bi-CMOS 与非门

Bi-CMOS 与非门电路图如图 2－68 所示。该电路与非门电路相比，输出方式是一样的，只是 CMOS 逻辑功能的连接不同。

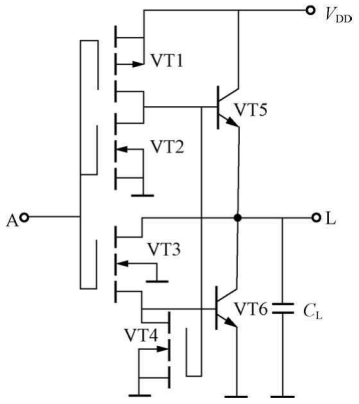


图 2－67　Bi-CMOS 非门电路图

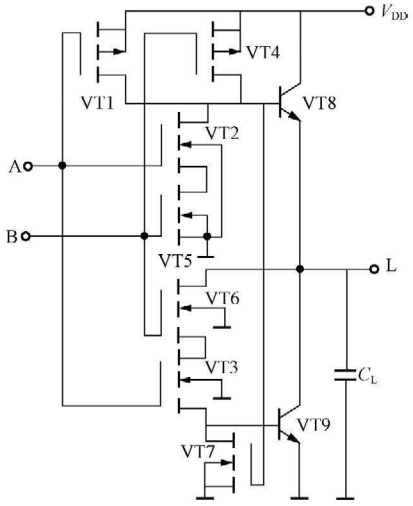


图 2－68　Bi-CMOS 与非门电路图

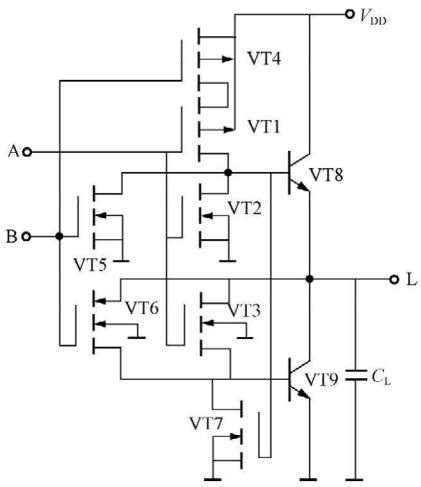


图 2-69 Bi-CMOS 或非门电路图

当输入端 A、B 中有低电平时，如 B 为低电平，则有 VT4、VT7 导通，VT5 截止，因此使 VT8 导通，VT9 截止，输出高电平。当输入端 A、B 全为高电平时，则有 VT2 和 VT5 导通，VT7 截止，因此使 VT8 截止，VT9 导通，输出低电平。在正逻辑体制下输出与输入之间实现与非逻辑。

（三）Bi-CMOS 或非门

Bi-CMOS 或非门电路图如图 2-69 所示。当输入端 A、B 中有高电平时，例如 A 为高电平，则有 VT2、VT3 导通，VT7 截止，使得 VT8 截止、VT9 导通，输出低电平。当输入端 A、B 全为低电平时，则有 VT1 和 VT4 导通，且 VT7 导通，使得 VT8 导通、VT9 截止，输出高电平。在正逻辑体制下输出与输入之间实现或非逻辑。

四、CMOS 系列集成电路

最早设计的 CMOS 门电路是 CD4000B 系列金属门集成电路，它和 TTL 集成电路相比具有以下几个方面的优点：电源电压工作范围宽（3～15V）；工作频率不高时，集成电路功耗非常小；噪声容限大（1V 噪声容限）。但是和 TTL 集成电路相比，CD4000B 系列金属门集成电路具有以下几个方面的不足：传播延迟大（100ns），驱动能力差（只能驱动 1 个 LS-TTL 负载），锁存效应（低阻状态过饱和），对静电敏感。CD4000B 系列金属门集成电路中常用的一些门电路有 4001（四 2 输入或非门）、4012（二 4 输入与非门）、4070（四 2 输入异或门）等。

其他系列 CMOS 集成电路编号采用 54 系列和 74 系列编号，通过字母 C 来标识。目前 CMOS 集成电路包括：74C 或 54C（CMOS），74HC（高速 CMOS），74HCT（高速 CMOS-TTL 兼容），74AC（先进 CMOS），74ACT（先进 CMOS-TTL 兼容）系列。

54C 系列（或 74C 系列）门电路和 54 系列（或 74 系列）TTL 相应门电路具有相同的功能和引脚排列。例如 7430 和 74C30 都是 8 输入与非门，且引脚排列完全相同；74HC 系列与 74HCT 系列和 74LS 系列的引脚完全兼容；74HC 系列门电路可以实现 TTL 和 4000 系列门电路的大部分功能。和 74LS 系列集成电路相比，74HC 和 74HCT 系列门电路的传播延迟要减小很多，其传播延迟为 10ns 左右。74HC 系列的电源电压为 2～6V；74HCT 系列的电源电压为 4.5～5.5V，它可以和 TTL 直接连接使用。

74AC111×××、74ACT111×××系列采用流水线结构设计。采用这种结构设计的集成电路，其电源引脚位于集成电路的中央，而不是通常的右上角。这种集成电路有两方面的优点：一是降低了集成电路内部的电感和电路高速开关产生的噪声，传播延迟降低至 5ns；二是输入引脚安排在集成电路的右侧，输入引脚安排在集成电路的左侧，这样可简化印刷电路板的设计和尺寸。不利因素是 AC111×××、ACT111×××系列与 TTL 系列的引脚不兼容。集成电路还克服了 CD4000B 系列的锁存效应和静电敏感等不足。

表 2-15 列出了几种 CMOS 系列集成电路的电源电压的取值。表 2-16 列出了几种 CMOS 系列集成电路在电源电压为 5V 时的输入、输出电平。

表 2-15　CMOS 系列集成电路的电源电压

电路种类	CD4000B	74C	74HC	74HCT	74AC	74ACT
电源电压/V	3～15	3～15	2～6	4.5～5.5	3～5.5	4.5～5.5

表 2-16　CMOS 系列集成电路输入、输出电平（电源电压为 5V）

参数名称＼电路种类	CD4000B	74HC	74HCT	74AC	74ACT
$V_{OH(min)}$/V	4.95	4.5	4.0	4.4	4.4
$V_{OL(max)}$/V	0.05	0.26	0.26	0.36	0.36
$V_{IH(min)}$/V	3.5	3.5	2.0	3.5	2.0
$V_{IL(max)}$/V	1.5	1.5	0.8	1.5	0.8

从表 2-16 可以看出，HCT、ACT 系列集成电路的输入高电平的下限为 2.0V，输入低电平的上限为 0.8 V，这些值和 TTL 电路的输入电平完全相同，因此，HCT、ACT 系列集成电路可以直接连接到 TTL 集成电路的输出端。TTL 集成电路是否可以直接连接到 HCT、ACT 系列集成电路的输出端呢？标准生产工艺的 TTL 系列集成电路的输出高电平的下限为 2.4V，LSTTL 系列集成电路的输出高电平的下限为 2.7V，而 HCT 系列集成电路的输出高电平的下限为 4.0V，ACT 系列集成电路的输出高电平的下限为 4.4V，位于 TTL 系列集成电路的范围之内，故 TTL 集成电路可以直接连接到 HCT、ACT 系列集成电路的输出端。所以 HCT、ACT 系列集成电路可以直接驱动 TTL 系列集成电路或者被驱动。

第六节　各种工艺的逻辑门之间的接口问题

一、TTL 与 CMOS 器件之间的接口问题

在实际使用中，根据系统设计要求，选择最佳的逻辑器件，再将它们组装在一起，显然可以达到更好的性能要求，因此不同生产工艺的逻辑器件混合使用是常常碰到的。这样，不同系列逻辑器件的接口，就是一个必须解决的问题。

两种不同类型的集成电路相互连接，驱动门必须要为负载门提供符合要求的高低电平和足够大的输入电流，即要满足下列条件：

（1）驱动门的 $U_{OH(min)}$ ≥负载门的 $U_{IH(min)}$；

（2）驱动门的 $U_{OL(max)}$ ≤负载门的 $U_{IL(max)}$；

（3）驱动门的 $I_{OH(max)}$ ≥负载门的 $I_{IH(总)}$；

（4）驱动门的 $I_{OL(max)}$ ≥负载门的 $I_{IL(总)}$。

（一）TTL 门驱动 CMOS 门

由于 TTL 门的 $I_{OH(max)}$ 和 $I_{OL(max)}$ 远远大于 CMOS 门的 I_{IH} 和 I_{IL}，所以 TTL 门驱动 CMOS 门时，主要考虑 TTL 门的输出电平是否满足 CMOS 输入电平的要求。

1. TTL 门驱动 4000 系列和 74HC 系列

表 2-17 给出了某些 TTL 系列和 CMOS 系列门电路的主要输入、输出特性参数。从表中看出，当电源电压都采用 5V 时，TTL 门的 $U_{OH(min)}$ 为 2.4V 或 2.7V，而 CMOS4000 系列

和 74HC 系列电路的 $U_{IH(min)}$ 为 3.5V，显然不满足要求。这时可在 TTL 电路的输出端和电源之间，接一上拉电阻 R_p，如图 2-70（a）所示。R_p 的阻值取决于负载器件的数目及 TTL 和 CMOS 器件的电流参数，一般在几百欧至几千欧间。

如果 TTL 和 CMOS 器件采用的电源电压不同，则应使用 OC 门，同时使用上拉电阻 R_p，如图 2-70（b）所示。

表 2-17　TTL 和 CMOS 门电路的主要输入、输出特性参数（电源电压为+5V）

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

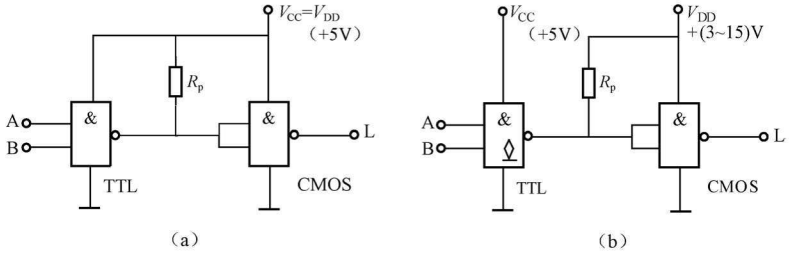


图 2-70　TTL 门驱动 4000 系列和 74HC 系列

（a）电源电压都为+5V 时的接口；（b）电源电压不同时的接口

2. TTL 门驱动 74HCT 系列

前面提到 74HCT 系列与 TTL 器件电压兼容。74HCT 的输入电压参数为 $U_{IH(min)}=2.0$V，而 TTL 的输出电压参数为 $U_{OH(min)}$ 为 2.4V 或 2.7V，因此两者可以直接相连，不需外加其他器件。

（二）CMOS 门驱动 TTL 门

从表 2-17 中看出，当电源电压都采用+5V 时，CMOS 门的 $U_{OH(min)}$ 大于 TTL 门的 $U_{IH(min)}$，CMOS 的 $U_{OL(max)}$ 小于 TTL 门的 $U_{IL(max)}$，两者电压参数相容。但是 CMOS 门的 I_{OH}、I_{OL} 参数较小，所以，这时主要考虑 CMOS 门的输出电流是否满足 TTL 输入电流的要求。

【例 2-6】 一个 74HC00 与非门电路能否驱动 4 个 7400 与非门？能否驱动 4 个 74LS00 与非门？如不能驱动，该如何改进？

解　从表 2-17 中查出，74 标准系列门的 $I_{IL}=1.6$mA，74LS 系列门的 $I_{IL}=0.4$mA，4 个 74 标准系列门的 $I_{IL(总)}=4\times1.6=6.4$（mA），4 个 74LS 门的 $I_{IL(总)}=4\times0.4=$

1.6（mA）。而 74HC 系列门的 $I_{OL}=4mA$，所以不能驱动 4 个 7400 与非门，可以驱动 4 个 74LS00 与非门。

要提高 CMOS 门的驱动能力，可将同一芯片上的多个门并联使用，如图 2-71（a）所示；也可在 CMOS 门的输出端与 TTL 门的输入端之间加一 CMOS 驱动器，如图 2-71（b）所示。

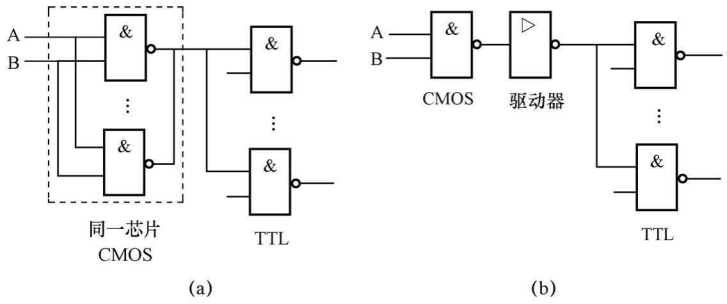


图 2-71　CMOS 驱动 TTL 门电路

（a）并联使用提高带负载能力；（b）用 CMOS 驱动器驱动 TTL 电路

二、TTL 和 CMOS 电路带负载时的接口问题

在工程实践中，常常需要用 TTL 或 CMOS 电路去驱动指示灯、发光二极管 LED、继电器等负载。

对于电流较小、电平能够匹配的负载可以直接驱动。图 2-72（a）所示为用 TTL 门电路驱动发光二极管的原理电路，这时只要在电路中串接一个几百欧的限流电阻即可。图 2-72（b）所示为用 TTL 门电路驱动 5V 低电流继电器，其中二极管 VD 作保护，用以防止过电压。

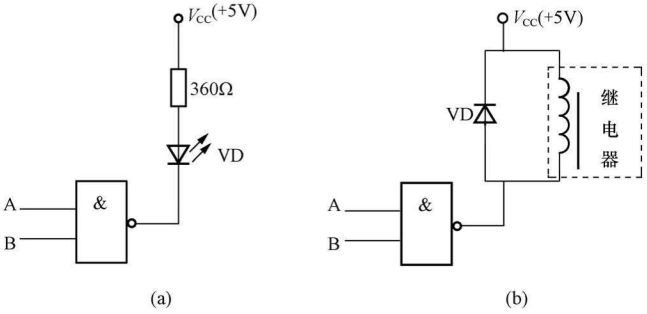


图 2-72　门电路带小电流负载

（a）驱动发光二极管；（b）驱动低电流继电器

如果负载电流较大，可将同一芯片上的多个门并联作为驱动器，如图 2-73（a）所示；也可在门电路输出端接三极管，以提高负载能力，如图 2-73（b）所示。

三、门电路多余输入端的处理

多余输入端的处理应以不改变电路逻辑关系及稳定可靠为原则。处理时通常采用下列方法：

（1）对于与非门及与门，多余输入端应接高电平，比如直接接电源正端，或通过一个上拉电阻（1～3kΩ）接电源正端，如图 2-74（a）所示；在前级驱动能力允许时，也可以与

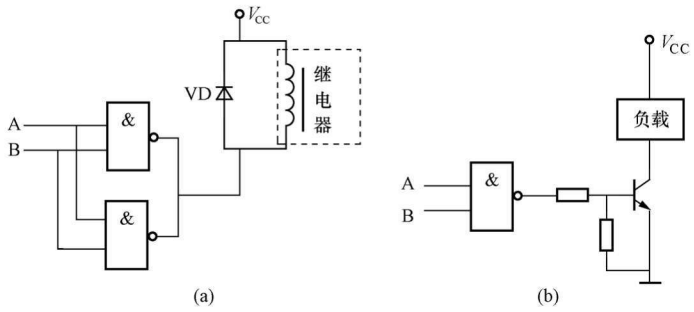


图 2-73 门电路带大电流负载

(a) 门电路并联使用；(b) 加驱动三极管

有用的输入端并联使用，如图 2-74（b）所示。

（2）对于或非门及或门，多余输入端应接低电平，比如直接接地，如图 2-75（a）所示；也可以与有用的输入端并联使用，如图 2-75（b）所示。

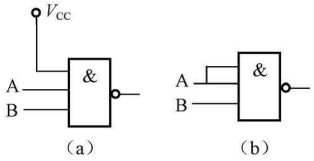


图 2-74 与非门多余输入端的处理

(a) 直接接电源正端；(b) 与输入端并联

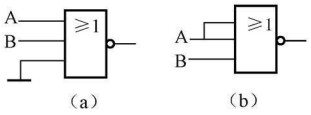


图 2-75 或非门多余输入端的处理

(a) 直接接地；(b) 与输入端并联

如果开路故障发生在 CMOS 门电路中，悬空引脚上的电压是浮动的，该引脚电压可以是高电平也可以是低电平。总之，CMOS 门电路中悬空引脚对门电路影响是不可预料的。

小　　结

二极管的开关特性包括静态开关特性和动态开关特性。二极管的静态开关特性是指二极管稳定地处于导通（开关处在关状态）和稳定地处于截止（开关处在断开状态）时的特性。二极管的动态开关特性是指二极管从一个状态到另一个状态的过渡过程中的特性。二极管从截止状态到导通状态的时间很短，可以近似认为瞬间完成。二极管从正向导通转为反向截止需要经历反向恢复过程，状态转换不能瞬间完成。

三极管的静态开关特性是当三极管稳定地处于饱和或截止状态时，三极管所呈现出的性质特点。三极管的动态开关特性是指三极管从一个状态到另一个状态的过渡过程中的特性。

正逻辑体制是指高电平表示 1，低电平表示 0；负逻辑体制是指高电平表示 0，低电平表示 1。

TTL 是应用最为广泛的一类集成电路。TTL 电路的主要参数有输出高电平 U_{OH}、标准输出高电平 U_{SH}、输出低电平 U_{OL}、标准输出低电平 U_{SL}、输入高电平 U_{IH}、输入高电平的下

限 $U_{IH(min)}$（开门电平 U_{ON}）、输入低电平 U_{IL}、输入低电平的上限 $U_{IL(max)}$（关门电平 U_{OFF}）、输入高电平噪声容限电压（最大允许负向干扰电压）、输入低电平噪声容限电压（最大允许正向干扰电压）、输入低电平电流 I_{IL}、输入高电平电流 I_{IH}、关门电阻 R_{OFF}、开门电阻 R_{ON}、最大输出低电平电流 $I_{OL(max)}$（也叫做最大灌电流）、输出低电平时的扇出系数 N_{OL}、最大输出高电平电流 $I_{OH(max)}$（也叫做最大拉电流）、输出高电平时的扇出系数 N_{OH}、取 N_{OL} 与 N_{OH} 中的较小值作为门电路的扇出系数 N_O、传输延迟时间、功率损耗（功耗）P_D、功耗—延时积 DP。

TTL 门电路除非门外，还有与非门、或非门、与或非门、集电极开路门、三态门、异或门等多种类型。

不使用 TTL 与门将两个要做与运算的信号 A、B 直接相连，即使用一根导线就实现了与运算，这就是线与。集电极开路的 TTL 门电路可以实现线与功能。为保持推拉式输出级的优点，并还能做线与连接，人们又开发了一种三态 TTL 门，它的输出除了具有一般 TTL 门的两种状态，即输出电阻小的高低电平外，还具有高输出电阻的第三态，称为高阻态。在普通 TTL 门电路的基础上增加控制端和控制电路就可以构成 TTL 三态输出门。

为了提高门电路的各种性能指标，除了标准生产工艺的 TTL 门电路以外，还有其他各种生产工艺类型的门电路，如肖特基 TTL 门电路、低功耗 TTL 门电路、低功耗肖特基 TTL 门电路、先进的低功耗肖特基 TTL 门电路、先进的肖特基 TTL 门电路、高速 TTL 门电路。

射极耦合逻辑电路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在 2ns 以下。

以 MOS 管作为开关元件的门电路叫做 MOS 逻辑门电路。就逻辑功能而言，它们与 TTL 门电路并无区别。MOS 门电路具有电路密度高、功耗低等许多优点，在数字电路中应用最为广泛。MOS 门电路与前面所介绍的 TTL 门电路最大的区别在于，MOS 门电路只有一种载流子，即自由电子或是空穴单一参与导电。

在实际使用中，根据系统设计要求，选择最佳的逻辑器件，再将它们组装在一起，显然可以达到更好的性能要求，因此不同生产工艺的逻辑器件混合使用是常常碰到的。这样，不同系列逻辑器件的接口问题，就是一个必须解决的问题。两种不同类型的集成电路相互连接，驱动门必须要为负载门提供符合要求的高低电平和足够大的输入电流，即要满足下列条件：

（1）驱动门的 $U_{OH(min)} \geqslant$ 负载门的 $U_{IH(min)}$；

（2）驱动门的 $U_{OL(max)} \leqslant$ 负载门的 $U_{IL(max)}$；

（3）驱动门的 $I_{OH(max)} \geqslant$ 负载门的 $I_{IH(总)}$；

（4）驱动门的 $I_{OL(max)} \geqslant$ 负载门的 $I_{IL(总)}$。

习　　题

2-1　在图 2-76 所示电路中，已知 $R_1=4.3\text{k}\Omega$，$R_2=16\text{k}\Omega$，$R_c=1.5\text{k}\Omega$，$V_{CC}=12\text{V}$，$V_{BB}=8\text{V}$，$V_{C1}=5\text{V}$，输入信号 u_I 的高电平为 5.5V，低电平为 0.3V。试问：

（1）当三极管 VT 的放大系数 $\beta=30$ 时，三极管能否饱和与截止？

（2）为了保证三极管 VT 的放大系数在输入信号为高电平时能饱和，β 值最小应为多少？

(3) 为了保证三极管 VT 的放大系数在输入信号为低电平时能截止，V_{BB}的最低值应为多少？

2-2 试分别在两种给定参数情况下：

(1) $R_c=1k\Omega$，$R_b=50k\Omega$，$\beta=100$；

(2) $R_c=1k\Omega$，$R_b=750k\Omega$，$\beta=100$。

判断图 2-77 所示电路的工作状态。

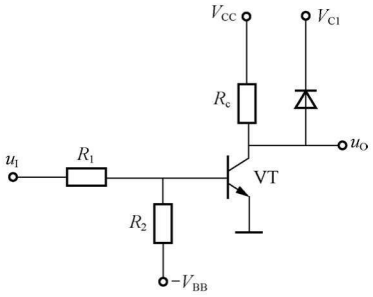


图 2-76 题 2-1 图

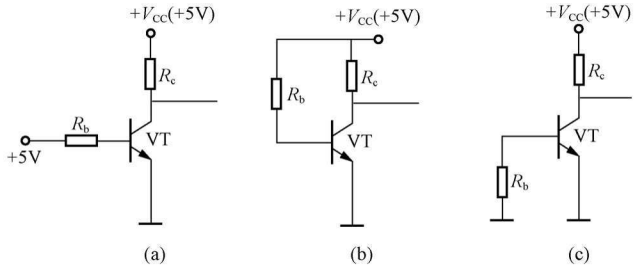


图 2-77 题 2-2 图

2-3 图 2-78 中，$R_b=300k\Omega$，$R_c=2k\Omega$，$V_{CC}=12V$，$\beta=50$，求三极管的状态。如果想改变三极管的状态，有哪些方法？

2-4 图 2-79 所示二极管门电路中，设二极管导通压降为 0.7V，输入信号 u_1、u_2 的 $U_{IH}=+5V$，$U_{IL}=0V$，试求在输入信号 u_1、u_2 的四种组合情况下所对应的输出电压 u_O。

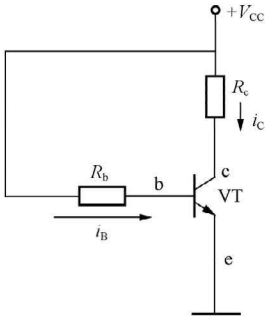


图 2-78 题 2-3 图

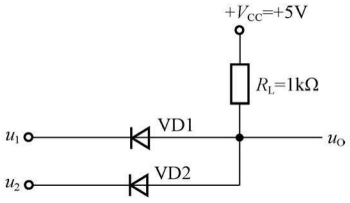


图 2-79 题 2-4 图

2-5 图 2-80 所示电路中，VD1、VD2 为硅二极管，导通压降为 0.7V。试回答下列问题：

(1) B 端接地，A 接 5V 时，u_O 等于多少伏？

(2) B 端接 10V，A 接 5V 时，u_O 等于多少伏？

(3) B 端悬空，A 接 5V，测 B 和 u_O 端电压，各应等于多少伏？

(4) A 接 10kΩ 电阻，B 悬空，测 B 和 u_O 端电压，各应为多少伏？

2-6 图 2-80 所示电路中，若在 A、B 端加入如图 2-81 所示波形，试画出 u_O 端对应的波形，并标明相应的电平值。假设二极管为理想二极管。

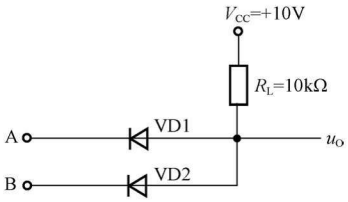


图 2-80　题 2-5 图

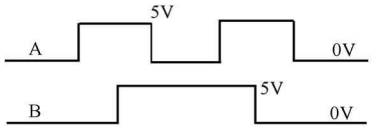


图 2-81　题 2-6 图

2-7　试分析图 2-82 所示电路，写出各输出函数 F_0，F_1，F_2，F_3 的逻辑表达式。

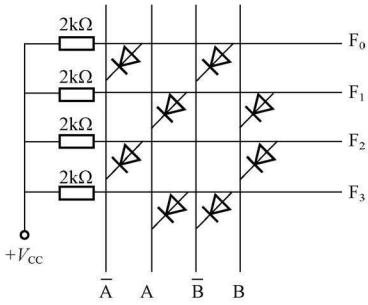


图 2-82　题 2-7 图

2-8　通过集成电路手册，查找 74LS83、74LS32、7486 集成电路的下列参数及取值：①高电平输出最小值；②高电平输入最小值；③低电平输出最大值；④低电平输入最大值；⑤从电压源获得的最大电流值；⑥低电平—高电平传播延迟最大值；⑦高电平—低电平传播延迟最大值；⑧低电平输出电流最大值；⑨低电平输入电流最大值。

2-9　根据题 2-8 得到的技术参数及取值，计算：

（1）74LS83 门电路的噪声容限；

（2）74LS83 可以驱动多少个 74LS32 门电路？

（3）74LS32 门电路可以驱动多少个 7486 门电路？

2-10　TTL 门电路如图 2-83 所示，已知门的参数为 $U_{SH}=3.6V$，$U_{SL}=0.1V$，$U_{IH(min)}=2.8V$，$U_{IL(max)}=0.4V$，$I_{IH}=20\mu A$，$I_{IL}=1mA$，$I_{OH(max)}=500\mu A$，$I_{OL(max)}=10mA$。为了实现 $F_1=\overline{AB}$，$F_2=\overline{\overline{AB}}$的逻辑关系，试确定电阻 R 的取值范围。

2-11　TTL 与非门接成图 2-84 所示电路。图中与非门为 74LS00 芯片，其有关参数如下：$U_{SH}=3.6V$，$U_{SL}=0.3V$，$U_{IH(min)}=2.8V$，$U_{IL(max)}=0.8V$，$R_{ON}=30k\Omega$，$R_{OFF}=2k\Omega$，$I_{OH(max)}=0.4mA$，$I_{OL(max)}=8mA$，$I_{IH}=20\mu A$。若要实现 $L_1=\overline{AB}$，$L_2=\overline{\overline{ABC}}$，试确定 R_1，R_2 的取值范围。

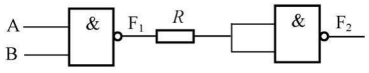


图 2-83　题 2-10 图

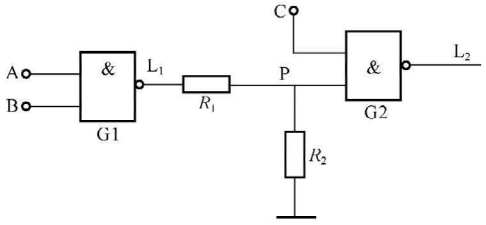


图 2-84　题 2-11 图

2-12　什么是上拉电阻？为什么集电极开路门需要使用上拉电阻？

2-13　电路如图 2-85（a）所示。已知当三极管 VT 导通时，$U_{BE}=0.7V$，当饱和时，$U_{CES}=0.3V$，$\beta=100$；OC 门的最大允许灌电流 $I_{OL(max)}=10mA$，输出低电平 $U_{OL}=0.3V$；TTL 与非门输入特性曲线如图 2-85（b）所示。当三极管集电极输出 P 端为高电平时 $U_{PH}=3.5V$，低电平时 $U_{PL}=0.3V$，试问：

(1) 若 P 端接 5 个与非门，则 R_b 的取值范围为多大？

(2) 若 $R_b=20k\Omega$，则此电路能带多少个与非门？

(3) 若将 OC 门换成普通 TTL 与非门，则电路会发生什么问题？

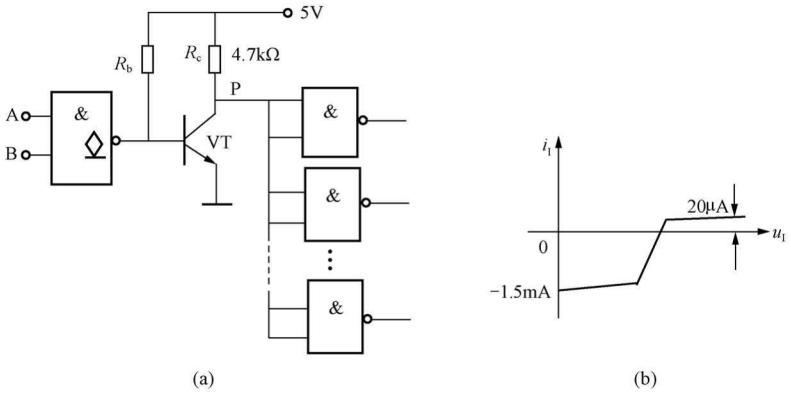


图 2-85 题 2-13 图

2-14 在图 2-86 由 74 系列 TTL 与非门组成的电路中，计算门 GM 能驱动多少同样的与非门。要求 GM 输出的高、低电平满足 $U_{OH}\geqslant 3.2V$，$U_{OL}\leqslant 0.4V$。与非门的输入电流为 $I_{IL}=1.6mA$，$I_{IH}=40\mu A$。$U_{OL}\leqslant 0.4V$ 时，输出电流最大值为 $I_{OL(max)}=16mA$；$U_{OH}\geqslant 3.2V$ 时，输出电流最大值为 $I_{OH(max)}=0.4mA$。GM 的输出电阻可以忽略不计。

2-15 试说明在下列情况下，用万用表（内阻为 20kΩ）测量图 2-87 的 u_{I2} 端得到的电压各为多少。①u_{I1} 悬空；②u_{I1} 接低电平（0.2V）；③u_{I1} 接高电平（3.2V）；④u_{I1} 经 51Ω 电阻接地；⑤u_{I1} 经 10kΩ 电阻接地。

2-16 试绘出图 2-88 电路的高电平输出特性曲线和低电平输出特性曲线。已知 $V_{CC}=5V$，$R_L=1k\Omega$。OC 门截止时，输出管的漏电流 $I_{OH}=200\mu A$。$u_{I1}=u_{I2}=U_{IH}$ 时，OC 门输出管饱和导通，在 $I_L<I_{LM}$ 的范围内导通内阻小于 20Ω。

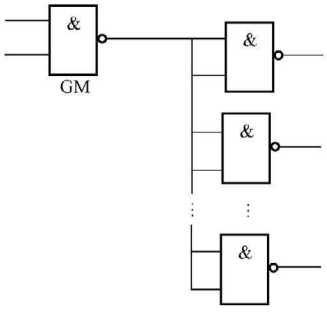


图 2-86 题 2-14 图

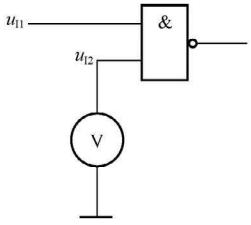


图 2-87 题 2-15 图

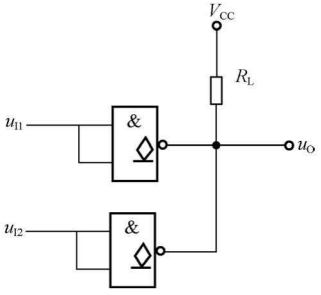


图 2-88 题 2-16 图

2-17 图 2-89（a）中，G1 为 TTL 三态门，G2 为 TTL 与非门，图 2-89（b）是 TTL 与非门的电压传输特性，图 2-89（c）是 TTL 与非门的输入负载特性。万用表的内阻 20kΩ/V，量程为 5V。当 C=0 和 C=1 时，试分别说明在下列情况下，万用表的读数、输出电压 u_O 各为多少伏？

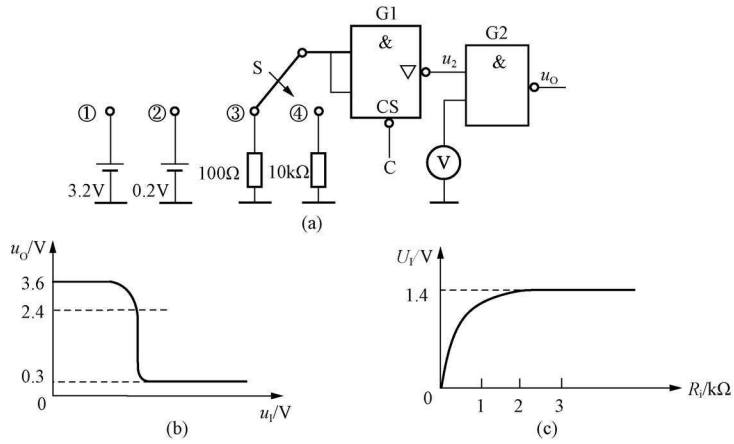


图 2-89　题 2-17 图

（a）电路图；（b）电压传输特性；（c）输入负载特性

（1）开关 S 悬空；（2）开关 S 接到①端；（3）开关 S 接到②端；（4）开关 S 接到③端；（5）开关 S 接到④端。

2-18　门电路如图 2-90（a）、（b）所示，请写出 F_1、F_2 的逻辑表达式。当向图 2-90（a）、（b）分别输入图 2-90（c）所示信号波形时，画出 F_1、F_2 端的波形。

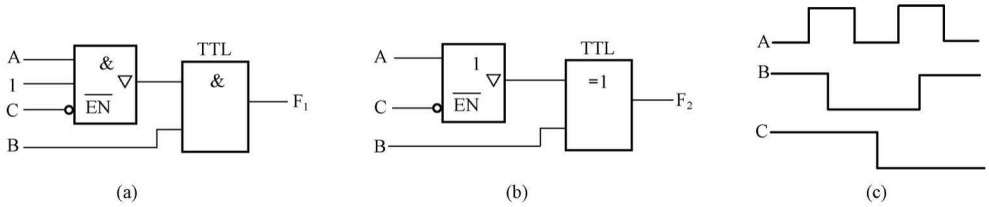


图 2-90　题 2-18 图

2-19　图 2-91 所示电路均为 TTL 电路，试问各电路能否实现给定的逻辑功能，如有错误，加以改正。

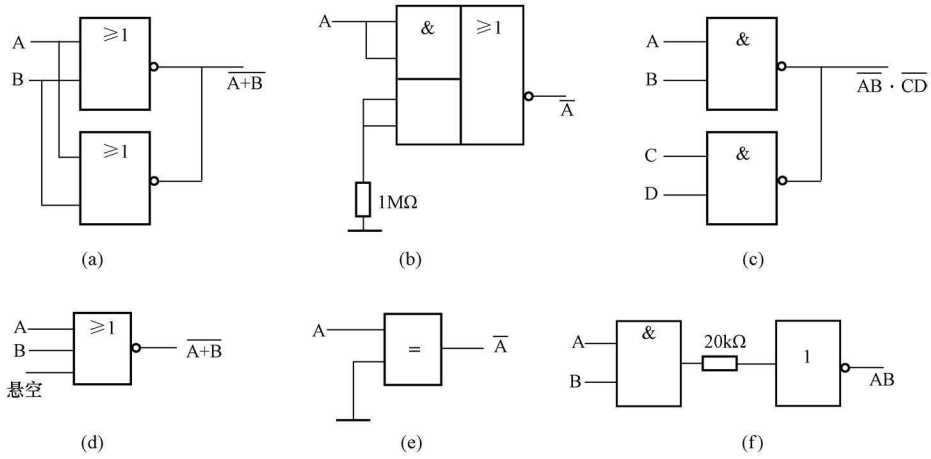


图 2-91　题 2-19 图

2-20 试分析如图 2-92 所示电路的逻辑功能，写出逻辑函数表达式。

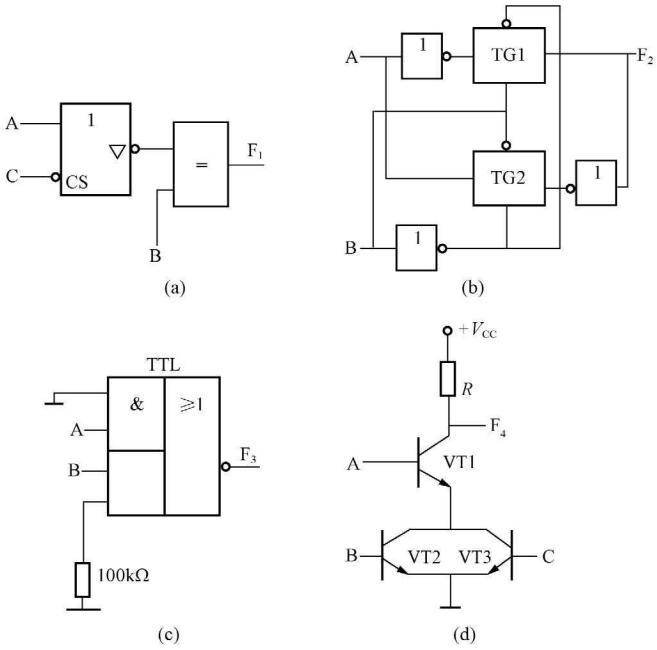


图 2-92 题 2-20 图

2-21 图 2-93 所示电路均为 CMOS 电路，试问各电路能否实现给定的逻辑功能，如有错误，加以改正。

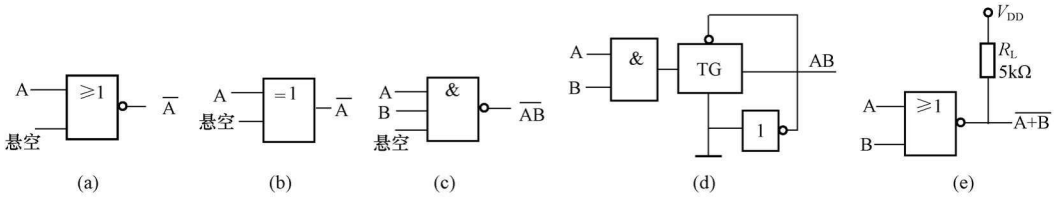


图 2-93 题 2-21 图

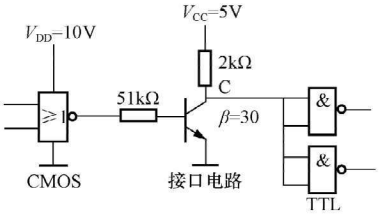


图 2-94 题 2-22 图

2-22 计算图 2-94 所示电路中接口电路输出端 C 的高、低电平，并说明接口电路参数的选择是否合理。CMOS 或非门的电源电压为 10V，空载输出的高、低电平分别为 $U_{OH}=9.95V$、$U_{OL}=0.05V$，门电路的输出电阻小于 200Ω。TTL 与非门的高电平输入电流 $I_{IH}=20\mu A$，低电平输入电流 $I_{IL}=0.4mA$。

2-23 图 2-95 所示逻辑门均为 CMOS 电路，已知电源电压 $V_{DD}=10V$，二极管的正向导通压降为 0.7V，写出各电路输出的逻辑表达式。

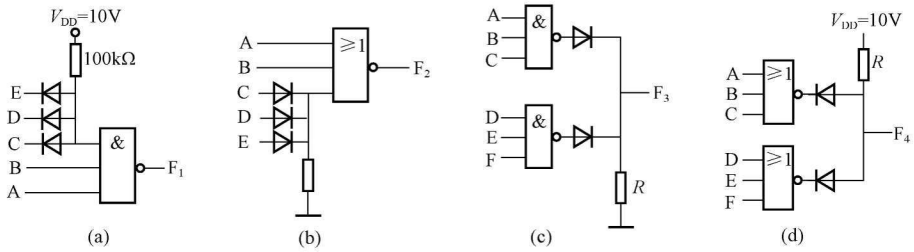


图 2-95　题 2-23 图

2-24　试分析图 2-96 所示 CMOS 门电路的逻辑功能。试写出图 2-96（a）中 L_1、L_2 的逻辑表达式，并判断图 2-96 所示两种连接方式的门电路能否用于 TTL 门电路。

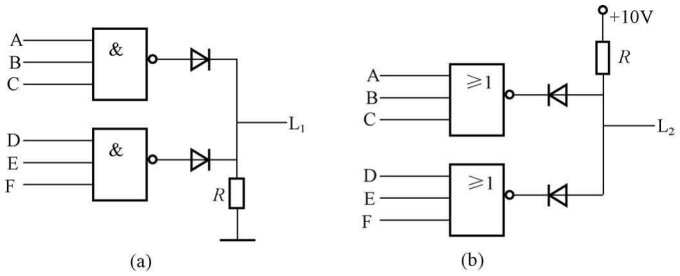


图 2-96　题 2-24 图

2-25　74HCT00 门电路可驱动多少个 74LS00 门电路？

2-26　工作电压为＋5V 时，74LS00 可以驱动多少个 74HC00 门电路？

第三章　组合逻辑电路

本章提要

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。前面介绍了构成数字系统的基本单元——逻辑门电路，分析和设计数字电路的工具——逻辑代数和卡诺图。本章将介绍如何运用这些基本知识分析和设计组合逻辑电路。首先介绍逻辑组合电路一般的分析方法和设计方法；然后着重从物理概念上说明竞争—冒险现象及其成因，并扼要地介绍消除竞争—冒险现象的常用方法；最后介绍常用的几种中规模集成的组合逻辑电路的工作原理和使用方法。

第一节　组合逻辑电路的分析

一、组合逻辑电路的特点及逻辑功能描述

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。组合逻辑电路在逻辑功能上的特点是在任意时刻，组合逻辑电路的输出信号仅仅取决于该时刻电路的各个输入信号，而与该时刻之前的输入信号的取值无关。

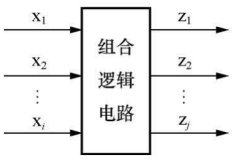


图 3－1　组合逻辑电路的一般框图

对于任何一个多输入信号、多输出信号的组合逻辑电路，其一般框图如图 3－1 所示。图中，x_1，x_2，…，x_i 为电路的 i 个输入信号，z_1，z_2，…，z_j 为电路的 j 个输出信号，输出信号与输入信号之间的逻辑关系可以用逻辑函数来描述，即

$$\begin{cases} z_1 = f_1(x_1, x_2, \cdots, x_i) \\ z_2 = f_2(x_1, x_2, \cdots, x_i) \\ z_j = f_j(x_1, x_2, \cdots, x_i) \end{cases} \tag{3-1}$$

如果写成向量函数的形式，则有

$$\mathbf{Z} = F(\mathbf{X}) \tag{3-2}$$

式中：$\mathbf{Z}=(z_1, z_2, \cdots, z_j)$；$\mathbf{X}=(x_1, x_2, \cdots, x_i)$。

组合逻辑电路在结构上的特点为：输入、输出之间没有反馈延迟通路；电路中不含有记忆单元。

二、组合逻辑电路的分析方法

所谓分析一个给定的组合逻辑电路，其目的就是通过分析找出该电路的逻辑功能。分析的第一步由逻辑电路图出发，从电路的输入到输出逐级写出逻辑函数表达式；第二步将得到的表示输出和输入关系的逻辑函数表达式进行化简或者变换（代数法、卡诺图法），目的是使逻辑关系简单明了，便于列写真值表；第三步根据真值表对该组合逻辑电路的功能进行描述。组合逻辑电路分析的一般步骤归纳如图 3－2 所示。特别说明，在第二个步骤中，通常并不需要一定得到最简的表达式，只是最简的表达式更加便于列写真值表而已。

图 3 - 2　组合逻辑电路分析的一般步骤

下面举例来说明组合逻辑电路的分析方法。

【例 3 - 1】 组合逻辑电路如图 3 - 3 所示，试分析该电路的逻辑功能。

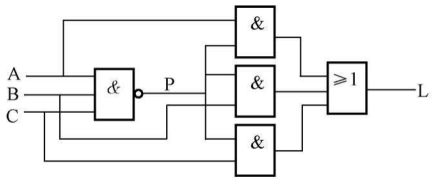


图 3 - 3 ［例 3 - 1］的逻辑电路图

解 （1）由逻辑电路图逐级写出逻辑函数表达式。为了写表达式方便，借助中间变量 P。

$$P = \overline{ABC}$$

$$L = AP + BP + CP = A\,\overline{ABC} + B\,\overline{ABC} + C\,\overline{ABC}$$

（2）化简与变换。因为下一步要列真值表，所以要通过化简、变换，使逻辑函数表达式有利于列真值表。L 化简变换为

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

（3）由逻辑函数表达式列出真值表。经过化简、变换的表达式为两个最小项之和的非，所以很容易列出真值表，见表 3 - 1。

表 3 - 1　　［例 3 - 1］ 的 真 值 表

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

（4）分析逻辑功能。由真值表可知，当 A、B、C 三个变量不一致时，电路输出为“1”，所以这个电路称为“不一致电路”。可见，一旦将电路的逻辑功能列成真值表，它的功能也就一目了然。

本例中输出变量只有一个，对于多输出变量的组合逻辑电路，分析方法完全相同。

第二节　用小规模集成电路（SSI）实现组合逻辑电路的设计

一、设计组合逻辑电路的一般方法

设计组合逻辑电路时要完成的工作是根据给出的实际逻辑问题，求出实现这一逻辑功能的最简单逻辑电路。这里所谓的“最简”，是指电路所用的器件数最少，器件的种类最少，而且器件之间的连线也最少。在第一章所介绍的用代数法和卡诺图法化简逻辑函数，就是为了获得最简的形式，以便能用最少的门电路来实现组合逻辑电路。但是，由于在设计中普遍采用中、小规模集成电路产品，因此，应根据具体情况，尽可能减少所用的器件数目和种类，使组装好的电路结构紧凑，达到工作可靠而且经济的目的。

组合逻辑电路的设计过程通常可按五个步骤进行。

1. 进行逻辑抽象、约定

在许多情况下，提出的设计要求是用文字描述的一个具有一定因果逻辑关系的事件。通过逻辑抽象，可以将这个因果关系抽象为用逻辑函数来描述。逻辑抽象过程通过一系列约定来完成，具体步骤如下：

（1）确定输入变量和输出变量。一般将引起事件发生的原因定为输入变量，而把事件的结果作为输出变量。

（2）定义逻辑“0”、“1”的含义。用二值逻辑的“0”、“1”分别代表每个变量的两种不同取值。而“0”、“1”的具体含义完全由设计者人为选定，这项工作也叫做逻辑变量赋值。

2. 列出逻辑真值表

根据给定的因果关系，列出真值表。至此，已将一个实际的逻辑问题抽象成一个逻辑函数。而且，这个逻辑函数首先是以真值表的形式描述的。

3. 写出逻辑函数表达式

为了便于对逻辑函数进行化简和变换，需要将真值表转换成为对应的逻辑函数表达式的形式。

4. 选定器件的类型

为了实现所需要的逻辑函数，既可以用小规模集成的门电路组成相应的逻辑电路，也可以用中规模集成的常用组合逻辑器件或者可编程器件等构成相应的逻辑电路。应根据对电路的具体要求和器件的资源情况决定采用哪种类型的器件。

5. 将逻辑函数表达式化简或者变换成适当的形式

在使用小规模集成电路（Small Scale Integration，SSI）进行设计时，为获得最简的设计结果，首先应该将逻辑函数表达式化成最简与或式，即函数式中的乘积项的个数最少，而且每个乘积项中的因子也最少。如果对所用器件的类型有附加的限制（如只允许用单一类型的与非门），则还应将函数式变换成与器件类型相应的形式（如将函数式化成与非—与非的形式）。使用小规模集成电路和中规模集成电路（Medium Scale Integration，MSI）的组合逻辑电路芯片设计组合逻辑电路的具体方法将在本章介绍。使用存储器和可编程逻辑器件（PLD）设计组合逻辑电路的具体方法将在第八章中介绍。

目前，用于逻辑设计的计算机辅助设计软件几乎都具有对逻辑函数进行化简或者变换的功能。因此，这一部分工作可以由计算机自动完成。

6. 画逻辑电路图

根据化简或者变换后的逻辑函数式，画出逻辑电路的连线图，至此，原理性设计（逻辑设计）已经完成。最后，还需进行工艺设计，完成组装，调试。对于组合逻辑电路的一般设计步骤归纳如图 3 - 4 所示。

图 3 - 4　组合逻辑电路设计的一般步骤

应当指出的是，上述的设计步骤并不是一成不变的。例如，有的设计要求直接以真值表的形式给出，就不用进行逻辑抽象了；有的问题逻辑关系比较简单直观，也可以不列真值表而直接写出表达式。

二、用小规模集成电路实现完全描述的组合逻辑电路设计

所谓完全描述，就是不含无关项的逻辑问题的描述。下面将举例说明用小规模集成电路实现完全描述的组合逻辑电路设计方法。

【例 3-2】 设计一个三人表决电路，其中一个人有最终的否决权，即只要这个人不同意，这件事就不能通过；但是这个人如果同意了这件事，这件事也不一定能通过，还要看另外两个人的意见，结果按“少数服从多数”的原则决定。

解 (1) 约定：三个人的意见分别用字母 A、B、C 表示，表决结果用字母 L 表示。设同意用逻辑“1”表示，不同意用逻辑“0”表示。表决结果通过用逻辑“1”表示，没通过用逻辑“0”表示。同时约定 A 是那个有最终否决权的人。

(2) 根据设计要求建立该逻辑函数的真值表。列出真值表如表 3-2 所示。

表 3-2　　[例 3-2] 的 真 值 表

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

(3) 把真值表中 L 取值为 1 的变量取值组合对应的最小项相或得标准与或表达式

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

(4) 化简。由于卡诺图化简法较方便易行，易于掌握，故一般用卡诺图进行化简。将该逻辑函数填入卡诺图，如图 3-5 所示。合并最小项，得该逻辑函数的最简与或表达式为

$$L = AB + AC$$

(5) 画出逻辑电路图，如图 3-6 所示。

(6) 如果要求用与非门实现该逻辑电路，就应将表达式转换成与非—与非表达式，即

$$L = AB + AC = \overline{\overline{AB}\,\overline{AC}}$$

画出逻辑图如图 3-7 所示。

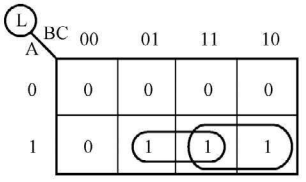


图 3-5 [例 3-2] 的卡诺图

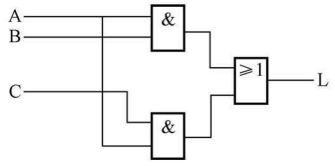


图 3-6 [例 3-2] 的逻辑电路图

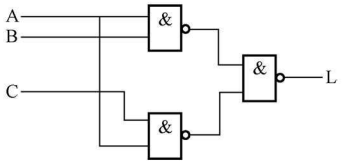


图 3-7 [例 3-2] 用与非门实现的逻辑电路图

在 [例 3-2] 中，对一件事情的看法，每个人的意见可能同意，也可能不同意，也就是说八种组合情况都有可能出现，这就是完全描述的逻辑问题。

【例 3-3】 设计一个电话机信号控制电路。电路有 I_0（火警）、I_1（盗警）和 I_2（日常业务）三种输入信号，通过排队电路分别从 L_0、L_1、L_2 输出，在同一时间只能有一个信号通过。如果同时有两个以上信号出现时，应首先接通火警信号，其次为盗警信号，最后是日常业务信号。试按照上述轻重缓急设计该信号控制电路。要求用集成门电路 7400（每片含 4 个 2 输入端与非门）实现。

解 （1）约定：I_0、I_1 和 I_2 分别表示有无火警、盗警和日常业务（因为题目中已经给出，所以要按题目中的要求来做），有火警、盗警和日常业务时，用“1”表示，无火警、盗警和日常业务时，用“0”表示；L_0、L_1、L_2 分别表示火警、盗警和日常业务的输出信号，这些信号能输出，用“1”表示，不能输出，用“0”表示。

（2）列真值表，如表 3-3 所示。

表 3-3 **［例 3-3］ 的 真 值 表**

I_0	I_1	I_2	L_0	L_1	L_2	I_0	I_1	I_2	L_0	L_1	L_2
0	0	0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	1	0	1	1	0	0
0	1	0	0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	1	1	1	1	0	0

（3）对三个输出信号，分别用卡诺图（图 3-8）化简，得到各输出的最简与或逻辑表达式为

$$L_0 = I_0, \quad L_1 = \overline{I_0} I_1, \quad L_2 = \overline{I_0}\,\overline{I_1} I_2$$

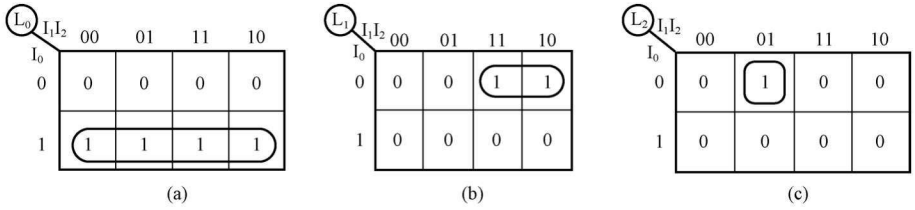


图 3-8 ［例 3-3］的卡诺图

（a）L_0 的卡诺图；（b）L_1 的卡诺图；（c）L_2 的卡诺图

（4）根据要求，将上式转换为与非表达式，即

$$L_0 = I_0, \quad L_1 = \overline{\overline{I_0 I_1}}, \quad L_2 = \overline{\overline{\overline{I_0}\,\overline{I_1} I_2}} = \overline{\overline{\overline{I_0}\,\overline{I_1} I_2}}$$

（5）画出逻辑图如图 3-9 所示，可用两片集成与非门 74X00 来实现。

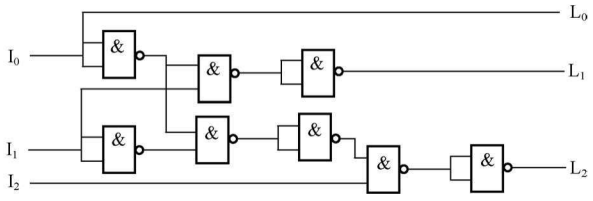


图 3-9 ［例 3-3］的逻辑电路图

可见，在实际设计逻辑电路时，有时并不是表达式最简单就能满足设计要求，还应考虑

所使用集成器件的种类，将表达式转换为能用所要求的集成器件实现的形式，并尽量使所用集成器件最少，就是设计步骤框图中所说的“最合理表达式”。

三、用小规模集成电路实现不完全描述的组合逻辑电路设计

所谓不完全描述，就是含无关项的逻辑问题的描述。下面将举例说明，用小规模集成电路实现不完全描述的组合逻辑电路设计方法。

【例 3-4】 设计一个将余 3 码变换成 8421BCD 码的组合逻辑电路。

解 （1）约定：用 $A_3A_2A_1A_0$ 表示输入的余 3 码，用 $L_3L_2L_1L_0$ 表示输出的 8421BCD 码。

（2）根据题目要求，列出真值表如表 3-4 所示。余 3 码中 0000、0001、0010、1101、1110、1111 没有使用，所以在真值表中作为无关项处理。

表 3-4　　[例 3-4] 的 真 值 表

输入（余 3 码）				输出（8421 码）				输入（余 3 码）				输出（8421 码）			
A_3	A_2	A_1	A_0	L_3	L_2	L_1	L_0	A_3	A_2	A_1	A_0	L_3	L_2	L_1	L_0
0	0	0	0	×	×	×	×	1	0	0	0	0	1	0	1
0	0	0	1	×	×	×	×	1	0	0	1	0	1	1	0
0	0	1	0	×	×	×	×	1	0	1	0	0	1	1	1
0	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
0	1	0	0	0	0	0	1	1	1	0	0	1	0	0	1
0	1	0	1	0	0	1	0	1	1	0	1	×	×	×	×
0	1	1	0	0	0	1	1	1	1	1	0	×	×	×	×
0	1	1	1	0	1	0	0	1	1	1	1	×	×	×	×

（3）用卡诺图进行化简。本题目为 4 个输入信号、4 个输出信号，故分别画出 4 个输出信号的 4 变量卡诺图，如图 3-10 所示。注意本例题中有 6 个无关项，应充分利用，使其逻

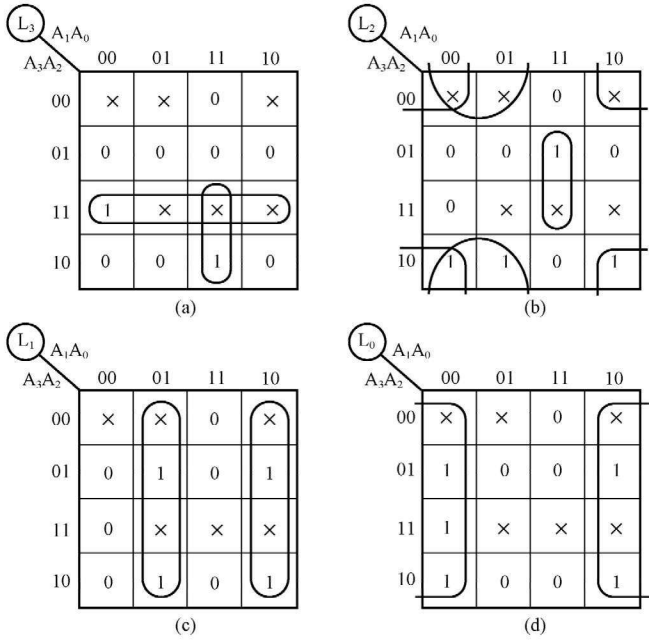


图 3-10　余 3 码变换成 8421BCD 码的卡诺图

（a）L_3 的卡诺图；（b）L_2 的卡诺图；（c）L_1 的卡诺图；（d）L_0 的卡诺图

辑函数尽量简单。

化简后得到的逻辑表达式分别为

$$L_0 = \overline{A_0}$$

$$L_1 = A_1\overline{A_0} + A_0\overline{A_1} = A_1 \oplus A_0$$

$$L_2 = \overline{A_2}\,\overline{A_0} + A_2A_1A_0 + \overline{A_2}\,\overline{A_1} = \overline{\overline{\overline{A_2}\,\overline{A_0}}\;\overline{A_2A_1A_0}\;\overline{\overline{A_2}\,\overline{A_1}}}$$

$$L_3 = A_3A_2 + A_3A_1A_0 = \overline{\overline{A_3A_2}\;\overline{A_3A_1A_0}}$$

（4）由逻辑表达式画出逻辑电路图如图 3 - 11 所示。

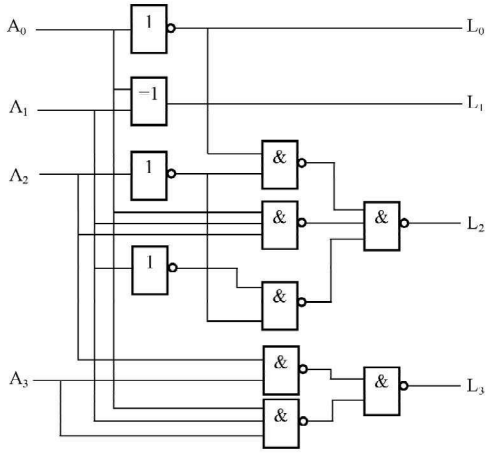


图 3 - 11　余 3 码变换成 8421BCD 码的逻辑电路图

需要提醒读者的是，在真值表中输出变量的位置画上“×”表示这个最小项是无关项，而在输入变量的位置画上“×”只是表示将真值表若干行压缩成了一行，并不是无关项。

第三节　组合逻辑电路中的竞争—冒险

前面在分析和设计组合逻辑电路时，考虑的是输入信号、输出信号已经处于稳定的逻辑电平的情况，没有考虑输入变化瞬间的情况。为了保证系统工作的可靠性，有必要再讨论当输入信号逻辑电平发生变化的瞬间电路的工作情况。由于门电路存在延迟时间，在输入变化的瞬间，在电路的输出端口可能会出现与预期不一样的尖峰脉冲，这种情况称为电路出现了冒险。冒险分静态冒险和动态冒险，静态冒险出现在二级电路中，动态冒险只出现在多级电路中。静态冒险又分为静态 1 冒险和静态 0 冒险。静态 1 冒险出现在二级与或电路中，静态 0 冒险出现在二级或与电路中。本节将重点讨论组合逻辑电路的静态冒险问题，在最后简单介绍动态冒险的基本概念。

一、静态冒险的定义

在组合逻辑电路中，如果输入变化前、后稳态输出为 1，而转换瞬间出现 0 的毛刺，称为静态 1 冒险，其示意图如图 3 - 12（a）所示；如果输入变化前、后稳态输出为 0，而转换瞬间出现 1 的毛刺，称为静态 0 冒险，其示意图如图 3 - 12（b）所示。

二、静态冒险现象及其产生的原因

首先看一个简单的例子。在图 3 - 13（a）所示的与门电路中，稳态下，无论 A＝1、B＝0，还是 A＝0、B＝1，输出都是 L＝0。但是在输入信号 A 从 0 跳变为 1 时，如果 B 从 1 跳变为 0，而且，信号 A、B 改变状态的时间有先有后（经过的传输线长度不同，门电路的传输时间不同），从而使信号 A 先上升到逻辑 1，信号 B 后下降到逻辑 0，则输出端 L 就产生了正向干扰脉冲（或者称为尖峰脉冲，电压毛刺），如图 3 - 13（b）所示。这个干扰脉冲不符合门电路稳态下的逻辑功能，因而它是系统内部的一种噪声。对于或门电路也有类似的现象。

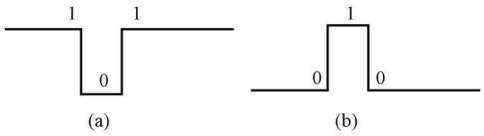


图 3 - 12　静态冒险示意图

（a）静态 1 冒险；（b）静态 0 冒险

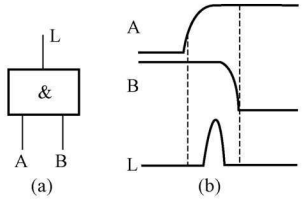


图 3 - 13　与门电路及其产生尖峰脉冲的示意图

（a）与门电路；（b）与门由竞争产生尖峰脉冲的示意图

门电路两个输入信号同时向相反的逻辑电平跳变（一个从 1 变 0，另一个从 0 变 1），而变化的时间有差异的现象称作竞争。逻辑表达式和真值表所描述的是静态逻辑关系，而竞争则发生在从一种稳态到另一种稳态的过渡过程中，因此，竞争是动态问题，它发生在输入变量变化时。当某个变量发生变化时，如果真值表所描述的逻辑关系受到短暂的破坏并在输出端出现不应有的尖脉冲，则称这种情况为冒险。应该指出，有竞争现象时不一定都会产生冒险。例如图 3 - 13 的与门电路中，如果在 A 上升到逻辑 1 之前，信号 B 已经下降到逻辑 0，这时输出不会产生干扰脉冲。

由于在实际的数字系统中，与门、或门的输入信号 A、B 是经过不同的传输途径到达的，在设计时往往很难准确地知道 A、B 到达次序的先后，以及它们在上升时间和下降时间上的细微差别。因此，只能说只要存在竞争现象，输出就有可能出现违背稳态逻辑关系的电压毛刺。

图 3 - 14（a）所示的电路中，逻辑函数表达式为 $L=A\overline{A}$，理想情况下，输出应恒等于 0。但是由于 G1 门的延迟时间为 t_{pd}，$\overline{A}$ 下降沿到达 G2 门的时间比 A 信号上升沿晚 t_{pd}，使 G2 输出端出现了一个正向窄脉冲，即静态 0 冒险，如图3 - 14（b)所示。

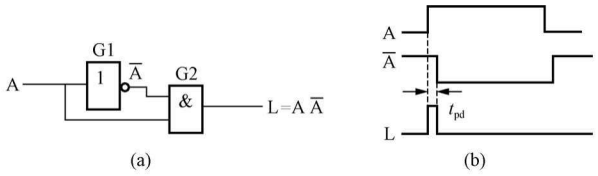


图 3 - 14　存在静态 0 冒险的电路及波形图

（a）逻辑电路图；（b）有静态 0 冒险的波形图

同理，在图 3 - 15（a）所示的电路中，由于 G1 门的延迟时间为 t_{pd}，会使 G2 输出端出现一个负向窄脉冲，即静态 1 冒险，如图3 - 15（b）所示。

冒险是一种干扰脉冲，有可能引起后级电路的错误动作，应在设计中采取措施加以避免。

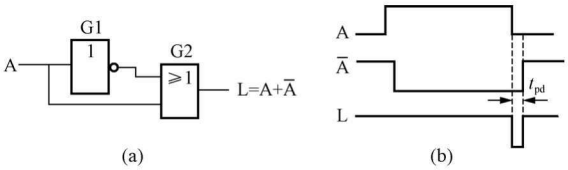


图 3 - 15　存在静态 1 冒险的电路及波形图

（a）逻辑电路图；（b）有静态 1 冒险的波形图

综上所述，产生竞争冒险的根本原因在于门电路存在延迟时间。当一个输入信号经过多条路径传送后又重新会合到某个门的输入端，由于不同路径上门的级数不同，或者门电路延迟时间的差异，导致到达会合点的时间有先有后。

三、判断静态冒险现象的方法

检查一个组合逻辑电路中是否存在静态冒险，最直接的方法就是逐级列出真值表，并找出哪些输入信号会发生竞争，即一个信号从 0 变为 1，而另一个信号同时从 1 变为 0，然后判断是否会在整个电路的输出端产生干扰脉冲。如果可能产生则有静态冒险，否则就没有。但这种方法由于其工作量庞大，因而并不实用。

在组合逻辑电路中，当某瞬间只有单个输入变量改变状态的简单情况时，分析有无静态冒险的简便方法为，写出函数的与或表达式，画出卡诺图，检查有无几何相邻的乘积项（两个不同的乘积项如果包含了几何相邻的最小项，则这两个乘积项称为几何相邻），有几何相邻的乘积项则有竞争—冒险，反之则无。也可采用代数法来判断一个组合逻辑电路是否存在冒险，其方法为写出组合逻辑电路的逻辑表达式，当某些逻辑变量取特定值（0 或 1）时，如果表达式能转换为 $L=A\overline{A}$，则存在静态 0 冒险；如果表达式能转换为 $L=A+\overline{A}$，则存在静态 1 冒险。

图 3 - 16（a）是一个四变量逻辑函数化简后的二级与或电路。下面分析图 3 - 16（a）所示电路，以对静态冒险的概念做进一步的解释。由图 3 - 16（a）所示的电路可得逻辑函数表达式为

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

其对应的卡诺图如图 3 - 16（b）所示，图中标示出了画圈的方法。

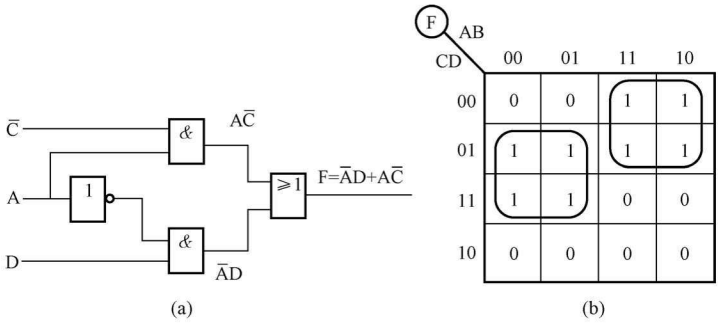


图 3 - 16　一个四变量逻辑函数简化后的二级与或逻辑电路和卡诺图

（a）逻辑电路图；（b）卡诺图

如图 3 - 17 所示，当输入 ABCD＝1100 时，F＝1；当输入 ABCD＝1101 时，F＝1。可见，在 ABCD 从 1100 变成 1101 的过程中，输出 F 一直维持为 1，没有静态冒险出现。观察图 3 - 16（b）所示的卡诺图，ABCD＝1100 和 ABCD＝1101 包含在一个圈中，即属于同一个乘积项，这种情况下，不会出现静态冒险。

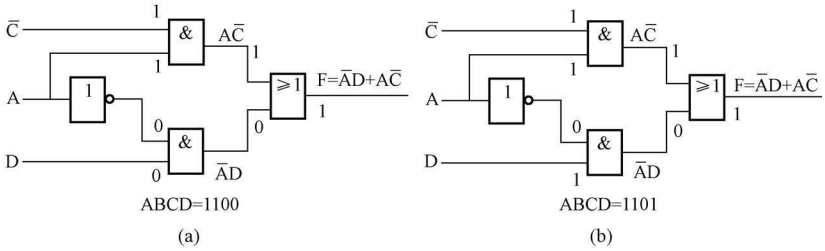


图 3 - 17　图 3 - 16（a）所示电路在给定输入信号时的值（无静态冒险）

（a）ABCD＝1100 时；（b）ABCD＝1101 时

在图 3 - 18 中，当输入 ABCD＝1101 时，F＝1；当输入 ABCD＝0101 时，F＝1。在 ABCD 从 1101 变成 0101 的过程中，输出 F 不能一直维持为 1，会出现短暂的 0，有静态冒险出现。静态冒险出现在 A 已经变成 0，但是$\overline{A}$还没有变成 1，仍然维持 0 的一段时间。观察图 3 - 16（b）所示的卡诺图，ABCD＝1101 和 ABCD＝0101 分别属于不同的圈，即不属于同一个乘积项，这种情况下，就会出现静态 1 冒险。

如果在图 3 - 16（b）所示的卡诺图中，增加一个圈，将 ABCD＝1101 和 ABCD＝0101 包含在同一个圈中，对应的表达式为 F＝$\overline{A}$D＋A $\overline{C}$＋$\overline{C}$D，增加了冗余项，就可以消除静态 1 冒险。

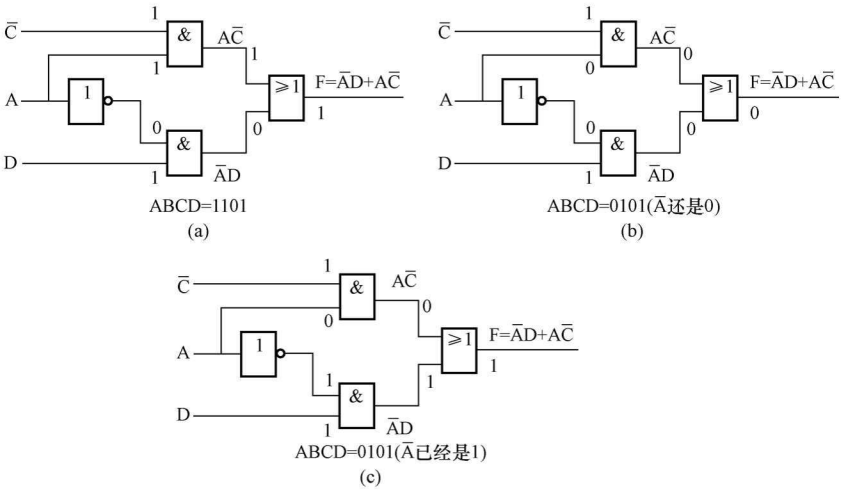


图 3 - 18　图 3 - 16（a）所示电路在给定输入信号时的值（出现静态冒险）

（a）ABCD＝1101 时；（b）ABCD＝0101 时；（c）ABCD＝0101 时

【例 3 - 5】　判断逻辑函数 L ＝（A ＋ B)($\overline{B}$ ＋ C）所对应的逻辑电路是否存在静态冒险。

解　如果令 A＝C＝0，则有 L＝B $\overline{B}$，因此，该逻辑电路存在静态 0 冒险。

读者也可以画出对应的卡诺图来观察某逻辑电路是否存在静态冒险。在卡诺图中，用圈 0 的方法来获得或项。

由逻辑函数表达式判断静态冒险的方法虽然简单，但局限性也太大。因为多数情况下输入变量都有两个以上同时改变状态的可能性，如果输入变量的数目又很多，就更难于从逻辑函数式上简单地找出所有产生竞争—冒险的情况了。

借助于计算机辅助分析的手段，可以有效地从原理上检查复杂数字电路的竞争—冒险现象。通过在计算机上运行数字电路的模拟仿真程序，可以迅速查出电路是否会存在竞争—冒险现象。目前，已经存在较成熟的应用程序。

另外一种方法是用实验的方法来检查电路的输出端是否存在竞争—冒险产生的毛刺电压。需要注意的是，加到输入端的信号波形应包含输入变量的所有可能发生的状态变化。

最后需要说明的是，使用计算机辅助分析的方法检查过的电路，也往往还需要经过实验的方法进行验证，才能最终确定是否存在竞争—冒险现象。这主要是因为用计算机软件模拟数字电路时，智能采用标准化的典型参数，有时还要做一些近似，所以，得到的模拟结果有时和实际电路的工作状态会有出入。因此可以认为，只有实验室检查的结果才是最终的结论。

四、静态冒险现象的消除方法

当组合逻辑电路存在静态冒险现象时，可以采取以下方法来消除静态冒险现象。

1. 加冗余项

在图 3 - 16 所示的电路中存在静态冒险现象。如在其逻辑函数表达式中增加乘积项$\overline{C}D$，使其变为 $F=\overline{A}D+A\overline{C}+\overline{C}D$。这个逻辑函数增加了乘积项$\overline{C}D$ 后，已不是“最简”，故这种乘积项称冗余项。

2. 变换函数逻辑式，消去互补变量

［例 3 - 5］函数的逻辑式 $L=(A+B)(\overline{B}+C)$ 存在静态冒险现象。如将其变换为 $L=A\overline{B}+AC+BC$，则在原来产生冒险的条件 $A=C=0$ 时，$L=0$，不会产生静态冒险。

3. 增加选通脉冲

在图 3 - 19 中，门 G1、G4 的输入端引进脉冲 P_2。由于 P_2 的作用时间取在电路达到新的稳定状态之后，所以 G1、G4 的输出端不再会有干扰脉冲的出现。不过，这时 G1、G4 正常的输出信号也变成脉冲形式了，而且宽度也与选通脉冲相同。如当输入信号 AB 变为 11 以后，L_3 并不立即变成高电平，而要等 P_2 出现时，才给出一个正脉冲。

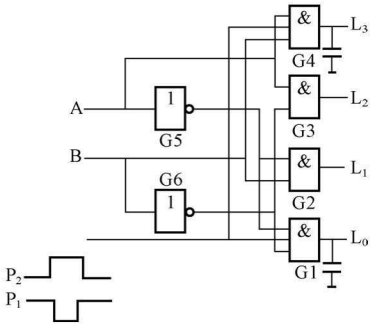


图 3 - 19 滤波电容克服静态冒险

4. 增加输出滤波电容

由于竞争—冒险产生的干扰脉冲的宽度一般都很窄，在可能产生冒险的门电路输出端并接一个滤波电容（一般为 4～20pF）。如在图 3 - 19 中 L_3、L_0 处接对地电容，利用电容两端的电压不能突变的特性，使输出波形上升沿和下降沿都变得比较缓慢，从而起到消除冒险现象的作用，将电压毛刺的幅度消弱至门电路的阈值电压以下。

这种方法简单易行，但增加了输出电压波形的上升时间和下降时间，使波形变坏。

5. 引入封锁脉冲

为了消除因竞争—冒险所产生的干扰脉冲，可以引入一个负脉冲（如图 3 - 19 中 P_1），将输入信号可能发生干扰脉冲的门封住。封锁脉冲须与输入信号的转换同步，而且宽度不应小于从一个稳态到另一个稳态所需要的过渡时间。

把上述几种方法比较一下可以看出，接入滤波电容的方法简单易行，但输出电压的波形随之变坏，因此只适用于对输出波形的前、后沿无严格要求的场合；引入选通脉冲和封锁脉冲的方法也比较简单，而且不需要增加电路元件，但使用这种方法时必须设法得到一个与输入信号同步的选通脉冲，对这个脉冲的宽度和作用的时间均有严格的要求；至于增加冗余项和消去互补变量的方法，倘若运用得当，有时可以收到令人满意的效果，但其局限性也很大。因此，要能很好地解决问题，必须在实践中积累经验和总结方法。

五、动态冒险的定义

在组合逻辑电路中，若在输入信号变化前后，稳定状态输出不同，则不会出现静态冒险。但如果在得到最终稳定输出之前，输出发生了三次变化，如图 3 - 20（a）所示，输出的变化序列为 0、1、0、1，称为动态 1 冒险；如图 3 - 20（b）所示，输出的变化序列为 1、0、1、0，称为动态 0 冒险。

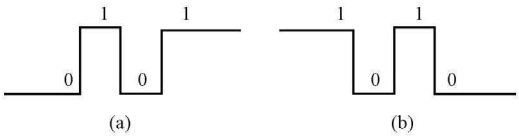


图 3 - 20　动态冒险示意图

（a）动态 1 冒险；（b）动态 0 冒险

第四节　编　码　器

由于人们在实践中遇到的逻辑问题层出不穷，因而为解决这些逻辑问题而设计的逻辑电路也不胜枚举。然而我们发现，其中有些逻辑电路经常、大量地出现在各种数字系统中。这些电路包括编码器、译码器、数据选择器、数据分配器、算术逻辑运算电路、比较器等。为了使用上的方便，人们已经把这些逻辑电路制成了中规模集成电路（MSI）的标准化产品。下面将分别介绍它们的工作原理和使用方法。在学习这些内容过程中，要以前面学习的逻辑代数、卡诺图等作为工具，进行逻辑电路的分析和设计。

编码的概念在第一章的码制一节中已经有解释。在日常生活中，常常需要给某些信息起名字，以对它们进行区分，用一个术语来说，这就是编码。日常生活中的编码可以用英文字母、汉字、数字等各种符号的组合，但是在数字电路系统中，只能用二进制代码的组合来给某些信息起名字，要区分的信息数越多，所使用的二进制代码的位数就越多。

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。用 n 位二进制代码可以实现最多对 $N = 2^n$ 个信号进行编码。按照输入信号有无优先级别来划分，编码器可以分为普通编码器和优先编码器。

一、普通编码器

所谓普通编码器，是指任何时刻只允许一个输入信号有效，否则输出将发生混乱。这种编码器要求输入信号是互相排斥的变量，它们之间存在约束关系，所以使用中受到一定限制。

1. 二进制普通编码器

用 n 位二进制代码对 2^n 个信号进行编码的电路称为二进制编码器。二进制编码器有 2^n 个输入端，n 个输出端。现通过一个 4 线—2 线编码器电路的设计例题详细讨论。

【例 3 - 6】 试设计一个 4 线—2 线编码器电路，可将 4 个信息编成 2 位代码。此处假设任一瞬间，4 个信息必须有一个而且只能有一个处于有效状态。

解 （1）约定：信息有效用 1 表示，无效用 0 表示。4 个信息分别用 I_0、I_1、I_2、I_3 表示，2 位代码用 A_1、A_0 表示，且对应的关系为：信息 I_0 的编码为 00，信息 I_1 的编码为 01，信息 I_2 的编码为 10，信息 I_3 的编码为 11。

（2）真值表如表 3 - 5 所示，因为有 4 个输入变量，所以真值表中共有 16 行，分别画在左右两侧，由双竖线隔开。每行对应了一种变量取值组合，根据题目中的叙述，其中 12 种变量取值组合不会出现，所以视为无关项。

表 3 - 5　　［例 3 - 6］ 的 真 值 表

输入				输出		输入				输出	
I_0	I_1	I_2	I_3	A_1	A_0	I_0	I_1	I_2	I_3	A_1	A_0
0	0	0	0	×	×	1	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	×	×
0	0	1	0	1	0	1	0	1	0	×	×
0	0	1	1	×	×	1	0	1	1	×	×
0	1	0	0	0	1	1	1	0	0	×	×
0	1	0	1	×	×	1	1	0	1	×	×
0	1	1	0	×	×	1	1	1	0	×	×
0	1	1	1	×	×	1	1	1	1	×	×

（3）用卡诺图（图 3 - 21）化简，得到输出 A_1 和 A_0 的逻辑表达式为

$$A_1 = \overline{I_0}\,\overline{I_1}$$

$$A_0 = \overline{I_0}\,\overline{I_2}$$

（4）用门电路实现该 4 线—2 线编码器的逻辑电路，如图 3 - 22 所示。

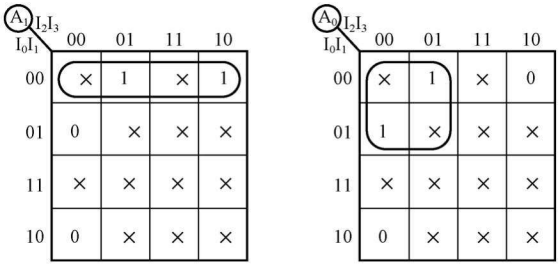


图 3 - 21　［例 3 - 6］的求 A_1 和 A_0 的卡诺图

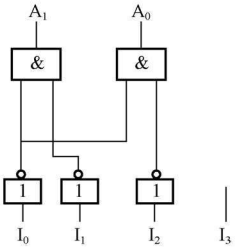


图 3 - 22　4 线—2 线编码器的逻辑电路图

在使用上述编码电路的过程中，必须保证输入 I_0、I_1、I_2、I_3 任一瞬间有一个而且只能有一个处于有效状态。因为在设计此电路的过程中使用了无关项，必须遵守无关项，否则该编码器不能给出正确的编码，从而导致编码混乱。比如 I_1 和 I_2 同时为 1 时，该编码器的输出 A_1A_0 为 00。

从前面的组合电路设计一节中已经知道，此电路的设计完全可以使用另外的一种约定。例如，输入信息有效用 0 表示，无效用 1 表示。4 个输入信息分别用 I_0、I_1、I_2、I_3 表示，2 位代码用 A_1、A_0 表示，且对应的关系为：信息 I_0 的编码为 11，信息 I_1 的编码为 01，信息

I_2 的编码为 10，信息 I_3 的编码为 00。由此可以列出对应的真值表，读者可以自行完成余下的设计工作。

2. 二—十进制编码器——键控 8421BCD 码编码器

所谓二—十进制编码器是将十进制的十个数码 0～9 分别编成对应的 8421BCD 代码的电路。输入信号 S_0～S_9 分别代表十进制的十个数码 0～9，输出信号 ABCD 代表对应的 8421BCD 代码。约定输入信号为低电平有效。该编码器真值表如表 3 - 6 所示，其中 GS 为控制使能标志，当 10 个输入信号全部无效时 GS＝0，当有输入信号有效时 GS＝1。因为有 10 个输入信号，真值表中输入变量的取值组合应该有 1024 行，1024 行的真值表很难完整地画出来。在这个编码器中，我们有这样的约定，任意时刻输入信号可以全部无效或者有一个有效，不可能出现两个或多于两个以上的输入信号同时有效。这样在真值表中两个或多于两个以上的输入信号同时有效的情况作为约束项看待，在表 3 - 6 中没有列出。

表 3 - 6　　键控 8421BCD 码编码器真值表

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

从前面所学的知识可以知道，不能画出 10 个变量的卡诺图，所以用如下的方法写出这个真值表对应的逻辑表达式。在真值表中，每一行只可能有一个 0，根据这个特点，写出逻辑表达式（没有利用无关项）为

$$A=\overline{S_8}+\overline{S_9}=\overline{S_8S_9}$$

$$B=\overline{S_4}+\overline{S_5}+\overline{S_6}+\overline{S_7}=\overline{S_4S_5S_6S_7}$$

$$C=\overline{S_2}+\overline{S_3}+\overline{S_6}+\overline{S_7}=\overline{S_2S_3S_6S_7}$$

$$D=\overline{S_1}+\overline{S_3}+\overline{S_5}+\overline{S_7}+\overline{S_9}=\overline{S_1S_3S_5S_7S_9}$$

由上述表达式画出逻辑电路图，如图 3 - 23 所示。

在图 3 - 23 中，左端的 10 个按键 S_0～S_9 分别代表输入的十进制的 10 个数码 0～9，输

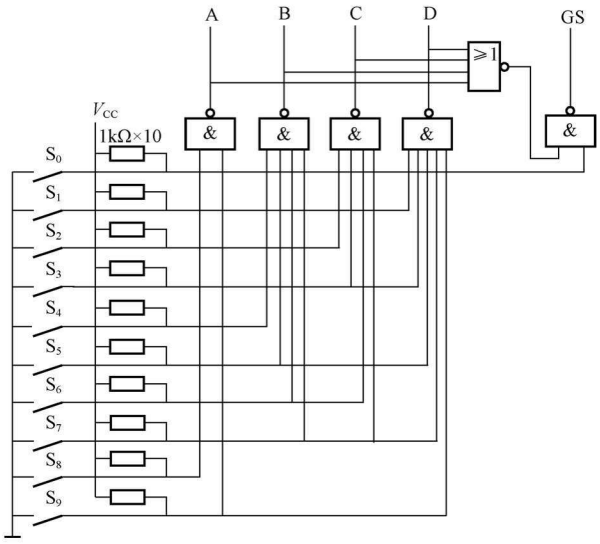


图 3 - 23 键控 8421BCD 码编码器的逻辑电路图

入为低电平有效，即某一按键按下，对应的输入信号为 0。输出为 4 位码，所以有 4 个输出端 A、B、C、D。输出代码正好对应 8421 码。当按下 S_0～S_9 中任意一个键时，使能端 GS=1，表示有信号输入；当 S_0～S_9 均没按下时，GS=0，表示没有信号输入，此时的输出代码 0000 为无效代码。使能端 GS 的存在，可以将真值表中第一行和第二行的两种情况区分开来。

二、优先编码器

优先编码器允许同时输入两个以上的编码信号，编码器给所有的输入信号规定了优先顺序，当多个输入信号同时出现时，只对其中优先级最高的一个进行编码。优先编码器不必对输入信号提出严格要求，而且使用可靠方便，目前得到广泛应用。例如计算机外设要工作时，都经编码器向主机提出申请时，这就需要把所有外设按轻重缓急进行排队，定出优先级别，并采用优先编码器。优先编码器也可以有二进制优先编码器和二—十进制优先编码器。下面首先介绍优先编码器的定义与功能，然后以二进制优先编码器 74X148 为例介绍这一类芯片的工作原理和使用方法。

1. 优先编码器的定义与功能

下面通过一个组合电路的设计问题来理解优先编码器的定义与功能。

【例 3 - 7】 设计一个 4 线—2 线优先编码器，要求任一时刻必须有一个输入有效，但允许多个输入同时有效。

解 (1) 约定。输入为高电平有效，信息有效用 1 表示，无效用 0 表示。4 个信息分别用 I_0、I_1、I_2、I_3 表示，2 位代码用 A_1、A_0 表示，且对应的关系为：I_0 的编码为 00（左边为 A_1、右边为 A_0），I_1 的编码为 01（左边为 A_1、右边为 A_0），I_2 的编码为 10（左边为 A_1、右边为 A_0），I_3 的编码为 11（左边为 A_1、右边为 A_0）。I_0、I_1、I_2、I_3 的优先级依次升高。

(2) 列出真值表，如表 3 - 7 所示。

表 3-7　　[例 3-7] 的真值表

输入				输出		输入				输出	
I_0	I_1	I_2	I_3	A_1	A_0	I_0	I_1	I_2	I_3	A_1	A_0
0	0	0	0	×	×	1	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1
0	0	1	0	1	0	1	0	1	0	1	0
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	0	1	1	1	0	0	0	1
0	1	0	1	1	1	1	1	0	1	1	1
0	1	1	0	1	0	1	1	1	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1

（3）用卡诺图化简，如图 3-24 所示。

（4）由卡诺图得最简与或逻辑表达式为

$$A_1 = I_2 + I_3$$

$$A_0 = I_3 + I_1 \overline{I_2}$$

根据逻辑表达式得到如图 3-25 所示的逻辑电路图。

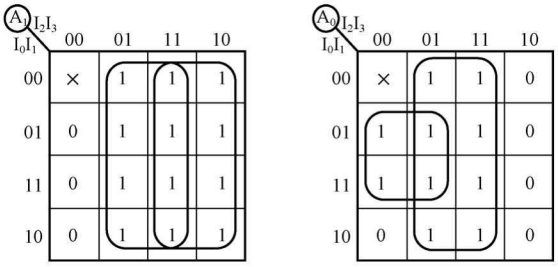


图 3-24　[例 3-7] 的卡诺图

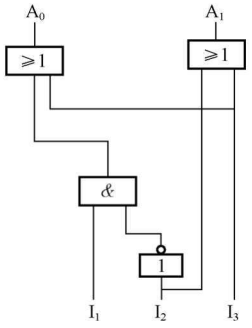


图 3-25　[例 3-7] 的逻辑电路图

2. 二进制优先编码器集成电路芯片 74X148

下面以 74X148 这个具体芯片为例，介绍优先编码器这一类集成电路芯片的功能，希望读者通过下面的学习，能够举一反三，触类旁通，掌握这一类芯片的功能和使用方法。74X148 中“X”是指 S、LS、ALS、HC、HCT 等芯片制造的各种工艺，这些不同工艺的芯片的逻辑功能、引脚排列完全相同，只是性能参数稍有不同。

74X148 芯片是一种常用的集成 8 线—3 线二进制优先编码器。从互联网上搜索 74X148 的数据手册（data sheet），可以获得有关这个芯片的所有信息。74X148 芯片是本书第一个详细解释的中规模（MSI）集成芯片。中规模集成芯片的种类很多，功能也千差万别，为了更好地记忆这些中规模集成芯片的功能，不至于在使用时频繁地去查阅这些芯片的使用手册，本书中电路的逻辑符号使用国家标准 GB/T 4728.12—1996，这个标准与美国国家标准委员会（ANSI）和电气电子工程师协会（IEEE）共同制定的的标准（ANSI/IEEE Std

91—1984，IEEE Standard Graphic Symbols for Logic Functions)、国际电工委员会（IEC）制定的标准 IEC617 完全兼容。使用该标准绘出的符号可以做到少用或不用其他参考文件就能确定所描述的逻辑电路的功能。

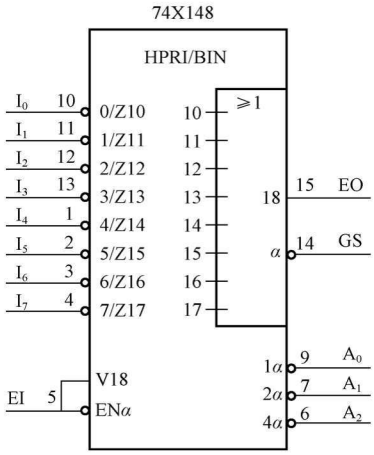


图 3 - 26　加了引脚名字的 74X148 的逻辑符号

中规模集成芯片的功能一般都比较复杂，要想全面了解其功能，就必须查阅生产芯片的厂家提供的数据手册，从功能表、内部逻辑电路图、引脚图、时序图等方面进行了解分析。为了能在其他设计中灵活方便地使用这些芯片，最好能记住这些芯片的功能和引脚，但是由于功能较复杂，芯片数目较多，一般不容易记忆，所以在本书中使用了国标逻辑符号。逻辑符号并没有规定引脚的名字，只用引脚的编号来区分，本书为了清晰起见，便于叙述，在逻辑符号的基础上还给定了一个引脚名。在国标逻辑符号中，一般情况下，输入信号标在左边，输出信号标在右边。

图 3 - 26 是加了引脚名字的 74X148 的逻辑符号，图 3 - 27 是 74X148 的内部逻辑电路图。

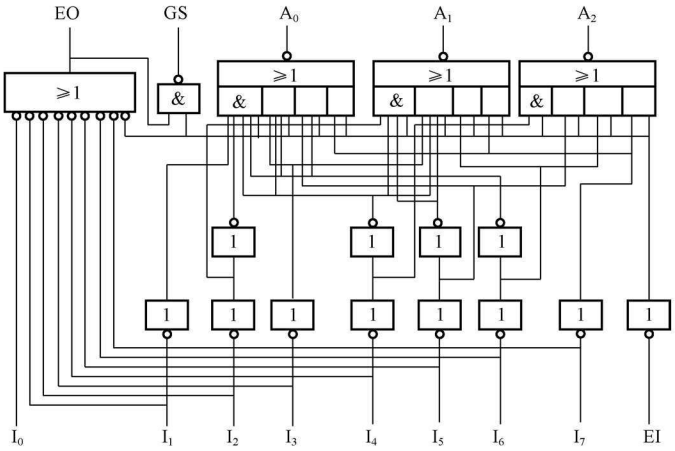


图 3 - 27　74X148 内部逻辑电路图

74X148 的真值表如表 3 - 8 所示。其中 $I_0 \sim I_7$ 为信号输入端，低电平有效；A_2、A_1、A_0 为编码输出端；I_7 的编码为 $A_2A_1A_0=000$，I_6 的编码为 $A_2A_1A_0=001$，I_5 的编码为 $A_2A_1A_0=010$，I_4 的编码为 $A_2A_1A_0=011$，I_3 的编码为 $A_2A_1A_0=100$，I_2 的编码为 $A_2A_1A_0=101$，I_1 的编码为 $A_2A_1A_0=110$，I_0 的编码为 $A_2A_1A_0=111$。本书中如不特别指明，全部使用正逻辑体制。

其他功能描述如下：

（1）EI 为使能输入端，低电平有效，即当 EI＝0 时，编码器工作，可以完成优先编码的功能；而当 EI＝1 时，则不论 8 个信号输入端为何种状态，3 个输出端 A_2、A_1、A_0 均为高电平，GS、EO 均为高电平，编码器处于非工作状态。

（2）优先顺序为 $I_7 \rightarrow I_0$，即 I_7 的优先级最高，然后是 I_6，I_5，…，I_0。输入信号为低电

平有效，当某一个输入端有低电平输入时，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信号的代码。例如，输入端 I_5 为 0，且优先级别比它高的 I_6 和 I_7 均为 1 时，输出代码为 010，即输出为输入 I_5 的编码，这就是优先编码器的工作原理。

（3）GS（Group Select）为编码器的工作标志，低电平有效。当输入使能端 EI 为 0（有效），编码器处于正常的工作状态时，$I_0 \sim I_7$ 信号输入端至少有一个处于有效时，优先编码器工作状态标志 GS 为 0，处于有效状态，表明编码器处于工作状态；当没有任何输入有效时，GS 为 1，处于无效状态。

（4）EO 为输出使能端，高电平有效。当输入使能端 EI 为 1（无效）时，输出使能端 EO 为 1（有效）。只有在 EI 为 0（有效），且 $I_0 \sim I_7$ 信号输入端无任何一个处于有效时，EO 输出 0，处于它的无效状态。

表 3-8　　优先编码器 74X148 的真值表

输入									输出				
EI	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A_2	A_1	A_0	GS	EO
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	0	0	0	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

根据真值表，可写出各个输出端的逻辑函数表达式。输出使能端 EO 的逻辑函数表达式为

$$\begin{aligned} EO &= EI + \overline{I_0} + \overline{I_1} + \overline{I_2} + \overline{I_3} + \overline{I_4} + \overline{I_5} + \overline{I_6} + \overline{I_7} \\ &= \overline{\overline{EI} \cdot I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7} \end{aligned} \tag{3-3}$$

编码器的工作标志 GS 的逻辑表达式

$$\begin{aligned} GS &= EI + \overline{EI} \cdot I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7 \\ &= EI + \overline{EO} \\ &= \overline{\overline{EI} \cdot EO} \\ &= \overline{\overline{EI}(EI + \overline{I_0} + \overline{I_1} + \overline{I_2} + \overline{I_3} + \overline{I_4} + \overline{I_5} + \overline{I_6} + \overline{I_7})} \end{aligned} \tag{3-4}$$

下面以 A_2 为例推导输出代码的逻辑函数表达式。A_2 的逻辑函数表达式为

$$\begin{aligned} A_2 = EI + \overline{EI}(&I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7 + \overline{I_0} I_1 I_2 I_3 I_4 I_5 I_6 I_7 \\ &+ \overline{I_1} I_2 I_3 I_4 I_5 I_6 I_7 + \overline{I_2} I_3 I_4 I_5 I_6 I_7 + \overline{I_3} I_4 I_5 I_6 I_7) \end{aligned}$$

利用 $A+\overline{A}B=A+B$ 和 $A+\overline{A}=1$ 的关系，化简得

$$A_2 = EI + I_4 I_5 I_6 I_7$$

加两次非号，变换可得输出信号 A_2 的逻辑函数表达式为

$$A_2 = \overline{(\overline{I_4} + \overline{I_5} + \overline{I_6} + \overline{I_7}) \cdot \overline{EI}} \tag{3-5}$$

同理，可得另外两个输出 A_1、A_0 的逻辑函数表达式为

$$A_1 = \overline{(\overline{I_2} I_4 I_5 + \overline{I_3} I_4 I_5 + \overline{I_6} + \overline{I_7}) \cdot \overline{EI}} \tag{3-6}$$

$$A_0 = \overline{(\overline{I_1} I_2 I_4 I_6 + \overline{I_3} I_4 I_6 + \overline{I_5} I_6 + \overline{I_7}) \cdot \overline{EI}} \tag{3-7}$$

不难看出，由上面分析得到的逻辑函数表达式与从逻辑电路图得到的逻辑函数表达式是一致的。

下面将解释 74X148 逻辑符号的含义。在逻辑符号中，具有同类功能的信号一般标注在一起，输入信号全部标注在长方形框的左边，输出信号全部标注在长方形框的右边。在图 3 - 26 所示的 74X148 的逻辑符号中，输入信号 EI、I_0、I_1、I_2、I_3、I_4、I_5、I_6、I_7 全部排列在左边，输出信号 A_2、A_1、A_0、GS、EO 全部排列在右边。国标符号规定，引脚只标有引脚的编号，引脚的名字是本书作者标上去的。在方框的外面、上部写有 74X148 的字样，表明是 74X148 芯片。

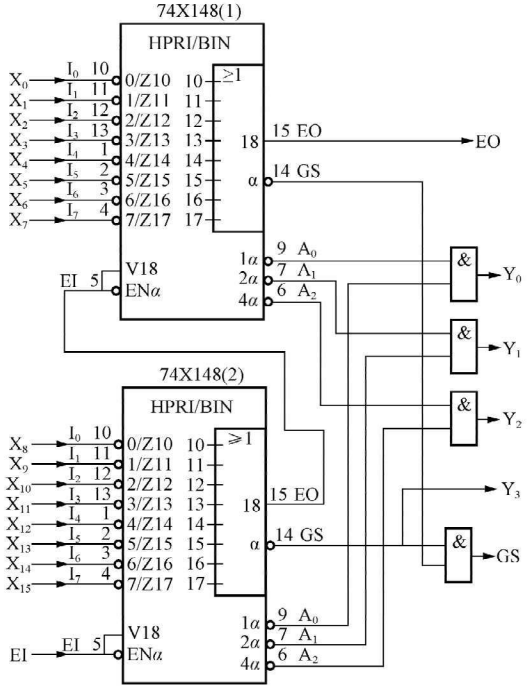


图 3 - 28　串行扩展实现的 16 线—4 线优先编码器

在图 3 - 26 所示的 74X148 的逻辑符号中，总定性符号 HPRI/BIN 表明该编码器是优先编码器，一般写在长方形框内部的上部。HPRI 是 HIGHEST PRIORITY 的缩写，PRIORITY 表示输入信号具有优先级的意思，HIGHEST 表示位权最大的高位输入信号的优先级别最高。“/”经常用来把编码器或译码器的输入和输出分开，BIN 是 BINARY 的缩写，表示把输入信号编成二进制码。芯片的 10、11、12、13、1、2、3、4 脚是 8 个被编码信号输入端，在长方形框的左上部，分别表示 10 号脚、11 号脚、12 号脚、13 号脚、1 号脚、2 号脚、3 号脚、4 号脚。为了描述方便、清晰起见，本书给 10 脚标上了 I_0 的名字，I 是 INPUT 的缩写，同理，11、12、13、1、2、3、4 脚的名字分别是 I_1、I_2、I_3、I_4、I_5、I_6、I_7，下标用来区分 8 个不同的输入信号。长方形框外部的小圆圈表示输入信号是低电平有效。长方形框的内部写有“0/Z10”，其中“/”用来分开一个内部信号的多种不同的功能，“/”左边的“0”是给这个信号一个序号。同理，“1/Z11”、“2/Z12”、…、“7/Z17”中“/”左边的数字也是分别给对应的引脚输入信号一个序号。“Z”是“ZIGZAG”的缩写，“Z”表示内部互连的意思，若“Z”后面的序号与其右边的数字相同，则表示它们是连接在一起的。“V18”中的“V”表示或的意思。“ENα”中的“EN”是使能的意思，表示 5 脚输入信号是使能信号。“ENα”中的“α”也是给这个使能信号一个序号，但是此处没有使用数字，而是使用了拉丁字母，主要是为了避免混淆。可以看到，如果将

方框中所有的“α”换成数字，则“1α”、“2α”、“4α”将发生混乱。“1α”、“2α”、“4α”中的数字是位权，表明8个输入信号与3个编码输出端信号的对应关系。从前面的叙述中已经知道，编码本身没有数量的概念，在此使用位权的概念，只是一种简便描述的手段。从表3-8中可以看出，I_7～I_0的编码与对应的二进制数正好相反，所以在逻辑符号方框的外部9脚、7脚、6脚处都画上了小圆圈。有效的编码输出是在使能端5脚“EN”有效的前提下得出的，用“α”来表示这种约束关系，同理，14脚GS输出有效也是在5脚“EN”有效的前提下得出的。

当输入使能端EI为1（无效）时，输出使能端EO为1（有效），在图3-26所示的74X148的逻辑符号中，就表示了这样的功能。74X148的5脚标有V18的字样，给5脚一个18的序号，V表示“或”的意思，从式（3-3）可以看出，EO与EI存在或的关系，只有在EI=0时，EO才具有其正常的功能，当EI=1时，74X148的15脚输出EO为1，靠近15脚内部的18的字样表明了这种依赖关系。

三、集成编码器的应用

1. 编码器的扩展

集成编码器的输入输出端的数目都是一定的，当需要更大规模的编码器时，利用编码器的输入使能端EI、输出使能端EO和优先编码工作标志GS，可以扩展编码器的输入输出端。

图3-28所示为用两片74X148优先编码器扩展实现的16线—4线优先编码器，表3-9为其真值表。

表3-9　　串行扩展实现的16线—4线优先编码器的真值表

输入																	输出					
EI	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	Y_3	Y_2	Y_1	Y_0	GS	EO
1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	0	0	0	0	0	0	1
0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	0	1	0	0	0	1	0	1
0	×	×	×	×	×	×	×	×	×	×	×	×	×	0	1	1	0	0	1	0	0	1
0	×	×	×	×	×	×	×	×	×	×	×	×	0	1	1	1	0	0	1	1	0	1
0	×	×	×	×	×	×	×	×	×	×	×	0	1	1	1	1	0	1	0	0	0	1
0	×	×	×	×	×	×	×	×	×	×	0	1	1	1	1	1	0	1	0	1	0	1
0	×	×	×	×	×	×	×	×	×	0	1	1	1	1	1	1	0	1	1	0	0	1
0	×	×	×	×	×	×	×	×	0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	×	×	×	×	×	×	×	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1
0	×	×	×	×	×	×	0	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1
0	×	×	×	×	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

图 3 - 28 所示的 16 线—4 线优先编码器共有 16 个编码输入端，用 $X_0 \sim X_{15}$ 表示；有 4 个编码输出端，用 $Y_0 \sim Y_3$ 表示；同时有 EI、EO、GS 等控制端，其功能与单片 74X148 芯片的功能非常类似。图中片（1）为低位片，其输入端作为总输入端的 $X_0 \sim X_7$；片（2）为高位片，其输入端作为总输入端的 $X_8 \sim X_{15}$。两片的输出端分别相与，作为总输出端 Y_0、Y_1、Y_2；片（2）的工作标志端作为总输出端 Y_3；片（1）的输出使能端作为电路总的输出使能端 EO；片（2）的输入使能端作为电路总的输入使能端 EI，在本电路中接 0，处于允许编码状态。片（2）的输出使能端 EO 接片（1）的输入使能端 EI，控制片（1）工作；两片的优先编码工作标志相与，作为总的优先编码工作标志 GS 端。

该编码器的工作原理为：当片（2）的输入端没有输入信号有效，即 $X_8 \sim X_{15}$ 全为 1 时，片（2）的 GS＝1（即 Y_3＝1），片（2）EO＝0［即片（1）的 EI＝0］，片（1）处于允许编码状态。设此时 X_5＝0，则片（1）的输出为 $A_2A_1A_0$＝010，由于片 2 输出 $A_2A_1A_0$＝111，所以总输出 $Y_3Y_2Y_1Y_0$＝1010。

当片（2）的输入端有输入信号有效，片（2）的 EO＝1［即片（1）的 EI＝1］，片（1）处于禁止编码状态。设此时 X_{12}＝0［即片（2）的 I_4＝0］，则片（2）的输出为 $A_2A_1A_0$＝011，且片（2）的 GS＝0。由于片（1）输出 $A_2A_1A_0$＝111，所以总输出 $Y_3Y_2Y_1Y_0$＝0011。

2. 组成 8421BCD 编码器

用 74X148 和门电路组成的 8421BCD 编码器如图 3 - 29 所示，具有十进制 10 个数码的输入端，$Y_3Y_2Y_1Y_0$ 为 8421BCD 码的输出端。$I_0 \sim I_9$ 输入仍为低电平有效。其工作原理为：

（1）当 I_8 和 I_9 无输入（即 I_8、I_9 均为高电平）时，与非门 G4 的输出 Y_3＝0，同时使 74X148 的 EI＝0，允许 74X148 工作，74X148 对输入 $I_0 \sim I_7$ 进行编码。例如，当 I_5 为低电平时，则 $A_2A_1A_0$＝010，经门 G1、G2、G3 处理后，$Y_2Y_1Y_0$＝101，所以总输出 $Y_3Y_2Y_1Y_0$＝0101。这正好是 5 的 8421BCD 码。

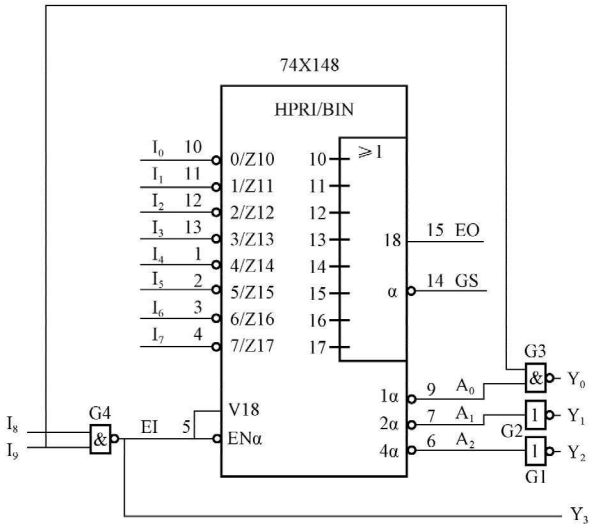


图 3 - 29　74X148 和门电路组成的 8421BCD 编码器的逻辑电路图

（2）当 I_9 或 I_8 有输入（低电平）时，与非门 G4 的输出 Y_3＝1，同时使 74X148 的 EI＝1，禁止 74X148 工作，使 $A_2A_1A_0$＝111。如果此时是 I_9＝0，总输出 $Y_3Y_2Y_1Y_0$＝1001；如

果此时是 $I_8=0$，总输出 $Y_3Y_2Y_1Y_0=1000$，正好分别是 9 和 8 的 8421BCD 码。

74X148 组成 8421BCD 编码器的真值表如表 3 - 10 所示。

表 3 - 10　　74X148 组成 8421BCD 编码器的真值表

输入										输出			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	Y_3	Y_2	Y_1	Y_0
×	×	×	×	×	×	×	×	×	0	1	0	0	1
×	×	×	×	×	×	×	×	0	1	1	0	0	0
×	×	×	×	×	×	×	0	1	1	0	1	1	1
×	×	×	×	×	×	0	1	1	1	0	1	1	0
×	×	×	×	×	0	1	1	1	1	0	1	0	1
×	×	×	×	0	1	1	1	1	1	0	1	0	0
×	×	×	0	1	1	1	1	1	1	0	0	1	1
×	×	0	1	1	1	1	1	1	1	0	0	1	0
×	0	1	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1	1	0	0	0	0

第五节　译　码　器

译码器有时也称之为解码器。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数码的组合转换成对应的信号。具有译码功能的逻辑电路称为译码器。常用的译码器可以分成通用译码器和显示译码器，而通用译码器又有二进制译码器和代码变换译码器之分。

一、二进制译码器

（一）二进制译码器的定义与功能

二进制译码器又称为变量译码器、全译码器、最小项译码器、唯一地址译码器。二进制译码器的 n 个输入信号和 N 个输出信号之间存在 $N=2^n$ 的关系，它常用于实现多输出的组合逻辑函数、计算机系统中的地址译码。

下面以 2 线—4 线译码器（即输入变量有 2 个，可译出 4 个输出信号）的设计为例，说明译码器的工作原理。

方法一：

（1）约定：两个输入信号分别用 A，B 表示，4 个输出信号分别用 Y_0、Y_1、Y_2、Y_3 表示，输出信号为低电平有效，AB=00、AB=01、AB=10、AB=11 分别对应 Y_0、Y_1、Y_2、Y_3 信号。

（2）按上述约定列出的真值表如表 3 - 11 所示。

表 3 - 11　　2 线—4 线译码器的解法一的真值表

输入		输出				输入		输出			
A	B	Y_0	Y_1	Y_2	Y_3	A	B	Y_0	Y_1	Y_2	Y_3
0	0	0	1	1	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1	1	1	1	0

（3）由真值表 3 - 11 可写出 Y_0、Y_1、Y_2、Y_3 各输出信号的逻辑表达式为

$$Y_0=\overline{\overline{A}\,\overline{B}},\quad Y_1=\overline{\overline{A}B},\quad Y_2=\overline{A\,\overline{B}},\quad Y_3=\overline{AB}$$

因为在各输出中 0 的个数少，所以可以先写出各输出信号的反函数，然后再求反得到原函数。

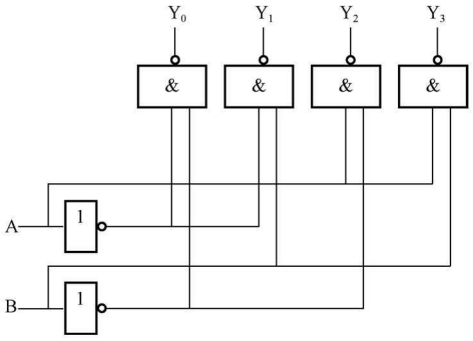


图 3 - 30　2 线—4 线译码器设计方法一的逻辑电路图

（4）用门电路实现 2 线—4 线译码器的逻辑电路如图 3 - 30 所示。

除上述约定外，完全可以作出另外的约定，下面按另外一种约定再做 2 线—4 线译码器设计。设计者还可根据需要做出其他的约定。

方法二：

（1）约定：两个输入信号分别用 A、B 表示，4 个输出信号分别用 Y_0、Y_1、Y_2、Y_3 表示，输出信号为高电平有效，AB＝01、AB＝00、AB＝11、AB＝10 分别对应 Y_0、Y_1、Y_2、Y_3 信号。

（2）按上述约定列出的真值表如表 3 - 12 所示。

表 3 - 12　　2 线—4 线译码器的解法二的真值表

输入		输出				输入		输出			
A	B	Y_0	Y_1	Y_2	Y_3	A	B	Y_0	Y_1	Y_2	Y_3
0	0	0	1	0	0	1	0	0	0	0	1
0	1	1	0	0	0	1	1	0	0	1	0

（3）由真值表 3 - 12 可写出 Y_0、Y_1、Y_2、Y_3 各输出函数表达式为

$$Y_0=\overline{A}B,\quad Y_1=\overline{A}\,\overline{B},\quad Y_2=AB,\quad Y_3=A\,\overline{B}$$

（4）用门电路实现 2 线—4 线译码器的逻辑电路如图 3 - 31 所示。

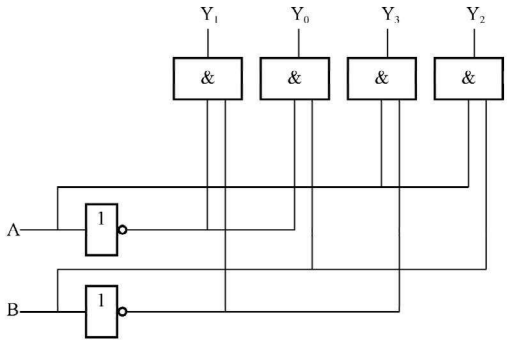


图 3 - 31　2 线—4 线译码器设计方法二的逻辑电路图

通过上面对 2 线—4 线译码器设计的介绍，希望读者能够了解译码器的基本概念和基本工作原理，同时还要体会二进制译码器的概念。两个输入二进制数码的 4 种组合均有对应的输出信号，这就是二进制译码器。

（二）典型的中规模集成电路芯片 74X138

74X138 是一种典型的二进制译码器。图 3 - 32 是加了引脚名字的 74X138 的逻辑符号，图 3 - 33 是 74X138 内部逻辑电路图。74X138 的真值表如表 3 - 13 所示。

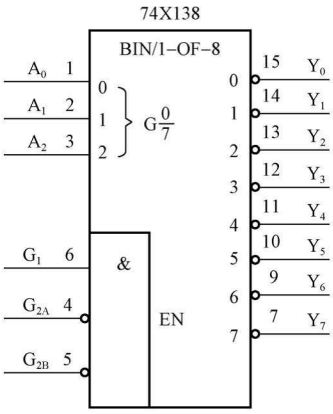


图 3 - 32　加了引脚名字的 74X138 的逻辑符号

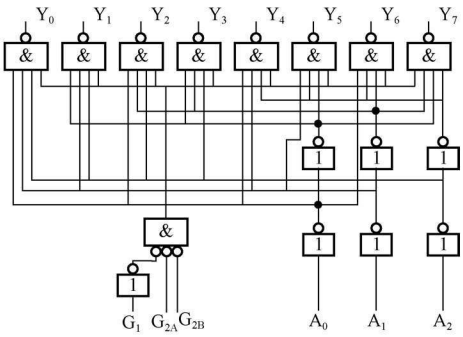


图 3 - 33　74X138 内部逻辑电路图

74X138 将 3 个二进制数码输入端 A_2、A_1、A_0 的 8 种组合翻译成对应的 8 个输出信号 Y_0～Y_7，对应关系如表 3 - 13 所示，所以常称其为 3 线—8 线译码器。它属于二进制译码器，输出为低电平有效。

表 3 - 13　　3 线—8 线译码器 74X138 真值表

输入						输出							
G_1	G_{2A}	G_{2B}	A_2	A_1	A_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×	1	×	×	×	×	1	1	1	1	1	1	1	1
×	×	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	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	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

生产厂家设计芯片的时候，除了最基本的功能外，为了方便用户使用，总是设计一些控制端，以满足不同用户的设计需求。在 74X138 中，设计了 3 个控制端 G_1、G_{2A} 和 G_{2B}，如图 3 - 32 所示，其中 G_1 为高电平有效，G_{2A} 和 G_{2B} 为低电平有效。当 G_1 为 1，G_{2A} 和 G_{2B} 均为 0 时，译码器处于工作状态。否则译码器被禁止，所有的输出端被封锁在高电平。

由真值表或者逻辑电路图，可以得到 74X138 的 8 个输出的逻辑表达式为

$$
\begin{cases}
Y_0(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_0}\\
Y_1(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,\overline{A_1}\,A_0}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_1}\\
Y_2(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,A_1\,\overline{A_0}}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_2}\\
Y_3(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,A_1\,A_0}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_3}\\
Y_4(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,\overline{A_1}\,\overline{A_0}}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_4}\\
Y_5(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,\overline{A_1}\,A_0}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_5}\\
Y_6(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,A_1\,\overline{A_0}}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_6}\\
Y_7(A_2,A_1,A_0)=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,A_1\,A_0}=\overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,m_7}
\end{cases}
\tag{3-8}
$$

当 $G_1=1$、$G_{2A}=0$ 和 $G_{2B}=0$ 时 8 个输出的逻辑表达式为

$$
\begin{cases}
Y_0(A_2,A_1,A_0)=\overline{\overline{A_2}\,\overline{A_1}\,\overline{A_0}}=\overline{m_0}\\
Y_1(A_2,A_1,A_0)=\overline{\overline{A_2}\,\overline{A_1}\,A_0}=\overline{m_1}\\
Y_2(A_2,A_1,A_0)=\overline{\overline{A_2}\,A_1\,\overline{A_0}}=\overline{m_2}\\
Y_3(A_2,A_1,A_0)=\overline{\overline{A_2}\,A_1\,A_0}=\overline{m_3}\\
Y_4(A_2,A_1,A_0)=\overline{A_2\,\overline{A_1}\,\overline{A_0}}=\overline{m_4}\\
Y_5(A_2,A_1,A_0)=\overline{A_2\,\overline{A_1}\,A_0}=\overline{m_5}\\
Y_6(A_2,A_1,A_0)=\overline{A_2\,A_1\,\overline{A_0}}=\overline{m_6}\\
Y_7(A_2,A_1,A_0)=\overline{A_2\,A_1\,A_0}=\overline{m_7}
\end{cases}
\tag{3-9}
$$

注意：在式（3-9）中，将 A_2、A_1、A_0 组成的与项看作最小项，以 A_2、A_1、A_0 的顺序，A_2 为高位对最小项编号时，8 个输出信号 Y 的下标与 8 个最小项 m 的下标一致，这样便于记忆这些输出的表达式。很容易可以验证，如果以其他的方式对最小项编号，输出信号 Y 的下标与最小项 m 的下标不可能完全一致，而且没有规律。

利用 74X138 译码器的每个输出信号分别对应一个最小项的非运算这个特点，可以方便地实现 3 变量多输出组合逻辑函数。从这个角度说，这种译码器又叫做最小项译码器。

下面介绍图 3-32 所示的 74X138 逻辑符号的含义。图中，输入信号 A_0、A_1、A_2、G_1、G_{2A}、G_{2B}全部排列在左边，输出信号 Y_0、Y_1、Y_2、Y_3、Y_4、Y_5、Y_6、Y_7 全部排列在右边。BIN/1-OF-8 是总限定符，BIN 是 BINARY 的缩写，表明是二进制译码器，1-OF-8 表明任意瞬间 8 个输出信号中只能有 1 个输出信号处于有效状态。在方框内部靠近输入信号 A_0、A_1、A_2 处分别标有 0、1、2，这是为了便于解释芯片的功能而人为给信号规定的位权，规定输入信号 A_0 的位权为 2^0，也就是 1；规定输入信号 A_1 的位权为 2^1，也就是 2；规定输入信号 A_2 的位权为 2^2，也就是 4。因为这是一个二进制译码器，所以此处标明的是位权的幂。如果不是二进制译码器，就必须直接标明位权，无法用幂来表示。大括号叫做分组符号（bit-grouping symbol），表明组合在一起的这些二进制位产生一个内部值（internal value），由一组二进制位产生的内部值影响其他的内部值或输出信号，这个内部值是人为规定了位权之后的和。如果所有位的位权都是 2 的幂，则这些位权就可以只用幂来表示，而将底数省略不写。G 表示和内部值的范围在 0～7 内的信号有关联。当 3 脚 A_2、2 脚 A_1、1 脚 A_0 输入

110 时，内部值就是 6，这时标有 6 字样的输出信号内部值为 1，在方框的外部有小圆圈，所以 9 脚 Y_6 输出 0，在正逻辑体制下，输出就是低电平。

（三）二进制译码器的应用

1. 二进制译码器的扩展

利用译码器的使能端可以方便地扩展译码器的容量。图 3 - 34 是将两片 74X138 扩展为 4 线—16 线译码器的电路图，表 3 - 14 是其真值表。DCBA 是四位二进制代码输入端，Z_0～Z_{15} 是 16 个信号输出端。由于 74X138 仅有 3 个地址输入端 $A_2A_1A_0$，如果想对四位二进制代码进行译码，可以利用一个附加控制端（G_1、G_{2A} 和 G_{2B} 中的一个）作为第四个二进制数码的输入端。图 3 - 34 的连接方案是将 74X138（1）的 G_{2A} 和 74X138（2）的 G_1 连接在一起作为第四个二进制数码的输入端 D，同时 74X138（1）的 G_1 接高电平，74X138（1）的 G_{2B} 和 74X138（2）的 G_{2A}、G_{2B} 连接在一起作为使能端 EN。于是，当 EN＝0 时，得到两片 74X138 的输出信号分别为

$$\begin{cases} Z_0 = \overline{\overline{D}\,\overline{C}\,\overline{B}\,\overline{A}} \\ Z_1 = \overline{\overline{D}\,\overline{C}\,\overline{B}A} \\ \quad\vdots \\ Z_7 = \overline{\overline{D}CBA} \end{cases}, \quad \begin{cases} Z_8 = \overline{D\,\overline{C}\,\overline{B}\,\overline{A}} \\ Z_9 = \overline{D\,\overline{C}\,\overline{B}A} \\ \quad\vdots \\ Z_{15} = \overline{DCBA} \end{cases} \tag{3-10}$$

当 EN＝0 且 D＝0 时，则 74X138（2）禁止，74X138（1）工作，此时将输入 DCBA 的 0000～0111 这 8 个代码译成 Z_0～Z_7 8 个低电平输出信号。当 EN＝0 且 D＝1 时，则 74X138（1）禁止，74X138（2）工作，将输入 DCBA 的 1000～1111 这 8 个代码译成 Z_8～Z_{15} 8 个低电平输出信号，从而实现了 4 线—16 线译码器功能。

表 3 - 14　　4 线—16 线译码器的真值表

输入					输出															
EN	D	C	B	A	Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}
1	×	×	×	×	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

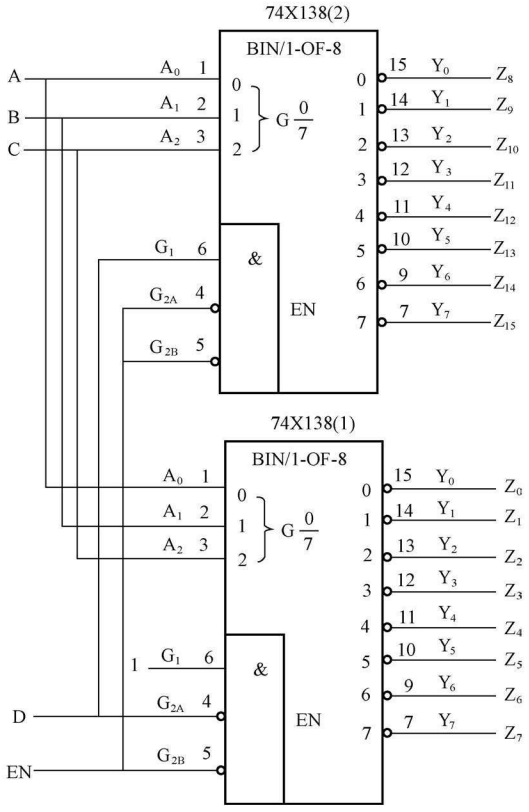


图 3 - 34　两片 74X138 扩展为 4 线—16 线译码器

2. 实现多输出组合逻辑函数

前面已经详细讲述了二进制译码器的电路结构和工作原理。由 3 线—8 线译码器 74X138 的真值表可以看出，当控制使能端 $G_1G_{2A}G_{2B}$ 为 100 时，若将 A_2、A_1、A_0 作为 3 个输入逻辑变量，则 8 个输出端 $Y_0 \sim Y_7$ 就是这 3 个输入变量的全部最小项的非，分别是 $\overline{m}_0 \sim \overline{m}_7$。利用附加的门电路将这些最小项适当地组合起来，便可产生任意的三变量组合逻辑函数。

将之推广可得，由于 n 位二进制译码器的输出给出了 n 变量的全部最小项，因而用 n 变量二进制译码器和“或门”（当译码器的输出为原变量 $m_0 \sim m_{2^n-1}$ 时）或者“与非门”（当译码器的输出为反变量 $\overline{m_0} \sim \overline{m_{2^n-1}}$ 时），则一定能实现任何形式的输入变量数目小于等于 n 的组合逻辑函数。

【例 3 - 8】 某多输出组合逻辑函数的真值表如表 3 - 15 所示，试用 74X138 译码器和必要的门电路实现该多输出组合逻辑函数。

表 3 - 15　　［例 3 - 8］ 的 真 值 表

输入			输出			输入			输出		
A	B	C	L	F	G	A	B	C	L	F	G
0	0	0	0	0	1	1	0	0	1	0	1
0	0	1	1	0	0	1	0	1	0	1	0
0	1	0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	0	1	1	1	1	0	0

解　(1) 由真值表可知，该组合逻辑函数具有 3 个输入变量，所以选用一片 74X138 即可。又由于 74X138 的译码输出是以最小项的非 $\overline{m}_0 \sim \overline{m}_7$ 的形式给出的，所以需要将逻辑函数转换成 $\overline{m}_0 \sim \overline{m}_7$ 的函数形式，即先写出各输出的最小项表达式为

$$L(A,B,C) = \overline{A}\,\overline{B}C + \overline{A}B\,\overline{C} + A\,\overline{B}\,\overline{C} + ABC = m_1 + m_2 + m_4 + m_7 = \overline{\overline{m_1}\,\overline{m_2}\,\overline{m_4}\,\overline{m_7}}$$

$$F(A,B,C) = \overline{A}BC + A\,\overline{B}C + AB\,\overline{C} = m_3 + m_5 + m_6 = \overline{\overline{m_3}\,\overline{m_5}\,\overline{m_6}}$$

$$G(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\,\overline{C} + A\,\overline{B}\,\overline{C} + AB\,\overline{C} = m_0 + m_2 + m_4 + m_6 = \overline{\overline{m_0}\,\overline{m_2}\,\overline{m_4}\,\overline{m_6}}$$

(2) 当 $G_1=1$、$G_{2A}=0$ 和 $G_{2B}=0$ 时，74X138 的 8 个输出的逻辑表达式如式（3 - 9）所示。

（3）将 $A=A_2$、$B=A_1$、$C=A_0$，L、F、G 的逻辑表达式与 74X138 的输出表达式相比较，有

$$L=\overline{Y_1Y_2Y_4Y_7}$$

$$F=\overline{Y_3Y_5Y_6}$$

$$G=\overline{Y_0Y_2Y_4Y_6}$$

用 1 片 74X138 加 3 个与非门就可实现该多输出组合逻辑函数，逻辑电路图如图3 - 35 所示。可见，用译码器实现多输出组合逻辑函数时，优点更明显。

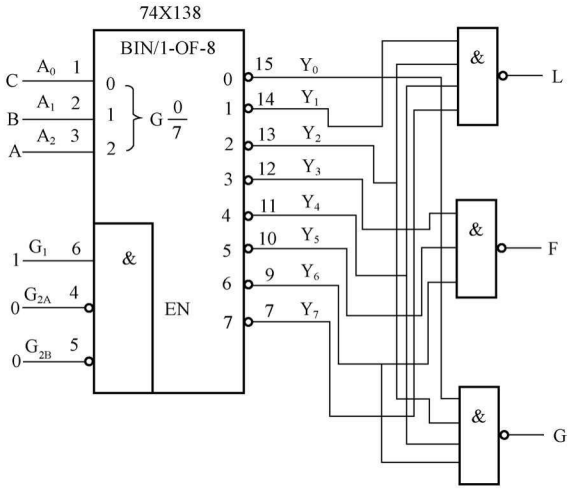


图 3 - 35　［例 3 - 8］的逻辑电路图

3. 二进制译码器可以作为数据分配器使用

数据分配是将来自一个数据源的数据，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。数据分配器的功能相当于多个输出的单刀多掷开关，有一个输入端和多个输出端，输入数据分配到哪个输出端上，则是由加到通道选择输入端上的通道选择信号（地址选择信号）决定的，如图 3 - 36 所示。表 3 - 16 是当 $n=3$ 时数据分配器的功能表。

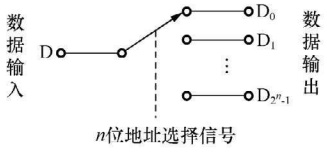


图 3 - 36　数据选择器功能示意图

表 3 - 16　　$n=3$ 时数据分配器功能表

地址选择信号			输　出	地址选择信号			输　出	地址选择信号			输　出
A_2	A_1	A_0		A_2	A_1	A_0		A_2	A_1	A_0	
0	0	0	$D_0=D$	0	1	1	$D_3=D$	1	1	0	$D_6=D$
0	0	1	$D_1=D$	1	0	0	$D_4=D$	1	1	1	$D_7=D$
0	1	0	$D_2=D$	1	0	1	$D_5=D$				

二进制译码器一个很重要的应用就是构成数据分配器。也正因为如此，市场上没有集成数据分配器产品，只有集成译码器产品。当需要数据分配器时，可以用二进制译码器改接。带选通端的二进制译码器可用作数据分配器，在用二进制译码器改接成数据分配器时，与数据通道 D 相对应的是译码器的选通控制信号端，与地址选择信号相对应的是二进制译码器输入的二进制代码，即集成数据分配器就是带选通控制端（又称为使能端）的集成二进制译码器。只要在使用时把二进制译码器的选通控制端当作数据输入端、二进制代码输入端当作选择控制端即可。例如，74X138 是集成 3 线—8 线译码器，也是集成 1 路—8 路数据分配器。74X139 是集成 2 线—4 线译码器，也是集成 1 路—4 路数据分配器。

下面以 74X138 为例讲解用二进制译码器做数据分配器的方法。用 74X138 作为数据分配器有三种方案可供选择。

（1）第一种方案：G_{2A}为数据输入端 D，Y_0、Y_1、Y_2、Y_3、Y_4、Y_5、Y_6、Y_7 分别为数据输出端 D_0、D_1、D_2、D_3、D_4、D_5、D_6、D_7、A_2、A_1、A_0 为地址选择信号，其逻辑电路图如图 3 - 37 所示。

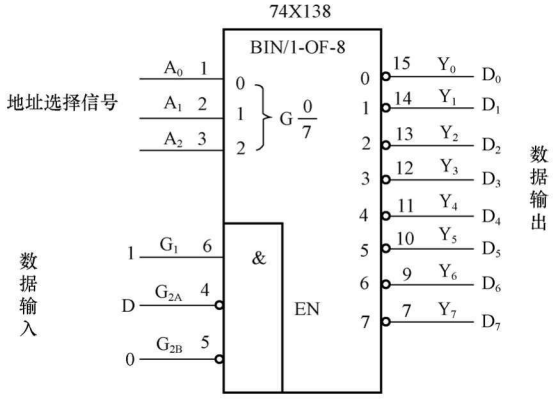


图 3 - 37　方案一的逻辑电路图

在 $G_{2B}=0$、$G_1=1$ 的前提下，74X138 的输出的逻辑表达式为

$$
\begin{aligned}
Y_0 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}} = \overline{\overline{G_{2A}}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}} \\
Y_1 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,\overline{A_1}\,A_0} = \overline{\overline{G_{2A}}\,\overline{A_2}\,\overline{A_1}\,A_0} \\
Y_2 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,A_1\,\overline{A_0}} = \overline{\overline{G_{2A}}\,\overline{A_2}\,A_1\,\overline{A_0}} \\
Y_3 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,\overline{A_2}\,A_1\,A_0} = \overline{\overline{G_{2A}}\,\overline{A_2}\,A_1\,A_0} \\
Y_4 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,\overline{A_1}\,\overline{A_0}} = \overline{\overline{G_{2A}}\,A_2\,\overline{A_1}\,\overline{A_0}} \\
Y_5 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,\overline{A_1}\,A_0} = \overline{\overline{G_{2A}}\,A_2\,\overline{A_1}\,A_0} \\
Y_6 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,A_1\,\overline{A_0}} = \overline{\overline{G_{2A}}\,A_2\,A_1\,\overline{A_0}} \\
Y_7 &= \overline{G_1\,\overline{G_{2A}}\,\overline{G_{2B}}\,A_2\,A_1\,A_0} = \overline{\overline{G_{2A}}\,A_2\,A_1\,A_0}
\end{aligned}
\tag{3-11}
$$

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=000$ 时 74X138 的输出分别是 $Y_0=G_{2A}$、$Y_1=Y_2=Y_3=Y_4=Y_5=Y_6=Y_7=1$。数据从 Y_0 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=001$ 时 74X138 的输出分别是 $Y_1=G_{2A}$、

$Y_0=Y_2=Y_3=Y_4=Y_5=Y_6=Y_7=1$。数据从 Y_1 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=010$ 时 74X138 的输出分别是 $Y_2=G_{2A}$、$Y_0=Y_1=Y_3=Y_4=Y_5=Y_6=Y_7=1$。数据从 Y_2 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=011$ 时 74X138 的输出分别是 $Y_3=G_{2A}$、$Y_0=Y_1=Y_2=Y_4=Y_5=Y_6=Y_7=1$。数据从 Y_3 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=100$ 时 74X138 的输出分别是 $Y_4=G_{2A}$、$Y_0=Y_1=Y_2=Y_3=Y_5=Y_6=Y_7=1$。数据从 Y_4 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=101$ 时 74X138 的输出分别是 $Y_5=G_{2A}$、$Y_0=Y_1=Y_2=Y_3=Y_4=Y_6=Y_7=1$。数据从 Y_5 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=110$ 时 74X138 的输出分别是 $Y_6=G_{2A}$、$Y_0=Y_1=Y_2=Y_3=Y_4=Y_5=Y_7=1$。数据从 Y_6 输出。

在 $G_{2B}=0$、$G_1=1$ 的前提下，当 $A_2A_1A_0=111$ 时 74X138 的输出分别是 $Y_7=G_{2A}$、$Y_0=Y_1=Y_2=Y_3=Y_4=Y_5=Y_6=1$。数据从 Y_7 输出。

（2）第二种方案：G_{2B}作为数据输入端 D，Y_0、Y_1、Y_2、Y_3、Y_4、Y_5、Y_6、Y_7 作为数据输出端 D_0、D_1、D_2、D_3、D_4、D_5、D_6、D_7，A_2、A_1、A_0 作为地址选择信号。逻辑电路图如图 3 - 38 所示。

（3）第三种方案：G_1 作为数据输入端 D，Y_0、Y_1、Y_2、Y_3、Y_4、Y_5、Y_6、Y_7 作为数据输出端 D_0、D_1、D_2、D_3、D_4、D_5、D_6、D_7，G_1 输入的数据被反相后输出，A_2、A_1、A_0 作为地址选择信号。逻辑电路图如图3 - 39 所示。

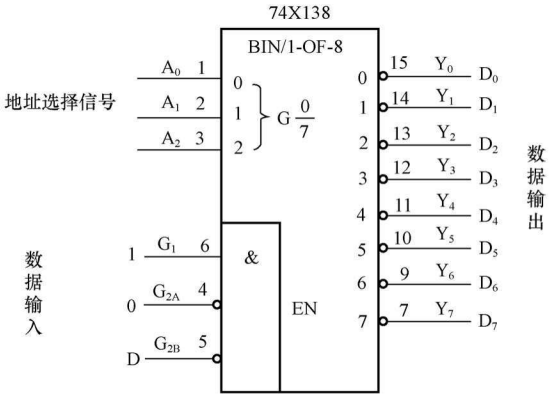


图 3 - 38　方案二的逻辑电路图

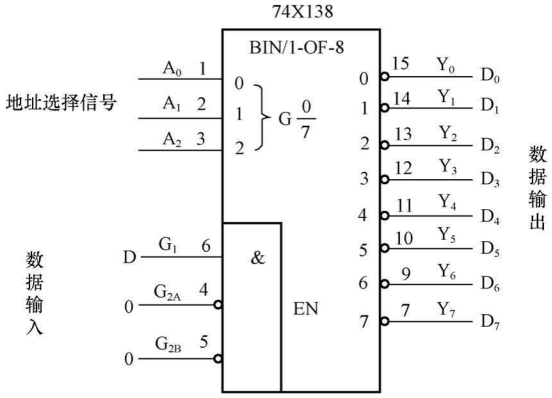


图 3 - 39　方案三的逻辑电路图

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；用它与数据选择器连接组成分时数据传送系统；另外它还可以与计数器结合组成脉冲分配器。

二、代码变换译码器

代码变换译码器的一个最典型的例子就是二—十进制译码器。二—十进制译码器有 4 个输入信号，用于输入 BCD 码的四位二进制码；有 10 个输出信号，分别对应表示十进制的 10 个数码的输出信号。这类译码器主要是用来解决人们不习惯直接识别二进制数的现实问题。

二—十进制译码器的典型芯片是 74X42。图 3 - 40 是加了引脚名字的 74X42 的逻辑符

号，图 3 - 41 是 74X42 的内部逻辑电路图。74X42 的真值表如表 3 - 17 所示。

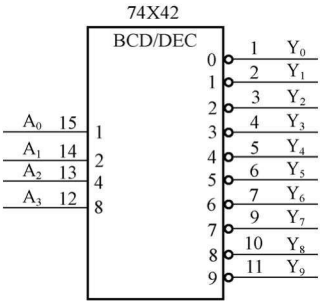


图 3 - 40　加了引脚名字的 74X42 的逻辑符号

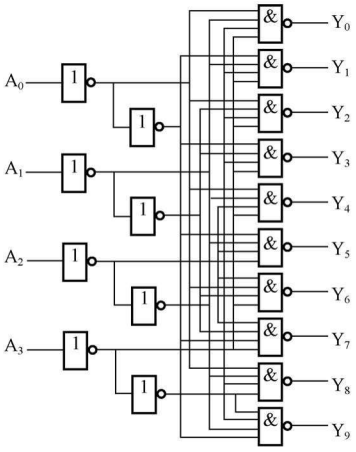


图 3 - 41　74X42 内部逻辑电路图

表 3 - 17　　二—十进制译码器 74X42 的真值表

序　号	输　入				输　出									
	A_3	A_2	A_1	A_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0
伪　码	1	0	1	0	1	1	1	1	1	1	1	1	1	1
	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	1	1	0	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	0	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1

根据芯片内部逻辑图可得逻辑表达式为

$$\begin{cases} Y_0 = \overline{\overline{A_3}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}} \\ Y_1 = \overline{\overline{A_3}\,\overline{A_2}\,\overline{A_1}\,A_0} \\ Y_2 = \overline{\overline{A_3}\,\overline{A_2}\,A_1\,\overline{A_0}} \\ Y_3 = \overline{\overline{A_3}\,\overline{A_2}\,A_1\,A_0} \\ Y_4 = \overline{\overline{A_3}\,A_2\,\overline{A_1}\,\overline{A_0}} \\ Y_5 = \overline{\overline{A_3}\,A_2\,\overline{A_1}\,A_0} \\ Y_6 = \overline{\overline{A_3}\,A_2\,A_1\,\overline{A_0}} \\ Y_7 = \overline{\overline{A_3}\,A_2\,A_1\,A_0} \\ Y_8 = \overline{A_3\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}} \\ Y_9 = \overline{A_3\,\overline{A_2}\,\overline{A_1}\,A_0} \end{cases} \tag{3-12}$$

对应 BCD 代码以外的伪码（即 1010～1111 共 6 个代码），Y_0～Y_9 均无低电平信号产生，译码器拒绝“翻译”，即这时的输出均为无效的高电平，没有任何信号被翻译出来，所以这个电路结构具有拒绝伪码的功能。

在图 3-40 所示的 74X42 的逻辑符号中，输入信号 A_0、A_1、A_2、A_3 全部排列在左边，输出信号 Y_0～Y_9 全部排列在右边。BCD/DEC 是总限定符，BCD 是 BINARY-CODED DECIMAL 的缩写，表明是二进制编码的十进制；DEC 是 DECIMAL 的缩写，表明是十进制数。在方框内部靠近输入信号 A_0、A_1、A_2、A_3 处分别标有 1、2、4、8，这是为了便于解释芯片的功能，而人为给信号规定的位权。规定输入信号 A_0 的位权为 1，A_1 的位权为 2，A_2 的位权为 4，A_3 的位权为 8。因为这不是一个二进制译码器，所以此处标明的不是位权的幂。例如，当 12 脚 A_3、13 脚 A_2、14 脚 A_1、15 脚 A_0 输入 1001 时，内部值就是 9，这时标有 9 字样的输出信号内部值为 1，在方框的外部有小圆圈，所以 11 脚 Y_9 输出 0，在正逻辑体制下，输出就是低电平。

三、显示译码器

在数字系统中，常常需要将数字量直观地显示出来，供人们读取或监视系统的工作情况。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显示译码器、驱动器和显示器等部分组成。在数字电路中，数字量都是以一定的代码形式出现的，所以这些数字量要先经过译码，才能送到数字显示器去显示。这种能把数字量翻译成数字显示器所能识别的信号的译码器称为数字显示译码器。通常，显示译码器也包含了驱动的功能。

常用的数字显示器有多种类型。

按显示方式分，有字形重叠式、点阵式、分段式等。所谓字形重叠式，是将不同字符的电极重叠起来，要显示某字符，只需使相应的电极发亮即可，如辉光放电管、边光显示管等。而所谓分段式，数码是由分布在同一平面上若干段发光的笔画组成，如荧光数码管等。所谓点阵式，是由一些按一定规律排列的可发光的点阵组成，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。

按发光物质分，有半导体显示器，又称发光二极管（LED）显示器；荧光显示器，如荧光数码管、场致发光数字板等；液晶显示器；气体放电管显示器，如辉光数码管、等离子体

显示板等。

目前应用最广泛的是由发光二极管构成的七段数码显示器。

（一）七段数码显示器原理

七段数码显示器就是将 7 个（加上小数点为 8 个）发光二极管按一定的方式排列起来，七段 a、b、c、d、e、f、g（小数点 DP）各对应一个发光二极管，发光二极管按照需要制作成条形或者圆点形。利用不同发光段的组合，显示不同的阿拉伯数字，如图 3 - 42 所示。在这 16 个字型中，6 可以如图中所示对应 c、d、e、f、g 段发光，还可以对应 a、c、d、e、f、g 段发光；9 可以如图中所示对应 a、b、c、f、g 段发光，还可以对应 a、b、c、d、f、g 段发光。在图 3 - 43 中段组合图 9 字形后面的字形依次为 10、11、12、13、14、15，但是这种表示方法非常不习惯，所以 10、11、12、13、14、15 经常用两位数码管显示，这样更符合人们日常生活中的习惯。

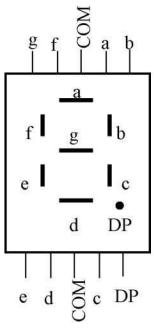


图 3 - 42　七段数码显示器

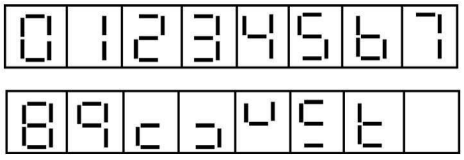


图 3 - 43　七段数码发光段组合图

按内部连接方式不同，七段数码显示器分为共阳极接法和共阴极接法两种，共阳极接法如图 3 - 44 所示。共阳极七段数码显示器的 8 个发光二极管的阳极接在一起，然后与高电平相接，要想让某个发光二极管发光，只要将这个发光二极管的阴极接低电平即可，若不想让某个发光二极管发光，只要将这个发光二极管的阴极接高电平即可。

共阴极七段数码显示器的 8 个发光二极管的阴极接在一起，然后与低电平相接，要想让某个发光二极管发光，只要将这个发光二极管的阳极接高电平即可。

共阳极七段数码显示器必须与输出信号为低电平有效的显示译码器配套使用，共阴极七段数码显示器必须与输出信号为高电平有效的显示译码器配套使用。

七段数码显示器的优点是工作电压较低（1.5～3V）、体积小、寿命长、亮度高、响应速度快、工作可靠性高；缺点是工作电流大，每个字段的工作电流约为 10mA 左右。

（二）七段显示译码器 74X48

七段显示译码器 74X48 是一种与共阴极数码显示器配合使用的集成译码器，它的功能是将输入的 4 位二进制代码转换成显示器所需要的 7 个段信号 a、b、c、d、e、f、g 输出为高电平有效。图 3 - 45 是加了引脚名字的 74X48 的逻辑符号。

表 3 - 18 是 74X48 的真值表。表中 A_3、A_2、A_1、A_0 是输入的 4 位二进制代码，a、b、c、d、e、f、g 为译码输出端，送给七段数码显示器进行显示。另外，它还有 3 个控制端：灯测试输入端 LT（lamp testing）、灭零输入端 RBI（ripple blanking input）、消隐输入端/灭零输出端 BI/RBO（blanking input/ripple blanking output）。

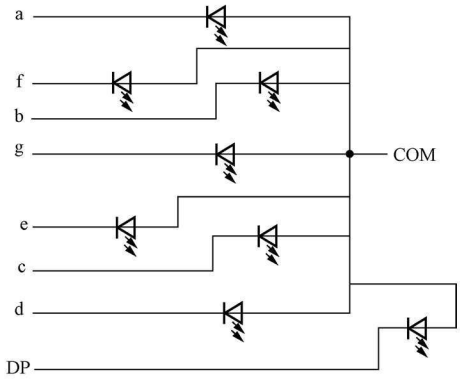


图 3 - 44　七段数码显示器的共阳极接法

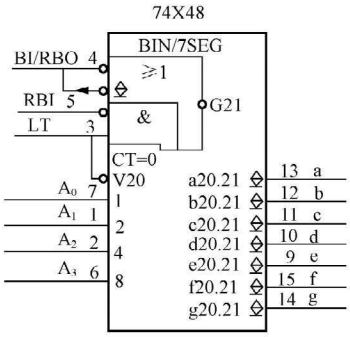


图 3 - 45　加了引脚名字的 74X48 的逻辑符号

表 3 - 18　　七段显示译码器 74X48 的逻辑真值表

功　能	输　入						输入/输出	输　出							显示字形
	LT	RBI	A_3	A_2	A_1	A_0	BI/RBO	a	b	c	d	e	f	g	
0	1	1	0	0	0	0	1（出）	1	1	1	1	1	1	0	0
1	1	×	0	0	0	1	1（出）	0	1	1	0	0	0	0	1
2	1	×	0	0	1	0	1（出）	1	1	0	1	1	0	1	2
3	1	×	0	0	1	1	1（出）	1	1	1	1	0	0	1	3
4	1	×	0	1	0	0	1（出）	0	1	1	0	0	1	1	4
5	1	×	0	1	0	1	1（出）	1	0	1	1	0	1	1	5
6	1	×	0	1	1	0	1（出）	0	0	1	1	1	1	1	6
7	1	×	0	1	1	1	1（出）	1	1	1	0	0	0	0	7
8	1	×	1	0	0	0	1（出）	1	1	1	1	1	1	1	8
9	1	×	1	0	0	1	1（出）	1	1	1	0	0	1	1	9
10	1	×	1	0	1	0	1（出）	0	0	0	1	1	0	1	c
11	1	×	1	0	1	1	1（出）	0	0	1	1	0	0	1	ɔ
12	1	×	1	1	0	0	1（出）	0	1	0	0	0	1	1	u
13	1	×	1	1	0	1	1（出）	1	0	0	1	0	1	1	c̣
14	1	×	1	1	1	0	1（出）	0	0	0	1	1	1	1	t
15	1	×	1	1	1	1	1（出）	0	0	0	0	0	0	0	
灭灯	×	×	×	×	×	×	0（入）	0	0	0	0	0	0	0	
灭零	1	0	0	0	0	0	0（出）	0	0	0	0	0	0	0	
试灯	0	×	×	×	×	×	1（出）	1	1	1	1	1	1	1	

灯测试输入端 LT 是一个输入信号，低电平有效；灭零输入端 RBI 是一个输入信号，低电平有效；消隐输入端/灭零输出端 BI/RBO 是一个双向信号，分时使用这个引脚来作消隐输入端和灭零输出端，其中做消隐输入端时是低电平有效，做灭零输出端时是低电平有效。下面就这三个控制端的功能分别介绍。

1. 消隐输入端/灭零输出端 BI/RBO

在消隐输入端/灭零输出端 BI/RBO 给定一个低电平输入信号，即作为输入端 BI 来使用

时，不管灯测试输入端 LT 和灭零输入端 RBI 处于有效还是无效的状态，不管二进制数码输入端 A_3、A_2、A_1、A_0 为何值，a～g 均输出 0，显示器全灭。这种情况主要是为了节省电能来设计的，很多情况下我们可能并不需要观察七段数码显示器的值，不妨让它们处于熄灭的状态，在想观察的时候，通过一定的控制手段，让它们再点亮。可以这么说，在这 3 个控制信号中，消隐输入端 BI 具有最高的优先级别。如表 3 - 18 的倒数第三行所示。

消隐输入端/灭零输出端 BI/RBO 不接输入信号，将作为输出信号来使用。

2. 灯测试输入端 LT

当消隐输入端/灭零输出端 BI/RBO 不作为输入端使用、灭零输入端 RBI 处于有效或者无效的状态下，只要在灯测试输入端 LT 给定一个低电平输入信号，无论输入 A_3、A_2、A_1、A_0 为任何值，a～g 输出全 1，数码显示器七段全亮。用这种方法可以检测显示器 7 个发光段的好坏。在这 3 个控制信号中，灯测试输入端 LT 比消隐输入端 BI 优先级别低，比灭零输入端 RBI 优先级别高，具有第二优先级别。在进行灯测试的过程中，消隐输入端/灭零输出端 BI/RBO 是作为输出信号来使用的，处于它的无效输出状态，即输出一个高电平。如表 3 - 18 的倒数第一行所示。

3. 灭零输入端 RBI

在消隐输入端/灭零输出端 BI/RBO 不作为输入端使用，且灯测试输入端 LT 都无效的情况下，在灭零输入信号 RBI 端给定一个低电平输入信号，当 $A_3A_2A_1A_0=0000$ 时，七段数码显示器的 0 不显示，7 个发光二极管全部熄灭，这时消隐输入端/灭零输出端 BI/RBO 作为输出端来使用，叫做灭零输出端，处于有效的低电平状态，输出一个 0，RBO＝0 如表 3 - 18 倒数第二行所示。当 $A_3A_2A_1A_0$ 处于除了 0000 以外的其他 15 种值时，七段数码显示器全部正常显示，这时消隐输入端/灭零输出端 BI/RBO 作为输出端来使用，叫做灭零输出端，处于无效的高电平状态，输出一个 1，RBO＝1。例如，有一个 8 位的数码显示电路，整数部分为 5 位，小数部分为 3 位，在显示 13.7 这个数时，将显示 00013.700 字样。如果将前、后多余的零熄灭，则显示的结果将更加醒目。用这些控制端就能实现这种效果。

4. 正常译码显示

当给定灯测试输入端 LT＝1（即处于无效状态）和灭零输入端 RBI＝1（即处于无效状态）时，对输入的十进制数 0～15 的二进制码（$A_3A_2A_1A_0=0000$～1111）进行译码，产生对应的七段显示信号。这时消隐输入端/灭零输出端 BI/RBO 做为输出端，且处于无效状态 BI/RBO＝1。如表 3 - 18 前 16 行所示。

将 BI/RBO 和 RBI 配合使用，可以实现多位数显示时的“无效 0 消隐”功能。在多位十进制数码显示时，整数前和小数后的 0 是无意义的，称为“无效 0”。在图 3 - 46 所示的多位数码显示系统中，就可将无效 0 灭掉。从图中可见，由于整数部分的 74X48 除最高位的 RBI 接 0、最低位的 RBI 接 1 外，其余各位的 RBI 均接高位的 RBO 输出信号。所以整数部分只有在高位是 0，而且被熄灭时，低位才有灭零输入信号。同理，小数部分除最高位的 RBI 接 1、最低位的 RBI 接 0 外，其余各位的 RBI 均接低位的 RBO 输出信号，所以小数部分只有在低位是 0，而且被熄灭时，高位才有灭零输入信号。这就实现了多位十进制数码显示器的“无效 0 消隐”功能。整数部分的个位和小数部分的最高位无论在何种情况下都要显示，比如 0.0 也要显示出来。

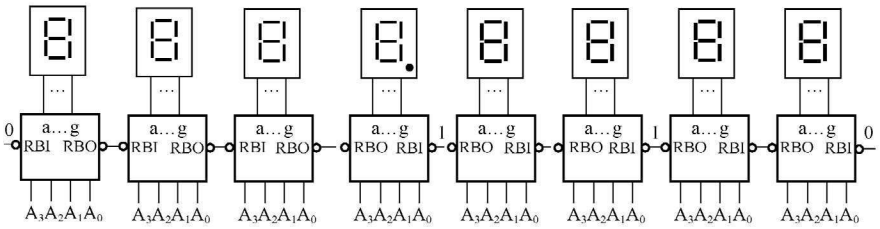


图 3 - 46　有灭零控制的 8 位数码显示

在图 3 - 46 中，8 个 74X48 芯片的灯测试端 LT 都处于无效状态，即全部接入高电平，为了清晰起见，在图中没有表示出来；消隐输入端/灭零输出端 BI/RBO 作为输出来使用。图 3 - 47 是数字 1208.7453 的显示。图 3 - 48 是数字 0000.0000 的显示。

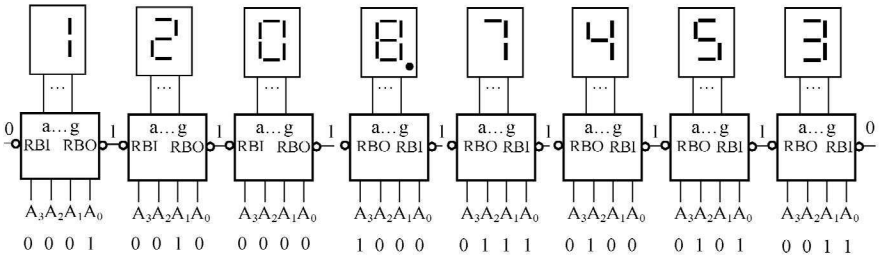


图 3 - 47　数字 1208.7453 的显示

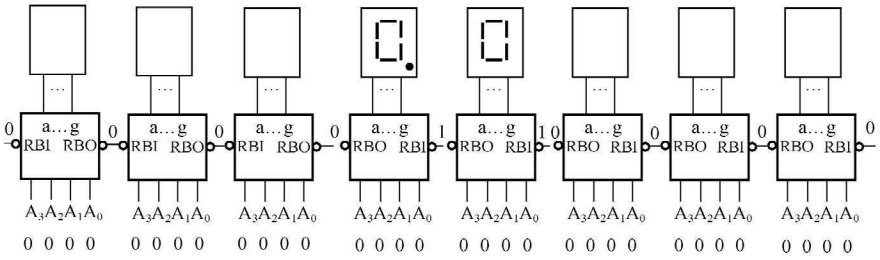


图 3 - 48　数字 0.0 的显示

在图 3 - 45 所示的 74X48 的逻辑符号中，输入信号 A_0、A_1、A_2、A_3、RBI、LT 全部排列在左边，输出信号 a、b、c、d、e、f、g 全部排列在右边。另外，BI/RBO 为双向信号，将 BI/RBO 标示在了左边，BI 消隐输入端是输入信号，所以没有什么特别的标注。RBO 灭零输出端是输出信号，所以在 4 脚处特别标示了一个箭头，箭头的方向向外，标明是输出功能。BIN/7SEG 是总限定符；BIN 是 BINARY 的缩写，表明输入是二进制数码；7SEG 表明是七段输出；SEG 是 SEGMENT 的缩写。OF-8 表明任意瞬间 8 个输出信号中只能有 1 个输出信号处于有效状态。在方框内部靠近输入信号 A_0、A_1、A_2、A_3 处分别标有 1、2、4、8，这是为了便于解释芯片的功能，而人为给输入信号规定的位权。输出端的菱形符号表示是无源上拉输出。“CT＝0”中的 CT 是 CONTENT 的缩写，表示内容的意思。根据它所在的位置可以说明，在 BI 消隐输入端为 0，或者 LT＝1 且 RBI＝0，这两种情况下 CT＝0，即 a、b、c、d、e、f、g 这 7 个输出信号均为 0，不显示字形。

当外部信号 LT=0 时，对应的内部信号产生内部值为 1，这个内部被标上了一个序号 20，用“V20”表示，其中的 V 表示“或”意思。符号中的内嵌框给输入信号的某种组合对应产生了一个内部信号，其标号为 21，表达式为$\overline{LT \cdot \overline{RBI}+\overline{BI}}$，“G21”前面的小圈表示非的意思。即在 LT=1、RBI=0 或者 BI=0，$\overline{LT \cdot \overline{RBI}+\overline{BI}}$为 0，给定了这个内部信号一个标号为 21，“G21”中的 G 表示与的意思。输出信号 13 脚、12 脚、11 脚、10 脚、9 脚、15 脚、14 脚的内部分别标有“a20，21”、“b20，21”、“c20，21”、“d20，21”、“e20，21”、“f20，21”、“g20，21”字样，其中字母 a、b、c、d、e、f、g 分别表示 7 段显示数码管的 7 个段，“20”表示输出信号与内部标有 20 序号的信号为“或”的关系，“21”表示输出信号与内部标有 21 序号的信号为“与”的关系。

第六节 数据选择器

一、数据选择器的基本概念及工作原理

在多路数据传送过程中，能够根据需要将其中任意一路数据挑选出来的电路称为数据选择器（multiplexer），也称为多路数据选择器或多路开关。它的作用与图 3-49 所示的单刀多掷开关相似。

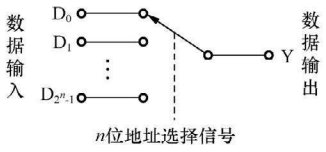


图 3-49 数据选择器功能示意图

常用的数据选择器有 4 选 1、8 选 1、16 选 1 等多种类型。下面以 8 选 1 数据选择器为例介绍数据选择器的工作原理和应用方法。

74X151 是一种典型的集成 8 选 1 数据选择器，其中 X 的含义是 LS、S、ALS、HC、HCT 等表征芯片工艺的标识符。不同工艺的 74X151 芯片的逻辑功能完全相同，只是电路参数不同。图 3-50 是加了引脚名字的 74X151 的逻辑符号，图 3-51 是 74X151 的内部逻辑电路图。

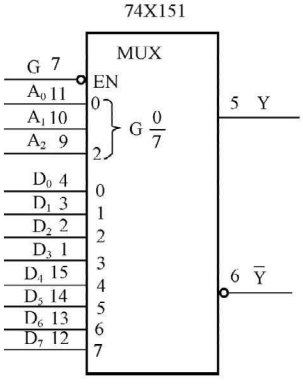


图 3-50 加了引脚名字的 74X151 的逻辑符号

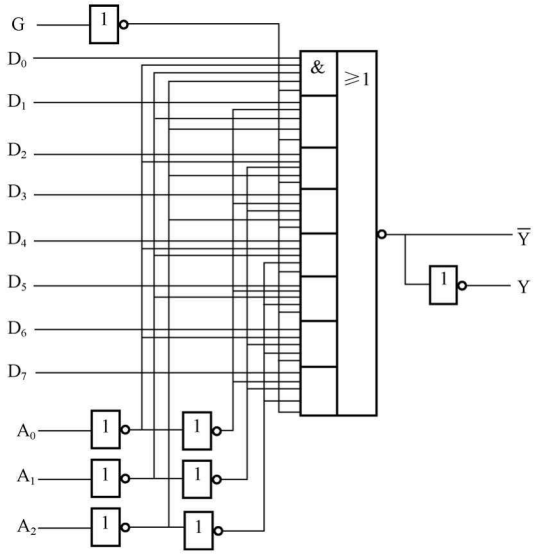


图 3-51 74X151 的内部逻辑电路图

由 74X151 内部逻辑电路图可知，其基本结构为“与—或”形式。它有 8 个数据源 D_0～D_7，3 个地址输入端 A_2、A_1、A_0，2 个互补的输出端 Y 和$\overline{Y}$，1 个使能输入端 G，使能端 G 为低电平有效。74X151 的功能表如表 3-19 所示。

表 3-19　　8 选 1 数据选择器 74X151 功能表

输入				输出		输入				输出	
使能	地址选择			Y	$\overline{Y}$	使能	地址选择			Y	$\overline{Y}$
G	A_2	A_1	A_0			G	A_2	A_1	A_0		
1	×	×	×	0	1	0	1	0	0	D_4	$\overline{D_4}$
0	0	0	0	D_0	$\overline{D_0}$	0	1	0	1	D_5	$\overline{D_5}$
0	0	0	1	D_1	$\overline{D_1}$	0	1	1	0	D_6	$\overline{D_6}$
0	0	1	0	D_2	$\overline{D_2}$	0	1	1	1	D_7	$\overline{D_7}$
0	0	1	1	D_3	$\overline{D_3}$						

在图 3-50 所示的 74X151 的逻辑符号中，总限定符 MUX 表示是一个数据选择器，大括号叫做分组符号（bit-grouping symbol），表明组合在一起的这些二进制位产生一个内部值（internal value），由一组二进制位产生的内部值影响其他的内部值或输出信号，这个内部值是人为规定了位权之后的和，每个位的位权写在相应输入的旁边，在方框的内部。如果所有位的位权都是 2 的幂，则这些位权就可以只用幂来表示，而将底数省略不写。G 表示和内部值的范围在 0～7 内的信号有关联。11 脚 A_0、10 脚 A_1、9 脚 A_2 的位权分别是 2^0、2^1、2^2。当 9 脚 A_2、10 脚 A_1、11 脚 A_0 输入 110 时，内部值就是 6。标有 6 的输入信号 D_6 被传送到输出端，从 5 脚输出。

为了对 8 路数据源进行选择，使用了 3 位地址码 $A_2A_1A_0$，产生 8 个地址信号：000、001、010、011、100、101、110、111，分别控制 8 个与门的开和闭。显然，任意时刻 $A_2A_1A_0$ 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 为低电平有效，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，与门的封锁解除，由地址码决定哪一个与门打开。比如，如果此时地址码为 010，则 D_2 所对应的与门打开，其余与门关闭，输出 Y 为 D_2。

根据图 3-51 所示的内部逻辑电路图可以写出输出 Y 的逻辑表达式为

$$\begin{aligned}Y=&\overline{G}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}D_0+\overline{G}\,\overline{A_2}\,\overline{A_1}A_0D_1+\overline{G}\,\overline{A_2}A_1\,\overline{A_0}D_2+\overline{G}\,\overline{A_2}A_1A_0D_3\\&+\overline{G}A_2\,\overline{A_1}\,\overline{A_0}D_4+\overline{G}A_2\,\overline{A_1}A_0D_5+\overline{G}A_2A_1\,\overline{A_0}D_6+\overline{G}A_2A_1A_0D_7\end{aligned}\tag{3-13}$$

当 G=0 时，输出 Y 的逻辑表达式为

$$\begin{aligned}Y(A_2,A_1,A_0)=&m_0D_0+m_1D_1+m_2D_2+m_3D_3\\&+m_4D_4+m_5D_5+m_6D_6+m_7D_7\end{aligned}\tag{3-14}$$

从式（3-14）可知，当以 A_2（9 脚）为高位，按照 A_2（9 脚）、A_1（10 脚）、A_0（11 脚）位号依次降低的顺序，把变量取值组合看成二进制数的方式对最小项编号时，m 的下标与 D 的下标正好一致，这样便于记忆输出的逻辑表达式。如果不是以这样的方式对最小项编号，并不是说不可以，但是 m 的下标与 D 的下标不一致［见式（3-15）］，没有什么规律，不便于记忆输出的逻辑表达式，这样的最小项编号的方式没有什么意义。

以 A_0 为高位，按照 A_0、A_1、A_2 位号依次降低的顺序，对最小项编号。当 G＝0 时，输出 Y 的逻辑表达式为

$$Y(A_0,A_1,A_2)=m_0D_0+m_4D_1+m_2D_2+m_6D_3+m_1D_4+m_5D_5+m_3D_6+m_7D_7 \quad (3-15)$$

常用的集成电路数据选择器有许多种类，并且有 CMOS 和 TTL 产品。除了上面介绍的 8 选 1 数据选择器 74X151 外，还有四 2 选 1 数据选择器 74X157（一片中有 4 个完全相同的 2 选 1 数据选择器）、双 4 选 1 数据选择器 74X153 等。还有一些数据选择器具有三态输出功能，例如与上述产品相对应具有三态输出功能的有 74X251、74X257、74X253。输出除了有正常的高电平、低电平外，当使能端无效时，输出为高阻状态。利用这一特点，可以将多个芯片的输出端连接在一起，共用一根数据传输线。

二、数据选择器的应用

（一）数据选择器的扩展

作为一种集成器件，常见的数据选择器类型是 8 选 1。选择数据的位数是 1，而数据源通道为 8。如果需要更大规模的数据选择器，可进行扩展。数据选择器的扩展分为两个方面的扩展：选择数据的位数扩展和数据源通道的扩展。

1. 选择数据位数的扩展

前面所讨论的是 1 位的数据选择器，如果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接在一起，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 3 - 52 所示。

在图 3 - 52 中，74X151（1）和 74X151（2）的 7 脚连接在一起，作为使能端，低电平有效。74X151（1）和 74X151（2）的 9 脚、10 脚、11 脚分别连接在一起，作为地址选择端 $A_2A_1A_0$。当使能端 G 为低电平，$A_2A_1A_0$＝000 时，数据 $D_{10}D_{00}$ 被选择，送到输出端 Y_1Y_0。$A_2A_1A_0$＝001 时，数据 $D_{11}D_{01}$ 被选择，送到输出端 Y_1Y_0。依次类推，$A_2A_1A_0$＝111 时，数据 $D_{17}D_{07}$ 被选择，送到输出端 Y_1Y_0。当需要进一步扩充数据选择器位数时，仅仅需要相应地增加器件的数目。

2. 数据通道源的扩展

当需要对数据通道源数目进行扩展时，可以把数据选择器的使能端作为地址选择输入。比如用两片 74X151 和 3 个门电路组成的 16 选 1 的数据选择器电路，如图 3 - 53 所示。

16 选 1 的数据选择器的地址选择输入应该有 4 位，用 $A_3A_2A_1A_0$ 表示。在图 3 - 54 中，最高位 A_3 与 74X151（1）的使能端 7 脚连接在一起；另外，经过一个反相器后与 74X151（2）的使能端连接。74X151（1）和 74X151（2）的 9 脚、10 脚、11 脚分别连接在一起作为 $A_2A_1A_0$。

当 $A_3A_2A_1A_0$＝0000 时，因为 A_3＝0，所以 74X151（1）被使能，而 74X151（2）被禁止，74X151（2）的 5 脚输出为 0，这时 D_0 数据通过 74X151（1）送到 5 脚输出，然后通过或门 G1 送到输出端 L。当 $A_3A_2A_1A_0$＝1000 时，因为 A_3＝1，所以 74X151（2）被使能，而 74X151（1）被禁止，74X151（1）的 5 脚输出为 0，这时 D_8 数据通过 74X151（2）送到 5 脚输出，然后通过或门 G1 送到输出端 L。

若要实现更多的数据源通道扩展，则不可能只借助于附加简单的门电路了，而是需要借助于译码器或者附加的数据选择器来实现。

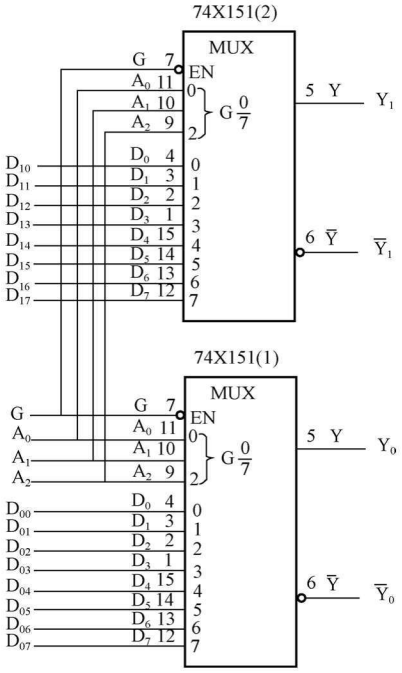


图 3 - 52　2 位的 8 选 1 数据选择器的逻辑电路图

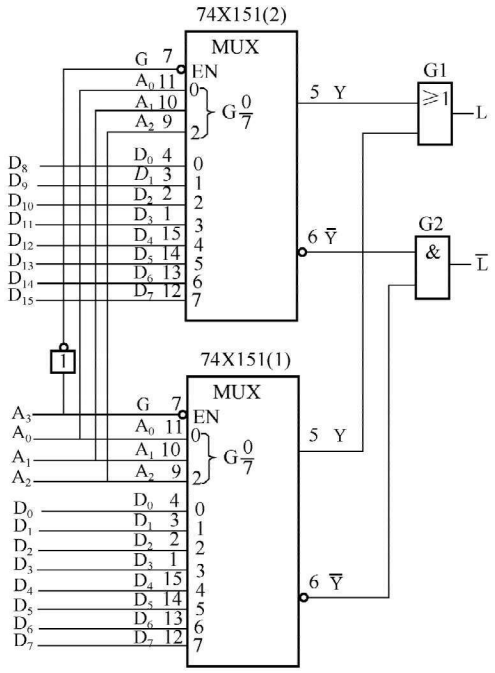


图 3 - 53　用两片 74X151 组成的 16 选 1 数据选择器的逻辑电路图

（二）实现单输出组合逻辑函数

从前面已经知道，74X151 当使能端 G＝0 时，数据选择器的输出 Y 是 $A_2A_1A_0$ 和输入数据 $D_7D_6D_5D_4D_3D_2D_1D_0$ 的与或函数，其表达式为

$$Y(A_2,A_1,A_0)=m_0D_0+m_1D_1+m_2D_2+m_3D_3+m_4D_4+m_5D_5+m_6D_6+m_7D_7$$
$$=\sum_{i=0}^{7}(m_iD_i)$$

【例 3 - 9】　试用 8 选 1 数据选择器 74X151 实现单输出组合逻辑函数

$$L=AB+BC+AC$$

解　（1）将要实现的逻辑函数转换成最小项表达式，即

$$L(A,B,C)=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC=m_3+m_5+m_6+m_7$$

此处要特别注意将逻辑函数写成最小项编号的形式时，要特别指明对最小项编号的方法，这部分的内容在讲解最小项的概念时已经强调过。

（2）当 G＝0 时，74X151 的输出函数表达式为

$$Y(A_2,A_1,A_0)=m_0D_0+m_1D_1+m_2D_2+m_3D_3+m_4D_4+m_5D_5+m_6D_6+m_7D_7$$
$$=\sum_{i=0}^{7}(m_iD_i)$$

（3）将本题目要实现的组合逻辑函数与 74X151 的输出表达式做对应。输入变量 A、B、C 接至数据选择器的地址输入端 A_2、A_1、A_0，即 $A=A_2$，$B=A_1$，$C=A_0$。输出变量接至数据选择器的输出端，即 L＝Y。将逻辑函数 L 的最小项表达式与 74X151 的输出表达式相

比较，L 式中出现的最小项，对应的数据输入端应接 1，L 式中没出现的最小项，对应的数据输入端应接 0，即 $D_3=D_5=D_6=D_7=1$，$D_0=D_1=D_2=D_4=0$。

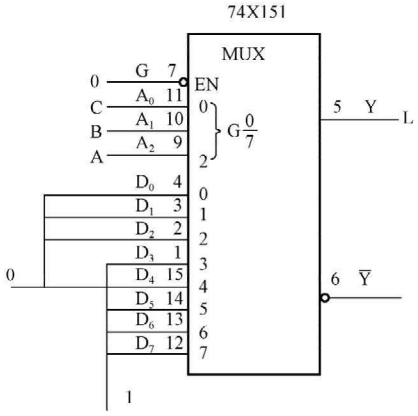


图 3 - 54 ［例 3 - 9］的逻辑电路图

（4）画出该逻辑函数对应的逻辑电路图如图 3 - 54所示。

在上面的这个例题中，要实现的逻辑函数中的变量个数与数据选择器的地址输入端的个数相同，将变量与数据选择器的地址输入端一一对应即可。如果要实现的逻辑函数中的变量个数与数据选择器的地址输入端的个数不同，不能用前述的简单办法，而应分离出多余的变量，把它们加到适当的数据输入端。下面举例说明这种方法。

【例 3 - 10】 试用 8 选 1 数据选择器 74X151 实现单输出 4 变量组合逻辑函数

$$L = ABCD + BC\overline{D} + AC$$

解 解法一：

将要实现的函数转换成最小项表达式，即

$$\begin{aligned} L &= ABCD + BC\overline{D} + AC \\ &= ABCD + ABC\overline{D} + \overline{A}BC\overline{D} + ABCD + A\overline{B}CD + ABC\overline{D} + A\overline{B}C\overline{D} \\ &= ABCD + ABC\overline{D} + \overline{A}BC\overline{D} + A\overline{B}CD + A\overline{B}C\overline{D} \end{aligned}$$

任意取 3 个变量，这三个变量要与 74X151 的 3 个地址选择端对应，并且将函数写成这 3 个变量的最小项表达式形式，不妨取 A、B、C 这 3 个变量，然后按这 8 个最小项对表达式进行整理。D 这个变量要从 8 个数据输入端进行输入，必要时还要使用门电路，则

$$\begin{aligned} L(A,B,C) &= m_7D + m_7\overline{D} + m_3\overline{D} + m_5D + m_5\overline{D} \\ &= m_7(D+\overline{D}) + m_3\overline{D} + m_5(D+\overline{D}) \\ &= m_7 1 + m_3\overline{D} + m_5 1 \end{aligned}$$

当 G＝0 时，74X151 的输出函数表达式为

$$\begin{aligned} Y(A_2,A_1,A_0) &= m_0D_0 + m_1D_1 + m_2D_2 + m_3D_3 + m_4D_4 + m_5D_5 + m_6D_6 + m_7D_7 \\ &= \sum_{i=0}^{7}(m_iD_i) \end{aligned}$$

将本题目要实现的组合逻辑函数与 74X151 的输出表达式做对应。输入变量 A、B、C 接至数据选择器的地址输入端 A_2、A_1、A_0，即 $A=A_2$，$B=A_1$，$C=A_0$。输出变量接至数据选择器的输出端，即 L＝Y。将逻辑函数 L 的最小项表达式与 74X151 的输出表达式相比较，L 式中没出现的最小项，对应的数据输入端应接 0，即 $D_0=D_1=D_2=D_4=D_6=0$，$D_3=\overline{D}$，$D_5=D_7=1$。画出连线图如图 3 - 55 所示。

解法二：

写出要实现函数的最小表达式，即

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

任意取 3 个变量，这 3 个变量要与 74X151 的 3 个地址选择端对应，并且将函数写成这 3 个变量的最小项表达式形式。本解法中取 B、C、D 这 3 个变量，然后按这 8 个最小项对表达式进行整理，得

$$L(B,C,D)=m_7A+m_6A+m_6\overline{A}+m_3A+m_2A$$
$$=m_7A+m_61+m_3A+m_2A$$

当 G＝0 时，74X151 的输出函数表达式为

$$Y(A_2,A_1,A_0)=m_0D_0+m_1D_1+m_2D_2+m_3D_3+m_4D_4+m_5D_5+m_6D_6+m_7D_7$$
$$=\sum_{i=0}^{7}(m_iD_i)$$

输入变量 B、C、D 接至数据选择器的地址输入端 A_2、A_1、A_0，即 $B=A_2$，$C=A_1$，$D=A_0$。输出变量接至数据选择器的输出端，即 L＝Y。将逻辑函数 L 的最小项表达式与 74X151 的输出表达式相比较，L 式中没出现的最小项，对应的数据输入端应接 0，即 $D_0=D_1=D_4=D_5=0$，$D_2=D_3=D_7=A$，$D_6=1$。画出连线图如图 3－56 所示。

从 A、B、C、D 中任意取 3 个变量，并任意规定位权对最小项编号，将有很多种求解方法。本处不再一一讲解。

通过上面的例子可以看出，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进行化简。

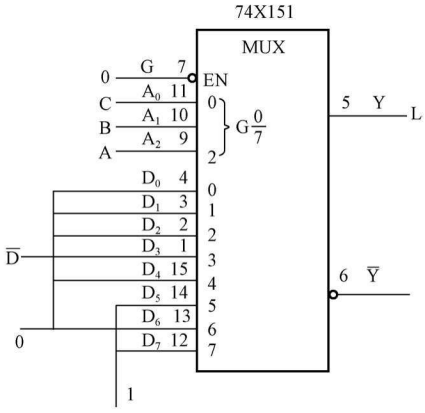


图 3－55　［例 3－10］解法一的逻辑电路图

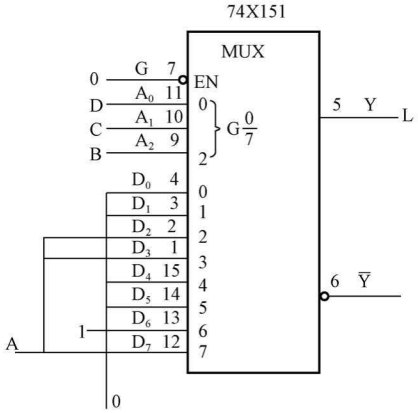


图 3－56　［例 3－10］解法二的逻辑电路图

（三）实现并行数据到串行数据的转换

图 3－57 是实现并/串行转换的逻辑电路图，图 3－58 是输出信号波形图。电路由 8 选 1 数据选择器和 1 个 3 位二进制计数器构成，计数器的概念将在第五章讲解。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 $Q_2Q_1Q_0$ 从 000→001→…→111 依次变换，循环往复。由于 Q_2、Q_1、Q_0 分别与选择器的地址输入端 A_2、A_1、A_0 相连，因此，$A_2A_1A_0$ 就随时钟脉冲的逐个输入从 000 向 111 变化，选择器的输出 *L* 随之接通 D_0，D_1，D_2，…，D_7。当选择器的数据输入端 D_0，D_1，D_2，…，D_7 与一个并行 8 位数据 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。

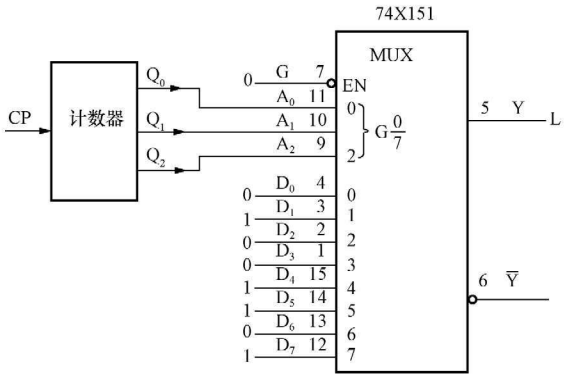


图 3 - 57 实现并/串行转换的逻辑电路图

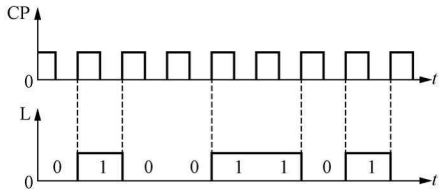


图 3 - 58 输出信号波形图

第七节 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。下面介绍加法运算和减法运算的逻辑电路。

一、加法器

加法器是实现两个二进制数的加法运算的电路。算术运算中的四则运算在电路中往往分解转化为加法进行，因此加法器是运算电路的核心部件。多位加法器的基础是 1 位加法器，而 1 位加法器又分为 1 位半加器和 1 位全加器两种电路。

（一）1 位加法器

1. 1 位半加器

如果不考虑来自低位的进位，只将两个 1 位二进制数相加，称为半加运算。实现半加运算的电路叫做 1 位半加器。即 1 位半加器就是只能进行本位加数、被加数的加法运算而不考虑低位进位的运算逻辑电路。列出 1 位半加器的真值表，如表 3 - 20 所示。

表 3 - 20　　1 位半加器真值表

输入	被加数 A	0	0	1	1
	加数 B	0	1	0	1
输出	和数 S	0	1	1	0
	进位数 C	0	0	0	1

由真值表直接写出逻辑表达式为

$$S = \overline{A}B + A\overline{B} = A \oplus B \tag{3-16}$$

$$C = AB \tag{3-17}$$

由上面的函数表达式知道，可以用一个异或门和一个与门实现 1 位半加器的逻辑电路，如图 3 - 59 所示。

如果想用与非门组成 1 位半加器，则将上式用代数法变换成与非形式，即

$$\begin{aligned} S &= \overline{A}B + A\overline{B} = \overline{A}B + A\overline{B} + A\overline{A} + B\overline{B} \\ &= A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) = A\overline{AB} + B\overline{AB} = \overline{\overline{A\overline{AB}}\,\overline{B\overline{AB}}} \end{aligned} \tag{3-18}$$

$$C = AB = \overline{\overline{AB}} \tag{3-19}$$

由此画出用与非门组成的 1 位半加器如图 3 - 60 所示，图 3 - 61 为 1 位半加器的逻辑符号。

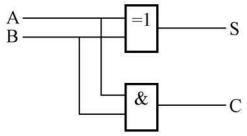


图 3 - 59　1 位半加器的逻辑电路图

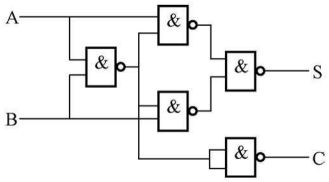


图 3 - 60　用与非门组成的 1 位半加器逻辑电路图

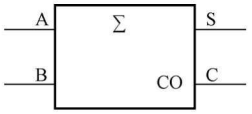


图 3 - 61　1 位半加器的逻辑符号

2. 1 位全加器

将两个多位二进制数相加时，除了最低位以外，每一位都应该考虑来自低位的进位，即将两个对应位的加数和来自低位的进位共 3 个数相加，这种运算称为全加。实现全加运算的电路叫做全加器，即全加器就是能同时进行本位数和相邻低位的进位信号相加的运算电路。1 位全加器真值表如表 3 - 21 所示。

表 3 - 21　　1 位 全 加 器 真 值 表

A_i	B_i	C_{i-1}	S_i	C_i	A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	1	1	1

由真值表直接写出逻辑表达式，再经代数法化简和转换得到

$$S_i = \overline{A_i}\,\overline{B_i}C_{i-1} + \overline{A_i}B_i\,\overline{C_{i-1}} + A_i\,\overline{B_i}\,\overline{C_{i-1}} + A_iB_iC_{i-1}$$

$$= \overline{(A_i \oplus B_i)}C_{i-1} + (A_i \oplus B_i)\,\overline{C_{i-1}} = A_i \oplus B_i \oplus C_{i-1} \tag{3-20}$$

$$C_i = \overline{A_i}B_iC_{i-1} + A_i\,\overline{B_i}C_{i-1} + A_iB_i\,\overline{C_{i-1}} + A_iB_iC_{i-1}$$

$$= A_iB_i + (A_i \oplus B_i)C_{i-1} \tag{3-21}$$

整理式（3 - 20）、式（3 - 21），用异或门实现，得到 1 位全加器的逻辑表达式为

$$S_i = A_i \oplus B_i \oplus C_{i-1} \tag{3-22}$$

$$C_i = A_iB_i + (A_i \oplus B_i)C_{i-1} \tag{3-23}$$

根据式（3 - 22）、式（3 - 23）画出 1 位全加器的逻辑电路图，如图 3 - 62 所示。1 位全加器的逻辑符号如图 3 - 63 所示。

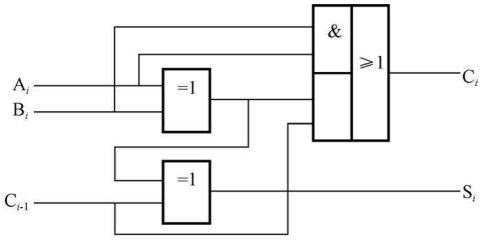


图 3 - 62　1 位全加器的逻辑电路图

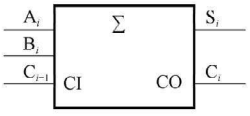


图 3 - 63　1 位全加器的逻辑符号

（二）多位加法器

1. 串行进位加法器

两个多位二进制数相加时每一位都是带进位相加的，因而必须使用 1 位全加器来实现此功能。只要依次将低位 1 位全加器的进位输出端接到高位 1 位全加器的进位输入端，就可以构成多位加法器。4 位串行进位二进制加法器如图 3 - 64 所示。

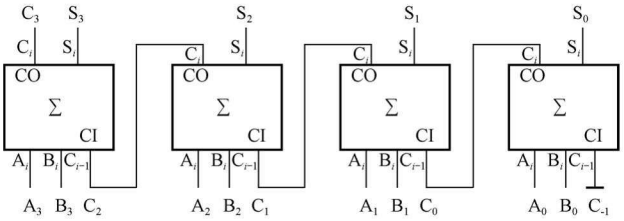


图 3 - 64　4 位串行进位加法器

这种串行进位的多位加法器的结构简单、应用面较广，但是其每一位的加法运算只能等低位进位信号产生之后才能进行，时间较长，运算速度较慢。图 3 - 65 中最后一位的进位输出 C_3 要经过四位全加器传递之后才能形成。如果位数增加，传输延迟时间将更长，工作速度更慢。串行进位加法器在运算速度要求不高的场合仍得到应用。

2. 超前进位加法器

为了提高运算速度，必须减少进位信号逐级传递所消耗的时间。因此，人们又设计了一种多位数超前进位（又称快速进位）的加法器。所谓超前进位，是指加法运算过程中，各级进位信号同时送到各位 1 位全加器的进位输入端。现在的集成加法器，大多采用这种方法。74X283 是一种典型的超前进位的集成加法器。

（1）超前进位的概念。1 位全加器 S_i 和 C_i 的输出逻辑表达式为

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_iB_i + (A_i \oplus B_i)C_{i-1}$$

考察进位信号 C_i 的表达式，可见：当 $A_i = B_i = 1$ 时，$A_iB_i = 1$，得 $C_i = 1$，即产生进位。所以，定义 $G_i = A_iB_i$，G_i 称为产生变量。当 $A_i \oplus B_i = 1$，则 $A_iB_i = 0$，得 $C_i = C_{i-1}$，即低位的进位信号能传送到高位的进位输出端。

定义 $P_i = A_i \oplus B_i$，称为传输变量。将 G_i 和 P_i 代入各进位信号的逻辑表达式为

$$C_0 = G_0 + P_0C_{-1} \tag{3-24}$$

$$C_1 = G_1 + P_1C_0 = G_1 + P_1G_0 + P_1P_0C_{-1} \tag{3-25}$$

$$C_2 = G_2 + P_2C_1 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_{-1} \tag{3-26}$$

$$C_3 = G_3 + P_3C_2 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{-1} \tag{3-27}$$

由式（3 - 25）、式（3 - 26）、式（3 - 27）可以看出：各位的进位信号都只与 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被加数 A_i 和加数 B_i 有关，它们是可以并行产生的，从而可实现超前进位。

（2）超前进位集成 4 位加法器 74X283。图 3 - 65 是加了引脚名字的 74X283 的逻辑符号，其中输入信号列在左边，输出信号列在右边；总定性符号 Σ 表示求和运算；被加数、加

数、进位输入等输入信号列在左边，和、进位输出列在右边；0～3规定了各位的位权，CI是进位输入端，CO是进位输出端。

（三）集成加法器的应用

1. 加法器级联实现多位二进制数加法运算

一片74X283只能进行4位二进制数的加法运算，将多片74X283进行级联，就可扩展参与加法运算的二进制数的位数。用2片74X283组成的8位二进制数加法电路如图3－66所示。$A_7A_6A_5A_4A_3A_2A_1A_0$是被加数，$B_7B_6B_5B_4B_3B_2B_1B_0$是加数，$S_7S_6S_5S_4S_3S_2S_1S_0$是和，C_7是最高位产生的进位，片（1）的C_{-1}接0，片（1）的9脚C_3是低四位加法运算的进位，送给片（2）的7脚。

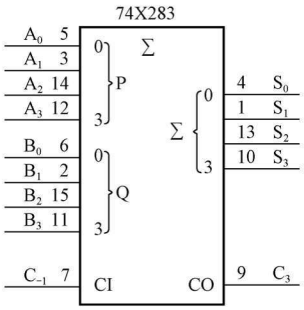


图3－65　加了引脚名字的74X283的逻辑符号

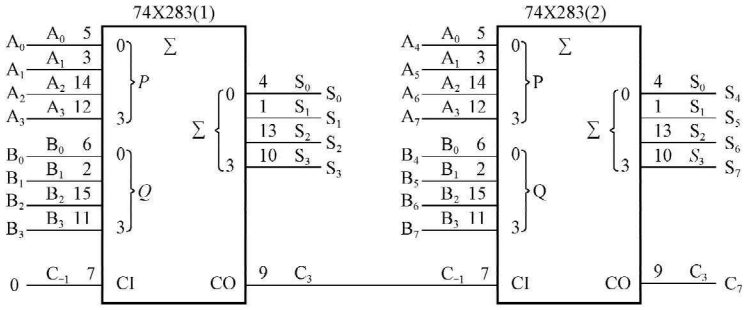


图3－66　2片74X283组成的8位二进制数加法电路图

2. 用74X283构成一位8421BCD码的加法器

8421BCD码的加法运算为十进制运算，逢十进一，而74X283是四位二进制数的加法运算，二者进位关系不同，若要用74X283构成一位8421BCD码的加法器，则必须对结果进行修正。当8421BCD码加法运算的和数大于9时，8421BCD码加法运算电路应该产生向高位的进位。当两个用8421BCD码表示的一位十进制数相加时，每个数都不会大于9（1001），再考虑到低位来的进位，最大的和为9＋9＋1＝19。

当用4位二进制数加法器74X283完成这个加法运算时，要用两片74X283。片（1）完成加法运算，片（2）完成修正运算。加法器输出的和$S_3S_2S_1S_0$是4位二进制数表示的和，而不是8421BCD码，因此，必须想办法将4位二进制数表示的和转换成8421BCD码。将0～19对应的二进制数与用8421BCD码表示的数进行比较发现，当数不大于1001（9）时，二进制码与8421BCD码相同；当数大于1001时，只要在二进制码上加0110（6）就可以把二进制码转换为8421BCD码，同时产生进位输出。这一转换可以由修正电路来完成。下面介绍这个修正电路的设计。

当两个一位8421BCD码相加时，不外乎这两种情况，或者是和数超过9但没有进位（即和数为10、11、12、13、14、15），或者是有进位（此时和数一定超过9，即和数为16、17、18、19），这两种情况都应该进行和数加6（即0110）修正。

设M为修正信号，N为片（1）最高位的进位输出信号，$M=N+C_{S>9}$。其中，$C_{S>9}$表示和数大于9但没有进位的情况。

列出求 $C_{S>9}$ 信号的真值表如表 3 - 22 所示。求 $C_{S>9}$ 信号的卡诺图如图 3 - 67 所示。化简得 $C_{S>9}=S_3S_2+S_3S_1$，则修正信号

$$M = N + C_{S>9} = N + S_3S_2 + S_3S_1 \tag{3-28}$$

表 3 - 22　　求 $C_{S>9}$ 信号的真值表

S_3	S_2	S_1	S_0	$C_{S>9}$	S_3	S_2	S_1	S_0	$C_{S>9}$	S_3	S_2	S_1	S_0	$C_{S>9}$
0	0	0	0	0	0	1	1	0	0	1	1	0	0	1
0	0	0	1	0	0	1	1	1	0	1	1	0	1	1
0	0	1	0	0	1	0	0	0	0	1	1	1	0	1
0	0	1	1	0	1	0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	0	1	0	1					
0	1	0	1	0	1	0	1	1	1					

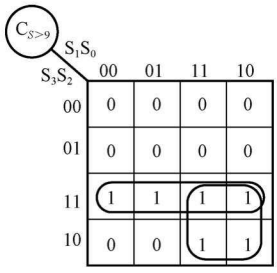


图 3 - 67　求 $C_{S>9}$ 信号的卡诺图

根据式（3 - 28）得到逻辑电路图如图 3 - 68 所示。当 M=1 时，把 0110 加到二进制加法器输入端即可。当 M=0 时，把 0000 加到二进制加法器 74X283（2）输入端，即不做修正。

【例 3 - 11】　试采用 74X283 完成 8421BCD 码到余 3 码的转换。

解　由于 8421BCD 码加 0011 即为余 3 码，所以其转换电路就是一个加法电路，如图 3 - 69 所示。

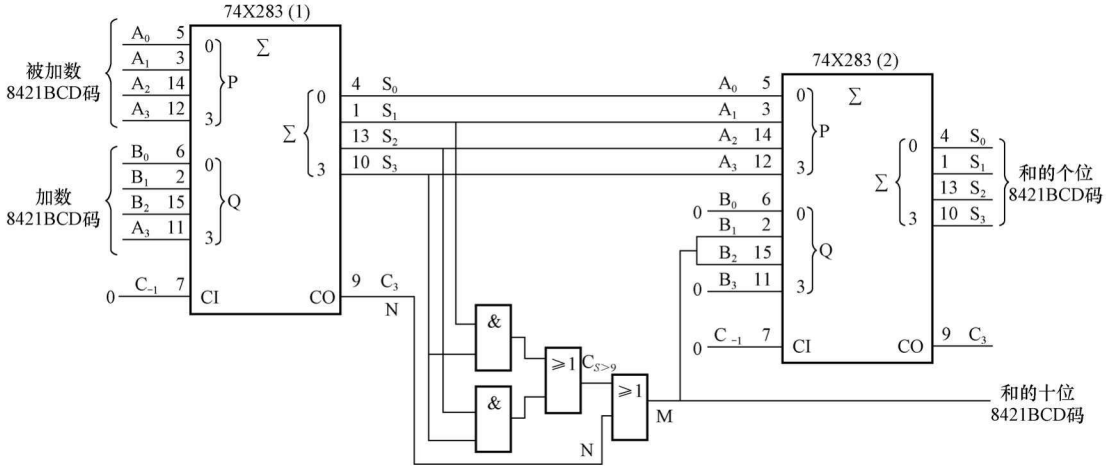


图 3 - 68　用 74X283 构成一位 8421BCD 码加法器逻辑电路图

二、用加法器实现减法和乘法运算

同加法一样，减法运算可以采用减法器来实现。但为了简化系统结构，通常不另行设计减法器，而是将减法运算变为加法运算来处理，使运算器即能实现加法运算，又可以实现减法运算。一般采用加补码的方法代替减法运算，下面介绍这种方法的原理。

（一）补码的定义

$(N)_2$ 的补码的定义为

$$[N]_2 = 2^n - (N)_2 \qquad (3-29)$$

式中：$(N)_2$ 表示 N 的二进制数，本身不包括符号位，只是数值本身；$[N]_2$ 表示 N 的补码，二进制数形式，本身不包括符号位，只是数值本身；n 是二进制数的位数。

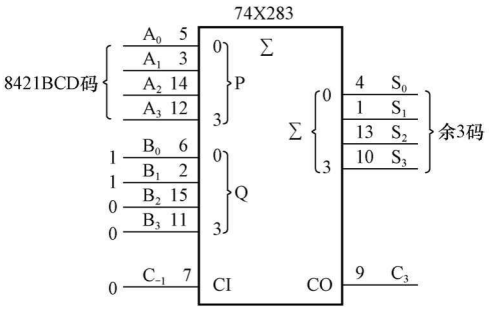


图 3-69　[例 3-11] 的电路图

【例 3-12】　求 $(N)_2=(01100101)_2$ 的补码。

解　从 $(01100101)_2$ 中可以观察出 $n=8$，所以有

$$[N]_2 = [01100101]_2 = 2^8 - (01100101)_2$$
$$= (100000000)_2 - (01100101)_2 = (10011011)_2$$

【例 3-13】　求 $(N)_2=(11010010)_2$ 的补码，并验证 $(N)_2+[N]_2=0$。

解　从 $(11010010)_2$ 中可以观察出 $n=8$，所以有

$$[N]_2 = [11010010]_2 = 2^8 - (11010010)_2$$
$$= (100000000)_2 - (11010010)_2 = (00101110)_2$$

对 $(N)_2$ 和 $[N]_2$ 做加运算，即

$$\begin{array}{r} 1\,1\,0\,1\,0\,0\,1\,0 \\ +\,0\,0\,1\,0\,1\,1\,1\,0 \\ \hline 1\,0\,0\,0\,0\,0\,0\,0\,0 \end{array}$$

其中最左边的 1 为进位，如果丢掉进位，可知 $(N)_2+[N]_2=0$，也就是说，可以用 $[N]_2$ 来表示 $-(N)_2$。

【例 3-14】　求 [例 3-13] 中计算得到的补码 $[N]_2=(00101110)_2$ 的补码。

解　从 $(00101110)_2$ 中可以观察出 $n=8$，所以有

$$[[N]_2]_2 = [00101110]_2 = 2^8 - (00101110)_2$$
$$= (100000000)_2 - (00101110)_2 = (11010010)_2$$

此外，$[[N]_2]_2$ 与 [例 3-13] 中的 $(N)_2$ 相等。

从 [例 3-14] 得知，对一个数两次求补码的结果还是这个数本身，即

$$[[N]_2]_2 = [-(N)_2]_2 = -(-(N)_2)_2 = (N)_2 \qquad (3-30)$$

【例 3-15】　求 $(N)_2=(10110)_2$ 的补码，二进制数的位数 $n=8$。

解　从 $(10110)_2$ 观察 $n=5$，但是题目已经明确要求二进制数的位数 $n=8$，所以有

$$[N]_2 = [10110]_2 = 2^8 - (10110)_2 = (100000000)_2 - (10110)_2 = (11101010)_2$$

（二）求补码的算法

从上面补码的定义来求补码比较麻烦，下面给出两个比较简便的算法。

(1) 算法一：从最低位开始向最高位依次观察，碰到的第一个 1 保留，从这一位往左，将所有位求反，就得到这个数的补码。

【例 3-16】　求 $(N)_2=(01100100)_2$ 的补码。

解　$N=0\,1\,1\,0\,0\,1\,0\,0$

$[N]_2=10011100$

【例 3-17】 求 $(N)_2=(01100101)_2$ 的补码。

解 $N=01100101$

$[N]_2=10011011$

(2) 算法二：将所有位求反后再加 1 即可得到补码。

(三) 数的真值

因为在一般情况下，数总是有正数和负数之分，所以这里讨论有符号数。对于有符号数，数的真值表示方法是，数值的前面加上相应的正号“+”或负号“−”。例如，$+(1101.001)_2$，$-(1101.001)_2$。

在数字系统中，有符号数表示为机器数（二进制数），正数的符号和负数的符号也用数字表示，正数的符号用 0 表示，负数的符号用 1 表示，即符号与数值部分统一用二进制代码表示。在机器数中根据数值部分形成的规则不同又分为原码、反码和补码。对于正数来说，原码、反码和补码均相同。下面重点讨论负数的原码、反码和补码。

(四) 原码系统

原码也叫符号—数值码（sign magnitude）。例如 $(-13)_{10}=-(1101)_2=(11101)_{2sm}$，其中，第一个数 $(-13)_{10}$ 是十进制数；第二个数 $-(1101)_2$ 是对应的二进制数的真值，第三个数 $(11101)_{2sm}$ 是对应的原码（符号−数值码），下标用 2sm 表示，$(11101)_{2sm}$ 中最左边的 1 表示符号，是负数。

数的原码表示方法简单直观，与真值之间转换容易。但是数的原码表示方法，当两个数进行加、减运算时较复杂。例如，用原码表示的两个数在做加法运算时，要判断两个数是同号还是异号。若同号，则做加法运算；若异号，则实际是做减法运算。当实际是做减法运算时，还要判断两个数的绝对值的大小，要用绝对值大的数减去绝对值小的数，差的符号与绝对值大的相同。这样在计算机中不仅要增加相应的判断电路，而且影响运算速度。数的补码表示方法可以使减法运算全部转换成加法运算。

【例 3-18】 给定 $(N)_2=(1100101)_2$，求 $\pm(N)_2$ 的原码，且 $n=8$。

解 $+(N)_2=(1100101)_2=(0,1100101)_{2sm}$

$-(N)_2=(1,1100101)_{2sm}$

注意，在这里“$-(N)_2$”中包括符号位在内一共为 8 位。

(五) 补码系统

在补码系统中，正数的表示方法与原码完全相同，而负数则用对应正数的补码表示。用 $(N)_{2cns}$ 表示在补码系统中的一个数。因此 $N=+(a_{n-2},\cdots,a_0)_2=(0,a_{n-2},\cdots,a_0)_{2cns}$，其中 $0\leqslant N\leqslant(2^{n-1}-1)$。如果用 $N=(a_{n-1},a_{n-2},\cdots,a_0)_2$ 表示，那么 $-N$ 就用 $[a_{n-1},a_{n-2},\cdots,a_0]_2$ 来表示，这里 $-1\geqslant -N\geqslant -2^{n-1}$。

【例 3-19】 给定 $(N)_2=(1100101)_2$，求 $\pm(N)_2$ 的补码，且 $n=8$。

解 $+(N)_2=(1100101)_2=(0,1100101)_{2cns}$

$-(N)_2=[+(N)_2]_2=2^8-(01100101)_2=(100000000)_2-(01100101)_2=(1,0011011)_{2cns}$

注意，在这里，包括符号位在内一共为 8 位。

【例 3-20】 给定 $(N)_2=(11001)_2$，求 $\pm(N)_2$ 的补码，且 $n=8$。

解 $+(N)_2=(11001)_2=(0,0011001)_{2cns}$

$-(N)_2=[+(N)_2]_2=2^8-(11001)_2=(100000000)_2-(11001)_2=(1,1100111)_{2cns}$

注意，在这里包括符号位在内一共为 8 位。

【例 3-21】 求 $(-13)_{10}$ 在补码系统中的表示方法，且 $n=8$。

解 $+(13)_{10}=+(1101)_2=(0,0001101)_{2cns}$

因此，$-(13)_{10}$ 在补码系统中的表示方法为 $(1,1110011)_{2cns}$，即对应正数的补码。

（六）补码系统中的加法和减法运算

【例 3-22】 使用 $n=5$ 的补码系统计算 $(9)_{10}+(5)_{10}$。

解 $+(9)_{10}=+(1001)_2=(0,1001)_{2cns}$

$+(5)_{10}=+(0101)_2=(0,0101)_{2cns}$

$$\begin{array}{r} 0\,1\,0\,0\,1 \\ +0\,0\,1\,0\,1 \\ \hline 0\,1\,1\,1\,0 \end{array}$$

$(0,1110)_{2cns}=+(1110)_2=+(14)_{10}$

【例 3-23】 使用 $n=5$ 的补码系统计算 $(12)_{10}+(7)_{10}$，并说明计算结果是否正确。

解 $+(12)_{10}=+(1100)_2=(0,1100)_{2cns}$

$+(7)_{10}=+(0111)_2=(0,0111)_{2cns}$

$$\begin{array}{r} 0\,1\,1\,0\,0 \\ +0\,0\,1\,1\,1 \\ \hline 1\,0\,0\,1\,1 \end{array}$$

$(1,0011)_{2cns}=-(1101)_2=-(13)_{10}$

[例 3-23] 结果表明两个正数相加得到了一个负数。这个结果显然不正确，$(12)_{10}+(7)_{10}$ 的结果应为 $+(19)_{10}$，而 $+(19)_{10}$ 已经超过了 $n=5$ 的补码系统所能表示的最大正数。$n=5$的补码系统所能表示的最大正数为 $(15)_{10}$，有溢出现象发生。不正确的符号位表明有溢出现象。

【例 3-24】 使用 $n=5$ 的补码系统计算 $(12)_{10}-(5)_{10}$。

解 $+(12)_{10}=+(1100)_2=(0,1100)_{2cns}$

$-(5)_{10}=-(0101)_2=(1,1011)_{2cns}$

$$\begin{array}{r} 0\,1\,1\,0\,0 \\ +\quad 1\,1\,0\,1\,1 \\ \hline 1\,0\,0\,1\,1\,1 \end{array}$$

丢掉进位，$(0,0111)_{2cns}=+(0111)_2=+(7)_{10}$

【例 3-25】 使用 $n=5$ 的补码系统计算 $(5)_{10}-(12)_{10}$。

解 $+(5)_{10}=+(0101)_2=(0,0101)_{2cns}$

$-(12)_{10}=-(1100)_2=(1,0100)_{2cns}$

$$\begin{array}{r} 00101 \\ +\ \ 10100 \\ \hline 11001 \end{array}$$

$(1,1001)_{2cns} = -(0111)_2 = -(7)_{10}$

【例 3-26】 使用 $n=5$ 的补码系统计算 $-(9)_{10}-(5)_{10}$。

解 $-(9)_{10} = -(1001)_2 = (1,0111)_{2cns}$

$-(5)_{10} = -(0101)_2 = (1,1011)_{2cns}$

$$\begin{array}{r} 10111 \\ +\ \ 11011 \\ \hline 110010 \end{array}$$

丢掉进位，即

$$(1,0010)_{2cns} = -(1110)_2 = -(14)_{10}$$

【例 3-27】 图 3-70 所示为用 74X283 和 74X157 实现的 4 位二进制数 A($A_3A_2A_1A_0$)、B($B_3B_2B_1B_0$) 的加、减运算电路，试分析其工作原理。

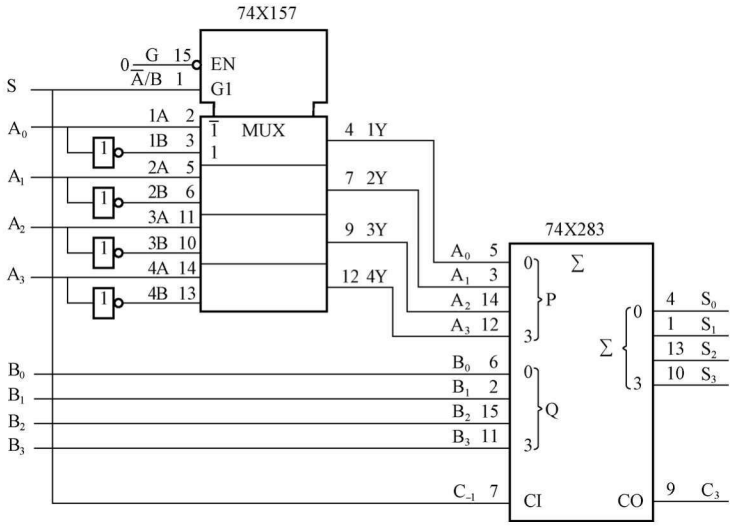


图 3-70 ［例 3-27］的逻辑电路图

解 当 S=0 时，74X283 的 7 脚与 S 相同，为 0。74X157 选择 A_3、A_2、A_1、A_0 送给其输出端，到达 74X283 的 12 脚、14 脚、3 脚、5 脚（A_3、A_2、A_1、A_0）输入端，然后与 $B_3B_2B_1B_0$ 进行加法运算。

当 S=1 时，74X283 的 7 脚与 S 相同，为 1。74X157 选择经过反相后的 A_3、A_2、A_1、A_0 送入其输出端，到达 74X283 的 12 脚、14 脚、3 脚、5 脚（A_3、A_2、A_1、A_0）输入端，与 B_3、B_2、B_1、B_0 进行加法运算，并且还与 74X283 的 7 脚的 1 相加，完成了 B 加 A 的补码的运算，即 B－A 的运算。B－A 的运算可以转换成 B 加 A 的补码。

第八节　数值比较器

在一些数字系统中，经常要求比较两个二进制数的大小。为了实现数值大小的比较功能，人们专门设计出各种用于数值比较的数字逻辑电路，称这些数字逻辑电路为数值比较器。数值比较器的作用就是用于比较两个数值的大小。

一、1 位二进制数值比较器

将两个 1 位数 A 和 B 进行比较大小，一般有 3 种可能，即 A＞B、A＜B 和 A＝B 。由此可知该比较器应有两个输入端 A、B；3 个输出端 $F_{A>B}$，$F_{A<B}$和 $F_{A=B}$，规定输出为高电平有效。比较结果共有 3 种情况：当 A＞B 时，$F_{A>B}=1$；当 A＝B 时，$F_{A=B}=1$；当 A＜B 时，$F_{A<B}=1$。可以得到 1 位二进制数值比较器的真值表，如表 3 - 23 所示。

表 3 - 23　　1 位二进制数值比较器的真值表

输入		输出			输入		输出		
A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$	A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	0	0	1

由真值表 3 - 23 得出的输出逻辑表达式为

$$F_{A>B}=A\overline{B} \tag{3-31}$$

$$F_{A<B}=\overline{A}B \tag{3-32}$$

$$F_{A=B}=\overline{A}\,\overline{B}+AB=\overline{\overline{A}B+A\overline{B}} \tag{3-33}$$

其逻辑电路图如图 3 - 71 所示。

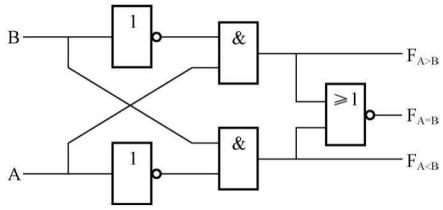


图 3 - 71　1 位二进制数值比较器逻辑电路图

请读者思考若规定输出为低电平有效，该如何设计电路？

二、2 位二进制数值比较器

1 位二进制数值比较器只能对两个 1 位二进制数进行比较，而实用的二进制数值比较器一般是多位的。下面以 2 位为例讨论这种数值比较器。2 位二进制数值比较器的功能表如表 3 - 24 所示。两个数进行比较时，先对高位比较，若高位已经有结果，低位就没必要再比较，只有当高位相等时，才对低位再进行比较。

表 3 - 24　　2 位二进制数值比较器的功能表

数值输入				输出			数值输入				输出		
A_1	B_1	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$	A_1	B_1	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1>B_1$		×	×	1	0	0	$A_1=B_1$		$A_0<B_0$		0	1	0
$A_1<B_1$		×	×	0	1	0	$A_1=B_1$		$A_0=B_0$		0	0	1
$A_1=B_1$		$A_0>B_0$		1	0	0							

比较 A_1A_0 和 B_1B_0 这两个二进制数的大小，2 位二进制数值比较器的真值表如表 3 - 25 所示。其中，A_1、B_1、A_0、B_0 为数值输入端，$F_{A>B}$、$F_{A<B}$、$F_{A=B}$为本位片 3 种不同比较结果输出端，规定输出端为高电平有效。

表 3-25　　2 位二进制数值比较器的真值表

输入				输出			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	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	0	0	1	1	0	0	0	1	0	0
0	0	0	1	0	1	0	1	0	0	1	1	0	0
0	0	1	0	1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	0	1	1	0	1	1	1	0	0
0	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	0	1	0	1	0	1	1	0	1	0	1	0
0	1	1	0	0	1	0	1	1	1	0	1	0	0
0	1	1	1	0	1	0	1	1	1	1	0	0	1

画出 3 个输出信号 $F_{A>B}$、$F_{A<B}$、$F_{A=B}$的卡诺图如图 3-72 所示。

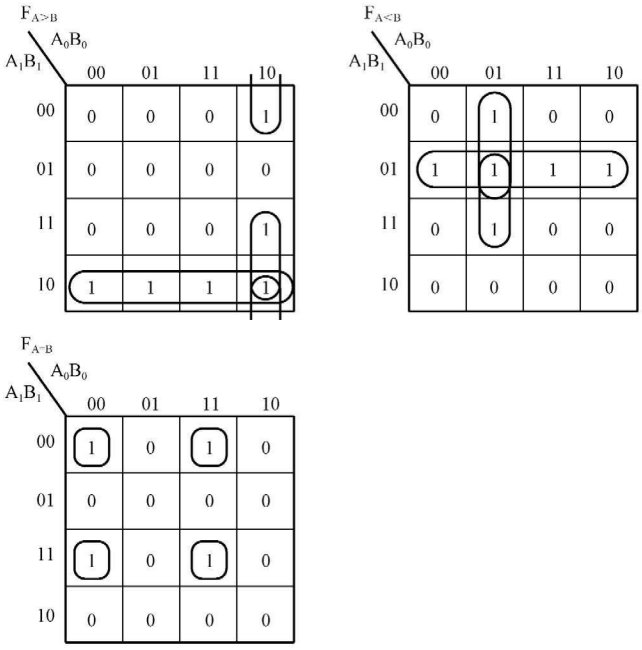


图 3-72　2 位二进制数值比较器的卡诺图

由卡诺图化简得到最简与或表达式为

$$F_{A>B}=A_1\overline{B_1}+A_1A_0\overline{B_0}+\overline{B_1}A_0\overline{B_0}$$

$$F_{A<B}=\overline{A_1}B_1+\overline{A_1}\,\overline{A_0}B_0+B_1\overline{A_0}B_0$$

$$F_{A=B}=\overline{A_1}\,\overline{B_1}\,\overline{A_0}\,\overline{B_0}+\overline{A_1}\,\overline{B_1}A_0B_0+A_1B_1\overline{A_0}\,\overline{B_0}+A_1B_1A_0B_0$$

由最简与或表达式可得逻辑电路图如图 3-73 所示。

三、集成 4 位二进制数值比较器 74X85

集成二进制数值比较器 74X85 是典型的集成 4 位二进制数值比较器。图 3-74 是加了引脚名字的 74X85 的逻辑符号。

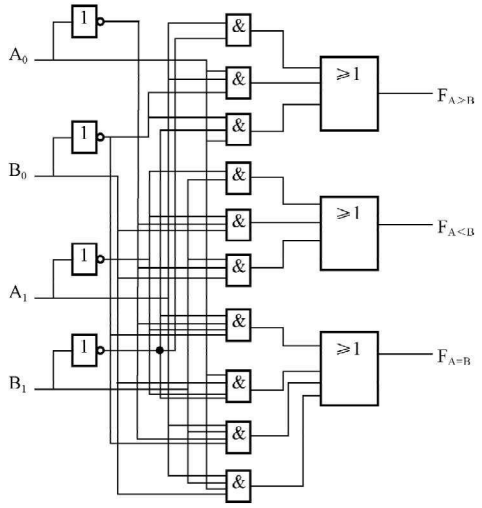


图 3 - 73　2 位二进制数值比较器的逻辑电路图

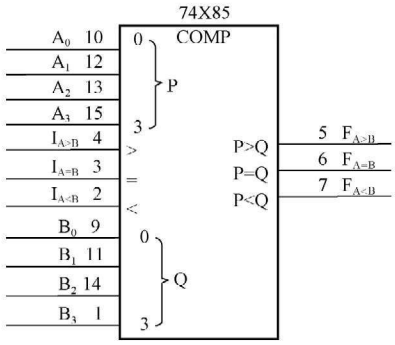


图 3 - 74　加了引脚名字的 74X85 的逻辑符号

4 位二进制数值比较器 74X85 的逻辑功能表如表 3 - 26 所示。其逻辑功能是比较 A 和 B 这两个二进制数的大小。A 是 4 位二进制数，用 $A_3A_2A_1A_0$ 表示；B 是 4 位二进制数，用 $B_3B_2B_1B_0$ 表示。A_3、B_3、A_2、B_2、A_1、B_1、A_0、B_0 为数值输入端，$F_{A>B}$、$F_{A<B}$、$F_{A=B}$ 为本位片三种不同比较结果输出端，规定输出端为高电平有效。两个数进行比较时，先对高位比较，若高位已经有结果，低位就没必要再比较，只有当高位相等时，才对低位再进行比较。74X85 还设计了 3 个级联输入端，用于功能的扩展。从逻辑功能表的倒数第 3 行至倒数第 6 行看出，当 $A_3A_2A_1A_0=B_3B_2B_1B_0$ 时，比较的结果决定于“级联输入”端，“级联输入”端是高电平有效。逻辑功能表的倒数 3 行没有什么实际意义，只是为了说明当这 3 个输入端处于其他输入组合情况下输出的逻辑电平。

表 3 - 26　　4 位二进制数值比较器 74X85 的逻辑功能表

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
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	H	L	L	H
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	×	×	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

在图 3 - 74 所示的逻辑符号中，总定性符 COMP 表示此电路为一个比较器电路，被比较的两个 4 位二进制数 $A_3A_2A_1A_0$、$B_3B_2B_1B_0$ 和级联输入等输入信号列在左边，表示比较结果的 3 个输出信号列在右边，0～3 规定了 4 位二进制数各位的位权。

四、数值比较器功能的扩展

当要比较的二进制数的位数多于 4 位时，用一片 74X85 不能够实现，所以要进行位数的扩展。扩展的方法有两种：串联方式扩展和并联方式扩展。

（一）串联方式扩展

要比较两个 8 位二进制数 $A_7A_6A_5A_4A_3A_2A_1A_0$ 和 $B_7B_6B_5B_4B_3B_2B_1B_0$ 的大小，可以用两片 4 位二进制数值比较器扩展连接成 8 位二进制数值比较器。将两片芯片串联连接，即将低位芯片的输出端 $F_{A>B}$、$F_{A<B}$和 $F_{A=B}$分别接高位芯片级联输入端 $I_{A>B}$、$I_{A<B}$和 $I_{A=B}$，串联的逻辑电路图如图 3 - 75 所示。当高四位都相等时，就可由低四位来决定两数的大小。

在图 3 - 75 中，74X85（1）是低位片，$A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 从 74X85（1）输入，74X85（2）是高位片，$A_7A_6A_5A_4$ 和 $B_7B_6B_5B_4$ 从 74X85（2）输入。低位片 74X85（1）的 4 脚 $I_{A>B}$接 0，2 脚 $I_{A<B}$接 0，3 脚 $I_{A=B}$接 1。此外 $I_{A=B}$端接 1，表示 A_0 和 B_0 就是位数最低的位，没有比其更低的数位了。因此，可以认为 A_0 和 B_0 更低的数位的比较结果为相等，所以 $I_{A=B}$端接 1。低位片 74X85（1）根据 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 比较的结果在 5 脚 $F_{A>B}$、6 脚 $F_{A=B}$、7 脚 $F_{A<B}$分别输出响应的高电平，然后送给 74X85（2）高位片。

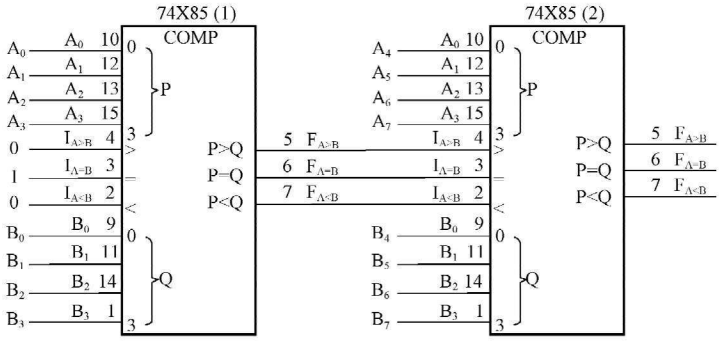


图 3 - 75　4 位比较器扩为 8 位比较器的逻辑电路图

当 74X85（2）高位片的输入信号 $A_7A_6A_5A_4$ 和 $B_7B_6B_5B_4$ 能区分出大小，这时就不用参考低位片 74X85（1）送来的 $I_{A>B}$、$I_{A<B}$和 $I_{A=B}$信号。只有当 $A_7A_6A_5A_4=B_7B_6B_5B_4$ 时，才根据低位片 74X85（1）送来的 $I_{A>B}$、$I_{A<B}$和 $I_{A=B}$信号来决定比较的结果。若 $I_{A>B}=1$，则 A>B；若 $I_{A<B}=1$，则 A<B；若 $I_{A=B}=1$，则 A=B。

（二）并联方式扩展

当两个要比较的二进制数的位数较多，且速度要求较快时，可以采用并联方式扩展。例如，可以用 5 片 4 位二进制数值比较器扩展为 16 位比较器，如图 3 - 76 所示。两个二进制数分别用 $A=A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$ 和 $B=B_{15}B_{14}B_{13}B_{12}B_{11}B_{10}B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0$ 表示。注意在图 3 - 76 中，74X85（5）～74X85（1）5 个芯片的 4 脚、3 脚、2 脚分别接 0、1、0。将待比较的两个 16 位二进制数分成 4 组，$A_{15}A_{14}A_{13}A_{12}$和

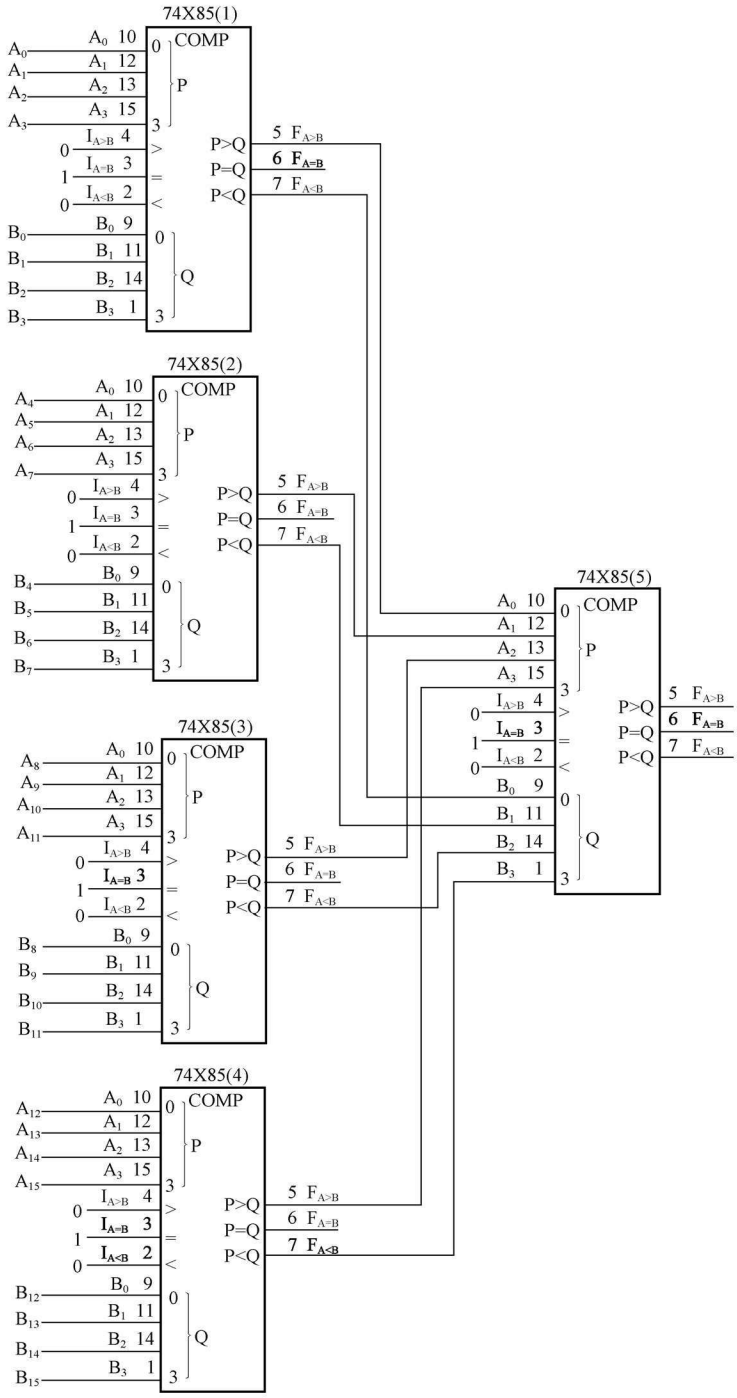


图 3 - 76　比较器并联方式扩展图

$B_{15}B_{14}B_{13}B_{12}$ 为一组，送入 74X85（4）去比较；$A_{11}A_{10}A_9A_8$ 和 $B_{11}B_{10}B_9B_8$ 为一组，送入 74X85（3）去比较；$A_7A_6A_5A_4$ 和 $B_7B_6B_5B_4$ 为一组，送入 74X85（2）去比较；$A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 为一组，送入 74X85（1）去比较。各组的 4 位二进制数的比较是同时进行的，

各组得到比较的结果后，分别从对应芯片的 5 脚、6 脚、7 脚输出结果，然后再将每组的比较结果输入到 74X85（5）去进行比较，最后得出比较结果。

要特别注意，在图 3 - 76 中，74X85（4）片的 5 脚 $F_{A>B}$ 一定要与 74X85（5）片的 15

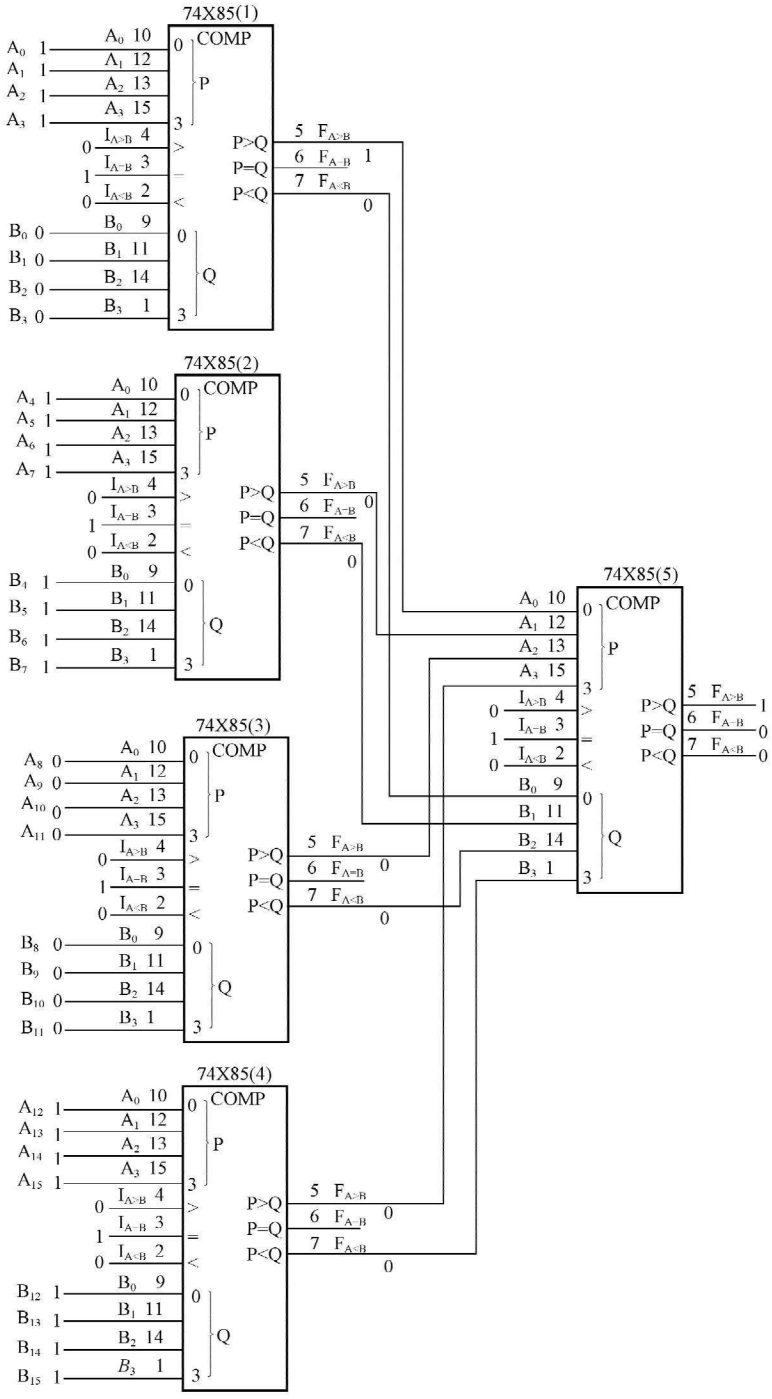


图 3 - 77 ［例 3 - 28］的逻辑电路图

脚 A_3 连接，74X85（4）片的 7 脚 $F_{A<B}$ 一定要与 74X85（5）片的 1 脚 B_3 连接；74X85（3）片的 5 脚 $F_{A>B}$ 一定要与 74X85（5）片的 13 脚 A_2 连接，74X85（3）片的 7 脚 $F_{A<B}$ 一定要与 74X85（5）片的 14 脚 B_2 连接；74X85（2）片的 5 脚 $F_{A>B}$ 一定要与 74X85（5）片的 12 脚 A_1 连接，74X85（2）片的 7 脚 $F_{A<B}$ 一定要与 74X85（5）片的 11 脚 B_1 连接；74X85（1）片的 5 脚 $F_{A>B}$ 一定要与 74X85（5）片的 10 脚 A_0 连接，74X85（1）片的 7 脚 $F_{A<B}$ 一定要与 74X85（5）片的 9 脚 B_0 连接。

如果 $A_{15}A_{14}A_{13}A_{12}>B_{15}B_{14}B_{13}B_{12}$，则 74X85（4）片的 5 脚 $F_{A>B}=1$，7 脚 $F_{A<B}=0$。这时 74X85（5）片的 15 脚 $A_3=1$；74X85（5）片的 1 脚 $B_3=0$，$A_3>B_3$；74X85（5）片的 5 脚 $F_{A>B}$ 就输出高电平，表示 A＞B。

这种并联扩展方式从数据输入到输出只需要两倍的 4 位二进制数值比较器的延迟时间，如果采用串联扩展方式时，则需要 4 倍的 4 位二进制数值比较器的延迟时间。

【例 3 - 28】 已知两个 16 位二进制数分别为 A＝1111000011111111，B＝1111000011110000。试用 74X85 并联扩展方式设计比较 A、B 大小的电路，要求画出逻辑电路图，并在图中标明电路中各个信号的具体数值。

解 比较 A、B 大小的逻辑电路图如图 3 - 77 所示。图中已标明了各个信号的具体数值。

【例 3 - 29】 用 74X85 设计一个比较两个 7 位二进制数 $A_6A_5A_4A_3A_2A_1A_0$ 和 $B_6B_5B_4B_3B_2B_1B_0$ 大小的电路。

解 解法一：一片 74X85 只能对 4 位二进制数进行比较，题目要求对两个七位二进制数进行比较，所以必须对 74X85 进行扩展。如果采用串联扩展方式，要采用两片 4 位二进制数值比较器 74X85，其逻辑电路图如图 3 - 78 所示。片（1）的 $I_{A>B}=0$、$I_{A<B}=0$、$I_{A=B}=1$。74X85（2）的 15 脚和 1 脚应该接相同的电平，1 和 0 均可。

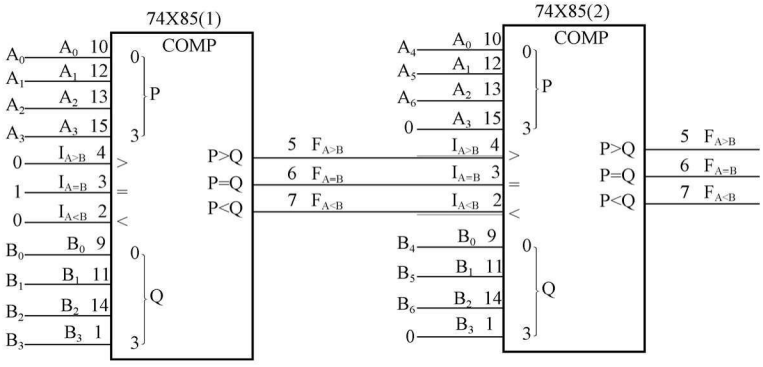


图 3 - 78 ［例 3 - 29］的解法一的逻辑电路图

解法二：如果采用并联扩展方式，要采用 3 片 4 位二进制比较器 74X85，其逻辑电路图如图 3 - 79 所示。

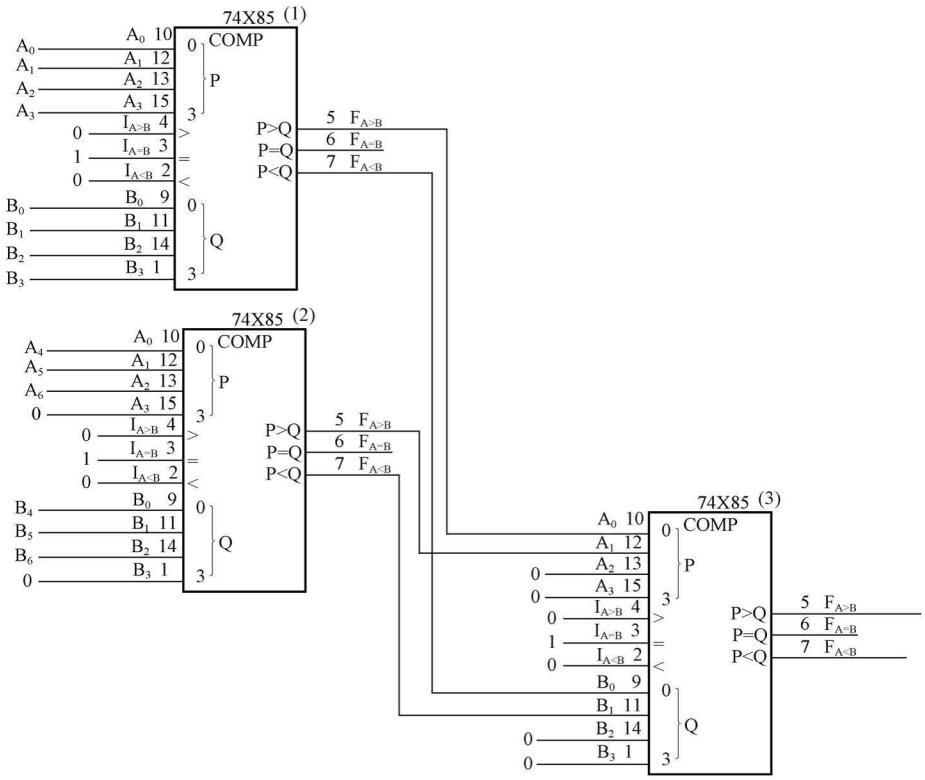


图 3 - 79 ［例 3 - 29］的解法二的逻辑电路图

小　　结

在任意时刻，组合逻辑电路的输出信号仅仅取决于该时刻电路的各个输入信号，而与该时刻之前的输入信号的取值无关。组合逻辑电路由门电路组合而成，电路中没有记忆单元，没有反馈通路。

组合逻辑电路的分析步骤为：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。

组合逻辑电路的设计步骤为：根据设计要求列出真值表→写出逻辑表达式（或填写卡诺图）→逻辑化简和变换→画出逻辑电路图。

组合逻辑电路的设计中应采取有效措施，避免发生竞争冒险。

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、算术逻辑运算电路等。这些组合逻辑器件除了具有基本逻辑功能以外，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。

应用中规模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采用门电路设计时基本一致，但也有其特殊之处：

（1）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。

（2）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来满足设计要求。

（3）可能出现只需一个组合器件的部分功能就可以满足设计要求的情况，这时需要对有关输入、输出信号作适当的处理；也可能出现一个组合器件不能满足设计要求的情况，这就需要对组合逻辑器件进行扩展，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

3－1　有一个组合逻辑部件，其内部结构不详，但测得其输入信号 A、B、C 和输出信号 Z 的波形如图3－80 所示。试写出其标准与或式。

3－2　某逻辑电路如图 3－81 所示，试分析其逻辑功能（包括逻辑表达式、真值表，并说明其逻辑功能）。

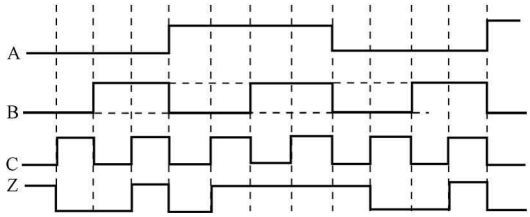


图 3－80　题 3－1 图

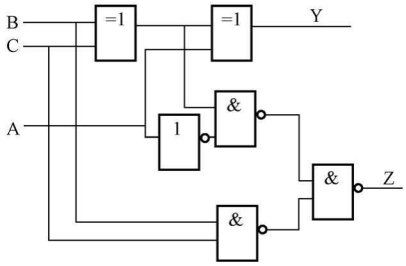


图 3－81　题 3－2 图

3－3　电路如图 3－82 所示，试分析当输入 A、B、C、D 为何种取值组合时有

（1）$F_1=1$；

（2）$F_2=1$；

（3）$F_1=F_2$。

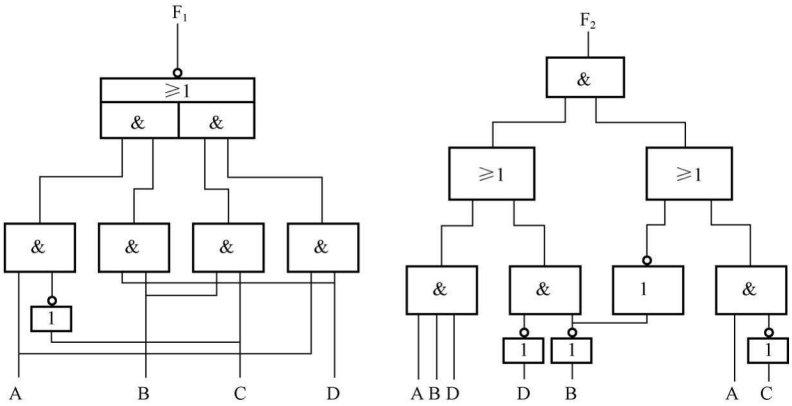


图 3－82　题 3－3 图

3－4　某物料传送系统示意图如图 3－83 所示。系统由 A、B、C 三台电动机拖动，为防止物料堆积，规定只有 C 开机时 B 才可开机，只有 B 开机时 A 才可开机，否则应给出告

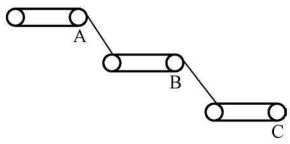


图 3-83 题 3-4 图

警信号。设 A、B、C 开机时状态信号为 1。试设计产生告警信号的逻辑电路，所用门电路可任选。

3-5 有 A、B、C、D 四对话路，正常工作时，最多只允许两对同时通话，并且 A 和 B、C 和 D、A 和 D 不允许同时通话。要求设计一个不能正常工作时，能立即显示的逻辑电路（用与或非门实现）。

3-6 试用与非门设计一个 2 位二进制数相加的逻辑电路。提示

$$\begin{array}{rcc} & A_1 & A_0 \\ + & B_1 & B_0 \\ \hline C_1 & S_1 & S_0 \end{array}$$

其中：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。

3-7 有一个三线排队的组合电路，A、B、C 为三路输入信号，F_1、F_2、F_3 为其对应的输出，电路在同一时间只允许通过一路信号，且优先权的顺序为 A、B、C。试用与非门组成三线排队电路，写出逻辑表达式。

3-8 设输入信号为 A、B、C，试分别设计无优先级和有优先级的编码器电路对输入信号进行编码，优先权的顺序为 A、B、C。

3-9 保密电锁上有 3 个键钮 A、B、C，要求当 3 个键钮同时按下时，或 A、B 两个同时按下时，或按下 A、B 中的任一键时，锁能打开；而当不符合上列组合状态时电铃报警（当没有按键按下时锁不打开也不报警）。试设计一个组合逻辑电路，并要求用两输入与非门实现。

3-10 试利用 3 线—8 线译码器 74X138 集成电路，并加入必要的门电路实现一个判别电路。输入 ABCD 为 4 位二进制代码，当输入代码能被 4 整除时电路输出为 1，否则为 0。

3-11 试用与非门设计一个译码器，译出对应 ABCD 分别为 0010、1010、1110 状态的 3 个信号。

3-12 在举重比赛中，有 3 名裁判，其中 1 名主裁判和两名副裁判。比赛时，只有主裁判判定运动员成绩有效，加上至少 1 名副裁判判定运动员成绩有效时，该运动员的成绩才有效。列出真值表，并用或非门实现该逻辑电路。

3-13 用异或门设计一个将 3 位二进制码转换成循环码的逻辑电路，其真值表如表 3-27 所示。写出该电路的逻辑表达式，并画出逻辑图。

表 3-27 题 3-13 真值表

二进制码 $A_2A_1A_0$	循环码 $F_2F_1F_0$	二进制码 $A_2A_1A_0$	循环码 $F_2F_1F_0$	二进制码 $A_2A_1A_0$	循环码 $F_2F_1F_0$
000	000	011	010	110	101
001	001	100	110	111	100
010	011	101	111		

3-14 判断下列函数对应的逻辑电路中有无静态冒险。若有，说明是静态 0 险，还是静态 1 冒险。修改设计消除静态冒险。

(1) $F(A,B,C,D) = AB + A\overline{B}C$；

(2) $F(A,B,C,D)=AC+\overline{B}C$；

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

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

3-15　试用一片译码器 74X138 和与非门实现下列多输出组合逻辑函数：

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

(2) $L(A,B,C)=\sum m(0,2,3,6,7)$；

(3) $M(A,B,C)=(A+C)(\overline{A}+B+C)$。

3-16　为了使 74X138 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。

3-17　用译码器 74X138 和适当的逻辑门实现函数 $F=\overline{A}\,\overline{B}\,\overline{C}+A\overline{B}\,\overline{C}+AB\overline{C}+ABC$。

3-18　应用 74X151 实现如下逻辑函数：

(1) $Y=A\overline{B}\,\overline{C}+A\overline{B}C+\overline{A}\,\overline{B}C$；

(2) $Y=(A\odot B)\odot C$。

3-19　试用 4 选 1 数据选择器 74X153 产生 1100 序列信号。

3-20　试用下列方法构成一个全减器，可以附加必要的逻辑门：

(1) 用双 4 选 1 数据选择器 74X153 实现。

(2) 用 74X138 译码器实现，并具有控制端 K。K=0 时，全减器功能被禁止；K=1 时，作全减运算。

3-21　试用两片 4 位二进制数值比较器 74X85 实现 3 个 4 位二进制数 A、B、C 的最大、最小和相等关系的比较。

3-22　由 4 位超前进位全加器 74X283 器件组成的代码转换电路如图 3-84 所示。试分析电路的转换功能。输入 ABCD 为余 3 码，输出用 MNOP 表示。

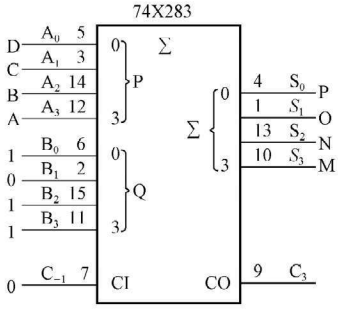


图 3-84　题 3-22 图

3-23　用 4 位全加器 74X283 实现 8421BCD 码至 5421BCD 码的转换。

3-24　设计一个码变换电路，可以把带符号的负的二进制数（包括符号位在内共 8 位），转换成该数的补码。要求用 4 位二进制全加器 74X283 和门电路实现。

3-25　已知 8 选 1 数据选择器 74LS151 芯片的选择输入端 A_2 的引脚折断，无法输入信号，但芯片内部功能完好。试问如何用它来实现函数 $F(A,B,C)=\sum m(2,3,4,7)$，画出逻辑电路图。

3-26　设计一个多功能组合逻辑电路，要求实现表 3-28 所示的功能。其中 M_1、M_0 为功能选择信号；A、B 为输入逻辑变量；F 为输出逻辑变量。试用 74X151 和门电路实现该电路。

表 3-28　　题 3-26 电路的功能表

M_1	M_0	F	M_1	M_0	F
0	0	$\overline{A+B}$	1	0	$A\oplus B$
0	1	$\overline{AB}$	1	1	$A\odot B$

3-27 分析图3-85所示的电路：

(1) 写出F、G、L的标准与或表达式；

(2) 化简得到最简或与式。

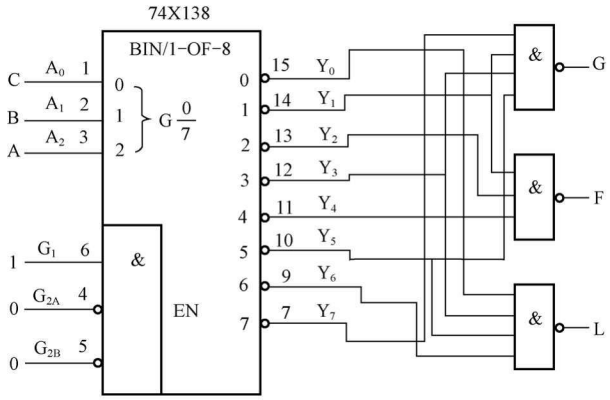


图3-85 题3-27图

3-28 设计一个能实现两个1位二进制数的全加运算和全减运算的组合逻辑电路，加减控制信号用M表示，当M=0时为全加运算，M=1时为全减运算。要求：用适当的门电路实现（画出逻辑电路图）。

3-29 设计一个电动机的故障指示电路，要求两台电动机同时工作时，绿灯亮；一台电动机发生故障时，黄灯亮；两台电动机同时故障时，红灯亮。要求用与非门实现。

3-30 设计一个血型配对指示器。输血时供血者和受血者的血型配对情况是：同一血型之间可以相互输血；AB型受血者可以接受任何血型的输出；O型血者可以给任何血型的受血者输血。要求：

(1) 用最少个数的与非门实现（画出逻辑电路图）；

(2) 用74X151实现（画出逻辑电路图）。

3-31 设计一个组合逻辑电路，输入为4位二进制数N，当$N \geqslant 9$时输出为1，否则为0。要求：

(1) 用最少个数的或非门实现（画出逻辑电路图）；

(2) 用74X151实现（画出逻辑电路图）；

(3) 用74X138加必要的门电路实现（画出逻辑电路图）。

3-32 设计一个能实现两个1位二进制数的全加运算和全减运算的组合逻辑电路。加减控制信号用M表示，当M=0时为全加运算，M=1时为全减运算。要求：

(1) 用最少个数的与非门实现（画出逻辑电路图）；

(2) 用74X151实现（画出逻辑电路图）；

(3) 用74X138加必要的门电路实现（画出逻辑电路图）。

3-33 CD4532B是一款CMOS 8线—3线集成优先编码器芯片，与74X148的功能非常类似，其功能表如表3-29所示。试说明其功能。

表 3-29　　题 3-33 芯片 CD4532B 的功能表

输入									输出				
EI	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	GS	Q_2	Q_1	Q_0	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	1	0	0
1	0	0	1	×	×	×	×	×	1	1	0	1	0
1	0	0	0	1	×	×	×	×	1	1	0	0	0
1	0	0	0	0	1	×	×	×	1	0	1	1	0
1	0	0	0	0	0	1	×	×	1	0	1	0	0
1	0	0	0	0	0	0	1	×	1	0	0	1	0
1	0	0	0	0	0	0	0	1	1	0	0	0	0

3-34　74X147 是一款 10 线—4 线集成优先编码器芯片，其功能表如表 3-30 所示，逻辑符号如图 3-86 所示。试回答下列问题：

（1）芯片有几个输入端，几个输出端？

（2）输入信号的优先级别是什么？输入信号是低电平有效还是高电平有效？

（3）对 0～9 这 10 个十进制的数码的编码分别是什么？

（4）如果想得到 0～9 这 10 个十进制的数码的 8421BCD 编码，该如何设计电路？

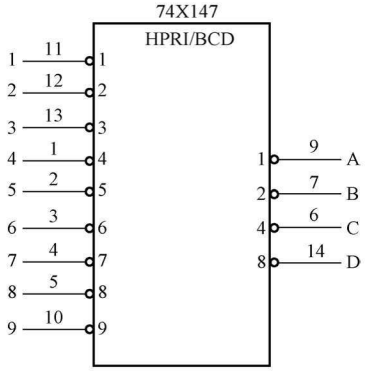


图 3-86　题 3-34 图

表 3-30　　题 3-34 芯片 74X147 的功能表

输入									输出			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
×	×	×	×	×	×	×	×	L	L	H	H	L
×	×	×	×	×	×	×	L	H	L	H	H	H
×	×	×	×	×	×	L	H	H	H	L	L	L
×	×	×	×	×	L	H	H	H	H	L	L	H
×	×	×	×	L	H	H	H	H	H	L	H	L
×	×	×	L	H	H	H	H	H	H	L	H	H
×	×	L	H	H	H	H	H	H	H	H	L	L
×	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

第四章 触 发 器

本 章 提 要

本章介绍构成数字系统的另一种基本逻辑单元——锁存器和触发器。它们是能够存储1位二进制信息的逻辑电路，有两个能自行保持的稳定状态，在适当的触发信号作用下，会从一个稳态跃变到另一个稳态。电路有两个互补输出端，具有记忆功能，其输出不仅与输入有关还与现在的状态有关。锁存器和触发器是最简单的时序逻辑电路，或者说是构成时序逻辑电路的基本单元电路。

本章主要内容包括锁存器和触发器的各种电路结构以及由于电路的结构不同而带来的不同动作特点、脉冲工作特性，逻辑功能表示方法、分类及相互之间的转换，几种典型集成触发器的主要性能指标。

按结构分主要有基本RS锁存器、逻辑门控RS锁存器、主从触发器、边沿触发器；按控制方式（即信号的输入方式以及触发器状态随输入信号变化的规律）的不同，触发器具有不同的逻辑功能，主要有RS触发器、JK触发器、D触发器、T触发器、T′触发器。

描述触发器功能的基本方法有功能表、特性方程、状态转换图、时序图。

第一节 触发器的电路结构及工作特点

锁存器和触发器是构成各种时序电路的存储单元电路，其共同特点是都具有0和1两种稳定状态，一旦状态被确定，就能自行保持，即长期存储1位二进制码，直到有外部信号作用时才有可能改变。锁存器是一种对脉冲电平敏感的存储单元电路，可以在特定输入脉冲电平作用下改变状态。而触发器则是一种对脉冲边沿敏感的存储电路，它们只有在时钟脉冲上升沿或下降沿的变化瞬间才能改变状态。

一、基本RS锁存器

基本RS锁存器是构成各类触发器的最基本逻辑单元，可以用来存储1位二进制的数码0和1。它可以用与非门组成，也可以用或非门组成。

（一）与非门组成的基本RS锁存器的电路结构和逻辑符号

由与非门组成的基本RS锁存器的逻辑电路图和逻辑符号如图4-1所示。从图4-1（a）可以看出，基本RS锁存器的两个与非门的输入、输出端交叉耦合，G1门的输出端送入G2门的输入端，G2门的输出端送入G1门的输入端。图4-1（b）是基本RS锁存器的另外一种画法，观察此图可以看出，此图与前面学过的组合逻辑电路的最大区别就是G2门的输出端又反馈送入G1门的输入端，这是时序逻辑电路的最根本的特征。基本RS锁存器有两个输入端，分别是R、S，输入是低电平有效。在图4-1（c）所示的逻辑符号中，R和S输入端边框外侧都画有小圆圈，表明输入是低电平有效。锁存器有两个输出端Q、$\overline{Q}$，通常把锁存器Q端的状态称为锁存器的状态。当Q=1、$\overline{Q}$=0时，称锁存器处于1

状态或置位状态；当 Q=0、$\overline{Q}$=1 时，称锁存器处于 0 状态或复位状态。一般情况下，Q、$\overline{Q}$的状态互补。

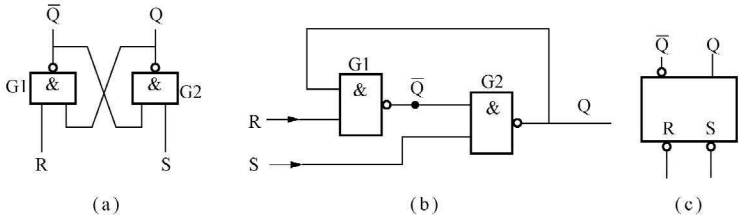


图 4-1　与非门组成的基本 RS 锁存器

（a）逻辑电路图；（b）逻辑电路图的另一种画法；（c）逻辑符号

（二）逻辑功能分析

根据输入信号 R 和 S 不同状态的组合，锁存器的输出与输入的关系有四种情况，下面分别进行分析。

1. R=0、S=1

由于 R=0，无论电路中原来的状态如何，都有$\overline{Q}$=1，再由 S=1 和$\overline{Q}$=1 共同作用，可得 Q=0。锁存器此时的状态为 0 态，称锁存器清 0 或复位。Q 接回到 G1 门的一个输入端，即使 R 由低电平变为高电平，保证了锁存器仍能维持 0 状态不变。由于使锁存器为 0 状态的输入信号是 R，所以 R（Reset）称为置 0 输入端，又称复位端，低电平有效。

2. R=1、S=0

由于 S=0，无论电路中原来的状态如何，都有 Q=1，再由 R=1 和 Q=1 共同作用，可得$\overline{Q}$=0。锁存器此时的状态为 1 态，称锁存器置 1 或置位。$\overline{Q}$接回到 G2 门的一个输入端，即使 S 由低电平变为高电平，保证了锁存器仍能维持 1 状态不变。由于使锁存器为 1 状态的输入信号是 S，所以 S（Set）称为置 1 输入端，又称置位端，低电平有效。

3. R=1、S=1

如果锁存器原来处于 0 态，即 Q=0，$\overline{Q}$=1。Q=0 加到 G1 门的输入端，必能保证$\overline{Q}$=1，而$\overline{Q}$=1 又加到 G2 门的输入端，因 R=S=1，所以能保证 Q=0，可见能继续保持 0 状态。

如果锁存器的原状态为 1，也可以保持 1 态不变。这体现了锁存器具有记忆功能，把锁存器的这种工作状态称为保持原状态。

4. R=0、S=0

显然，在此条件下，两个锁存器的输出端 Q 和$\overline{Q}$全为 1，破坏了两个输出信号互补的规则，从存储二进制信号的角度看，既不是存储了 0，也不是存储了 1。当 R 和 S 同时由 0 跳变到 1 状态以后，由于两个与非门的传输延迟时间存在差别，使锁存器的输出状态可能是 0，也可能是 1，这取决于两个与非门的传输延迟时间的大小。当 G1 门的传输延迟时间小于 G2 门的传输延迟时间，$\overline{Q}$先变为 0，从而使 Q=1，锁存器为 1 状态；当 G1 门的传输延迟时间大于 G2 门的传输延迟时间，Q 先变为 0，从而使$\overline{Q}$=1，锁存器为 0 状态；当 G1 门的传输延迟时间和 G2 门的传输延迟时间完全相同，当 R 和 S 同时由 0 跳变到 1 状态以后，$\overline{Q}$和 Q 同时变 0，$\overline{Q}$和 Q 的 0 同时送入 G2 门和 G1 门的输入端，使$\overline{Q}$和 Q 同时变 1，从而在输出端引起振荡。

当 R 和 S 同时由 0 跳变到 1 状态以后，由于与非门的传输延迟时间具有分散性，锁存

器的状态不定，锁存器的这种工作状态称为不定态，要避免这种情况发生。如果 R 和 S 的低电平不同时消失，输出的状态是可以确定的。在正常使用锁存器时，需要对输入信号加约束条件，即 R+S=1，保证不出现 R=S=0 的情况，避免出现不定态。

如果锁存器的初始状态为 1，只有 R=0，S=1，才能使锁存器变为 0 态，如果锁存器的初始状态为 0，只有 S=0，R=1，才能使锁存器变为 1 态。锁存器 1 到 0 或者由 0 到 1 的变化过程称为翻转。

若想让基本 RS 锁存器存储 0，只要在复位输入端 R 输入有效的低电平（此时置位输入端 S 必须处于无效的高电平状态），就可以使 Q=0，$\overline{Q}$=1；若想让基本 RS 锁存器存储 1，只要在置位输入端 S 输入有效的低电平（此时复位输入端 R 必须处于无效的高电平状态），就可以使 Q=1，$\overline{Q}$=0。

综上所述，将与非门组成的基本 RS 锁存器的功能列成真值表如表 4-1 所示。锁存器接收输入信号之前所处的状态，即锁存器原来的状态，称为现态，用 Q^n 表示。锁存器接收输入信号之后所处的状态，即锁存器新的状态，称为次态，用 Q^{n+1} 表示，次态是输入信号作用后的状态。由于锁存器的次态不仅与输入信号有关，还与锁存器的现态有关，所以在真值表中把现态 Q^n 作为一个变量列入。含有现态 Q^n 的锁存器的真值表又叫做锁存器的特性表（或功能表）。

表 4-1　　与非门组成的基本 RS 锁存器的特性表

R	S	Q^n	Q^{n+1}	功能说明	R	S	Q^n	Q^{n+1}	功能说明
0	0	0	×	不定（不允许）	1	0	0	1	置 1（置位）
0	0	1	×		1	0	1	1	
0	1	0	0	置 0（复位）	1	1	0	0	保持原状态
0	1	1	0		1	1	1	1	

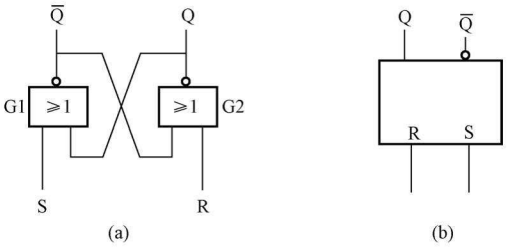


图 4-2　或非门组成的基本 RS 锁存器

（a）逻辑图；（b）逻辑符号

（三）或非门构成的基本 RS 锁存器

基本 RS 锁存器也可以由或非门交叉耦合连接而构成，如图 4-2 所示。两个或非门构成的基本 RS 锁存器和由两个与非门组成的电路比较起来，电路中 R 和 S 的位置对换了，逻辑符号输入端无圆圈，表示输入信号为高电平有效。不难分析，此电路的特性表如表 4-2 所示。对输入信号的约束条件为 RS=0。

表 4-2　　用或非门组成的基本 RS 锁存器的特性表

R	S	Q^n	Q^{n+1}	功能说明	R	S	Q^n	Q^{n+1}	功能说明
0	0	0	0	保持原状态	1	0	0	0	置 0（复位）
0	0	1	1		1	0	1	0	
0	1	0	1	置 1（置位）	1	1	0	×	不定（不允许）
0	1	1	1		1	1	1	×	

基本 RS 锁存器具有电路简单的优点，由电平触发，能存储一位二进制数，是构成各种性能更完善的锁存器的基础。但是因为输入信号直接控制，输入信号在全部时间内都能直接决定 Q 的状态，所以使用基本 RS 锁存器的局限性大，且对输入信号 R 和 S 有约束条件。

【例 4-1】 用与非门组成的基本 RS 锁存器中，设初始状态为 0，已知输入 R、S 的波形，画出 Q 的波形。

解　与非门组成的基本 RS 锁存器如图 4-1 所示，由表 4-1 可知：

当 R=1、S=1 时，保持原状态。当 R=1、S=0 时，锁存器置 1。当 R=0、S=1 时，锁存器置 0。当 R、S 同时是 0 时，Q=1，$\overline{Q}$=1。这时如果 R、S 不同时变 1，状态可定；如果 R、S 同时变 1，状态不定，用斜线区域表示。输入输出波形如图 4-3 所示。

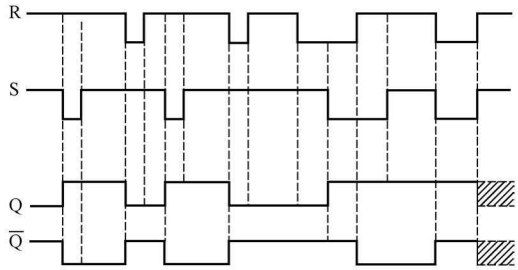


图 4-3　[例 4-1] 输出波形

（四）基本 RS 锁存器的应用

基本 RS 锁存器具有电路简单、操作方便的特点，常被应用于防抖动、信息存储等电路中。

【例 4-2】 某机械开关的电路如图 4-4（a）所示，当机械开关切换时，由于开关触点振动会使电压或电流波形产生“毛刺”，波形如图 4-4（b）所示。在电子电路中开关的抖动会导致电路产生误动作，应当设法消除。利用基本 RS 锁存器的记忆功能可以消除上述开关抖动产生的影响，开关与锁存器的连接方法如图 4-5（a）所示，试分析其工作原理，并画出 Q 的波形。

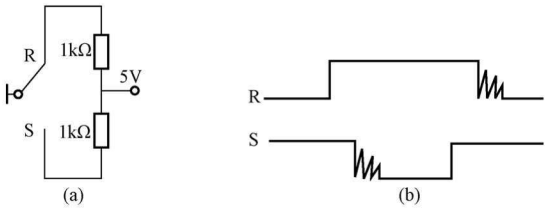


图 4-4　机械开关的工作情况

（a）机械开关电路；（b）开关动作后 R 和 S 端的电压波形

解　已知开关接通 R，此时 R=0、S=1。当开关转向 S 的过程中，会出现 R=S=1 的状态，之后，R=1，S 端由于开关接通过程的振动，会有图示的毛刺现象，最后稳定在 0 上；同理开关再从 S 端转换到 R 端，经历同时为 1 后，S 一直为 1，而 R 端也会出现毛刺然后再稳定在 0 上的现象。因此机械开关动作时 R 和 S 端的电压波形，会出现如图 4-4（b）所示的现象。

如当将 R 和 S 作为锁存器的输入，分析 Q 的波形。开关合向 S 端，R=1，S 一旦出现低电平锁存器的状态 Q 就为 1，即使由于抖动使 S 出现高电平，Q 也保持 1 状态；同理，R 发生抖动，S=1，Q 置 0，之后尽管 R 有 0、1 的变化，Q 仍然保持 0 态。所以 Q 的波形没有毛刺，消除了抖动现象，波形如图 4-5（b）所示。

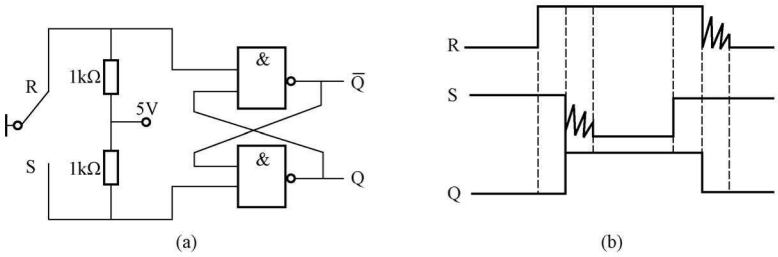


图 4-5　利用基本 RS 锁存器消除机械抖动的影响

（a）电路；（b）电压波形

二、逻辑门控 RS 锁存器

基本 RS 锁存器的状态直接由输入信号控制，而在实际应用中，常常要求锁存器的工作状态不仅受 R、S 端的信号控制，还要求它受某个控制信号的控制。只有当控制信号到达时，锁存器的状态才可以变化，或者说被触发，若控制信号没有到来，即使输入信号 R、S 改变了，锁存器的状态也不会变，为此给锁存器引入一个时钟脉冲信号 CP（Clock Pulse 的缩写）。因为具有时钟脉冲控制的锁存器的状态的改变与时钟脉冲同步，所以被称为逻辑门控锁存器。这种受时钟信号控制的锁存器统称为逻辑门控锁存器。

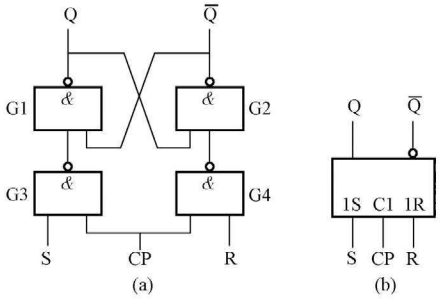


图 4-6　逻辑门控 RS 锁存器

（a）逻辑电路；（b）逻辑符号

（一）电路结构和逻辑符号

逻辑门控 RS 锁存器的电路结构和逻辑符号如图 4-6 所示。其电路包括两个部分：由 G1 和 G2 组成的基本 RS 锁存器；由 G3 和 G4 组成的输入控制电路。

（二）逻辑功能分析

当 CP=0 时，G3 和 G4 门被封锁，输入信号 R、S 不起作用，输出全为 1，基本 RS 锁存器状态不变。

当 CP=1 时，G3 和 G4 门打开，输入信号 R、S 经 G3、G4 门作用于基本 RS 锁存器，使 Q 的状态跟随输入状态的变化而改变，其特性表如表 4-3 所示。

表 4-3　　逻辑门控 RS 锁存器的特性表

CP	R	S	Q^n	Q^{n+1}	CP	R	S	Q^n	Q^{n+1}
0	×	×	×	Q^n	1	1	0	0	0
1	0	0	0	0		1	0	1	0
	0	0	0	1	1	1	1	0	×
1	0	1	0	1		1	1	1	×
	0	1	1	1					

可以看到，只有在 CP＝1 时锁存器的状态才受输入信号的控制，而且特性表中 CP＝1 期间与或非门组成的基本 RS 锁存器的特性表相同，并且输入信号同样有约束条件 RS＝0。电路的三个输入信号中 CP 控制状态转换的时刻，R、S 决定状态转换的方向。

（三）逻辑门控 RS 锁存器功能的几种描述方法

逻辑门控 RS 锁存器的功能除了用特性表来描述之外，还可以用特性方程、状态转换图、驱动表、时序图来描述，下面分别介绍。

1. 特性方程

锁存器次态 Q^{n+1} 与输入信号 R、S 及现态 Q^n 之间的表达式称为特性方程。根据特性表 4-3，通过卡诺图化简（如图 4-7 所示），可以得到逻辑门控 RS 锁存器的特性方程，即

$$\begin{cases} Q^{n+1} = S + \overline{R}Q^n \\ RS = 0(\text{约束条件}) \end{cases}, \text{CP} = 1 \text{ 时有效}$$

2. 状态转换图

状态转换图表示在时钟满足时，锁存器从一个状态变化到另一个状态或保持原状态不变时，对输入信号的要求。用两个圆圈代表两个状态，用箭头表示状态转换的方向，同时在箭头的旁边注明转换的条件，由此构成的图称为状态转换图，它可以更形象地表示出 RS 锁存器的逻辑功能。逻辑门控 RS 锁存器的状态转换图如图 4-8 所示。

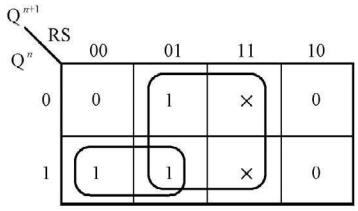


图 4-7　逻辑门控 RS 锁存器 Q^{n+1} 的卡诺图化简

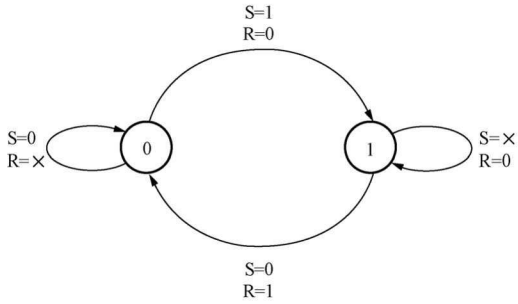


图 4-8　逻辑门控 RS 锁存器的状态转换图

3. 驱动表

驱动表是用表格的方式表示锁存器从一个状态变化到另一个状态或保持原状态不变时，对输入信号的要求。根据表 4-3 获得的逻辑门控 RS 锁存器的驱动表如表 4-4 所示。驱动表对时序逻辑电路的设计是很有用的。

表 4-4　逻辑门控 RS 锁存器的驱动表

$Q^n \to Q^{n+1}$		R	S
0	0	×	0
0	1	0	1
1	0	1	0
1	1	0	×

4. 波形图

触发器的功能也可以用输入输出波形直观地表示出来。

【例 4-3】 已知逻辑门控 RS 锁存器的输入信号波形如图 4-9 所示。分析锁存器输出 Q 的波形（设锁存器的初始状态为 0）。

解　CP＝0 时触发器保持原状；CP＝1 期间，根据表 4-3 分析输入信号及现态，得到相应的输出次状态，画出波形如图 4-9 所示。

（四）逻辑门控 RS 锁存器存在的问题——空翻现象

如图 4-6（a）所示，在 CP=1 的全部时间内 S 和 R 信号通过 G3 和 G4 门加到基本 RS 锁存器上，所以 CP=1 期间，R 和 S 的变化都会引起锁存器状态的响应改变。这就是同步 RS 锁存器的特点。

如 CP=1 期间输入信号多次发生变化，则锁存器的状态可能发生多次翻转，这种现象被叫做空翻，如图 4-10 所示。这大大降低了电路抵御干扰信号的能力。在时钟信号 CP 为 0 期间，锁存器的状态保持不变，锁存器的触发翻转只是被控制在一个时间间隔内，而不能被控制在某一时刻进行。

电路仍有不定状态存在，为避免不定状态的发生，必须对输入信号进行约束，R、S 不能同时为 1，即 RS=0。

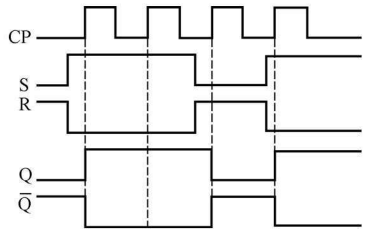


图 4-9 ［例 4-3］的波形图

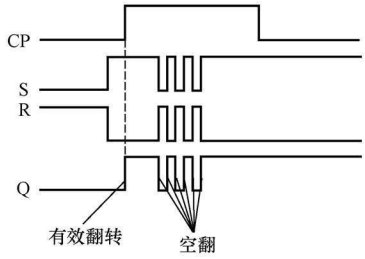


图 4-10 逻辑门控 RS 锁存器的空翻波形

三、主从触发器

主从触发器是由逻辑门控 RS 锁存器改进而得到的。在这种触发器中，先将输入信号接入主锁存器，然后再传送给从锁存器并输出，整个过程分两步进行，具有主从控制特点。

（一）主从 RS 触发器

主从 RS 触发器由两级逻辑门控 RS 锁存器组成，其逻辑电路如图 4-11（a）所示。图中 G5、G6、G7、G8 组成主锁存器，直接接收输入信号，主锁存器的时钟控制信号是 CP；G1、G2、G3、G4 组成从锁存器，接收主锁存器的输出信号，时钟控制信号是$\overline{CP}$。图 4-11（b）为主从 RS 触发器的逻辑符号，其中的“┌”表示主从结构，小圆圈表示 CP 下降沿有效。

1. 主从 RS 触发器的工作原理

图 4-11（a）中，主锁存器接收输入信号，其状态直接由输入信号决定；从锁存器的输入与主锁存器的输出连接，其状态由主锁存器的状态决定。主从锁存器受互补时钟控制，触发器的工作分两步进行。

当 CP=1 时，这时 G7 和 G8 的门打开，主锁存器工作，主锁存器接收 R、S 信号。如果 R=0，S=1，由逻辑门控 RS 锁存器的功能表 4-3 可知，$Q'=1$，$\overline{Q}'=0$，相当于把 R 和 S 信号暂时存在了主锁存器的输出端。这时从锁存器被封锁，保持原有状态。

当 CP 由 1 变 0 时刻（CP 的下降沿），从锁存器接收主锁存器的状态，从锁存器翻转到与主锁存器相同的状态；而 CP=0 后，主锁存器封锁，不再接收信号，因为主锁存器的状态不会再变化，所以从锁存器的状态也不会变化。此时 $Q'=1$，$\overline{Q}'=0$，则从锁存器的输出也翻转到$Q=1$，$\overline{Q}=0$。主从 RS 触发器的状态变化是在 CP 由 1 变 0 时刻，即 CP 的下降沿

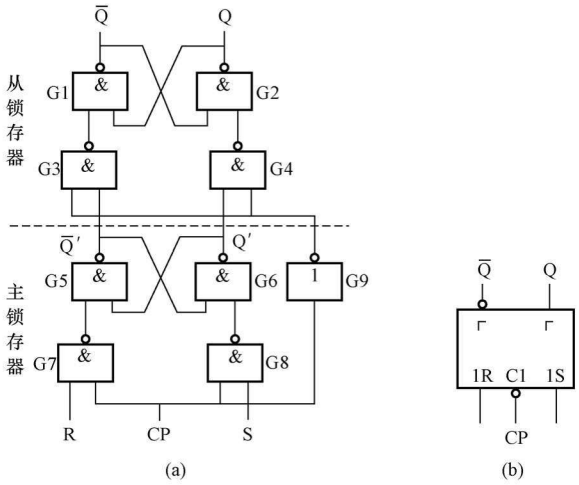


图 4-11　由两个逻辑门控 RS 锁存器组成的主从 RS 触发器

（a）逻辑电路图；（b）逻辑符号

进行的。

主从 RS 触发器的特性方程与逻辑门控 RS 锁存器相同，其真值表、状态转换图、驱动表也相同。

2. 主从 RS 触发器的基本特点

主从 RS 触发器由两个逻辑门控 RS 锁存器组成，它们受互补时钟控制，只在下降沿触发翻转，时钟一旦变成低电平，主触发器被封锁，不再受 R 和 S 信号的影响，对输入信号的敏感时间大大缩短，因此克服了空翻现象。

另外，由于主从 RS 触发器是由两个逻辑门控 RS 锁存器组成的，所以仍然具有约束条件，即 RS=0（即两个输入信号不能同时有效）。

【例 4-4】 已知主从 RS 触发器输入信号 S、R 的波形如图 4-12 所示。分析触发器输出 Q 的波形（设触发器的初始状态为 0）。

解　根据特性方程 $Q^{n+1}=S+\overline{R}Q^n$，或特性表，或状态转换图，可画出输出 Q 端的波形如图 4-12 所示。

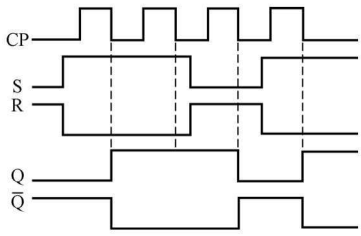


图 4-12　［例 4-4］的波形图

波形分析的时候需要注意：触发器的触发翻转发生在时钟的边沿（这里是下降沿），其他时间保持；在 CP=1 期间输入信号无变化，判断主从触发器次态的依据是时钟脉冲下降沿前瞬间输入端的状态；如果在 CP=1 期间输入信号的状态发生了改变，必须考虑主触发器的全部工作过程，根据主触发器的状态分析从触发器的输出状态。

（二）主从 JK 触发器

主从 RS 触发器的输入信号存在约束条件。而实际应用中，需要触发器的每一种输入组合都是有意义的。为此，在主从 RS 触发器的基础上，稍加改动而产生了主从 JK 触发器。

主从 JK 触发器的逻辑电路图及逻辑符号分别如图 4-13（a）、（b）所示。它是在主从 RS 触发器的基础上引入两根反馈线，一根从 Q 引回到 G7 输入端，另一根从$\overline{Q}$引回到 G8 的

输入端。主从 RS 触发器的置位输入信号 S 改称为 J，主从 RS 触发器的复位输入信号 R 改称为 K，即 J 为主从 JK 触发器的置位输入信号，K 为主从 JK 触发器的复位输入信号，输入信号为高电平有效。

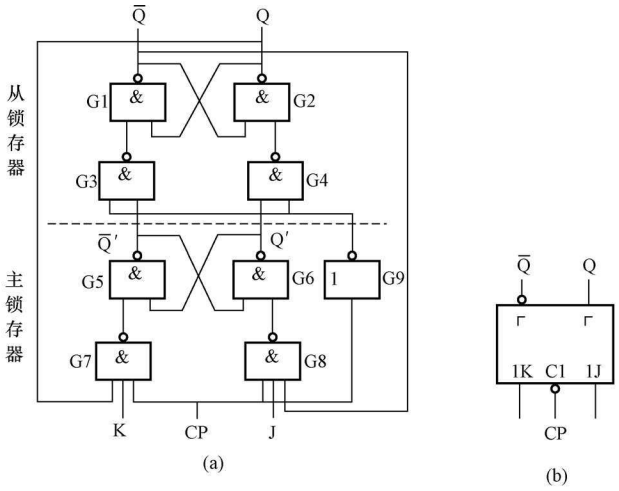


图 4-13 主从 JK 触发器

(a) 逻辑电路图；(b) 逻辑符号

1. 主从 JK 触发器的逻辑功能分析

主从 JK 触发器的工作原理与主从 RS 触发器十分类似。主从 JK 触发器的主锁存器是在时钟 CP=1 期间接收数据，在 CP 下降沿时主锁存器的状态转移到从锁存器，Q 端才改变状态。因为 Q 引回到 G7 输入端，$\overline{Q}$引回到 G8 的输入端，所以主从 JK 触发器解决了主从 RS 触发器的输入信号有约束的问题，但是又引入了新的问题——一次变化问题。

主从 JK 触发器的工作原理描述如下：

如图 4-13 (a) 所示，当 CP=1 时，这时 G7 和 G8 的门打开，主锁存器工作，接收 J、K 输入信号。如果 K=0、J=1（即置位信号有效），由逻辑门控 RS 锁存器的功能表知，$Q'=1$，$\overline{Q}'=0$，相当于把 J 和 K 输入信号暂时存在了主锁存器的输出端。这时从锁存器被封锁，保持原有状态。

当 CP 由 1 变 0 时刻（CP 的下降沿），从锁存器接收主锁存器的状态，从锁存器翻转到与主锁存器相同的状态；而 CP=0 后，主锁存器封锁，不再接收信号，因为主锁存器的状态不会再变化，所以从锁存器的状态也不会变化。此时 $Q'=1$，$\overline{Q}'=0$，则从锁存器的输出也翻转到 $Q=1$，$\overline{Q}=0$。主从 JK 触发器的状态变化是在 CP 由 1 变 0 时刻，即 CP 的下降沿进行的。

因为 Q 引回到门 G7 输入端，$\overline{Q}$引回到门 G8 的输入端，Q 和$\overline{Q}$总是互补的，所以主从 JK 触发器的主锁存器的门 G7 和门 G8 的输出总有一个是 1，也就是主锁存器中的基本 RS 锁存器的输入端不会同时处于有效的低电平状态，所以解决了主从 RS 触发器的输入信号有约束的问题，但是又引入了新的问题，即一次变化问题。后面会详细介绍一次变化问题。

由上分析，可得到主从 JK 触发器的特性表如表 4-5 所示。

表 4-5　　主从 JK 触发器的特性表

J	K	Q^n	Q^{n+1}	功能说明	J	K	Q^n	Q^{n+1}	功能说明
0	0	0	0	保持原状态	1	0	0	1	置位
0	0	1	1		1	0	1	1	
0	1	0	0	复位	1	1	0	1	与原状态相反
0	1	1	0		1	1	1	0	

主从 JK 触发器与主从 RS 触发器比较，相当于在主从 RS 触发器中的输入端有 $S=J\overline{Q}^n$、$R=KQ^n$，代入主从 RS 触发器的特性方程，可得主从 JK 触发器的特性方程，即 $Q^{n+1}=S+\overline{R}Q^n=J\overline{Q}^n+\overline{KQ^n}Q^n=J\overline{Q^n}+\overline{K}Q^n$（只在 CP 下降沿到来时有效）。

关于主从 JK 触发器的特性方程也可以通过特性表的卡诺图化简得到，结果相同。

主从 JK 触发器的状态转换图如图 4-14 所示。根据特性表可以得到主从 JK 触发器的驱动表，如表 4-6 所示。

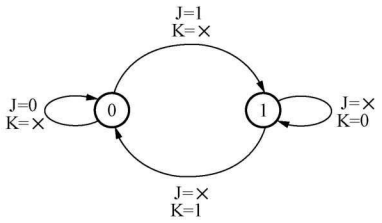


图 4-14　主从 JK 触发器的状态转换图

表 4-6　　主从 JK 触发器的驱动表

$Q^n \to Q^{n+1}$		J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

【例 4-5】　已知主从 JK 触发器的输入信号 J、K 的波形，如图 4-15 所示。分析触发器输出 Q 和 $\overline{Q}$ 的波形（设触发器的初始状态为 0）。

解　从输入信号的波形可知，在 CP=1 期间输入信号没有变化，所以只需在下降沿根据特性方程 $Q^{n+1}=J\overline{Q^n}+\overline{K}Q^n$ 分析 Q 的状态，其他时间 Q 保持。画出输出 Q 端的波形如图 4-15 所示（也可以利用主从 JK 触发器的特性表直接分析）。

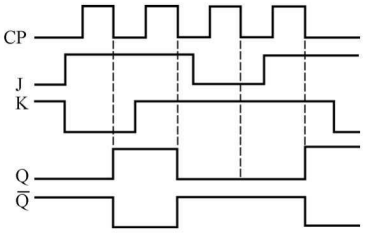


图 4-15　[例 4-5] 的波形图

2. 主从 JK 触发器的脉冲工作特性

触发器的脉冲工作特性是指触发器对时钟脉冲、输入信号和它们之间在时间关系上的相互配合的要求，包括输入信号的建立时间 t_{set}、输入信号的保持时间 t_H、输出信号的传输延迟时间、时钟脉冲 CP 高电平的维持时间 t_{CPH}、时钟脉冲 CP 低电平的维持时间等。掌握这种工作特性对触发器的应用非常重要。

输入信号的建立时间是指以有效的时钟脉冲沿为基准，要求输入信号提前准备好的最小时间量，用 t_{set} 表示。

输入信号的保持时间是指以有效的时钟脉冲沿为基准，要求输入信号在有效的时钟脉冲沿之后必须再维持的最小时间量，用 t_H 表示。

输出信号的传输延迟时间有两个，t_{CPHL} 和 t_{CPLH}。t_{CPLH} 是指以有效的时钟脉冲沿为基准，输出信号由低电平跳变为高电平所经历的时间。t_{CPHL} 是指以有效的时钟脉冲沿为基准，输出

信号由高电平跳变为低电平所经历的时间。

时钟脉冲 CP 高电平的维持时间 t_{CPH} 是指电路能正常工作的时钟脉冲 CP 高电平的最小维持时间。

时钟脉冲 CP 低电平的维持时间 t_{CPL} 是指电路能正常工作的时钟脉冲 CP 低电平的最小维持时间。

主从 JK 触发器的脉冲工作特性如图 4-16 所示。在图 4-13（a）所示的主从 JK 触发器逻辑电路中，当时钟脉冲 CP 上升沿到达时，输入信号 J、K 进入主触发器，由于 J、K 和 CP 同时接到 G7、G8 门的输入端，所以只要 J、K 信号不迟于 CP 上跳沿即可，所以，J、K 输入信号的建立时间 $t_{set}=0$。

在 CP 上跳沿到达后，要经过三级与非门的延迟时间 t_{pd}，主触发器才翻转完毕。所以要求时钟脉冲 CP 高电平的维持时间 $t_{CPH}\geqslant 3t_{pd}$。

等 CP 下跳沿到达后，从触发器翻转，主触发器立即被封锁，所以，输入信号 J、K 可以不再保持，即输入信号的保持时间 $t_H=0$。

从 CP 下跳沿到达到触发器输出状态稳定，也需要一定的传输时间，即 CP=0 的状态也必须保持一段时间，这段时间称为时钟脉冲 CP 低电平的维持时间 t_{CPL}。

由图 4-13 可以看出，该电路的输出信号的传输延迟时间 $t_{CPLH}=2t_{pd}$，输出信号的传输延迟时间 $t_{CPHL}=3t_{pd}$，所以要求触发器的时钟脉冲 CP 低电平的维持时间 $t_{CPL}\geqslant t_{CPHL}=3t_{pd}$。

综上所述，主从 JK 触发器要求 CP 的最小工作周期 $T_{min}=t_{CPH}+t_{CPL}$。

3. 主从 JK 触发器的一次变化问题

图 4-17 给出了主从 JK 触发器的输入、输出波形，包括输入信号 J、K 及主触发器、从触发器的输出波形（已知主触发器和从触发器的初状态为 0）。由图 4-17 可见，在 CP=1 期间，主触发器工作，先有 J=K=0，所以 Q′=0，维持原态；然后当 J 由 0 变为 1 时，G8 输出为 0，G7 输出为 1，主触发器的输出 Q′=1；接着 J 再由 1 变回 0 时，输入信号 J、K 无效，主触发器保持原状态（J=1 时 Q′的状态），Q′仍为 1。当 CP 下降沿到来时，主触发器把 Q′=1 送给从触发器做输入，从触发器翻转为 1。

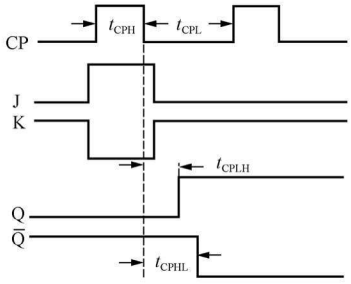


图 4-16 主从 JK 触发器的脉冲工作特性

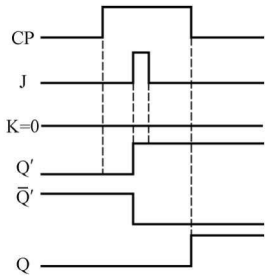


图 4-17 主从 JK 触发器的输入、输出波形

主从 JK 触发器在 CP=1 期间，输入信号 J、K 的变化可能引起主触发器状态的改变，但是主触发器的状态只能变化一次，第二次变化再也不能发生，这种现象称为一次变化问题。

一次变化问题是有害的，如果在 CP=1 期间，输入端出现干扰信号引起 J、K 的变化，从而引起主触发器状态的改变，当干扰信号从 J、K 输入端消失后，J、K 恢复正常，但是

主触发器的状态再也不能根据正常的 J、K 输入而改变其状态。为了避免发生一次变化问题，在使用主从 JK 触发器时，要保证在 CP=1 期间，输入信号 J、K 保持不变。

要解决主从 JK 触发器的一次变化问题，仍应从电路结构上入手，让触发器只接收 CP 触发沿到来前一瞬间的输入信号，这种触发器称为边沿触发器。

（三）T 触发器和 T′触发器

如果将 JK 触发器的 J 和 K 相连作为一个输入端，将此输入端用 T 表示，就构成了 T 功能触发器，其逻辑电路图和逻辑符号如图 4-18 所示。T 触发器的特性表，如表 4-7 所示。

J=K=T，代入 JK 触发器的特性方程，得到 T 触发器的特性方程为

$$Q^{n+1} = T\overline{Q^n} + \overline{T}Q^n = T \oplus Q^n$$

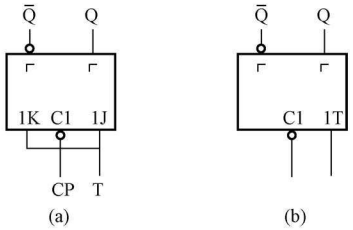


图 4-18 用 JK 触发器构成的 T 触发器

（a）逻辑电路图；（b）逻辑符号

表 4-7 T 触发器的特性表

T	Q^n	Q^{n+1}	功 能 说 明
0	0	0	保持原状态
0	1	1	
1	0	1	每输入一个脉冲
1	1	0	输出状态改变一次

T 触发器的状态转换图，如图 4-19 所示。T 触发器的驱动表，如表 4-8 所示。

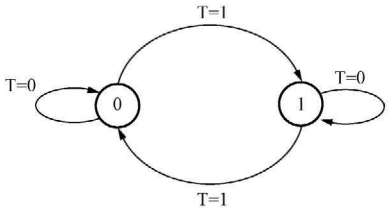


图 4-19 T 触发器的状态转换图

表 4-8 T 触发器的驱动表

$Q^n \rightarrow Q^{n+1}$		T
0	0	0
0	1	1
1	0	1
1	1	0

当 J=K=T=1 时，触发器的特性方程为

$$Q^{n+1} = \overline{Q^n}$$

触发器每输入一个时钟脉冲 CP，状态将翻转一次。这种状态称为计数状态，这种触发器称为 T′触发器，也被称作计数型触发器。

（四）CMOS 主从结构 D 触发器

1. CMOS 主从结构 D 触发器的电路结构

图 4-20 所示是用 CMOS 逻辑门和 CMOS 传输门组成的主从 D 触发器。D 触发器只有一个输入端 D，它由两部分组成，虚线的左边，G1、G2 和 TG1、TG2 组成主锁存器，虚线的右边，G3、G4 和 TG3、TG4 组成从锁存器。CP 和$\overline{CP}$为互补的时钟脉冲。由于引入了传输门，该电路虽为主从结构，却没有一次变化问题，与前面讲的主从结构触发器的工作特点不同。

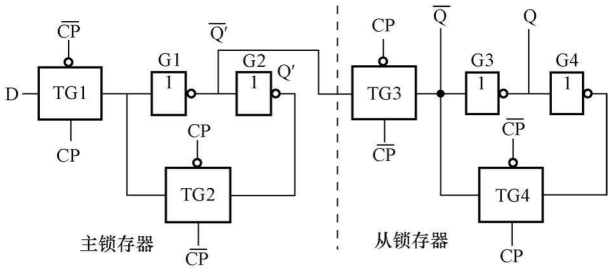


图 4-20 CMOS 主从结构 D 触发器

2. CMOS 主从结构 D 触发器的功能分析

触发器的工作过程分为两个节拍：

(1) 当 CP 正跳变后，CP 为 1，则$\overline{CP}$为 0。这时 TG1 开通，TG2 关闭。主锁存器接收输入端 D 的信号。设 D=1，经 TG1 传到 G1 的输入端，使$\overline{Q'}=0$，$Q'=1$。同时，TG3 关闭，切断了主、从两个锁存器间的联系，TG4 开通，从锁存器保持原状态不变。

(2) 当 CP 负跳变后，CP 为 0，则$\overline{CP}$变为 1。这时 TG1 关闭，切断了 D 输入信号与主锁存器的联系，使 D 信号不再影响主锁存器的状态，而 TG2 开通，将 G1 的输入端与 G2 的输出端连通，使主锁存器保持原状态不变。与此同时，TG3 开通，TG4 关闭，将主锁存器的状态$\overline{Q'}=0$ 送入从锁存器，使$\overline{Q}=0$，经 G3 反相后，输出 Q=1。至此完成了整个触发器接收信号的全过程。

D 触发器只有一个触发输入端 D，因此，逻辑关系非常简单，其功能表如表 4-9 所示。

表 4-9　　D 触发器的功能表

D	Q^n	Q^{n+1}	功能说明
0	0	0	输出状态与 D 状态相同
0	1	0	
1	0	1	
1	1	1	

D 触发器的特性方程为

$$Q^{n+1}=D$$

D 触发器的状态转换图如图 4-21 所示，驱动表如表 4-10 所示。

如果将传输门的控制信号 CP 和$\overline{CP}$互换，可使触发器变为 CP 上跳沿触发。

为了实现异步置位、复位功能，需要引入 R_D 和 S_D 信号，因为 R_D 和 S_D 是以高电平作为置 1 和置 0 信号的，所以需要把图 4-20 中的四个反相器改成或非门，形成如图 4-22 所示的电路。注意，该电路的 R_D 和 S_D 端都为高电平有效。此处“异步”的含义是直接实现置位、复位功能，不需要有效的时钟沿配合。

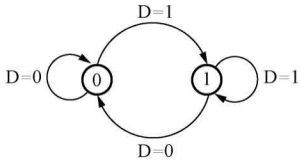


图 4-21 D 触发器的状态转换图

表 4-10　　D 触发器的驱动表

$Q^n \to Q^{n+1}$		D
0	0	0
0	1	1
1	0	0
1	1	1

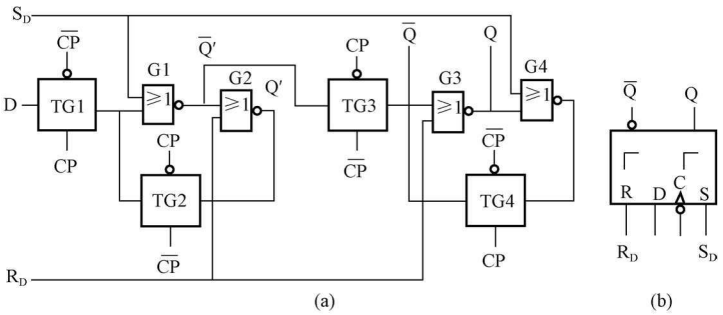


图 4-22　带有 R_D 和 S_D 端的 CMOS 边沿触发器

（a）逻辑图；（b）逻辑符号

3. D 触发器的基本特点

D 触发器在 CP 控制下分两个节拍动作，这一点与前面讲的主从结构的触发器相同。

由前面的分析可知，D 触发器是在利用 4 个传输门交替地开通和关闭将触发器的触发翻转控制在 CP 下降沿到来的一瞬间，并接收 CP 下降沿到来前一瞬间的 D 信号。因此，在 CP 的一个周期内，触发器只在 CP 的触发沿时刻接收信号并使输出状态翻转。

总之，CMOS 主从结构的 D 边沿触发器具有主从触发器的电路结构，边沿触发器的工作特点。

四、边沿触发器

边沿触发器不仅可以将触发器的触发翻转控制在 CP 触发沿到来的前一瞬间，而且可以将接收输入信号的时间也控制在 CP 触发沿到来的前一瞬间，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。因此，边沿触发器既没有空翻现象，也没有一次变化现象，提高了触发器工作的可靠性和抗干扰能力。

目前已应用于数字集成电路产品中的边沿触发器有维持阻塞触发器、利用传输延迟的边沿触发器等。

（一）维持阻塞 D 触发器

1. 电路结构

维持阻塞 D 触发器的逻辑电路图及逻辑符号如图 4-23（a）、（b）所示。该触发器由 6 个与非门和四根反馈线 L1、L2、L3 和 L4 构成，其中 G1 和 G2 构成基本 RS 锁存器。一个输入信号 D，时钟 CP 上升沿触发。

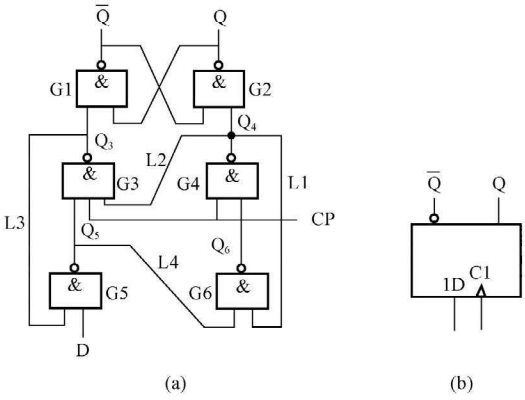


图 4-23　维持阻塞 D 触发器

（a）逻辑电路图；（b）逻辑符号

2. 工作原理

输入信号 D=1 的情况。在 CP=0 时，G3、G4 被封锁，$Q_3=1$、$Q_4=1$，G1、G2 组成的基本 RS 锁存器保持原状态不变。因为 D=1，G5 输入全 1，输出 $Q_5=0$，它使 $Q_3=1$，$Q_6=1$。当 CP 由 0 变 1 时，G4 输入全 1，输出 Q_4 变为 0。继而，Q 翻转为 1，$\overline{Q}$翻转为 0，

完成了使锁存器翻转为1状态的全过程。同时，一旦 Q_4 变为0，通过反馈线L1封锁了G6门，这时如果D信号由1变为0，只会影响G5的输出，不会影响G6的输出，维持了锁存器的1状态。因此，称L1线为置1维持线。同理，Q4变0后，通过反馈线L2也封锁了G3门，从而阻塞了置0通路，故称L2线为置0阻塞线。

输入信号D=0的情况。在CP=0时，G3、G4被封锁，$Q_3=1$、$Q_4=1$，G1、G2组成的基本RS锁存器保持原状态不变。因D=0，$Q_5=1$，G6输入全1，输出 $Q_6=0$。当CP由0变1时，G3输入全1，输出 Q_3 变为0。继而，$\overline{Q}$翻转为1，Q翻转为0，完成了使触发器翻转为0状态的全过程。同时，一旦 Q_3 变为0，通过反馈线L3封锁了G5门，这时无论D信号再怎么变化，也不会影响G5的输出，从而维持了触发器的0状态。因此，称L3线为置0维持线。同理，Q_5 的1信号通过反馈线L4、Q_4 的1信号通过反馈线L1，送到门G6的输入端，使 Q_6 保持0封锁了G4门，从而阻塞了置1通路，故称L4线为置1阻塞线。

通过分析，可以得到如表4-9所示的特性表。维持阻塞D触发器的特性方程、状态转换图以及状态驱动表与CMOS主从结构的D触发器相同，这里不再重述。

3. 脉冲工作特性

维持阻塞D触发器的脉冲工作特性如图4-24所示。在CP上跳沿到来时，G3、G4门将根据G5、G6门的输出状态控制触发器翻转。因此在CP上跳沿到达之前，G5、G6门必须要有稳定的输出状态。而从信号加到D端开始到G5、G6门的输出稳定下来，需要经过一段时间，这段时间称为触发器的建立时间 t_{set}，即输入信号必须比CP脉冲早 t_{set} 时间到达。由图4-23（a）可以看出，该电路的建立时间为两级与非门的延迟时间，即 $t_{set}=2t_{pd}$。

其次，为使触发器可靠翻转，信号D还必须维持一段时间，在CP触发沿到来后输入信号需要维持的时间称为触发器的保持时间 t_H。当D=0时，这个0信号必须维持到 Q_3 由1变0后将G5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。所以D=0时的保持时间 $t_H=1t_{pd}$。当D=1时，CP上跳沿到达后，经过 t_{pd} 的时间 Q_4 变0，将G6封锁。但若D信号变化，传到G6的输入端也同样需要 t_{pd} 的时间，所以D=1时的保持时间 $t_H=0$。综合以上两种情况，取 $t_H=1t_{pd}$。

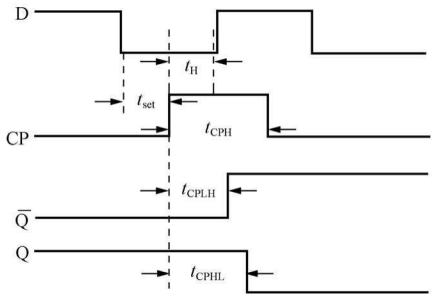


图4-24 维持阻塞D触发器的脉冲工作特性

另外，为保证触发器可靠翻转，CP=1的状态也必须保持一段时间，直到触发器的Q、$\overline{Q}$端电平稳定，这段时间称为触发器的维持时间 t_{CPH}。时钟脉冲触发沿开始到一个输出端由0变1所需的时间称为 t_{CPLH}；时钟脉冲触发沿开始到另一个输出端由1变0所需的时间称为 t_{CPHL}。由图4-23（a）可以看出，该电路的 $t_{CPLH}=2t_{pd}$，$t_{CPHL}=3t_{pd}$，所以触发器的 $t_{CPH}\geqslant t_{CPHL}=3t_{pd}$。

4. 基本特点

通过逻辑电路的功能分析可知，维持阻塞D触发器是利用了维持线和阻塞线，将触发器的触发翻转控制在CP上跳沿到来的一瞬间。触发器的次态只取决于CP上跳沿到来前一

瞬间的 D 信号，而在这之前或之后，输入信号 D 的变化对触发器的输出状态没有影响。这种特点有效地提高了触发器电路的抗干扰能力，因而也提高了电路的工作可靠性，维持阻塞 D 触发器也因此而得名。图 4-23（b）中，“∧”表示边沿触发，外侧无圆圈，表示上升沿触发。

【例 4-6】 维持阻塞 D 触发器如图 4-23（a）所示，设初始状态为 0，已知输入 D 的波形图如图 4-25 所示，试画出输出 Q 的波形图。

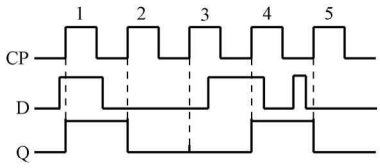


图 4-25 ［例 4-6］波形图

解 由于该触发器是边沿触发器，在分析波形图时，应注意：触发器的触发翻转发生在时钟脉冲的触发沿（这里是上升沿）；判断触发器次态的依据是时钟脉冲触发沿前一瞬间（这里是上升沿前一瞬间）输入端的状态。

根据 D 触发器的功能表或特性方程或状态转换图，可画出输出端 Q 的波形图如图 4-25 所示。

5. 集成高速 CMOS 边沿 D 触发器 74HC74

实际的集成触发器一般都设有直接清零和置 1 端。具有异步置位和复位的维持阻塞 D 触发器，其逻辑电路和逻辑符号如图 4-26 所示。图中直接置 0 端为 R_D，直接置 1 端 S_D，该电路 R_D 和 S_D 端都为低电平有效。R_D 和 S_D 信号不受时钟信号 CP 的制约，具有最高的优先级，所以又称异步清零端、异步置 1 端。

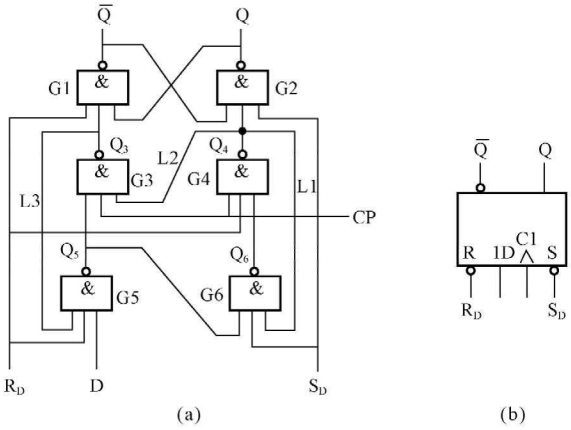


图 4-26 具有异步置位和复位的维持阻塞 D 触发器

（a）逻辑电路图；（b）逻辑符号

R_D 和 S_D 的作用主要是用来给触发器设置初始状态，或对触发器的状态进行特殊的控制。在使用时要注意，任何时刻只能一个信号有效，不能同时有效。

集成 D 触发器的定型产品种类很多，这里将介绍双 D 触发器 74HC74。一个 74HC74 片子里封装着两个相同的 D 触发器，每个触发器只有一个 D 端，它们都带有直接置 0 端 R_D 和直接置 1 端 S_D，为低电平有效，CP 上升沿触发。74HC74 的逻辑符号和引脚排列分别如图 4-27（a）、（b）所示，其功能表如表 4-11 所示。

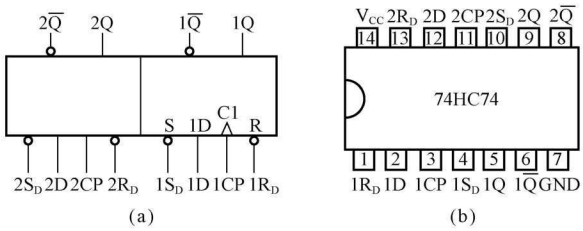


图 4-27 高速 CMOS 边沿 D 触发器 74HC74

(a) 逻辑符号；(b) 引脚排列图

表 4-11 74HC74 的功能表

输入				输出	
R_D	S_D	CP	D	Q	$\overline{Q}$
0	1	×	×	0	1
1	0	×	×	1	0
1	1	↑	0	0	1
1	1	↑	1	1	0

（二）利用传输延迟的边沿 JK 触发器

1. 电路结构

另一种边沿触发器是利用门电路的传输延迟时间而构成的，CPF 降沿触发的边沿 JK 触发器逻辑电路和逻辑符号如图 4-28 所示。图 4-28（a）中电路包含一个由 G11、G12、G13 和 G21、G22、G23 组成的基本 RS 锁存器，两个输入控制门 G3、G4，而且门的传输延迟时间大于基本 RS 锁存器的翻转时间。图 4-28（b）中的小圆圈和“∧”结合表示在时钟的下降沿触发器被触发。

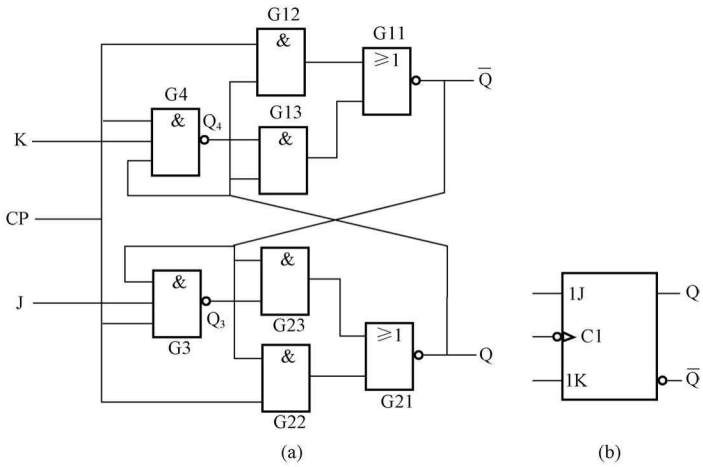


图 4-28 利用传输延迟的边沿 JK 触发器

(a) 逻辑图；(b) 逻辑符号

2. 工作原理

如图 4-28（a）所示，CP＝0 时，触发器处于一个稳态，门 G3、G4 被封锁，不接收 J、K 信号，$Q_3=Q_4=1$。另一方面，G12、G22 也被 CP 封锁，G11、G13 和 G21、G23 构成的基本 RS 锁存器处于一个稳定状态，使 Q，$\overline{Q}$保持不变。

CP 由 0 变 1 时，触发器不翻转，为接收 J、K 输入信号做准备。CP 信号可以通过两个信号通道影响触发器的输出状态：一条是通过与门 G12 和 G22 直接影响触发器的输出；另一条通道是通过与非门 G3 和 G4，再经与门 G23 和 G13 影响触发器的输出状态。前一个信号通道只需经一级与门的传输延迟时间，而后一个通道则需要一级与非门和一级与门的传输延迟时间，显然前一通道影响输出状态比后者要快得多。由于时间上的差别，所以触发器的

状态由前一通道的信号决定。设触发器的初始状态为 Q=0，$\overline{Q}$=1。CP 由 0 变 1 时，首先 G22 的输出由 0 变 1，无论 G23 的输出为何种状态（无论 J 和 K 为何种状态），对应的或非门 G21 的输出 Q 仍然为 0；无论 Q_4 是何状态，Q=0 传递给 G12、G13，使它们输出均为 0，对应的 G11 的门输出$\overline{Q}$为 1。同理可以分析，当触发器的原状态为 Q=1，$\overline{Q}$=0 时，会有 G12 输出为 1，保证$\overline{Q}$=0，G22 和 G23 输出为 0，从而 Q 仍然为 1。总之，CP 由 0 变 1 时，J 和 K 不影响触发器的状态，触发器不发生翻转，但是 G3 和 G4 门打开，准备接收 J 和 K 信号。

CP 下降沿到达时，触发器的状态取决于前一时刻 J 和 K 的状态。设输入信号 J=1、K=0，触发器的原状态 Q=0，$\overline{Q}$=1，则有 Q_3=0，Q_4=1，CP 由 1 变 0 时，G12、G22 首先被封锁，输出为 0，G3 和 G4 也被封锁。但是 Q_3、Q_4 经 G23、G13 影响触发器的输出，有 G23 的输出为 0，又因为 G22 的输出也为 0，所以 Q=1，使 G13 输出为 1，$\overline{Q}$=0，触发器翻转。虽然 CP 变 0 后，G3、G4、G12 和 G22 被 CP 封锁，Q_3=Q_4=1。但由于与非门的延迟时间比与门长（由制造工艺予以保证），因此 Q_3=Q_4=1 这一新状态的稳定是在触发器翻转之后。此时因 CP=0，所以不管 J、K 状态如何变化，触发器必保持在翻转后的 1 状态。在对 J 和 K 为不同取值时触发器的工作状态逐一分析后，可得到表 4-5 所示的特性表。

利用传输延迟的边沿 JK 触发器基本特点是在 CP 下降沿前接收 J、K 信号，在下降沿触发翻转，在下降沿后输入被封锁，它消除了一次变化现象。

3. 脉冲工作特性

电路要求 J、K 输入信号先于 CP 下跳沿传输到 G3、G4 的输出端，因此要求 J、K 输入信号至少比 CP 下降沿提前一级与非门的延迟时间，这段时间是 J、K 输入信号的建立时间 t_{set}。图 4-29 所示为边沿 JK 触发器的脉冲工作特性。

J、K 输入信号在 CP 下降沿到来后就不必保持，t_H=0。原因在于即使原来的 J、K 输入信号变化，还要经一级与非门的延迟时间才能传输到 G3、G4 的输出端，在此之前，触发器已由 G12、G13、G22 和 G23 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 J、K 输入信号的保持时间极短，从而具有很高的抗干扰能力，且因缩短 t_{CPL} 而提高工作速度。

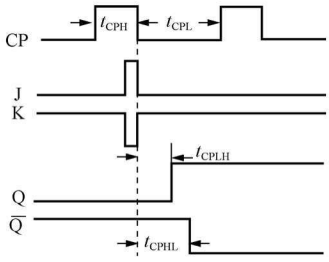


图 4-29　边沿 JK 触发器的脉冲工作特性

从 CP 下降沿开始到触发器输出状态稳定，也需要一定的延迟时间，这个时间应大于两级与或非门的延迟时间。

4. 集成边沿 JK 触发器 74HC76

集成边沿 JK 触发器 74HC76，该器件内含两个相同的 JK 触发器，都带有预置和清零输入端，属于下降沿触发的边沿触发器。74HC76 的逻辑符号如图 4-30（a）、引脚图如图 4-30（b）所示，功能表如表 4-12 所示。

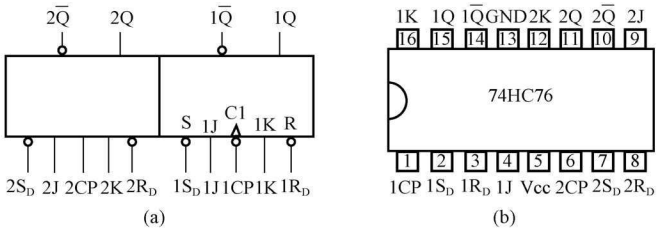


图 4-30 74HC76 逻辑图及引脚图

(a) 逻辑符号；(b) 引脚图

表 4-12 74HC76 的功能表

输入					输出		输入					输出	
R_D	S_D	CP	J	K	Q	$\overline{Q}$	R_D	S_D	CP	J	K	Q	$\overline{Q}$
0	1	×	×	×	0	1	1	1	↓	0	1	0	1
1	0	×	×	×	1	0	1	1	↓	1	0	1	0
1	1	↓	0	0	保持		1	1	↓	1	1	翻转	

第二节 触发器的逻辑功能及功能转换

一、触发器按逻辑功能的分类

从上一节可以看到，触发器电路的信号输入方式不同，触发器的状态随输入信号的变化规律不同，所以逻辑功能也不完全相同。

需要指出的是触发器的逻辑功能和电路结构是两个不同的概念。所谓逻辑功能，是指触发器的次态和现态及输入信号之间在稳态下的逻辑关系，这种逻辑关系可以用特性表、特性方程和状态转换图来表达。按照逻辑功能的不同特点，通常将时钟控制的触发器分为 RS 触发器、JK 触发器、D 触发器、T 触发器等几种类型。各种触发器的功能描述如表 4-13 所示。而基本 RS、同步 RS 触发器、主从触发器、边沿触发器等是指电路结构的不同形式，由于电路的结构不同，带来了各自相异的动作特点。

同一种逻辑功能的触发器可以用不同的电路结构实现，反过来说，用同一种电路结构形式可以构成不同逻辑功能的触发器，因此逻辑功能与电路结构无固定的对应关系。

触发器每输入一个时钟脉冲 CP，状态将翻转一次，这种状态称为计数状态，这种状态的触发器称为 T′触发器，其特性方程为

$$Q^{n+1} = \overline{Q^n}$$

读者可以自行分析得到其特性表、状态转换图、驱动表。

二、触发器功能的转换

由于输入信号为双端的情况下 JK 触发器的逻辑功能最完善，而输入信号为单端的情况下 D 触发器用起来最方便，所以目前的集成触发器以 JK 触发器和 D 触发器最常见。在必须使用其他逻辑功能的触发器时，可以通过逻辑功能转换的方法，将 JK 触发器或 D 触发器转换为所需要的逻辑功能的触发器，当然逻辑功能的转换可以用于任何两种逻辑功能触发器之间的相互转换。

表 4 - 13 **各种功能触发器的功能描述**

触发器	RS 触发器	JK 触发器	D 触发器	T 触发器
特性方程	$Q^{n+1}=S+\overline{R}Q^n$ RS=0(约束条件)	$Q^{n+1}=J\overline{Q^n}+\overline{K}Q^n$	$Q^{n+1}=D$	$Q^{n+1}=T\overline{Q^n}+\overline{T}Q^n$
特性表				

RS 触发器特性表：

R	S	Q^n	Q^{n+1}	功能说明
0	0	0	0	保持原状态
0	0	1	1	
0	1	0	1	置 1(置位)
0	1	1	1	
1	0	0	0	置 0(复位)
1	0	1	0	
1	1	0	×	不定(不允许)
1	1	1	×	

JK 触发器特性表：

J	K	Q^n	Q^{n+1}	功能说明
0	0	0	0	保持原状
0	0	1	1	
0	1	0	0	置 0(复位)
0	1	1	0	
1	0	0	1	置 1(置位)
1	0	1	1	
1	1	0	1	与原状态相反
1	1	1	0	

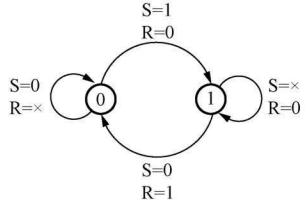
D 触发器特性表：

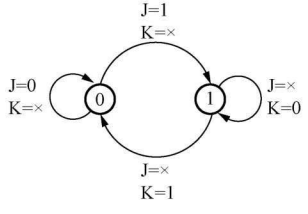
D	Q^n	Q^{n+1}	功能说明
0	0	0	输出状态与 D 状态相同
0	1	0	
1	0	1	
1	1	1	

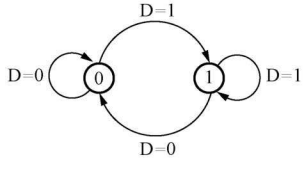
T 触发器特性表：

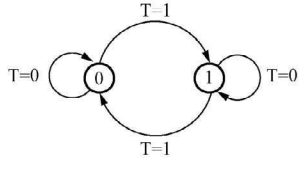
T	Q^n	Q^{n+1}	功能说明
0	0	0	保持原状态
0	1	1	
1	0	1	每输入一个脉冲输出状态改变一次
1	1	0	

状态转换图：









驱动表：

$Q^n\to Q^{n+1}$		R	S
0	0	×	0
0	1	0	1
1	0	1	0
1	1	0	×

$Q^n\to Q^{n+1}$		J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

$Q^n\to Q^{n+1}$		D
0	0	0
0	1	1
1	0	0
1	1	1

$Q^n\to Q^{n+1}$		T
0	0	0
0	1	1
1	0	1
1	1	0

（一）用 JK 触发器转换成其他功能的触发器

1. JK 触发器转换成 D 触发器

已知的是 JK 触发器，待求的是 D 触发器。

首先写出已知触发器的特性方程，即 JK 触发器的特性方程为

$$Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n \tag{4-1}$$

再写出要求触发器的特性方程，并变换为已知触发器的特性方程的形式。D 触发器的特性方程为

$$Q^{n+1} = D = D(\overline{Q^n} + Q^n) = D\overline{Q^n} + DQ^n \tag{4-2}$$

比较式（4-1）和式（4-2）两个特性方程，求出已知触发器的输入信号的方程为

$$J = D, K = \overline{D}$$

画出用 JK 触发器转换成 D 触发器的逻辑电路图如图 4-31（a）所示。

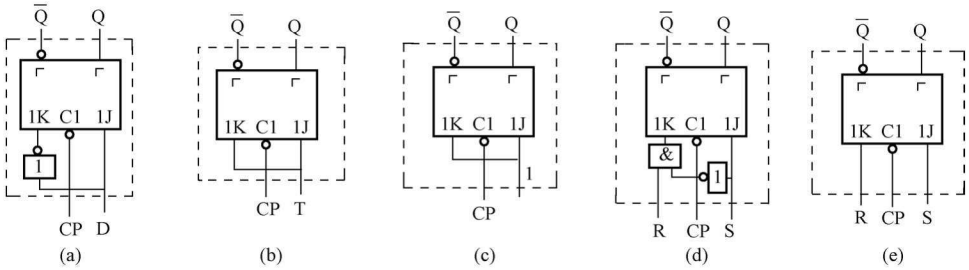


图 4-31 JK 触发器转换成其他功能触发器的逻辑电路图

（a）用 JK 触发器转换成 D 触发器；（b）用 JK 触发器转换成 T 触发器；

（c）用 JK 触发器转换成 T′触发器；（d）用 JK 触发器转换成 RS 触发器；

（e）简化后的用 JK 触发器转换成 RS 触发器

2. JK 触发器转换成 T（T′）触发器

同样可以按照上述方法进行转换。

写出 T 触发器的特性方程，即

$$Q^{n+1} = T\overline{Q^n} + \overline{T}Q^n$$

与 JK 触发器的特性方程比较得

$$J = T, K = T$$

画出用 JK 触发器转换成 T 触发器的逻辑电路图，如图 4-31（b）所示。令 T=1，即可得 T′触发器，如图 4-31（c）所示。

3. JK 触发器转换成 RS 触发器

首先写出已知触发器的特性方程，即 JK 触发器的特性方程为

$$Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n \tag{4-3}$$

再写出所求触发器的特性方程，并变换为已知触发器的特性方程的形式。即 RS 触发器的特性方程如式为

$$Q^{n+1} = S + \overline{R}Q^n = S(\overline{Q^n} + Q^n) + \overline{R}Q^n = S\overline{Q^n} + SQ^n + \overline{R}Q^n \tag{4-4}$$

比较式（4-3）、式（4-4）得

$$J = S, K = \overline{S}R$$

即可实现 RS 触发器的功能，接线图如图 4-31（d）所示。

利用约束条件还能对式（4-4）进一步化简，可得

$$Q^{n+1}=S\overline{Q^n}+SQ^n(R+\overline{R})+\overline{R}Q^n=S\overline{Q^n}+RSQ^n+\overline{R}SQ^n+\overline{R}Q^n$$

因为 RS=0，则有

$$Q^{n+1}=S\overline{Q^n}+\overline{R}SQ^n+\overline{R}Q^n=S\overline{Q^n}+(\overline{R}S+\overline{R})Q^n=S\overline{Q^n}+\overline{R}Q^n \tag{4-5}$$

比较式（4-3）、式（4-5）得

$$J=S,K=R$$

画出简化后的用 JK 触发器转换成 RS 触发器的逻辑电路图如图 4-31（e）所示。

（二）用 D 触发器转换成其他功能的触发器

1. D 触发器转换成 JK 触发器

写出 D 触发器和 JK 触发器的特性方程分别为

$$Q^{n+1}=D \tag{4-6}$$

$$Q^{n+1}=J\overline{Q^n}+\overline{K}Q^n \tag{4-7}$$

比较式（4-6）和式（4-7），得到已知触发器输入信号的表达式为

$$D=J\overline{Q^n}+\overline{K}Q^n$$

画出用 D 触发器转换成 JK 触发器的逻辑电路图如图 4-32（a）所示。

2. D 触发器转换成 T 触发器

写出 D 触发器和 T 触发器的特性方程，即

$$Q^{n+1}=D$$

$$Q^{n+1}=T\overline{Q^n}+\overline{T}Q^n$$

比较上两式，得到已知触发器的输入信号 D 的表达式为

$$D=T\overline{Q^n}+\overline{T}Q^n=T\oplus Q^n$$

画出用 D 触发器转换成 T 触发器的逻辑电路图如图 4-32（b）所示。

3. D 触发器转换成 T′触发器

写出 D 触发器和 T′触发器的特性方程，即

$$Q^{n+1}=D$$

$$Q^{n+1}=\overline{Q^n}$$

比较上面两式，得到已知触发器的输入信号 D 的表达式为

$$D=\overline{Q^n}$$

画出用 D 触发器转换成 T′触发器的逻辑电路图如图 4-32（c）所示。

4. D 触发器转换成 RS 触发器

写出 D 触发器和 RS 触发器的特性方程，即

$$Q^{n+1}=D$$

$$Q^{n+1}=S+\overline{R}Q^n$$

比较上两式，得到已知触发器的输入信号 D 的表达式为

$$D=S+\overline{R}Q^n$$

画出用 D 触发器转换成 RS 触发器的逻辑电路图如图 4-32（d）所示。

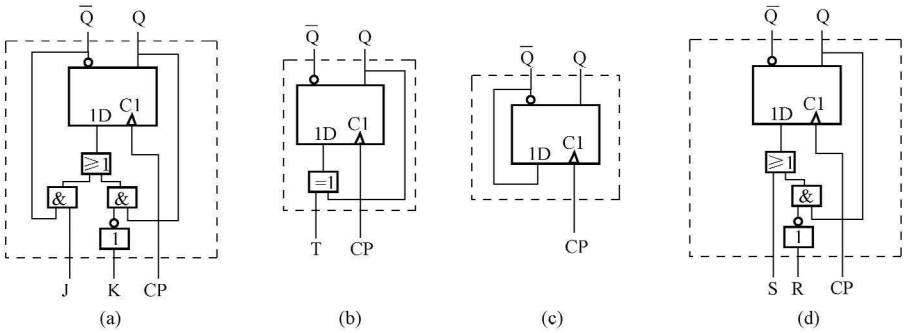


图 4-32 D 触发器转换成功能的触发器逻辑电路图

(a) 用 D 触发器转换成 JK 触发器；(b) 用 D 触发器转换成 T 触发器；

(c) 用 D 触发器转换成 T′触发器；(d) 用 D 触发器转换成 RS 触发器

第三节 集成触发器的主要参数

学习集成触发器的参数，便于更好地使用触发器，下面从两个方面介绍触发器参数的含义。

一、直流参数

因为触发器实质也是由门电路构成，所以触发器的参数与门电路的参数雷同。下面介绍几个常见参数。

(1) 电源电流 I_{CC}：由于一个触发器由许多门构成，无论 0 态还是 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不大。规定所有输入和输出端悬空时电源向触发器提供的电流为电源电流。

(2) 低电平输入电流 I_{IL}：指某输入端接地，其他各输入、输出端悬空时，从该输入端流向地的电流，它表明对驱动电路输出低电平时的加载情况。触发器的每个输入端都有低电平输入电流，比如 JK 触发器的该项参数包括 J、K 端，时钟端和直接置 0、置 1 端的低电平输入电流。

(3) 高电平输入电流 I_{IH}：指将某输入端接电源，其他输入端、输出端悬空时测得的电流就是 I_{IH}，它表明对驱动电路输出高电平时的加载情况。

(4) 输出高电平 U_{OH} 和输出低电平 U_{OL}：触发器输出端输出高电平时的对地电压值为 U_{OH}，输出低电平时的对地电压值为 U_{OL}。

二、开关参数

(1) 最高时钟频率 f_{max}：指触发器在计数状态下能正常工作的最高频率，它是表明触发器工作速度的一个指标。在测试该参数时，输出端带上额定的电流负载和电容负载，在生产厂家的产品手册中均有明确的规定。

(2) 时钟信号的延时时间（t_{CPLH}，t_{CPHL}）：从时钟脉冲的触发沿到触发器的输出端由 0 变为 1 的延迟时间为 t_{CPLH}；从时钟脉冲的触发沿到触发器的输出端由 1 变为 0 的延迟时间为 t_{CPHL}。一般 t_{CPLH} 比 t_{CPHL} 约大一级门的延迟时间。时钟信号的延时时间表明对时钟脉冲 CP 的要求。

(3) 对直接置 0 或置 1 端的延迟时间（t_{RHL}，t_{RLH}，t_{SLH}，t_{SHL}）：从置 0 脉冲触发沿

到输出端由 0 变为 1 的延迟时间为 t_{RLH}，到输出端由 1 变为 0 的延迟时间为 t_{RHL}。从置 1 脉冲触发沿到输出端由 0 变为 1 的延迟时间为 t_{SLH}，到输出端由 1 变为 0 的延迟时间为 t_{SHL}。

触发器的应用非常广泛，是时序逻辑电路重要的组成部分，其典型应用将在第五章中作较详细的介绍。

小 结

锁存器和触发器是时序逻辑电路的基本单元路，有两个基本性质：①在一定条件下，触发器（或锁存器）可维持在两种稳定状态（0 或 1 状态）之一而保持不变；②在一定的外加信号作用下，触发器（或锁存器）可从一个稳定状态转变到另一个稳定状态。这就使得电路能够记忆二进制信息 0 和 1，常被用作二进制存储单元。

按照电路结构形式不同，有基本 RS 锁存器、逻辑门控 RS 锁存器、主从触发器、维持阻塞触发器、利用 CMOS 传输门的边沿触发器和利用传输延迟时间的边沿触发器等不同类型。介绍这些电路结构的主要目的，在于说明由于电路结构不同而带来的不同动作特点。只有了解这些不同的动作特点，才能正确地使用这些触发器。

触发器的逻辑功能是指触发器输出的次态与输出的现态及输入信号之间的逻辑关系。由于输入方式以及触发器状态随输入信号变化的规律不同，各种触发器在具体的逻辑功能上有所差别。根据这些差异，将触发器分成了 RS、JK、T、T′、D 等几种逻辑功能的类型。描写触发器逻辑功能的方法主要有特性表、特性方程、驱动表、状态转换图和波形图（又称时序图）等。

触发器的电路结构形式和逻辑功能是两个不同的概念，两者没有固定的对应关系。同一电路结构的触发器可以做成不同的逻辑功能；同一逻辑功能的触发器可以用不同的电路结构来实现；不同结构的触发器具有不同的触发条件和动作特点，触发器逻辑符号中 CP 端有小圆圈的为下降沿触发；没有小圆圈的为上升沿触发。利用特性方程可实现不同功能触发器间逻辑功能的相互转换。

为了保证触发器在动态时能可靠地翻转，输入信号、时钟信号和它们在时间上相互配合应满足相应集成触发器的开关参数。

习 题

4-1 基本 RS 锁存器和逻辑门控 RS 锁存器的主要区别是什么？

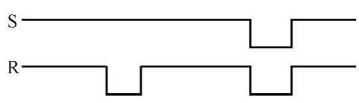


图 4-33 题 4-2 图

4-2 已知与非门组成的基本 RS 锁存器的直接置 0 端和直接置 1 端的输入波形，如图 4-33 所示。试画出锁存器 Q 端和 $\overline{Q}$ 端的波形。

4-3 或非门组成的基本 RS 锁存器如图 4-34 所示，已知输入信号的波形，分析输出波形。

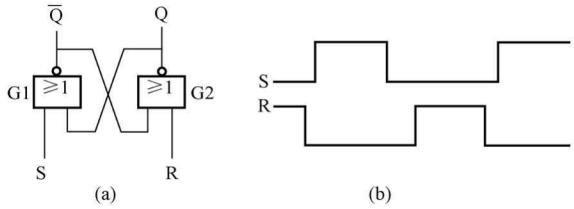


图 4-34　题 4-3 图

4-4　某电路如图 4-35 所示，已知输入信号的波形，分析输出 Q 的波形。设触发器的初始状态为 0。

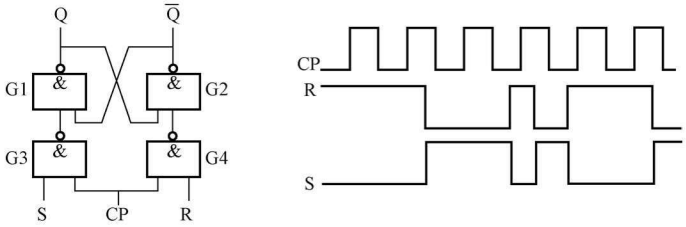


图 4-35　题 4-4 图

4-5　已知主从结构 RS 触发器的输入信号的波形如图 4-36 所示，试分析输出端的波形。设触发器的初始状态为 0。

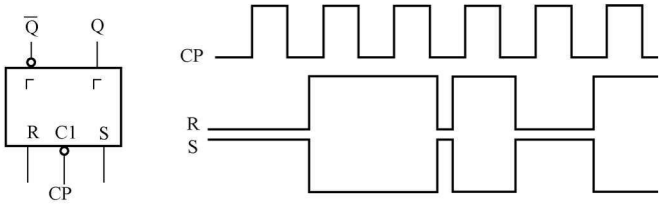


图 4-36　题 4-5 图

4-6　已知主从结构 JK 触发器的输入信号的波形如图 4-37 所示，试分析输出端 Q 的波形。设触发器的初始状态为 0。

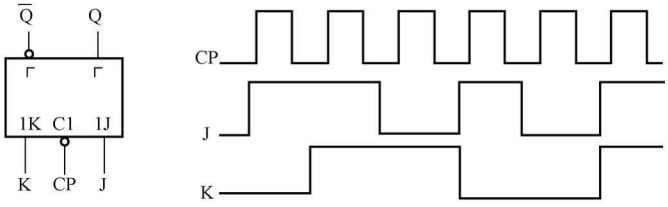


图 4-37　题 4-6 图

4-7　已知主从 JK 触发器的输入信号的波形如图 4-38 所示，设触发器的初始状态为 0，试分析输出端的波形。

4-8　用维持阻塞结构 D 触发器组成的脉冲分频电路如图 4-39 所示，分析 Y 的波形。设触发器的初状态为 0。

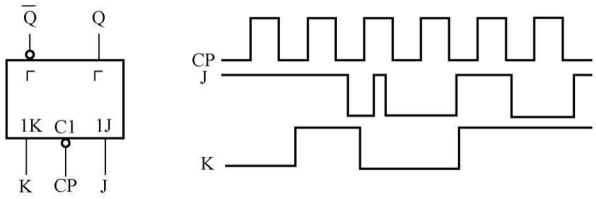


图 4-38　题 4-7 图

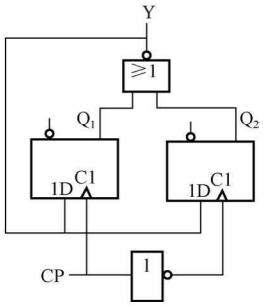


图 4-39　题 4-8 图

4-9　说明基本 RS 锁存器、逻辑门控 RS 锁存器、主从结构的触发器、边沿结构的触发器的工作特点。

4-10　解释边沿触发器的工作速度高于主从结构触发器的原因。

4-11　设图 4-40 中各 TTL 工艺触发器的初始状态为 0，写出各触发器的状态方程，并画出 CP 波形作用下各触发器输出端的波形。

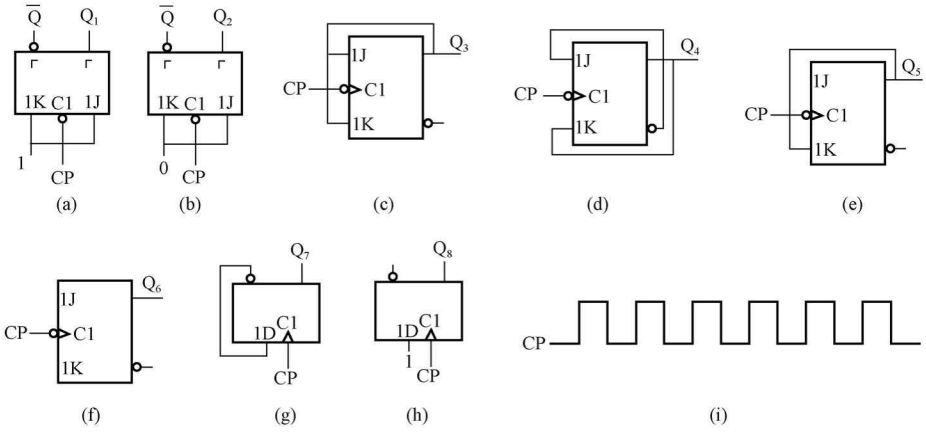


图 4-40　题 4-11 图

4-12　维持阻塞 D 触发器组成的电路如图 4-41 所示。已知输入信号的波形［见图 4-41（b）］，试画出与之对应的输出 Q 的波形。设触发器的初始状态为 0。

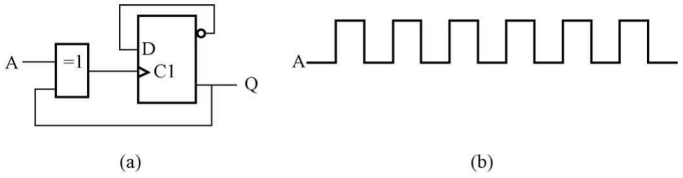


图 4-41　题 4-12 图

4-13　已知边沿 JK 触发器各输入端的波形如图 4-42 所示，试画出触发器 Q 端波形。

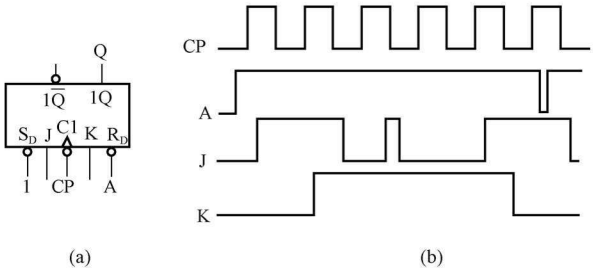


图 4-42 题 4-13 图

4-14 写出图 4-43（a）、（b）中触发器 Q 的特性方程，在已知信号［见图 4-43（c）］的作用下画出 Q 的波形。

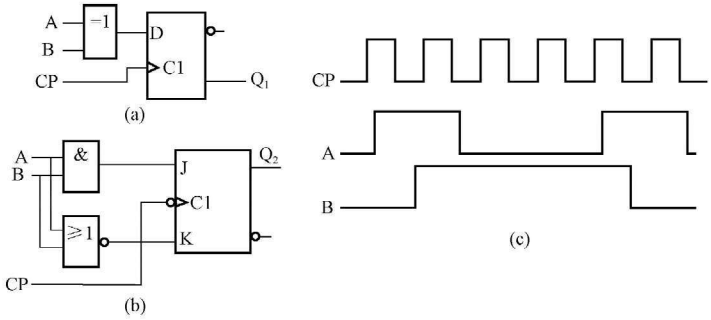


图 4-43 题 4-14 图

4-15 主从结构的 JK 触发器中，已知 CP 和输入信号 T 的波形如图 4-44（b）所示，试画出触发器输出端 Q 和$\overline{Q}$的波形。设触发器的初状态为 0。

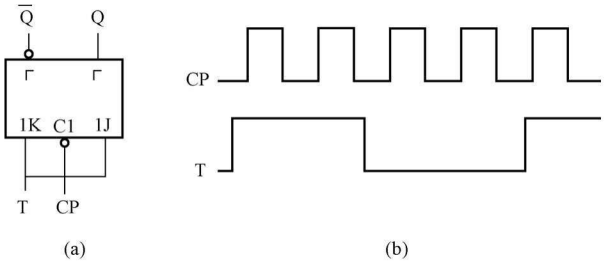


图 4-44 题 4-15 图

4-16 触发器组成的电路图及波形图分别如图 4-45（a）、（b）所示，设触发器的初始状态为 0，试画出电路输出端 Y、Z 的波形。

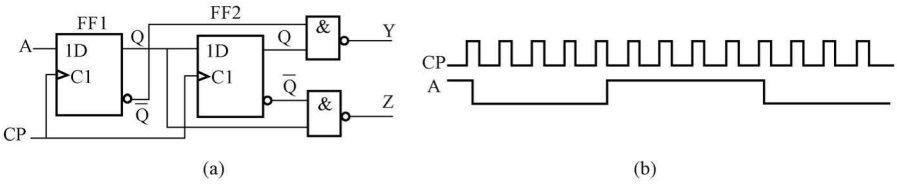


图 4-45 题 4-16 图

4-17　试画出图 4-46 所示电路在一系列 CP 作用下 Q_1、Q_2、Q_3 端输出电压波形，并说明 Q_1、Q_2、Q_3 输出信号的频率与 CP 信号的频率之间的关系。触发器均为边沿结构，初始状态为 0。

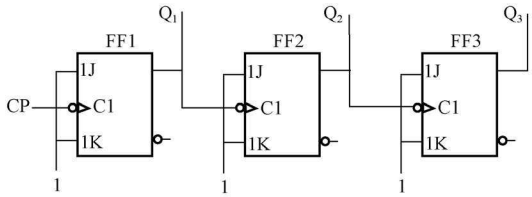


图 4-46　题 4-17 图

4-18　某电路如图 4-47（a）所示，试画出在 CP、R_D 信号［见图 4-47（b）］作用下 Q_1、Q_2、Q_3 的输出电压波形，并说明 Q_1、Q_2、Q_3 输出信号的频率与 CP 信号的频率之间的关系。

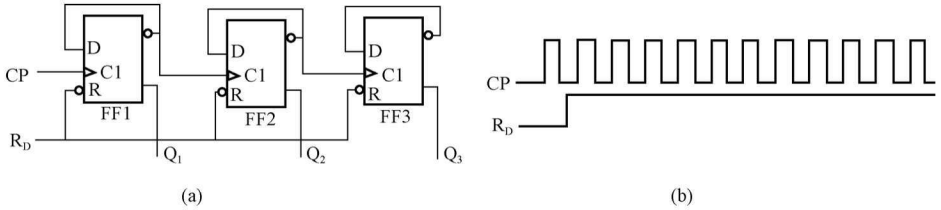


图 4-47　题 4-18 图

4-19　一个触发器的特性方程是 $Q^{n+1}=X\oplus Y\oplus Q^n$，试分别用 JK 触发器、D 触发器实现其功能，画出逻辑电路图。

4-20　设计一个三人抢答电路。要求满足下列条件：三人各控制一个按钮和一个发光二极管，用按动按钮发出抢答信号；竞赛开始后，谁先按下按钮，谁的发光二极管就亮，同时使其他人的抢答无效。

第五章　时 序 逻 辑 电 路

本　章　提　要

本章首先介绍时序逻辑电路的基本概念、特点、分类和描述方法，以及时序逻辑电路的一般分析方法，然后介绍同步时序逻辑电路的设计方法，最后讨论典型时序逻辑部件计数器和寄存器的工作原理、逻辑功能、集成芯片及其使用方法和典型应用。

第一节　概　　述

按照逻辑功能和电路组成的不同特点，常常把数字电路分成两大类：一类是在第三章已经介绍了的组合逻辑电路；还有一类就是时序逻辑电路（Sequential Circuit）。

组合逻辑电路的框图如图 5－1 所示，其中 x_1，x_2，…，x_i 是其输入信号，z_1，z_2，…，z_j 是其输出信号。在某一时刻该组合逻辑电路的输出信号 z_1，z_2，…，z_j 只与该时刻的输入信号 x_1，x_2，…，x_i 有关。时序逻辑电路的框图如图 5－2 所示，x_1，x_2，…，x_i 是其输入信号，z_1，z_2，…，z_j 是其输出信号。在某一时刻该时序逻辑电路的输出信号 z_1，z_2，…，z_j 不但与该时刻的输入信号 x_1，x_2，…，x_i 有关，还与该时刻以前的输入信号的取值有关。该时刻以前的输入信号的曾经的取值，即输入信号的历史，表示为时序逻辑电路现在的状态。状态记忆的是输入信号的历史，要记忆的输入信号的历史时间越长，电路的状态数设置得就要越多。状态是时序逻辑电路的一个非常重要的概念。

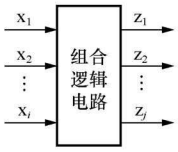


图 5－1　组合逻辑电路的框图

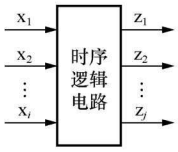


图 5－2　时序逻辑电路的框图

假定两个时刻 t_1 和 t_2，对组合逻辑电路而言，若这两个时刻的输入信号 x_1，x_2，…，x_i 完全相同，那么这两个时刻的输出信号 z_1，z_2，…，z_j 的值一定相同。而对时序逻辑电路而言，若这两个时刻的输入信号 x_1，x_2，…，x_i 完全相同，这两个时刻的输出信号 z_1，z_2，…，z_j 的值不一定相同。因为在某一时刻时序逻辑电路的输出信号 z_1，z_2，…，z_j 不但与该时刻的输入信号 x_1，x_2，…，x_i 有关，还与该时刻以前的输入信号的取值有关，时刻 t_1 之前的输入信号和时刻 t_2 之前的输入信号的取值不一定相同。

一、时序逻辑电路的结构及特点

在时序逻辑电路中，任一时刻的输出不仅与该时刻的输入有关，而且还与该时刻电路的状态有关，或者说与该时刻之前的电路的输入有关，即时序逻辑电路具有记忆功能。因此，时序电路中必须含有具有记忆能力的存储电路。存储电路能把输入信号的历史存储起来，不

同的输入信号的历史表现为电路的不同的状态。存储电路可以由触发器构成，也可以由带有反馈的组合电路构成。

时序逻辑电路的基本结构框图如图 5 - 3 所示。从结构上看时序逻辑电路有两个特点：一是由组合逻辑电路和存储电路两部分组成；二是输出、输入之间至少有一条反馈路径。

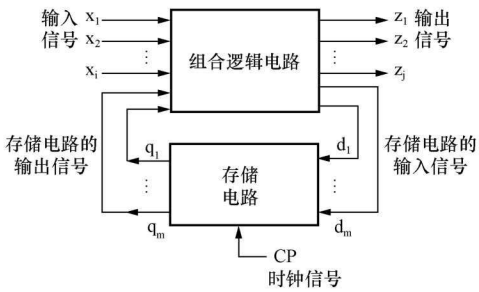


图 5 - 3　时序逻辑电路的基本结构框图

图 5 - 3 中，x_1，x_2，…，x_i 代表时序逻辑电路的输入信号，用 X（x_1，…，x_i）表示；z_1，z_2，…，z_j 代表时序逻辑电路的输出信号，用 Z(z_1，…，z_j）表示；d_1，…，d_m 代表存储电路的输入信号，称为激励信号或驱动信号，用 D(d_1，…，d_m）表示；q_1，…，q_m 代表存储电路的输出信号，用 Q(q_1，…，q_m）表示，它代表了时序逻辑电路在某一时刻所处的状态，所以又称为时序逻辑电路的状态信号。这些信号之间的逻辑关系可表示为

$$Z = F_1(X, Q^n) \tag{5-1}$$

$$D = F_2(X, Q^n) \tag{5-2}$$

$$Q^{n+1} = F_3(D, Q^n) = F_3(X, Q^n) \tag{5-3}$$

式中：Q^{n+1}是次态；Q^n是现态。

式（5 - 1）是输出方程，式（5 - 2）是存储电路的激励方程（或称驱动方程），式（5 - 3）是电路的状态方程。

二、时序逻辑电路的分类

时序逻辑电路可分为同步时序逻辑电路（Synchronous Sequential Circuit）和异步时序逻辑电路（Asynchronous Sequential Circuit）两大类。异步时序逻辑电路又分为脉冲型异步时序逻辑电路和电平型异步时序逻辑电路。同步时序逻辑电路和脉冲型异步时序逻辑电路中的存储电路由触发器构成。本章主要介绍这种存储电路由触发器构成的时序逻辑电路。电平型异步时序逻辑电路的运算速度快，是很受关注的一个研究领域，这方面的资料可参阅有关文献，本书不做介绍。

在同步时序逻辑电路中，所有触发器的时钟脉冲信号是同一个信号，因此，所有触发器状态的变化都是在所加的时钟脉冲信号的控制下同时发生的；在脉冲型异步时序逻辑电路中，触发器没有统一的时钟脉冲信号，各触发器状态的变化不是同时发生的。

从电路输出信号和输入信号的关系而言，时序逻辑电路又可分为米里（Mealy）型和莫尔（Moore）型两种。米里型时序逻辑电路的一般框图如图 5 - 4 所示。在米里型电路中，输出信号同时取决于电路状态和输入信号。莫尔型时序逻辑电路的一般框图如图 5 - 5 所示。在莫尔型电路中，输出信号只取决于电路状态，与输入信号没有直接的关系。

三、时序逻辑电路的描述方法

描述一个时序逻辑电路的逻辑功能可以采用逻辑方程组（驱动方程、输出方程、状态方程）、状态转移表、状态转移图、时序图等方法。这些方法从不同角度描述了时序逻辑电路逻辑功能的特点，它们在本质上是完全相同的，可以相互转换，是分析和设计时序逻辑电路的基本工具。

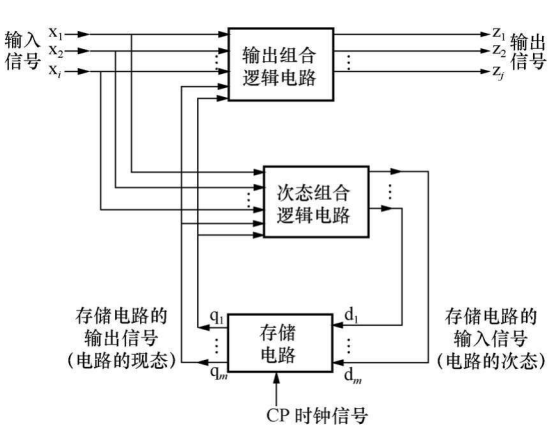


图 5-4 米里型时序逻辑电路的一般框图

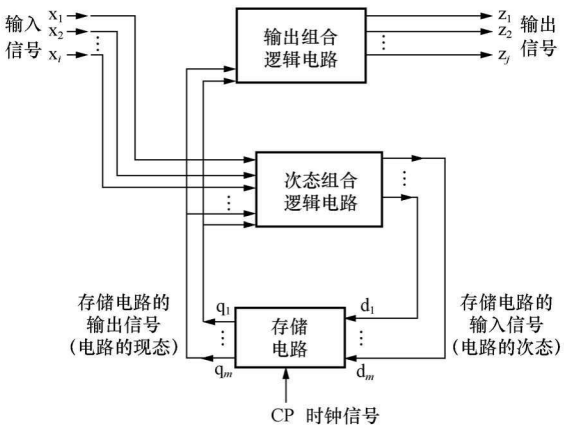


图 5-5 莫尔型时序逻辑电路的一般框图

1. 逻辑方程组

时序逻辑电路的驱动方程、输出方程和状态方程清楚简练地描述了时序逻辑电路，但不够直观，根据这三个方程式不能判断时序逻辑电路的逻辑功能。此外，设计时序逻辑电路时，也很难根据给定的逻辑要求而直接写出电路的逻辑方程。

2. 状态转换表（State Transition Table）

表 5-1 米里型时序逻辑电路的状态表的画法

$Q_1^{n+1}Q_0^{n+1}/Z$ \ X ; $Q_1^nQ_0^n$	0	1
00	01/0	00/0
10	10/0	00/0
01	10/1	00/0
11	10/0	00/0

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转移表，简称状态表。表 5-1 是某个时序逻辑电路的状态表，该电路是一个米里型电路，有一个输入信号 X，一个输出信号 Z，有两个触发器。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 Q^n，表的内部列出次态 Q^{n+1} 和输出 Z。状态表给出了时序逻辑电路在给定的现态 Q^n 和输入 X 不同取值组合情况下，该电路的输出信号 Z 和次态 Q^{n+1} 的取值。在表 5-1 中，现态 Q^n 和输入 X 共有八种不同的取值组合。

3. 状态转换图（State Transition Diagram）

状态转换图，简称状态图，是反映时序逻辑电路状态转换规律与相应输入、输出取值关系的图形。状态图比其他的描述方法更形象。状态图可以从状态表得到，从状态表 5-1 得到的状态图如图 5-6 所示。

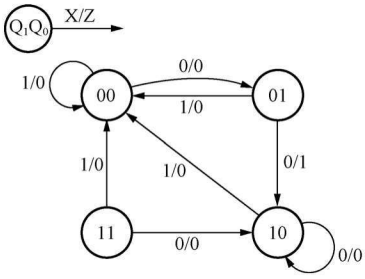


图 5-6 表 5-1 给定的时序逻辑电路的状态图

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，则表示经过一个时钟脉冲的有效沿作用之后电路的状态不变。标在连线一侧的数字表示状态转换前输入信号的取值和在输入信号作用之下的输出信号的取值，用“/”分隔开。通常将输入信号取值写在“/”的左侧，输出信号的取值写在“/”的右侧。它清楚

地表明：在该输入取值作用下，将产生相应输出值，稍后电路将发生如箭头所指的状态转换。

例如，电路现在的状态为 $Q_1Q_0=00$，在输入 X=0 时，输出 Z 为 0，在时钟脉冲的有效沿作用之后，转为次态 $Q_1Q_0=01$；在输入 X=1 时，输出 Z 为 0，在时钟脉冲的有效沿作用之后，状态不变，仍为 $Q_1Q_0=00$。

4. 时序图

时序图，又称波形图，它能直观地描述时序逻辑电路的输入信号 X、时钟信号 CP、输出信号 Z 及电路的状态 Q 在时间上的对应关系。时序图由状态表或状态图得到，状态图 5-6 所描述的时序逻辑电路，在给定的输入信号 X 作用下的时序图如图 5-7 所示。假设触发器为下降沿触发，电路的初态为 $Q_1Q_0=00$，为了清晰起见，没有考虑组合逻辑电路和触发器的输出延迟。此时序逻辑电路是米里型电路，输出信号 Z 是输入信号 X 和现态 Q_1Q_0 的函数，只要输入信号 X 或触发器的 Q_1Q_0 任意一个有变化，输出信号就要根据表 5-1 所示的表格变化，而次态要等到紧邻的一个时钟的下降沿到来时才改变。

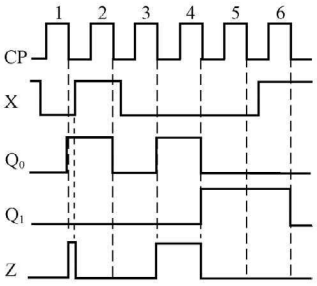


图 5-7　表 5-1 给定的时序逻辑电路的时序图

第二节　时序逻辑电路的分析

所谓时序逻辑电路的分析就是要找出给定时序逻辑电路的逻辑功能。具体说，就是根据给定的时序逻辑电路，分析在时钟信号和输入信号的作用下，时序逻辑电路的输出信号的变化规律和电路的状态的转换规律，进而说明该时序逻辑电路的逻辑功能。

一、时序逻辑电路分析的一般步骤

(1) 观察电路的结构，确定电路是同步时序逻辑电路还是异步时序逻辑电路，是米里型电路还是莫尔型电路；

(2) 写出各触发器的时钟方程；

(3) 写出时序逻辑电路的输出方程；

(4) 写出各触发器的驱动方程；

(5) 将各触发器的驱动方程代入其特性方程，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；

(6) 根据状态方程得到该时序逻辑电路的状态表；

(7) 根据状态表得到该时序逻辑电路的状态图；

(8) 在给定的输入信号作用下得到该时序逻辑电路的时序图；

(9) 根据状态图分析该时序逻辑电路的功能。

需要说明的是，上述步骤不是必须遵循的固定步骤，实际应用中可根据具体情况加以取舍。如在分析同步时序逻辑电路时，各触发器的时钟信号的逻辑表达式就可以不写。

二、同步时序逻辑电路的分析举例

(一) 米里型同步时序逻辑电路的分析

下面通过例题进行米里型同步时序逻辑电路的分析。

【例 5-1】 试分析图 5-8 所示的时序逻辑电路的功能。

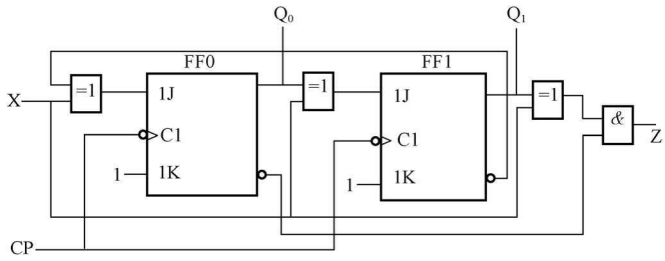


图 5-8 [例 5-1] 的时序逻辑电路图

解 (1) 观察电路的结构，可知有两个 JK 触发器，分别记为 FF0 和 FF1。这两个 JK 触发器的时钟信号相同，所以这个电路是同步时序逻辑电路。该电路有一个输入信号 X，一个输出信号 Z。观察输出信号 Z，可以看出它既与触发器状态 Q 有关，又与外部输入信号 X 有关，所以这是米里型同步时序逻辑电路。时钟信号虽然也是输入信号，但是它的作用与输入信号 X 的作用有些不同，时钟信号是控制整个电路工作步调的一个时间基准信号。所以在描述时序逻辑电路的输入信号的时候，不把时钟信号当作一般意义的输入信号。

(2) 写出各触发器的时钟方程。由于图 5-8 为逻辑门控时序逻辑电路，图中的两个触发器都接至同一个时钟脉冲源 CP，所以各触发器的时钟方程可以不写。

(3) 写出时序逻辑电路的输出方程，即

$$Z = (X \oplus Q_1^n)\overline{Q_0^n} \tag{5-4}$$

(4) 写出各触发器的驱动方程，FF0 触发器的输入信号分别记为 J_0 和 K_0，FF1 触发器的输入信号分别记为 J_1 和 K_1，表达式分别为

$$J_0 = X \oplus \overline{Q_1^n}, K_0 = 1 \tag{5-5}$$

$$J_1 = X \oplus Q_0^n, K_1 = 1 \tag{5-6}$$

(5) 将各触发器的驱动方程代入其特性方程，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。

JK 触发器的特性方程为 $Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n$，将式 (5-5) 和式 (5-6) 表示的驱动方程代入 JK 触发器的特性方程，得 *FF*0 和 *FF*1 触发器的次态方程，也就是时序逻辑电路的状态方程为

$$Q_0^{n+1} = J_0\overline{Q_0^n} + \overline{K_0}Q_0^n = (X \oplus \overline{Q_1^n})\overline{Q_0^n} \tag{5-7}$$

$$Q_1^{n+1} = J_1\overline{Q_1^n} + \overline{K_1}Q_1^n = (X \oplus Q_0^n)\overline{Q_1^n} \tag{5-8}$$

表 5-2 [例 5-1] 的时序逻辑电路的状态表

$Q_1^{n+1}Q_0^{n+1}/Z$ \ X ; $Q_1^nQ_0^n$	0	1
00	01/0	10/1
01	10/0	00/0
10	00/1	01/0
11	00/0	00/0

(6) 根据状态方程得到该时序逻辑电路的状态表。列状态表是分析时序逻辑电路功能的关键性的一步，其具体做法是：先填入电路现态 Q^n 的所有可能组合以及输入信号 X 的所有组合；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} 的相应值。状态表描述了电路经过一个时钟脉冲的有效沿作用之后电路状态变化的规律。

（7）根据状态表得到该时序逻辑电路的状态图如图 5 - 9 所示。因为此时序逻辑电路有一个输入信号 X，所以从每个状态出来两个箭头，分别表示当输入信号 X＝0 时和 X＝1 时电路将要到达的次态。

（8）在给定的输入信号作用下得到该时序逻辑电路的时序图。给定输入信号 X 的取值如图 5 - 10 中的波形所示，假定电路的初态为 $Q_1Q_0=00$。在一系列 CP 脉冲作用下，FF0 和 FF1 触发器的状态 Q_0、Q_1 和输出 Z 的波形如图 5 - 10 所示。在画波形的过程中，可以根据状态图先画出 Q_0、Q_1 的波形，然后再根据式（5 - 4）给定的输出信号表达式 $Z=(X\oplus Q_1^n)\overline{Q_0^n}$，画出 Z 的波形。从图 5 - 10 中可以看出，在 t_1 时刻电路处于 $Q_1Q_0=00$ 的状态下，当 X 从 0 变为 1 时，输出 Z 经过几个门电路的延时（在此波形图中为了更清楚地表达电路的基本逻辑功能，将门电路和触发器的延迟均忽略）从 0 变为 1，在紧接着的 CP 脉冲的下降沿 t_2 时刻电路的状态变为 $Q_1Q_0=10$，即输出信号先到达预定的值，然后电路再进入预定的状态，这就是米里型时序逻辑电路。其实从式（5 - 4）输出方程也可以看出输出信号、输入信号与现态有关，可得出该电路是米里型时序逻辑电路。

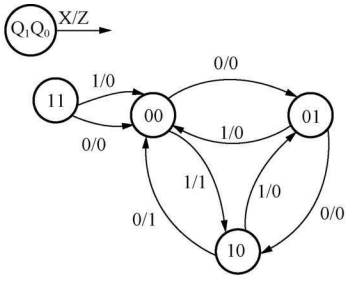


图 5 - 9　［例 5 - 1］的时序逻辑电路的状态图

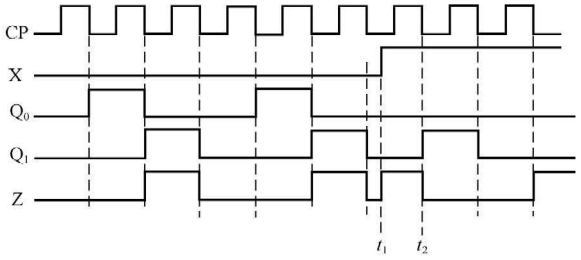


图 5 - 10　［例 5 - 1］的时序逻辑电路的波形图

（9）根据状态图分析该时序逻辑电路的功能。为了便于分析电路的功能，分别画出当 X＝0 时的状态图和 X＝1 时的状态图，如图 5 - 11 和图 5 - 12 所示。

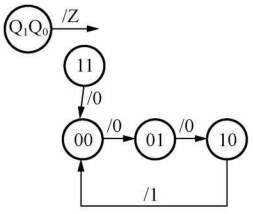


图 5 - 11　当 X＝0 时的状态图

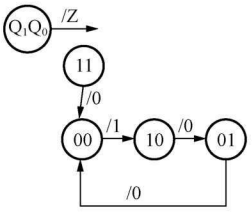


图 5 - 12　当 X＝1 时的状态图

在图 5 - 11 中，当 X＝0 时，电路的状态按 00→01→10→00 循环变化，若电路处于 11 状态，经过一个 CP 信号的下降沿之后，电路进入 00 状态。在 10 状态下，当 X＝0 到来时，输出信号 Z＝1，其他情况下输出信号 Z＝0。这是一个 3 进制计数器，状态按照 00→01→10→00 的规律变化。电路的状态是用触发器状态的组合来表示的，要特别注意理解，状态本身不是二进制数，但是在描述这些时序逻辑电路时，为了方便，习惯把这些状态人为赋予相应的位权，看成二进制数，这样在描述电路的功能时能够用更简练的语言表达。例如，00→01→10→00 可以看成 0→1→2→0，次态比现态多 1，为“加”计数器。在 10 状态

下，当 X=0 到来时，输出信号 Z=1。

在图 5-12 中，当 X=1 时，电路的状态按 10→01→00→10 循环变化，若电路处于 11 状态，经过一个 CP 信号的下降沿之后，电路进入 00 状态。在 00 状态下，当 X=1 到来时，输出信号 Z=1，其他情况下输出信号 Z=0。所以这是一个 3 进制计数器，状态按照 10→01→00→10 的规律变化。如果给状态赋予相应的位权可以看成 2→1→0→2，次态比现态减 1，为“减”计数器。在 00 状态下，当 X=1 到来时，输出信号 Z=1。

本例中的时序逻辑电路一共有 4 个状态，即 00、01、10、11，并具有自启动能力。自启动能力是指当电路由于意外的干扰进入无效的状态（不是预先设计的有用状态）时，经过几个有限的时钟脉冲之后，电路可以进入预先设计的有用状态循环，具有抗干扰能力。从状态图上看，就是所有的状态之间都有箭头相连，没有分割成几个独立的图。

综上所述，该电路是一个可控的 3 进制计数器。当 X=0 时，作“加”计数，Z 是进位信号；当 X=1 时，作“减”计数，Z 是借位信号。

计数器的状态数叫做计数器的容量，也称为模。[例 5-1] 所讲的计数器是 3 进制计数器，有 3 个状态，也叫模 3 计数器。计数是指这个时序逻辑电路可以记忆已经到来的 CP 脉冲周期的个数，电路的不同状态表示已经到来的 CP 脉冲的周期的个数不同。在 [例 5-1] 中，CP 脉冲周期的个数只能记忆 3 个，若想记忆得更多，必须增加电路的状态，使用更多的触发器。

在 [例 5-1] 中，当 X=0 时，$Q_1Q_0=00$ 表示这是第 1 个周期的 CP 脉冲，$Q_1Q_0=01$ 表示这是第 2 个周期的 CP 脉冲，$Q_1Q_0=10$ 表示这是第 3 个周期的 CP 脉冲。当 X=1 时，$Q_1Q_0=10$ 表示这是第 1 个周期的 CP 脉冲，$Q_1Q_0=01$ 表示这是第 2 个周期的 CP 脉冲，$Q_1Q_0=00$ 表示这是第 3 个周期的 CP 脉冲。

【例 5-2】 分析图 5-13 所示时序逻辑电路的功能。

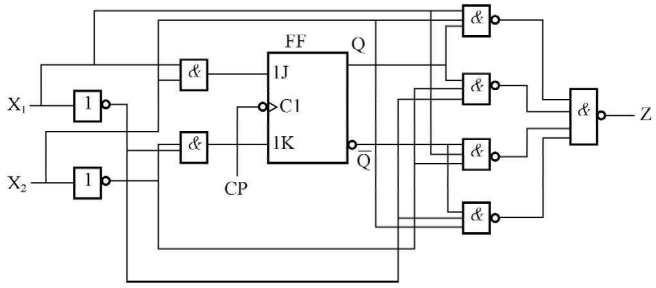


图 5-13 [例 5-2] 的逻辑电路图

解 (1) 观察电路的结构，可知有一个 JK 触发器，该电路有两个输入信号 X_1、X_2，一个输出信号 Z。观察输出信号 Z，可以看出它既与触发器状态 Q 有关，又与外部输入信号 X_1、X_2 有关，所以这是米里型时序逻辑电路。因为只有一个触发器，所以不用区分是同步时序逻辑电路还是异步时序逻辑电路。

(2) 写出时序逻辑电路的输出方程为

$$
\begin{aligned}
Z &= \overline{\overline{X_1X_2Q^n}\ \overline{\overline{X_1}\ \overline{X_2}Q^n}\ \overline{X_1\ \overline{X_2}\ \overline{Q^n}}\ \overline{\overline{X_1}X_2\ \overline{Q^n}}} \\
&= X_1X_2Q^n + \overline{X_1}\ \overline{X_2}Q^n + X_1\ \overline{X_2}\ \overline{Q^n} + \overline{X_1}X_2\ \overline{Q^n}
\end{aligned}
\tag{5-9}
$$

（3）写出 JK 触发器的驱动方程，JK 触发器的输入信号分别记为 J 和 K，即

$$J = X_1X_2, K = \overline{X_1}\,\overline{X_2} \tag{5-10}$$

（4）将触发器的驱动方程代入其特性方程，求得触发器的次态方程，也就是时序逻辑电路的状态方程。

JK 触发器的特性方程为 $Q^{n+1}=J\overline{Q^n}+\overline{K}Q^n$，将式（5-10）表示的驱动方程代入 JK 触发器的特性方程，得触发器的次态方程，也就是时序逻辑电路的状态方程为

$$Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n = X_1X_2\overline{Q^n} + (X_1 + X_2)Q^n \tag{5-11}$$

（5）根据状态方程式（5-11）得到该时序逻辑电路的状态表，如表 5-3 所示。

（6）根据状态表 5-3 得到该时序逻辑电路的状态图，如图 5-14 所示。

表 5-3　　［例 5-2］的状态表

Q^{n+1}/Z \ X_1X_2 (Q^n)	00	01	10	11
0	0/0	0/1	0/1	1/0
1	0/1	1/0	1/0	1/1

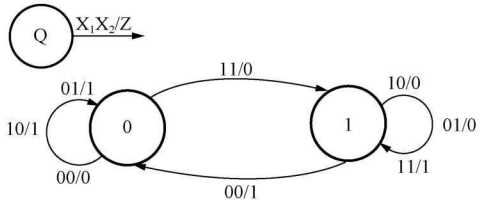


图 5-14　［例 5-2］的状态图

（7）在给定的输入信号作用下得到该时序逻辑电路的时序图，如图 5-15 所示。

（8）根据状态图分析该时序逻辑电路的功能。这个电路是一个串行加法运算电路，X_1 串行输入被加数的值，每个值维持一个时钟周期长，X_2 串行输入加数的值，每个值维持一个时钟周期长，Z 串行输出被加数与加数相加得到的和，相加产生的进位保存在触发器中，触发器的初态要设置为 0。图 5-15 的时序图描述了 011+001 的运算，低位首先送入串行输入端，被加数 $X_1=1$、加数 $X_2=1$ 相加，得到的和为 0，即 Z=0，进位为 1，即 Q=1；在第 1 个时钟的下降沿后送来被加数 $X_1=1$ 和加数 $X_2=0$，与低位来的进位 1 相加，得到和 Z=0，进位Q=1；在第 2 个时钟的下降沿后送来被加数 $X_1=0$ 和加数 $X_2=0$，与低位来的进位 1 相加，首先得到和 Z=1，然后在第 3 个时钟的下降沿后进位 Q=0。这是米里型同步时序逻辑电路。Mealy 型电路中，输出会出现毛刺，可在 Z 输出端经 D 触发器输出，从而去掉毛刺。通过此例题，我们可以体会到，二进制数的加法运算分别可以用组合逻辑电路和时序逻辑电路来实现。用组合逻辑电路时，电路中门的个数较多，尤其当做加法运算的二进制数位数较多时，但电路的运算速度快。而用时序逻辑电路实现多位二进制数加法运算时，电路规模较小，电路中门电路的个数较少，但电路的运算速度慢，尤其是当想增加二进制数位数时，只需在输入端联系输入即可，不需改变电路结构。在实际应用中，到底选用组合电路还是时序电路，取决于运算速度和电路复杂程度的要求。

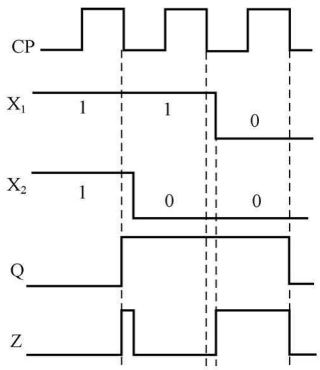


图 5-15　［例 5-2］的时序图

（二）莫尔型同步时序逻辑电路的分析

下面通过例题进行莫尔型同步时序逻辑电路的分析。

【例 5-3】 分析图 5-16 所示时序逻辑电路的功能。

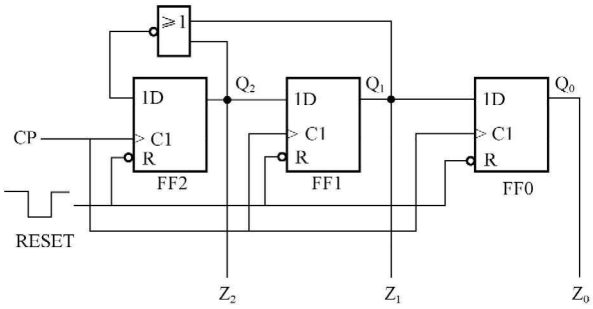


图 5-16 ［例 5-3］的逻辑电路图

解 （1）观察电路的结构，有三个上升沿触发的 D 触发器，这三个触发器的时钟信号是同一个信号，所以该电路是同步时序逻辑电路。该时序逻辑电路没有一般意义的输入信号，有三个输出信号 Z_2、Z_1、Z_0，这三个输出信号正好分别是三个触发器的状态信号，所以是莫尔型电路。RESET 信号是给触发器置初态的信号，低电平有效。在电路正常工作时，RESET 信号应处于无效状态。

（2）写出各触发器的时钟方程。由于图 5-16 为同步时序逻辑电路，图中的三个触发器都接至同一个时钟脉冲源 CP，所以各触发器的时钟方程可以不写。

（3）写出时序逻辑电路的输出方程，即

$$Z_2 = Q_2^n, Z_1 = Q_1^n, Z_0 = Q_0^n$$

（4）写出各触发器的驱动方程，即

$$\begin{cases} D_2 = \overline{Q_2^n + Q_1^n} \\ D_1 = Q_2^n \\ D_0 = Q_1^n \end{cases}$$

（5）将各触发器的驱动方程代入其特性方程 $Q^{n+1}=D$，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为

$$\begin{cases} Q_2^{n+1} = \overline{Q_2^n + Q_1^n} \\ Q_1^{n+1} = Q_2^n \\ Q_0^{n+1} = Q_1^n \end{cases}$$

（6）根据状态方程得到该时序逻辑电路的状态表。由于此电路中 $Z_2Z_1Z_0=Q_2^nQ_1^nQ_0^n$，输出信号与现态完全相同，所以状态表中不再列出输出信号。又因为这个电路中没有输入信号 X，状态表中的次态仅有一列。此电路的状态表可简化成表 5-4 所示的形式。

（7）根据状态表 5-4 得到该时序逻辑电路的状态图，如图 5-17 所示。

表 5-4　［例 5-3］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}	Z_2	Z_1	Z_0
0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0	1
0	1	0	0	0	1	0	1	0
0	1	1	0	0	1	0	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	0	1
1	1	0	0	1	1	1	1	0
1	1	1	0	1	1	1	1	1

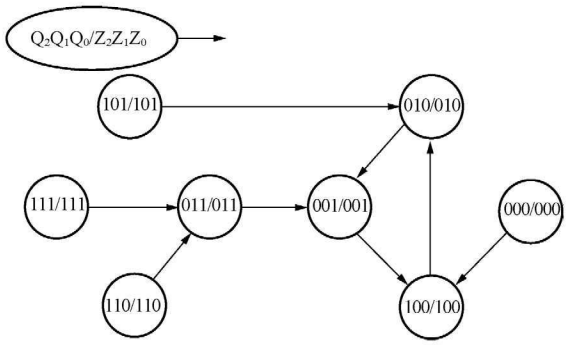


图 5-17　［例 5-3］的状态图

（8）在给定的输入信号作用下得到该时序逻辑电路的时序图。设电路的初态为 $Q_2Q_1Q_0=000$，根据状态表和状态图，可画出在一系列 CP 脉冲作用下该电路的时序图，如图 5-18 所示。

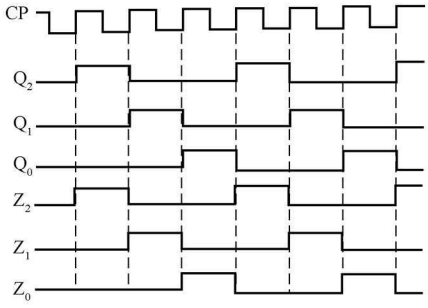


图 5-18　［例 5-3］的时序图

（9）根据状态图分析该时序逻辑电路的功能。

在状态图 5-17 中，100、010、001 这 3 个状态形成了闭合圈，取这 3 个状态作为有效循环状态，构成一个 3 进制计数器。000、011、101、110、111 这 5 个状态被称为无效状态。对于含有无效状态的时序电路来说，假如电路由于某种原因进入无效状态，在有限个时钟脉冲作用下，最终能走到有效循环中去，则称此电路具有自启动能力；反之，如果无效状态之间构成死循环，则称此电路不具有自启动能力。对于本例，无论电路处在 000、011、101、110、111 中的哪一个，经过有限个时钟脉冲作用后总是可以回到有效状态循环中，所以该电路具有自启动能力。

从时序图 5-18 来看，该时序电路的逻辑功能可以认为是脉冲分配器或节拍脉冲产生器。电路在正常工作时，各触发器的 Q 端轮流出现了一个脉冲信号，其宽度为一个 CP 周期，即 $1T_{CP}$，循环周期为 $3T_{CP}$。这个动作可以看成是在 CP 脉冲作用下，电路把宽度为 $1T_{CP}$ 的脉冲依次分配给 Z_2、Z_1、Z_0 各端。

三、异步时序逻辑电路的分析举例

异步时序电路的分析和同步时序电路的分析方法和步骤基本相同，不同之处在于异步时序电路需考虑各个触发器的时钟方程 CP。由于在异步时序逻辑电路中，没有统一的时钟脉冲，因此分析时必须写出时钟方程。在分析异步时序电路的状态转换时，要特别注意各触发器是否具备有效的触发信号（上升沿或下降沿），仅在具备有效触发信号时触发器才可能动作，否则将保持原状态不变。

现举例说明异步时序逻辑电路的分析过程。

【例 5-4】　试分析图 5-19 所示的时序逻辑电路的功能。

解　（1）观察电路的结构，有两个上升沿触发的 D 触发器，分别记为 FF0 和 FF1。这两个 D 触发器的时钟信号不是同一个信号，所以该电路是脉冲型异步时序逻辑电路。该电

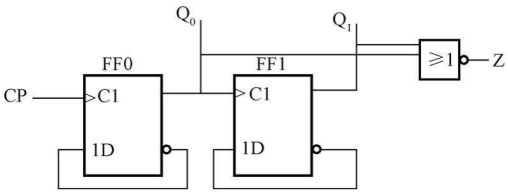


图 5 - 19 ［例 5 - 4］的时序逻辑电路图

路没有一般意义的输入信号，有一个输出信号 Z，输出信号只与两个 D 触发器的状态有关，所以是莫尔型电路。

（2）写出各触发器的时钟方程。因为是脉冲型异步时序逻辑电路，所以必须写出时钟方程。

FF0 触发器的时钟方程为 $CP_0=CP$（时钟脉冲源的上升沿触发）。

FF1 触发器的时钟方程为 $CP_1=Q_0$（当 FF0 的 Q_0 由 0→1 时，Q_1 才可能改变状态，否则 Q_1 将保持原状态不变）。

（3）写出时序逻辑电路的输出方程，即

$$Z=\overline{Q_1^n+Q_0^n}=\overline{Q_1^n}\,\overline{Q_0^n}$$

（4）写出各触发器的驱动方程，即

$$D_0=\overline{Q_0^n},D_1=\overline{Q_1^n}$$

（5）将各触发器的驱动方程代入其特性方程，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，即

$$Q_0^{n+1}=D_0=\overline{Q_0^n}(\text{CP 由 } 0\rightarrow 1 \text{ 时此式有效})$$

$$Q_1^{n+1}=D_1=\overline{Q_1^n}(Q_0 \text{ 由 } 0\rightarrow 1 \text{ 时此式有效})$$

（6）根据状态方程得到该时序逻辑电路的状态表。列状态表的方法与同步时序逻辑电路基本相似，只是要特别注意各触发器是否具备有效的时钟信号（此处为上升沿作用），为了清晰起见，在状态表中增加了各触发器 CP 信号的状况。特别注意，不具备时钟信号的上升沿时的 CP 用 0 表示，即此处的 0 不表示 CP 时钟信号为 0，只是表示时钟沿不具备而已。具备时钟信号的上升沿时用一个向上的箭头表示，表示有一个上升沿。状态表如表 5 - 5 所示。

表 5 - 5　［例 5 - 4］的状态表

现态		次态		输出	时钟脉冲	
Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	Z	CP_1	CP_0
0	0	1	1	1	↑	↑
0	1	0	0	0	0	↑
1	0	0	1	0	↑	↑
1	1	1	0	0	0	↑

（7）根据状态表得到该时序逻辑电路的状态图如图 5 - 20 所示。

（8）在时钟脉冲信号作用下的该时序逻辑电路的时序图如图 5 - 21 所示。

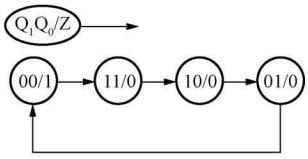


图 5 - 20 ［例 5 - 4］的状态图

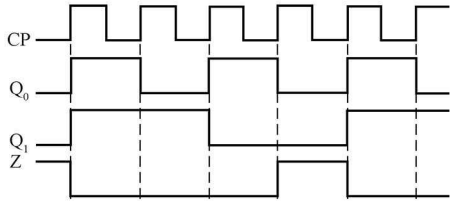


图 5 - 21 ［例 5 - 4］的时序图

（9）根据状态图分析该时序逻辑电路的功能。由图 5 - 20 可知，该电路一共有 4 个状态，即 00、01、10、11，在时钟脉冲作用下，按照“减 1”规律循环变化，所以是一个 4 进制减法计数器。$Q_1Q_0=00$ 时 Z=1，Q_1Q_0 为其他值时 Z=0。Z 是借位信号。

由时序图可以看出，该电路也是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 $1T_{CP}$。

第三节　同步时序逻辑电路的设计

同组合逻辑电路的设计类似，所谓时序逻辑电路的设计也是要求设计者从实际的逻辑问题出发，设计出满足逻辑功能要求的电路，并力求最简。当选用小规模集成电路做设计时，电路最简的标准是所用的触发器和门电路的数目最少，而且触发器和门电路的输入端数目也最少。而当使用中、大规模集成电路时，电路最简的标准是使用的集成电路个数最少，种类最少，而且相互间的连线也最少。

时序逻辑电路的状态表中，若在每一种输入组合情况下，其次态和输出都是唯一确定的，则称这种时序逻辑电路为完全规定的时序逻辑电路。若在每一种输入组合情况下，其次态或输出出现了没有明确给定的状态或输出值，则称这种时序逻辑电路为不完全规定的时序逻辑电路。

异步时序逻辑电路设计的一般方法比较复杂，此处不做讨论。本书只讨论如何用触发器和门电路来设计同步时序逻辑电路。

一、设计同步时序逻辑电路的一般步骤

1. 逻辑抽象，建立原始状态图

首先根据设计要求，分析题目要求实现的逻辑功能，确定输入信号、输出信号，分析电路应该具有的状态，并用符号表示这些状态。然后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的状态及相应的输出，便可求得符合题意的状态图。这种直接由要求实现的逻辑功能求得的状态图叫做原始状态图。这一步是基础，也是关键，因为原始状态图正确与否，将决定着所设计的电路能否实现预定的逻辑功能。在设计前，还要确定是采用 Meaty 型电路还是 Moore 型电路。

2. 状态化简，得到最简状态图

原始状态图不一定是最简的，很可能包含有多余的状态，即可以合并的状态。电路的状态数目越多，设计的电路就越复杂，因此需要进行状态化简，得出最简的状态图。下面所讲的状态化简方法是针对米里型完全规定的时序逻辑电路的状态的化简。莫尔型完全规定的时序逻辑电路的状态的化简可以使用类似的方法进行。在详细讲解状态化简的方法之前，先给出几个术语的定义。

（1）输入序列：与一串时钟脉冲相对应的一串输入组合。以图 5 - 15 所示的二进制串行加法器的波形图为例，对应序列长度为 3 的情况，图中所展示的 X_1 的输入序列为 110，X_2 的输入序列为 100。

（2）输出序列：在一串时钟脉冲及相对应的输入序列作用下产生的一串输出组合。以图 5 - 15 所示的二进制串行加法器的时序图为例，对应序列长度为 3 的情况，图中所展示的 Z 的输出序列为 001。

（3）序列长度：所考察的输入或输出序列所对应的时钟脉冲的个数。以图 5 - 15 所示的二进制串行加法器的时序图为例，图中所展示的时钟脉冲的个数为 3，所以序列长度为 3。

（4）等价状态对的定义之一：将某时序逻辑电路的初态分别设置为两个状态 S_i 和 S_j，在完全相同的任意长度 k 的输入序列的作用下，该时序逻辑电路的输出序列完全相同，此时不能分辨出这个时序逻辑电路到底是从初态 S_i，还是从初态 S_j 开始运行的。这样的两个状

态 S_i 和 S_j 被称为等价状态对。输入序列的长度 k 为任意长的意思是为无穷长的时间。

上述等价状态对的定义不具有可操作性，因为在判断两个状态是否等价时不可能施加无穷长的输入序列，所以再给出等价状态对的定义之二。

（5）等价状态对的定义之二：设状态 S_i 和 S_j 是某时序逻辑电路的两个状态，分别对其施加各种可能的输入组合 I_p，若同时满足以下条件：

1）输出完全相同，即

$$Z(S_i, I_p) = Z(S_j, I_p)\text{（此处“=”表示相同）} \tag{5-12}$$

2）后继状态等价，即

$$N(S_i, I_p) = N(S_j, I_p)\text{（此处“=”表示等价）} \tag{5-13}$$

则称 S_i 与 S_j 等价，它们是等价状态对。

后继状态等价的含义是下面四种情况中的任意一种：①两个次态完全相同；②两个次态为其现态本身或交错；③两个次态为状态封闭链中的一个状态对；④两个次态的某一后续状态对可以合并。第③、④条稍显复杂，本书后面不涉及，若读者感兴趣，可参考其他书籍。

3. 状态分配（或状态编码），画出编码后的状态图及状态表

在得到简化的状态图后，要对每一个状态指定一个二进制代码，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以有利于所选触发器的驱动方程及电路输出方程的简化为原则。编码方案确定后，根据简化的状态图，画出编码形式的状态表及状态表。

4. 选定触发器类型和个数

触发器的类型选得合适，可以简化电路结构。因为时序逻辑电路的状态是用触发器状态的不同组合来表示的，对于有 M 个状态的时序电路来说，应按式 $2^{n-1} < M \leqslant 2^n$ 确定触发器的个数 n。

5. 求出电路的输出方程和驱动方程

根据编码状态表以及所采用的触发器的逻辑功能，导出待设计电路的输出方程和驱动方程。

6. 画逻辑电路图

根据得到的方程式画出逻辑电路图。

7. 检查所设计电路能否自启动

画全状态转换图，检查设计的电路能否自启动。如果电路不能自启动，则应该修改设计。

表 5-6 ［例 5-5］的状态表

S^{n+1}/Z \ X（S^n）	0	1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

【例 5-5】 某时序逻辑电路的状态表如表 5-6 所示，根据等价状态对的定义之一判断这个时序逻辑电路的哪些状态等价。

解 （1）输入序列长度 k 为 1 时观察输出序列。电路的初始状态是 A、B、C 中的任意一个，当 X=0 时，电路的输出均为 1；当 X=1 时，电路的输出均为 0，此时称状态 A、B、C 是 1-等价（1-equivalent）。1-等价中的 1 是输入序列长度为 1 的意思。

电路的初始状态是 D、E 中的任意一个，当 X=0 时，电路的输出均为 0；当 X=1 时：电路的输出均为 1，状态 D、E 是 1-等价（1-equivalent）。

（2）当输入序列长度 k 为 2 时观察输出序列，如图 5 - 22 所示。可以看出 B、C 是 2-等价，D、E 是 2-等价。

（3）当输入序列长度 k 为 3 时观察输出序列，如图 5 - 23 所示。可以看出 B、C 是 3-等价。而 D、E 不再是 3-等价，因为当输入序列为 110 时，它们的输出序列不再相同。当输入序列为 111 时，它们的输出序列也不相同。

（4）当输入序列长度 k 为 4 时观察输出序列，观察 B、C 是否是 4-等价，若还是 4-等价，则继续进行下去，一直到 B、C 对无穷长的序列仍然等价，即可推出结论 B、C 是等价状态。若 B、C 不是 4-等价的，就可以得出结论说 B、C 状态不等价。

可以看出，这样的验证过程几乎不可能进行下去，所以验证状态是否等价应采用定义二。

初始状态	输入序列			
	00	01	10	11
A	11	10	01	00
B	11	10	00	01
C	11	10	00	01
D	00	01	11	10
E	00	01	11	10

图 5 - 22 ［例 5 - 5］在序列长度为 2 的情况下的输出序列

初始状态	输入序列							
	000	001	010	011	100	101	110	111
A	111	110	100	101	011	010	000	001
B	111	110	100	101	000	001	011	010
C	111	110	100	101	000	001	011	010
D	000	001	011	010	111	110	100	101
E	000	001	011	010	111	110	101	100

图 5 - 23 ［例 5 - 5］在序列长度为 3 的情况下的输出序列

【例 5 - 6】 根据等价状态对的定义之二，判断［例 5 - 5］给定的时序逻辑电路的哪些状态等价。

解　在表 5 - 6 中，只有一个输入信号 X，则各种可能的输入组合 I_p 就是 X＝0 和 X＝1。通过观察可以看出，对状态 B、C 而言，当 X＝0 时电路的输出均为 1，当 X＝1 时电路的输出均为 0，即输出完全相同。当 X＝0 时 B 的次态是 C，C 的次态是 B，即两个次态为交错关系；当 X＝1 时 B 的次态是 E，C 的次态也是 E，即两个次态相同。根据等价状态对的定义之二，可以得知状态 B 和 C 等价。

【例 5 - 7】 判断图 5 - 24 所示的状态图中的状态是否有等价状态；若有等价状态，则试合并等价状态得出简化的状态图。

解　将图 5 - 24 转换成状态表，如表 5 - 7 所示。

表 5 - 7 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，记为 S_2，取消 S_3，即将图 5 - 24 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，得到简化后的状态图如图 5 - 25 所示。

表 5 - 7 ［例 5 - 7］的状态表

S^n \ X（S^{n+1}/Z）	0	1
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_0/0$	$S_3/1$
S_3	$S_0/0$	$S_3/1$

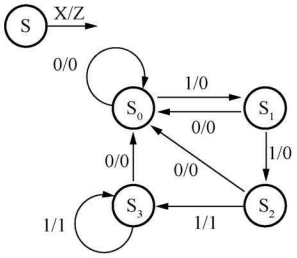


图 5 - 24 ［例 5 - 7］的原始状态图

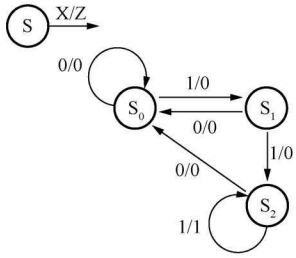


图 5 - 25 ［例 5 - 7］的简化后的状态图

二、同步时序逻辑电路设计举例

1. 简单同步时序逻辑电路的设计

如果没有外部输入变量 X，则设计过程比较简单。下面举例说明这种简单同步时序逻辑电路设计的方法。

【例 5 - 8】 设计一个同步 5 进制加法计数器，当计满后产生进位输出 Y＝1，其他情况下 Y＝0。

解 （1）要设计的同步 5 进制加法计数器电路的对外输入和输出信号如图 5 - 26 所示。前面已经介绍过，不把 CP 当作一般的输入信号，而是当作时序电路工作的时间基准。所以，此电路中没有一般意义上的输入信号，这种电路的设计比较简单。

（2）根据设计要求，设定状态，画出原始状态图。5 进制计数器应该有 5 个不同的状态，5 进制计数器的状态只要是 5 个就行，至于是哪 5 个状态，则无所谓。题目中要求是加法计数器，“加法”的含义在［例 5 - 1］中已经有介绍。5 个状态至少需要 3 个触发器来实现，得到状态图如图 5 - 27 所示，3 个触发器的状态分别用 $Q_2Q_1Q_0$ 来表示。在 CP 脉冲作用下，电路在 5 个状态中依次循环。约定状态 100 表示计数器已收到第 5 个计数器脉冲。此时输出 Y 为 1。

图 5 - 26 ［例 5 - 8］的对外输入和输出信号示意图

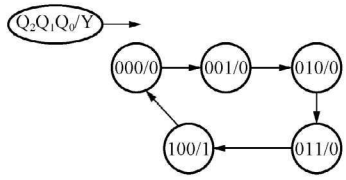


图 5 - 27 ［例 5 - 8］的状态图

（3）5 进制计数器的状态很明确，不需化简。

（4）根据状态图得到状态表如表 5 - 8 所示。

（5）选用 JK 触发器来设计该计数器。

（6）求各触发器的驱动方程和进位输出方程。根据 JK 触发器的驱动表 5 - 9，可以列出表 5 - 10 求各触发器的驱动方程和进位输出方程的列表。表 5 - 11 是把表 5 - 10 中的次态 Q_2^{n+1}、Q_1^{n+1}、Q_0^{n+1} 三列删去而得到的表格，其他内容与表 5 - 10 完全相同，这样做是为了让读者更清楚地看出驱动信号和输出信号是现态的函数。根据表 5 - 11 画出驱动信号和输出信号的卡诺图（如图 5 - 28 和图 5 - 29 所示），对驱动信号和输出信号进行化简。

表 5-8　［例 5-8］的状态表

现态			次态			进位输出
Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Y
0	0	0	0	0	1	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	0	0	0	1
1	0	1	×	×	×	×
1	1	0	×	×	×	×
1	1	1	×	×	×	×

表 5-9　JK 触发器的驱动表

$Q^n \to Q^{n+1}$		J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

表 5-10　求各触发器的驱动方程和进位输出方程的列表

现态			次态			输出	驱动信号					
Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Y	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	0	×	0	×	1	×
0	0	1	0	1	0	0	0	×	1	×	×	1
0	1	0	0	1	1	0	0	×	×	0	1	×
0	1	1	1	0	0	0	1	×	×	1	×	1
1	0	0	0	0	0	1	×	1	0	×	0	×
1	0	1	×	×	×	×	×	×	×	×	×	×
1	1	0	×	×	×	×	×	×	×	×	×	×
1	1	1	×	×	×	×	×	×	×	×	×	×

表 5-11　求各触发器的驱动方程和进位输出方程的真值表

现态			输出	驱动信号					
Q_2^n	Q_1^n	Q_0^n	Y	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	×	0	×	1	×
0	0	1	0	0	×	1	×	×	1
0	1	0	0	0	×	×	0	1	×
0	1	1	0	1	×	×	1	×	1
1	0	0	1	×	1	0	×	0	×
1	0	1	×	×	×	×	×	×	×
1	1	0	×	×	×	×	×	×	×
1	1	1	×	×	×	×	×	×	×

根据图 5-28 和图 5-29 得到驱动方程和输出方程，即

$$J_0 = \overline{Q_2^n}, K_0 = 1$$

$$J_1 = Q_0^n, K_1 = Q_0^n$$

$$J_2 = Q_0^n Q_1^n, K_2 = 1$$

$$Y = Q_2^n$$

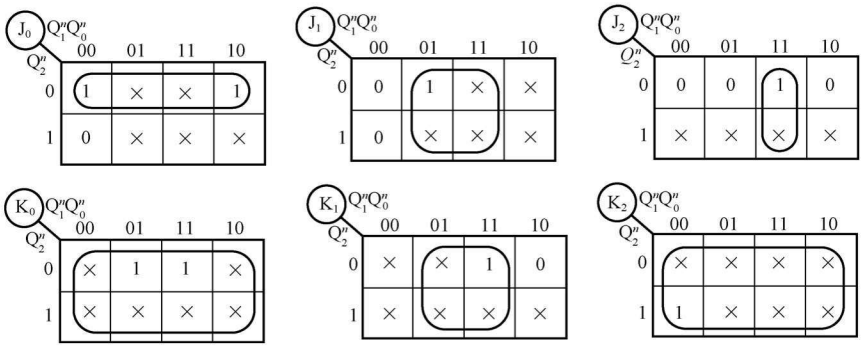


图 5 - 28 ［例 5 - 8］各触发器的驱动信号的卡诺图

（7）画逻辑电路图。根据驱动方程和输出方程，画出五进制计数器的逻辑电路图如图 5 -30 所示。

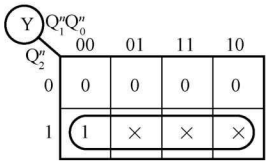


图 5 - 29 ［例 5 - 8］的输出信号的卡诺图

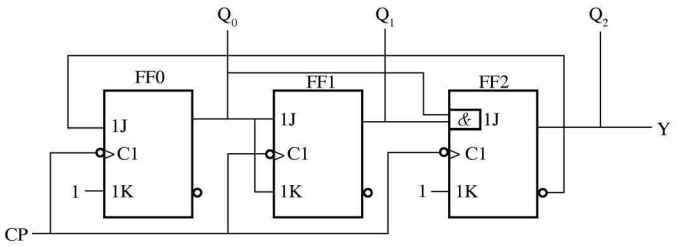


图 5 - 30 ［例 5 - 8］的逻辑电路图

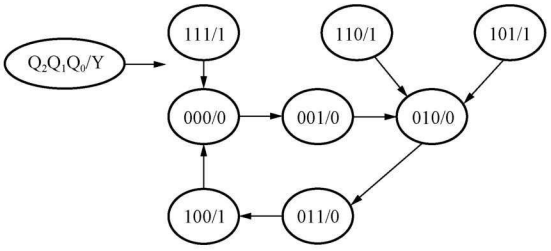


图 5 - 31 ［例 5 - 8］电路的完整的状态图

（8）检查电路能否自启动。根据图 5 - 30 的电路图，分析得到它的完整的状态图，如图 5 - 31 所示。可见，如果电路进入无效状态 101、110、111 时，在 CP 脉冲作用下，可以分别进入有效状态 010、010、000。所以电路能够自启动。

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。其方法是在表 5 - 8 中，为 101、110、111 这三个现态明确规定其次态，可以取次态与现态只有一位不同，不再全部当做无关项来处理，得到新的驱动方程和逻辑电路图，再检查其自启动能力，直到能够自启动为止。

2. 串行数据检测器的设计

串行数据检测器（有时也称为序列检测器）在数字系统中有广泛的应用，为使两个独立的电子设备能协调地工作，常由主设备向从设备发出一组串行码，当从设备完整地收到这一组串行码后，即开始进行预定的操作。这组串行码有时称为同步码或同步序列。串行数据检测器也叫串行译码器。作为对比，在第三章学过的 74X138 可以称为并行译码器。

下面举例说明串行数据检测器的设计方法。

【例 5-9】 设计一个串行数据检测器。要求该检测器可以检测 111 序列的到来，并有一个输入端 X，一个输出端 Y。当在输入端连续输入 111 时，该电路输出 Y=1，否则输出Y=0。

解 进行设计之前一定要想清楚使用米里型电路还是莫尔型电路设计，因为采用不同类型的电路设计结果将不同。本题将采用米里型设计。图 5-32 是在 CP 和 X 给定的前提下输出 Y 的波形图，从这个波形图中可以更直观地理解要设计的 111 序列检测器的功能。有这样一个问题：一个完整的 111 到来后，前面的 1 要不要作为后面 111 序列的开始？图 5-32（a）和图 5-32（b）是前面的 1 要作为后面 111 序列的开始（称为可重叠）的序列检测器的波形图。图 5-32（c）是前面的 1 不作为后面 111 序列的开始（称为不可重叠）的序列检测器的波形图。通过对比不难看出它们的区别。因为本题中没有给出明确的要求，本处将设计一个图 5-32（a）所示的序列检测器。

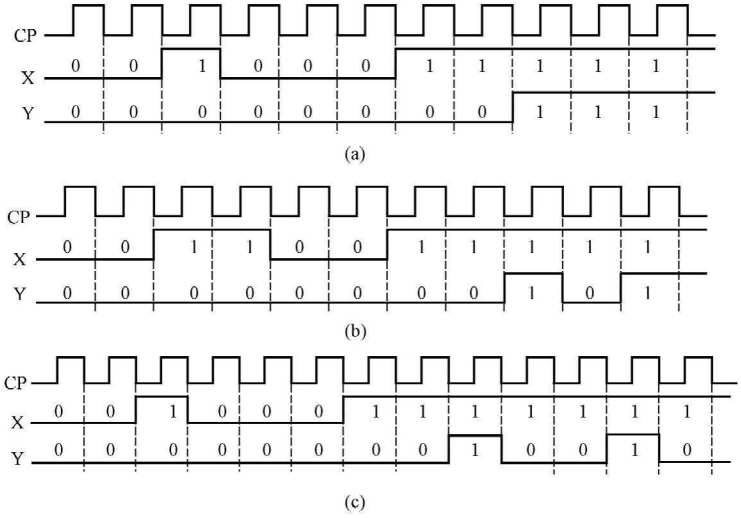


图 5-32 ［例 5-9］的 111 序列检测器在不同输入下的波形图

（a）输入为 00100011111，输出为 00000000111（前面序列最后的 11 作为后面序列的开始）；

（b）输入为 00110011111，输出为 00000000101（前面序列最后的 1 作为后面序列的开始）；

（c）输入为 0010001111111，输出为 0000000010010（前面的 1 不作为后面 111 序列的开始）

（1）进行逻辑抽象，建立原始状态图。这一步的注意力应放在状态图的正确性上，保证不遗漏任何必要的状态。至于状态是否多余，则不必过多考虑，因为即使有多余状态，在下步状态化简时也可以消去。所以，建立原始状态图（表）时，形成状态的原则是“宁多勿丢”。

首先要确定电路内部状态的转换规律，列出如图 5-33 所示的流程图，此图给出了在不同的输入 X 的作用下，电路状态变化规律及输出 Y 的值。图中有向线段上的标注反映 X 的输入序列，有向线段从初态出发指向次态，“/”下面的值表示输出 Y 的值。

如果电路处在状态 a，若输入 X=0，则状态进入 b 且输出 Y=0，若输入 X=1，则状态进入 c 且输出 Y=0，也就是说，状态 b 反映的是输入信号的历史是已经到来 1 个周期长的

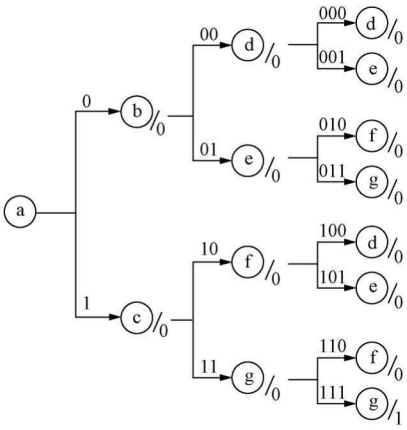


图 5-33 ［例 5-9］的状态转换流程图

0，状态 c 反映的是输入信号的历史是已经到来 1 个周期长的 1。

如果电路处在初始状态 c，若输入 X=0，则状态进入 f 且输出Y=0，若输入 X=1，则状态进入 g 且输出 Y=0，也就是说状态 f 反映的是输入信号的历史是已经连续输入 1 个时钟周期长的 0 和 1 个时钟周期长的 1，状态 g 反映的是输入信号的历史是已经连续输入 2 个时钟周期长的 2 个 1。

如在初始状态 g，若输入 X=0，则状态进入 f 且输出Y=0；若输入 X=1，则状态进入 g 且输出 Y=1，要找的“111”序列已经找到，输出 Y=1，离现在最远的那个周期的输入信号的历史已经没有必要记住，因为此处采用的是米里型电路设计，也就是说对于输入信号的记忆时间只需要 2 个时钟周期长。根据图 5-33 得到原始的状态表，如表 5-12 所示。

（2）状态化简。这一步主要是考察一下原始状态图中是否有多余状态。

观察表 5-12 的原始状态表，可以看出状态 b、d、f 等价，合并为一个状态，用 p 表示；c、e 等价，合并为一个状态，用 q 表示。得简化后的状态表如表 5-13 所示。

表 5-12 ［例 5-9］的原始状态表

S^{n+1}/Y \ X / S^n	0	1
a	b/0	c/0
b	d/0	e/0
c	f/0	g/0
d	d/0	e/0
e	f/0	g/0
f	d/0	e/0
g	f/0	g/1

表 5-13 ［例 5-9］的简化后的状态表

S^{n+1}/Y \ X / S^n	0	1
a	p/0	q/0
p	p/0	q/0
q	p/0	g/0
g	p/0	g/1

观察表 5-13，可以看出状态 a、p 等价，合并为一个状态，用 S_0 表示，q 用 S_1 表示，g 用 S_2 表示，得到最简的状态表如表 5-14 所示。

表 5-14 ［例 5-9］的最简的状态表

S^{n+1}/Y \ X / S^n	0	1
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_0/0$	$S_2/1$

对于本例，要设计的电路实际上是一个可重叠的 111 序列检测器，其序列长度为 3，根据题意可以直接得到最简的状态图。状态的概念在本章的开始给出了明确的定义。在某一时刻时序逻辑电路的输出信号不但与该时刻的输入信号有关，还与该时刻以前的输入信号的取值有关。该时刻以前的输入信号的曾经的取值表示为时序逻辑电路现在的状态。电路的状态反映的是输入信号的历史。111 序列检测器要检测的序列长度为 3，因为

采用米里设计，所以电路的状态数只要能够记忆 2 个时钟周期长的输入的历史就可以了。从前面的分析过程也能体会出这一点。设定电路的状态如下：

S_0：表示初始状态或收到一个时钟周期长的 0 时的状态或没有收到一个 1 时的状态；

S_1：收到一个时钟周期长的 1 后的状态；

S_2：连续收到两个时钟周期长的 11 后的状态。

本检测器只要三个状态就可以将 111 序列检测出来。不需要设置记忆输入 3 个时钟周期长的输入的历史，因为假如电路处在连续收到 11 后的状态 S_2，因为采用米里设计，这时输入端再来一个 1 时，输出就变为 1，表示要检测的序列已经到来，其他的电路可以使用这个信号了，所以采用米里设计只要记忆 2 个时钟周期长的输入的历史就可以了。如果要采用莫尔设计，就必须记忆 3 个时钟周期长的输入的历史，详细内容可以参考［例 5 - 10］。

最简状态图如图 5 - 34 所示。

（3）状态分配。首先需确定触发器的个数 n。因为有三个状态，所以最少需要两个触发器。两个触发器共有四种状态组合，“00”、“01”、“10”、“11”。只需要使用其中的三个状态即可。编码方式不同，得到的逻辑图也不同。编码方式的选择是否恰当，应以得到的电路最简，并且能够自启动为标准。这里取 $S_0=00$，$S_1=01$，$S_2=11$，据此可得到编码后的状态图如图 5 - 35 所示，其相应的状态表如表 5 - 15 所示。表 5 - 16 是［例 5 - 9］编码后的状态表的另一种画法，表 5 - 17 是求各触发器的驱动方程和输出方程的列表，为了更清晰起见，画出了表 5 - 18。表 5 - 18 将次态那列删掉了，很清楚地表明了两个触发器的驱动信号、输出信号与触发器现态的函数关系，这就是真值表。图 5 - 36 是求各触发器的驱动方程和进位输出方程的卡诺图。

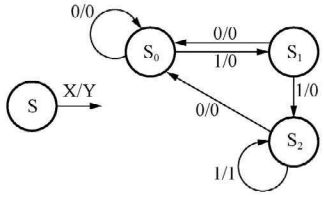


图 5 - 34　［例 5 - 9］的最简状态图

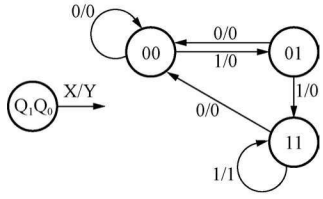


图 5 - 35　［例 5 - 9］编码后的状态图

表 5 - 15　［例 5 - 9］编码后的状态表

$Q_1^{n+1}Q_0^{n+1}/Y$ \ X ; $Q_1^nQ_0^n$	0	1
00	00/0	01/0
01	00/0	11/0
11	00/0	11/1
10	××/×	××/×

表 5 - 16　［例 5 - 9］编码后的状态表的另一种画法

输入变量	现态		次态		输出变量
X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	Y
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	×	×	×
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	×	×	×
1	1	1	1	1	1

（4）选定触发器类型，选用上升沿触发的 D 触发器。

（5）求出电路的驱动方程和输出方程。

表 5-17　　求各触发器的驱动方程和输出方程的列表

输入变量	现态		次态		驱动信号		输出变量	输入变量	现态		次态		驱动信号		输出变量
X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	D_1	D_0	Y	X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	D_1	D_0	Y
0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	0
0	1	0	×	×	×	×	×	1	1	0	×	×	×	×	×
0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1

表 5-18　　求各触发器的驱动方程和进位输出方程的真值表

输入变量	现态		驱动信号		输出变量	输入变量	现态		驱动信号		输出变量
X	Q_1^n	Q_0^n	D_1	D_0	Y	X	Q_1^n	Q_0^n	D_1	D_0	Y
0	0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	1	1	1	0
0	1	0	×	×	×	1	1	0	×	×	×
0	1	1	0	0	0	1	1	1	1	1	1

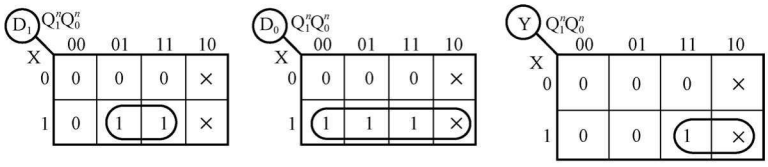


图 5-36　求各触发器的驱动方程和进位输出方程的卡诺图

根据图 5-36 得到驱动方程和输出方程为

$$D_0 = X$$

$$D_1 = XQ_0^n$$

$$Y = XQ_1^n$$

（6）根据驱动方程和输出方程，得到逻辑电路图，如图 5-37 所示。

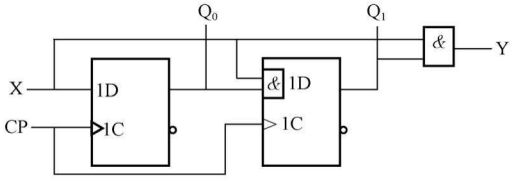


图 5-37　［例 5-9］的逻辑电路图

（7）给定输入信号 X 的值，得到波形图如图 5-38 所示，同时给出了两个触发器状态的波形。

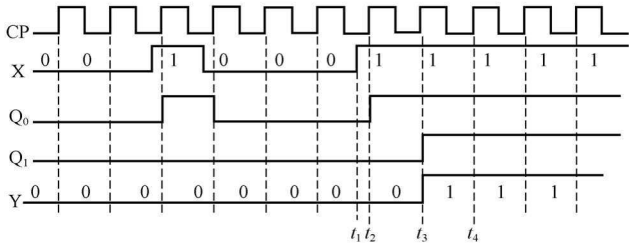


图 5 - 38　［例 5 - 9］的波形图

在 t_1 时刻，输入 X 变为 1，在 t_2 时刻（一个 CP 的上升沿），电路的状态变为 $Q_1Q_0=01$，表示电路已经收到了一个 1。t_2 时刻过后，输入 X 还是 1，那么在 t_3 时刻（一个 CP 的上升沿），电路的状态变为 $Q_1Q_0=11$，表示电路已经连续收到了两个 1。t_3 时刻过后，输入 X 还是 1，那么输出 Y 变为 1，表示电路已经连续收到了三个 1，并在 t_4 时刻（一个 CP 的上升沿），电路进入 $Q_1Q_0=11$ 状态，体现了米里型电路的特点。111 序列检测器比较特殊，要检测三个连续的 1，三个 1 之间没有变化，在波形上米里型电路的特点不易观察，后面［例 5 - 11］将以 101 序列检测器的设计再来体会这些概念。另外，在图 5 - 38 中，当第二个 1 到来时，输出 Y 就已经输出 1。但是这是 111 序列检测器，只能当第三个 1 到来时，Y 才能输出 1，解决这个问题的办法是将输出再经过 D 触发器输出。

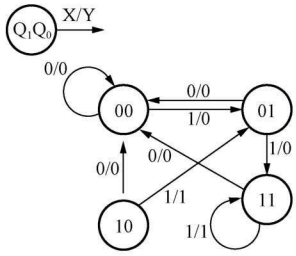


图 5 - 39　［例 5 - 9］的完整的状态转换图

（8）画全状态转换图，检查电路的自启动。完整的电路状态转换图如图 5 - 39 所示。假如电路处于无效状态 10，将它代入各触发器的驱动方程和特性方程得到：当X＝0 时，在 CP 作用下转为 00 状态，此时 Y＝0；而当 X＝1 时，在 CP 作用下，转为 01 状态，此时Y＝1。仔细分析可知，若电路由于干扰进入无效状态 10 时，再接收输入信号的工作是不正确的。在 10 状态下，若输入一个 1，电路应该输出 0，据此修改表 5 - 15，重新设计。因此，对序列检测器的设计，不要应用无关项，或者用 Moore 电路来设计。

【例 5 - 10】　用莫尔型电路实现［例 5 - 9］的 111 序列检测器。

解　（1）111 序列检测器要检测的序列长度为 3，因为采用莫尔型设计，所以电路的状态数要能够记忆 3 个时钟周期长的输入的历史。设定电路的状态如下：

S_0：表示初始状态或收到一个 0 时的状态；

S_1：收到一个 1 后的状态；

S_2：连续收到 11 后的状态；

S_3：连续收到 111 后的状态。

莫尔型电路设计需要三个状态，与米里型电路设计相比，多了一个状态。莫尔型电路需要记忆三个时钟周期长的历史，当连续收到 111 后电路进入 S_3 状态，在这个状态下输出 Y 变为 1，表示 111 序列已经到来。

（2）根据题意，得到状态图，分三种情况：①前一个 111 序列的最后一个 1 不作为后面新序列的开始，这种情况称为不可重叠，状态图如图 5 - 40 所示；②前一个 111 序列的最后两个 1 作为后面新序列的开始，这种情况称为可重叠，状态图如图 5 - 41 所示；③前一个

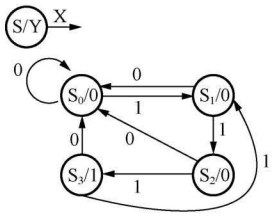


图 5 - 40 ［例 5 - 10］的不可重叠的 111 序列检测器的莫尔型设计状态图

111 的最后一个 1 作为后面新序列的开始，也称为可重叠。

（3）状态不需化简。

（4）状态赋值。下面对不可重叠的 111 序列检测器进行设计，可重叠的 111 序列检测器的设计读者可根据同样的方法自行完成。首先需确定触发器的个数 n。因为有四个状态，所以最少需要两个触发器。两个触发器共有四种状态组合，“00”、“01”、“10”、“11”。这里取 $S_0=00$，$S_1=01$，$S_2=10$，$S_3=11$，据此可得到编码后的状态图如图 5 - 42 所示。

（5）得到状态表如表 5 - 19 所示。

（6）表 5 - 20 是［例 5 - 10］的驱动信号和输出信号表。

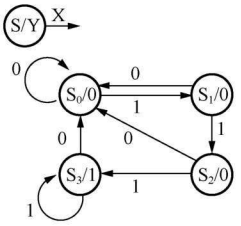


图 5 - 41 ［例 5 - 10］的可重叠的 111 序列检测器的莫尔型设计状态图

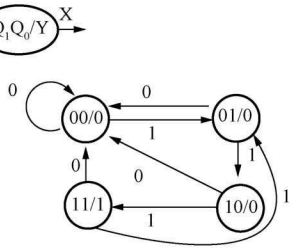


图 5 - 42 图5 - 40 的编码后的状态图

表 5 - 19 ［例 5 - 10］的状态表

输入变量	现态		次态		输出变量
X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	Y
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1

表 5 - 20 ［例 5 - 10］的驱动信号和输出信号表

输入变量	现态		次态		驱动信号		输出变量
X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	D_1	D_0	Y
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	1	1	1	1	0
1	1	1	0	1	0	1	1

（7）表 5 - 21 是［例 5 - 10］的求驱动信号的真值表，表 5 - 22 是［例 5 - 10］的求驱输出信号的真值表。因为这是莫尔型设计，所以两个表格分开画。

表 5 - 21　［例 5 - 10］的驱动信号表

输入变量	现　态		驱动信号	
X	Q_1^n	Q_0^n	D_1	D_0
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

表 5 - 22　［例 5 - 10］的输出信号表

现　态		输出变量
Q_1^n	Q_0^n	Y
0	0	0
0	1	0
1	0	0
1	1	1
0	0	0
0	1	0
1	0	0
1	1	1

（8）根据表 5 - 21 画出求驱动信号的卡诺图，如图 5 - 43 所示。根据表 5 - 22 可以直接写出输出信号的逻辑表达式

$$D_1 = X\overline{Q_1^n}Q_0^n + XQ_1^n\overline{Q_0^n}$$

$$D_0 = X\overline{Q_0^n} + XQ_1^n$$

$$Y = Q_0^n Q_1^n$$

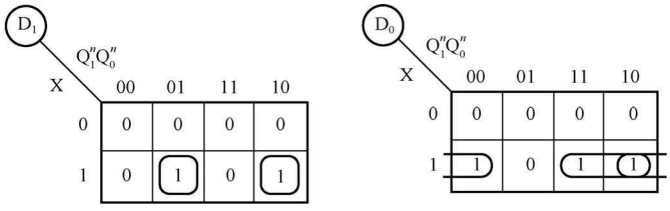


图 5 - 43　求驱动方程的卡诺图

（9）根据驱动方程和输出方程，得到逻辑电路图，如图 5 - 44 所示。

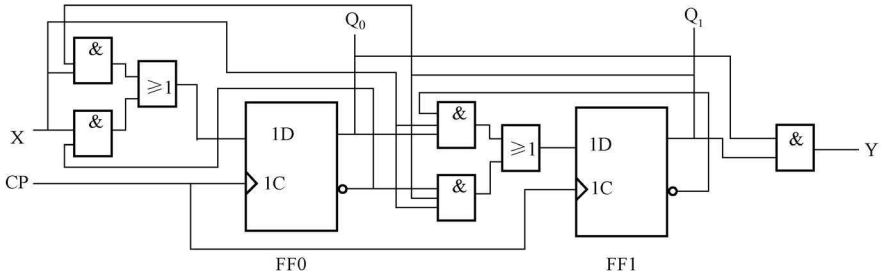


图 5 - 44　［例 5 - 10］的逻辑电路图

（10）不用检查能否自启动，因为两个触发器的四种状态组合全部被使用了。

（11）［例 5 - 10］的波形图如图 5 - 45 所示。

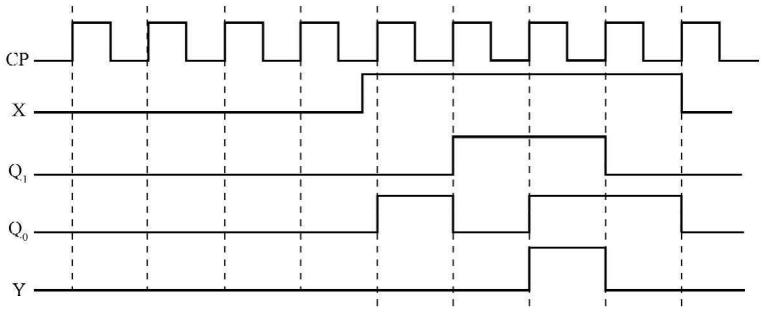


图 5-45 ［例 5-10］的波形图

【例 5-11】 设计一个 101 脉冲序列检测电路，X 为输入，Z 为输出，当检测到 X 连续输入 101 时，Z=1，否则，Z=0。X 输入的 101 序列中最后一个 1 不可以当作下一个序列的第一个 1，如 X=01010110100，则 Z=00010000100。

解 进行设计之前一定要想清楚使用米里型电路还是莫尔型电路设计，因为采用不同类型的电路设计结果将不同。本题将采用米里型设计一个前面的 1 不作为后面 101 序列的开始（称为不可重叠）的序列检测器。

（1）进行逻辑抽象，建立原始状态图。由题意要求可知，要设计的电路实际上是一个不可重叠的“101”序列检测器，其有效序列长度为 3。因此要求该电路能记忆收到的输入为 0，收到一个 1，连续收到 1、0 后的状态，故需设如下三个状态：

S_0：初始状态或没有收到 1 时的状态；

S_1：收到一个 1 后的状态；

S_2：连续收到 1、0 后的状态。

电路未收到 1 之前的初态为 S_0，此时，若 X 输入一个 0，电路仍维持原态不变，Z=0；若输入一个 1，就转入 S_1 态，Z=0。当电路在 S_1 状态下，X 输入一个 1，状态仍维持在 S_1，Z=0；反之，输入一个 0，电路状态转为 S_2，Z=0。当电路在 S_2 状态下，X 输入一个 0，电路回到 S_0 态，Z=0；若输入一个 1，则输出 Z=1，电路状态变为 S_0。电路的原始状态图如图 5-46 所示，对应的原始状态表如表 5-23 所示。

（2）状态化简。本例中已是最简状态。

表 5-23 ［例 5-11］的状态表

S^{n+1}/Z \ X / S^n	0	1
S_0	$S_0/0$	$S_1/0$
S_1	$S_2/0$	$S_1/0$
S_2	$S_0/0$	$S_0/1$

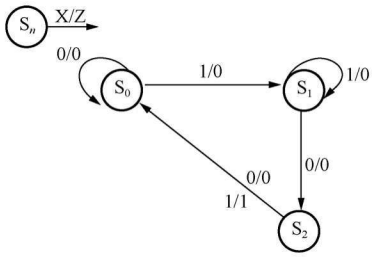


图 5-46 ［例 5-11］的状态图

（3）状态分配。首先需确定触发器的个数。因为有三个状态，所以最少需要两个触发器。两个触发器共有“00”、“01”、“10”、“11”四种状态组合，只需要使用其中的三个状态

即可。编码方式不同，得到的逻辑图也不同。编码方式的选择是否恰当，应以得到的电路最简，并且以能够自启动为标准。这里取 $S_0=00$，$S_1=01$，$S_2=10$，据此可得到编码后的状态转换图如图 5－47 所示，其相应的状态表如表 5－24 所示。

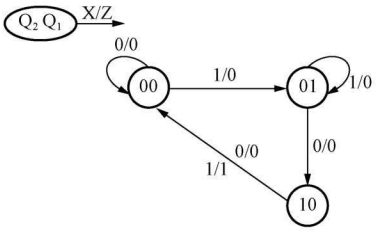


图 5－47　［例 5－11］编码后的状态图

表 5－24　［例 5－11］编码后的状态表

$Q_2^{n+1}Q_1^{n+1}/Z$ \ X ; $Q_2^nQ_1^n$	0	1
00	00/0	01/0
01	10/0	01/0
10	00/0	00/1
11	××/×	××/×

（4）选定触发器类型，求出电路的驱动方程和输出方程。若选定上升沿触发的维持阻塞 D 触发器来构成这个检测电路，则求电路的驱动方程和输出方程时，应根据状态图，列出状态表，再根据触发器的激励表，求得状态转换驱动表，如表 5－25 所示。将表 5－25 中的 Q_2^{n+1}、Q_1^{n+1} 两列删去，得到表 5－26。表 5－26 清晰地给出了触发器的驱动信号 D_2 和 D_1，及输出信号 Z_1，与输入 X、现态 Q_2^n、Q_1^n 的关系。

表 5－25　电路的状态转换驱动表

输入信号	现态		次态		驱动信号		输出信号
X	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	D_2	D_1	Z
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	1	×	×	×	×	×
1	0	0	0	1	0	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	1
1	1	1	×	×	×	×	×

表 5－26　表 5－25 删去次态后的电路状态转换驱动表

输入变量	现态		驱动信号		输出信号
X	Q_2^n	Q_1^n	D_2	D_1	Z
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	1	×	×	×
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	×	×	×

根据表 5－26，可画出函数 Z 和驱动信号 D_2、D_1 的卡诺图，分别如图 5－48、图 5－49 所示。

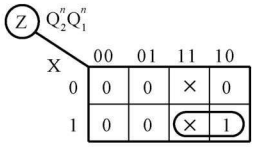


图 5－48　输出函数 Z 的卡诺图

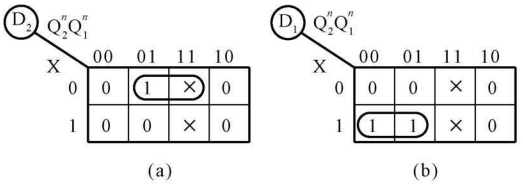


图 5－49　驱动信号 D_2、D_1 的卡诺图

（a）驱动信号 D_2；（b）驱动信号 D_1

经化简得到

$$Z = XQ_2^n$$

$$D_1 = X\overline{Q_2^n}$$

$$D_2 = \overline{X}Q_1^n$$

（5）根据输出方程和驱动方程画出逻辑电路图，如图 5 - 50 所示。

（6）对图 5 - 50 进行分析，画出完整的状态图，检查电路的自启动。假如电路处于无效状态 11，将它代入各触发器的驱动方程和特性方程得到：当 X＝0 时，在 CP 作用下转为 10 状态，此时 Z＝0；而当 X＝1 时，在 CP 作用下，转为 00 状态，此时 Z＝1。其完整的电路状态图如图 5 - 51 所示。观察图 5 - 51 可以看出，此电路并不能符合设计要求。在无效状态 11 下，此时输入一个 1，电路应该输出 0，次态进入 01；在无效状态 11 下，此时输入一个 0，电路应该输出 0，次态进入 00。据此修改表 5 - 24，重新设计。

（7）波形图如图 5 - 52 所示。

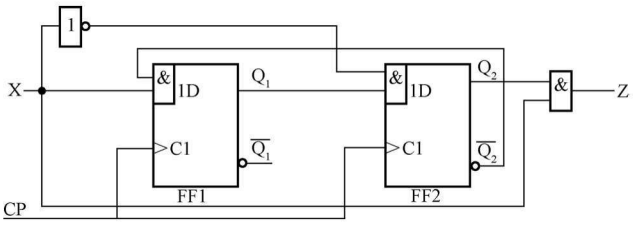


图 5 - 50 ［例 5 - 11］的逻辑电路图

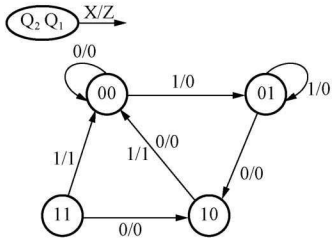


图 5 - 51 ［例 5 - 11］的完整的状态图

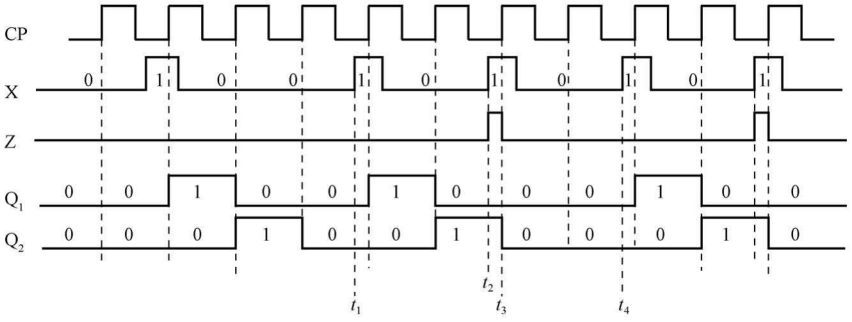


图 5 - 52 ［例 5 - 11］的波形图

【例 5 - 12】 用 JK 触发器设计一个自动售票机的时序逻辑电路。每次只允许投入一枚 5 角硬币或一枚 1 元硬币，累计投币 1 元时售票一张。若投币 5 角后再投币 1 元，则售票的同时找币 5 角。

解 （1）该时序逻辑电路有一个检测硬币面值的输入信号 X，两个输出信号 Y 和 Z 分别表示是否找钱的信号、是否售票的信号，框图如图 5 - 53 所示。规定 X 为 0 表示投入一枚 5 角硬币，X 为 1 表示投入一枚 1 元硬币；Y 为 0 表示不找钱，Y 为 1 表示找币 5 角；Z 为 0 表示不售票，Z 为 1 表示售票一张。采用米里型电路设计此自动售票机。

（2）根据题意，设 S_0 表示没有任何交易的等待状态，S_1 表示累计投币 5 角状态，S_2 表

示累计投币 1 元状态，S_3 表示累计投币 1 元 5 角状态，则原始状态图如图 5 - 54 所示。

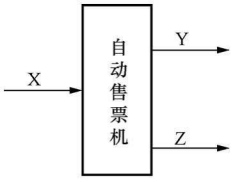


图 5 - 53　［例 5 - 12］自动售票机的框图

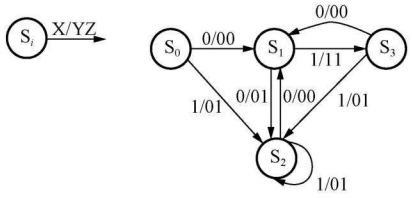


图 5 - 54　［例 5 - 12］的原始状态图

（3）状态化简。从原始状态图得原始状态表，如表 5 - 27 所示。

从原始状态表可观察出 S_0、S_2、S_3 是等价状态，合并为一个状态，并用 S_0 表示。简化后的状态表如表 5 - 28 所示。

表 5 - 27　［例 5 - 12］自动售票机的原始状态表

S^{n+1}/YZ \ X（S^n）	0	1
S_0	$S_1/00$	$S_2/01$
S_1	$S_2/01$	$S_3/11$
S_2	$S_1/00$	$S_2/01$
S_3	$S_1/00$	$S_2/01$

表 5 - 28　［例 5 - 12］自动售票机的简化状态表

S^{n+1}/YZ \ X（S^n）	0	1
S_0	$S_1/00$	$S_0/01$
S_1	$S_0/01$	$S_0/11$

简化后的状态表中只有两个状态，S_0 表示没有任何交易的等待状态，S_1 表示累计投币 5 角的状态。可以这样想，因为是采用米里型电路来设计此电路的，在 S_1 状态下，再投币 1 元，售票后电路应回到没有任何交易的等待状态，没有必要设立累计投币 1 元的状态，因为售票工作已经完成。在 S_0 状态下，投币 1 元，售票即可进行，售票后电路应回到没有任何交易的等待状态。

（4）状态编码。该电路共有两个状态，用一个 JK 触发器即可实现。触发器的状态用 Q 表示。编码后的状态表如表 5 - 29 所示。表 5 - 30 是编码后的状态表的另一种形式，表 5 - 31 是驱动信号和输出信号的表格。

（5）表 5 - 32 是求驱动信号和输出信号的表格。

表 5 - 29　编码后的状态表

Q^{n+1}/YZ \ X（Q^n）	0	1
0	1/00	0/01
1	0/01	0/11

表 5 - 30　编码后的状态表的另一种形式

X	Q^n	Q^{n+1}	Y	Z
0	0	1	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	1

表 5-31 驱动信号和输出信号列表

X	Q^n	Q^{n+1}	J	K	Y	Z
0	0	1	1	×	0	0
0	1	0	×	1	0	1
1	0	0	0	×	0	1
1	1	0	×	1	1	1

表 5-32 求驱动信号和输出信号的表格

X	Q^n	J	K	Y	Z
0	0	1	×	0	0
0	1	×	1	0	1
1	0	0	×	0	1
1	1	×	1	1	1

根据表 5-32，按照组合逻辑电路从真值表写表达式的方法写出表达式，并化简；或者用卡诺图化简，本处设有使用无相项，写出驱动方程 J、K 的表达式和输出方程 Y、Z 的表达式为

$$J = \overline{X}\,\overline{Q^n}$$

$$K = Q^n$$

$$Y = XQ^n$$

$$Z = X + Q^n$$

（6）根据驱动方程和输出方程，画出逻辑电路图如图 5-55 所示。

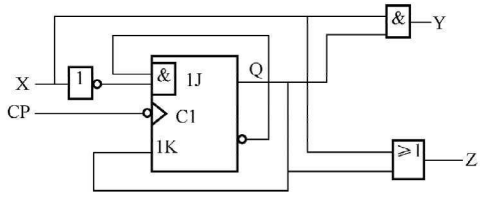


图 5-55 ［例 5-12］的逻辑电路图

第四节 计 数 器

计数器的基本功能是统计时钟脉冲的个数，是一种应用十分广泛的时序逻辑电路，除用于计数、分频外，还广泛用于数字测量、运算和控制。从小型数字仪表到大型数字电子计算机，计数器几乎无所不在，是任何现代数字系统中不可缺少的组成部分。

计数器的种类很多，从不同角度，有不同的分类方法：

（1）按计数容量可分为：二进制计数器和非二进制计数器。

（2）按数字的增减趋势可分为：加法计数器、减法计数器和可逆计数器。

（3）按计数器中各个触发器的时钟信号是否是同一个可分为：同步计数器和异步计数器。

一、二进制计数器

（一）二进制异步计数器

1. 二进制异步加法计数器

图 5-56 所示为由 4 个下降沿触发的 JK 触发器组成的 4 位二进制异步加法计数器的逻辑电路图。图中，JK 触发器都接成 T′触发器（即 J＝K＝1）。触发器 FF0 的时钟脉冲输入端接计数脉冲 CP，其他触发器的时钟脉冲输入端接相邻触发器的 Q 端。显然，这是一个异

步时序电路。

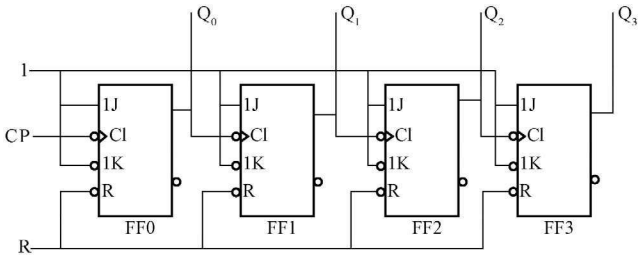


图 5 - 56　由 JK 触发器组成的 4 位异步二进制加法计数器的逻辑图

由于该电路的连线简单且规律性强，可以不用前面介绍的分析步骤进行分析，只需作简单的观察与分析就可逐级画出各输出端的波形图，再由波形图画出状态图，这种分析方法称为“观察法”。

该电路的波形图如图 5 - 57 所示。设开始工作前各触发器均处于 0 状态（由清零脉冲所置）。每输入一个计数脉冲 CP，FF0 就向相反的状态翻转一次；当 Q_0 由 1 变 0 时，就会向 FF1 触发器的时钟端输入一个负跳变脉冲，FF1 向相反的状态翻转一次；同样，当 Q_1 由 1 变 0 时，FF2 向相反的状态翻转一次；当 Q_2 由 1 变 0 时，FF3 向相反的状态翻转一次。

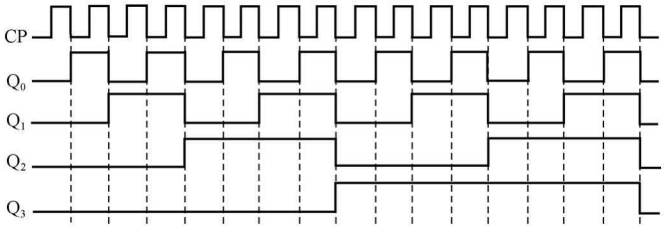


图 5 - 57　图 5 - 56 所示时序逻辑电路的波形图

根据图 5 - 57 所示的波形图，就可以画出电路的状态图如图 5 - 58 所示。该计数器有 0000～1111 共 16（$=2^4$）个状态，是 16 进制计数器或模 16 计数器，也可以称为 4 位二进制计数器。

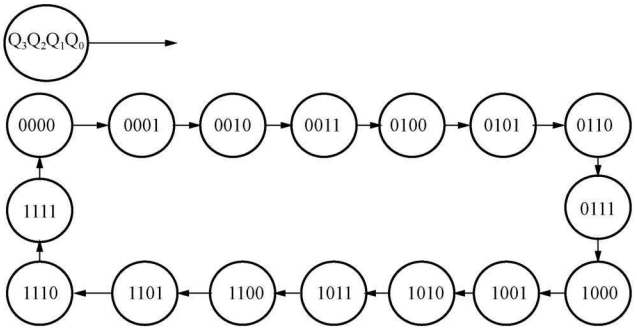


图 5 - 58　图 5 - 56 所示时序逻辑电路的状态图

从波形图可以看出，Q_0、Q_1、Q_2、Q_3 信号的周期分别是 CP 脉冲周期的 2 倍、4 倍、8 倍、16 倍。也就是说，Q_0、Q_1、Q_2、Q_3 的频率分别是 CP 脉冲的 1/2、1/4、1/8、1/16，即

分别对 CP 信号进行了二分频、四分频、八分频、十六分频。因而计数器也可作为分频器。

异步二进制计数器结构简单，通过改变级联触发器的个数，可以很方便地改变二进制计数器的位数。n 个触发器可构成 n 位二进制计数器、模 2^n 计数器或 2^n 分频器。

2. 二进制异步减法计数器

将图 5 - 56 所示电路中 FF1、FF2、FF3 的时钟脉冲输入端改接到相邻低位触发器的$\overline{Q}$端就可构成二进制异步减法计数器。其工作原理请读者自行分析。

图 5 - 59 所示是用 4 个上升沿触发的 D 触发器组成的 4 位异步二进制减法计数器的逻辑图。图中各触发器的 D 输入端与该触发器的$\overline{Q}$输出端相连（即 $D_i=\overline{Q_i^n}$），把 D 触发器转换成计数型触发器 T′。同时，各 Q 端又与相邻高 1 位触发器的时钟输入端相连。计数脉冲 CP 加至触发器 FF0 的时钟脉冲输入端。

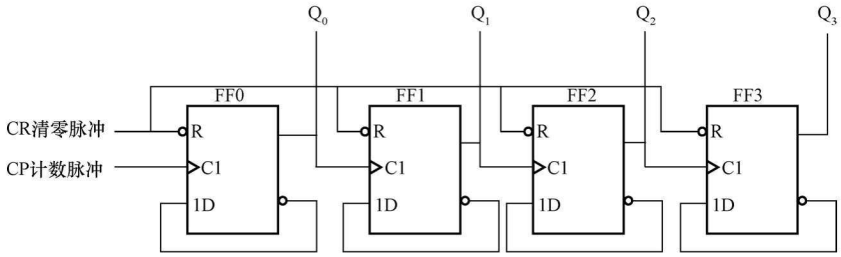


图 5 - 59　用 D 触发器组成的 4 位二进制异步减法计数器的逻辑图

同样，可以用“观察法”逐级画出各输出端的波形图，再由波形图画出状态图。该电路的波形图如图 5 - 60 所示，状态图如图 5 - 61 所示。图中各个触发器从初态 0000 开始，在第一个计数脉冲 CP 作用后，触发器 FF0 由 0 翻转为 1，此上升沿使 FF1 也由 0 翻转为 1，这个上升沿又使 FF2 也由 0 翻转为 1，此上升沿使 FF3 也由 0 翻转为 1，即计数器由 0000 变成了 1111 状态。此后，每输入一个计数脉冲，计数器的状态按二进制递减（减 1）。输入第 16 个计数脉冲后，又回到 0000 状态，完成一次循环。

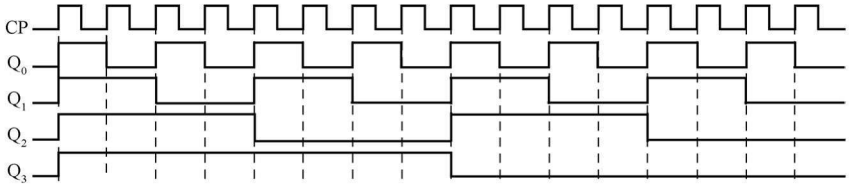


图 5 - 60　图 5 - 59 电路的波形图

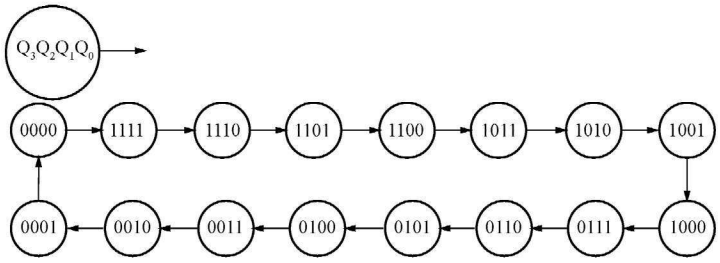


图 5 - 61　图 5 - 59 电路的状态图

从图 5 - 56 和图 5 - 59 可见，用 JK 触发器和 D 触发器都可以很方便地组成二进制异步计数器。方法是先将触发器都接成 T′触发器，然后根据加、减计数方式及触发器为上升沿还是下降沿触发来决定各触发器之间的连接方式。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相连。对于减计数器，各触发器间的连接方式则相反。

在二进制异步计数器中，高位触发器的状态翻转必须在相邻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，所以异步计数器的工作速度较低，又称这种类型的计数器为串行计数器或纹波计数器。为了提高计数速度，可采用同步计数器。

（二）二进制同步计数器

1. 二进制同步加法计数器

图 5 - 62 所示为由 4 个 JK 触发器组成的 4 位二进制同步加法计数器的逻辑电路图。用前面讲过的时序逻辑电路的分析方法对该电路进行分析。

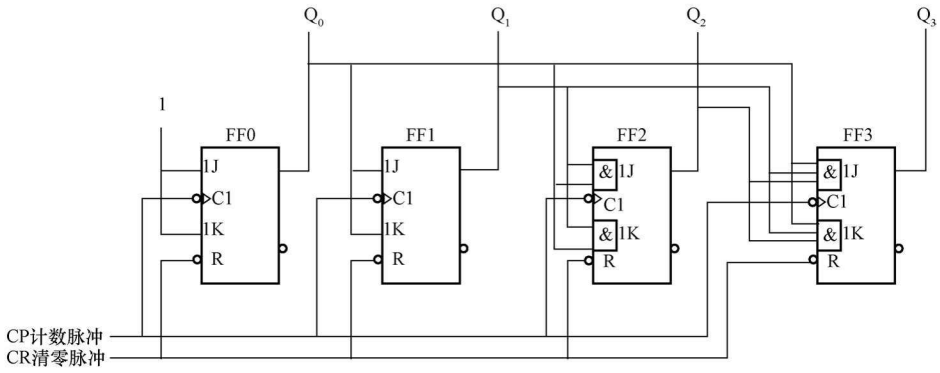


图 5 - 62　4 位二进制同步加法计数器的逻辑电路图

时钟脉冲输入端接同一计数脉冲 CP，显然，这是一个同步时序逻辑电路。

各触发器的驱动方程分别为

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0^n$$

$$J_2 = K_2 = Q_0^n Q_1^n$$

$$J_3 = K_3 = Q_0^n Q_1^n Q_2^n$$

将驱动方程代入特性方程 $Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n$ 得到电路的状态方程为

$$Q_0^{n+1} = \overline{Q_0^n}$$

$$Q_1^{n+1} = Q_0^n\,\overline{Q_1^n} + \overline{Q_0^n}\,Q_1^n = Q_0^n \oplus Q_1^n$$

$$Q_2^{n+1} = Q_0^n Q_1^n\,\overline{Q_2^n} + \overline{Q_0^n Q_1^n}\,Q_2^n = (Q_0^n Q_1^n) \oplus Q_2^n$$

$$Q_3^{n+1} = Q_0^n Q_1^n Q_2^n\,\overline{Q_3^n} + \overline{Q_0^n Q_1^n Q_2^n}\,Q_3^n = (Q_0^n Q_1^n Q_2^n) \oplus Q_3^n$$

表 5 - 33 为图 5 - 62 所示 4 位二进制同步加法计数器的状态表，图 6 - 63、图 6 - 64 分别为其状态图和波形图。

表 5-33　　图 5-62 所示 4 位二进制同步加法计数器的状态表

$Q_3^n Q_2^n Q_1^n Q_0^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$	$Q_3^n Q_2^n Q_1^n Q_0^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$
0 0 0 0	0 0 0 1	1 0 0 0	1 0 0 1
0 0 0 1	0 0 1 0	1 0 0 1	1 0 1 0
0 0 1 0	0 0 1 1	1 0 1 0	1 0 1 1
0 0 1 1	0 1 0 0	1 0 1 1	1 1 0 0
0 1 0 0	0 1 0 1	1 1 0 0	1 1 0 1
0 1 0 1	0 1 1 0	1 1 0 1	1 1 1 0
0 1 1 0	0 1 1 1	1 1 1 0	1 1 1 1
0 1 1 1	1 0 0 0	1 1 1 1	0 0 0 0

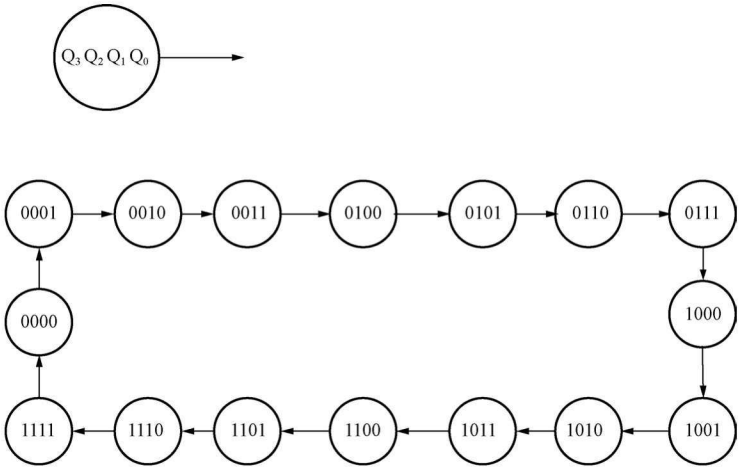


图 5-63　4 位二进制同步加法计数器的状态图

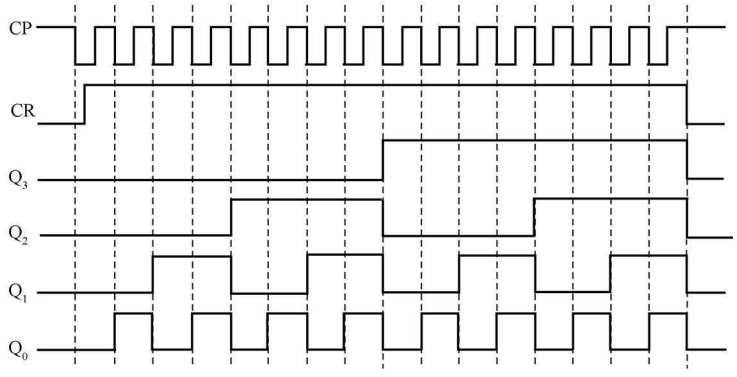


图 5-64　4 位二进制同步加法计数器的波形图

由于同步计数器的计数脉冲 CP 同时接到各位触发器的时钟脉冲输入端，当计数脉冲到来时，应该翻转的触发器同时翻转，所以速度比异步计数器高，但电路结构比异步计数器复杂。

2. 二进制同步减法计数器

将图 5 - 62 中各触发器的 J、K 端由原来接低位 Q 端改接为$\overline{Q}$端，就构成了 4 位二进制同步减法计数器。其逻辑电路图如图 5 - 65 所示。

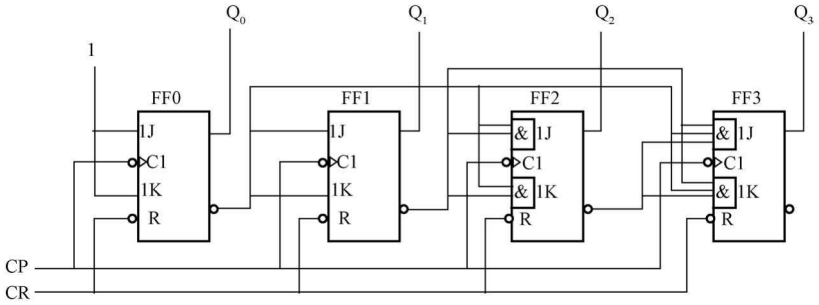


图 5 - 65　4 位二进制同步减法计数器逻辑电路图

各触发器的驱动方程为

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = \overline{Q_0^n}$$

$$J_2 = K_2 = \overline{Q_0^n}\,\overline{Q_1^n}$$

$$J_3 = K_3 = \overline{Q_0^n}\,\overline{Q_1^n}\,\overline{Q_2^n}$$

将驱动方程代入特性方程 $Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n$ 中，得到电路的状态方程为

$$Q_0^{n+1} = \overline{Q_0^n}$$

$$Q_1^{n+1} = \overline{Q_0^n}\,\overline{Q_1^n} + Q_0^n Q_1^n$$

$$Q_2^{n+1} = \overline{Q_0^n}\,\overline{Q_1^n}\,\overline{Q_2^n} + (Q_0^n + Q_1^n)Q_2^n$$

$$Q_3^{n+1} = \overline{Q_0^n}\,\overline{Q_1^n}\,\overline{Q_2^n}\,\overline{Q_3^n} + (Q_0^n + Q_1^n + Q_2^n)Q_3^n$$

4 位二进制同步减法计数器的状态表如表 5 - 34 所示，其状态图、波形图分别如图 5 - 66、图 5 - 67 所示。

表 5 - 34　　4 位二进制同步减法计数器的状态表

$Q_3^n Q_2^n Q_1^n Q_0^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$	$Q_3^n Q_2^n Q_1^n Q_0^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$
0 0 0 0	1 1 1 1	1 0 0 0	0 1 1 1
0 0 0 1	0 0 0 0	1 0 0 1	1 0 0 0
0 0 1 0	0 0 0 1	1 0 1 0	1 0 0 1
0 0 1 1	0 0 1 0	1 0 1 1	1 0 1 0
0 1 0 0	0 0 1 1	1 1 0 0	1 0 1 1
0 1 0 1	0 1 0 0	1 1 0 1	1 1 0 0
0 1 1 0	0 1 0 1	1 1 1 0	1 1 0 1
0 1 1 1	0 1 1 0	1 1 1 1	1 1 1 0

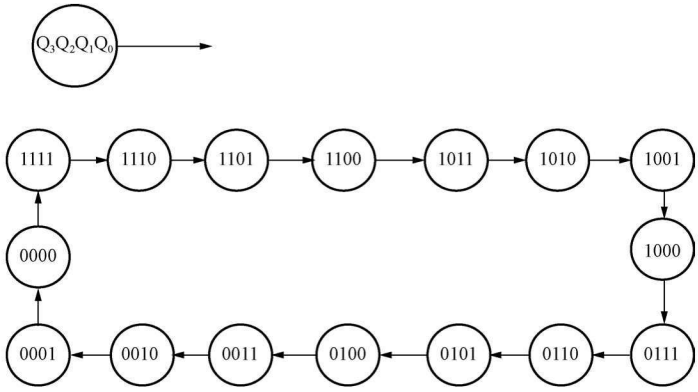


图 5 - 66　4 位二进制同步减法计数器的状态图

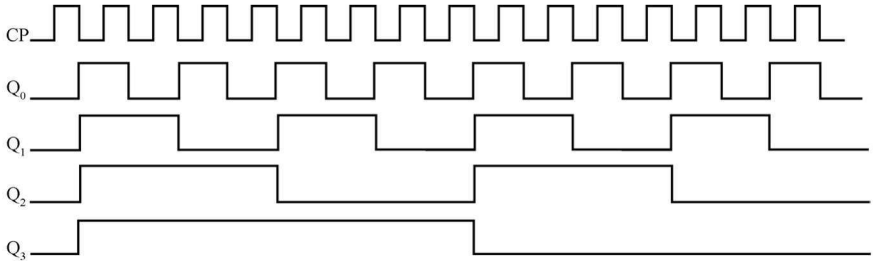


图 5 - 67　4 位二进制同步减法计数器的波形图

3. 二进制同步可逆计数器

既能作加计数，又能作减计数的计数器称为可逆计数器。将前面介绍的 4 位二进制同步加法计数器和减法计数器合并起来，并引入一加/减控制信号 X 便构成 4 位二进制同步可逆计数器。其逻辑电路图如图 5 - 68 所示。

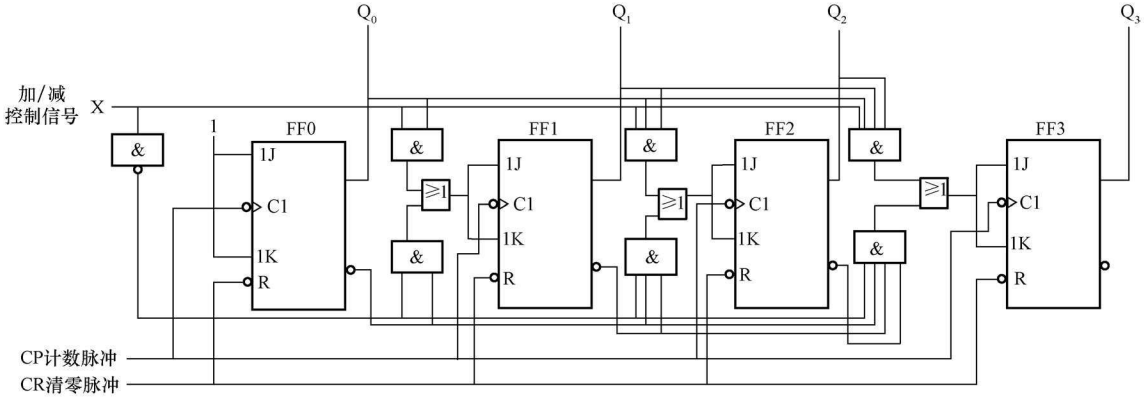


图 5 - 68　4 位二进制可逆计数器的逻辑电路图

由图可知，各触发器的驱动方程为

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = XQ_0^n + \overline{X}\,\overline{Q_0^n}$$

$$J_2 = K_2 = XQ_0^nQ_1^n + \overline{X}\,\overline{Q_0^n}\,\overline{Q_1^n}$$

$$J_3 = K_3 = XQ_0^nQ_1^nQ_2^n + \overline{X}\,\overline{Q_0^n}\,\overline{Q_1^n}\,\overline{Q_2^n}$$

将驱动方程代入特性方程得到状态方程为

$$Q_0^{n+1} = \overline{Q_0^n}$$

$$Q_1^{n+1} = (XQ_0^n + \overline{X}\,\overline{Q_0^n}) \oplus Q_1^n$$

$$Q_2^{n+1} = (XQ_0^nQ_1^n + \overline{X}\,\overline{Q_0^n}\,\overline{Q_1^n}) \oplus Q_2^n$$

$$Q_3^{n+1} = (XQ_0^nQ_1^nQ_2^n + \overline{X}\,\overline{Q_0^n}\,\overline{Q_1^n}\,\overline{Q_2^n}) \oplus Q_3^n$$

根据状态方程得到状态表如表 5 - 35 所示。

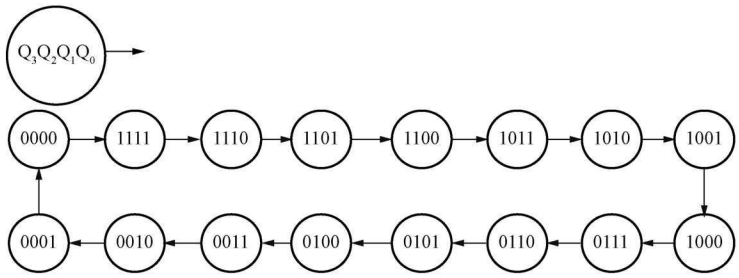
图 5 - 69 所示为 4 位二进制可逆计数器的状态图。由图可见，当控制信号 X=1 时，作加法计数，当控制信号 X=0 时，作减法计数，实现了可逆计数器的功能。

表 5 - 35　　图 5 - 68 的状态表

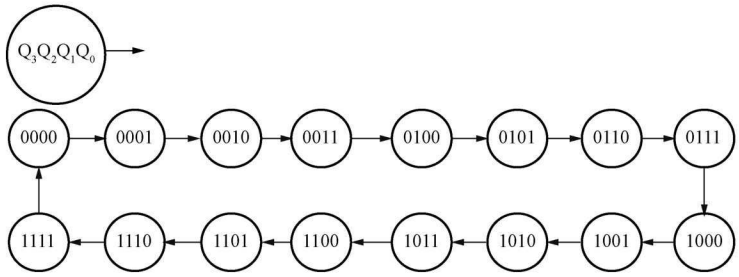
输入	现态				次态				输入	现态				次态			
X	Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	X	Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	1	1
0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	0	0
0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	1	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0
0	0	1	1	0	0	1	0	1	1	0	1	1	0	0	1	1	1
0	0	1	1	1	0	1	1	0	1	0	1	1	1	1	0	0	0
0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	1
0	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	1	0
0	1	0	1	0	1	0	0	1	1	1	0	1	0	1	0	1	1
0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	0
0	1	1	0	0	1	0	1	1	1	1	1	0	0	1	1	0	1
0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	1	0
0	1	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0

二、非二进制计数器

N 进制计数器又称模 N 计数器，当 $N=2^n$ 时，就是前面讨论的 n 位二进制计数器；当 $N\neq2^n$ 时，为非二进制计数器。非二进制计数器中最常用的是十进制计数器。下面讨论 8421BCD 码十进制计数器。



(a)



(b)

图 5 - 69　4 位二进制可逆计数器的状态图

（a）当 X=0 时的减计数的状态图；（b）当 X=1 时的加计数的状态图

（一）8421BCD 码同步十进制加法计数器

图 5 - 70 所示为由 4 个下降沿触发的 JK 触发器组成的 8421BCD 码同步十进制加法计数器的逻辑电路图。

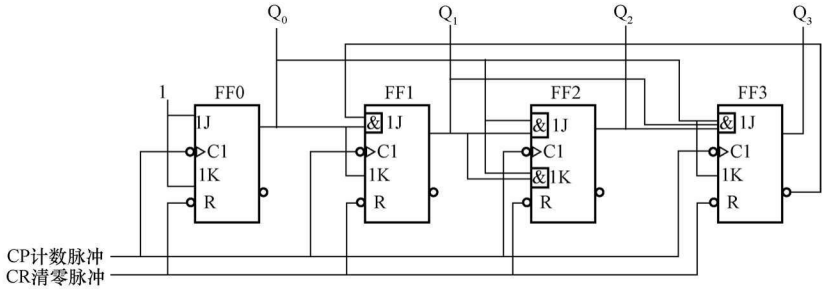


图 5 - 70　8421BCD 码同步十进制加法计数器的逻辑电路图

用前面介绍的同步时序逻辑电路分析方法对该电路进行分析。

（1）写出驱动方程，即

$$J_0 = 1, K_0 = 1$$

$$J_1 = \overline{Q_3^n}Q_0^n, K_1 = Q_0^n$$

$$J_2 = Q_1^nQ_0^n, K_2 = Q_1^nQ_0^n$$

$$J_3 = Q_2^nQ_1^nQ_0^n, K_3 = Q_0^n$$

（2）写出 JK 触发器的特性方程 $Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n$，然后将各驱动方程代入 JK 触发器

的特性方程，得各触发器的次态方程为

$$Q_0^{n+1} = J_0\overline{Q_0^n} + \overline{K_0}Q_0^n = \overline{Q_0^n}$$

$$Q_1^{n+1} = J_1\overline{Q_1^n} + \overline{K_1}Q_1^n = \overline{Q_3^n}\,Q_0^n\,\overline{Q_1^n} + \overline{Q_0^n}Q_1^n$$

$$Q_2^{n+1} = J_2\overline{Q_2^n} + \overline{K_2}Q_2^n = Q_1^nQ_0^n\,\overline{Q_2^n} + \overline{Q_1^n}\,\overline{Q_0^n}Q_2^n$$

$$Q_3^{n+1} = J_3\overline{Q_3^n} + \overline{K_3}Q_3^n = Q_2^nQ_1^nQ_0^n\,\overline{Q_3^n} + \overline{Q_0^n}Q_3^n$$

（3）作状态表。如表 5 - 36 所示。

表 5 - 36　　图 5 - 70　电路的状态表

现　态				次　态				现　态				次　态			
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1
0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0
0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	1
0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1	1	1	0	0	1	1	0	1
1	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	1
1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	0

（4）作状态图及时序图。根据状态转换表作出电路的状态图如图 5 - 71 所示，时序图如图 5 - 72 所示。由状态表、状态图或时序图可见，该电路为 8421BCD 码十进制加法计数器，电路能够自启动。

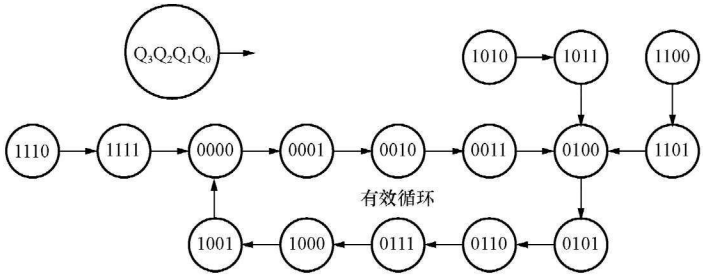


图 5 - 71　图 5 - 70 的状态图

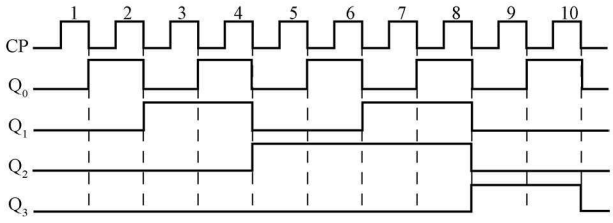


图 5 - 72　图 5 - 70 的时序图

（二）8421BCD 码异步十进制加法计数器

图 5 - 73 所示为由 4 个下降沿触发的 JK 触发器组成的 8421BCD 码异步十进制加法计数器的逻辑电路图。

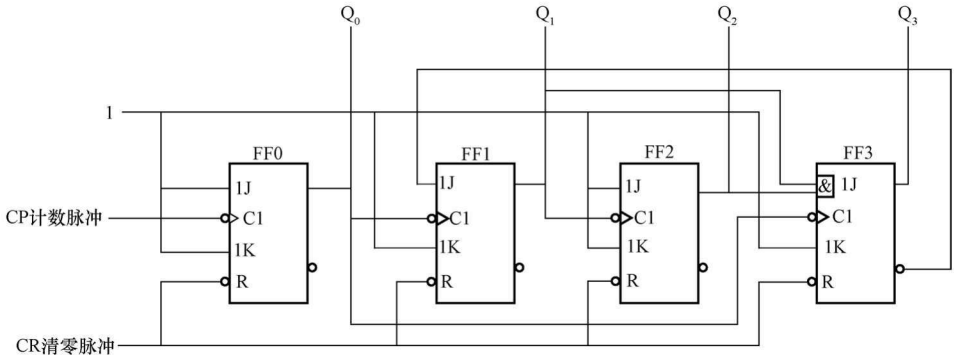


图 5 - 73　8421BCD 码异步十进制加法计数器的逻辑电路图

用前面介绍的异步时序逻辑电路分析方法对该电路进行分析。

（1）写出各逻辑方程式。

1）时钟方程：

$CP_0 = CP$（时钟脉冲源的下降沿触发）

$CP_1 = Q_0$（当 FF0 的 Q_0 由 1→0 时，Q_1 才可能改变状态，否则 Q_1 将保持原状态不变）

$CP_2 = Q_1$（当 FF1 的 Q_1 由 1→0 时，Q_2 才可能改变状态，否则 Q_2 将保持原状态不变）

$CP_3 = Q_0$（当 FF0 的 Q_0 由 1→0 时，Q_3 才可能改变状态，否则 Q_3 将保持原状态不变）

2）各触发器的驱动方程为

$$J_0 = 1, K_0 = 1$$

$$J_1 = \overline{Q_3^n}, K_1 = 1$$

$$J_2 = 1, K_2 = 1$$

$$J_3 = Q_2^n Q_1^n, K_3 = 1$$

（2）将各驱动方程代入 JK 触发器的特性方程，得各触发器的次态方程为

$$Q_0^{n+1} = J_0\overline{Q_0^n} + \overline{K_0}Q_0^n = \overline{Q_0^n}\text{（CP 由 } 1 \to 0 \text{ 时此式有效）}$$

$$Q_1^{n+1} = J_1\overline{Q_1^n} + \overline{K_1}Q_1^n = \overline{Q_3^n}\,\overline{Q_1^n}\text{（}Q_0\text{ 由 } 1 \to 0 \text{ 时此式有效）}$$

$$Q_2^{n+1} = J_2\overline{Q_2^n} + \overline{K_2}Q_2^n = \overline{Q_2^n}\text{（}Q_1\text{ 由 } 1 \to 0 \text{ 时此式有效）}$$

$$Q_3^{n+1} = J_3\overline{Q_3^n} + \overline{K_3}Q_3^n = Q_2^n Q_1^n\,\overline{Q_3^n}\text{（}Q_0\text{ 由 } 1 \to 0 \text{ 时此式有效）}$$

（3）作状态转换表，如表 5 - 37 所示。

表 5 - 37　　图 5 - 73 电路的状态表

现　态				次　态				时　钟　脉　冲			
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	CP_3	CP_2	CP_1	CP_0
0	0	0	0	0	0	0	1	0	0	0	↓
0	0	0	1	0	0	1	0	↓	0	↓	↓
0	0	1	0	0	0	1	1	0	0	0	↓

续表

现　态				次　态				时　钟　脉　冲			
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	CP_3	CP_2	CP_1	CP_0
0	0	1	1	0	1	0	0	↓	↓	↓	↓
0	1	0	0	0	1	0	1	0	0	0	↓
0	1	0	1	0	1	1	0	↓	0	↓	↓
0	1	1	0	0	1	1	1	0	0	0	↓
0	1	1	1	1	0	0	0	↓	↓	↓	↓
1	0	0	0	1	0	0	1	0	0	0	↓
1	0	0	1	0	0	0	0	↓	0	↓	↓
1	0	1	0	1	0	1	1	0	0	0	↓
1	0	1	1	0	1	0	0	↓	↓	↓	↓
1	1	0	0	1	1	0	1	0	0	0	↓
1	1	0	1	0	1	0	0	↓	0	↓	↓
1	1	1	0	1	1	1	1	0	0	0	↓
1	1	1	1	0	0	0	0	↓	↓	↓	↓

（4）作状态图及时序图。根据状态表作出电路的状态图如图 5 - 74 所示，时序图如图 5 - 75所示。由状态表、状态图或时序图可见，该电路为 8421BCD 码十进制加法计数器。

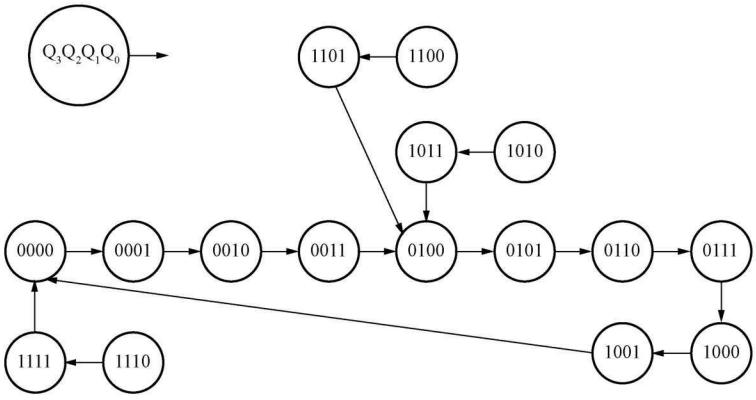


图 5 - 74　图 5 - 73 的状态图

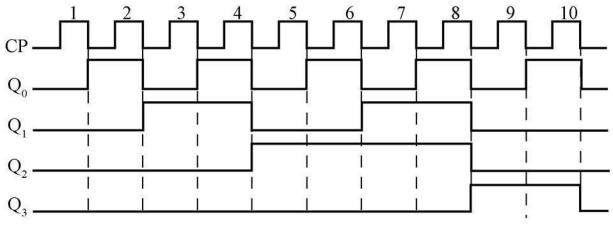


图 5 - 75　图 5 - 73 的时序图

第五节 常用中规模计数器芯片及应用

一、常用中规模计数器芯片

（一）4 位二进制同步加法计数器芯片 74X161

74X161 是 4 位二进制同步加法计数器集成模块。图 5-76 是加了引脚名字的 74X161 的逻辑符号，图 5-77 是其内部逻辑电路图。74X161 芯片共有 16 个引脚。8 脚和 16 脚分别是地和电源。74X161 芯片内部有四个触发器，11 脚 Q_3、12 脚 Q_2、13 脚 Q_1、14 脚 Q_0 是这四个触发器的状态信号。2 脚 CP 是时钟脉冲信号。一般情况下，芯片生产厂家为了扩大芯片的使用范围，在实现芯片基本功能的基础上，增加了一些控制功能。1 脚 CR 是清零端，可以实现芯片的置 $Q_3Q_2Q_1Q_0=0000$ 状态的功能。9 脚 LD 是预置数控制端，与 6 脚 D_3、5 脚 D_2、4 脚 D_1、3 脚 D_0 预置数据输入端一起共同完成芯片的置 $Q_3Q_2Q_1Q_0$ 为任意状态的功能，CR 和 LD 是为了给芯片置初态的。7 脚 EP 和 10 脚 ET 是芯片的使能端，控制芯片是否处于计数状态。15 脚 RCO 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。

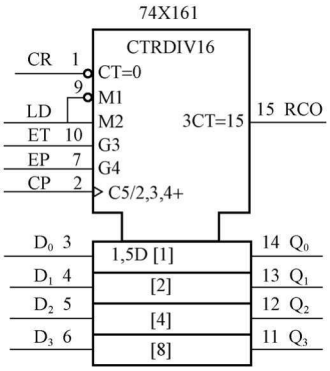


图 5-76 加了引脚名字的 74X161 的逻辑符号

中规模集成芯片的种类很多，功能也千差万别，为了更好地记忆这些中规模集成芯片的功能，不至于在使用时频繁地去查阅这些芯片的使用手册，在本书中采用了国家标准 GB/T4728.12——1996《电气简图用图形符号——二进制逻辑元件》所规定的原则绘制的符号。该标准与国际电工委员会标准 IEC617——12 是一致的。使用该标准绘出的符号可以做到少用或不用其他参考文件就能确定所描述的逻辑电路的功能。

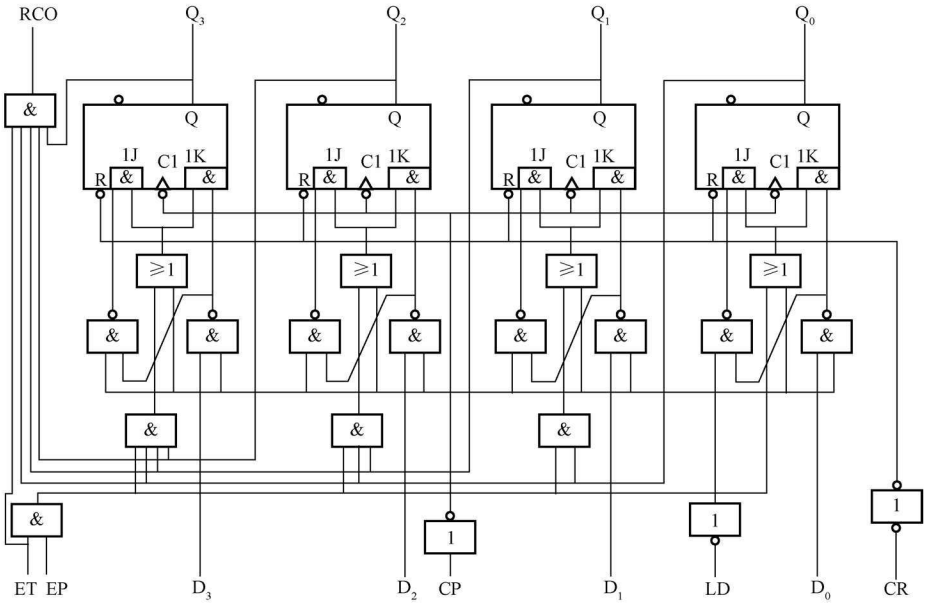


图 5-77 74X161 的内部逻辑电路图

表 5－38 为 74X161 的功能表，从该表的第一行可以看出 CR 清零信号的功能，CR 为 0，表明为低电平有效，在 CP 处为“×”，表明 CR 为异步清零。注意此处的“异步清零”中的“异步”的含义与前面所讲的“异步计数器”中的“异步”的含义不同。前面已经知道，“异步计数器”中的“异步”是指计数器中的各个触发器的时钟信号之间的关系，“异步”是指各个触发器的时钟信号不是同一个信号。而此处的“异步清零”中的“异步”是指 CR 与 CP 之间的关系，当 CR 有效的低电平信号到来时 $Q_3Q_2Q_1Q_0$ 就被置为 0000，不需要 CP 有效的时钟沿的配合。CR 的异步清零功能也可以从图 5－77 的 74X161 的内部逻辑电路图观察出来，在图 5－77 中，右下脚的 CR 信号直接连接到了各个触发器的直接清零端。CR 信号具有最高的优先级别，从表中可以看出其他的输入信号全部为“×”。

表 5－38　　74X161 的功能表

清零	预置数	使能		时钟	预置数据输入				输出				工作模式
CR	LD	ET	EP	CP	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	
0	×	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	0	×	×	↑	D	C	B	A	D	C	B	A	同步置数
1	1	0	×	×	×	×	×	×	保持				数据保持
1	1	×	0	×	×	×	×	×	保持				数据保持
1	1	1	1	↑	×	×	×	×	计数				加法计数

从表 5－38 的第二行可以看出 LD 预置数信号的功能，在 CR 清零信号无效的前提下，LD 为 0，表明为低电平有效，在 CP 处为↑，表明 LD 为同步预置数。注意此处的“同步预置数”中“同步”的含义与前面所讲的“同步计数器”中的“同步”的含义不同。“同步计数器”中的“同步”是指计数器中的各个触发器的时钟信号之间的关系，“同步”是指各个触发器的时钟信号是同一个信号。而此处的“同步预置数”中的“同步”是指 LD 与 CP 之间的关系，LD 与 CP 必须共同作用才能完成置数的功能。当 LD 有效的低电平信号到来时 $Q_3Q_2Q_1Q_0$ 就被置为 $D_3D_2D_1D_0$ 送来的数据 DCBA，但是这种置数需要 CP 有效的时钟沿（上升沿）的配合。LD 的同步预置数功能也可以从图 5－77 的 74X161 的内部逻辑电路图观察出来。在图 5－77 中，LD 信号与 $D_3D_2D_1D_0$ 经过几个门之后连接到了触发器的 JK 输入端，JK 输入端的信号需要时钟沿的配合才能改变触发器的状态。

从表 5－38 的第三行和第四行可以看出，在清零信号和预置数信号无效的前提下，使能端 EP 和 ET 不同时为 1 时，74X161 处于保持状态，即虽然有效的时钟沿源源不断地到来，但是触发器的状态并不改变。

从表 5－38 的第五行可以看出，在清零信号和预置数信号无效的前提下，使能端 EP 和 ET 同时为 1（有效）时，计数器处于正常的计数功能。从表 5－38 中只能知道计数规律是“加”计数，但不能看出更具体的计数规律，可以从图 5－78 所示的 74X161 的时序图中观察出它的计数规律。74X161 的状态图如图 5－79 所示，在该图中要特别注意 $Q_3Q_2Q_1Q_0$ 的顺序。

从图 5－77 内部逻辑电路图和图 5－78 时序图中也可以观察 RCO 的规律。$RCO=Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot ET$。当 $Q_3=1$、$Q_2=1$、$Q_1=1$、$Q_0=1$、ET＝1 时 RCO＝1。

在图 5－78 中，首先加入一清零信号 CR＝0，使各触发器的状态为 0，即计数器清零。CR 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 5 - 78 中为 DCBA＝1100），这就是预置操作。接着是 EP＝ET＝1，在此期间 74X161 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 CP 负跳变前的状态不变，图 5 - 78 中为 $Q_3Q_2Q_1Q_0$＝0010，RCO＝0。

在了解了 74X161 的基本功能之后，下面对 74X161 的逻辑符号进行必要的解释。

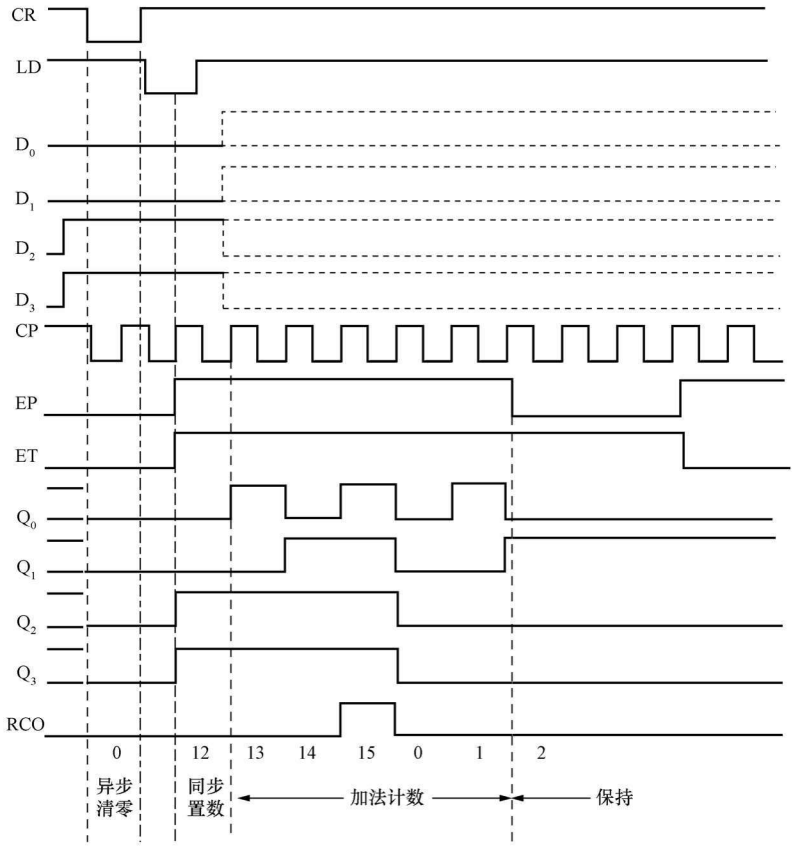


图 5 - 78　74X161 的时序图

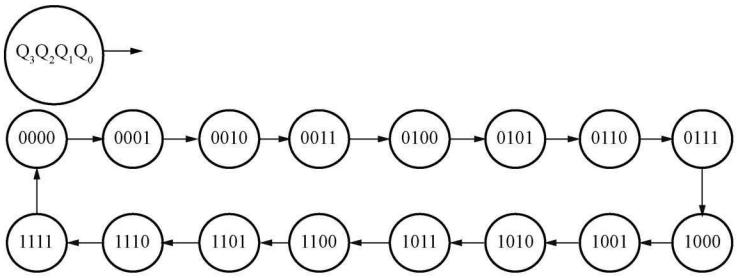


图 5 - 79　74X161 的状态图

在开始了解国标符号之前，觉得国标符号非常复杂，不易使用。其实当我们对国标符号有了一定的理解时，会觉得非常便于记忆不同集成芯片的功能。在国标符号中，具有同类功

能的信号一般标注在一起，输入信号全部标注在长方形框的左边，输出信号全部标注在长方形框的右边。总定性符号 CTRDIV16 表明是模 16 计数器。CTR 是 COUNTER 的缩写，表示是计数器。我们知道计数器也可以当作分频器使用，DIV 是 DIVISION 的缩写，表示是分频器。16 表示计数容量是 16，即模是 16，具有 16 个状态。总定性符号 CTRDIV16 一般写在长方形框的上部。国标符号的引脚只标有引脚的编号，并不包括引脚的名字，引脚的名字是本书作者为了便于解释芯片的使用而标上去的，不包括在标准之内。芯片的 1 脚是清零信号输入端，在长方形框的左上部，1 表示是 1 号脚，本书给 1 脚标上了 CR，CR 是 CLEAR 的缩写，表示清除的意思。紧挨着 1 脚，在长方形框的内部，写有“CT＝0”，CT 是 CONTENT（内容）的缩写，表示的内容是 $Q_3Q_2Q_1Q_0$ 的值，“CT＝0”即 $Q_3Q_2Q_1Q_0$＝0000，小圆圈表示是低电平有效（在正逻辑体制下，本书如不做特别说明，均使用正逻辑）。“CT＝0”的前面没有任何数字，可以说明是异步清零。

芯片的 9 脚是预置数信号输入端，在长方形框的左上部，9 表示是 9 号脚。本书给 9 脚标上了 LD 的名字，LD 是 LOAD 的缩写，表示装载的意思。紧挨着 9 脚，在长方形框的内部，写有“M1”、“M2”，M 是 MODE 的缩写，表示是模式的意思。9 脚 LD 在不同输入电平下定义为不同的工作方式，输入低电平定义为模式 1，输入高电平定义为模式 2。“M1”、“M2”中的“1”、“2”表示给它们编了序号“1”、“2”。

芯片的 10 脚和 7 脚是使能信号输入端，没有小圆圈表示是高电平有效。本书给 10 脚标上了 ET，在长方形框的内部，写有“G3”，G 是“与关联”的意思，表示其他信号能否起作用与这个信号有关系，关系是当此信号为 1 时才起作用，即与的关系。按“M1”、“M2”中的序号“1”、“2”排下来，“G3”中的“3”表示给 10 脚编了一个序号“3”。同理，给 7 脚标上了 EP，在长方形框的内部，写有“G4”，“G4”中的“4”表示给 7 脚编了一个序号“4”。

芯片的 2 脚是时钟脉冲输入端，在长方形框内部的“＞”表示是动态输入端，即时钟脉冲输入端，外部没有小圆圈，表示是上升沿有效。本书给 2 脚标上了 CP，在长方形框的内部“＞”的右面写有“C5/2，3，4＋”，其中，“C”表示控制；“5”表示给 2 脚时钟脉冲输入端编了一个序号“5”；“/”、“2，3，4”表示当 M2（LD）＝1、G3（ET）＝1、G4（EP）＝1 时进行计数；“＋”表示是“加”计数，“加”的概念在前面已经有详细的解释，注意理解其实质。

芯片的 15 脚是进位输出端，本书给 15 脚标上了 RCO，在长方形框的内部，写有“3CT＝15”。“3CT＝15”中的“3”表示与序号为“3”的信号有关，序号为“3”的信号是 10 脚 ET 使能端，从内部逻辑电路图 5－77 已经知道，RCO 是 5 个信号的与，其中包括 ET 信号，即 RCO＝$Q_3Q_2Q_1Q_0$ET；“CT＝15”指当内容为 15，即 $Q_3Q_2Q_1Q_0$＝1111 时芯片的 15 脚 RCO 输出为 1。

以上所讲的这些信号在国标符号中均位于上部的公共控制框内。

芯片的 3 脚、4 脚、5 脚和 6 脚是与芯片的 9 脚（预置数信号输入端 LD）配合使用的数据输入端，用来给计数器置初态，本书中分别给 3 脚、4 脚、5 脚和 6 脚标上了 D_0、D_1、D_2 和 D_3 的名字。紧挨着 3 脚，在长方形框的内部，写有“1，5D”，表明与序号为“1”、“5”的信号有关。序号为“1”的信号是 9 脚 LD 输入为低电平时定义的模式 1，序号为“5”的信号是 2 脚时钟脉冲输入端，为同步预置数；“1，5D”中的“D”是数据的意思。虽然紧挨着 4 脚、5 脚和 6 脚的长方形框的内部没有写“1，5D”，但是也有 3 脚内部的“1，5D”的功能。

芯片的 11 脚、12 脚、13 脚和 14 脚是触发器的状态信号，在本书中分别给 11 脚、12 脚、

13 脚和 14 脚标上了 Q_3、Q_2、Q_1 和 Q_0。长方形框内部的［1］、［2］、［4］和［8］表示输出端 Q_0、Q_1、Q_2 和 Q_3 的权依次为 1、2、4 和 8。注意电路的状态是用触发器的状态的组合表示的，只是 0、1 的组合而已，本身没有数的概念，没有位权。但是为了更方便地描述计数器的功能，人为地给触发器赋予了一定的位权，这就是所谓的“加”计数和“减”计数的概念。

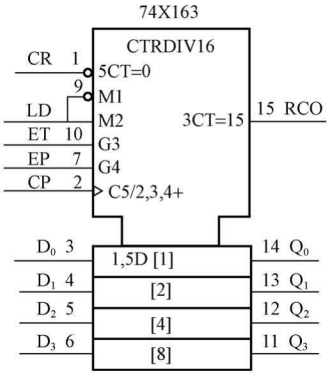


图 5－80　加了引脚名字的 74X163 的逻辑符号

（二）4 位二进制同步加法计数器芯片 74X163

74X163 芯片也是 4 位二进制同步加法计数器，要想了解 74X163 芯片的详细功能，读者可以从网上搜索它的数据手册，从数据手册中可以查看内部逻辑电路图、功能表、时序图，从而了解它的详细功能。此处为了节省篇幅，不再赘述。74X163 芯片与 74X161 芯片的功能基本相同，唯一不同的是采用同步清零方式。当清零端 CR 为 0 时，无论置数控制端 LD 和计数控制端 EP、ET 为任何状态，必须在时钟脉冲 CP 的上升沿来到时，所有触发器的输出 $Q_3Q_2Q_1Q_0$ 才被清零。图 5－80 是加了引脚名字的 74X163 的逻辑符号，图 5－81 是其状态图。

比较图 5－76 和图 5－80 所示的 74X161 和 74X163 两个芯片的逻辑符号，可以看出，74X161 芯片的 1 脚处为“CT＝0”，而 74X163 芯片的 1 脚处为“5CT＝0”，其中的“5”表示“CT＝0”清零功能的实现必须依赖于标号为“5”的那个信号，标号为“5”的信号是时钟脉冲，所以 74X163 芯片的清零信号为同步清零。

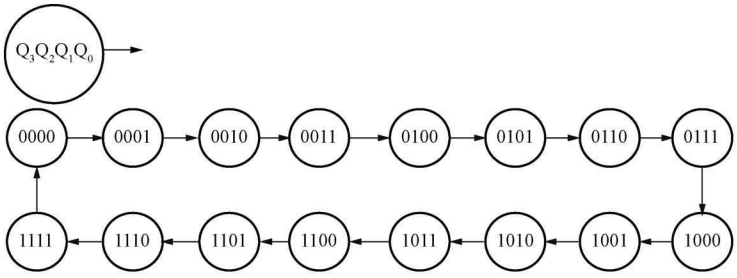


图 5－81　74X163 的状态图

（三）4 位二进制同步可逆计数器芯片 74X191

图 5－82 是加了引脚名字的 74X191 的逻辑符号。表 5－39 是其功能表。74X191 芯片没有清零功能。74X191 的 11 脚是置数控制端，异步且低电平有效。4 脚是使能端，低电平有效。5 脚是“加”、“减”控制端，当为高电平时为“减”计数，当为低电平时为“加”计数；12 脚、13 脚是进位输出端。

图 5－83、图 5－84 分别是 74X191 在“减”计数、“加”计数时的状态图。74X191 的时序图如图 5－85 所示。

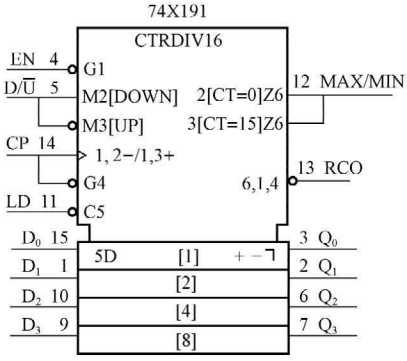


图 5－82　加了引脚名字的 74X191 的逻辑符号

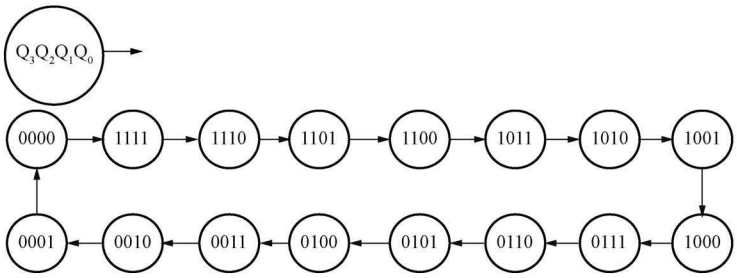


图 5 - 83　74*X*191 在“减”计数时的状态图

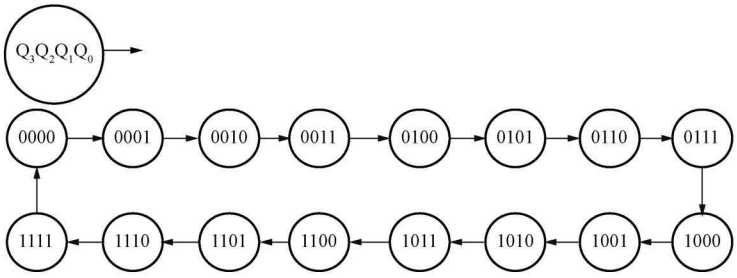


图 5 - 84　74*X*191 在“加”计数时的状态图

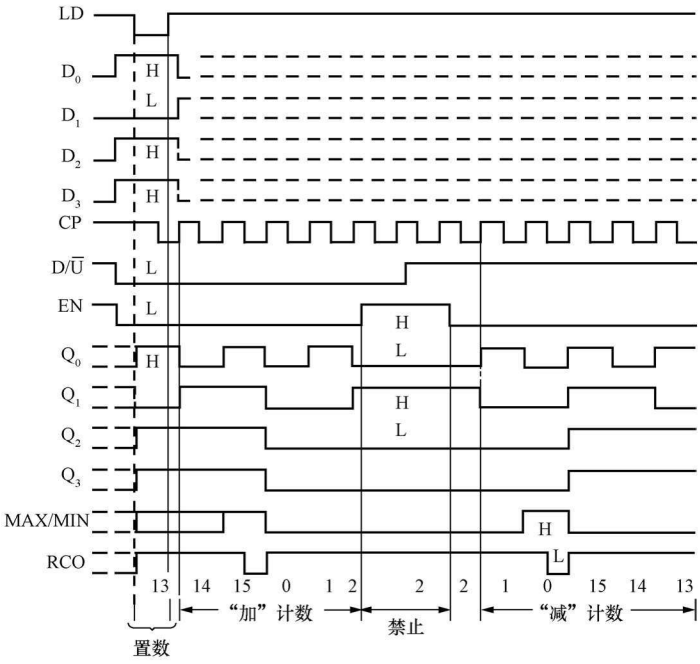


图 5 - 85　74*X*191 的时序图

在图 5 - 82 所示的 74X191 的逻辑符号中，总定性符号 CTRDIV16 表示是 16 进制计数器。4 脚右侧的 G1 表示 4 脚对应的内部信号被编号为 1，G 表示那些写有 1 字样的引脚对应的内部信号与 4 脚对应的内部信号之间是与的逻辑关系。5 脚的右侧写有 M2［DOWN］，另外，经过一个小圆圈之后内部写有 M3［UP］，M 表示模式的意思，2、3 分别是对应的内部

信号的编号，DOWN 表示是减计数模式，UP 表示是加计数模式。14 脚的右侧写有“>”字样，表示是时钟输入端，“1，2－/1，3＋”表示计数的规律，“/”表示“或”的意思，“1，2－”表示在标号为 1 的内部信号为 1、标号为 2 的内部信号为 1、在 14 脚输入的时钟上升沿的作用下，计数器按减计数规律进行状态的转换，－表示减计数的意思。“1，3＋”表示在标号为 1 的内部信号为 1、标号为 3 的内部信号为 1、在 14 脚输入的时钟上升沿的作用下，计数器按加计数规律进行状态的转换，＋表示加计数的意思，若要使标号为 3 的内部信号为 1，在 5 脚必须输入 0，因为有小圆圈存在。14 脚经过小圆圈之后定义了一个标号为 4 的内部信号，G 表示那些写有 4 字样的引脚对应的内部信号与标号为 4 的内部信号之间是与的逻辑关系。11 脚经过小圆圈之后定义了标号为 5 的内部信号，C 表示控制关系。15 脚的右侧写有 5D 字样，表示在标号为 5 的内部信号为 1 时将 15 脚的数据传输给 3 脚，这是异步置数，1 脚、10 脚、9 脚的右侧省略了 5D 字样，同理，在标号为 5 的内部信号为 1 时将 1 脚的数据传输给 2 脚，将 10 脚的数据传输给 6 脚，将 9 脚的数据传输给 7 脚。[1]、[2]、[4]、[8] 分别给 15 脚、1 脚、10 脚、9 脚对应的内部信号规定了位权。2 [CT＝0] Z6 表示在标号为 2 的内部信号为 1（即减计数模式）且内容为 0（即 $Q_3Q_2Q_1Q_0=0000$）时，在 12 脚输出 1，并且把 12 脚对应的内部信号标号为 6。3 [CT＝15] Z6 表示在标号为 3 的内部信号为 1（即加计数模式）且内容为 15（即 $Q_3Q_2Q_1Q_0=1111$）时，在 12 脚输出 1，并且把 12 脚对应的内部信号标号为 6。13 脚左侧的“6，1，4”字样表示在标号为 6 的内部信号为 1、标号为 1 的内部信号为 1、标号为 4 的内部信号为 1 时 13 脚对应的内部信号为 1，经过小圆圈反相后在 13 脚输出 0。

表 5 - 39　　74X191 的功能表

预置	使能	加/减控制	时钟	预置数据输入				输出				工作模式
LD	EN	$D/\overline{U}$	CP	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	
0	×	×	×	d_3	d_2	d_1	d_0	d_3	d_2	d_1	d_0	异步置数
1	1	×	×	×	×	×	×	保持				数据保持
1	0	0	↑	×	×	×	×	加法计数				加法计数
1	0	1	↑	×	×	×	×	减法计数				减法计数

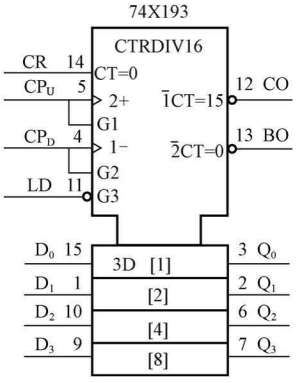


图 5 - 86　加了引脚名字的 74X193 的逻辑符号

（四）4 位二进制同步可逆计数器芯片 74X193

74X193 是 4 位二进制同步可逆计数器。图 5 - 86 是加了引脚名字的 74X193 的逻辑符号，CR 是异步清零端，LD 是预置数控制端，$D_3D_2D_1D_0$ 是预置数据输入端，CP_U 是“加”计数时钟脉冲输入端，CP_D 是“减”计数时钟脉冲输入端，CO 是进位输出端，BO 是借位输出端。74X193 和 74X191 都是 4 位二进制同步可逆计数器，它们的区别是 74X191 设置了一个“加”和“减”计数的控制端，“加”和“减”计数使用同一个时钟脉冲；而 74X193 为“加”计数和“减”计数分别设置了两个时钟脉冲 CP_U 和 CP_D。图 5 - 87 是其时序图。图 5 - 88、图 5 - 89 分别为 74X193 在“减”计数和“加”计数时的状态图。

先将逻辑符号各部分简要说明如下：

CTRDIV16 是总定性符，16 表明它是十六进制计数器。“CT＝0”表明内容是 0，说明这个引脚是清零输入端。“CT＝0”的左边没有其他字样，表明是异步清零，没有小圆圈表明是高电平有效。

“＞2＋”表明 5 脚是时钟输入端，在内部标号为“2”的信号 4 脚 CP_D 为高电平的情况下，将进行“加”计数。5 脚没有小圆圈表明是上升沿有效。

“＞1－”表明 4 脚是时钟输入端，在内部标号为“1”的信号 5 脚 CP_U 为高电平的情况下，将进行“减”计数。4 脚没有小圆圈表明是上升沿有效。

11 脚 LD 低电平有效，异步置数。15 脚 3D 表示 G3＝1（即 LD＝0）时，置数操作。

12 脚 CO 是进位输出端，低电平有效。$\overline{1}$CT＝15：表示 G1（即 CP_U）＝0，且计数器状态 $Q_3Q_2Q_1Q_0$＝1111 时，产生进位输出，因为输出端画有小圆圈，所以输出为 0。

13 脚 BO 是借位输出端，低电平有效。$\overline{2}$CT＝0：表示 G2（即 CP_D）＝0，且计数器状态为 0，即 $Q_3Q_2Q_1Q_0$＝0000 时，产生借位输出，因为输出端画有小圆圈，所以输出为 0。

从表 5－40 可以看出 74X193 具有以下功能：

（1）异步清零。当 CR＝1 时，不管其他输入端的状态如何，不论有无时钟脉冲 CP，计数器输出将被直接置零（$Q_3Q_2Q_1Q_0$＝0000），称为异步清零。CR 高电平有效，具有最高的优先级别。

（2）异步并行预置数。在异步清零端 CR 无效（CR＝0）的情况下，当 LD＝0 时，不论有无时钟脉冲，并行输入端的数据 $D_3D_2D_1D_0$ 被置入计数器的输出端，即 $Q_3Q_2Q_1Q_0$＝$D_3D_2D_1D_0$。异步并行预置数端 LD 为低电平有效。优先级别比异步清零端 CR 低，即必须在异步清零端 CR 无效的情况下才能预置数。

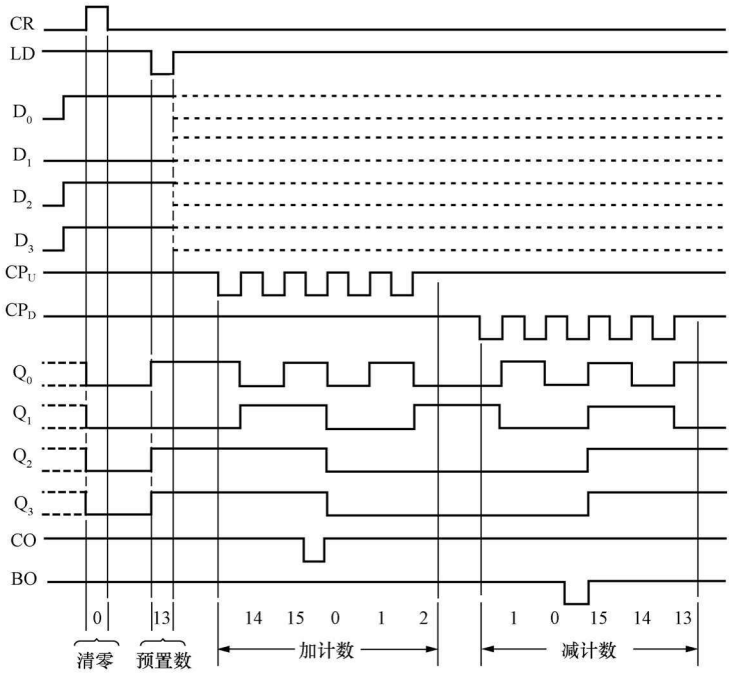


图 5－87　74X193 的时序图

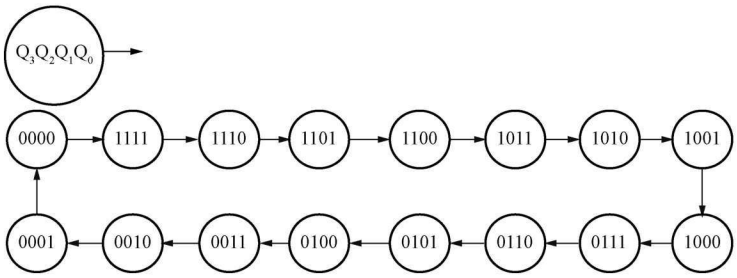


图 5 - 88　74X193 在“减”计数时的状态图

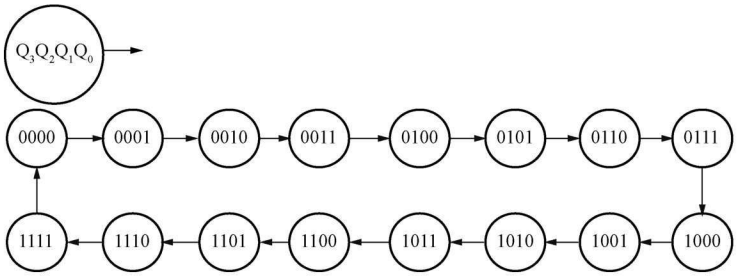


图 5 - 89　74X193 在“加”计数时的状态图

表 5 - 40　　74X193 的功能表

清零	预置	“加”计数时钟	“减”计数时钟	预置数据输入				输出				工作模式
CR	LD	CP_U	CP_D	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	
1	×	×	×	×	×	×	×	0	0	0	0	异步清零
0	0	×	×	D	C	B	A	D	C	B	A	异步置数
0	1	1	1	×	×	×	×	保持				数据保持
0	1	↑	1	×	×	×	×	计数				加法计数
0	1	1	↑	×	×	×	×	计数				减法计数

（3）保持。当异步清零端 CR 无效、异步并行预置数端 LD 无效时，且 $CP_U = CP_D = 1$，计数器保持原来的状态不变。

（4）“加”计数。当 $CP_D = 1$ 时，计数器响应 CP_U 端的上升沿进行“加”计数，当 $Q_3Q_2Q_1Q_0 = 1111$，且 $CP_U = 0$ 时，$CO = \overline{Q_3Q_2Q_1Q_0\ \overline{CP_U}} = 0$。

（5）“减”计数。当 $CP_U = 1$ 时，计数器响应的 CP_D 上升沿进行减法计数，当 $Q_3Q_2Q_1Q_0 = 0000$，且 $CP_D = 0$ 时，$BO = \overline{\overline{Q_3}\ \overline{Q_2}\ \overline{Q_1}\ \overline{Q_0}\ \overline{CP_D}} = 0$。

（五）8421BCD 码同步加法计数器 74X160 芯片

74X160 是 8421BCD 码同步加法计数器。图 5 - 90 是加了引脚名字的 74X160 的逻辑符号。图 5 - 91 是 74X160 的状态图。74X160 的功能表如表 5 - 41 所示。读者查阅生产厂家的数据手册可以得到 74X160 芯片的内部逻辑电路图，从内部逻辑电路图可以观察出进位输出端 RCO 的逻辑表达式为

$$RCO = ET \cdot Q_3Q_0$$

从功能表 5 - 41 可以看出，CR 清零信号为低电平有效，异步清零。注意引处的“异步清零”，在 CR 清零信号无效的前提下，预置数信号 LD 为低电平有效，同步预置数。在清零信号和预置数信号无效的前提下，使能端 EP 和 ET 不同时为 1 的前提下，74X160 处于保持状态，即虽然有效的时钟沿源源不断的到来，但是触发器的状态并不改变。在清零信号和预置数信号无效的前提下，使能端 EP 和 ET 同时为 1（有效）的前提下，计数器处于正常的计数功能，从表 5 - 41 中只能知道计数规律是“十进制”计数，更具体的计数规律从表 5 - 41 中不能看出，可以从 74X160 的时序图中观察出它的计数规律，本书没有给出，读者可以察看数据手册，其实从图 5 - 90 的 74X160 的逻辑符号也可以观察出来。在图 5 - 91 中要特别注意 $Q_3Q_2Q_1Q_0$ 的顺序，顺序不能改变。

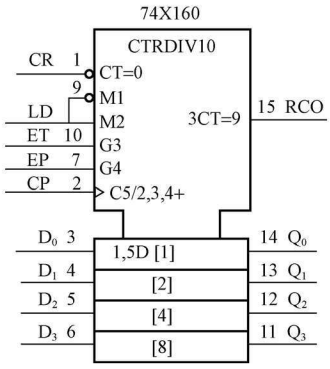


图 5 - 90　加了引脚名字的 74X160 的逻辑符号

（六）二—五—十进制异步加法计数器 74X290

74X290 是二—五—十进制异步加法计数器。图 5 - 92 是加了引脚名字的 74X290 的逻辑符号。图 5 - 93 是 74X290 的内部逻辑电路图。$R_{0(1)}$、$R_{0(2)}$ 为复位输入端，$S_{9(1)}$、$S_{9(2)}$ 为置位输入端。它包含一个独立的 1 位二进制计数器和一个独立的异步五进制计数器。二进制计数器的时钟输入端为 CP_1，输出端为 Q_0；五进制计数器的时钟输入端为 CP_2，输出端为 Q_1、Q_2、Q_3。图 5 - 94 是 1 位二进制计数器的状态图，图 5 - 95 是五进制计数器的状态图。

表 5 - 41　　74X160 的功能表

清零	预置数	使能		时钟	预置数据输入				输出				工作模式
CR	LD	ET	EP	CP	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	
0	×	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	0	×	×	↑	D	C	B	A	D	C	B	A	同步置数
1	1	0	×	×	×	×	×	×	保持				数据保持
1	1	×	0	×	×	×	×	×	保持				数据保持
1	1	1	1	↑	×	×	×	×	十进制计数				加法计数

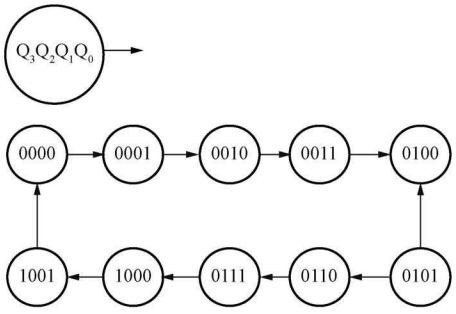


图 5 - 91　74X160 的状态图

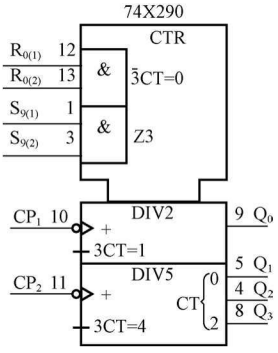


图 5 - 92　加了引脚名字的 74X290 的逻辑符号

表 5-42 是 74X290 的功能表。由表可知，74X290 具有以下功能：

（1）异步清零。功能表的第一、二行表示：当复位输入端 $R_{0(1)}=R_{0(2)}=1$（两个复位端同时有效）且置位输入 $S_{9(1)}S_{9(2)}=0$（至少有一端无效）时，不论有无时钟脉冲 CP，计数器输出将被直接清零。

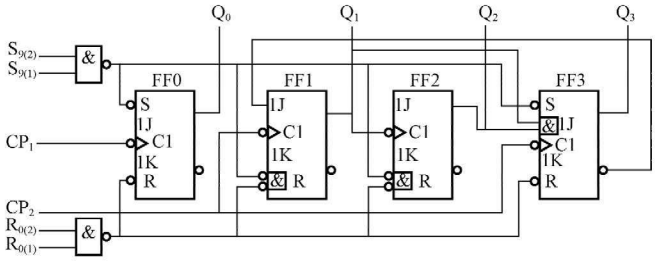


图 5-93　74X290 的内部逻辑电路图

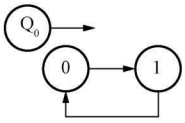


图 5-94　74X290 二进制计数器状态图

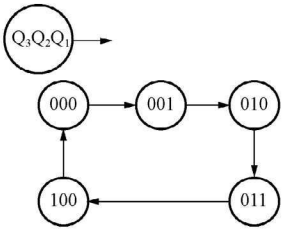


图 5-95　74X290 五进制计数器状态图

（2）异步置 9。功能表的第三行表示：当置位输入 $S_{9(1)}=S_{9(2)}=1$ 时，无论其他输入端状态如何，计数器输出将被直接置 9（即 $Q_3Q_2Q_1Q_0=1001$）。置 9 信号比清零信号优先级别高。

（3）计数。功能表的第四、五、六、七行表示：当 $R_{0(1)}\cdot R_{0(2)}=0$ 且 $S_{9(1)}\cdot S_{9(2)}=0$ 时，在各自计数脉冲（下降沿）作用下，分别按二进制计数器和五进制计数器运行。

表 5-42　　74X290 的功能表

复位输入		置位输入		时钟	输出				工作模式
$R_{0(1)}$	$R_{0(2)}$	$S_{9(1)}$	$S_{9(2)}$	CP	Q_3	Q_2	Q_1	Q_0	
1	1	0	×	×	0	0	0	0	异步清零
1	1	×	0	×	0	0	0	0	
×	×	1	1	×	1	0	0	1	异步置数
0	×	0	×	↓	计数				加法计数
0	×	×	0	↓	计数				
×	0	0	×	↓	计数				
×	0	×	0	↓	计数				

74X290 的国标符号各部分简要说明如下：

在图 5-92 中，上面部分为公共控制框，下面为两个独立的输出框，分别表示一个独立的二进制计数器和一个独立的五进制计数器。在公共控制框中，74X290 的 1 脚 $S_{9(1)}$、3 脚 $S_{9(2)}$ 相与后标有字样“Z3”，当 $S_{9(1)}=S_{9(2)}=1$ 时标号为 3 的内部信号的值为 1，Z 表示与其他的内部信号（有字样“3CT=1”、“3CT=4”）有连接，“3CT=1”表示当标号为 3 的内部信号为 1 时 CT=1，即内容为 1，$Q_0=1$。“3CT=4”表示当标号为 3 的内部信号为 1 时

CT＝4，即内容为 4，$Q_3Q_2Q_1$＝100。74X290 的 12 脚 $R_{0(1)}$、13 脚 $R_{0(2)}$ 相与后标有字样“$\overline{3}$CT＝0”，意思是 1 脚 $S_{9(1)}$ 或 3 脚 $S_{9(2)}$ 无效的前提下，12 脚 $R_{0(1)}$ 和 13 脚 $R_{0(2)}$ 均有效时 $Q_3Q_2Q_1Q_0$ 被置成 0000。可以说置 9 信号比清零信号优先级别高。

74X290 的 1 脚 $S_{9(1)}$、3 脚 $S_{9(2)}$ 没有小圆圈，表明是高电平有效，相应的位置也没有标示与 CP 有关的字样，所以为异步置 9。

74X290 的 12 脚 $R_{0(1)}$、13 脚 $R_{0(2)}$ 没有小圆圈，表明是高电平有效。相应的位置也没有标示与 CP 有关的字样，所以为异步复位。

有这样一个规律，异步计数器的清零端和置数端均为异步的，不可能为同步的。而同步计数器的清零端和置数端有可能为异步的，也有可能为同步的。

74X290 的逻辑符号的上面的输出框 DIV2 表示一个独立的二进制计数器，10 脚为时钟输入端，小圆圈表示下降沿有效，其中的“＋”号表示“加”计数。“3CT＝1”表示在标号为“3”的信号为 1 时，即 1 脚 $S_{9(1)}$、3 脚 $S_{9(2)}$ 均有效时 9 脚 Q_0 被置成 1。

74X290 的逻辑符号的下面的输出框 DIV5 表示一个独立的五进制计数器，11 脚为时钟输入端，小圆圈表示下降沿有效，其中的“＋”号表示“加”计数。“3CT＝4”表示在标号为“3”的信号为 1 时，即 1 脚 $S_{9(1)}$、3 脚 $S_{9(2)}$ 均有效时 $Q_3Q_2Q_1$ 被置成 100。

$CT\begin{cases}0\\2\end{cases}$ 表明了 8 脚 Q_3、4 脚 Q_2、5 脚 Q_1 的位权分别为 2^2、2^1、2^0。

如果将 Q_0 与 CP_2 相连，CP_1 作时钟脉冲输入端，Q_3～Q_0 作输出端，则为 8421BCD 码十进制计数器，如图 5－96 所示。该计数器的状态图如图 5－97 所示。图 5－98 是 74X290 组成的 8421BCD 码十进制计数器的波形图。

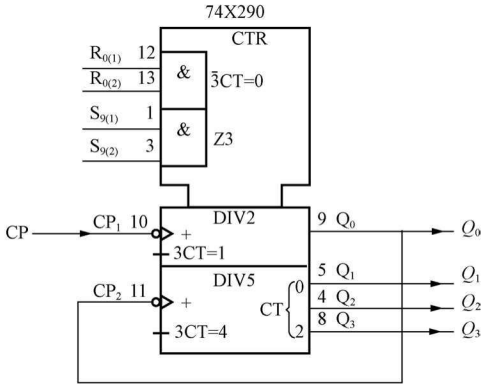


图 5－96　74X290 组成的 8421BCD 码十进制计数器的逻辑电路图

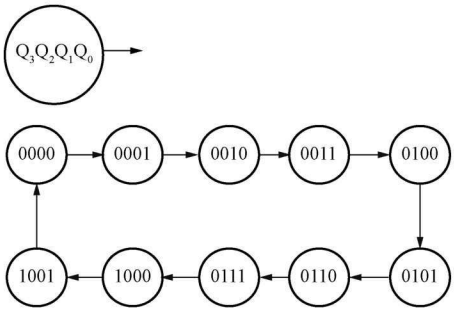


图 5－97　74X290 组成的 8421BCD 码十进制计数器的状态图

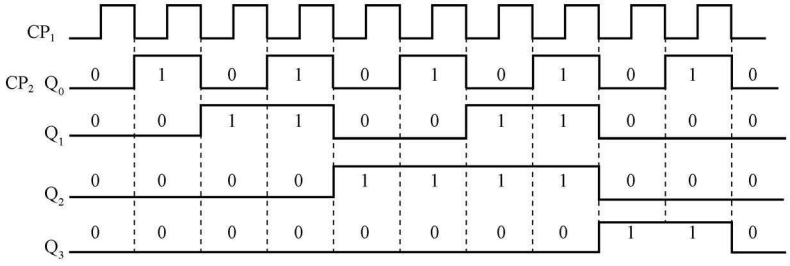


图 5－98　74X290 组成的 8421BCD 码十进制计数器的波形图

如果将 Q_3 与 CP_1 相连，CP_2 作时钟脉冲输入端 CP，$Q_3 \sim Q_0$ 作输出端，则为 5421BCD 码十进制计数器，如图 5 - 99 所示，该计数器的状态图如图 5 - 100 所示。图 5 - 101 是 74X290 组成的 5421BCD 码十进制计数器的波形图。

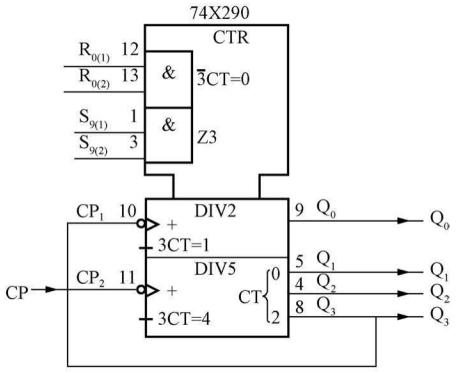


图 5 - 99　74X290 组成的 5421BCD 码十进制计数器的逻辑电路图

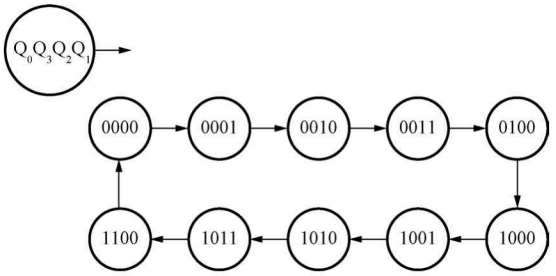


图 5 - 100　74X290 组成的 5421BCD 码十进制计数器的状态图

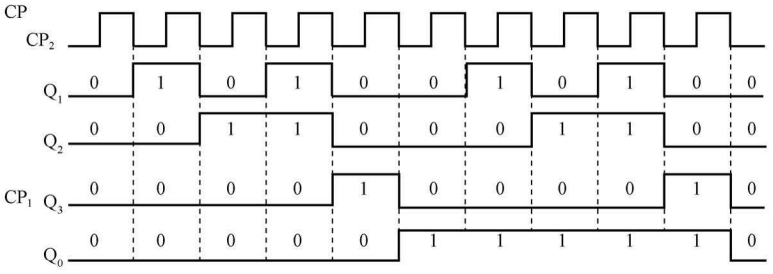


图 5 - 101　74X290 组成的 5421BCD 码十进制计数器的波形图

中规模集成计数器种类较多，功能强，使用方便。表 5 - 43 列举了几种集成计数器产品。

表 5 - 43　　几种集成计数器

触发器的 CP 之间的关系	型号	计数模式	清零方式	预置数方式
同步	74X160	十进制“加”计数器	异步（低电平有效）	同步（低电平有效）
	74X161	4 位二进制“加”计数器	异步（低电平有效）	同步（低电平有效）
	74X162	十进制“加”计数器	同步（低电平有效）	同步（低电平有效）
	74X163	4 位二进制“加”计数器	同步（低电平有效）	同步（低电平有效）
	74X190	单时钟可逆十进制计数器	无	异步（低电平有效）
	74X191	单时钟可逆 4 位二进制计数器	无	异步（低电平有效）
	74X192	双时钟可逆十进制计数器	异步（高电平有效）	异步（低电平有效）
	74X193	双时钟可逆 4 位二进制计数器	异步（高电平有效）	异步（低电平有效）

续表

触发器的 CP 之间的关系	型号	计数模式	清零方式	预置数方式
异步	74X290	二—五—十进制“加”计数器	异步（高电平有效）	预置 9，异步（高电平有效）
	74X293	二—八—十六进制“加”计数器	异步（高电平有效）	无
	74X90	二—五—十进制“加”计数器	异步（高电平有效）	预置 9，异步（高电平有效）
	74X92	二—六—十二进制“加”计数器	异步（高电平有效）	无
	74X93	二—八—十六进制“加”计数器	异步（高电平有效）	无

二、集成计数器的应用

（一）计数器容量扩展

为了扩大计数范围，要把多个计数器进行级联。m 个模 N 计数器级联，可以实现模 N^m 的计数器。计数器级联的方式有两种：一种是级间采用串行进位方式，即异步级联方式，这种方式是将低位计数器的进位输出直接作为高位计数器的时钟脉冲，异步方式的速度较慢；另一种是级间采用并行进位方式，即同步级联方式，这种方式一般是把各计数器的 CP 端连在一起接统一的时钟脉冲，而低位计数器的进位输出送高位计数器的计数控制端。

1. 同步级联方式

图 5 - 102 是用两片 4 位二进制加法计数器 74X161 采用同步级联方式构成的 8 位二进制同步加法计数器，模为 16×16＝256。两片 74X161 的 CP 端均与计数脉冲 CP 连接，因而是同步计数器。片（1）的计数使能端 ET＝EP＝1，因而它总处于计数状态；片（2）的计数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 5 - 103 是其波形图。

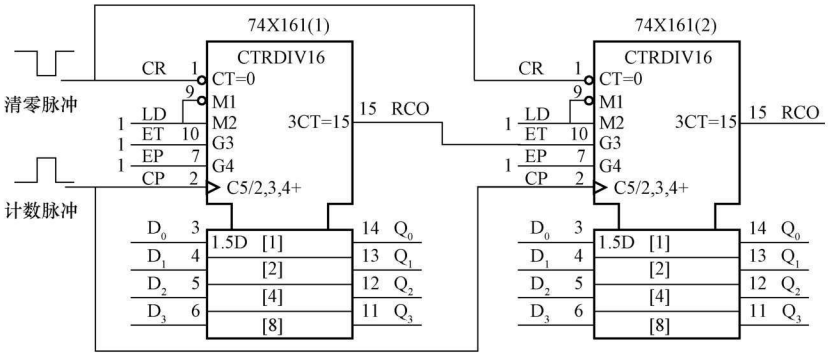


图 5 - 102　两片 74X161 同步级联组成 8 位二进制加法计数器的逻辑电路图

2. 异步级联方式

（1）两片 74X161 异步级联构成 8 位二进制计数器（256 进制计数器）。用两片 74X161

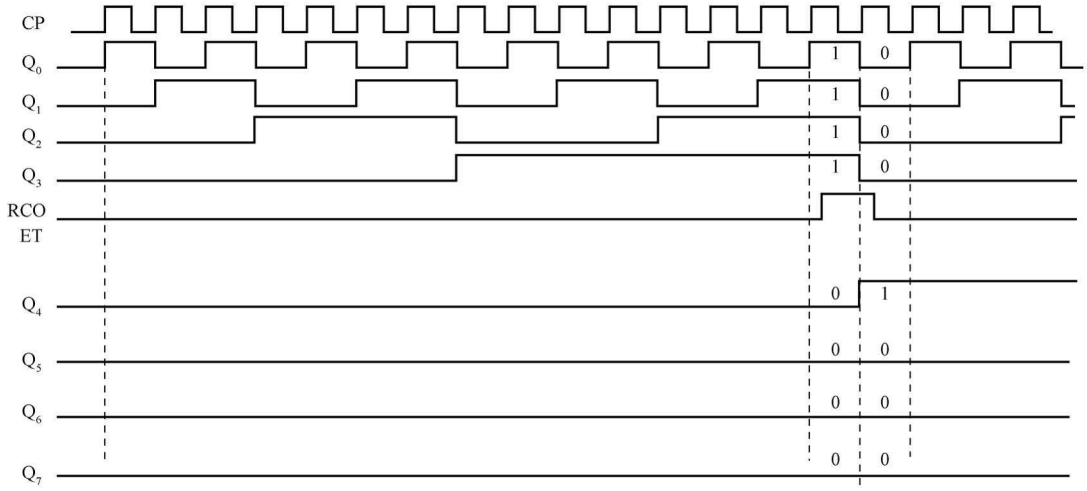


图 5 - 103　两片 74X161 同步级联组成 8 位二进制加法计数器的波形图

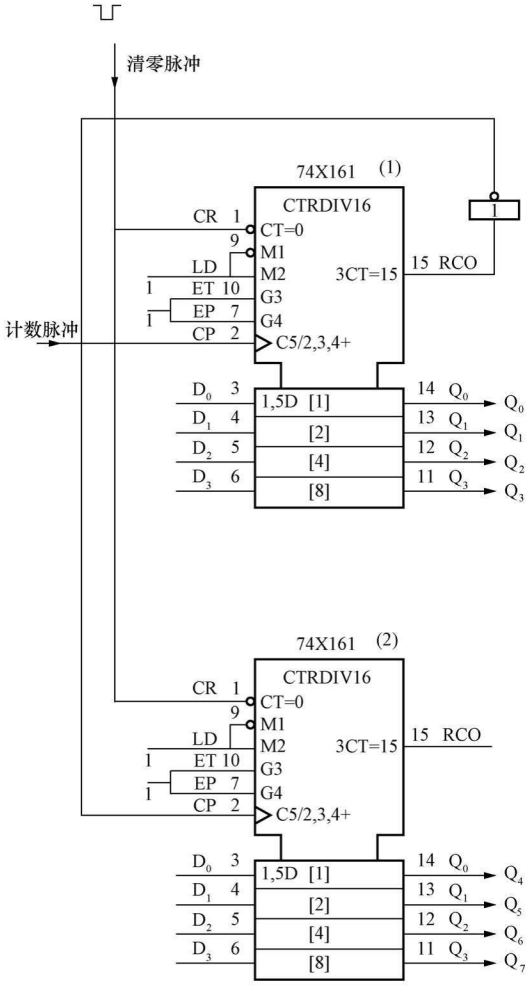


图 5 - 104　两片 74X161 异步级联组成 8 位二进制加法计数器的逻辑电路图

采用异步级联方式构成的 8 位二进制异步加法计数器如图 5 - 104 所示。图中，片（1）的 2 脚 CP 做为 256 进制计数器的时钟信号。片（1）和片（2）的使能端均有效，片（1）和片（2）的 LD 端均无效。片（1）的进位输出信号 RCO 经过一个非门反相后作为片（2）的计数脉冲 CP，因而是异步级联方式。只有在片（1）的 $Q_3Q_2Q_1Q_0$ 由 1111 变 0000 状态的时刻，使其 RCO 由 1 变为 0，经过非门反相后信号由 0 变为 1，片（2）的 CP 才能出现一个脉冲的上升沿，片（2）的状态改变一次。其他情况下，片（2）都将保持原有状态不变。图5 - 105是该 8 位二进制计数器的波形图。

（2）两片 74X193 异步级联构成 8 位二进制计数器（256 进制计数器）。用两片 74X193 采用异步级联方式构成的 8 位二进制异步加法计数器如图 5 - 106 所示。图中，片（1）和片（2）的 CP_D 都接 1，时钟脉冲从 CP_U 输入，而片（1）的 CP_U 做为 256 进制计数器的时钟输入端。两片 74X193 都按“加”计数器工作，其中，片（1）的进位输出信号 CO 作为片（2）的计数脉冲 CP_U，因而是异步级联方式。只有在片（1）的 $Q_3Q_2Q_1Q_0$ 由 1111 变 0000 状态的时刻，使其 CO 由 0 变为 1，片（2）的

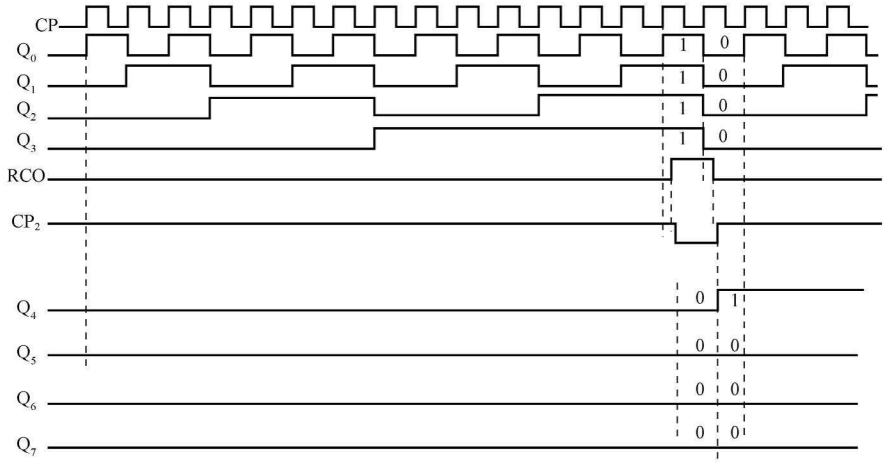


图 5 - 105　两片 74X161 异步级联组成 8 位二进制加法计数器的波形图

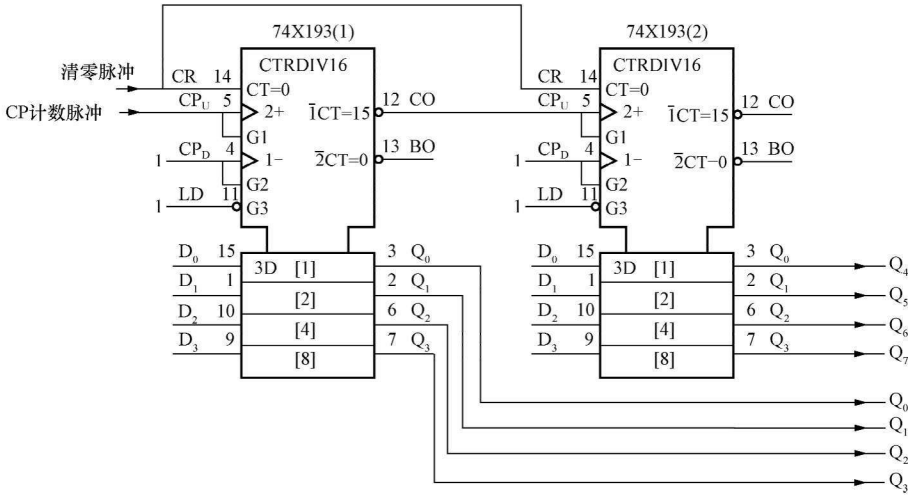


图 5 - 106　74X193 异步级联组成 8 位二进制加法计数器

CP_U 才能出现一个脉冲的上升沿，片（2）的状态改变一次。其他情况下，片（2）都将保持原有状态不变。图 5 - 107 是该 8 位二进制计数器的波形图。

（3）两片 74X290 异步级联构成 100 进制计数器。有的集成计数器没有进位/借位输出端，这时可根据具体情况，用计数器的输出信号 Q_3、Q_2、Q_1、Q_0 产生一个进位/借位。如用两片二—五—十进制异步加法计数器 74X290 采用异步级联方式组成的 100 进制加法计数器如图 5 - 108 所示，图 5 - 109 是其波形图。

（二）组成任意进制计数器

市场上能买到的集成计数器一般为二进制和 8421BCD 码十进制计数器，如果需要其他进制的计数器，可用现有的二进制或十进制计数器，利用其清零端或预置数端，外加适当的门电路连接而成。采用的方法有两种：一种是反馈清零法，另一种是反馈预置数法。

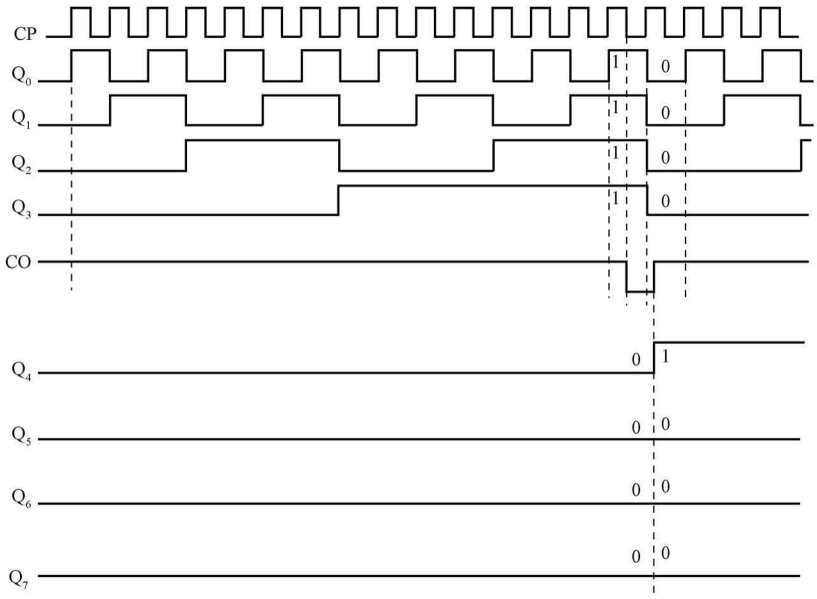


图 5-107　74X193 异步级联组成 8 位二进制加法计数器的波形图

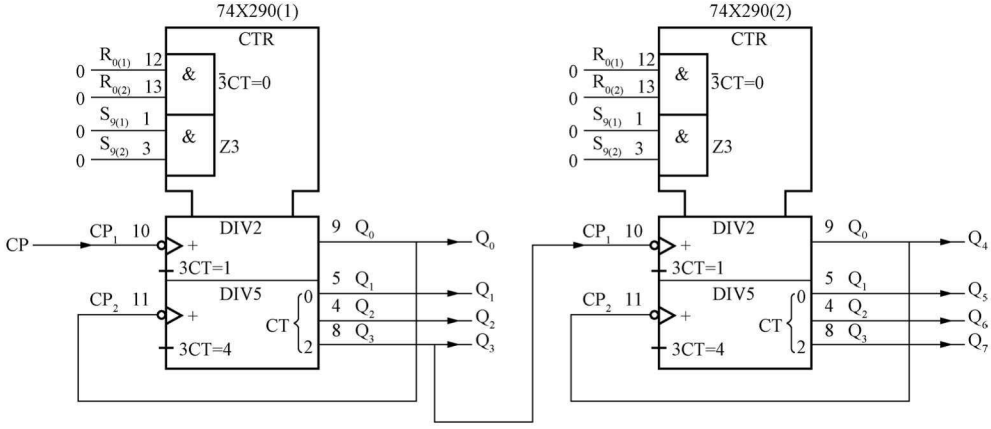


图 5-108　74X290 异步级联组成 100 进制计数器的逻辑电路图

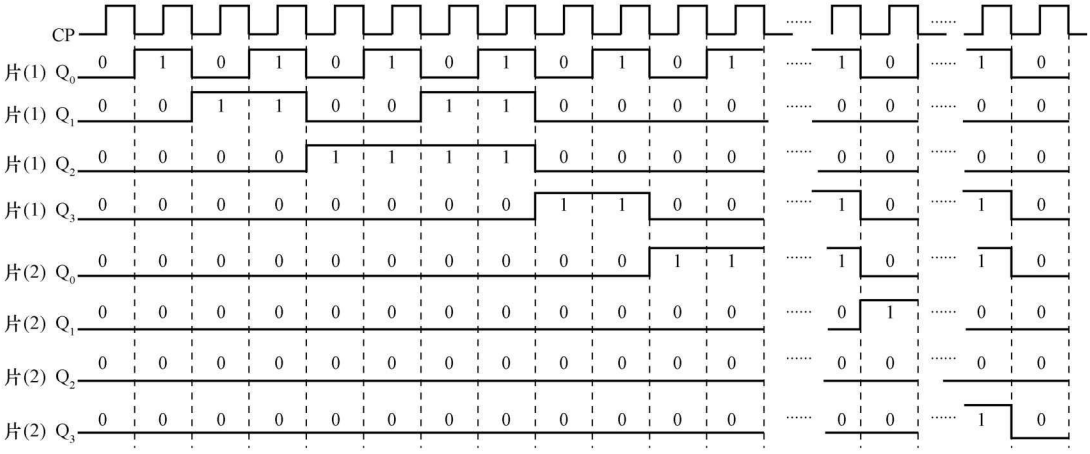


图 5-109　74X290 异步级联组成 100 进制计数器的波形图

用现有的最大计数值为 N 的计数器去实现一个任意模值为 M 的计数器，如果 $M<N$，则只需一片 N 进制计数器，使计数器在 N 进制的计数过程中，跳过 $N-M$ 个状态，就可以得到模 M 计数器。如果 $M>N$，要求实现的模值 M 超过单片计数器的计数范围时，必须首先将多片计数器级联，以扩大计数范围。级联的方法采用上面讲过的同步级联和异步级联两种方法；然后采用反馈清零或反馈置数的方式构成 M 进制计数器。

1. 反馈清零法

反馈清零法适用于有清零输入信号端的集成计数器。

（1）同步反馈清零法。现举例说明用同步反馈清零法设计计数器的方法。

【例 5-13】 用集成计数器 74X163 和必要的门电路组成 6 进制计数器，要求使用反馈清零法。

解 74X163 的状态图见图 5-81，74X163 本身有 16 个状态。6 进制计数器有 6 个状态，至于是哪 6 个状态，无所谓，只要状态的数目是 6 个即可。但是此题目明确要求使用反馈清零法，所以状态图中必须包括 $Q_3Q_2Q_1Q_0$ 为 0000 的状态，状态图如图 5-110 所示。

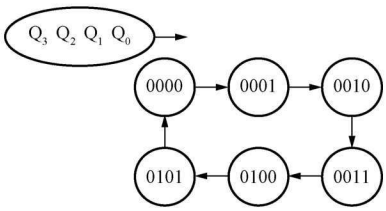


图 5-110 ［例 5-13］的状态图

集成计数器 74X163 具有同步清零功能，在其计数过程中，如果在同步清零输入端 CR 加入一低电平，还需要 CP 脉冲上升沿的配合，即 74X163 的状态在即将到来的 CP 脉冲上升沿到达时回到 0000 状态。清零信号（CR=0）消失后，就从 0000 状态开始重新计数。

设计要求电路当 $Q_3Q_2Q_1Q_0$ 为 0101 状态时产生一个低电平信号，这个信号送给 74X163 的同步清零输入端 CR。表 5-44 是求 CR 清零信号的真值表。图 5-111 是求 CR 清零信号的卡诺图。从卡诺图写出 CR 的表达式为 $CR=\overline{Q_0^n}+\overline{Q_2^n}=\overline{Q_0^nQ_2^n}$。图 5-112 是所求电路的逻辑电路图，图 5-113 是其波形图。

表 5-44　　求 CR 清零信号的真值表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR	Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR
0	0	0	0	1	1	0	0	0	×
0	0	0	1	1	1	0	0	1	×
0	0	1	0	1	1	0	1	0	×
0	0	1	1	1	1	0	1	1	×
0	1	0	0	1	1	1	0	0	×
0	1	0	1	0	1	1	0	1	×
0	1	1	0	×	1	1	1	0	×
0	1	1	1	×	1	1	1	1	×

（2）异步反馈清零法。现举例说明异步反馈清零法设计计数器的方法。

【例 5-14】 用集成计数器 74X161 和必要的门电路构成 6 进制计数器，要求使用反馈清零法。

解 74X161 的状态图见图 5-79，74X161 本身有 16 个状态，6 进制计数器有 6 个状态。题目明确要求使用反馈清零法，所以状态图中必须包括 $Q_3Q_2Q_1Q_0$ 为 0000 的状态，在此取 6 个状态如图 5-114 所示。

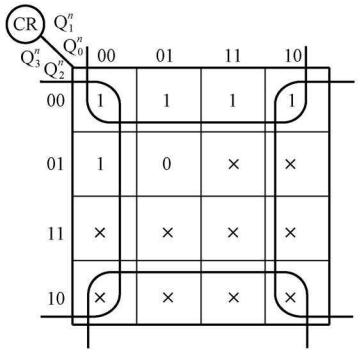


图 5 - 111 求 CR 清零信号的卡诺图

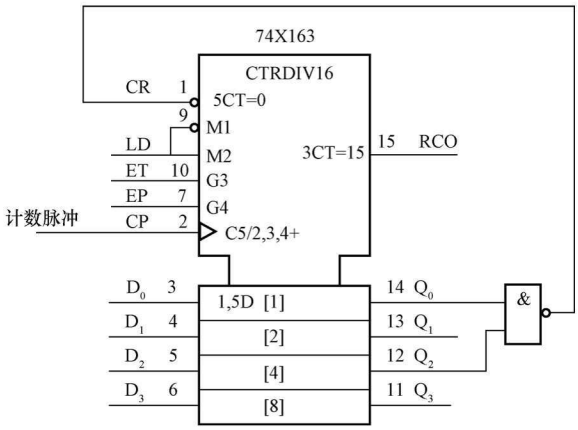


图 5 - 112 ［例 5 - 13］的逻辑电路图

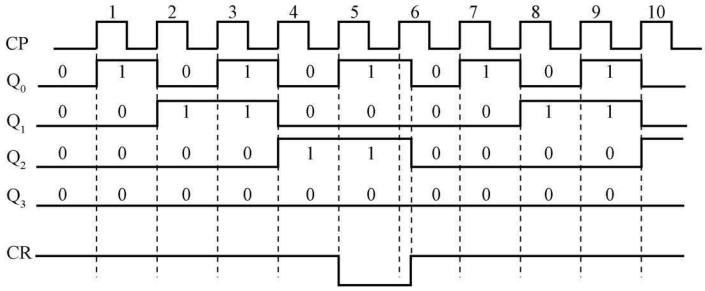


图 5 - 113 ［例 5 - 13］的波形图

集成计数器 74X161 具有异步清零功能，在其计数过程中，不管它处于哪一个状态，只要在异步清零输入端 CR 加入一低电平，74X161 的状态就回到 0000 状态。清零信号（CR＝0）消失后，74X161 就从 0000 状态开始重新计数。

此设计的重点是 $Q_3Q_2Q_1Q_0$ 在 0101 状态下产生一个低电平信号，送给异步清零输入端 CR。此题与［例 5 - 13］最大的不同是 74X161 具有异步清零功能，而 74X163 具有同步清零功能，所以必须在 $Q_3Q_2Q_1Q_0$ 为 0110 时产生一个低电平信号 CR，而不能在 $Q_3Q_2Q_1Q_0$ 为 0101 时产生一个低电平信号 CR。增加了 0110 过渡状态的状态图如图 5 - 115 所示。

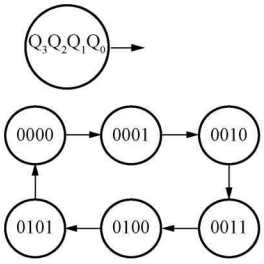


图 5 - 114 ［例 5 - 14］的状态图

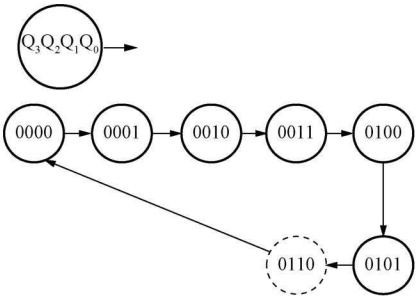


图 5 - 115 增加了 0110 过渡状态的［例 5 - 14］的状态图

表 5 - 45 是求 CR 清零信号的真值表，表 5 - 46 是［例 5 - 14］的完整的求 CR 清零信号的真值表。图 5 - 116 是求 CR 清零信号的卡诺图。

表 5 - 45　　［例 5 - 14］求 CR 清零信号的真值表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR	Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR
0	0	0	0	1	0	1	0	0	1
0	0	0	1	1	0	1	0	1	1
0	0	1	0	1	0	1	1	0	0
0	0	1	1	1					

表 5 - 46　　［例 5 - 14］的完整的求 CR 清零信号的真值表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR	Q_3^n	Q_2^n	Q_1^n	Q_0^n	CR
0	0	0	0	1	1	0	0	0	×
0	0	0	1	1	1	0	0	1	×
0	0	1	0	1	1	0	1	0	×
0	0	1	1	1	1	0	1	1	×
0	1	0	0	1	1	1	0	0	×
0	1	0	1	1	1	1	0	1	×
0	1	1	0	0	1	1	1	0	×
0	1	1	1	×	1	1	1	1	×

从图 5 - 116 的卡诺图写出 CR 的表达式为

$$CR = \overline{Q_1^n} + \overline{Q_2^n} = \overline{Q_1^n \cdot Q_2^n}$$

根据此表达式画出逻辑电路图如图 5 - 117 所示，画出波形图如 5 - 118 所示。

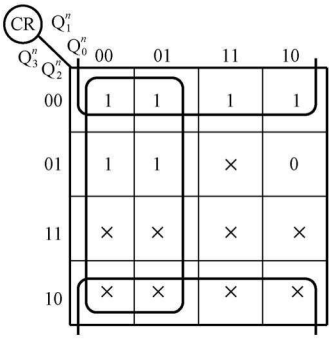


图 5 - 116　［例 5 - 14］求 CR 清零信号的卡诺图

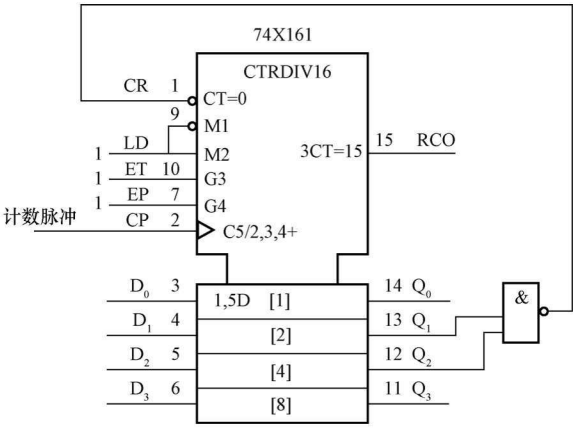


图 5 - 117　［例 5 - 14］的逻辑电路图

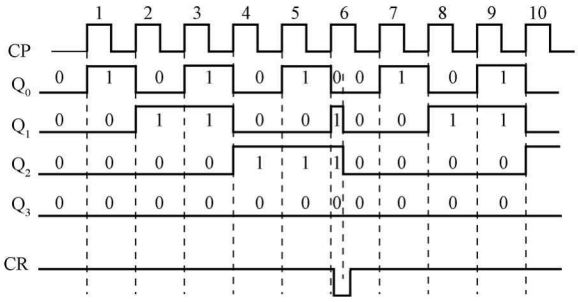


图 5 - 118 ［例 5 - 14］的波形图

在图 5 - 118 所示的波形图中，当输入第 6 个 CP 脉冲时，在其上升沿，输出 $Q_3Q_2Q_1Q_0$ 跳变为 0110，Q_2Q_1 的 11 通过与非门译码后，反馈给 CR 端一个清零信号，输出会立即从 0110 状态回到 0000 状态（0110 只出现一瞬间，为过渡状态，又立即被置成 0000 状态，所以在状态图5 - 115 中用虚线表示）。0000 状态和 0110 状态所维持的时间加起来为一个时钟周期，当 $Q_3Q_2Q_1Q_0$ 为 0000，CR 端的清零信号也随之消失，74X161 又从 0000 状态开始重新计数。这样就跳过了 0110～1111 共 10 个状态，实现了 6 进制计数器。图 5 - 119 为所求电路的完整状态图。

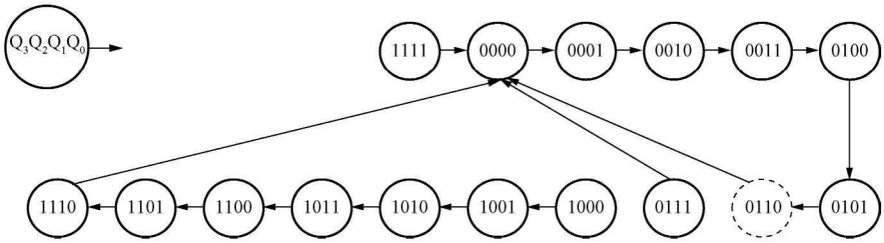


图 5 - 119 ［例 5 - 14］的完整状态图

2. 反馈置数法

反馈置数法一般适用于有预置功能的集成计数器。

（1）同步反馈置数法。现举例说明同步反馈置数法设计计数器的方法。

【例 5 - 15】 用集成计数器 74X160 和必要的门电路组成 7 进制计数器，要求该电路的有效状态是 $Q_3Q_2Q_1Q_0$ 按“加 1”的顺序从 0011 到 1001 循环变化。

解 8421BCD 码同步加法集成计数器 74X160 的状态图见图5 - 91，74X160 本身共有 10 个状态，具有同步预置功能。此题目明确要求使用反馈置数法，而且规定了是哪 7 个状态，其状态图如图 5 - 120 所示。

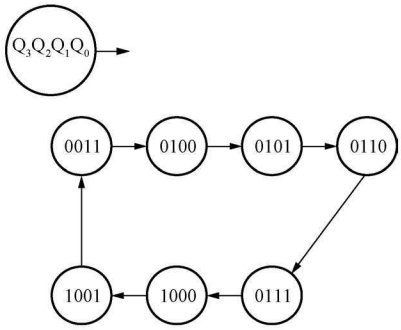


图 5 - 120 ［例 5 - 15］的状态图

集成计数器 74X160 具有同步反馈置数功能，在其计数过程中，在同步反馈置数输入端 LD 加入一低电平，还需要 CP 脉冲上升沿的配合，即 74X160 的状态在最近马上要到来的一个 CP 脉冲上升沿到达时把 $D_3D_2D_1D_0$ 送入 $Q_3Q_2Q_1Q_0$。置数信号（LD＝0）消失后，74X160 就从刚才进入的状态开始重新计数。

本设计的重点是在 $Q_3Q_2Q_1Q_0=1001$ 状态下产生一个低电平信号，送给同步置数输入端 LD。表 5 - 47 是求 LD 置数信号的真值表。图 5 - 121 是求 LD 置数信号的卡诺图。从卡诺图写出的 LD 表达式为

$$LD=\overline{Q_0^n}+\overline{Q_3^n}=\overline{Q_0^nQ_3^n}$$

根据此表达式画出逻辑电路图如图 5 - 122 所示。在图 5 - 122 中，$D_3D_2D_1D_0$ 接入 0011。也可以用图 5 - 123 所示的电路来实现这个计数器。

表 5 - 47　　［例 5 - 15］的求 LD 置数信号的真值表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	LD	Q_3^n	Q_2^n	Q_1^n	Q_0^n	LD
0	0	0	0	×	1	0	0	0	1
0	0	0	1	×	1	0	0	1	0
0	0	1	0	×	1	0	1	0	×
0	0	1	1	1	1	0	1	1	×
0	1	0	0	1	1	1	0	0	×
0	1	0	1	1	1	1	0	1	×
0	1	1	0	1	1	1	1	0	×
0	1	1	1	1	1	1	1	1	×

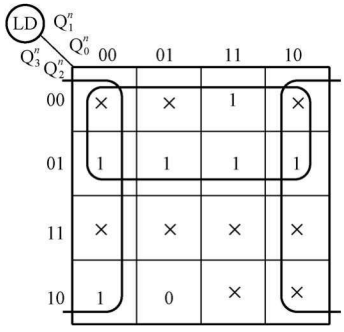


图 5 - 121　［例 5 - 15］的求 LD 置数信号的卡诺图

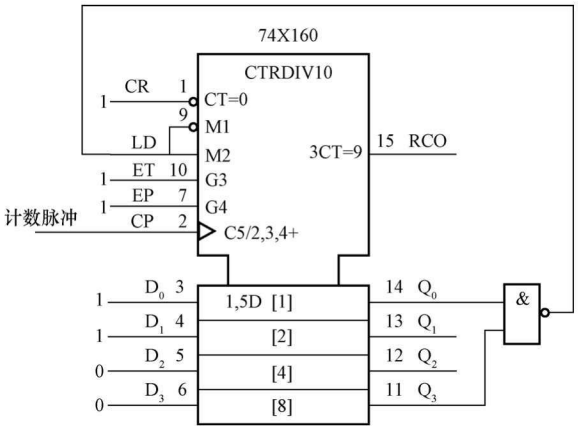


图 5 - 122　［例 5 - 15］的逻辑电路图（用 Q 端来实现）

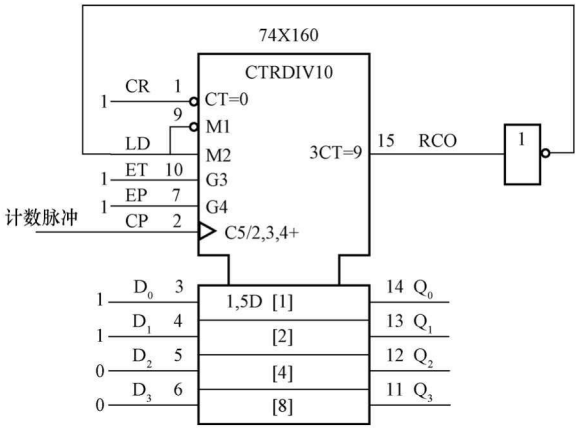


图 5 - 123　［例 5 - 15］的逻辑电路图（用 RCO 端来实现）

在图 5 - 123 中，当 74X160 计数到 1001 状态且 ET＝1 时产生进位信号 RCO＝1，把此进位信号通过非门，反馈到预置数控制端。预置数据输入端置成 $D_3D_2D_1D_0$＝0011 状态。

假设 74X160 的初态为 $Q_3Q_2Q_1Q_0$＝0011，图 5 - 124 是在 CP 脉冲作用下的波形图。

在图 5 - 124 中，该电路从 $Q_3Q_2Q_1Q_0$＝0011 状态开始加 1 计数，输入第 6 个 CP 脉冲后到达 1001 状态，产生进位信号 RCO＝1，LD＝0；因为是同步置数，所以在第 7 个 CP 脉冲作用后，$Q_3Q_2Q_1Q_0$ 被置成 0011 状态，同时 RCO＝0，LD＝1，又开始新的计数周期。图 5 - 125是其完整的状态图。

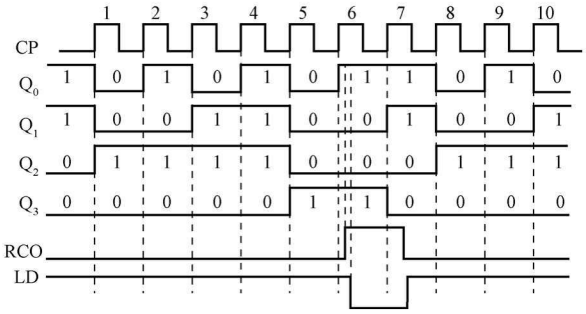


图 5 - 124 ［例 5 - 15］的波形图

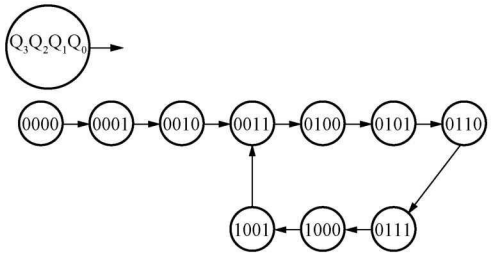


图 5 - 125 ［例 5 - 15］的完整的状态图

（2）异步反馈置数法。现举例说明异步反馈置数法设计计数器的方法。

【例 5 - 16】 用集成计数器 74X193 和必要的门电路组成 10 进制计数器，要求用异步反馈置数法实现。

解 74X193 的“加”计数的状态图见图 5 - 89。74X193 本身有 16 个状态。10 进制计数器有 10 个状态，至于是哪 10 个状态，无所谓，只要状态的数目是 10 个即可，不妨取状态图如图 5 - 126 所示。此题目明确要求使用反馈置数法实现。

集成计数器 74X193 具有异步预置功能，在其计数过程中，不管它的输出处于哪一个状态，只要在异步预置控制端加入一低电平电压，使 LD＝0，74X193 就会把预置数输入端 $D_3D_2D_1D_0$ 的值置入 $Q_3Q_2Q_1Q_0$ 输出端。当预置数控制信号（LD＝0）消失后，74X193 就从被置入的状态开始重新计数。因为 74X193 具有异步预置功能，所以应该在 $Q_3Q_2Q_1Q_0$＝1101 下去置数。图 5 - 127 给出了带有 1101 状态的状态图。

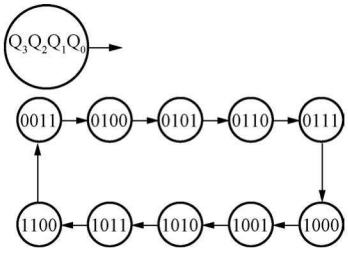


图 5 - 126 ［例 5 - 16］的状态图

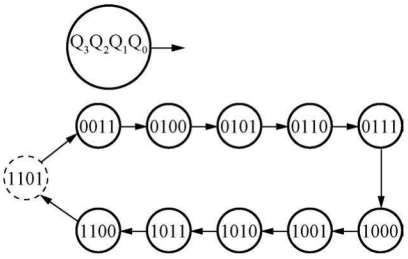


图 5 - 127 ［例 5 - 16］的带有 1101 状态的状态图

本设计要求电路在 $Q_3Q_2Q_1Q_0$ 为 1101 状态时产生一个低电平信号，送给异步置数输入端 LD。因为 74X193 是异步预置功能，所以必须在 $Q_3Q_2Q_1Q_0$ 为 1101 时产生一个低电平信号送给 LD，而不能在 $Q_3Q_2Q_1Q_0$ 为 1100 时产生一个低电平信号。可以利用前面的方法列

出求 LD 的真值表，然后用卡诺图进行化简，根据表达式画出逻辑电路图，如图 5 - 128 所示。当对这种方法已经足够熟悉后，可以利用观察法得出表达式。图 5 - 129 是图 5 - 128 所示电路的完整的状态图。这是一个可以自启动的电路。

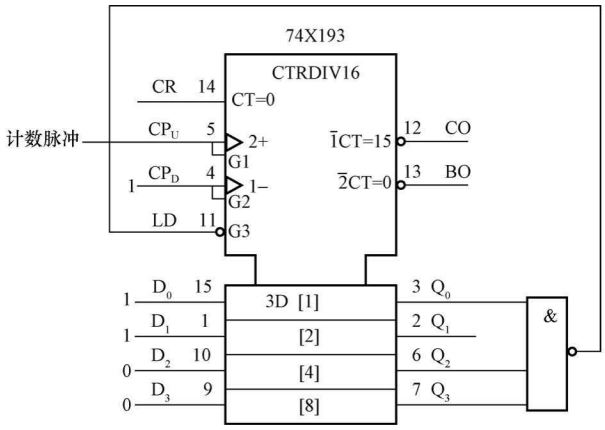


图 5 - 128 ［例 5 - 16］的逻辑电路图

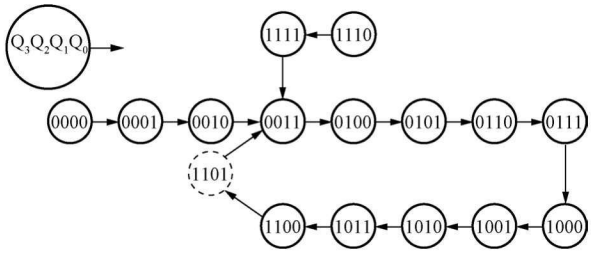


图 5 - 129 ［例 5 - 16］的完整的状态图

【例 5 - 17】 用 74X160 组成 48 进制计数器。

解 因为 $N=48$，$N=(48)_D=(0100\ 1000)_{8421BCD}$，而 74X160 为模 10 计数器，所以要用两片 74X160 构成此计数器。表 5 - 48 列出了要使用的 48 个状态。先将两芯片采用同步级联方式连接成 100 进制计数器，然后再借助 74X160 异步清零功能，在输入第 48 个计数脉冲后，计数器输出状态为 0100 1000 时，高位片（2）的 Q_2 和低位片（1）的 Q_3 同时为 1，使与非门输出 0，加到两芯片异步清零端上，使计数器立即返回 0000 0000 状态，状态 0100 1000 仅在极短的瞬间出现，为过渡状态。这样就组成了 48 进制计数器，其逻辑电路如图5 - 130所示。

表 5 - 48　　**［例 5 - 17］的状态表**

状态序号	Q_7^n	Q_6^n	Q_5^n	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_0^n	状态序号	Q_7^n	Q_6^n	Q_5^n	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_0^n
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	5	0	0	0	0	0	1	0	1
2	0	0	0	0	0	0	1	0	6	0	0	0	0	0	1	1	0
3	0	0	0	0	0	0	1	1	7	0	0	0	0	0	1	1	1

续表

状态序号	Q_7^n	Q_6^n	Q_5^n	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_0^n	状态序号	Q_7^n	Q_6^n	Q_5^n	Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_0^n
8	0	0	0	0	1	0	0	0	19	0	0	0	1	1	0	0	1
9	0	0	0	0	1	0	0	1	20	0	0	1	0	0	0	0	0
10	0	0	0	1	0	0	0	0	⋮			⋮			⋮		
11	0	0	0	1	0	0	0	1	29	0	0	1	0	1	0	0	1
12	0	0	0	1	0	0	1	0	30	0	0	1	1	0	0	0	0
13	0	0	0	1	0	0	1	1	⋮			⋮			⋮		
14	0	0	0	1	0	1	0	0	39	0	0	1	1	1	0	0	1
15	0	0	0	1	0	1	0	1	40	0	1	0	0	0	0	0	0
16	0	0	0	1	0	1	1	0	⋮			⋮			⋮		
17	0	0	0	1	0	1	1	1	47	0	1	0	0	0	1	1	1
18	0	0	0	1	1	0	0	0	48	0	1	0	0	1	0	0	0

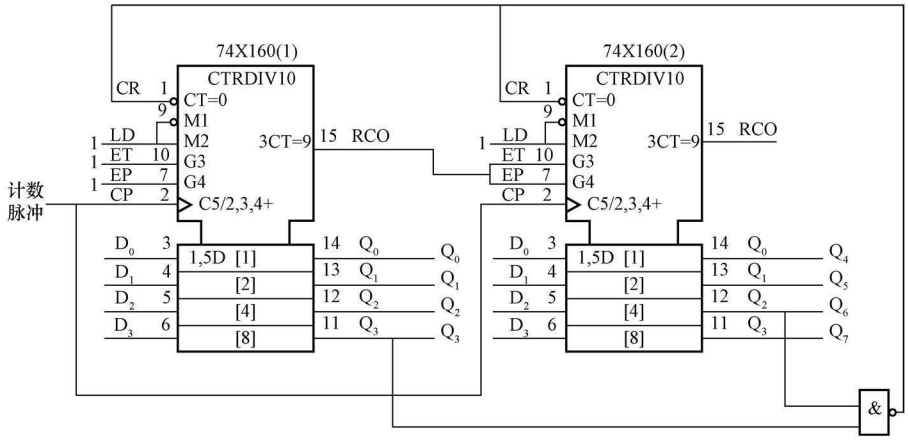


图 5 - 130 ［例 5 - 17］的逻辑电路图

还有另一种方法是大模分解法。如果 M 可以分解为多个因数相乘（每个因数小于单片计数器的最大值），则可先用 n 片计数器分别组成模值为 M_1，$M_2\cdots$，M_n 的计数器，然后将它们级联组成 $M=M_1\times M_2\times\cdots\times M_n$ 的计数器。

【例 5 - 18】 用 74X161 加必要的门电路实现一个 6 进制计数器的各种解法。

解 清零法一：用异步反馈清零方法实现，见［例 5 - 14］。

置数法二：用同步反馈置数（置 0000）方法实现。状态图如图 5 - 131 所示，逻辑电路图如图 5 - 132 所示。

置数法三：用同步反馈置数（置 0001）方法实现。状态图如图 5 - 133 所示，逻辑电路图如图 5 - 134 所示。

置数法四：用同步反馈置数（置 0010）方法实现。状态图如图 5 - 135 所示，逻辑电路图如图 5 - 136 所示。

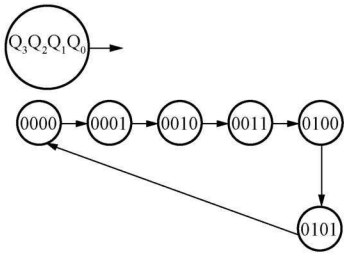


图 5 - 131　置数法二的状态图

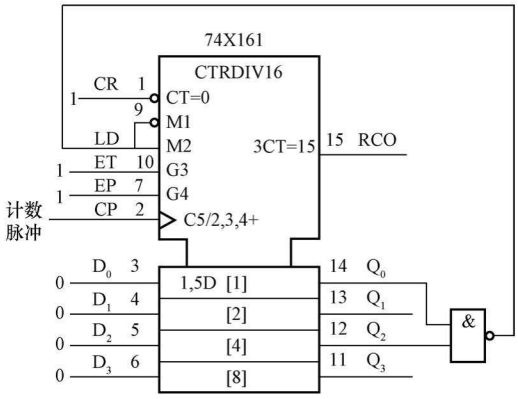


图 5 - 132　置数法二的逻辑电路图

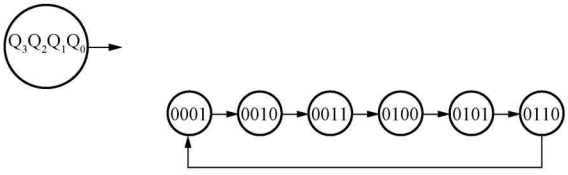


图 5 - 133　置数法三的状态图

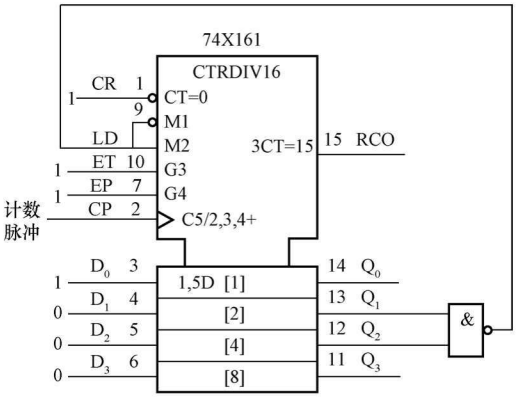


图 5 - 134　置数法三的逻辑电路图

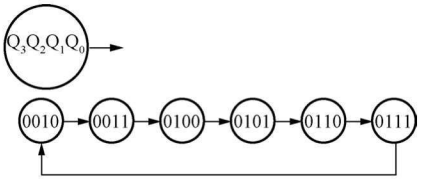


图 5 - 135　置数法四的状态图

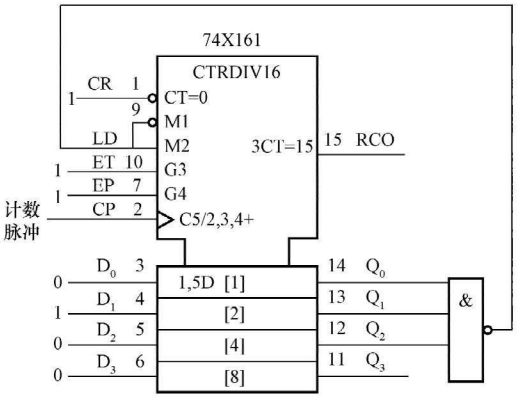


图 5 - 136　置数法四的逻辑电路图

置数法五：用同步反馈置数（置 1010）方法实现，使用 Q 信号。状态图如图 5 - 137 所示。图 5 - 138 是置数法五的逻辑电路图。

置数法六：用同步反馈置数（置 1010）方法实现，使用 RCO 信号。状态图如图 5 - 139 所示。图 5 - 140 是置数法六的逻辑电路图。

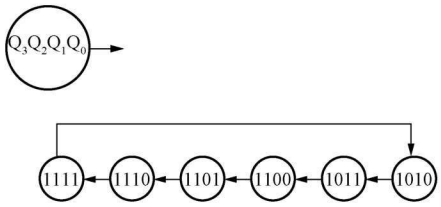


图 5 - 137　置数法五的状态图

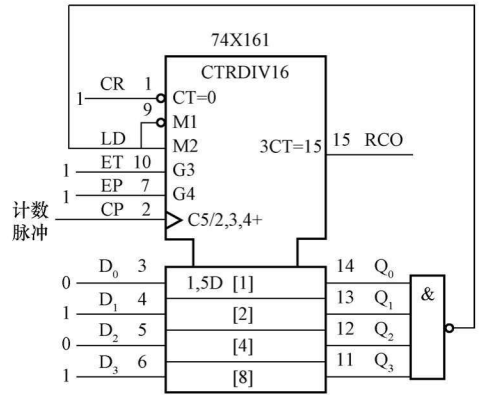


图 5 - 138　置数法五的逻辑电路图

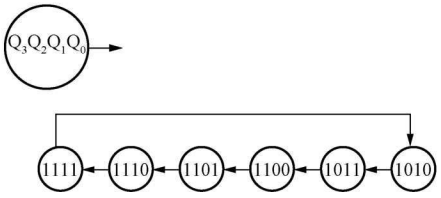


图 5 - 139　置数法六的状态图

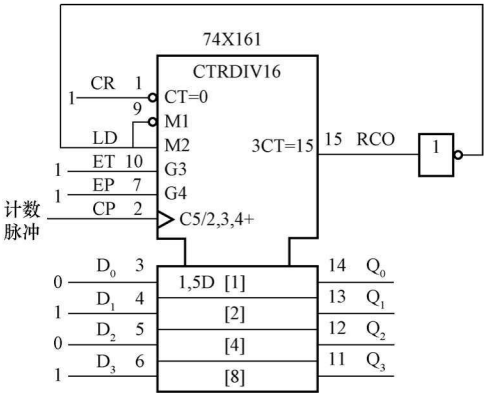


图 5 - 140　置数法六的逻辑电路图

置数法七：用同步反馈置数（置 1111）方法实现。状态图如图 5 - 141 所示。图 5 - 142 是置数法七的逻辑电路图。

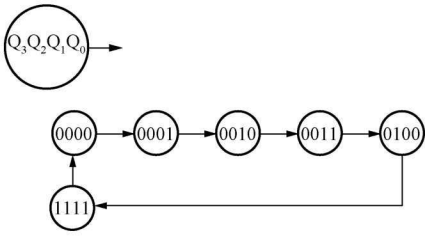


图 5 - 141　置数法七的状态图

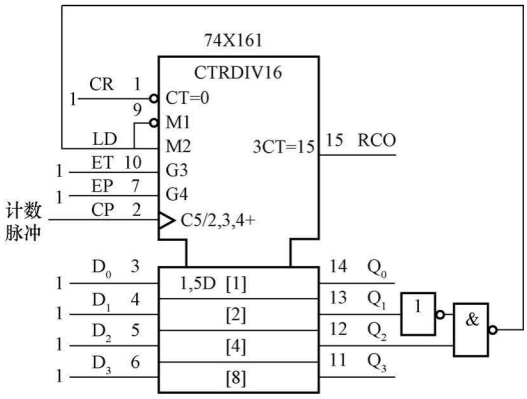


图 5 - 142　置数法七的逻辑电路图

除上述介绍的解法外，也可采用同步反馈置数法，分别置 0011、0100、0101、0110、0111、1000、1001、1011、1100、1101、1110 来实现电路设计要求。求解过程此处略，请读者自己分析。

（三）组成分频器

前面提到，模 M 计数器进位输出端输出脉冲的频率是输入脉冲频率的 $1/M$，因此可以用模 M 计数器组成 M 分频器。现举例说明。

【例 5 - 19】 某石英晶体振荡器输出脉冲信号的频率为 32768Hz，用 74X161 组成分频器，将其分频为频率为 1Hz 的脉冲信号。

解 因为 $32768=2^{15}$，经 15 级二分频，就可获得频率为 1Hz 的脉冲信号。因此将四片 74X161 级联，从高位片（4）的 Q_2 输出即可，其逻辑电路如图 5 - 143 所示。

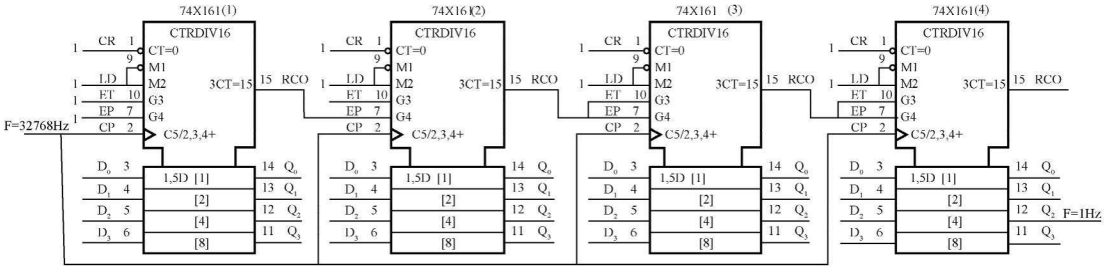


图 5 - 143 ［例 5 - 19］的逻辑电路图

（四）组成序列信号发生器

序列信号是在时钟脉冲作用下产生的按一定规则排列的一串周期性二进制信号。它通常用来作为数字系统的同步信号或地址码，也可以作为可编程逻辑电路的控制信号等，因此在通信、雷达、遥控、遥测、数字式噪声源等方面都有广泛的应用。

（1）用计数器辅以组合逻辑电路可以方便地构成各种序列发生器。构成的方法如下：

第一步，根据序列码的长度 P，设计一个模 P 计数器；第二步，设计适当的组合逻辑电路，计数器的状态做为该组合逻辑电路的输入，要设计的序列就是该组合逻辑电路的输出。

【例 5 - 20】 试分析计数器 74X161 及门电路构成的如图 5 - 144 所示逻辑电路的功能，并画出 Z 信号的波形。

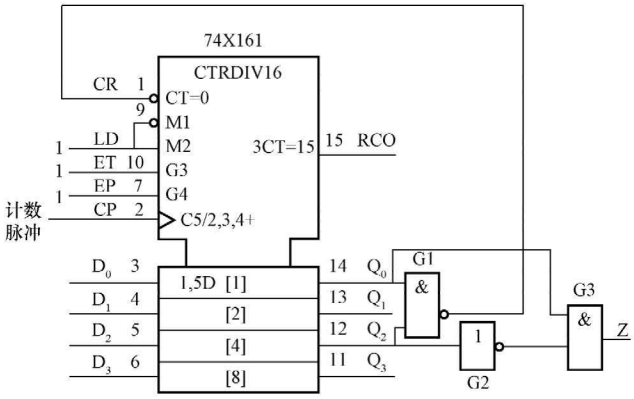


图 5 - 144 ［例 5 - 20］的逻辑电路图

解 74X161 和与非门 G1 构成了一个模 5 计数器，采用了异步清零功能。在 CP 作用下，计数器的状态变化如表 5 - 49 所示。由于 $Z=Q_0\overline{Q_2}$，故不同状态下的输出 Z 如表 5 - 50 的右列所示。因此，这是一个 01010 序列信号发生器，序列长度 $P=5$。Z 信号的波形图如图 5 - 145 所示。

表 5 - 49 ［例 5 - 20］的状态表

现态			次态		
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	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

表 5 - 50 ［例 5 - 20］的 Z 信号的真值表

现态			输出
Q_2^n	Q_1^n	Q_0^n	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0

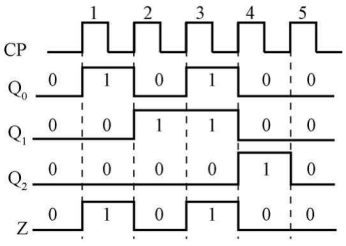


图 5 - 145 ［例 5 - 20］的 Z 信号的波形图

【例 5 - 21】 设计一个产生 110001001110 序列码的计数器型序列码发生器，要求组合逻辑电路部分用门电路实现。

解 （1）设计计数器。因序列长度 $P=12$，可选用 74X161 设计一个模 12 计数器。现采用同步预置法，计数器有效状态为 0100～1111。

（2）设计组合输出电路，求 Z 的真值表如表 5 - 51 所示，输入信号是计数器中四个触发器的状态信号。图 5 - 146 是求 Z 的卡诺图。图 5 - 147 是产生 110001001110 序列码的逻辑电路图。

表 5 - 51 ［例 5 - 21］的求 Z 的真值表

Q_3^n	Q_2^n	Q_1^n	Q_0^n	Z	Q_3^n	Q_2^n	Q_1^n	Q_0^n	Z
0	1	0	0	1	1	0	1	0	0
0	1	0	1	1	1	0	1	1	0
0	1	1	0	0	1	1	0	0	1
0	1	1	1	0	1	1	0	1	1
1	0	0	0	0	1	1	1	0	1
1	0	0	1	1	1	1	1	1	0

从图 5 - 146 可得出输出信号的逻辑函数表达式为

$$Z=Q_2^n\overline{Q_1^n}+\overline{Q_1^n}Q_0^n+Q_3^nQ_2^n\overline{Q_0^n}$$

用计数器辅以数据选择器可以方便地构成各种序列发生器。构成的方法如下：

第一步，根据序列码的长度 P，设计一个模 P 计数器；第二步，选择适当的数据选择

器，把欲产生的序列按规定的顺序加在数据选择器的数据输入端，把地址输入端与计数器的输出端适当地连接在一起。

由于计数器的状态设置和输出序列没有直接关系，因此这种结构更改输出序列比较方便，而且还能同时产生多组序列码。

【例 5-22】 设计一个产生 110001001110 序列码的计数器型序列码发生器，要求组合逻辑电路部分用 74X151 数据选择器实现。

解 （1）设计计数器。因序列长度 $P=12$，可选用 74X161 设计一个模 12 计数器。现采用同步预置法，计数器有效状态为 0100～1111。

（2）设计组合输出电路。令计数器计数过程中每一状态的输出符合给定序列的要求，可列出真值表如 5-52 所示。若采用 8 输入数据选择器 74X151 实现该组合逻辑函数，将 74X161 的 Q_3、Q_2、Q_1 作用于数据选择器的地址输入端 A_2、A_1、A_0，则选择器的数据输入端 $D_0=D_1=D_3=D_5=0$，$D_2=D_6=1$，且 $D_4=Q_0$，$D_7=\overline{Q_0}$，其逻辑电路图如图 5-148 所示。

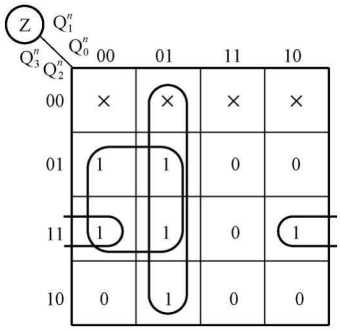


图 5-146 ［例 5-21］的求 Z 的卡诺图

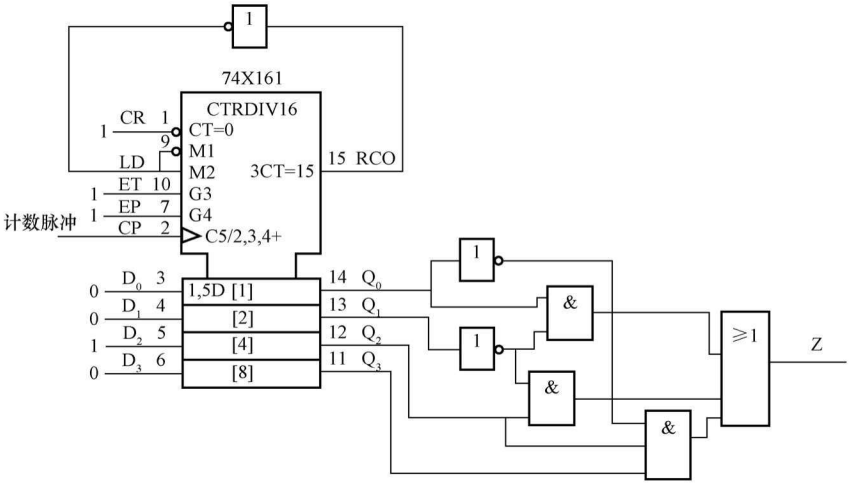


图 5-147 ［例 5-21］的逻辑电路图

表 5-52　　**［例 5-22］的真值表**

Q_3	Q_2	Q_1	Q_0	Z	Q_3	Q_2	Q_1	Q_0	Z
0	1	0	0	1	1	0	1	0	0
0	1	0	1	1	1	0	1	1	0
0	1	1	0	0	1	1	0	0	1
0	1	1	1	0	1	1	0	1	1
1	0	0	0	0	1	1	1	0	1
1	0	0	1	1	1	1	1	1	0

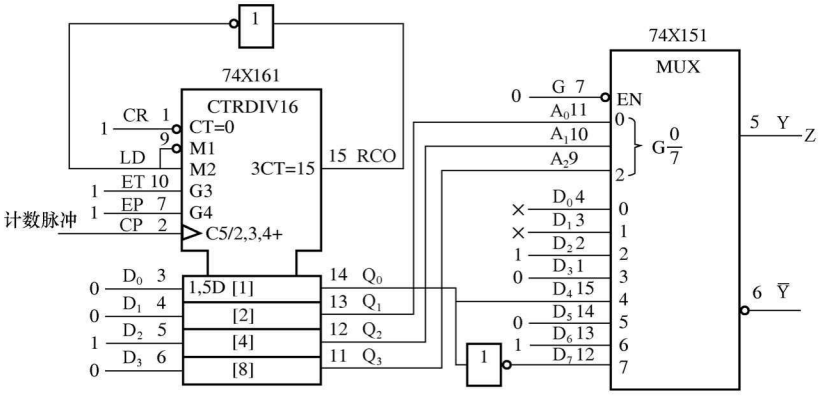


图 5 - 148 ［例 5 - 22］计数器型序列码发生器的逻辑电路图

（五）组成顺序脉冲发生器

在数控装置和数字计算机中，往往需要机器按照人们事先规定的顺序进行运算或操作，这就要求机器的控制部分不仅能正确地发出各种控制信号，而且要求这些控制信号在时间上有一定的先后顺序。通常采取的方法是，用一个顺序脉冲发生器来产生时间上有先后顺序的脉冲，以控制系统各部分协调地工作。

在数字电路中，能按一定时间、一定顺序轮流输出脉冲波形的电路称为顺序脉冲发生器。顺序脉冲发生器也称脉冲分配器或节拍脉冲发生器，一般由计数器（包括移位寄存器型计数器）和译码器组成。作为时间基准的计数脉冲由计数器的输入端送入，译码器将计数器状态译成输出端上的顺序脉冲，使输出端上的状态按一定时间、一定顺序轮流为 1，或者轮流为 0。

图 5 - 149 为一个由计数器 74X161 和译码器 74X138 组成的脉冲分配器。74X161 构成模 8 计数器，输出状态 $Q_2Q_1Q_0$ 在 000～111 之间循环变化，从而在译码器输出端 Y_0～Y_7 分别得到图 5 - 150 所示的脉冲序列。

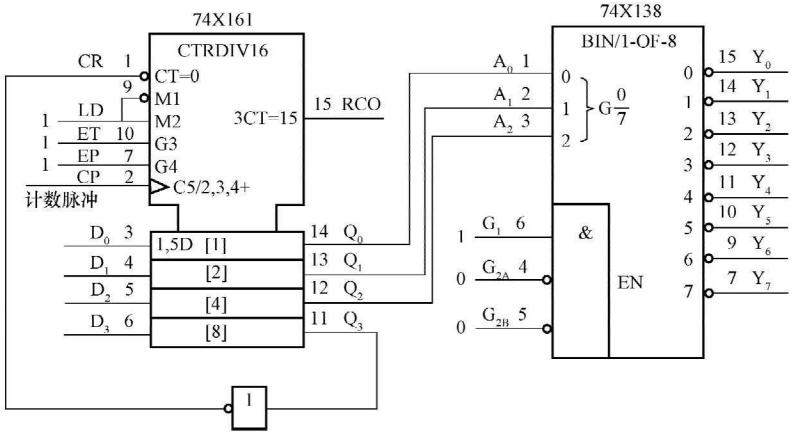


图 5 - 149 由计数器 74X161 和译码器 74X138 组成的脉冲分配器

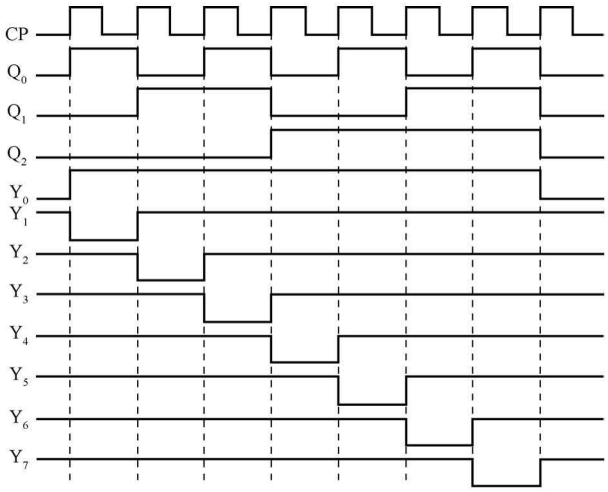


图 5 - 150　由计数器 74X161 和译码器 74X138 组成的脉冲分配器脉冲序列

第六节　数码寄存器与移位寄存器

寄存器是用来存放二进制数据或代码的电路，是一种基本时序电路。任何现代数字系统都必须把需要处理的数据和代码先寄存起来，以便随时取用。

寄存器是由具有存储功能的触发器组合起来构成的。一个触发器可以存储 1 位二进制代码，存放 n 位二进制代码的寄存器，需用 n 个触发器来构成。

按照功能的不同，可将寄存器分为基本寄存器和移位寄存器两大类。基本寄存器只能并行送入数据，需要时也只能并行输出。移位寄存器中的数据可以在移位脉冲作用下依次逐位右移或左移，数据既可以并行输入、并行输出，也可以串行输入、串行输出，还可以并行输入、串行输出，串行输入、并行输出，十分灵活。

寄存器的应用很广，特别是移位寄存器，不仅可将串行数码转换成并行数码，或将并行数码转换成串行数码，还可以很方便地构成移位寄存器型计数器和顺序脉冲发生器等电路。

一、数码寄存器

数码寄存器——存储二进制数码的时序电路组件，具有接收和寄存二进制数码的逻辑功能。前面介绍的各种集成触发器，就是一种可以存储一位二进制数的寄存器，用 n 个触发器就可以存储 n 位二进制数。

（一）单拍工作方式数码寄存器

图 5 - 151 所示是一个由 D 触发器组成的单拍工作方式寄存器，有四个数码输入端 D_3、D_2、D_1、D_0，四个数码输出端 Q_3、Q_2、Q_1、Q_0，一个送数控制端 CP。

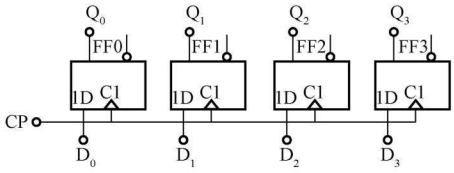


图 5 - 151　由 D 触发器组成的单拍工作方式寄存器

D 触发器的特性方程为

$$Q^{n+1} = D$$

所以，无论寄存器中原来的内容是什么，只要送数控制时钟脉冲 CP 上升沿到来，加在并行数据输入

端的数据 D_3、D_2、D_1、D_0，就立即被送进寄存器中，即

$$Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}=D_3D_2D_1D_0$$

由于这种电路一步就完成了对输入数码的接收，故称为单拍工作方式。

（二）双拍工作方式数码寄存器

如图 5 - 152 所示是一个由 D 触发器组成的双拍工作方式寄存器，有四个数码输入端 D_3、D_2、D_1、D_0，四个数码输出端 Q_3、Q_2、Q_1、Q_0，一个送数控制端 CP，一个清零端 CR。其接收数码过程如下：

（1）先异步清零，用一个负脉冲（CR=0）将所有触发器复位到 0 状态，即

$$Q_3^nQ_2^nQ_1^nQ_0^n=0000$$

（2）后送数，CR=1 时，用一个正脉冲（CP 上升沿）把输入数码放进寄存器中，并保存起来，即

$$Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}=D_3D_2D_1D_0$$

（3）在 CR=1、CP 上升沿以外时间，寄存器内容将保持不变。

由于整个工作过程是分两步进行的，故称为双拍工作方式。

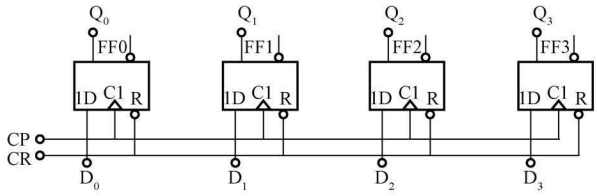


图 5 - 152　由 D 触发器组成的双拍工作方式寄存器

（三）4 位集成寄存器 74X175

74X175 是由 D 触发器组成的 4 位集成寄存器，图 5 - 153 是 74X175 的逻辑符号。图 5 - 154 是 74X175 的内部逻辑电路图，CR 是异步清零控制端。D_3、D_2、D_1 和 D_0 是并行数据输入端，CP 为时钟脉冲端，Q_3、Q_2、Q_1、Q_0 是并行数据输出端，$\overline{Q_3}$、$\overline{Q_2}$、$\overline{Q_1}$、$\overline{Q_0}$ 分别是 Q_3、Q_2、Q_1、Q_0 的反码数据输出端。

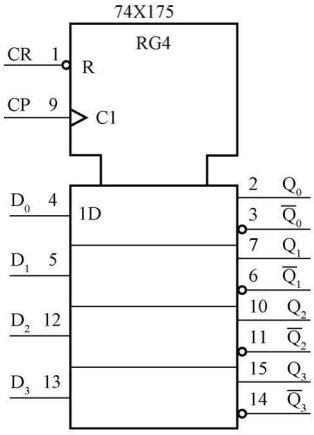


图 5 - 153　74X175 的逻辑符号

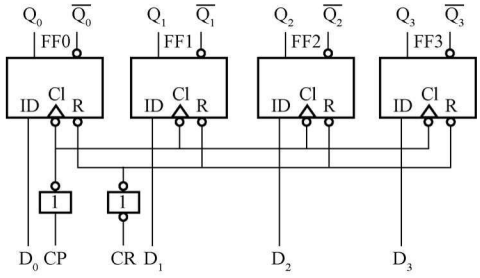


图 5 - 154　74X175 的内部逻辑电路图

该电路的数码接收过程为：将需要存储的 4 位二进制数码送到数据输入端 D_3、D_2、D_1 和 D_0，在 CP 端送一个时钟脉冲，脉冲上升沿作用后，4 位数码并行地出现在四个触发器 Q_3、Q_2、Q_1 和 Q_0 端。

74X175 的功能示于表 5 - 53。其逻辑功能如下：

（1）异步清零。当 CR＝0 时，不管其他输入端的状态如何，不论有无时钟脉冲 CP，寄存器输出将被直接置零（$Q_3Q_2Q_1Q_0$＝0000）。

（2）数码寄存。当 CR＝1 时，在输入时钟脉冲 CP 上升沿的作用下，并行输入端的数据 D_3、D_2、D_1 和 D_0 被置入寄存器的输出端，即 $Q_3Q_2Q_1Q_0$＝DCBA。

（3）数据保持。当 CR＝1，且无时钟脉冲 CP 上升沿，则寄存器保持原来的状态不变。

表 5 - 53　　74X175 的功能表

清零	时钟	输入				输出				工作模式
CR	CP	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	
0	×	×	×	×	×	0	0	0	0	异步清零
1	↑	D	C	B	A	D	C	B	A	数码寄存
1	1	×	×	×	×	保持				数据保持
1	0	×	×	×	×	保持				数据保持

二、移位寄存器

移位寄存器不但可以寄存数码，而且在移位脉冲作用下，寄存器中的数码可根据需要向左或向右移动 1 位。移位寄存器也是数字系统和计算机中应用很广泛的基本逻辑部件。

（一）单向移位寄存器

1. 右向移位寄存器

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 5 - 155 所示。数据从串行输入端 D_I输入，左边触发器的输出作为右邻触发器的数据输入，各触发器都受时钟脉冲 CP 的控制。

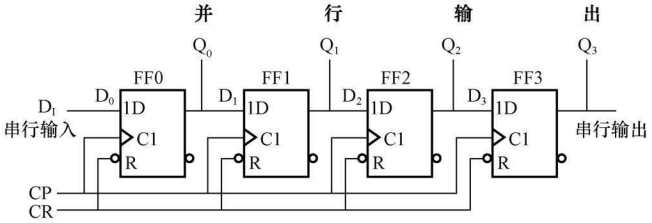


图 5 - 155　D 触发器组成的 4 位右移寄存器

时钟方程为

$$CP_0 = CP_1 = CP_2 = CP_3 = CP$$

驱动方程为

$$D_0 = D_I、D_1 = Q_0^n、D_2 = Q_1^n、D_3 = Q_2^n$$

状态方程为

$$Q_0^{n+1} = D_I、Q_1^{n+1} = Q_0^n、Q_2^{n+1} = Q_1^n、Q_3^{n+1} = Q_2^n$$

在时钟脉冲作用后，由于D触发器的输出Q由CP上升沿到来之前瞬时的输入D决定，因此每来一个时钟脉冲，左边触发器的输出状态就会移到右邻触发器中去，使数据向右移动一位。

设串行输入数码为1101，并从左至右依次送到D_I端，每位数码维持一个时钟周期长，如图5-156所示。在移位寄存器开始工作前，在CR清零端加低电平，使移位寄存器的初始状态为0000。当第一个时钟脉冲的上升沿到来时，$D_I=1$送入触发器FF0中，使$Q_0=1$；当第二个时钟脉冲作用后，$Q_0=1$移入触发器FF1中，使$Q_1=1$，且$D_I=1$送入触发器FF0中，使$Q_0=1$。依此类推，在4个移位脉冲作用后，输入的4位串行数码1101全部存入了寄存器中，使$Q_3=1$、$Q_2=1$、$Q_1=0$、$Q_0=1$。电路的状态表如表5-54所示。

表5-54　　右移寄存器的状态表

移位脉冲	输入数码	输出				移位脉冲	输入数码	输出			
CP	D_I	Q_0	Q_1	Q_2	Q_3	CP	D_I	Q_0	Q_1	Q_2	Q_3
0		0	0	0	0	3	0	0	1	1	0
1	1	1	0	0	0	4	1	1	0	1	1
2	1	1	1	0	0						

由表5-54可知，输入数码依次由左边触发器移到右邻触发器，所以是右向移位寄存器。经过4个时钟脉冲后，数码1101出现在触发器的输出端$Q_3=1$、$Q_2=1$、$Q_1=0$、$Q_0=1$，这样便将由D_I端串行输入的数码转换为并行输出的数码。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。

时序图如图5-156所示。由图可见，在第8个时钟脉冲作用后，数码已从Q_3端全部移出寄存器，即数码可由Q_3端串行输出。所以，移位寄存器具有串行输入—并行输出和串行输入—串行输出两种工作方式。

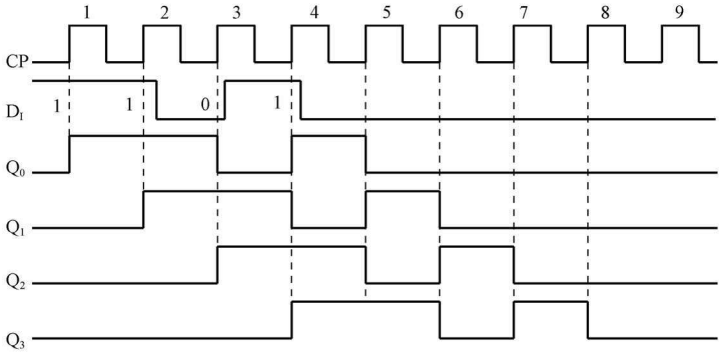


图5-156　图5-155电路的时序图

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。

2. 左向移位寄存器

若将图 5 - 155 所示电路中各触发器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，数据从最右边触发器串行输入端 D_I 输入，则可组成图 5 - 157 所示的左向移位寄存器。

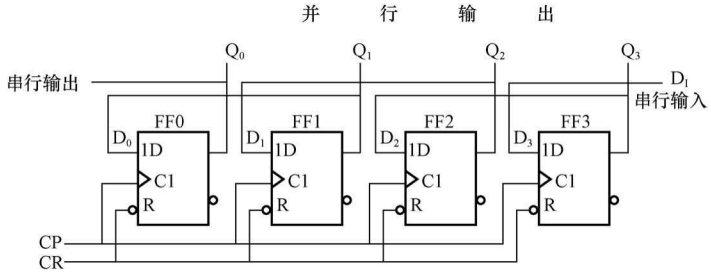


图 5 - 157　D 触发器组成的 4 位左移寄存器

单向移位寄存器具有以下主要特点：

(1) 单向移位寄存器中的数码，在 CP 脉冲操作下，可以依次右移或左移。

(2) n 位单向移位寄存器可以寄存 n 位二进制代码。n 个 CP 脉冲即可完成串行输入工作，此后可从 $Q_0 \sim Q_{n-1}$ 端获得并行的 n 位二进制数码，再用 n 个 CP 脉冲又可实现串行输出操作。

(3) 若串行输入端状态为 0，则 n 个 CP 脉冲后，寄存器便被清零。

(二) 双向移位寄存器

将图 5 - 155 所示的右移寄存器和图 5 - 157 所示的左移寄存器组合起来，并引入一控制端 S 便构成既可左移又可右移的双向移位寄存器，如图 5 - 158 所示。图中，每个触发器的数据输入端 D 同与或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 状态。D_{SR} 为右移串行输入端，D_{SL} 为左移串行输入端。

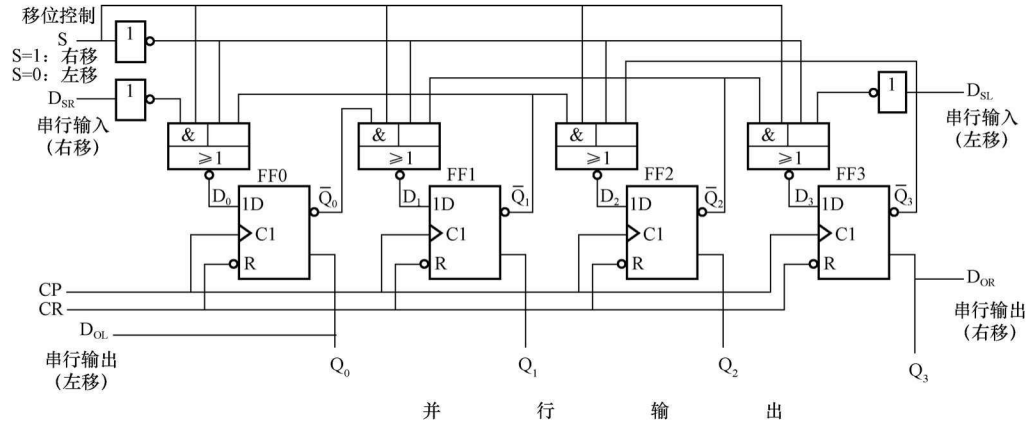


图 5 - 158　D 触发器组成的 4 位双向移位寄存器

由图 5 - 158 可知该电路的驱动方程为

$$D_0 = \overline{S\,\overline{D_{SR}} + \overline{S}\,\overline{Q_1}}$$

$$D_1 = \overline{S\,\overline{Q_0} + \overline{S}\,\overline{Q_2}}$$

$$D_2 = \overline{S\,\overline{Q_1} + \overline{S}\,\overline{Q_3}}$$

$$D_3 = \overline{S\,\overline{Q_2} + \overline{S}\,\overline{D_{SL}}}$$

当 S=1 时，$D_0 = D_{SR}$、$D_1 = Q_0$、$D_2 = Q_1$、$D_3 = Q_2$，相当于触发器 FF0、FF1、FF2、FF3 的 D 端分别与 D_{SR}、Q_0、Q_1、Q_2 相连，在 CP 脉冲作用下，实现右移操作；

当 S=0 时，$D_0 = Q_1$、$D_1 = Q_2$、$D_2 = Q_3$、$D_3 = D_{SL}$，相当于触发器 FF0、FF1、FF2、FF3 的 D 端分别与 Q_1、Q_2、Q_3、D_{SL}相连，在 CP 脉冲作用下，实现左移操作。

三、4 位双向集成移位寄存器 74X194

集成移位寄存器 74X194 是由 4 个 RS 触发器组成的功能很强的 4 位移位寄存器。

图 5 - 159 是加了引脚名字的 74X194 的逻辑符号。图 5 - 160 是 74X194 的内部逻辑电路图。74X194 芯片共有 16 个引脚。8 脚和 16 脚分别是地和电源。74X194 芯片内部有 4 个触发器，12 脚 Q_3、13 脚 Q_2、14 脚 Q_1、15 脚 Q_0 是这四个触发器的状态信号。11 脚 CP 是时钟脉冲信号。一般情况下，生产厂家为了扩大芯片的使用范围，在实现芯片基本功能的基础上，增加了一些控制功能。1 脚 CR 是清零端，可以实现芯片的置 $Q_3Q_2Q_1Q_0=0000$ 状态的功能。9 脚、10 脚 S_0、S_1 是控制端，共有 4 种组合，6 脚 D_3、5 脚 D_2、4 脚 D_1、3 脚 D_0 预置数据输入端和 S_0、S_1 控制端一起共同完成芯片的置 $Q_3Q_2Q_1Q_0$ 为任意状态的功能，2 脚 D_{SR}为右移串行数据输入端，7 脚 D_{SL}为左移串行数据输入端。

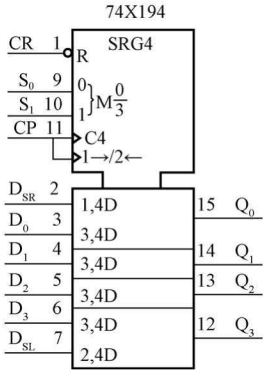


图 5 - 159　加了引脚名字的 74X194 的逻辑符号

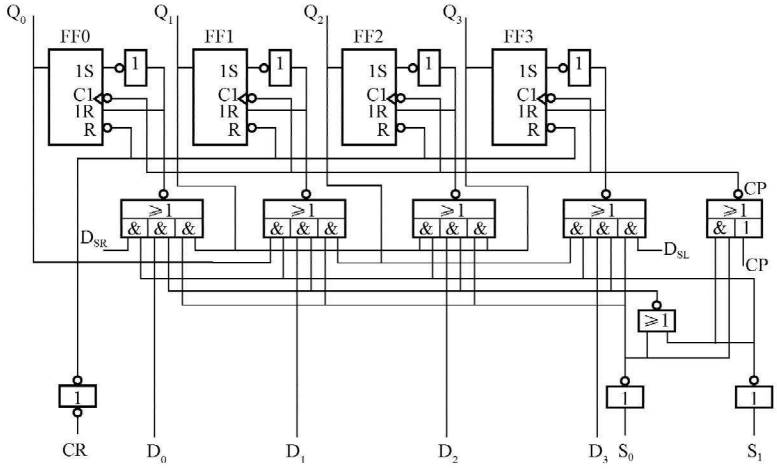


图 5 - 160　74X194 的内部逻辑电路图

以触发器 FF1 为例，由 RS 触发器的特性方程可得

$$Q_1^{n+1} = S + \overline{R}Q_1^n$$

由于 $S=\overline{R}$，所以

$$Q_1^{n+1} = \overline{R} + \overline{R}Q_1^n = \overline{R}(1+Q_1^n) = \overline{R} = \overline{S_0}Q_2^n + \overline{S_1}Q_0^n + S_0S_1D_1$$

由上式可知：当 $S_1=1$、$S_0=0$ 时，$Q_1^{n+1}=Q_2^n$，即为左移寄存器；当 $S_1=0$、$S_0=1$ 时，$Q_1^{n+1}=Q_0^n$，即为右移寄存器；当 $S_1=S_0=1$ 时，$Q_1^{n+1}=D_1$，即有并行存入功能；当 $S_1=S_0=0$ 时，CP 被封锁，触发器维持原态不变，使寄存器具有保持功能。

74X194 功能表如表 5 - 55 所示。由表 5 - 55 可以看出 74X194 具有如下功能。

（1）异步清零。当 CR＝0 时清零，与其他输入状态及 CP 无关。

（2）S_1S_0 是控制输入。当 CR＝1 时 74X194 有如下 4 种工作方式：

1）当 S_1S_0＝00 时，不论有无 CP 到来，各触发器状态不变，为保持工作状态。

2）当 S_1S_0＝01 时，在 CP 的上升沿作用下，实现右移操作，流向是 $D_{SR}\rightarrow Q_0\rightarrow Q_1\rightarrow Q_2\rightarrow Q_3$。

3）当 S_1S_0＝10 时，在 CP 的上升沿作用下，实现左移操作，流向是 $D_{SL}\rightarrow Q_3\rightarrow Q_2\rightarrow Q_1\rightarrow Q_0$。

4）当 S_1S_0＝11 时，在 CP 的上升沿作用下，实现置数操作 $D_0\rightarrow Q_0$，$D_1\rightarrow Q_1$，$D_2\rightarrow Q_2$，$D_3\rightarrow Q_3$。

表 5 - 55　　74X194 的功能表

输入										输出				工作模式
清零	控制		串行输入		时钟	并行输入								
CR	S_1	S_0	D_{SL}	D_{SR}	CP	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3	
0	×	×	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	0	0	×	×	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n	保持
1	0	1	×	1	↑	×	×	×	×	1	Q_0^n	Q_1^n	Q_2^n	右移 D_{SR} 为串行输入，Q_3 为串行输出
			×	0	↑	×	×	×	×	0	Q_0^n	Q_1^n	Q_2^n	
1	1	0	1	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	1	左移 D_{SL} 为串行输入，Q_0 为串行输出
			0	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	0	
1	1	1	×	×	↑	A	B	C	D	A	B	C	D	并行置数

74X194 芯片的控制信号的设计与其他计数器芯片稍有不同，74X194 芯片没有单独设计一个并行置数端，而是使用了 4 个模式中的一种作为并行置数功能；也没有单独设计使能端，而是使用了 4 个模式中的一种作为保持功能。

下面详细介绍图 5 - 159 所示的 74X194 的逻辑符号。

SRG4 是总限定符号。SRG 是 Shift Register 的缩写；4 是指移位寄存器的位数，即 4 位移位寄存器。

74X194 芯片的 1 脚、9 脚、10 脚、11 脚属于国标逻辑符号的公共控制框内，表示对其他的信号都有控制作用。

74X194 芯片长方形框的外部 1 脚处画有小圆圈，表示是低电平有效，内部标有 R 字样，指复位为 0 的意思，而且是异步清零。在本书中为了使用起来方便，在长方形框的外部 1 脚处还标上了 CR 的字样，更清楚地表明是将 74X194 芯片内部的触发器的状态清为 0。

在本书中给 74X194 芯片的 10 脚和 9 脚分别标注了 S_1、S_0，在长方形框的内部 9 脚处写有“0”的字样，10 脚处写有“1”的字样，这分别是给这两个信号规定的位权。“0”和“1”的右边有一个大括号，表明这是一组信号。大括号的右边写有 M 的字样，M 是 Mode

的缩写，表示模式的意思，共有四种模式，用 M $\frac{0}{3}$ 表示。$S_1S_0=00$ 对应 M=0，$S_1S_0=01$ 对应 M=1，$S_1S_0=10$ 对应 M=2，$S_1S_0=11$ 对应 M=3，S_1S_0 的 4 种组合决定了 74X194 的 4 种操作方式，如表 5-56 所列。

表 5-56　S_1S_0 的 4 种组合操作

S_1S_0	00	01	10	11
操作模式	保持	右移	左移	置数

74X194 芯片的 11 脚是时钟输入端，在长方形框的外部本书标注了 CP 字样的名字，在长方形框的 11 脚的内部标有“>C4”，“>”表示上升沿有效的动态输入，即我们常说的时钟信号，“>C4”中的 C 表示控制的意思，4 表示给定了一个关联编号为 4。

74X194 芯片的 11 脚的内部还标有“>1→/2←”，其中，“1→”中的 1 是 M=1，“→”表示在模式 1 下（$S_1S_0=01$）右移（Shift Right）的意思。“2←”中的 2 是 M=2，“←”表示在模式 2 下（$S_1S_0=10$）左移（Shift Left）的意思。

74X194 芯片的 2 脚的外部标有 D_{SR} 字样，这是本书给定的信号的名字。在长方形框的 2 脚的内部标有“1，4D”的字样，其中的 1 表示模式 1 即 M=1，模式 1 为右移模式；4 表示受标号为 4 的信号即时钟信号控制，也就是说在时钟脉冲的上升沿这个输入端的数据送入 74X194 芯片的 15 脚。

同理，74X194 芯片的 7 脚的外部标有 D_{SL} 字样。在长方形框的 7 脚的内部标有“2，4D”的字样，其中的 2 表示模式 2 即 M=2，模式 2 为左移模式；4 表示受标号为 4 的信号即时钟信号控制，也就是说在时钟脉冲的上升沿这个输入端的数据送入 74X194 芯片的 12 脚。

74X194 芯片的 3 脚、4 脚、5 脚、6 脚的外部分别标有 D_0、D_1、D_2、D_3 字样。在长方形框的 3 脚、4 脚、5 脚、6 脚的内部都标有“3，4D”的字样，其中的 3 表示模式 3 即 M=3，模式 3 为并行置数模式；4 表示受标号为 4 的信号即时钟信号控制，也就是说在时钟脉冲的上升沿将 3 脚、4 脚、5 脚、6 脚的数据送入 74X194 芯片的 15 脚、14 脚、13 脚、12 脚。

用 74X194 位双向移位寄存器可以方便地组成 8 位双向移位寄存器，这时只要将其中一片的最低位输出 Q_0 与另一片的 D_{SR} 相连，将另一片的最高位输出 Q_3 与这一片的 D_{SL} 相连，同时再将两片的 S_1、S_0、CP 和 CR 并联起来。

四、74X194 的应用

（一）移位寄存器构成的移位型计数器

在实际应用中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，称为移位型计数器。

典型的移位型计数器有环型和扭环型计数器。它们的共同特点是结构简单，状态变化顺序遵循移位的规律，但状态利用率低，存在无效循环。

1. 环形计数器

图 5-161 和图 5-162 分别是用 74X194 构成的环形计数器的逻辑电路图和状态图。当正脉冲启动信号 START 到来时，使 $S_1S_0=11$，从而不论移位寄存器 74X194 的原状态如

何，在 CP 作用下总是执行置数操作使 $Q_0Q_1Q_2Q_3=1000$。当 START 由 1 变 0 之后，$S_1S_0=01$，在 CP 作用下移位寄存器进行右移操作。在第四个 CP 到来之前 $Q_0Q_1Q_2Q_3=0001$。这样在第四个 CP 到来时，由于 $D_{SR}=Q_3=1$，故在此 CP 作用下 $Q_0Q_1Q_2Q_3=1000$。可见该计数器共四个状态，为模 4 计数器。

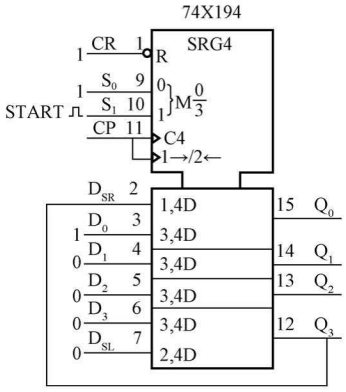


图 5 - 161　用 74X194 构成的环形计数器的逻辑电路图

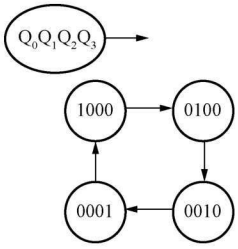


图 5 - 162　用 74X194 构成的环形计数器的状态图

环形计数器的电路十分简单，N 位移位寄存器可以计 N 个数，实现模 N 计数器，且状态为 1 的输出端的序号即代表收到的计数脉冲的个数，通常不需要任何译码电路。它的缺点是没有自启动能力。

2. 扭环形计数器

为了增加有效计数状态，扩大计数器的模，将上述接成右移寄存器的 74X194 的末级输出 Q_3 反相后，接到串行输入端 D_{SR}，就构成了扭环形计数器，如图 5 - 163 所示。图 5 - 164 为其状态图，由图可见该电路有 8 个计数状态，为模 8 计数器。一般来说，N 位移位寄存器可以组成模 $2N$ 的扭环形计数器，只需将末级输出反相后，接到串行输入端。

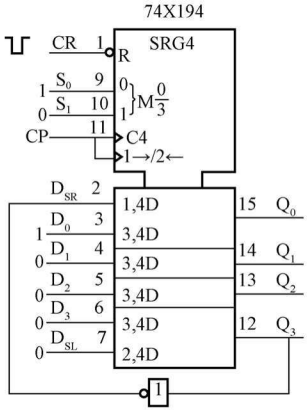


图5 - 163　用 74X194 构成的扭环形计数器逻辑电路图

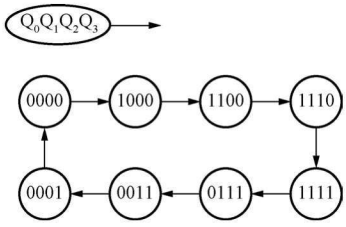


图 5 - 164　用 74X194 构成的扭环形计数器的状态图

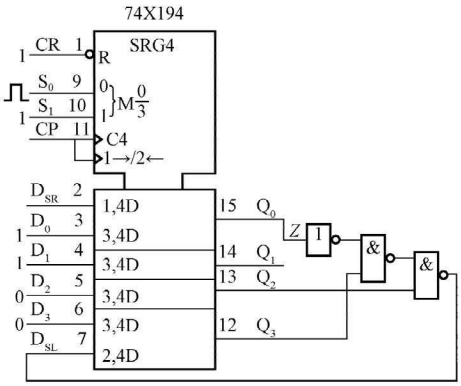


图 5 - 165　采用 74X194 移位寄存器构成的序列信号发生器

（二）移位寄存器构成的序列信号发生器

采用 74X194 移位寄存器构成的序列信号发生器如图 5 - 165 所示。

当在 9 脚输入正脉冲启动信号时，$S_1S_0=11$，移位寄存器完成并行置数操作，此时并行输出的数据 $Q_0Q_1Q_2Q_3=1100$。启动信号消除后，$S_1S_0=10$ 时，移位寄存器完成左移操作。左移串行输入数据 D_{SL} 为移位寄存器输出信号，此输出信号经过组合逻辑电路反馈后得到，即 $D_{SL}=\overline{Q_2\cdot\overline{Q_0}\cdot Q_3}$。据此，可以得到移位寄存器的状态表 5 - 57 和与之对应的状态图 5 - 166。从序列信号发生器 Z 端（即 Q_0 端）输出的序列码为 110011。

表 5 - 57　74X194 构成的序列信号发生器的状态表

CP	Q_0	Q_1	Q_2	Q_3	D_{SL}
0	1	1	0	0	1
1	1	0	0	1	1
2	0	0	1	1	1
3	0	1	1	1	1
4	1	1	1	1	0
5	1	1	1	0	0
6	1	1	0	0	1

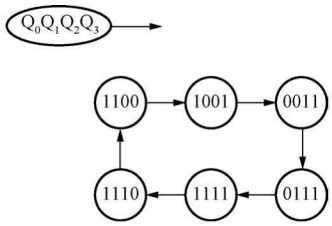


图 5 - 166　74X194 构成的序列信号发生器的状态图

小　　结

在某一时刻某时序逻辑电路的输出信号 z_1，z_2，…，z_j 不但与该时刻的输入信号 x_1，x_2，…，x_i 有关，还与该时刻以前的输入信号的取值有关。该时刻以前的输入信号的曾经取值表示为该时序逻辑电路现在的状态。状态是时序逻辑电路的一个非常重要的概念。

时序逻辑电路逻辑功能的描述方法有逻辑方程（输出方程、驱动方程和状态方程）、状态表、状态图和时序图等。

时序逻辑电路的分析步骤为逻辑图→时钟方程、驱动方程、输出方程→状态方程→状态转换真值表→状态转换图和时序图→逻辑功能。

时序逻辑电路的设计步骤为设计要求→原始状态表→最简状态表→驱动和输出表→驱动卡诺图和输出卡诺图→驱动方程、输出方程→逻辑电路图。

计数器是一种常用的时序逻辑器件，在计算机和其他数字系统中起着非常重要的作用。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。

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，采用的方法有异步清零法、同步清零法、异步置数法和同步置数法，根据集成计数器的清零方式和置数方式

来选择。当 $M>N$ 时，用 1 片 M 进制计数器即可；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。当需要扩大计数器的容量时，可将多片集成计数器进行级联。

寄存器也是一种常用的时序逻辑器件。寄存器分为数码寄存器和移位寄存器两种，移位寄存器又分为单向移位寄存器和双向移位寄存器。集成移位寄存器使用方便、功能全、输入和输出方式灵活。用移位寄存器可实现数据的串行—并行转换，组成环形计数器、扭环计数器、顺序脉冲发生器等。

掌握时序逻辑器件的国标逻辑符号的含义与应用。

习　　题

5 - 1　已知一时序电路的状态表如表 5 - 58 所示，试作出相应的状态图。

表 5 - 58　　题 5 - 1 表

S^{n+1}/Z 输入 / S^n	X	
	0	1
S_0	$S_3/1$	$S_1/1$
S_1	$S_3/1$	$S_2/1$
S_2	$S_3/1$	$S_2/1$
S_3	$S_1/1$	$S_2/1$

5 - 2　试分析图 5 - 167（a）所示时序逻辑电路，列出状态表，画出状态图。设电路的初始状态为 0，试画出在图 5 - 167（b）所示波形作用下，Q 和 Z 的波形图。

5 - 3　试分析图 5 - 168（a）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 5 - 168（b）所示波形作用下 Q 和 Z 的波形图。

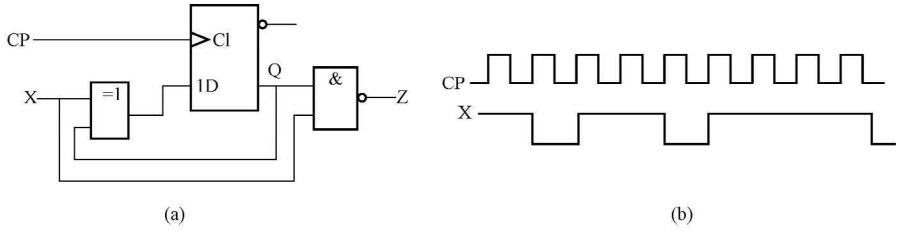


图 5 - 167　题 5 - 2 图

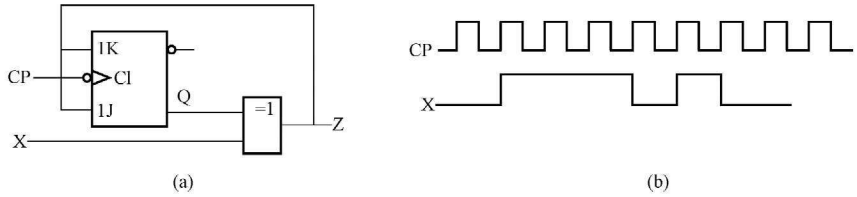


图 5 - 168　题 5 - 3 图

5 - 4　试分析图 5 - 169 所示时序逻辑电路，列出状态表，画出状态图和波形图，并说明其功能。

5 - 5　试分析图 5 - 170 所示的时序逻辑电路，写出它的驱动方程、状态方程，列出状态表和状态图，说明其功能。

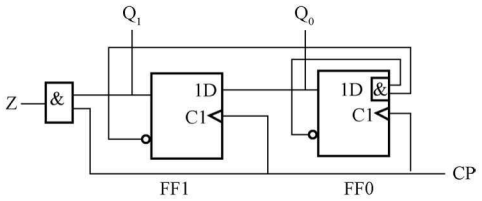


图 5 - 169　题 5 - 4 图

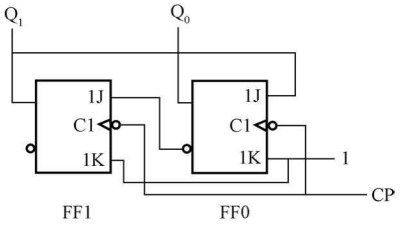


图 5 - 170　题 5 - 5 图

5 - 6　试分析图 5 - 171 所示的时序逻辑电路，写出它的驱动方程、状态方程，列出状态表和状态图，画出时序图，说明其功能。

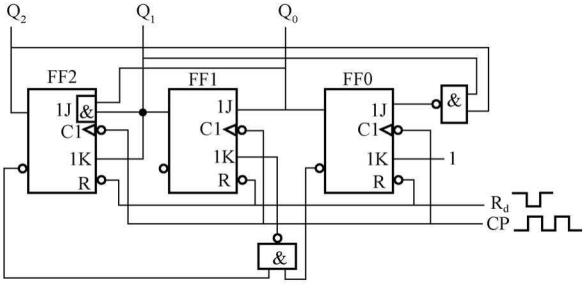


图 5 - 171　题 5 - 6 图

5 - 7　试画出如图 5 - 172（a）所示时序电路的状态图，并画出对应于图 5 - 172（b）所示 CP 波形的 Q_1、Q_0 和输出 Z 的波形。设电路的初始状态为 00。

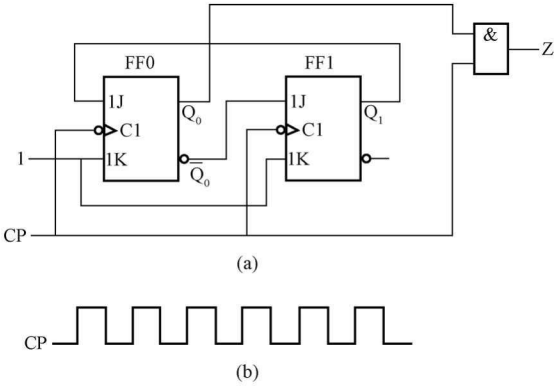


图 5 - 172　题 5 - 7 图

5 - 8　分析图 5 - 173 所示时序逻辑电路，画出其状态图，说明其功能。

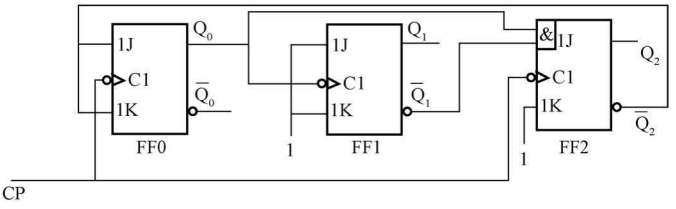


图 5 - 173　题 5 - 8 图

5 - 9　已知某序列码检测器的状态图如图 5 - 174 所示。试问：

（1）该电路的输入 X 为何种序列时输出 Z 才为 1？

（2）列出该电路的最简状态表。

5 - 10　试用 JK 触发器设计一同步时序电路，其状态图如图 5 - 175 所示，要求电路最简。

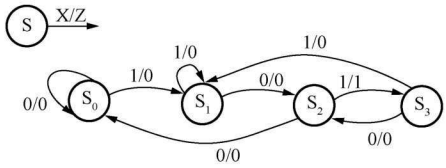


图 5 - 174　题 5 - 9 图

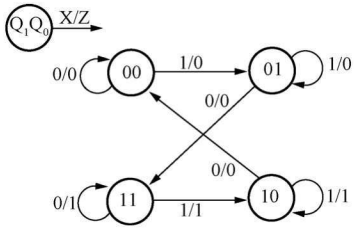


图 5 - 175　题 5 - 10 图

5 - 11　试用 JK 触发器设计一个同步七进制计数器，当计满时输出一个 0。

5 - 12　设计一个 4 位二进制数的串行奇校验电路。电路从复位状态开始，从单一的一根输入线 X 上串行接收 4 位数码。当第 4 位到达时，如果接收到偶数个 1，则电路输出为 1，否则为 0；而接收第 4 位以后，电路又回到初始状态。要求：

（1）画出状态图；

（2）列出状态表，并尽可能简化；

（3）写出用 JK 触发器实现的逻辑表达式；

（4）画出逻辑电路图。

5 - 13　图 5 - 176（a）所示电路是一个未完成的逻辑电路，图 5 - 176（b）给出该电路的 CP 和输出 Z 的波形，试分析并完成该逻辑电路的连接。图 5 - 17（b）中触发器的初态为 0。

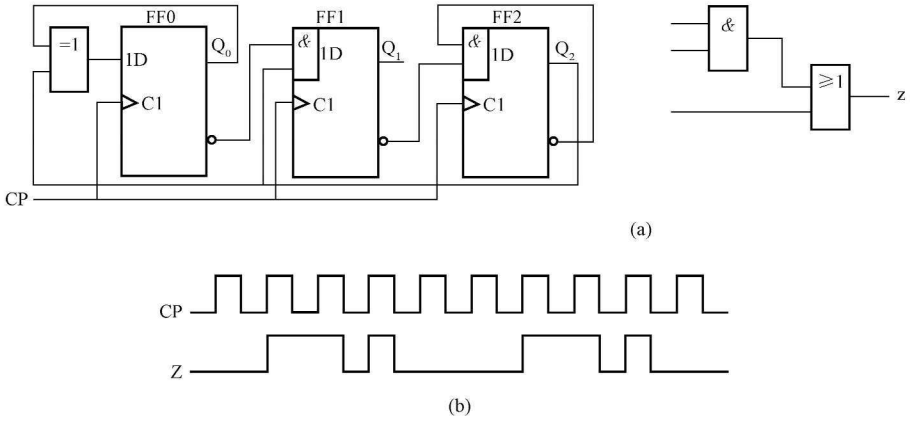


图 5 - 176　题 5 - 13 图

5 - 14　在某计数器的输出端观察到如图 5 - 177 所示的波形，试确定该计数器的模。

5 - 15　试用下降沿 JK 触发器组成 3 位二进制异步减计数器，画出逻辑电路图。

5 - 16　试用下降沿 D 触发器组成 3 位二进制异步加计数器，画出逻辑电路图。

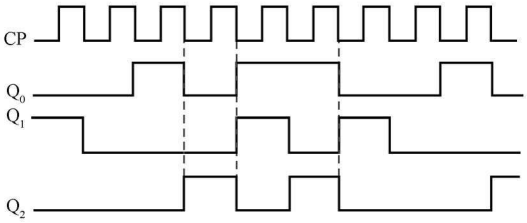


图 5 - 177 题 5 - 14 图

5 - 17 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，画出逻辑电路图。

5 - 18 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。

5 - 19 图 5 - 178 所示电路用具有异步清零功能的集成计数器 74X161 构成。试分析该电路并画出状态图，并说明是几进制计数器。

5 - 20 图 5 - 179 所示电路用具有同步清零功能的集成计数器 74X163 构成。试分析该电路并画出它的状态图，说明它是几进制计数器。

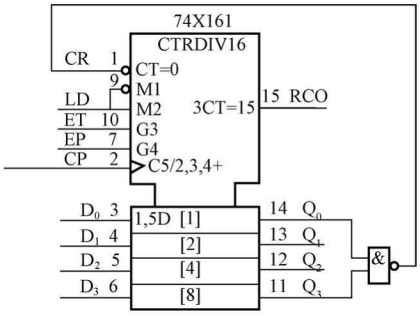


图 5 - 178 题 5 - 19 图

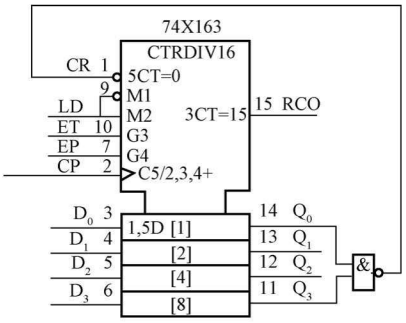


图 5 - 179 题 5 - 20 图

5 - 21 图 5 - 180 所示电路用具有异步预置功能的集成计数器 74X193 构成。试分析该电路并画出它的状态图，说明它是几进制计数器。

5 - 22 图 5 - 181 所示电路用具有同步预置功能的 8421BCD 码同步加法集成计数器 74X160 构成。试分析该电路并画出其状态图，并说明是几进制计数器。

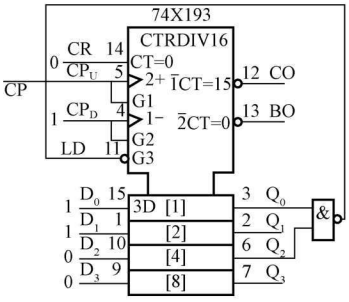


图 5 - 180 题 5 - 21 图

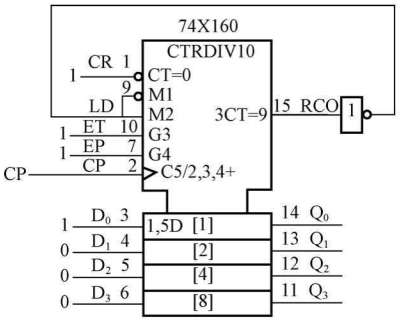


图 5 - 181 题 5 - 22 图

5 - 23 试分析图 5 - 182 所示的电路，画出它的状态图，说明它是几进制计数器。

5 - 24 试用计数器 74X290 设计一个八进制计数器，要求采用两种不同的方法。

5 - 25 试用计数器 74X161 设计一个同步六十进制计数器，要求采用两种不同的方法。

5 - 26 试分析如图 5 - 183 所示的时序逻辑电路，要求：

（1）列出驱动方程；

（2）列出状态方程；

（3）列出状态表；

（4）画出状态图；

（5）画出在 CP 脉冲作用下三个触发器的状态信号和 Z 的波形图，设三个触发器的初态均为 0。

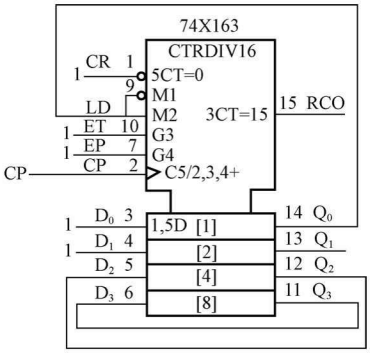


图 5 - 182　题 5 - 23 图

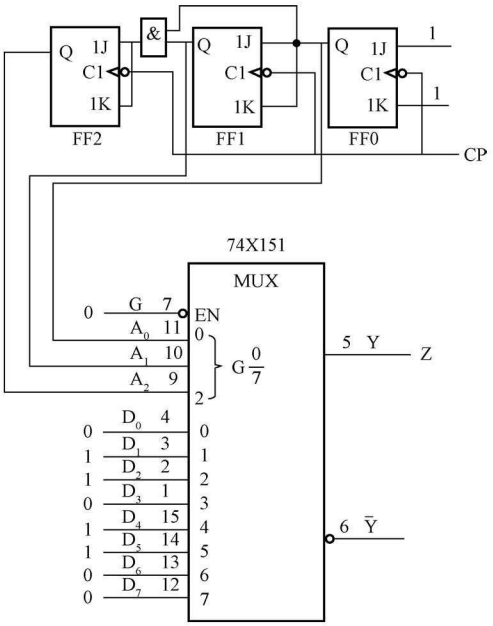


图 5 - 183　题 5 - 26 图

5 - 27　试用计数器 74X161 和数据选择器设计一个 01100011 序列发生器。

5 - 28　用同步可逆计数器 74X193 设计八进制的“加/减”计数，要求有加/减控制电路。当加/减控制信号 X 为 0 时，实现“加”计数，当加/减控制信号 X 为 1 时，实现“减”计数。要求：

（1）画出状态图；

（2）画出逻辑电路图。

5 - 29　序列码发生器电路如图 5 - 184 所示。要求：

（1）画出该电路的状态图；

（2）指出 Z 的输出序列码。

5 - 30　图 5 - 185 所示序列信号发生器由一个计数器和 4 选 1 数据选择器构成。分析计数器的工作原理，确定电路的模和状态转换关系，确定在计数器输出控制下，数据选择器的输出序列。

5 - 31　由 74X138 和 74X194 构成的电路如图 5 - 186 所示，要求：

（1）给出 74X194 构成的时序电路的状态图；

（2）设初态 $Q_0Q_1Q_2Q_3=1000$，画出 M 和 N 在 CP 作用下的波形图。

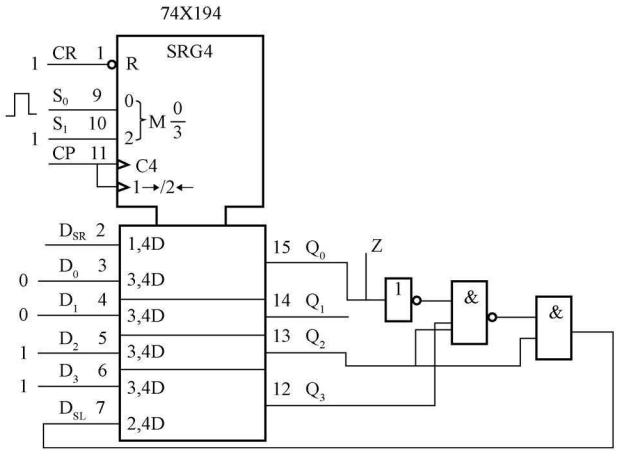


图 5 - 184　题 5 - 29 图

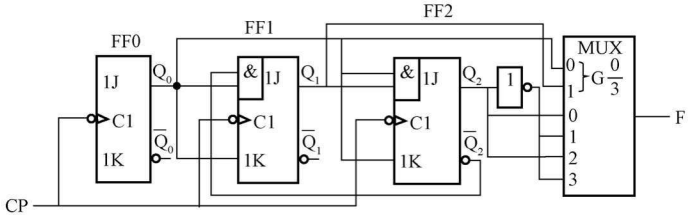


图 5 - 185　题 5 - 30 图

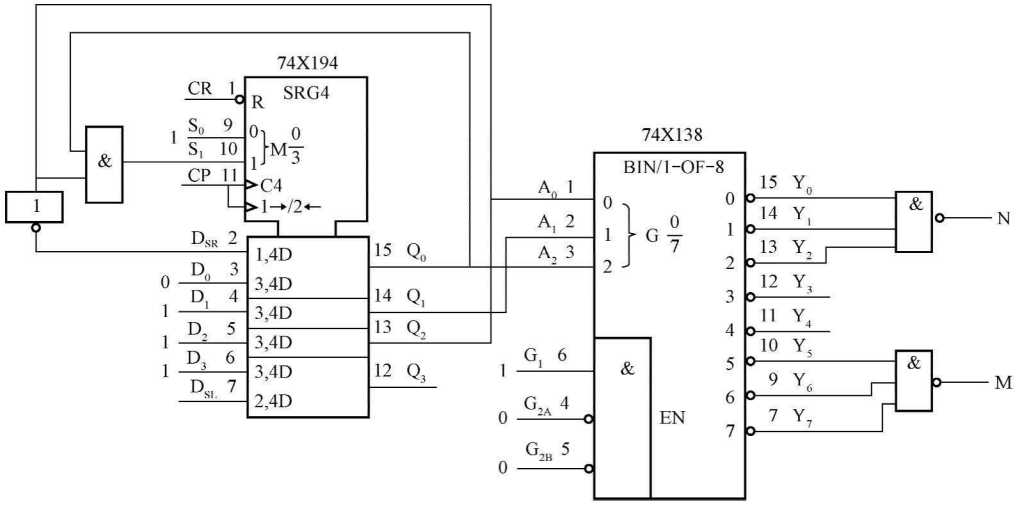


图 5 - 186　题 5 - 31 图

5 - 32　试分析如图 5 - 187 所示的时序逻辑电路，列出该时序电路的状态表，画出状态图，以及三个触发器的波形图和 74X138 八个输出的波形图，设触发器的初态为 0。说明给定逻辑电路输出的逻辑功能。

5－33　触发器的状态转换图如图 5－188 所示，写出该触发器的特性方程。如用 JK 触发器实现同样的功能，写出相应的逻辑关系表达式，并画出电路图。

5－34　触发器的状态转换图如图 5－189 所示，写出该触发器的特性方程。如用 JK 触发器实现同样的功能，写出相应的逻辑关系表达式，并画出电路图。

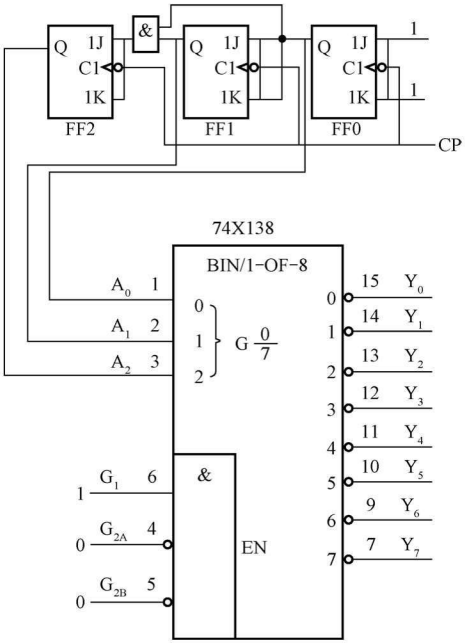


图 5－187　题 5－32 图

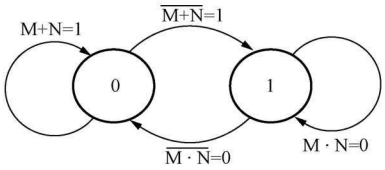


图 5－188　题 5－33 图

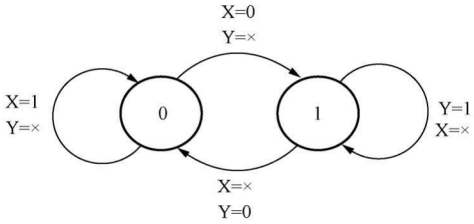


图 5－189　题 5－34 图

5－35　分析如图 5－190 所示时序逻辑电路，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。

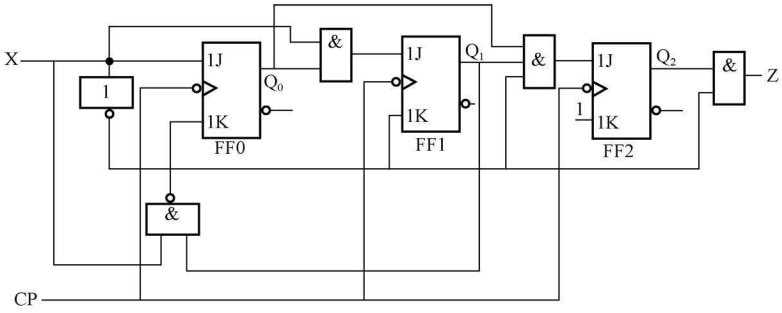


图 5－190　题 5－35 图

5－36　用边沿 JK 触发器设计一个同步三进制减法计数器，状态图如图 5－191 所示。试回答下列问题：

（1）最少用几个触发器？

（2）列出状态转换表；

（3）求出最简驱动方程；

（4）画出逻辑电路图；

（5）检查能否自启动。

5－37　用维持阻塞D触发器设计一个同步计数器，计数器的输出 Q_3、Q_2、Q_1 的状态转换关系如图5－192所示。要求：

（1）列出状态转换表；

（2）求出最简驱动方程；

（3）画出逻辑电路图；

（4）检查能否自启动，如果不能自启动，修改电路。

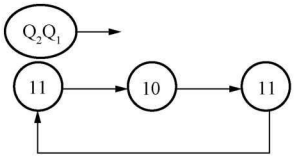


图5－191　题5－36图

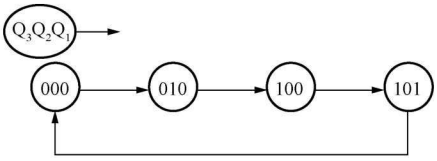


图5－192　题5－37图

5－38　用D触发器设计011序列检测器，要求分别用米里型电路和莫尔型电路来设计。

第六章　脉冲波形的产生与整形

本　章　提　要

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；另一种则是通过对已有信号进行整形、变换，使之满足系统的要求。本章主要介绍脉冲波形的产生电路和整形电路。脉冲波形产生电路主要包括由555定时器构成的多谐振荡器、对称和非对称式多谐振荡器、环形振荡器和石英晶体振荡器等。脉冲波形的整形电路主要是施密特电路和单稳态电路。

第一节　概　　述

在数字电路中，基本工作信号是二进制的数字信号，该信号只有0、1两个取值，用波形图表示就是矩形脉冲波形。矩形脉冲波形的特性主要由图6-1所示的指标来描述。各指标的含义如下：

（1）脉冲周期 T：两个相邻脉冲间的间隔。

（2）频率 f：单位时间内脉冲重复的次数。

（3）脉冲幅度 U_m：脉冲电压的最大变化幅度。

（4）脉冲宽度 t_W：从脉冲前沿上升到 $0.5U_m$ 处开始，到脉冲后沿下降到 $0.5U_m$ 为止的一段时间。

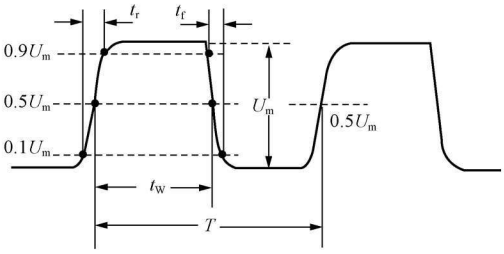


图6-1　矩形脉冲波形的主要指标

（5）上升时间 t_r：脉冲前沿从 $0.1U_m$ 上升到 $0.9U_m$ 所需要的时间。

（6）下降时间 t_f：脉冲后沿从 $0.9U_m$ 下降到 $0.1U_m$ 所需要的时间。

在后面的叙述中，为了方便起见，均采用理想矩形脉冲。理想矩形脉冲的上升时间和下降时间均为零。

第二节　555 定 时 器 芯 片

555定时器芯片是一种将模拟电路和数字电路结合在一起的中规模集成电路。它的功能灵活，适用范围广，只要在它的外部配接两三个电阻和电容元件，就可以方便地组成单稳态电路、多谐振荡器和施密特电路，所以555定时器芯片在波形的产生和整形、工业控制、定时、仿声、电子乐器及防盗报警等方面应用广泛。

555定时器芯片的产品型号繁多，有双极型（TTL型）555定时器芯片和单极型（CMOS型）555定时器芯片，其型号分别有NE555型（或5G555型）和C7555型等多种。

它们的结构、工作原理以及外部引脚排列基本相同。一般双极型（TTL 型）555 定时器芯片具有较大的驱动能力，而单极型（CMOS 型）555 定时器芯片具有低功耗、输入阻抗高等优点。双极型 555 定时器芯片的电源电压范围为 5～16V，输出最大负载电流可达 200mA；单极型 555 定时器芯片电源电压使用范围为 3～18V，但输出最大负载电流只有 4mA。

一、555 定时器芯片的内部电路

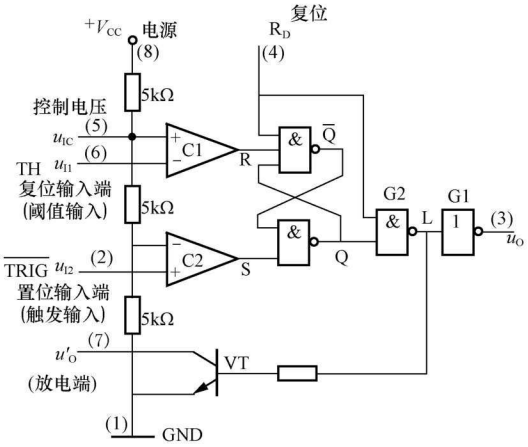


图 6-2 双极型 555 定时器芯片内部电路

图 6-2 所示为双极型 555 定时器芯片的内部电路，该电路由五部分组成。

1. 两个运算放大器

两个运算放大器 C1、C2 处于开环状态，工作于非线性区，构成电压比较器。电压比较器有两个输入端，两个运算放大器的同相输入端标有“＋”号，反相输入端标有“－”号，分别用 U_+ 和 U_- 表示相应输入端的电位，则当 $U_+>U_-$ 时其输出为高电平，$U_+<U_-$ 时输出为低电平。由模拟电路知识可知，运算放大器的两个输入端基本上不向外电路索取电流，即输入电阻趋近于无穷大。

2. 3 个 5kΩ 电阻

3 个 5kΩ 电阻组成分压电路（555 定时器芯片名称由此而来）。因为运算放大器的两个输入端的电流可以近似认为是 0，所以 3 个电阻可以认为是串联连接，那么比较器 C2 的反相输入端电位为 $V_{CC}/3$，比较器 C1 的同相输入端电位为 $2V_{CC}/3$。如果在 5 脚电压控制端 u_{IC}（也是比较器 C1 的同相输入端）外接电压源，则可改变 C1 的同相输入端电位和 C2 的反相输入端电位。例如在 u_{IC} 端外接 3V 的理想电压源，正极接 5 脚，负极接地，则比较器 C1 的同相输入端电位为 3V，比较器 C2 的反相输入端电位为 1.5V。若不使用 5 脚 u_{IC} 端，一般都通过一个 0.01μF 的电容接地，以旁路高频干扰。

3. 直接复位端

R_D 是专门设置的复位输入端，可从外部对 555 定时器芯片的 3 脚输出端 u_O 直接进行置 0（复位），低电平有效。当 $R_D=0$ 时，使 555 定时器芯片的 3 脚输出端 u_O 为低电平，处于复位状态。正常工作时，R_D 端应该接无效的高电平 1。

4. 基本 RS 锁存器

基本 RS 锁存器由两个与非门组成。

5. 三极管开关

三极管 VT 构成开关，可以处于截止状态和饱和状态。G2 门的输出信号 L 控制三极管 VT 的状态。当 L 为高电平，三极管 VT 处于饱和状态；当 L 为低电平，三极管 VT 处于截止状态。三极管 VT 又称为放电管，555 定时器芯片工作时，三极管 VT 是放电回路中的一个元件。

当 555 定时器芯片的 3 脚输出 u_O 为低电平时，G2 门的输出信号 L 为高电平，使三极管 VT 处于饱和状态，三极管 VT 的集电极（555 定时器芯片的 7 脚）为地电位。

当 555 定时器芯片的 3 脚输出 u_O 为高电平，G2 门的输出信号 L 为低电平，VT 处于截

止状态，三极管 VT 的集电极（555 定时器芯片的 7 脚）悬空。

6. 输出缓冲器

输出缓冲器就是接在输出端的反相器 G1，其作用是提高定时器的带负载能力和隔离负载对定时器的影响。

555 定时器芯片引脚图如图 6-3 所示，555 定时器芯片共有 8 个引脚。图中的数字符号为外部引脚序号，其中 1 脚为接地端 GND；2 脚为电压比较器 C2 的同相输入端$\overline{\text{TRIG}}$（也称为置位输入端，触发输入端）；3 脚为电路输出端 u_O；4 脚为直接复位端 R_D，该脚接低电平可使电路复位，即输出为低电平；5 脚为电压控制端 u_{IC}；6 脚为电压比较器 C1 的反相输入端 TH（也称为复位输入端，阈值输入端）；7 脚为放电端，当电路输出 u_O为高电平时，放电管 VT 截止，当电路输出 u_O为低电平时，放电管 VT 处于饱和状态；8 脚为正电源端 V_{CC}。

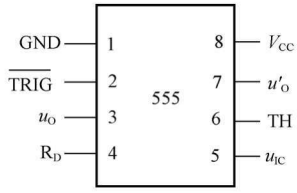


图 6-3　555 定时器芯片的引脚图

二、555 定时器的工作原理

1. 直接复位

555 定时器芯片的 3 脚输出信号 u_O为低电平的状态称 555 定时器芯片处于复位状态。当 $R_D=0$ 时，无论其他输入端如何，555 定时器芯片的 3 脚输出信号 u_O为低电平，称为直接复位，因为此时复位与置位输入端 2 脚和复位输入端 6 脚没有关系，所以称为直接复位。555 定时器芯片在复位状态时放电管 VT 处于饱和状态。

2. 置位

555 定时器芯片的 3 脚输出信号 u_O为高电平的状态称 555 定时器芯片处于置位状态。当直接复位输入端 R_D 无效（$R_D=1$）时，只要在 555 定时器芯片的 2 脚输入信号端$\overline{\text{TRIG}}$输入一个小于 $V_{CC}/3$ 的电压，电压比较器 C2 的输出 S 为低电平，S 为低电平使 Q=1，又因为 $R_D=1$，所以 555 定时器芯片的 3 脚输出信号 u_O为高电平，称为处于置位状态。L 为低电平使放电管 VT 截止。

在 5 脚悬空的情况下，将$\overline{\text{TRIG}}$端小于 $V_{CC}/3$ 的电压定义为低电平。555 定时器芯片的 2 脚输入信号$\overline{\text{TRIG}}$又称为置位输入端，为低电平有效。

3. 复位

555 定时器芯片的 3 脚输出信号 u_O为低电平的状态称 555 定时器芯片处于复位状态。当直接复位输入端 R_D 无效，同时 555 定时器芯片的 2 脚输入信号端$\overline{\text{TRIG}}$也无效的前提下，如果在 555 定时器芯片的 6 脚 TH 端输入一个大于 $2V_{CC}/3$ 的电压，则电压比较器 C1 的输出 R 为低电平，因为这时电压比较器 C2 的输出 S 为高电平，使$\overline{Q}=1$，Q=0。555 定时器芯片的 3 脚输出信号 u_O为低电平，称为复位状态，L 为高电平使放电管 VT 饱和。

在 5 脚悬空的情况下，6 脚 TH 端大于 $2V_{CC}/3$ 的电压定义为高电平。TH 端的电位大于 $2V_{CC}/3$ 的状态称为处于有效状态，TH 端的电位小于 $2V_{CC}/3$ 的状态称为处于无效状态。555 定时器芯片的 6 脚输入信号端 TH 端又称为复位输入端，为高电平有效。

4. 复位输入端和置位输入端均无效的情况

如果 555 定时器芯片的 6 脚 TH 复位输入端无效，同时 555 定时器芯片的 2 脚$\overline{\text{TRIG}}$置位输入端也无效，则电压比较器 C1、C2 输出均为高电平，基本 RS 锁存器保持原来状态不

变，因此 u_O、VT 也保持原来状态不变。

555 定时器芯片的功能如表 6-1 所示。应注意，表 6-1 的第 2 行表明，只要 555 定时器芯片的 2 脚输入信号端$\overline{TRIG}$输入一个小于 $V_{CC}/3$ 的电压，555 定时器芯片就置位，与 6 脚 TH 端是否有效无关。

表 6-1　　555 定时器芯片功能表

复位输入	置位输入	复位（R_D）	输出（u_O）	放电管 VT	复位输入	置位输入	复位（R_D）	输出（u_O）	放电管 VT
×	×	0	0	导通	$>\frac{2}{3}V_{CC}$	$>\frac{1}{3}V_{CC}$	1	0	导通
×	$<\frac{1}{3}V_{CC}$	1	1	截止	$<\frac{2}{3}V_{CC}$	$>\frac{1}{3}V_{CC}$	1	不变	不变

第三节　脉冲波形产生电路

脉冲波形产生电路不需要外加输入信号，只要接通供电电源，就自动产生矩形脉冲信号。由于其产生的矩形波信号按傅里叶级数分解，可看成是由许多不同频率的正弦波信号叠加而成，即含有丰富的谐波分量，所以称之为多谐振荡器。

一、由 555 定时器芯片组成的多谐振荡器

（一）电路结构

由 555 定时器芯片组成的多谐振荡器电路如图 6-4 所示。R_1、R_2 和 C 是外接定时元件，555 定时器芯片的 TH（6 脚）、$\overline{TRIG}$（2 脚）端连接起来接电容 C 的一端，放电管 VT 的集电极（555 定时器芯片的 7 脚）接到电阻 R_1、R_2 的连接点 P。555 定时器芯片的 5 脚经去耦电容接地，因此比较器 C2 的反相输入端电位为 $V_{CC}/3$，比较器 C1 的同相输入端电位为 $2V_{CC}/3$。

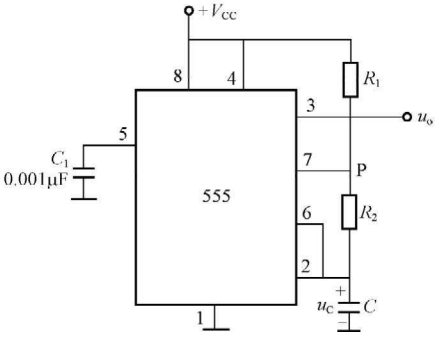


图 6-4　555 定时器芯片组成的多谐振荡器电路

（二）工作原理

1. 起始状态

接通电源前电容 C 上无电荷，所以接通电源瞬间，$u_C=0$，u_C 就是输入给 6 脚复位输入端 TH 和 2 脚置位输入端$\overline{TRIG}$的电压。由于 $u_C=0<V_{CC}/3$，所以 2 脚置位输入端$\overline{TRIG}$有效，555 定时器芯片置位，$u_O=U_{OH}$，放电管 VT 截止。

2. 暂稳态Ⅰ

当 $u_O=U_{OH}$时，放电管 VT 截止，放电管的集电极（555 定时器芯片的 7 脚）不接地，处于悬空状态。电源经过 R_1、R_2 向电容 C 充电，充电回路是 $V_{CC}\rightarrow R_1\rightarrow R_2\rightarrow C\rightarrow$地，电容 C 两端的电压 u_C 逐渐升高，充电时间常数是 $\tau_1=(R_1+R_2)C$。暂稳态Ⅰ维持时间为 t_{W1}，如图 6-5 所示。

3. 自动翻转Ⅰ

当电容 C 充电到其两端的电压 u_C 比 $2V_{CC}/3$ 稍微大一点时，555 定时器芯片的 6 脚复位

输入端 TH 有效，2 脚置位输入端$\overline{\text{TRIG}}$无效，555 定时器芯片复位，$u_O=U_{OL}$，放电管 VT 饱和。

4. 暂稳态Ⅱ

当 $u_O=U_{OL}$时，放电管 VT 饱和，放电管的集电极（555 定时器芯片的 7 脚）接地，电容 C 放电，放电回路是 $C\rightarrow R_2\rightarrow$VT→地，时间常数是 $\tau_2=R_2C$（忽略放电管 VT 饱和导通电阻 R_{CES}）。暂稳态Ⅱ维持时间为 t_{W2}，如图 6-5 所示。

5. 自动翻转Ⅱ

当电容 C 放电到其两端的电压 u_C 比 $V_{CC}/3$ 稍微小一点时，555 定时器芯片的 2 脚置位输入端$\overline{\text{TRIG}}$有效，555 定时器芯片置位，$u_O=U_{OH}$，放电管 VT 截止。

随着时间的推移，电容 C 充电和放电的过渡过程总是周而复始重复进行的，电路就在两个暂稳态之间来回翻转振荡，于是在 555 定时器芯片的 3 脚输出端就产生了矩形脉冲。该多谐振荡器的工作波形如图 6-5 所示。

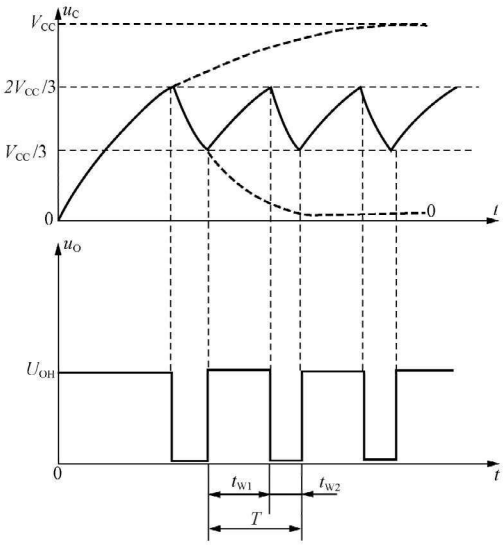


图 6-5　555 定时器芯片组成的多谐振荡器工作波形

（三）振荡频率的估算

1. 暂稳态Ⅰ维持时间 t_{W1}的估算

暂稳态Ⅰ维持时间 t_{W1}就是电容 C 从 $V_{CC}/3$ 充电到 $2V_{CC}/3$ 的时间。当电容 C 充电时，时间常数 $\tau_1=(R_1+R_2)C$，起始值为 $u_C(0^+)=V_{CC}/3$，终止值为 $u_C(\infty)=V_{CC}$，转换值 $u_C(t_{W1})=2V_{CC}/3$，代入 RC 过渡过程计算公式进行计算，可得

$$t_{W1}=\tau_1\ln\frac{u_C(\infty)-u_C(0^+)}{u_C(\infty)-u_C(t_{W1})}=\tau_1\ln\frac{V_{CC}-\frac{1}{3}V_{CC}}{V_{CC}-\frac{2}{3}V_{CC}}=\tau_1\ln2=0.7(R_1+R_2)C \quad (6-1)$$

2. 暂稳态Ⅱ维持时间 t_{W2}的估算

暂稳态Ⅱ维持时间 t_{W2}就是电容 C 从 $2V_{CC}/3$ 放电到 $V_{CC}/3$ 的时间。当电容 C 放电时，时间常数 $\tau_2=R_2C$，起始值 $u_C(0^+)=2V_{CC}/3$，终止值 $u_C(\infty)=0$，转换值 $u_C(t_{W2})=V_{CC}/3$，利用三要素法进行计算，可得

$$t_{W2}=\tau_2\ln\frac{u_C(\infty)-u_C(0^+)}{u_C(\infty)-u_C(t_{W2})}=\tau_2\ln\frac{0-\frac{2}{3}V_{CC}}{0-\frac{1}{3}V_{CC}}$$

$$=\tau_2\ln2=0.7R_2C \quad (6-2)$$

3. 电路振荡频率 f

振荡周期为

$$T=t_{W1}+t_{W2}=0.7(R_1+R_2)C+0.7R_2C=0.7(R_1+2R_2)C$$

振荡频率为

$$f=\frac{1}{T}=\frac{1}{0.7(R_1+2R_2)C}\approx\frac{1.43}{(R_1+2R_2)C}$$

4. 占空比

高电平的维持时间与重复周期之比，称为占空比，用 q 表示。计算 q 的公式为

$$q=t_{W1}/T$$

图 6-4 所示 555 定时器组成的多谐振荡器的占空比为

$$q=t_{W1}/T=(R_1+R_2)/(R_1+2R_2)$$

在图 6-4 所示电路中，由于电容 C 的充电时间常数为 $\tau_1=(R_1+R_2)C$，放电时间常数为 $\tau_2=R_2C$，所以总是 $t_{W1}>t_{W2}$，u_O的波形高电平维持的时间和低电平维持的时间不相等，所以不是方波，而且波形的占空比不能调节。

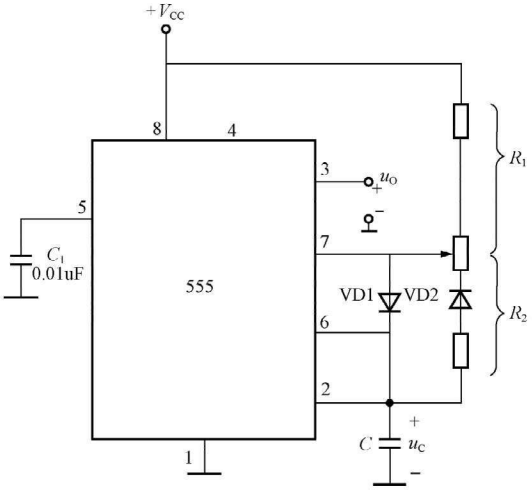


图 6-6 占空比可调的多谐振荡器电路

（四）占空比可调的多谐振荡器电路

利用半导体二极管的单向导电特性，把电容 C 充电和放电回路隔离开来，再加上一个电位器，便可以得到占空比可调的多谐振荡器。其电路如图 6-6 所示。从图中电路可以看出，电容 C 充电的路径为 $V_{CC}\rightarrow R_1$、VD1→C→地，充电时间常数为 $\tau_1=R_1C$。电容 C 放电的路径为 $C\rightarrow R_2\rightarrow$VD2→VT→地，放电时间常数为 $\tau_2=R_2C$。

忽略 VD1、VD2、VT 的导通电阻，通过计算，可得

$$t_{W1}=0.7R_1C,t_{W2}=0.7R_1C$$

则该多谐振荡器的占空比为

$$q=\frac{t_{W1}}{T}=\frac{t_{W1}}{t_{W1}+t_{W2}}=\frac{0.7R_1C}{0.7R_1C+0.7R_2C}=\frac{R_1}{R_1+R_2}$$

由上式可知，只要改变电位器活动端的位置，就可以方便地调节占空比 q。当 $R_1=R_2$ 时，$q=0.5$，u_O将成为对称的矩形脉冲（方波）。

二、环形多谐振荡器

（一）电路组成

利用门电路的延迟时间，将大于或等于 3 的奇数个非门首尾相连接成为环形，可以构成环形多谐振荡器。环形多谐振荡器是利用延迟负反馈来产生振荡的，其逻辑电路如图 6-7 所示。

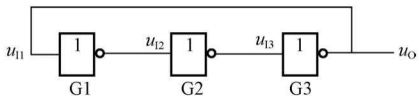


图 6-7 环形振荡器的逻辑电路

（二）工作原理

在图 6-7 电路中，假定三个非门 G1、G2、G3 的延迟时间都是 t_{pd}。任何一个非门的输出都不可能稳定在高电平或低电平状态，例如，在 0 时刻，当 u_{I1}信号突然变为高电平（$u_{I1}=1$），经过 G1 的传输延迟时间 t_{pd}之后使 $u_{I2}=0$，u_{I2}经过 G2 门的传输延迟时间 t_{pd}之后使 $u_{I3}=1$，$u_{I3}=1$ 又经过 G3 的传输延迟时间 t_{pd}之后，使输出端 u_O为 0，u_O就是 u_{I1}；$u_{I1}=0$ 经过 G1 的传输延迟时间之后使 $u_{I2}=1$，$u_{I2}=1$ 经过 G2 门的传输延迟时间后使 $u_{I3}=0$，

$u_{I3}=0$又经过 G3 的传输延迟时间之后使输出端 u_O为 1，u_O就是 u_{I1}，u_{I1}又送回 G1 的输入端。如此周而复始，该电路产生了自激振荡，振荡周期为 $6t_{pd}$，各点的工作波形如图 6-8 所示。

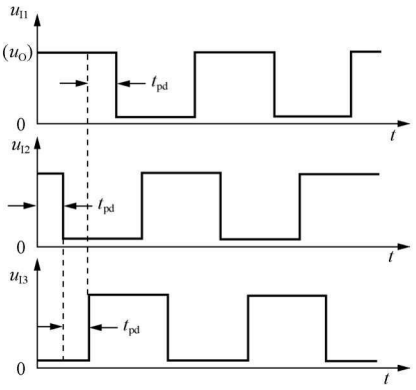


图 6-8　图 6-7 电路的工作波形图

（三）改进型的环形振荡器

由图 6-7 所示电路构成的多谐振荡器虽然简单，但很不实用。因为非门的传输延迟时间过短（仅为纳秒数量级），导致多谐振荡器的振荡频率高，而且振荡不是很稳定。另外，因为非门的传输延迟时间固定，振荡周期不可调。所以要获得较低振荡频率，且频率可调的信号是比较困难的。为此在实用电路中常加入 RC 电路。

图 6-9（a）所示为接入 RC 电路的原理性电路。RC 电路增加了 G2 门的传输延迟时间，有助于获得较低振荡频率的信号。为了进一步加大延迟时间，可将电容 C 的接地端改接到 G1 的输出端上，如图 6-9（b）所示。因为 G1 的输出端到地之间还有一个等效的电阻，相当于增加了充放电回路中的电阻。该电路没有稳定状态，只有两个暂稳态。电路由一个暂稳态自动翻转到另一个暂稳态，是由电容 C 的充放电实现的，可以通过调节 R 或 C 的值来调节振荡频率。由于 RC 电路的延迟时间远远大于门电路的延迟时间，分析时可以忽略门电路的延迟时间，近似认为每个门的输入、输出的跳变同时发生。R_S 为限流电阻，阻值较小，所以 u_{I3}可以视为是门 G3 的输入电压。图 6-9（c）是第一暂稳态期间 u_{O2}通过电阻 R 给电容 C 充电的电路示意图，图 6-9（d）第二暂稳态期间 u_{O1}通过电阻 R 给电容 C 反向充电（放电）的电路示意图。

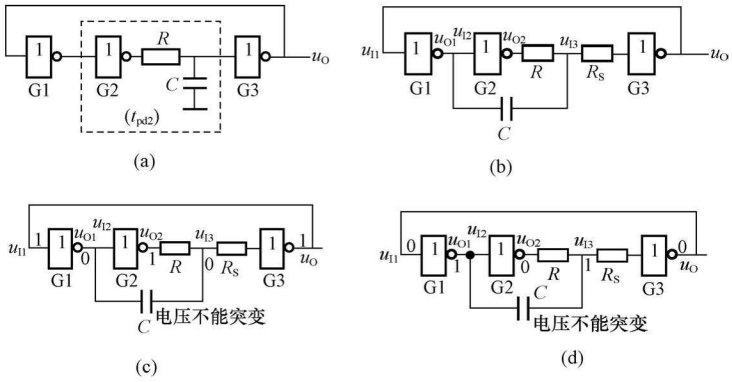


图 6-9　带 RC 延迟电路的环形振荡器

（a）原理性电路；（b）实用的改进电路；（c）第一暂稳态期间 u_{O2}通过电阻 R 给电容 C 充电；（d）第二暂稳态期间 u_{O1}通过电阻 R 给电容 C 反向充电（放电）

图 6-10 给出了图 6-9（b）所示电路的工作波形。分析这一部分内容时，要联系门电路内部的电路结构来理解。此处非门采用标准生产工艺的 TTL 门电路 7404。7404 的输入端有一个保护二极管。本章中要考虑这个二极管的影响。

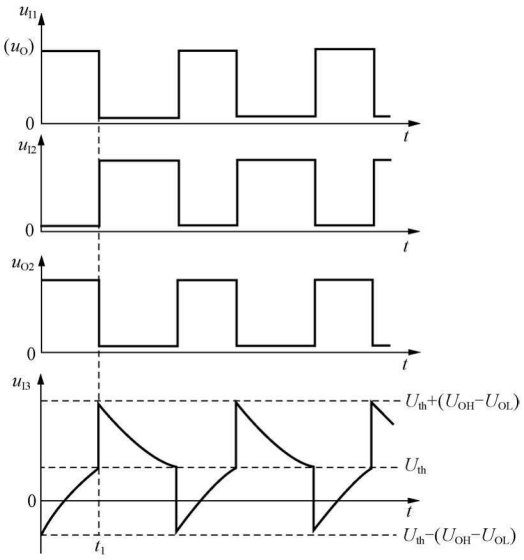


图 6-10 图 6-9（b）电路的工作波形

1. 第一暂稳态及其自动翻转过程

在 0 时刻，设门 G3 的输出 u_O（即门 G1 的输入 u_{I1}）由低电平跳变为高电平，则门 G1 的输出 u_{O1}（即门 G2 的输入 u_{I2}）由高电平跳变为低电平，u_{O2} 由低电平跳变为高电平，同时 u_{I2} 经过电容 C 使 u_{I3} 首先由高电平跳变为低电平（因为电容 C 两端电压不能突变），维持 u_O 为高电平。此时为第一暂稳态。如图 6-9（c）所示，在此期间 u_{O2} 通过电阻 R 给电容 C 充电，使 u_{I3} 的电位逐渐上升。充电回路为 $u_{O2}\rightarrow R\rightarrow u_{I3}\rightarrow C\rightarrow u_{O1}$，这是电流实际流动的方向。当 u_{I3} 上升到门的阈值电压 U_{th} 时，门 G3 导通，使门 G3 的输出 u_O 由高电平跳变为低电平，至此第一暂稳态结束。电路自动翻转到第二暂稳态。

2. 第二暂稳态及其自动翻转过程

在 t_1 时刻，设门 G3 的输出 u_O（即门 G1 的输入 u_{I1}）由高电平跳变为低电平，则门 G1 的输出 u_{O1}（即门 G2 的输入 u_{I2}）由低电平跳变为高电平。同时 u_{I2} 经过电容 C 使 u_{I3} 首先由低电平跳变为高电平（因为电容 C 两端电压不能突变），维持 u_O 为低电平。此时为第二暂稳态。如图 6-9（d）所示，在此期间 u_{O1} 通过电阻 R 给电容 C 反向充电（放电），电容 C 上的压降增加，使 u_{I3} 的电位逐渐下降。反向充电回路为 $u_{O1}\rightarrow C\rightarrow u_{I3}\rightarrow R\rightarrow u_{O2}$，这是电流实际流动的方向。当 u_{I3} 下降到小于门的阈值电压 U_{th} 时，门 G3 截止，使门 G3 的输出 u_O 由低电平跳变为高电平，至此第二暂稳态结束。电路自动翻转到第一暂稳态。

依靠电容 C 的充电、放电，使两个暂稳态周而复始地转换，形成周期性振荡，在门 G3 的输出端 u_O 得到的就是矩形脉冲波形。

在图 6-11（a）中，因为 u_{I3} 为低电平，所以门 G3 的发射结导通，二极管处于截止状态。

由图 6-11（c）所示的等效电路（发射结的等效电阻忽略），根据戴维南定理，可求得第一暂稳态期间电容 C 充电时求等效电阻 R_E 和等效电压源 U_E 分别为

$$U_E = U_{OH} + (V_{CC} - U_{BE} - U_{OH})\frac{R}{R + R_1 + R_S} \tag{6-3}$$

$$R_E = \frac{R(R_1 + R_S)}{R + R_1 + R_S} \tag{6-4}$$

电容 C 从电压值 $U_{th}-(U_{OH}-U_{OL})$ 充电到电压值 U_{th}，可求得电容 C 的充电时间为

$$t_{W1} = R_E C \ln\frac{U_E - [U_{th} - (U_{OH} - U_{OL})]}{U_E - U_{th}} \tag{6-5}$$

在图 6-11（b）中，电容 C 从电压值 $U_{th}+(U_{OH}-U_{OL})$ 放电到电压值 U_{th}，可求得电容 C 的放电时间为

$$t_{W2} = RC\ln\frac{U_{OH} + (U_{th} - U_{OL}) - U_{OL}}{U_{TH} - U_{OL}}$$

$$= RC\ln\frac{U_{OH}+U_{th}-2U_{OL}}{U_{th}-U_{OL}} \tag{6-6}$$

若 $R_1+R_S \geqslant R$，$U_{OL}\approx 0$，则 $U_E \approx U_{OH}$，$R_E \approx R$，则式（6-5）和式（6-6）可分别简化为

$$t_{W1} \approx RC\ln\frac{2U_{OH}-U_{th}}{U_{OH}-U_{th}} \tag{6-7}$$

$$t_{W2} \approx RC\ln\frac{U_{OH}+U_{th}}{U_{th}} \tag{6-8}$$

从而可得图 6-9（b）电路的振荡周期为

$$\begin{aligned} T &= t_{W1}+t_{W2} \\ &\approx RC\ln\left(\frac{2U_{OH}-U_{th}}{U_{OH}-U_{th}}\cdot\frac{U_{OH}+U_{th}}{U_{th}}\right) \end{aligned} \tag{6-9}$$

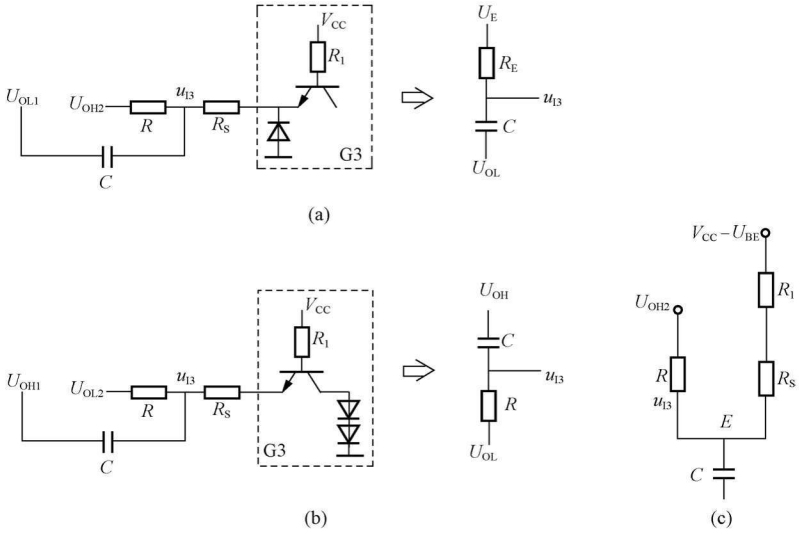


图 6-11　图 6-9（b）电路中电容的充放电等效电路

（a）第一暂稳态期间电容 C 充电时的等效电路；（b）第二暂稳态期间电容 C 放电时的等效电路；（c）第一暂稳态期间电容 C 充电时求等效电阻 R_E 和等效电压源 U_E 的等效电路

三、对称式多谐振荡器

（一）电路组成

由 TTL 反相器组成的多谐振荡器的逻辑电路如图 6-12 所示。图中，电容 C_1、C_2 把两个反相器的输出端和输入端耦合起来，通过电阻 R_{F1} 和 R_{F2} 把每个门的输出信号反馈到输入端。电容 C_1、C_2 有两个作用：一是将一个门的输出信号的跳变量耦合到另一个门的输入端；二是作为充放电元件，以获得一定宽度的输出脉冲。显然，这是由两个反相器利用两个电容相互耦合而成的正反馈环路。为了保证电路能够可靠起振，必须使 G1、G2 静态时工作在电压传输特性的转折区，因为这时反相器相当于一个放大能力很强的放大器，对

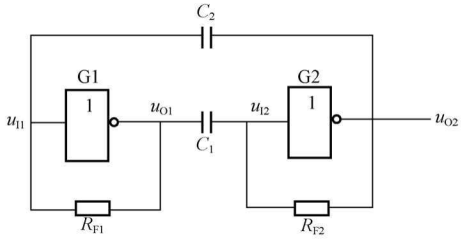


图 6-12　对称式多谐振荡器的逻辑电路

输入信号 u_I 有很强的放大作用，才能满足振荡条件。R_{F1} 和 R_{F2} 就是为了满足这个要求而加的输入偏置电路。对于 74 系列的门电路而言，R_{F1} 一般在0.5～1.9kΩ 之间取值。此处反相器采用标准生产工艺的 TTL 门电路 7404。

（二）工作原理

对称式多谐振荡器电路有两个暂稳态。

第一暂稳态：G1 截止，$u_{O1}=U_{OH}$；G2 导通，$u_{O2}=U_{OL}$。其间 C_2 充电，充电路径为 $u_{O1}\rightarrow R_{F1}\rightarrow u_{I1}\rightarrow C_2\rightarrow u_{O2}$，$C_1$ 放电，放电路径为 $u_{O1}\rightarrow C_1\rightarrow u_{I2}\rightarrow R_{F2}\rightarrow u_{O2}$。

第二暂稳态：G1 导通，$u_{O1}=U_{OL}$；G2 截止，$u_{O2}=U_{OH}$。其间 C_1 充电，充电路径为 $u_{O2}\rightarrow R_{F2}\rightarrow u_{I2}\rightarrow C_1\rightarrow u_{O1}$。$C_2$ 放电，放电路径为 $u_{O2}\rightarrow C_2\rightarrow u_{I1}\rightarrow R_{F1}\rightarrow u_{O1}$。

随着 C_1 和 C_2 充、放电的进行，该电路在两个暂稳态之间周而复始地相互转换，便形成了多谐振荡。

图 6-13 所示为第二暂稳态下 C_1 充电和 C_2 放电的等效电路。图 6-13（a）中的 R_{E1} 和 U_{E1} 是根据戴维南定理求得的等效电阻和等效电压源，它们分别为

$$R_{E1}=\frac{R_1R_{F2}}{R_1+R_{F2}} \tag{6-10}$$

$$U_{E1}=U_{OH}+\frac{R_{F2}}{R_1+R_{F2}}(V_{CC}-U_{BE}-U_{OH}) \tag{6-11}$$

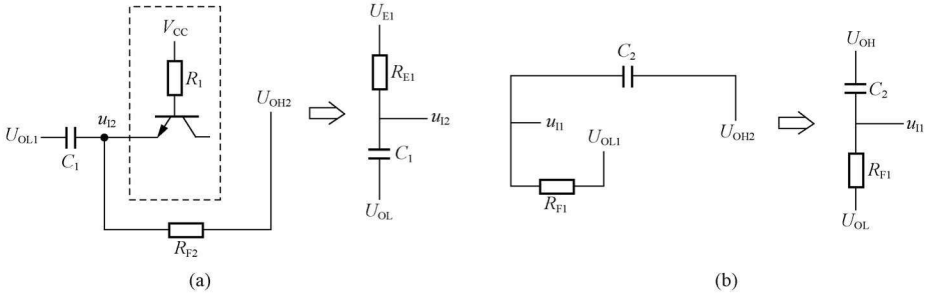


图 6-13　图 6-12 电路中第二暂稳态下电容的充、放电等效电路

（a）C_1 充电的等效电路；（b）C_2 放电的等效电路

第二暂稳态的时间 t_{W2} 是 C_1 从 -0.7V 充电到门的阈值电压 U_{th} 所经历的时间。如果认为 $U_{OL}=0$，那么 C_1 两端的电压就是 u_{I2}。电路中各点的电压波形如图 6-14 所示。

时间常数 $\tau_2=R_{E1}C_1$，起始值 $u_C(0^+)=-0.7$，终了值 $u_C(\infty)=U_{E1}$、转换值 $u_C(t_{W2})=U_{th}$，代入 RC 过渡过程公式进行计算，有

$$t_{W2}=\tau_2\ln\frac{u_C(\infty)-u_C(0^+)}{u_C(\infty)-u_C(t_{W2})}=\tau_2\ln\frac{U_{E1}+0.7}{U_{E1}-U_{th}}$$

因为电路的对称性，总的周期应该等于 t_{W2} 的两倍，即

$$T=t_{W1}+t_{W2}=2t_{W2}$$

四、非对称式多谐振荡器

（一）电路结构

对于图 6-12 所示的电路还可以进一步简化。由于门电路 G1 在静态时工作在电压传输特性转折区，所以把 G1 的输出直接接在 G2 的输入端，G2 就可得到一个介于高、低电平之

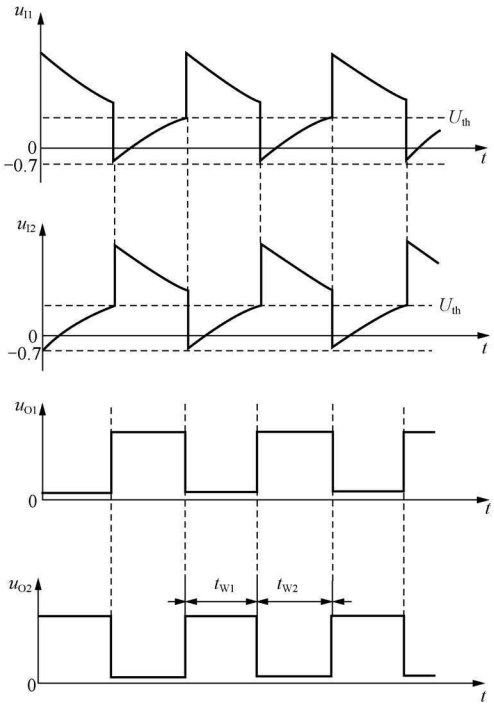


图 6-14　图 6-12 电路中各点电压波形

间的静态偏置电压，并可以工作在电压传输特性的转折区上。把 C_1 和 R_{F2} 去掉，在反馈环路中保留电容 C_2，就可以使电路没有稳定状态，而只能在两个暂稳态之间往复振荡。由于该电路结构不对称，因此称其为非对称式多谐振荡器，它的电路如图 6-15 所示，其中的 G1、G2 为 CMOS 非门。

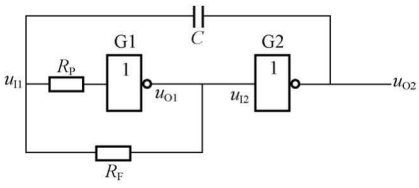


图 6-15　非对称式多谐振荡器电路

（二）工作原理

非对称式多谐振荡器电路没有稳定状态，设在 t＝0 时刻，$u_{O1}=U_{OH}$，此时 G2 导通，$u_{O2}=U_{OL}$，u_{O1} 的高电平经 R_F、C 和 G2 的输出电阻对 C 进行充电，u_{I1} 随之升高，电路有如下的正反馈过程

$$u_{I1}\uparrow \rightarrow u_{O1}\downarrow \rightarrow u_{O2}\uparrow$$

使 u_{O1} 跳变为低电平，u_{O2} 跳变为高电平，电路进入第一个暂稳态。u_{O2} 的高电平经 C、R_F 和 G1 的输出电阻对电容 C 进行反向充电，u_{I1} 随之下降，电路有如下的正反馈过程

$$u_{I1}\downarrow \rightarrow u_{O1}\uparrow \rightarrow u_{O2}\downarrow$$

使 u_{O1} 由低电平跳变为高电平，u_{O2} 由高电平跳变为低电平，电路进入第二个暂稳态。电路在第二个暂稳态并不能稳定，u_{O1} 的高电平又经 R_F、C 和 G2 的输出电阻对 C 进行充电，u_{I1} 随之升高，u_{O1} 跳变为低电平，u_{O2} 跳变为高电平，又返回第一暂稳态。两个暂稳态相互交替，使得电路输出周期性的矩形脉冲。图 6-16 为该电路的工作波形。

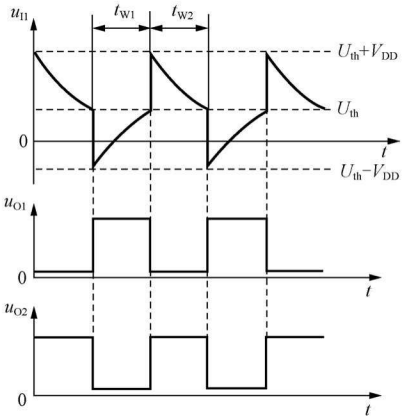


图 6-16 非对称式多谐振荡器的工作波形

五、石英晶体多谐振荡器

在实际的数字系统中，为了得到频率高度稳定的脉冲波形，普遍采用石英晶体多谐振荡器。图 6-17 给出了石英晶体的符号和电抗频率特性。在电路上，只要把石英晶体与上述基本多谐振荡器中的一个耦合电容 C_1 或 C_2 串接起来就可组成石英晶体多谐振荡器。这种石英晶体多谐振荡器的振荡频率仅取决于石英晶体的固有谐振频率，而与外接电阻、电容的参数无关。在非对称式多谐振荡器电路中，同样可将石英晶体与耦合电容串联，以达到稳频的目的。

图 6-18 所示为一典型的石英晶体振荡器电路。该电路的特点是在对称式多谐振荡器的电容回路中串接了石英晶体，两个 R_F 电阻用于确定反相器 G1 和 G2 的工作点，使其工作在放大区。

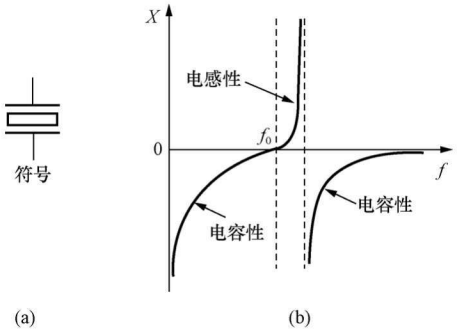


图 6-17 石英晶体的电抗频率特性和符号
(a) 符号；(b) 电抗频率特性

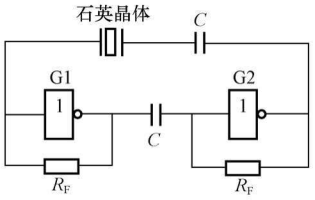


图 6-18 石英晶体多谐振荡器电路

第四节 脉冲波形整形电路

脉冲波形整形电路是对已有的波形进行整形和变换，使之符合系统的要求。脉冲波形整形电路主要有施密特电路和单稳态电路，这是两种用途不同的脉冲整形电路。施密特电路主要用来将变化缓慢的或快速变化的非矩形脉冲变换成上升沿和下降沿都很陡峭的矩形脉冲，而单稳态电路则主要用于将宽度不符合要求的脉冲变换成宽度符合要求的矩形脉冲。

一、施密特电路

施密特电路是一种常用的脉冲波形整形电路，可以将正弦波或其他不规则波形变换成矩形波。其主要特点是：

(1) 有两个稳态，所以广义上说也是一种双稳态电路。

(2) 属电平触发型，即依靠输入信号的电压幅度来触发和维持电路状态。输入信号 u_I 超过某值时，电路处于一种稳态；输入信号 u_I 低于某值时，电路处于另一种稳态。

(3) 两个稳态的相互转换电平不等，即电路从一种稳态转变为另一种稳态时的 u_I 转换电平（U_{T+}）不等于从另一种稳态返回到原来稳态时的 u_I 转换电平（U_{T-}），这种特性通常称之为施密特电路的滞回特性或回差特性。

施密特电路可由 TTL/CMOS 门组成，也可由分立元件组成，但更多的是直接选用集成施密特电路芯片，如 CC40106。施密特电路的用途很广，除可用于波形变换、脉冲整形和脉冲鉴幅外，还可用来组成单稳态电路和多谐振荡器等。

（一）555 定时器芯片组成的施密特电路

1. 电路结构

图 6-19 所示为 555 定时器芯片组成的施密特电路电路图。图中将 555 定时器芯片的置位输入端$\overline{\text{TRIG}}$（2 脚）和复位输入端 TH（6 脚）端连接在一起，作为信号输入端 u_I，555 定时器芯片的 3 脚作为输出信号端 u_{O1}。555 定时器芯片中的三极管 VT 的集电极引出端（7 脚）经过上拉电阻 R 接电源 V_{DD}，称为输出端 u_{O2}，其输出电压可通过改变 V_{DD}的大小进行调节。因为 555 定时器芯片的 5 脚悬空，所以这时 555 定时器芯片的高电平为 $2V_{CC}/3$，低电平为 $V_{CC}/3$。如果 555 定时器芯片的 5 脚不悬空，这时就要重新计算高电平和低电平的值。

2. 工作原理

如图 6-19 所示，当输入信号 u_I 小于 $V_{CC}/3$，置位输入端$\overline{\text{TRIG}}$（2 脚）有效，所以信号输出端 u_{O1}就置位；当输入信号 u_I 大于 $2V_{CC}/3$，复位输入端 TH（6 脚）有效，同时置位输入端$\overline{\text{TRIG}}$（2 脚）无效，所以信号输出端 u_{O1}就复位。

当输入信号 u_I 的波形为如图 6-20 所示的三角波时，其输出波形如图 6-20 所示。下面分以下几种情况来分析该施密特电路的工作原理。

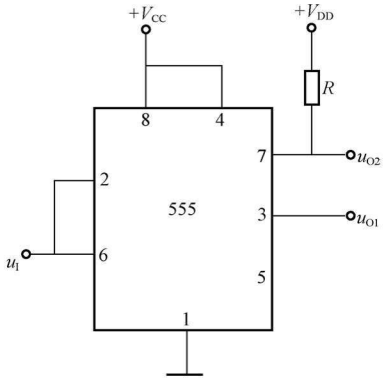


图 6-19　555 定时器芯片组成的施密特电路

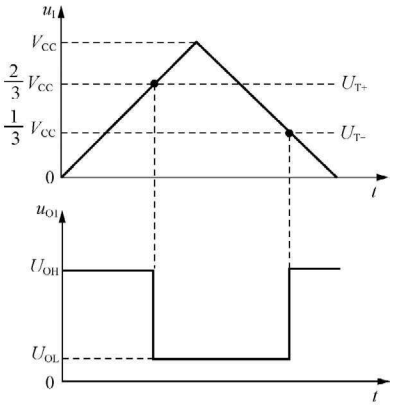


图 6-20　施密特电路在输入为三角波时的输出信号 u_{O1}的波形

（1）当 $u_I=0$V 时，555 定时器芯片的置位输入端$\overline{\text{TRIG}}$（2 脚）和复位输入端 TH（6 脚）端为 0V，这时 555 定时器芯片的置位输入端$\overline{\text{TRIG}}$（2 脚）电压小于 $V_{CC}/3$，为其有效状态，所以信号输出端 u_{O1}置位。这时，比较器 C1 输出为 1、比较器 C2 输出为 0，基本 RS 锁存器的 Q=1、$\overline{Q}=0$，$\overline{Q}=0$ 使三极管 VT 截止，VT 的集电极引出端（7 脚）u_{O2}经过上拉电阻 R 接电源 V_{DD}，只要 u_I 小于 $2V_{CC}/3$，电路的这种状态就不会改变。

（2）当 u_I 大于 $2V_{CC}/3$ 时，555 定时器芯片的置位输入端$\overline{\text{TRIG}}$（2 脚）和复位输入端 TH（6 脚）端都大于 $2V_{CC}/3$，这时 555 定时器芯片的复位输入端 TH（6 脚）为其有效状

态，同时置位输入端$\overline{TRIG}$（2 脚）无效，所以信号输出端 u_{O1}复位。这时，比较器 C1 输出为 0，比较器 C2 输出为 1，基本 RS 锁存器的 Q=0、$\overline{Q}$=1，$\overline{Q}$=1 使三极管 VT 饱和，VT 的集电极引出端（7 脚）u_{O2}接地，只要 u_I 不小于 $V_{CC}/3$，电路的这种状态就不会改变。

（3）当 u_I 下降到小于 $V_{CC}/3$ 时，555 定时器芯片的置位输入端$\overline{TRIG}$（2 脚）和复位输入端 TH（6 脚）端小于 $V_{CC}/3$，这时电路状态与上述（1）中情况相同。

如图 6-20 所示，施密特电路可将输入的缓慢变化的三角波 u_I 整形成为矩形脉冲。

3. 阈值电压

当输出发生跳变时，所对应的输入电压的值，定义为阈值电压。

（1）上限阈值电压。在输入信号电压从小向大变化的过程中，当输出发生跳变时所对应的输入电压的值，称为上限阈值电压，用 U_{T+} 表示。图 6-19 所示的施密特电路的上限阈值电压为 $U_{T+}=\frac{2}{3}V_{CC}$。

（2）下限阈值电压。在输入信号电压从大向小变化的过程中，当输出发生跳变时所对应的输入电压的值，称为下限阈值电压，用 U_{T-} 表示。图 6-19 所示的施密特电路的下限阈值电压为 $U_{T-}=\frac{1}{3}V_{CC}$。

4. 回差电压

电路的上限阈值电压与电路的下限阈值电压的差，称为回差电压，用 ΔU_T 表示。其值为

$$\Delta U_T = U_{T+} - U_{T-} = \frac{1}{3}V_{CC} \tag{6-12}$$

5. 电压传输特性曲线

图 6-21 所示为图 6-19 所示的施密特电路的电压传输特性曲线，为反相电压传输特性。反相电压传输特性中的反相是指当输入为高电平时，输出为低电平；当输入为低电平时，输出为高电平。

（二）由门电路组成的施密特电路

1. 电路结构

由门电路组成的施密特电路的电路图如图 6-22（a）所示，该电路将两级反相器 G1 和 G2 串接起来，同时通过分压电阻把输出端的电压反馈到输入端，来控制反相器的工作状态。图 6-22（b）为施密特电路的逻辑符号。输出端没有圆圈的为同相输出施密特电路的逻辑符号，输出端有圆圈的为反相输出施密特电路的逻辑符号。

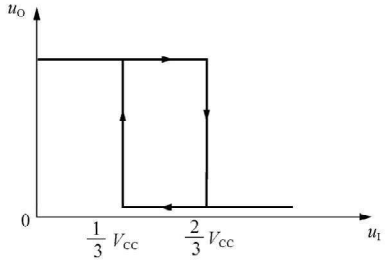


图 6-21 图 6-19 电路的电压传输特性曲线

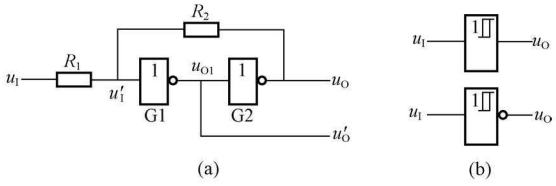


图 6-22 由门电路组成的施密特电路图

（a）逻辑电路图；（b）逻辑符号

2. 工作原理

假定 G1 和 G2 两个反相器是 CMOS 反相器，其电源电压为 V_{DD}，且 $R_1 < R_2$。CMOS 反相器的阈值电压 $U_{th} = \frac{1}{2}V_{DD}$。

（1）初始状态。当输入信号 $u_I = 0V$ 时，因 G1、G2 组成正反馈电路，所以输出 $u_{O1} = U_{OH} \approx V_{DD}$，$u_O = U_{OL} \approx 0V$。此时有

$$u_I' = \frac{R_2}{R_1 + R_2} u_I \tag{6-13}$$

（2）第一次翻转。当输入信号 u_I 从小到大增加时，并达到使 $u_I' = U_{th}$ 时，G1 进入放大区，引发了如下的正反馈过程

$$u_I' \uparrow \rightarrow u_{O1} \downarrow \rightarrow u_O \uparrow$$

正反馈使得电路的状态迅速发生转换，即 G1 开通，G2 关闭，输出 $u_O = U_{OH} \approx V_{DD}$。上限阈值电压 U_{T+} 可由下列公式得出

$$u_I' = U_{th} = \frac{R_2}{R_1 + R_2} U_{T+}$$

$$U_{T+} = \frac{R_1 + R_2}{R_2} U_{th} = \left(1 + \frac{R_1}{R_2}\right) U_{th} \tag{6-14}$$

此后，输入电压 u_I 继续增大时，电路状态保持不变。

（3）第二次翻转。当输入信号 u_I 从高向低下降时，由于此时 $u_O = U_{OH} \approx V_{DD}$，则

$$u_I' = \frac{V_{DD} - u_I}{R_1 + R_2} R_1 + u_I \tag{6-15}$$

当输入电压 u_I 下降到使 $u_I' = U_{th}$ 时，电路又有如下正反馈过程

$$u_I' \downarrow \rightarrow u_{O1} \uparrow \rightarrow u_O \downarrow$$

该反馈过程使电路又发生了第二次翻转，G1 关闭，G2 导通，输出 $u_O = U_{OL} \approx 0V$。下限阈值电压 U_{T-} 可由下列公式得出

$$u_I' = U_{th} = \frac{V_{DD} - U_{T-}}{R_1 + R_2} R_1 + U_{T-} \tag{6-16}$$

将 $V_{DD} = 2U_{th}$ 代入式（6-16）可得

$$U_{T-} = \left(1 - \frac{R_1}{R_2}\right) U_{th} \tag{6-17}$$

回差电压为

$$\Delta U_T = U_{T+} - U_{T-} \tag{6-18}$$

3. 电压传输特性

由门电路组成的施密特电路的电压传输特性如图 6-23 所示。其中，以 u_O 作为输出端时，由于输出与输入信号的高低电平是同相的，所以这种施密特电路称为同相输出的施密特电路，如图 6-23（a）所示。如果以 u'_O 作为输出端，则其电压传输特性为图 6-23（b）所示，该特性为反相电压传输特性。

U_{T+}、U_{T-} 和回差电压的大小可以通过改变电阻 R_1 和 R_2 的比值来实现，但是要求 R_1 必须小于 R_2。

（三）施密特电路的应用

1. 波形变换

施密特电路可以把边沿变化缓慢的周期性信号变换为边沿很陡峭的矩形脉冲信号，波形变换示意图如图 6 - 24 所示。

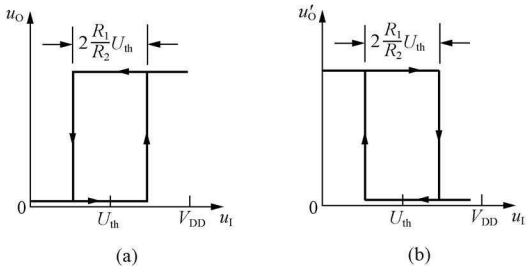


图 6 - 23　电压传输特性

（a）u_O作为输出的电压传输特性曲线；（b）u'_O作为输出的电压传输特性曲线

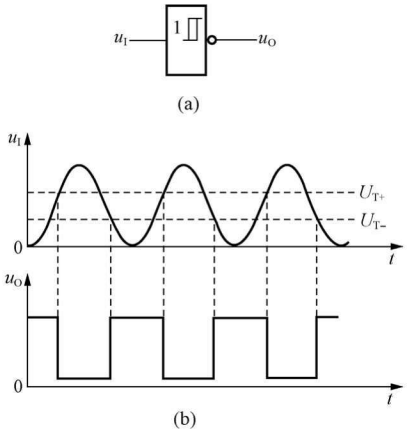


图 6 - 24　施密特电路的波形变换示意图

（a）电路图；（b）波形图

2. 脉冲整形

脉冲信号受到干扰后，往往发生畸变，将受到干扰的信号作为施密特电路的输入信号，可得到比较理想的矩形脉冲。对于图 6 - 24（a）所示的电路，其输入输出波形如图6 - 25所示。

3. 脉冲幅值鉴别

当输入信号为一组幅值不等的脉冲并要求去掉幅值较小的脉冲时，可将该信号作为施密特电路的输入信号，只有那些幅值大于施密特电路的上限阈值电压 U_{T+} 的脉冲才能在输出端产生信号，从而去掉了幅值较小的脉冲。其脉冲幅值鉴别示意图如图 6 - 26 所示。

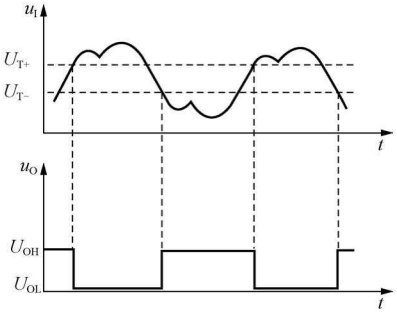


图 6 - 25　脉冲整形

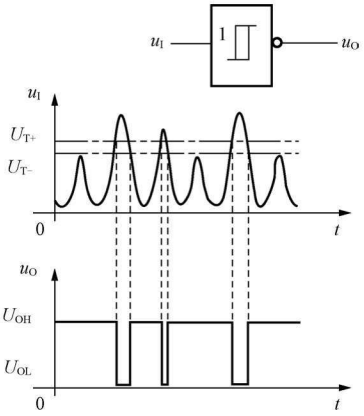


图 6 - 26　脉冲幅值鉴别示意图

4. 利用施密特电路构成多谐振荡器

如图 6 - 27 所示，将施密特电路的反相输出端经 RC 积分电路反馈回输入端，当直流电

源接通时，设电容初始电压为零，所以输出为高电平，输出信号经电阻 R 向电容 C 充电，当电容 C 上的电压升高到上限阈值电压时，输出跳变为低电平。而电容 C 上的电压又经过电阻 R 开始放电，电容电压下降，当其下降到下限阈值电压时，输出信号又跳变为高电平，电容 C 重新开始充电。如此周而复始，电路就实现了振荡。该电路输出波形如图 6-28 所示。

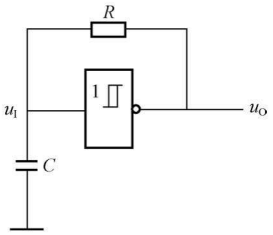


图 6-27　用施密特电路构成的多谐振荡器

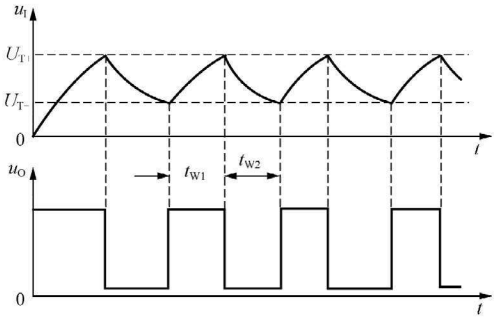


图 6-28　图 6-27 电路的电压波形图

二、单稳态电路

单稳态电路是常用的脉冲整形和延时电路，其功能特点为：

（1）电路有一个稳态、一个暂态。

（2）在外来触发信号作用下，电路由稳态翻转到暂态。

（3）暂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。暂态的持续时间取决于 RC 电路的参数，与触发信号无关。

由于单稳态电路有上述几个特点，所以被广泛应用于脉冲的整形、定时和延时。根据 RC 电路的不同接法，单稳态电路可分为微分型和积分型两种；根据电路及工作状态的不同，单稳态电路又可分为非可重复触发型（如 CT54/74121/221、CC74HC121 等）和可重复触发型（如 MC14528/14538、CT54/74123/122 等）两种；根据触发信号的有效沿和暂态电平的不同，单稳态电路可分为四种，即触发信号下降沿有效、暂态为高电平；触发信号下降沿有效、暂态为低电平；触发信号上升沿有效、暂态为高电平；触发信号上升沿有效、暂态为低电平。

（一）由 555 定时器芯片组成的单稳态电路

用 555 定时器芯片组成的单稳态电路如图 6-29 所示。电阻 R、电容 C 是定时元件，u_I 是输入触发信号，下降沿有效，加在 555 定时器芯片的置位输入端 $\overline{TRIG}$（2 脚），555 定时器芯片的 3 脚 u_O 是输出信号。

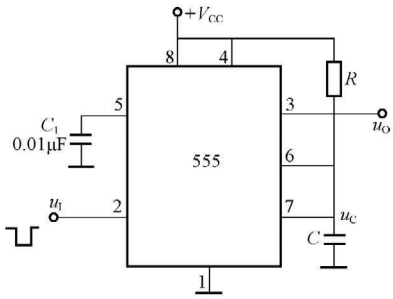


图 6-29　由 555 定时器芯片组成的单稳态电路图

1. 工作原理

（1）上电初始状态。在电源接通的瞬间，假定电容 C 没有存储电荷，即电容 C 两端电压为 0V，复位输入端 TH（6 脚）端为 0V，不大于 $2V_{CC}/3$，为无效电平。u_I 输入触发信号不小于 $V_{CC}/3$，为无效电平。这时，555 定时器芯片的置位输入端 $\overline{TRIG}$（2 脚）和复位输入端 TH（6

脚）端都无效。因为电容 C 两端电压为 0V，所以电源通过电阻 R 向电容 C 充电，当电容 C 两端电压大于 $2V_{CC}/3$ 时，复位输入端 TH（6 脚）有效，所以信号输出端 u_O 复位，三极管 VT 饱和，VT 的集电极引出端（7 脚）接地，电容 C 放电，电容 C 两端电压为 0V。此后，只要 u_I 输入触发信号不小于 $V_{CC}/3$，电路就一直维持这个状态不变。

（2）稳态。只要 u_I 输入触发信号不小于 $V_{CC}/3$，电路就一直维持信号输出端 u_O 复位的状态不变。这就是单稳态电路的稳态。

（3）暂态。当 u_I 输入触发信号小于 $V_{CC}/3$ 时，555 定时器芯片的置位输入端 $\overline{TRIG}$（2 脚）电压小于 $V_{CC}/3$，为其有效状态，而复位输入端 TH（6 脚）为 0V，小于 $2V_{CC}/3$，为其无效状态，所以信号输出端 u_O 置位。这时，比较器 C1 输出为 1、比较器 C2 输出为 0，基本 RS 锁存器的 Q=1、$\overline{Q}$=0，$\overline{Q}$=0 使三极管 VT 截止，VT 的集电极引出端（7 脚）不接地，电源通过电阻 R 向电容 C 充电，充电回路是 V_{CC}→R→C→地，时间常数为 $\tau_1=RC$。当电容 C 两端电压大于 $2V_{CC}/3$ 时，复位输入端 TH（6 脚）有效，信号输出端 u_O 复位，三极管 VT 饱和，VT 的集电极引出端（7 脚）接地，电容 C 放电至两端电压为 0V。

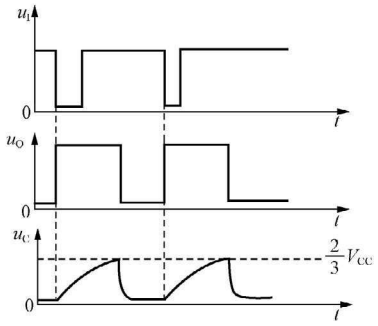


图 6-30 单稳态电路的工作波形图

单稳态电路的工作波形如图 6-30 所示。其中，触发信号下降沿有效，暂态为高电平。一定要注意，输入触发信号的低电平的时间不能太长，要小于暂态的时间。

2. 主要参数的估算

（1）输出脉冲宽度 t_W。对于图 6-29 所示的单稳态电路，输出脉冲宽度指输出为高电平的时间。由工作原理分析可知，输出脉冲宽度是等于暂稳态时间的，也就是定时电容 C 的充电时间。由图 6-30 所示工作波形可得：$u_C(0^+)\approx 0$，$u_C(\infty)\approx V_{CC}$，$u_C(t_W)=2V_{CC}/3$，代入 RC 过渡过程计算公式，可得

$$t_W=\tau_1\ln\frac{u_C(\infty)-u_C(0^+)}{u_C(\infty)-u_C(t_W)}=RC\ln\frac{V_{CC}-0}{V_{CC}-\frac{2}{3}V_{CC}}=RC\ln 3=1.1RC \qquad (6\text{-}19)$$

式（6-19）表明，单稳态电路输出脉冲宽度 t_W 仅取决于定时元件 R、C 的取值，与输入触发信号和电源电压无关，调节 R、C 即可改变脉冲宽度。

（2）恢复时间。恢复时间 t_{re} 是暂态结束后，定时电容 C 经饱和导通的三极管 VT 放电的时间，一般取 $t_{re}=(3\sim5)\tau_2$，即认为经过 3～5 倍时间常数的时间之后，电容就放电完毕。由于 $\tau_2=R_{CES}C$，而 R_{CES} 很小，所以恢复时间极短。

（3）最高工作频率。若输入触发信号 u_I 是周期为 T 的连续脉冲时，为了保证单稳态电路能够正常工作，要求 $T>t_W+t_{re}$，即 u_I 周期的最小值 $T_{min}=t_W+t_{re}$。因此，单稳态电路的最高工作频率为

$$f_{max}=\frac{1}{T_{min}}=\frac{1}{t_W+t_{re}} \qquad (6\text{-}20)$$

（二）集成单稳态电路芯片

单稳态电路的应用十分广泛，在 TTL 和 CMOS 数字集成电路中都有单片集成电路的单稳态电路芯片。使用这些器件时只需要外接很少的元件和连线，极为方便。集成单稳态电路

从触发脉冲的边沿不同来分类，有上升沿和下降沿之分；从暂态来分类，有高电平和低电平之分；从触发方式的不同来分类，可以分为可重复触发的单稳态电路和不可重复触发的单稳态电路。两种不同触发特性的单稳态电路的主要区别是：不可重复触发单稳态电路在进入暂态期间，如有触发脉冲作用，电路的工作过程不受影响，只有当电路的暂态结束后，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电路在暂态期间，如有触发脉冲作用，电路会重新被触发，使暂态继续延迟一个 t_Δ 时间，直至触发脉冲的间隔超过单稳态输出脉宽，电路才返回稳态。两种单稳态电路的工作波形分别如图 6-31（a）、（b）所示。图 6-31（a）是上升沿触发，暂态为高电平，稳态为低电平，不可重复触发。图 6-31（b）是下降沿触发，暂态为高电平，稳态为低电平，可重复触发。

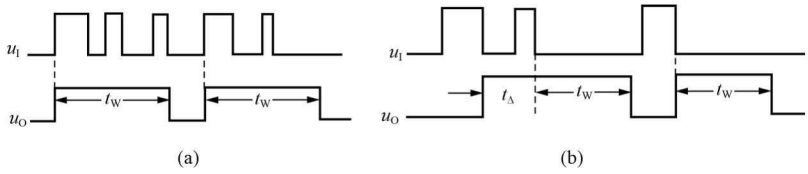


图 6-31　两种单稳态电路的工作波形

（a）上升沿触发，暂态为高电平，不可重复触发；（b）下降沿触发，暂态为高电平，可重复触发

目前市场上比较常见的是可重复触发单稳态电路芯片是 MC14528（CD4098B）。MC14528 中有两个单稳态电路。图 6-32 是 MC14528 的引脚图，表 6-2 是 MC14528 的功能表。

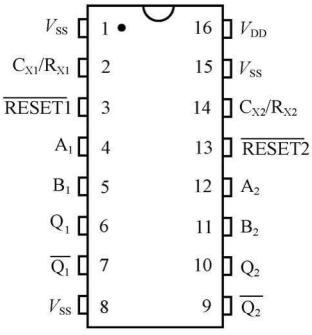


图 6-32　MC14528 的引脚图

表 6-2　　MC14528 的功能表

输入信号			输出信号	
$\overline{RESET}$	A	B	Q	$\overline{Q}$
H	↗	H	⎍	⎎
H	L	↘	⎍	⎎
H	↗↘	L	不触发	
H	H	↗↘	不触发	
H	L，H，↘	H	不触发	
H	L	L，H，↗	不触发	
L	X	X	L	H
↘↗	X	X	不触发	

（三）单稳态电路的应用

1. 延时与定时

（1）延时。图 6-33 单稳态电路作为延时与定时电路的示意图，其中的单稳态电路是下降沿触发、暂态为高电平。在图 6-33 中，u'_O 的下降沿比 u_I 的下降沿滞后了时间 t_W，实现了延时功能。

（2）定时。在图 6-33 中，当 $u'_O=1$ 时，与门打开，$u_O=u_F$。当 $u'_O=0$ 时，与门关闭，u_O为低电平。显然与门打开的时间是恒定不变的，就是单稳态输出脉冲 u'_O 的宽度 t_W。工作波形如图 6-34 所示。

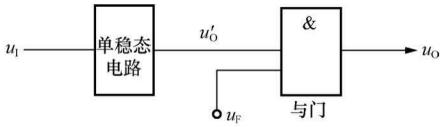


图 6-33 单稳态电路作为延时与定时电路的示意图

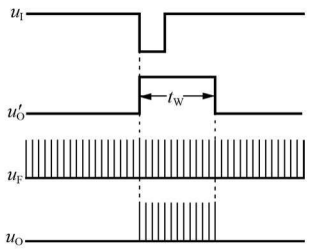


图 6-34 图 6-33 电路的工作波形

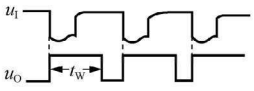


图 6-35 单稳态电路作为整形电路的波形

2. 整形

单稳态电路能够把不规则的输入信号 u_I，整形成为幅度和宽度都相同的标准矩形脉冲 u_O。u_O的幅度取决于单稳态电路输出的高、低电平，宽度 t_W 取决于单稳态电路暂稳态时间。某种单稳态电路的工作波形如图 6-35 所示，下降沿触发，暂态为高电平。

小　　结

本章重点讨论了用于脉冲波形的产生和整形电路的构成、工作原理及应用。脉冲波形产生电路主要包括由 555 定时器构成的多谐振荡器、对称和非对称式多谐振荡器、环形振荡器以及石英晶体振荡器等。脉冲波形的整形电路主要是施密特电路和单稳态电路。施密特电路和单稳态电路均可以利用 555 定时器构成。

555 定时器是一种应用十分广泛的集成器件，是模拟电路和数字电路的结合，只要外接几个阻容元件，就可以构成单稳态电路、多谐振荡器、施密特电路等。

多谐振荡器无需外加输入信号就能在接通电源后自动产生脉冲波形。多谐振荡器可以由 555 定时器或门电路组成。在频率要求较高的场合通常采用石英晶体振荡器。

施密特电路和单稳态电路可以构成各种脉冲波形的整形和变换电路，在数字电路中有十分广泛的应用。施密特电路具有滞后特性，有两个阈值电压，属于“电平触发”型电路。单稳态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。

集成单稳态电路分为可重复触发单稳态电路和不可重复触发单稳态电路两大类，在暂稳态期间出现的触发信号对不可重复触发单稳态电路没有影响，而对可重复触发单稳态电路可起到连续触发作用。

习　　题

6-1　试分别画出当图 6-36 所示电路中，开关 S 断开及闭合情况下，R_L 上的波形，并标出其主要波形参数。

6-2　某电路和其输入波形如图 6-37（a）、（b）所示，试回答：

（1）电路的名称；

（2）已知 $t_{WI}=5\mu s$，TTL 门的 $U_{OH}=3.6V$，$U_{OL}=0.3V$，$U_{th}=1.4V$，在给定参数下，

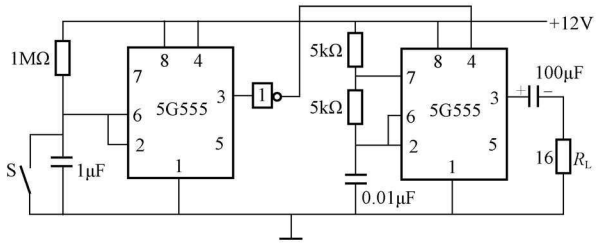


图 6-36　题 6-1 图

求输出脉冲的幅度 U_m、输出脉宽 t_{WO}；

（3）画出 u_I、u_C、u_O 的对应波形。

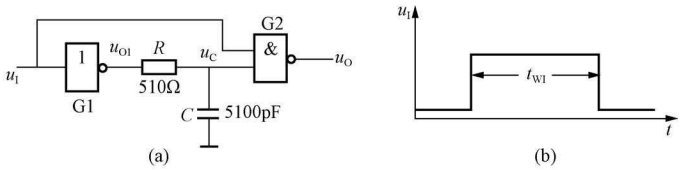


图 6-37　题 6-2 图

6-3　由主从 JK 触发器 FF 和 555 定时器组成的电路如图 6-38（a）所示，已知 CP 为 10Hz 的方波，如图 6-38（b）所示，$R_1=10k\Omega$，$R_2=56k\Omega$，$C_1=1000pF$，$C_2=4.7\mu F$，触发器端及 555 输出端（3 脚）初态均为 0。试完成：

（1）画出 Q 端，u_I、u_O 相对于 CP 脉冲的波形；

（2）求 Q 端输出波形的周期。

6-4　图 6-39（a）、（b）分别为 555 定时器构成的脉冲鉴幅电路和输入信号波形。试问为了将输入信号中大于 10V 的脉冲信号检出，电源电压 V_{CC} 应为多少？

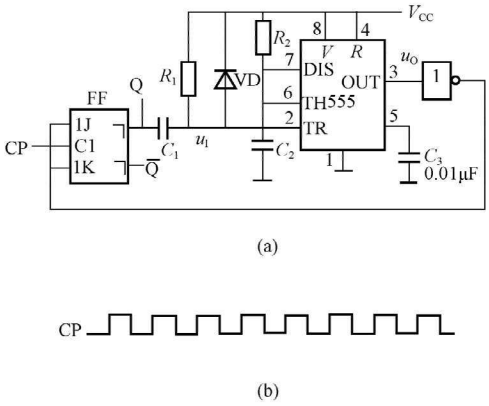


图 6-38　题 6-3 图

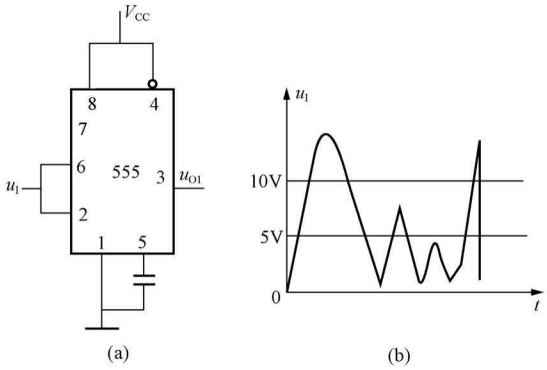


图 6-39　题 6-4 图

6-5　试回答多谐振荡器、单稳态电路、双稳态触发器各有几个暂稳态和能自动保持的稳定状态？

6-6　由集成单稳态电路 74LS121 组成的延时电路如图 6-40 所示，$C_{ext}=1\mu F$，$R_p=20k\Omega$，$R_1=5.1k\Omega$。试计算输出信号脉冲宽度的变化范围。

6-7 如图 6-41 所示，电路为由 555 定时器构成的占空比可调的多谐振荡器。电位器阻值为 1MΩ，设滑动触头到上端的极限位置时，滑动触头上边的电阻值是 R 额定值的 5%；同理，滑动触头到下端的极限位置时，滑动触头下边的电阻值是 R 额定值的 5%。假定输出电压 u_O 的高电平与电源电压相等，输出电压 u_O 的低电平是 0V，$V_{CC}=12V$，忽略二极管的导通压降及导通电阻。试回答：

（1）说明电路的工作原理；

（2）画出电压 u_C，u_O 的波形；

（3）计算输出信号 u_O 的频率；

（4）计算输出信号 u_O 的占空比变化范围。

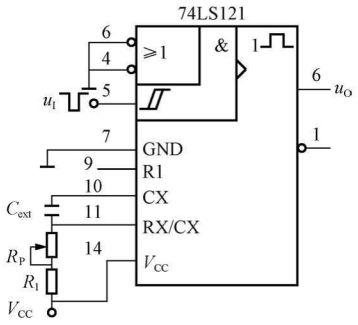


图 6-40 题 6-6 图

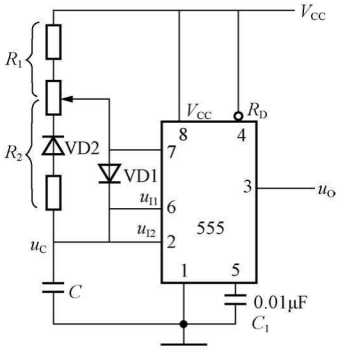


图 6-41 题 6-7 图

6-8 试说明单稳态电路的工作特点和主要用途。

6-9 利用集成施密特电路组成如图 6-42（a）所示电路，并已知该集成施密特电路的电压传输特性曲线如图 6-42（b）所示。试回答：

（1）定性画出 u_a 和 u_O 的电压波形；

（2）已知 $R=20k\Omega$，$C=0.022\mu F$，试求 u_O 的振荡周期 T。

6-10 图 6-43 为光控音调发生器电路，当光敏三极管 VT 受光较强时，5 号引脚电压 U_C 升高，反之，光弱时，U_C 减小，从而改变输出 u_O 的振荡频率，使音调发生变化。试分别计算当 U_C 为 5V 和 3V 时，u_O 的频率。

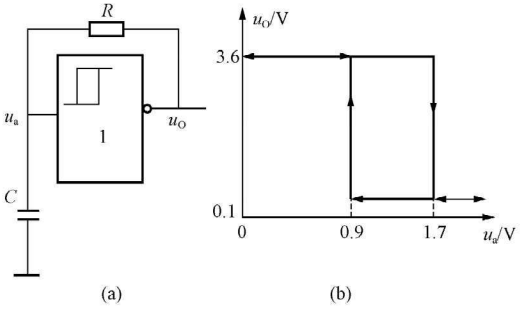


图 6-42 题 6-9 图

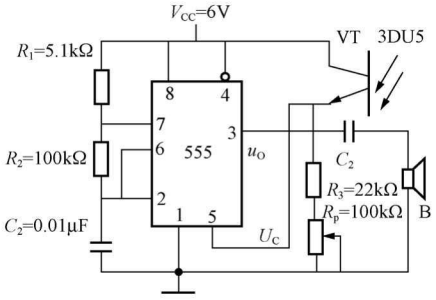


图 6-43 题 6-10 图

6-11 分析图 6-44（a）所示的 555 定时器芯片组成的电路，要求：

（1）说明电路的功能；

（2）给出上限阈值电压值和下限阈值电压的值；

（3）给定输入信号的波形如图 6-44（b）所示，画出输出 u_O的波形。

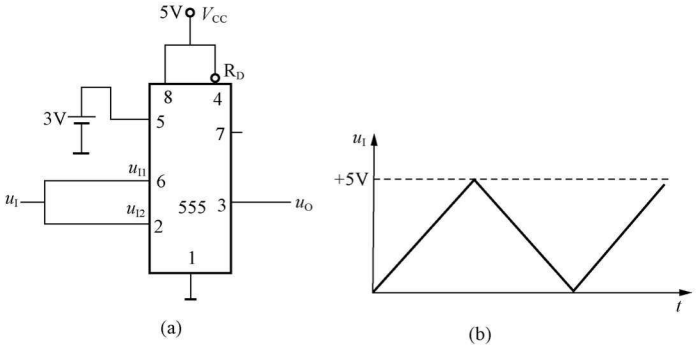


图 6-44　题 6-11 图

6-12　555 定时器芯片接成的电路如图 6-45（a）所示，回答下列问题：

（1）该电路的名称是什么？

（2）若 $V_{CC}=+10V$，$R=10k\Omega$，$C=300pF$，计算其暂态脉冲宽度 t_{po}。

（3）给定输入 u_I的波形如图 6-45（b）所示，画出电容电压 u_C的波形和对应的输出 u_O波形。

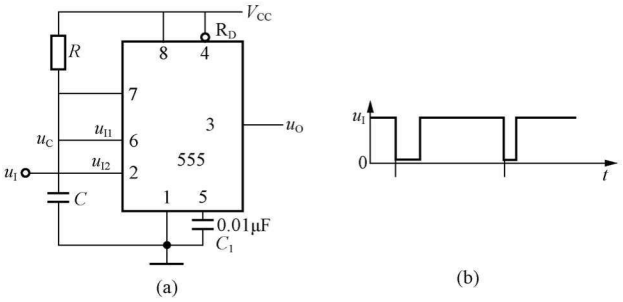


图 6-45　题 6-12 图

6-13　由 555 定时器构成的电路如图 6-46 所示。设输出高电平为 5V，输出低电平为 0V，二极管为理想二极管。已知 $R_1=33k\Omega$，$R_2=27k\Omega$，$R_3=3.3k\Omega$，$R_4=2.7k\Omega$。试回答：

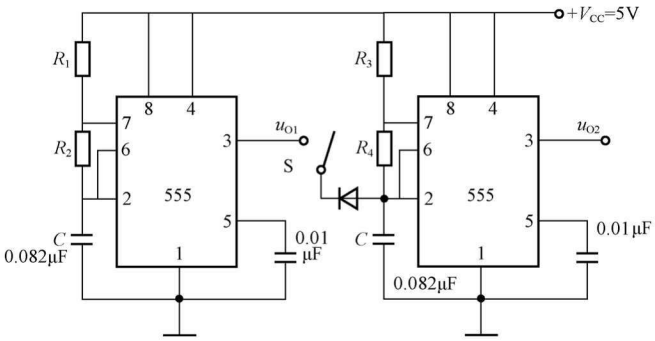


图 6-46　题 6-13 图

（1）若开关S断开，两个555定时器各构成什么电路，并计算输出信号 u_{O1}、u_{O2} 的频率之比值。

（2）若开关S闭合，画出信号 u_{O1}、u_{O2} 的波形。

6-14　某电路如图6-47所示。试回答：

（1）叙述电路工作原理；

（2）定性画出 u_{O1}、u_{O2} 的波形；

（3）计算后一级输出信号的频率。

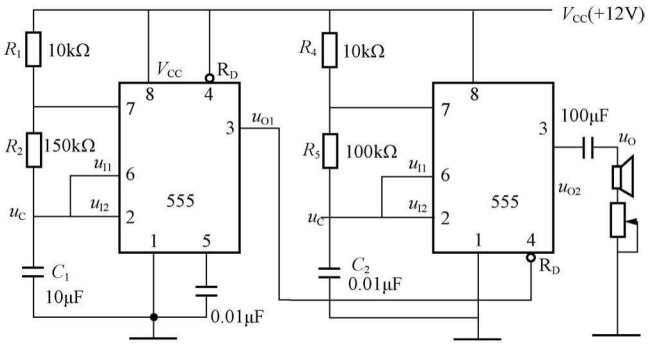


图6-47　题6-14图

6-15　试问选取何种电路才能实现如图6-48所示的各输入、输出波形。

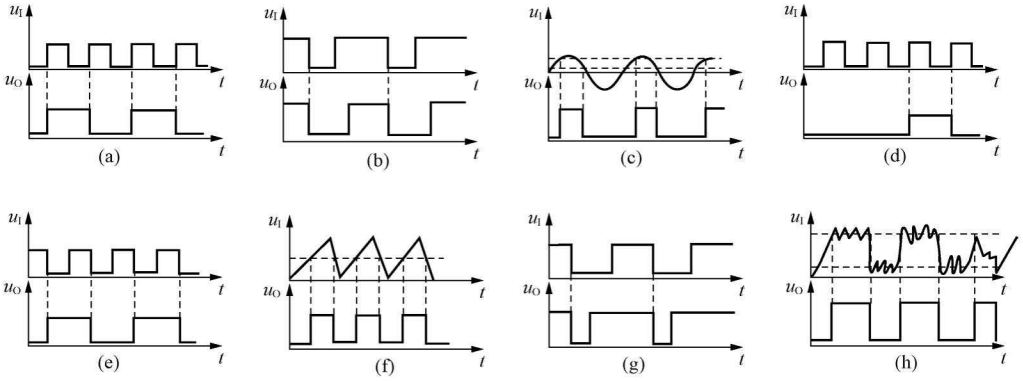


图6-48　题6-15图

6-16　用555定时器接成的电路如图6-49所示，试回答：

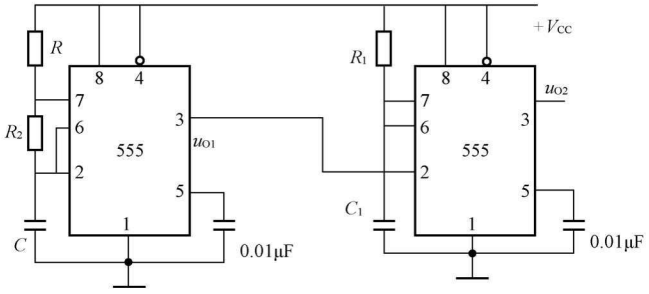


图6-49　题6-16图

（1）该电路的功能是什么？

（2）定性画出 u_{O1}、u_{O2} 的波形。

（3）若 $+V_{CC}=+10V$，$R=R_2=10k\Omega$，$C=300pF$，计算输出信号 u_{O1} 的周期 T。

6-17　电路如图 6-50 所示，在 555 定时器芯片的外部 5 脚接了一个电位器，求电压比较器 C1 的同相输入端电位和电压比较器 C2 的反相输入端电位。

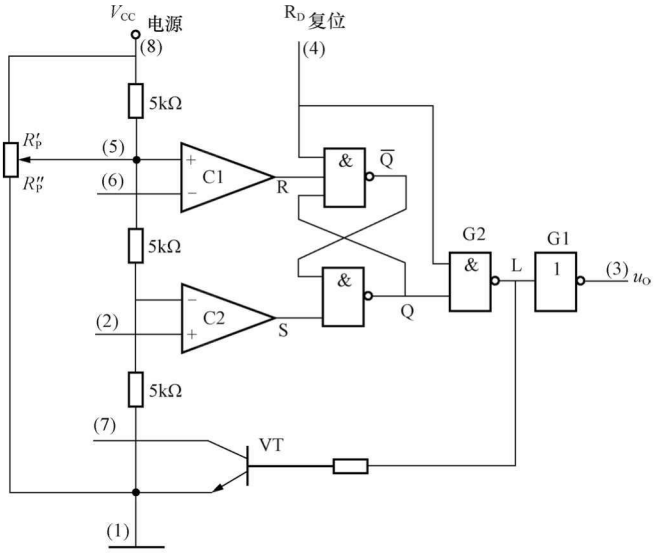


图 6-50　题 6-17 图

第七章　半 导 体 存 储 器

本　章　提　要

本章主要介绍半导体存储器的基本概念和分类，只读存储器（ROM，Read Only Memory）和随机存取存储器（RAM，Random Access Memory）的功能、结构和应用，以及存储器容量的扩展。

第一节　存储器的基本概念和分类

一、存储器的基本概念

存储器（Memory）是数字系统中用于存储大量二进制信息的部件，是现代计算机的重要组成部分之一。存储器用来存放程序和数据。中央处理器（CPU）将程序或数据从存储器中取出，经译码后使电路完成相应的操作。

二、存储器的分类

根据存储器在计算机系统中的不同地位，存储器可分为主存储器（Main Memory）和辅助存储器（Secondary Memory）。主存储器用来存放当前运行所需要的程序和数据，以便向中央处理器快速地提供信息。相对于辅助存储器而言，主存储器的存取速度快，但容量有限，且价格较高。由于主存储器设置在主机内部，所以又可以称为内存储器。辅助存储器用来存放当前暂时不参与运行的程序和数据，其特点是存储容量大，价格低，但存取速度慢。由于辅助存储器设置在主机外部，故又称为外存储器。

根据存储器的存储介质的不同来划分，存储器有半导体存储器、磁存储器、光存储器和激光光盘存储器，其中磁存储器包括磁芯、磁泡、磁鼓、磁带和磁盘等。半导体存储器的存取速度较快，在计算机系统中用做主存储器（内存储器）。磁存储器、光存储器和激光光盘存储器的速度较慢，在计算机系统中用做辅助存储器（外存储器）。随着集成电路技术的发展，半导体存储器也开始作为外存储器使用。

本章只介绍半导体存储器。

第二节　半 导 体 存 储 器

一、半导体存储器的分类

半导体存储器是由半导体材料制作成的。半导体存储器的种类繁多，从不同的角度来看有不同的分类：

1. 从制造工艺的角度分类

半导体存储器按其制造工艺的不同可以分为双极型半导体存储器和单极型半导体存储器。双极型半导体存储器是用双极型半导体三极管工艺制成的存储器，其特点是工作速度快，功耗不大，集成度低。计算机中的高速缓冲存储器经常采用双极型半导体存储器。单极

型半导体存储器是用 MOS 电路工艺制成的存储器，其特点是集成度高，功耗低，价格便宜，而且随着半导体集成工艺和技术的长足进展，目前这种 MOS 存储器的工作速度也已经能和双极型半导体存储器相比美。

2. 从工作方式的角度分类

半导体存储器按其不同的工作方式可以分为只读存储器（ROM）和随机存取存储器（RAM）两大类。其详细分类如下所示。

- 半导体存储器
  - RAM
    - 异步 RAM
      - 异步静态 RAM（ASRAM）
      - 异步动态 RAM（ADRAM）
    - 同步 RAM
      - 同步静态 RAM（SSRAM）
      - 同步动态 RAM（SDRAM）
  - ROM
    - 掩模 ROM（MROM）
    - 一次可编程 ROM（OTPROM）
    - 紫外线擦除多次可编程 ROM（UVEPROM）
    - 电擦除多次可编程 ROM（EEPROM）
    - 闪速存储器（FLASH MEMORY）

二、半导体存储器的三个主要技术指标

1. 存储容量

存储容量是指存储器可以存储的二进制信息量。

2. 存储速度

存储器的基本操作是读出与写入，总称为“访问”或“存取”。有关存储器的存储速度有以下两个时间参数：

（1）访问时间：指从启动一次存储器操作，到完成该操作所经历的时间。

（2）存储周期：指启动两次独立的存储器操作所需的最小时间间隔。

3. 存储器的可靠性

存储器的可靠性用平均故障间隔时间来衡量。平均故障间隔时间越短，说明存储器的可靠性越高。

第三节　只 读 存 储 器（ROM）

早期的只读存储器（Read Only Memory，ROM）中所保存的数据在数字系统正常工作状态下只能从 ROM 中读出，不能往 ROM 中写入，ROM 中所保存的数据是 ROM 在进入正常工作状态之前用专用的设备写入的。这种早期的 ROM 根据电路结构和不同的信息写入方式可分为掩模 ROM、一次可编程 ROM、紫外线擦除多次可编程 ROM。

（1）掩模 ROM（Mask-programmed ROM，MROM）也叫做固定 ROM，其中的数据在工厂制作时写入，出厂后用户使用时不能再对其进行修改。

（2）一次可编程 ROM（One Time Programmable ROM，OTPROM）可由用户在自己的实验室使用专用的写入设备将数据写入，一旦写入后内容不能再更改，即只能写入一次，若想再写入新的数据，必须更换一片新的芯片。

（3）紫外线擦除多次可编程 ROM（Ultra Violet Erasable Programmable ROM，UVEPROM）可由用户在自己的实验室用专用的写入设备将数据写入，若想修改数据，可以用专

用的设备，比如紫外线灯等设备，照射芯片 30min 左右就可将原来的数据擦除，用专用的写入设备可以再次写入新的数据。芯片可以重复使用多次。

随着半导体技术的发展，目前的 ROM 产品在数字系统正常工作状态下不但可以读出数据，而且可以写入数据。这种类型的产品有电擦除多次可编程 ROM（Electricatly Erasable Programmable ROM，EEPROM）和闪速存储器（FLASH MEMORY）。ROM 产品具有非易失性，即系统掉电后所存储的数据不丢失。

一、只读存储器（ROM）的一般结构

（一）只有一个地址译码器的 ROM

只有一个地址译码器的 ROM 的一般结构示意图如图 7-1 所示。ROM 由地址译码器、存储矩阵和输出缓冲电路三部分组成。输出缓冲电路用的是三态门，它有两个作用：一是提高带负载能力；二是实现对输出端状态的控制，以便于和系统总线连接。

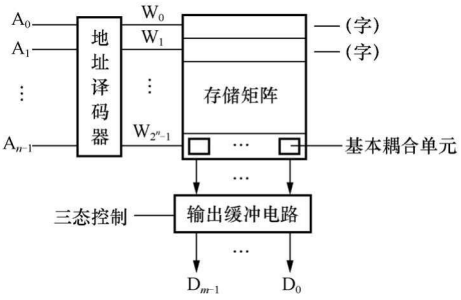


图 7-1 只有一个地址译码器的 ROM 的一般结构示意图

ROM 以若干个基本耦合单元为单位进行数据的读出，这样的一个单位称为一个字。存储矩阵中所有字的个数的总和称为字数。每个字中的基本耦合单元的个数称为字长。ROM 的存储矩阵中有很多的字，那么如何来区分这些不同的字呢？在日常生活中，我们人类每个个体都有一个名字，用来区分不同的个体。在数字系统中，电路无法识别汉字组合而成的名字，但是它可以识别由若干位二进制数码组合而成的名字。在 ROM 中，每个字被赋予了若干位二进制数码，这样的一组二进制数码的组合称为这个字的地址。可以这样理解，地址是给存储矩阵中各个字所起的名字，这个名字是用一组二进制数码的组合来表示的。存储矩阵中字的个数越多，则地址的位数就越多。地址译码器的输出线（见图 7-1）W_0，W_1，…，W_{2^n-1}称为字线。存储矩阵中的每个字分别对应了一根字线，当要读出这个字中的基本耦合单元中的数据时，应该使这根字线有效。对于某个具体的存储器芯片来说，可能是低电平有效，也可能是高电平有效。对于一个存储器来说，如果它有 n 根地址线，其最大可能的字数为 2^n 个。

每个字中包含的基本耦合单元的数据线称为该 ROM 的位线，位线的根数称为位数。ROM 中存放二进制信息的多少用存储容量来表示。存储容量是该 ROM 的字数和位数的乘积，即总的基本耦合单元的个数。对于有 n 位地址输入，m 位数据输出端的 ROM，其存储容量为 2^n（字）$\times m$（位）。

用二极管组成的地址译码器和存储矩阵的 2^3（字）$\times 4$（位）的 ROM 的电路结构如图 7-2 所示。地址译码器输出高电平有效，即字线高电平有效。

在第二章中曾讲过的二极管正与门电路如图 7-3 所示，逻辑关系为 M=XY。二极管正与门电路构成地址译码器。在第二章中曾讲过的二极管正或门电路如图 7-4 所示，逻辑关系为 L=A+B。二极管组成的正或门在此作为基本耦合单元构成存储电路。

下面以图 7-2 中所示的 2^3（字）$\times 4$（位）的 ROM 为例来具体说明二进制数据的读出过程。

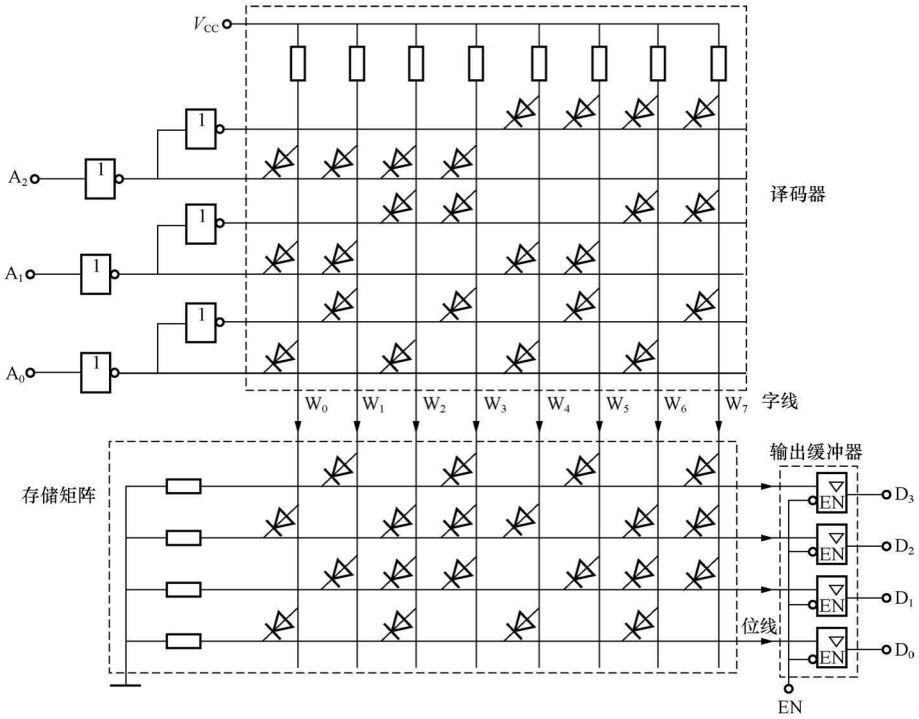


图 7-2　用二极管组成的 ROM 的电路结构

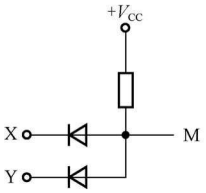


图 7-3　二极管正与门电路

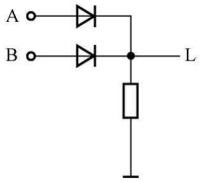


图 7-4　二极管正或门电路

当 $A_2A_1A_0=000$ 时，字线 W_0 输出高电平，其他字线均输出低电平，得知 $D_3D_2D_1D_0=0101$。图 7-5 给出了当 $A_2A_1A_0=000$ 时每根信号的电平。

同理，可得出当 $A_2A_1A_0$ 分别为 001、010、011、100、101、110、111 时，$D_3D_2D_1D_0$ 分别为 1010、0111、1110、0101、1010、0111、1110。

ROM 的电路结构比较复杂，为了抓住主要矛盾，能够清楚地看到存储矩阵中的存储内容，引入了 ROM 的一种简便表示方法，即 ROM 的阵列图。图 7-2 所示 ROM 电路的阵列图表示如图 7-6 所示。在译码器电路（二极管与门电路）中，地址输入线与字线相交处有二极管的地方用一个圆点表示，二极管则被省略掉。在存储矩阵（二极管或门电路）中，字线与位线相交处有二极管的地方用一个圆点表示，二极管则被省略掉。在阵列图表示的存储矩阵中，不妨可以这样想，有黑点表示存储了 1，没有黑点表示存储了 0。图 7-7 用表格的方法列出了存储矩阵中各个字的内容。

（二）具有两个地址译码器的 ROM

对于存储容量稍大的存储器，其地址码或地址线位数较多，如果如前面所述用一个地址

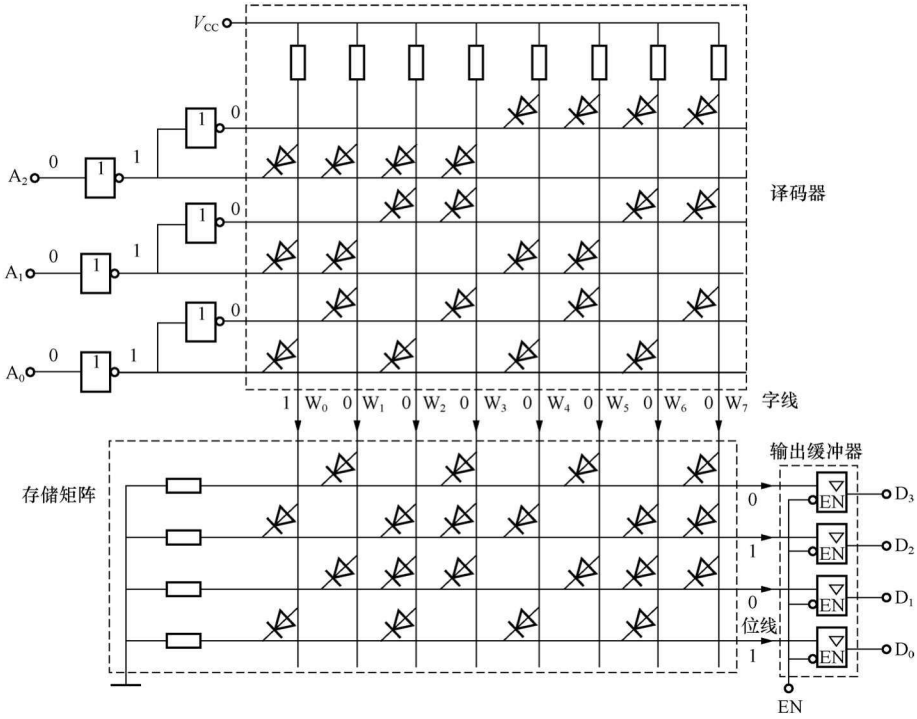


图 7-5 当 $A_2A_1A_0=000$ 时的数据读出过程示意图

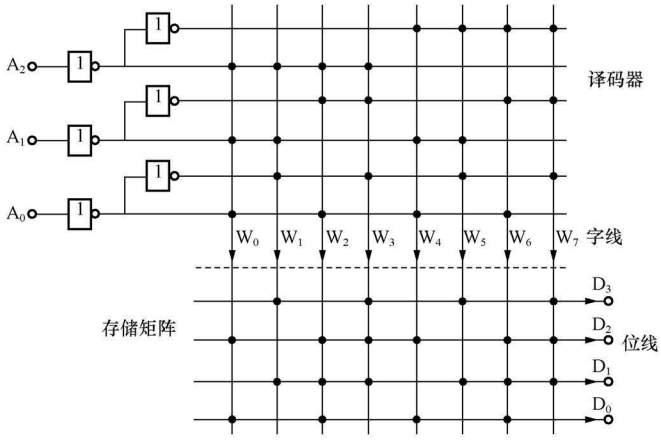


图 7-6 ROM 电路的阵列图表示

译码器直接对地址进行译码，仅地址译码器电路就非常庞大。所以为了简化电路，通常是将基本耦合单元组成阵列方式，使其垂直高度和水平高度的尺寸为同一数量级，即高宽比约为 1∶1。这样在同一行中就会有多个字，行地址译码器的输出信号用来选中某一行，但我们只希望选中其中的某一个字，为此必须加入列地址译码器，其作用就是从被选中的行中将所需的某个字选出来，即只有同

D_3	0	1	0	1	0	1	0	1
D_2	1	0	1	1	1	0	1	1
D_1	0	1	1	1	0	1	1	1
D_0	1	0	1	0	1	0	1	0
	字	字	字	字	字	字	字	字

图 7-7 ROM 存储矩阵中各个字的内容

时被行地址译码器和列地址译码器选中的某个字，才能进行读操作。

具有两个地址译码器的 2^{10}（字）×1（位）ROM 的一般结构示意图如图 7-8 所示。

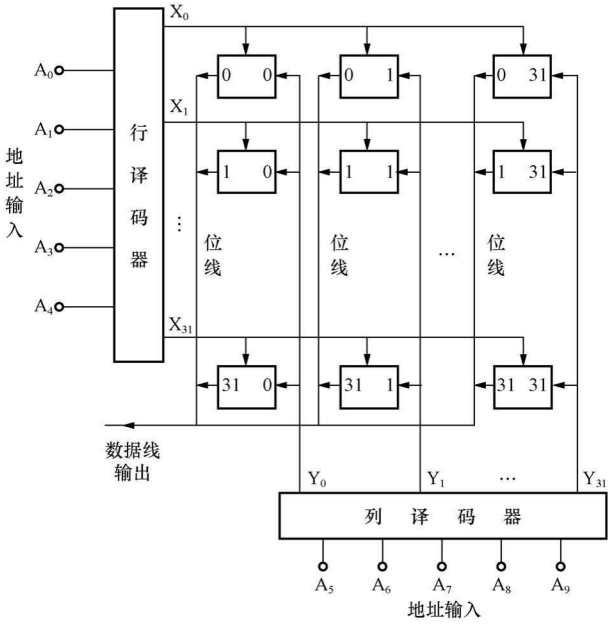


图 7-8　具有两个地址译码器的 2^{10}（字）×1（位）ROM 的一般结构示意图

举例说明数据读出过程：图 7-8 使用正逻辑体制。假设此 ROM 电路的两个地址译码器输出均为高电平有效，基本耦合单元的电路结构没有给出，只是用一个方框图表示。在此假定下当选中某个字时数据线上输出 1，即基本耦合单元中存储了 1。

当 $A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0=0000000000$ 时，数据读出过程如图 7-9 所示。此时，

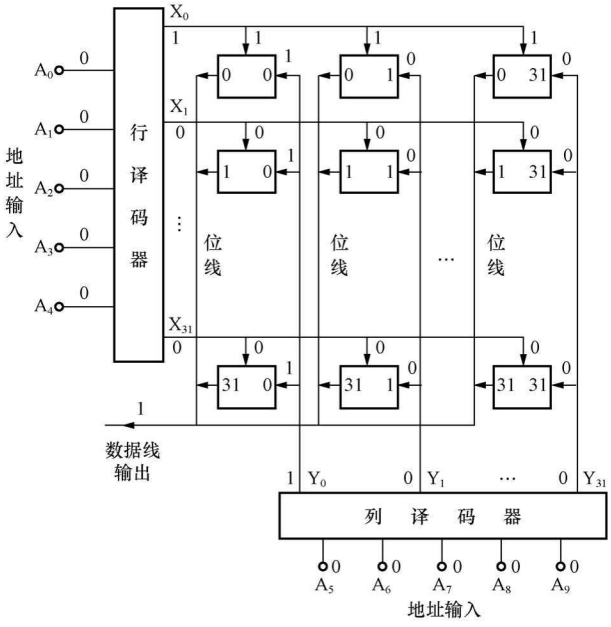


图 7-9　数据读出过程示意图

行地址译码器的 32 根输出线 X_0，X_1，…，X_{31} 中，只有 X_0 是高电平 1，其余全部是 0。列地址译码器的 32 根输出线 Y_0，Y_1，…，Y_{31} 中，只有 Y_0 是高电平 1，其余全部是 0。所以只有连接 X_0 和 Y_0 的那个基本耦合单元被选中，其余所有的基本耦合单元都不能被选中，数据从连接 X_0 和 Y_0 的那个基本耦合单元输出 1。

二、ROM 的基本耦合单元

从图 7-5 可以看出，ROM 中所存放的“1”或“0”信息，是靠字线和位线交叉处有无跨接于其间的二极管来实现的。有跨接于其间的二极管表示存储了“1”，没有跨接于其间的二极管则表示存储了“0”。然而在某些 ROM 存储矩阵中可能与之相反，这取决于地址译码器输出是高电平有效还是低电平有效和存储矩阵的电路结构。

严格地说，这种二极管并无存储信息的功能，只是反映了一种通、断的状态，所以将这种存储单元称为基本耦合单元。除了二极管外，双极型三极管、MOS 管等也可以作为基本耦合单元。

（一）掩模 ROM（MROM）的基本耦合单元

掩模 ROM 材料成本低廉，但是掩模制作成本较高，适用于大批量成熟产品的定制生产。二极管构成基本耦合单元的电路在前面已经有详细地介绍，下面介绍双极型三极管和 MOS 管构成的掩模 ROM 的基本耦合单元电路。

1. MOS 管构成的基本耦合单元

MOS 管构成基本耦合单元的掩模 ROM 电路结构如图 7-10 所示。MOS 管为 P 沟道增强型 MOS 管，此处采用了简化符号。MOS 管的栅极分别接 4 根字线 W_0、W_1、W_2、W_3，所有源极接地，漏极分别接 4 根位线上，位线经上拉电阻接正电压源，位线经三态反相器接输出数据线 D_3、D_2、D_1、D_0。

图 7-10 中，地址译码器输出的字线为低电平有效，当基本耦合单元中有 MOS 管时表示存储了 1。当 $A_1A_0=00$ 时，数据的读出过程如图 7-11 所示。$W_0=0$，$W_1=1$，$W_2=1$，$W_3=1$，因为字线 W_0 为低电平，又因为 MOS 管为 P 沟道增强型，所以与 W_0 相连的两个 MOS 管导通，漏极和源极相当于连接在一起。MOS 管的源极接地，所以这两个 MOS 管的漏极接地，对应的位线被拉成低电平，因为 EN=0，经反相后输出高电平。所以当 $A_1A_0=00$ 时，$D_3D_2D_1D_0=1010$。同理可得出，当 $A_1A_0=01$ 时，$D_3D_2D_1D_0=1001$；当 $A_1A_0=10$ 时，$D_3D_2D_1D_0=0101$；当 $A_1A_0=11$ 时，$D_3D_2D_1D_0=1111$。

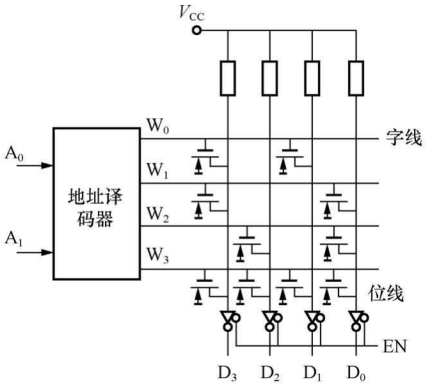


图 7-10　2^2 字×4（位）MOS 管 ROM 电路结构图

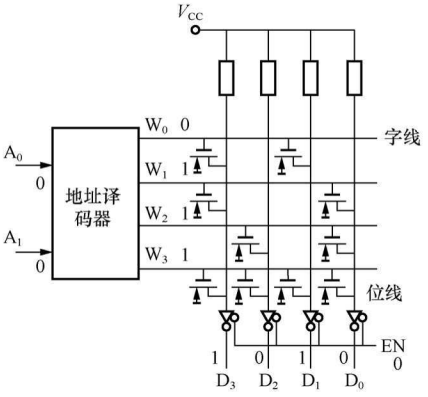


图 7-11　图 7-10 所示电路数据的读出过程示意图

2. 双极型三极管构成的基本耦合单元

双极型三极管构成的基本耦合单元的电路图如图 7-12 所示。

当字线为高电平时，三极管处于饱和状态，三极管的发射极电位和集电极电位近似相等，为 V_{CC}，所以位线输出高电平。当字线为低电平时，三极管处于截止状态，三极管的发射极与位线断开，位线经过下拉电阻接地，所以位线输出低电平。

（二）一次可编程 ROM（OTPROM）的基本耦合单元

OTPROM 在出厂时，所有的基本耦合单元为全通或全断，也即内部为全 1 或全 0。用户使用前可用专门的设备改变某些基本耦合单元的通、断状态，从而将自己的数据写入 OTPROM 中，数据一旦写入后不可再作修改。OTPROM 相当于由用户自己完成掩模 ROM 生产中写入信息的工序，其余工序与掩模 ROM 完全相同。OTPROM 适用于小批量生产。

OTPROM 的基本耦合单元有熔丝型、结破坏型和反熔丝型三种。

1. 熔丝型

熔丝型的 OTPROM 芯片出厂时，每个字线、位线交叉处均接有带易熔金属丝的 MOS 管或双极型三极管，如图 7-13 所示。图 7-13（a）中的 MOS 管为 N 沟道增强型，此处采用了简化符号。用户写入信息时，通过地址输入线逐字、逐位扫描 OTPROM 中的基本耦合单元，并根据要写入的数据内容，有选择地将某些耦合管的熔丝用规定的脉冲电流烧断，而其余耦合管的熔丝则保留。耦合管的熔丝的通、断反映了互补的两种逻辑状态。

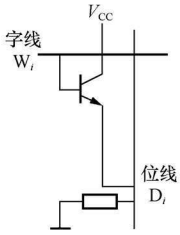


图 7-12　双极型三极管构成的基本耦合单元

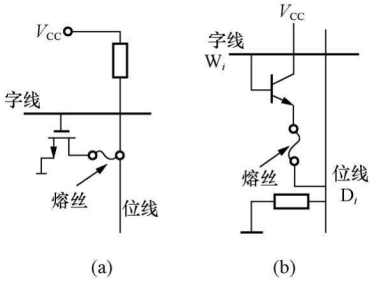


图 7-13　熔丝型的 OTPROM 的基本耦合单元

（a）熔丝型的 MOS 管作为基本耦合单元；（b）熔丝型的双极型三极管作为基本耦合单元

在图 7-13（a）中，当地址译码器输出的字线为高电平有效时，对于 N 沟道增强型 MOS 管来说，漏极和源极相当于连接在一起。因为 MOS 管的源极接地，若熔丝通则位线被拉至低电平，在正逻辑体制下，表示输出 0；若熔丝烧断则位线被上拉电阻接至电源，为高电平，在正逻辑体制下，表示输出 1。

在图 7-13（b）中，双极型 NPN 三极管的集电极接正电压源，发射极经熔丝接位线，基极接字线。当地址译码器输出的字线为高电平有效时，NPN 三极管处于饱和状态，发射极相当于与电源连接在一起。若熔丝通则位线被接至高电平，在正逻辑体制下，表示输出 1；若熔丝烧断则位线被下拉电阻接至低电平，在正逻辑体制下，表示输出 0。

2. 结破坏型

结破坏型的 OTPROM 芯片出厂时，每个字线、位线交叉处均接有一对阴极与阴极相接的二极管，如图 7-14（a）所示。无论字线为高电平还是低电平，二极管 VD1 和 VD2 不能

同时导通，所以不能使字线和位线之间建立起电路的连接。

如图 7-14（b）所示，若使用规定的电流将二极管 VD2 击穿短路，则当字线为高电平时，二极管 VD1 导通，使字线和位线连接在一起，位线电位和字线电位相等，使位线输出为高电平。当字线为低电平时，二极管 VD1 截止，位线经下拉电阻接地，所以位线输出为低电平。

3. 反熔丝型

反熔丝型的 OTPROM 的基本耦合单元中通过击穿介质达到连通线路的目的。Actel 公司的可编程低阻电路元件 PLICE（Programmable Low Impedance Circuit Element）多晶硅反熔丝的结构如图 7-15 所示。PLICE 反熔丝是位于 N^+ 扩散区（Diffusion）和多晶硅（Polysilicon）之间的介质（Dielectric），是和 CMOS 以及其他工艺（如双极型、BiCMOS 等）相兼容的。在未编程状态下，反熔丝呈现十分高的阻抗（>100MΩ）；当 18V 的编程电压加在其上时，介质被击穿，两层导电材料连在一起，接通电阻小于 1kΩ。反熔丝占用的硅片面积非常小，十分适宜于作集成度很高的可编程器件的编程元件。出厂时 PLICE 反熔丝是断开的，通过编程可使其导通，这与熔丝工艺的工作原理正好相反，所以叫反熔丝工艺。

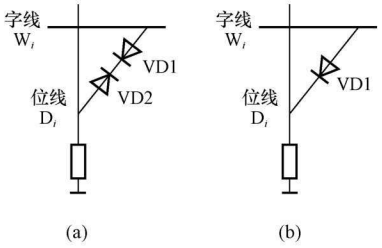


图 7-14 结破坏型的 OTPROM 的基本耦合单元

（a）基本耦合单元的一种逻辑状态；（b）基本耦合单元的另一种逻辑状态

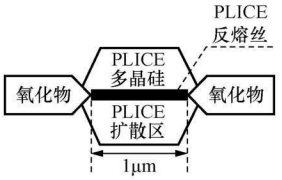


图 7-15 PLICE 多晶硅反熔丝的结构示意图

（三）紫外线擦除多次可编程 ROM（UVEPROM）的基本耦合单元

UVEPROM 可由用户在自己的实验室用专用的写入设备将数据写入，若想修改数据可以用专用的设备（如紫外线灯）照射芯片一定的时间就可将原来的数据擦除，然后用专用的写入设备可以再次写入新的数据。芯片可以重复使用 100 次以上。UVEPROM 的基本耦合单元采用浮栅雪崩注入 MOS 管（Floating Avalanche MOS，FAMOS）或叠栅注入 MOS 管（Stacked gate avalanche-Injection-type MOS，SIMOS）构成。

1. FAMOS 管

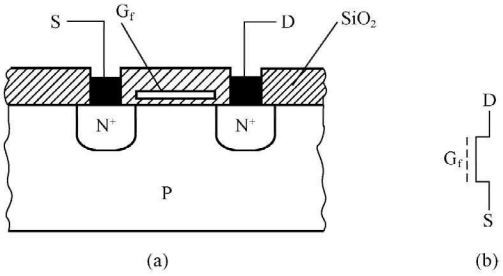


图 7-16 N 沟道增强型的 FAMOS 管的横断面示意图和图形符号

（a）横断面示意图；（b）图形符号

FAMOS 管与普通的 MOS 管类似，FAMOS 管可制作成 P 沟道增强型或 N 沟道增强型。下面以 N 沟道增强型为例进行讲解。N 沟道增强型的 FAMOS 管的横断面示意图和图形符号如图 7-16 所示。该 FAMOS 管用多晶硅材料制作栅极，栅极被二氧化硅绝缘层包围，因为栅极没有对外引出连线，好像浮在氧化层中，故称为“浮栅”。如果在漏极和源极之间加上高电压，则可使漏极和衬底之间的 PN 结产生雪崩击穿，漏极的 N^+

区中的电子在强电场的作用下以很高的速度向外射出，其中速度最快的一部分电子穿过二氧化硅绝缘层而到达浮栅，被浮栅俘获而形成栅极的存储电荷，这个过程叫做雪崩注入。漏极和源极之间的高电压去掉后，由于注入到栅极上的电荷没有放电通路，所以能长久保存下来。在栅极获得足够的电荷后，漏极和源极之间便形成导电沟道，使 FAMOS 导通。

若要擦去浮栅上的存储电荷，可以用专用设备产生的紫外线照射芯片的石英玻璃窗口，使浮栅上的电子获得足够的光子能量，而穿越绝缘层回到衬底中。这样芯片就又恢复到初始状态，这个擦去信息的芯片就又可以重新写入。写入数据后要用黑胶纸把芯片的玻璃窗口封好，以防光线干扰片内存储的信息。

2. 用 N 沟道增强型 FAMOS 管构成的基本耦合单元

用 N 沟道增强型 FAMOS 管构成的基本耦合单元示意图如图 7 - 17 所示。用 N 沟道增强型的 FAMOS 管做基本耦合单元时，还要用一只普通的 NMOS 管 VT1 与之串联，这只普通 NMOS 管的栅极受字线控制。产品在出厂时所有的 FAMOS 都处于截止状态，所以字线和位线断开，位线经一个处于饱和区的 NMOS 管 VT2 连接到正电压源上，位线输出高电平；所有的基本耦合单元中都存储了 1。

在进行写入数据 0 操作时，首先选好地址，使需要写入数据的那些单元对应的字线为高电平，然后在需要写入 0 的那些基本耦合单元的位线上加上正脉冲，使被选中的基本耦合单元中的 FAMOS 管发生雪崩击穿，就相当于在基本耦合单元中写入了 0。

在正常工作状态需要读出数据时，选好需要读出数据的字单元的地址，经地址译码器译码后，这些字单元所对应的字线为高电平。这根字线所接的一行字单元中栅极已经注入电荷的 FAMOS 管导通，使所接的位线变成低电平，读出 0；栅极没有注入电荷的 FAMOS 管不导通，使所接的位线变成高电平，读出 1。

因为采用 FAMOS 管的基本耦合单元需要用两只普通 MOS 管，所以单元面积较大，而且产生雪崩击穿所需要的电压也比较高。此外 NMOS 管的速度也比较低。所以目前多使用叠栅注入 MOS（SIMOS）管制作 UVEPROM 的基本耦合单元。

3. SIMOS 管

SIMOS 管与普通的 MOS 管类似，可制作成 P 沟道增强型或 N 沟道增强型。下面以 N 沟道增强型为例进行讲解。N 沟道增强型 SIMOS 管的横断面示意图和图形符号如图 7 - 18 所示。它有两个多晶硅栅极 G_c 和 G_f，上面的栅极 G_c 与普通 NMOS 管的栅极作用相同，称为控制栅极，用于控制读出和写入。下面的栅极 G_f 埋在二氧化硅绝缘层中，称为“浮栅”，用于长期保存电荷。

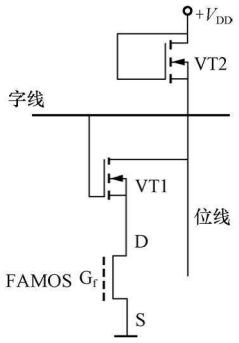


图 7 - 17　FAMOS 管构成的基本耦合单元

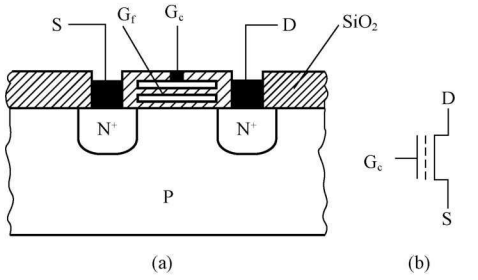


图 7 - 18　SIMOS 管的横断面示意图和图形符号

（a）横断面示意图；（b）图形符号

（1）SIMOS 管浮栅注入电子的过程。用户编程时，如图 7-19（a）所示，可在 SIMOS 管的漏极和源极之间加上高电压（20V），使得沟道内的电场足够强而形成雪崩，产生很多高能电子。此时在控制栅极上加上高压正脉冲（比如 20V 的 50ms 宽的正脉冲），借助控制栅极正脉冲电压的吸引，就会有一部分高能电子穿过二氧化硅绝缘层进入浮栅，这时编程过程完成。

如图 7-19（b）所示，当控制栅极上的高电压正脉冲去掉后，由于浮栅被绝缘层包围，它所俘获的电子很难泄露掉，故可长期保留。

（2）正常工作时 SIMOS 管的工作状态。在存储器芯片正常工作时，在控制栅极上加入 5V 的正电压，如图 7-20 所示。

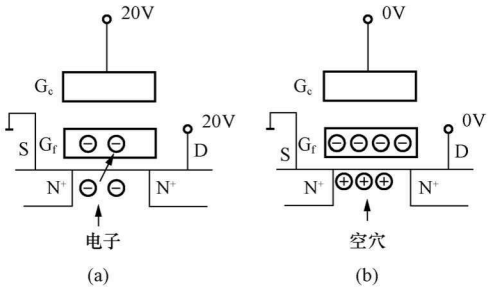


图 7-19　SIMOS 管浮栅注入电子的过程示意图

（a）雪崩注入过程示意图；（b）去掉编程电压后 SIMOS 管的状态示意图

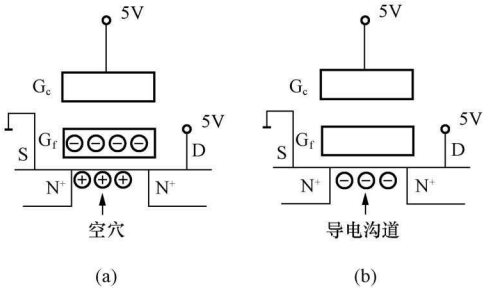


图 7-20　SIMOS 管正常工作时的状态

（a）浮栅带上电子后 SIMOS 管的工作状态；（b）浮栅不带电子时 SIMOS 管的工作状态

如图 7-20（a）所示，当浮栅带有电子时，衬底表面感应的是带正电的空穴，此时的开启电压将比浮栅未带上电子时的高。对于 N 沟道增强型 SIMOS 管来说，导电沟道是自由电子层。此时在控制栅极上加上＋5V 的电压，不能抵消浮栅上电子的影响，不能形成导电沟道。

如图 7-20（b）所示，当浮栅没有电子时，衬底表面没有感应产生的带正电的空穴，此时的开启电压将比浮栅带上电子时的低，能够形成导电沟道。

（3）SIMOS 管浮栅电子被擦除的过程。若要擦去所写入的数据，可以用专用设备产生的紫外线照射 UVEPROM 芯片的石英玻璃窗口，对每个基本耦合单元中的浮栅均照射一定时间，使浮栅上的电子获得足够的光子能量，而穿越绝缘层回到衬底中。这样芯片就又恢复到初始状态，擦去信息的芯片就又可以重新写入。

4. 用 N 沟道增强型 SIMOS 管构成的基本耦合单元

用 N 沟道增强型 SIMOS 管构成的基本耦合单元示意图如图 7-21 所示。

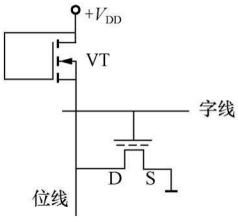


图 7-21　SIMOS 管构成的基本耦合单元

用 N 沟道增强型的 SIMOS 管做基本耦合单元时，不再需要一只普通的 NMOS 管与之串联，一只 SIMOS 管就可以构成一个基本耦合单元。SIMOS 管的控制栅极与字线相连，漏极与位线相连，源极接地，位线经处于饱和区的 N 沟道增强型 MOS 管 VT 接电源。芯片封装出厂时，片内所有基本耦合单元都接有这种 SIMOS 管，并且它们的浮栅均无电子。在存储器芯片正常工作时，在控制栅极上加入 5V 的正电压时，可以使该管导通，从而使位线为低电平，即从

正逻辑的角度看，所有的基本耦合单元中都存储了 0。

在正常工作状态需要读出数据时，选好需要读出数据的字单元的地址，经地址译码器译码后，这些字单元所对应的字线为＋5V 的高电平。这根字线所接的一行字单元中栅极已经注入电子的 SIMOS 管不能导通，使所接的位线输出高电平，从正逻辑的角度看读出 1；栅极没有注入电子的 SIMOS 管导通，使所接的位线变成低电平，读出 0。

（四）电擦除电可编程 ROM（EEPROM）的基本耦合单元

EEPROM 的基本耦合单元是隧道 MOS 管（FLOating-gate Tunneling OXide，FLOTOX）。同 SIMOS 管一样，FLOTOX 管也是利用浮栅上是否存有电荷来存储二值数据的，不同的是 FLOTOX 管是用电擦除的，并且擦除的速度要快得多。

1. FLOTOX 管

FLOTOX 管可制作成 P 沟道增强型或 N 沟道增强型。下面以 N 沟道增强型的 FLOTOX 管为例进行讲解。FLOTOX 管的横断面示意图和图形符号如图 7－22 所示。与 SIMOS 管相似，FLOTOX 管也有两个栅极，即控制栅 G_c 和浮栅 G_f，不同的是 FLOTOX 管的浮栅与漏极区（N^+）之间有一小块面积极薄的二氧化硅绝缘层的区域，称为隧道区。

当漏极接地，控制栅极上加上足够高的电压，在隧道区将产生电场。当电场强度大到一定程度时，漏区和浮栅之间出现导电隧道，电子通过绝缘层到达浮栅，使浮栅带负电荷。

若将控制栅极接地，漏极加上足够高的电压，则产生与上述相反的过程，浮栅上的负电荷消失。

FLOTOX 管的浮栅上负电荷的消失不再像 UVEPROM 那样需要借助于紫外线，所以比 UVEPROM 使用起来更灵活方便，而且还可以以字为单位进行擦除，而不像紫外线擦除那样整片内容被擦掉。

2. FLOTOX 管构成的基本耦合单元

N 沟道增强型的 FLOTOX 管构成的基本耦合单元如图 7－23 所示。为了提高写入和擦除数据的可靠性，用 N 沟道增强型的 FLOTOX 管做基本耦合单元时，还要用一只普通的 N 沟道增强型的 MOS 管与之串联，这只普通 NMOS 管的栅极受字线控制，漏极与位线相连，漏极经过上拉电阻接＋5V 电压源。

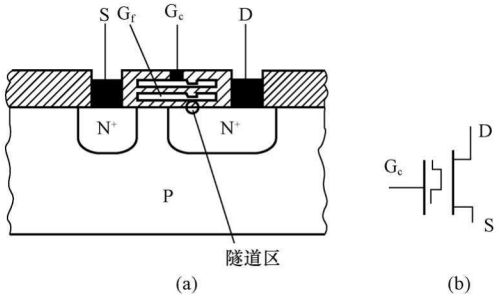


图 7－22　FLOTOX 管的横断面示意图和图形符号

（a）横断面示意图；（b）图形符号

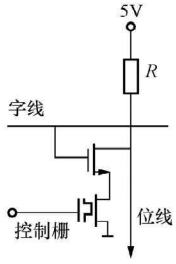


图 7－23　FLOTOX 管构成的基本耦合单元

在写入数据时，FLOTOX 管的控制栅极接高电压。在擦除数据时，FLOTOX 管的控制栅极接地。

在正常工作状态读出数据时，FLOTOX 管的控制栅极接＋5V 的电压。选好需要读出数据的字单元的地址，经地址译码器译码后，这些字单元所对应的字线为＋5V 的高电平，使普通 N 沟道增强型 MOS 管导通，其源极与漏极连接在一起。芯片出厂时所有的 FLOTOX 管的浮栅都没有存储负电荷，在控制栅极＋5V 电压的作用下，FLOTOX 管就可以导通，使普通 N 沟道增强型 MOS 管的源极接地，所以位线就被强制为低电平，所以所有的基本耦合单元中都存储了 0。

若某些基本耦合单元的 FLOTOX 管的浮栅被注入了负电荷，在控制栅极＋5V 电压的作用下，FLOTOX 管不能导通。普通 N 沟道增强型 MOS 管在字线为高电平也不能导通，所以位线经上拉电阻被强制为高电平，这时的基本耦合单元中就存储了 1。

3. FLOTOX 管构成的二进制耦合单元写入数据和擦除数据

（1）写入数据。如图 7-24 所示，字线接 15V 电压，位线接 0V 电压，控制栅极接 15V 电压。这样可以使浮栅获得负电荷。字线接 15V 电压可以使普通 NMOS 管导通，普通 NMOS 管的漏极电位和源极电位相等，为 0V。FLOTOX 管的漏极电位也为 0V。当漏极接地，控制栅极上加上足够高的电压，在隧道区将产生电场，当电场强度大到一定程度时，漏区和浮栅之间出现导电隧道，电子通过绝缘层到达浮栅，使浮栅带负电荷，实现数据的写入。

（2）擦除数据。如图 7-25 所示，字线接 15V 电压，位线接 15V 电压，控制栅极接 0V 电压。字线接 15V 电压，可以使普通 NMOS 管导通，普通 NMOS 管的漏极电位和源极电位相等，为 15V。FLOTOX 管的漏极电位也为 15V，控制栅极接地，则产生与编程相反的过程，浮栅上的负电荷消失，使数据被擦除。

（3）保留 FLOTOX 管浮栅电荷。若不想擦除浮栅电荷，则可如图 7-26 所示，使字线接 0V 电压，位线接 15V 电压，控制栅极接 0V 电压。FLOTOX 管浮栅电荷可以被保留。

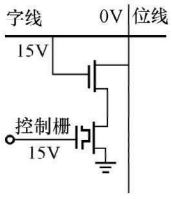


图 7-24　浮栅充电电路

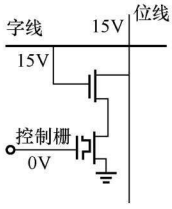


图 7-25　浮栅放电电路

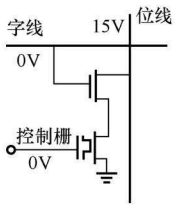


图 7-26　保留浮栅电荷的电路

EEPROM 芯片的擦除方式共有三种：字节为擦除单位，页为擦除单位，块为擦除单位。有些芯片支持三种擦除方式，有些芯片支持其中的一种或两种擦除方式。芯片的擦除方式对芯片的寿命有一定的影响。当一次只改变一个字节的内容时使用字节为擦除单位的方式。

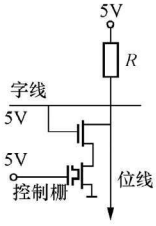


图 7-27　读出数据的电路

4. 正常工作时读出 FLOTOX 二进制耦合单元的内容的方法

当 EEPROM 芯片处在正常的数据读出状态而非写入和擦除状态时，控制栅上一直加＋5V 的电压，如图 7-27 所示。地址译码器输出的字线为高电平有效。

当地址译码器输出的字线为高电平时，若浮栅上有负电荷，则

FLOTOX管不能导通，位线输出高电平；若浮栅上没有负电荷，则FLOTOX管能导通，位线输出低电平。

（五）快闪存储器（FLASH MEMORY）的基本耦合单元

1. 快闪存储器的基本耦合单元中的MOS管

图7-28是快闪存储器的基本耦合单元中MOS管的横断面示意图和图形符号。从上面对EEPROM的介绍中可以看到，EEPROM的基本耦合单元用了两只MOS管，这无疑将限制EEPROM的集成度的进一步提高。而快闪存储器则使用了类似于UVEPROM的单SIMOS管基本耦合单元。但是有两点不同，一是快闪存储器的基本耦合单元中的MOS管的源极N^+区面积大于漏极N^+区，而SIMOS管的源极N^+区和漏极N^+区是对称的；二是快闪存储器的基本耦合单元中的MOS管从浮栅到P型衬底的绝缘层更薄。

快闪存储器的数据写入方式与UVEPROM中的SIMOS相同，即利用雪崩注入的方法使浮栅充电。

快闪存储器的擦除数据操作是利用隧道效应进行的，类似于EEPROM中的擦除操作。在擦除状态下，令控制栅处于低电平，同时在源极加入幅度为12V左右、宽度为100ms的正脉冲，在浮栅和源区间极小的重叠部分产生隧道效应，使浮栅上的电荷经隧道释放。

由于快闪存储器的所有基本耦合单元中的MOS管的源极是连在一起的，所以不能像EEPROM那样按字节擦除，而是类似UVEPROM那样整片擦除。

2. 快闪存储器的基本耦合单元

图7-29是快闪存储器的基本耦合单元。若浮栅上没有存储电荷，当字线为高电平时，MOS管导通，位线输出低电平；当字线为低电平时，MOS管截止，位线输出高电平。若浮栅上有存储电荷，相当于基本耦合单元中的字线和位线间没有MOS管，所以不管字线的电平如何，位线永远输出高电平。

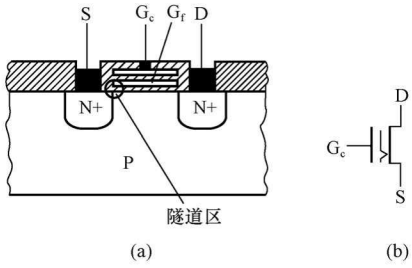


图7-28　快闪存储器的基本耦合单元中MOS管的横断面示意图和图形符号

（a）横断面示意图；（b）图形符号

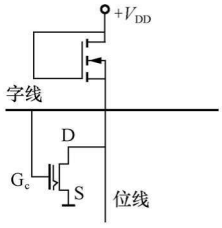


图7-29　快闪存储器的基本耦合单元

三、常用集成ROM存储器芯片

（一）常用EPROM集成存储器芯片

工业应用中标准的EPROM芯片是27×××系列，其中×××表明存储器的存储容量（单位为KB）。字长一般为8位，16位和32位字长的EPROM也有，但是很少。

早期的芯片有 2708、2716、2732、2764、27128、27256 型和 27512 型。除此之外，还有 27010、27020、27040 型，其存储容量分别为 1、2、4MB，存取时间在 200ns～500ns 之间。

1. Intel 公司的 2716 型 EPROM 芯片

2716 型 EPROM 芯片采用 NMOS 工艺制造，其双列直插式引脚排列图如图 7-30 所示。存储容量为 2K×8 位。其引脚功能说明如下。

A_0～A_{10}：11 条地址输入线，其中 7 条用于行译码，4 条用于列译码。

O_0～O_7：8 位数据线。编程写入时是输入线，正常读出时是输出线。

$\overline{CS}$：片选信号，低电平有效。当$\overline{CS}$=0 时，允许 2716 读出。

PD/PGM：待机/编程控制信号，输入。

V_{PP}：编程电源。在编程写入时，V_{PP}=+25V；正常读出时，V_{PP}=+5V。

V_{CC}：工作电源，为+5V。

2. 2764 型 EPROM 芯片

2764 型 EPROM 芯片采用 CMOS 工艺制造，其双列直插式引脚如图 7-31 所示。存储容量为 8K×8 位。CMOS 工艺的 EPROM 芯片编程电压低，存取时间在 45ns 左右。

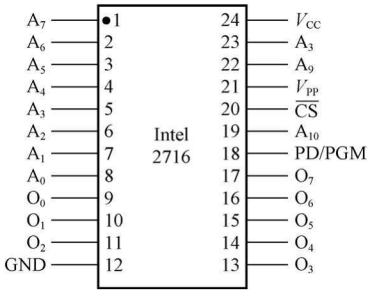


图 7-30　2716 型 EPROM 芯片引脚图

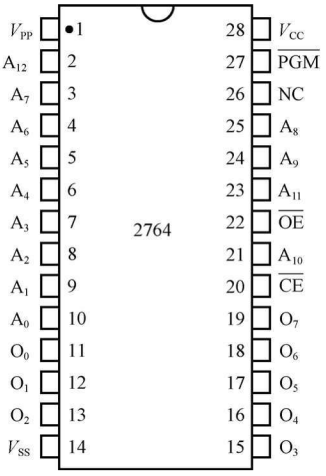


图 7-31　2764 型 EPROM 芯片引脚图

(1) 引脚功能介绍：

A_0～A_{12}：13 条地址输入线。

O_0～O_7：8 位数据线。编程写入时是输入线，正常读出时是输出线。

$\overline{CE}$：片选信号，低电平有效。当$\overline{CE}$=0 时，允许 2764 读出。

$\overline{PGM}$：编程使能信号，低电平有效。

V_{PP}：编程电源。在编程写入时，V_{PP}=+12V；正常读出时，V_{PP}=+5V。

V_{CC}：工作电源，为+5V。

V_{SS}：地。

NC：没有内部连接。

$\overline{OE}$：输出使能端，用来决定是否将 ROM 的输出送到数据总线上去。当$\overline{OE}=0$ 时，输出被使能；当$\overline{OE}=1$ 时，输出被禁止，ROM 数据输出端为高阻态。

（2）芯片编程。在进行编程时，$\overline{PGM}$引脚接低电平，V_{PP}引脚接高电平（编程电平+12V），数据由数据总线输入。

（3）芯片处于正常工作状态。在正常使用时，$V_{CC}=+5V$，V_{PP}引脚接+5V，$\overline{PGM}$引脚接高电平，数据由数据总线输出。ROM 输出能否被使能决定于$\overline{CS}+\overline{OE}$的结果。当$\overline{CS}+\overline{OE}=0$ 时，ROM 输出使能，否则将被禁止，输出端为高阻态。另外，当$\overline{CS}=1$ 时，还会停止对 ROM 内部的译码器等电路供电，其功耗降低到 ROM 工作时的 10%以下，这样会使整个系统中 ROM 芯片的总功耗大大降低。

（二）常用并行 EEPROM 芯片

常用的并行 EEPROM 芯片有 NMOS 工艺和 CMOS 工艺两种，前者为 28XXX 系列，后者为 28CXXX 系列。28XXX 系列常用的型号有 2816，2832，2864，28128 型和 28256 型等。28CXXX 系列常用的型号有 28C16，28C32，28C64，28C128 型和 28C256 型等。28C64 型是与 27C64 型 EPROM 芯片完全兼容的 EEPROM 芯片。

（三）常用的 FLASH 芯片

从 20 世纪 90 年代开始，FLASH 芯片在非易失性存储领域占据了越来越大的市场份额，它的主导地位一直维持到今天。FLASH 芯片的擦除次数一般不能超过 1 百万次，否则不能保证数据的正确性。使用 FLASH 芯片的产品通常都包含有一个管理算法来保证在产品报废之前，产品中的 FLASH 芯片的擦除次数不会超过上限。

从存储单元的电路结构看，FLASH 芯片分 NOR FLASH 和 NAND FLASH 两类。NOR FLASH 是随机存取结构，读出数据的过程类似于 EPROM。NOR FLASH 存储器可以直接和 CPU 接口，适合于存储计算机系统的引导程序和其他计算机程序。NAND FLASH 是顺序存取结构，将存储矩阵分成若干页，一般一页为 256 字节或 512 字节。NAND FLASH 没有类似 NOR FLASH 那样的随机存取接口，但是密度比较大，适用于大容量存储的场合。如果程序存储在 NAND FLASH 中，因为 NAND FLASH 与 CPU 的读写时序不匹配，所以在执行之前要将程序导入 RAM 中然后才能执行。因此 NAND FLASH 更多用于消费类电子产品中用做存储卡。

K9F2808U0C 是韩国三星公司的 NAND FLASH 芯片，存储容量为 16M×8B。目前最大容量的 NAND FLASH 的存储容量可达 8GB。NAND FLASH 芯片的存储容量大，价格低，在存储市场中占据的份额越来越大。

四、ROM 的应用

ROM 在数字系统中除了可当作存储元件外，还可以用来实现多输出的组合逻辑函数。在第三章介绍了用小规模集成电路（SSI）和中规模集成电路（MSI）来实现任意多输出组合逻辑函数的方法。此处介绍用 ROM 实现任意多输出组合逻辑函数。

任何组合逻辑函数都可以写成最小项之和的标准形式，即标准与或式（最小项表达式）。除了从存储器的角度来观察 ROM 外，还可以将 ROM 看成与门阵列和或门阵列，地址译码器可以看作与门阵列，可以产生变量的全部最小项；存储矩阵可以看作或门阵列，完成有关最小项的或运算。因此，只要合理地对 ROM 中的或阵列进行编程，ROM 的这种结构可以实现任意 n 个输入变量的 m 个输出组合逻辑函数。

【例 7-1】 有下列五个输出组合逻辑函数：

$$\begin{cases} X = \overline{A}\,\overline{B}C + \overline{A}B\,\overline{C} + A\,\overline{B}\,\overline{C} + ABCD \\ Y = BC + CA \\ M = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}BC\,\overline{D} + A\,\overline{B}\,\overline{C}D + AB\,\overline{C}\,\overline{D} \\ N = ABC + ABD + ACD + BCD \\ Q(A,B,C,D) = \sum_m(0,1,2,3) \end{cases}$$

试用 ROM 实现这五个组合逻辑函数，要求画出 ROM 阵列图。

解 五个函数均为四变量的逻辑函数，写出五个函数的标准与或表达式（最小项表达式）。按 A、B、C、D 顺序排列变量，A 为高位将变量取值组合看成二进制数，对最小项进行编号，可得

$$\begin{cases} X(A,B,C,D) = \sum_m(2,3,4,5,8,9,15) \\ Y(A,B,C,D) = \sum_m(6,7,10,11,14,15) \\ M(A,B,C,D) = \sum_m(0,3,6,9,12) \\ N(A,B,C,D) = \sum_m(7,11,13,14,15) \\ Q(A,B,C,D) = \sum_m(0,1,2,3) \end{cases}$$

因为有四个输入信号，所以字线有 2^4 根。因为有五个输出信号，所以位线有五根。选用 $2^4 \times 5$ 位的 ROM，画出 ROM 阵列图如图 7-32 所示。

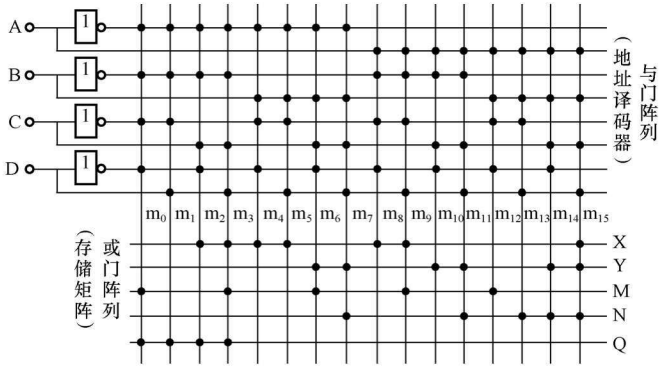


图 7-32 ［例 7-1］的 ROM 阵列图

【例 7-2】 用 ROM 实现 BCD 码转换成余 3 码的组合逻辑电路，要求画出 ROM 阵列图。

解 约定：输入是四位自然二进制码 A、B、C、D，输出是四位余 3 码 $Y_3 \sim Y_0$。

四位二进制码转换为余 3 码的真值表如表 7-1 所示，由此可写出输出函数的最小项表达式。因为没有必要化简，所以无关项没有用处。

写出四个输出函数的最小项表达式为

$$\begin{cases} Y_3(A,B,C,D) = \sum_m(5,6,7,8,9) \\ Y_2(A,B,C,D) = \sum_m(1,2,3,4,9) \\ Y_1(A,B,C,D) = \sum_m(0,3,4,7,8) \\ Y_0(A,B,C,D) = \sum_m(0,2,4,6,8) \end{cases}$$

表 7-1　　**[例 7-2] 的 真 值 表**

输入信号				输出信号				输入信号				输出信号			
A	B	C	D	Y_3	Y_2	Y_1	Y_0	A	B	C	D	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1
0	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0
0	0	1	0	0	1	0	1	1	0	1	0	×	×	×	×
0	0	1	1	0	1	1	0	1	0	1	1	×	×	×	×
0	1	0	0	0	1	1	1	1	1	0	0	×	×	×	×
0	1	0	1	1	0	0	0	1	1	0	1	×	×	×	×
0	1	1	0	1	0	0	1	1	1	1	0	×	×	×	×
0	1	1	1	1	0	1	0	1	1	1	1	×	×	×	×

画出实现该函数的 ROM 阵列图，如图 7-33 所示。

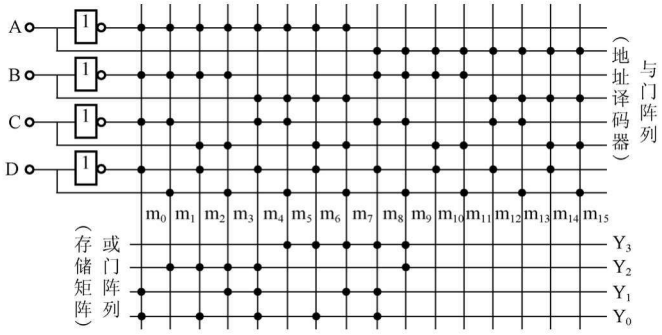


图 7-33　[例 7-2] 的 ROM 阵列图

【例 7-3】 用 ROM 实现三位二进数的平方运算，要求画出 ROM 阵列图。

解　数学运算是数控装置和数字系统中需要经常进行的操作，如果事先把要用到的基本函数变量在一定范围内的取值和相应的函数取值列成表格，写入只读存储器中，则在需要时只要给出规定地址就可以快速地得到相应的函数值。这种 ROM 实际上已经成为函数运算表电路。

约定三位二进数分别用 A、B、C 表示，六位乘积分别用 D_5、D_4、D_3、D_2、D_1、D_0 表示。根据题意列出真值表如表 7-2 所示。

表 7-2　　**[例 7-3] 的真值表**

输入			输出					
A	B	C	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

写出六个输出函数的最小项表达式为

$$\begin{cases} D_5(A,B,C)=\sum_m(6,7) \\ D_4(A,B,C)=\sum_m(4,5,7) \\ D_3(A,B,C)=\sum_m(3,5) \\ D_2(A,B,C)=\sum_m(2,6) \\ D_1(A,B,C)=0 \\ D_0(A,B,C)=\sum_m(1,3,5,7) \end{cases}$$

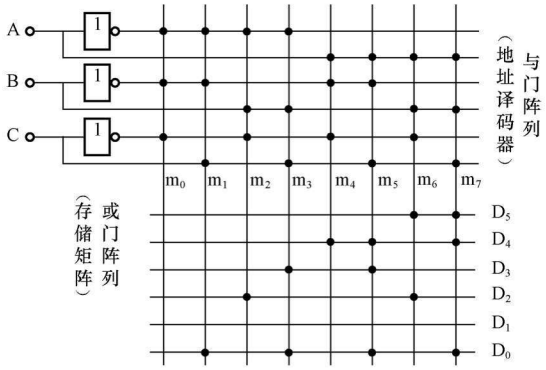


图 7-34 ［例 7-3］的 ROM 阵列图

画出实现该函数的 ROM 阵列图如图 7-34所示。

用可编程 ROM 来实现组合逻辑函数的最大不足之处在于对芯片的利用率不高。这是因为 ROM 中的与阵列是一个固定的全译码阵列，每一个乘积项都是一个最小项，只能实现组合逻辑函数的最小项表达式，不能进行化简，而且实际上大多数的组合逻辑函数也并不需要所有的最小项。因此，ROM 在绝大多数场合还是被作为存储器使用。

【例 7-4】 二进制加计数器的状态转移图如图 7-35 所示。利用 ROM 构成的任意波形发生器如图 7-36 所示，改变 ROM 的内容，即可改变输出波形。当 ROM 的内容如表 7-3 所示时，画出输出端电压随 CP 脉冲变化的波形。

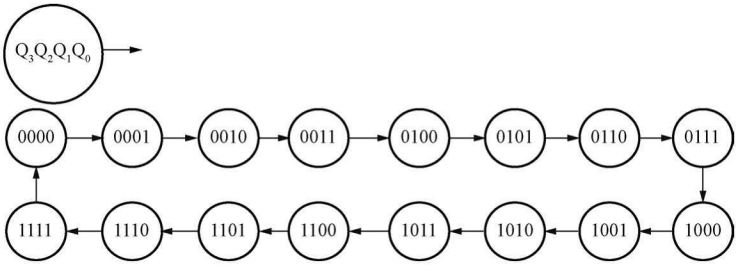


图 7-35 二进制加计数器的状态转移图

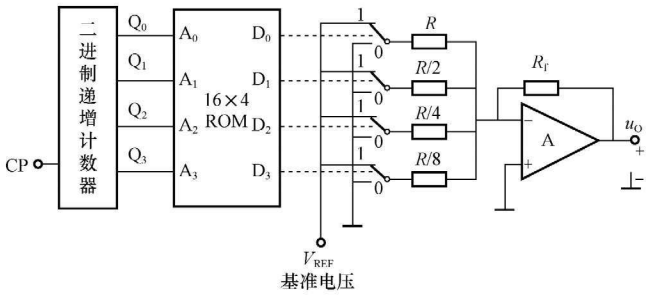


图 7-36 用 ROM 构成的任意波形发生器

表 7-3　　ROM 的内容列表

地址输入信号				ROM 的内容				地址输入信号				ROM 的内容			
A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0	A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	1	0	1	1	0	0	1	0	0	1	1
0	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1	1	0	1	1	0	0	0	1
0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0
0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	1
0	1	1	0	0	1	1	0	1	1	1	0	0	0	1	0
0	1	1	1	0	1	0	1	1	1	1	1	0	0	1	1

解　由图 7-36 可见，计数器的输出 Q_3、Q_2、Q_1、Q_0 送入 ROM 的地址输入端 A_3、A_2、A_1、A_0，ROM 的数据输出端 D_3、D_2、D_1、D_0 又作为后级 D/A 转换器的输入。而对于 D/A 转换器来说有

$$u_O = -\frac{R_f U_{REF}}{R}(2^3 D_3 + 2^2 D_2 + 2^1 D_1 + 2^0 D_0)$$

$$= -K(2^3 D_3 + 2^2 D_2 + 2^1 D_1 + 2^0 D_0)$$

$$K = \frac{R_f U_{REF}}{R}$$

设计数器初始状态为 0000，即 $A_3A_2A_1A_0=0000$。由表 7-3 可知，对应的 $D_3D_2D_1D_0=0100$，再根据上式计算可得 $u_O/K=-4$；在第一个 CP 脉冲的作用下，计数器输出为 0001，即 $A_3A_2A_1A_0=0001$，由表 7-3 知，$D_3D_2D_1D_0=0101$，根据上式计算得 $u_O/K=-5$。依照上述方法逐个计算，即得 CP 脉冲作用下各 u_O/K 的值。据此即可画出输出端电压随 CP 脉冲变化的波形，如图 7-37 所示。

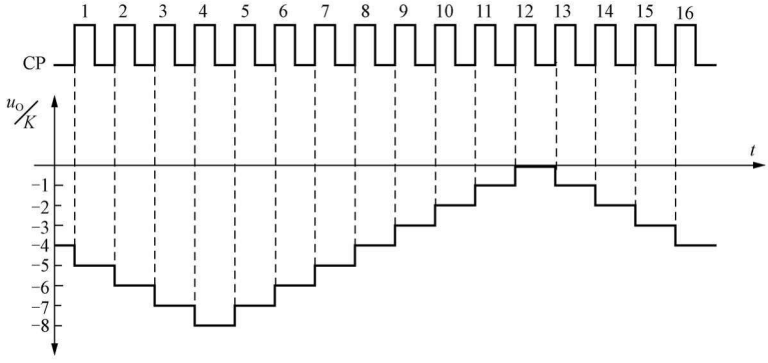


图 7-37　输出端电压随 CP 脉冲变化的波形

五、ROM 容量的扩展

当一片 ROM 芯片不能满足存储容量的要求时，可以用若干片 ROM 相连接，构成一个存储容量更大的 ROM。扩大存储容量的方法，通常有扩展字长和扩展字数两种方法。

（一）扩展字长

如果一片 ROM 芯片的字数已经够用，而每字的位数不够时，可以用扩展字长的方法来扩展存储容量。图 7-38 所示为两片 2764（8K×8 位）扩展成 8K×16 位 EPROM 的逻辑电路图。

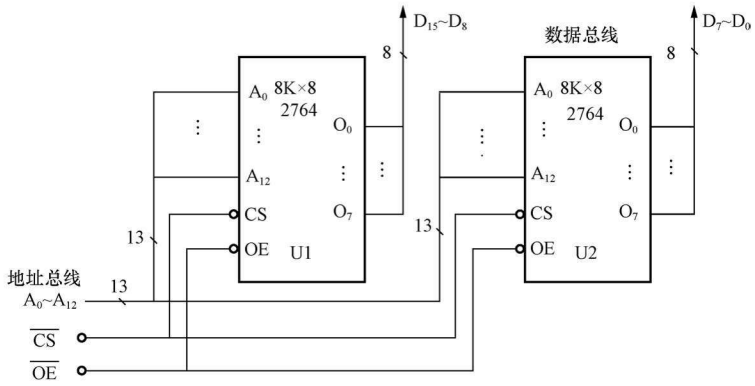


图 7-38 扩展字长的逻辑电路图

（二）扩展字数

如果一片 ROM 芯片的位数已经够用，但总的字数不够时，可以用扩展字数的方法来扩展存储容量。图 7-39 所示为用 8 片 2764（8K×8）扩展成 64K×8 位 EPROM。图中 74LS138 是地址译码器，三个控制端 G_1、G_{2A}、G_{2B}分别处于它们各自的有效电平，74LS138

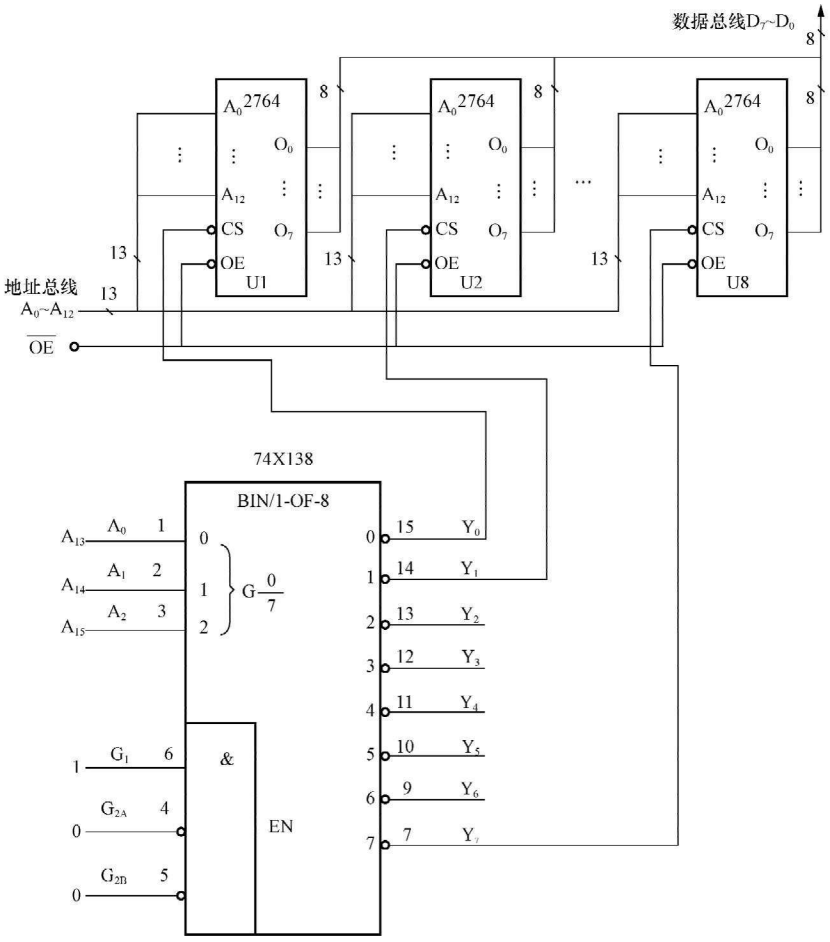


图 7-39 扩展字数的逻辑电路图

处在正常的译码状态。当地址线 $A_{15}A_{14}A_{13}$ 取值从 000 时，Y_0 输出 0，其他各输出端均为 1，这时选中 2764 的 U1 芯片，其他各芯片的输出均处于高阻状态。

如果一片 ROM 芯片的字数和位数都不够用，可以用同时扩展字数和位数的方法来扩展存储容量。

第四节　随机存取存储器(RAM)

随机存取存储器（Random Access Memory，RAM），也可以称为读写存储器。其特点是当 RAM 处于正常工作状态时可以从 RAM 中读出数据，也可以往 RAM 中写入数据。随机的含义是数据的存取时间与存储单元所处的物理位置无关。RAM 具有易失性，即一旦电路失掉电源，所存储的数据立即丢失。

从构成 RAM 的三极管来看，RAM 有双极型和 MOS 型两种，前者主要用于小容量高速存储器，后者主要用于大容量主存储器。按存储器件能否长时间保存信息来分，RAM 有静态 RAM（Static RAM，SRAM）和动态 RAM（Dynamic RAM，DRAM）两种。前者用双稳态触发器来保存信息，只要不掉电，信息就不会丢失；后者利用 MOS 管的电容来存储信息，使用时需定时给电容充电才能使信息得以保存。SRAM 的集成度低，功耗较大；DRAM 的集成度高，功耗小，主要用做大容量主存储器。

一、RAM 的基本结构

RAM 的基本结构与 ROM 非常相似。除了存储矩阵、地址译码器以外，因为 RAM 不但可以读出数据，而且在正常工作环境下还可以写入数据，所以还有读写控制器电路。

RAM 电路中字长、字数、存储容量、字线、位线等概念与 ROM 电路完全相同。二者最大的区别是二进制存储单元的结构不同。在 ROM 中，二进制存储单元更准确地被称为二进制基本耦合单元，ROM 是一个组合逻辑电路。而在 RAM 中，二进制存储单元是一个触发器或者是一个 MOS 管的电容，RAM 是时序逻辑电路。图 7-40 是具有一个地址译码器的 RAM 结构示意框图。图 7-41 是具有 X 行地址译码器和 Y 列地址译码器的两个地址译码器的 RAM 结构示意框图。

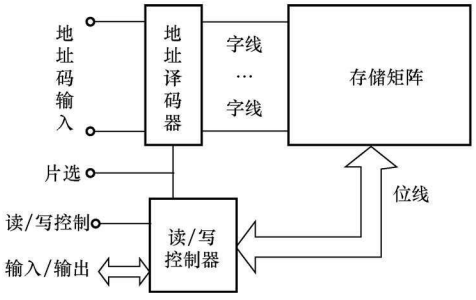


图 7-40　具有一个地址译码器的 RAM 的结构示意框图

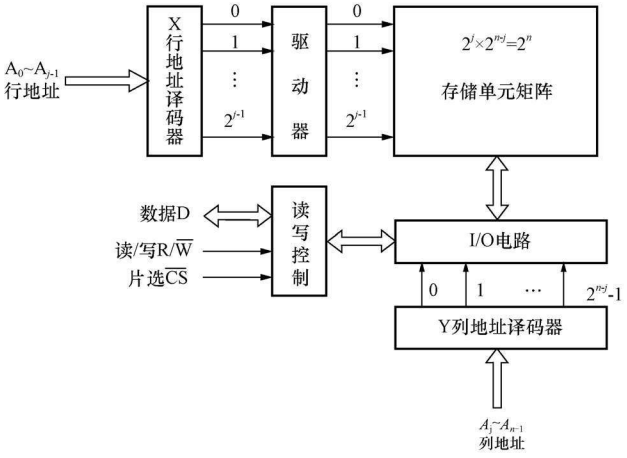


图 7-41　具有两个地址译码器的 RAM 的结构示意框图

（一）读/写控制电路

RAM 通过输入/输出端 I/O 与计算机的中央处理单元（CPU）交换数据。输入/输出端被分时使用，读出时它是输出端，写入时它是输入端，即一线二用，由读/写控制线控制。输入/输出端数据线的条数，与一个地址中所对应的二进制存储单元的个数（字长）相同。例如在 1024×1 位的 RAM 中，每个地址对应的字中只有 1 个二进制存储单元，因此只有 1 条输入/输出数据线；而在 256×4 位的 RAM 中，每个地址对应的字中有 4 个二进制存储单元，所以有 4 条输入/输出数据线。也有的 RAM 芯片输入数据线和输出数据线是分开的。RAM 的数据输出端一般都具有集电极开路或三态输出结构。图 7-42 给出了一个简单的读/写控制电路。

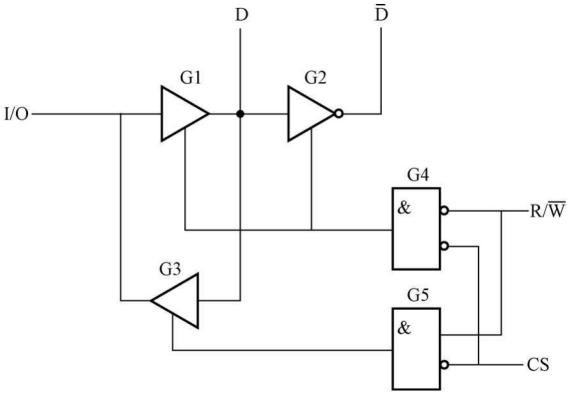


图 7-42 读/写控制电路

由于受 RAM 的集成度限制，一台计算机的存储器系统往往是由许多片 RAM 组合而成。CPU 访问存储器时，一次只能访问 RAM 中的某一片（或几片），即存储器中只有一片（或几片）RAM 中的一个地址接受 CPU 访问，与其交换信息，而其他片 RAM 与 CPU 不发生联系，片选就是用来实现这种控制的。通常一片 RAM 有一根或几根片选线，当某一片的片选线接入有效电平时，该片被选中，地址译码器的输出信号控制该片某个地址中的字的位线与 CPU 接通；当片选线接入无效电平时，则该片与 CPU 之间处于断开状态。

访问 RAM 时，对被选中的字，究竟是读还是写，通过读/写控制线 R/$\overline{W}$进行控制。如果是读，则被选中的字存储的数据经数据线 D、输入/输出线传送给 CPU；如果是写，则 CPU 将数据经过输入/输出线、数据线 D 存入被选中的字。

在图 7-42 所示读/写控制电路中，RAM 的读/写控制线 R/$\overline{W}$高电平为读，低电平为写，用一根线来控制读和写。有的 RAM 芯片中的读/写控制线是分开的，一根为读，另一根为写。

在图 7-42 中，当选片信号 CS=1 时，G5、G4 输出为 0，三态门 G1、G2、G3 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。当 CS=0 时，芯片被选通，这时如果 R/$\overline{W}$=1 时，G5 输出高电平，G3 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；如果 R/$\overline{W}$=0 时，G4 输出高电平，G1、G2 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 D 和 $\overline{D}$ 上，并被存入到所选中的存储单元，存储器执行写操作。

（二）RAM 的工作时序

为保证存储器准确无误地工作，加到存储器上的地址、数据和控制信号必须遵守几个时间边界条件。

1. RAM 的读操作时序

图 7-43 给出了某种具有单个地址译码器的 RAM 读出过程的时序关系。读出操作过程如下：

（1）读出单元的地址加到存储器的地址输入端；

（2）送入有效的片选信号 CS，此处为低电平有效；

（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，在图 7-43 中 R/$\overline{W}$为高电平，因为平时 R/$\overline{W}$总是处在高电平状态，所以此处忽略不画；

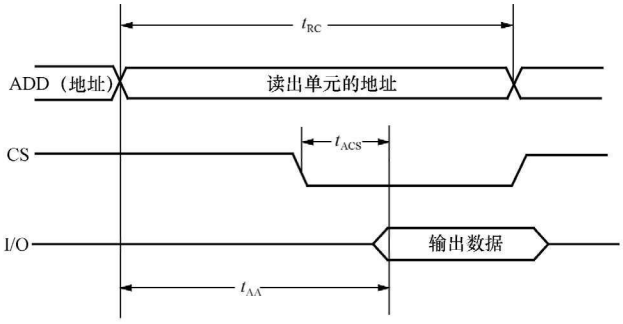


图 7-43　RAM 读操作时序图

（4）让片选信号 CS 无效，I/O 端呈高阻态，本次读出过程结束。

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}。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，被读存储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，表示该芯片连续进行两次读操作必需的时间间隔。

2. RAM 的写操作时序

图 7-44 给出了某种具有单个地址译码器的 RAM 写数据过程的时序关系。写数据的操作过程如下：

（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 端加上有效电平（此处为低电平有效），使 RAM 选通；

（3）将待写入的数据加到数据输入端 I/O；

（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。

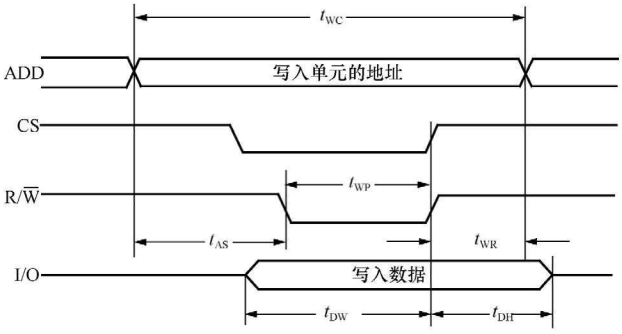


图 7-44　RAM 写操作时序图

由于地址改变时，新地址的稳定需要经过一段时间，如果在这段时间内加入写控制信号（即 R/$\overline{W}$变低），就可能将数据错误地写入其他单元。为防止这种情况出现，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，地址信号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该芯片的写脉冲宽度 t_{WP}。此外，对于写入的数据，应在写信号 t_{DW}时间内保持稳定，且在写信号失效后继续保持 t_{DH}时间。在时序图中还给出了写周期 t_{WC}，它反应了连续进行两次写操作所需要的最小时间间隔。对大多数静态半导体存储器来说，读周期和写周期是相等的，一般为十几纳秒到几十纳秒。

二、RAM 的二进制存储单元

二进制存储单元是存储器（RAM）的核心部分。二进制存储单元的电路形式多种多样，按工作方式不同可分为静态和动态两类，按所用元件类型又可分为双极型和 MOS 型两种。

（一）四管动态 MOS 二进制存储单元

动态 MOS 存储单元存储信息的原理，是利用 MOS 管栅极电容具有暂时存储信息的作用。由于漏电流的存在，栅极电容上存储的电荷不可能长久保持不变，因此为了及时补充漏掉的电荷，避免存储信息丢失，需要定时地给栅极电容补充电荷，通常把这种操作称作刷新或再生。

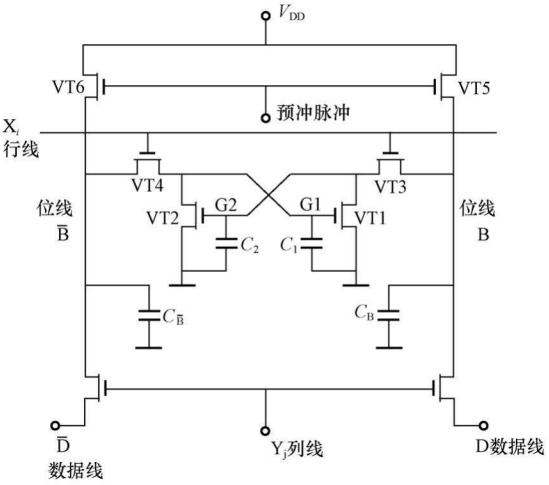


图 7-45　四管动态 MOS 存储单元

图 7-45 所示是四管动态 MOS 存储单元电路，对应有行和列两个地址译码器的 RAM 电路。VT1 和 VT_2 交叉连接，信息（电荷）存储在 C_1、C_2 上。C_1、C_2 上的电压控制 VT1、VT2 的导通或截止。当 C_1 充有电荷（电压大于 VT1 的开启电压），C_2 没有电荷（电压小于 VT2 的开启电压）时，VT1 导通、VT2 截止，称此时存储单元为 0 状态；当 C_2 充有电荷，C_1 没有电荷时，VT2 导通、VT1 截止，则称此时存储单元为 1 状态。VT3 和 VT4 是门控管，控制存储单元与位线的连接。

VT5 和 VT6 组成对位线的预充电电路，并且为一列中所有存储单元所共用。在访问存储器开始时，VT5 和 VT6 栅极上加“预充”脉冲，VT5、VT6 导通，位线 B 和$\overline{B}$被接到电源 V_{DD}而变为高电平。当预充脉冲消失后，VT5、VT6 截止，位线与电源 V_{DD}断开，但由于位线上分布电容 C_B 和 $C_{\overline{B}}$的作用，可使位线上的高电平保持一段时间。

在位线保持为高电平期间，当进行读操作时，X_i行线变为高电平，VT3 和 VT4 导通，若存储单元原来为 0 态，即 VT1 导通（G2 点为低电平）、VT2 截止（G1 点为高电平），此时 C_B 通过导通的 VT3 和 VT1 放电，使位线 B 变为低电平。而由于 VT2 截止，虽然此时 VT4 导通，位线$\overline{B}$仍保持为高电平，这样就把存储单元的状态读到位线 B 和$\overline{B}$上。如果此时列线 Y_j 亦为高电平，则 B、$\overline{B}$的信号将通过数据线被送至 RAM 的输出端 D 和 $\overline{D}$。

位线的预充电电路起什么作用呢？在 VT3、VT4 导通期间，如果位线没有事先进行预充电，那么位线$\overline{B}$的高电平只能靠 C_1 通过 VT4 对 $C_{\overline{B}}$充电建立，这样 C_1 上将要损失掉一部分电荷。由于位线上连接的元件较多，$C_{\overline{B}}$甚至比 C_1 还要大，这就有可能在读一次后便破坏了 G1 的高电平，使存储的信息丢失。采用了预充电电路后，由于位线$\overline{B}$的电位比 G1 的电位还要高一些，所以在读出时，C_1 上的电荷不但不会损失，反而还会通过 VT4 对 C_1 再充电，使 C_1 上的电荷得到补充，即进行一次刷新。

当进行写操作时，RAM 的数据输入端通过数据线、位线控制存储单元改变状态，把信息存入其中。

（二）单管动态 MOS 二进制存储单元

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 7-46所示。电容 C_S 的状态决定了这个 DRAM 单位的逻辑状态是 1 还是 0。一个电容器可以存储一定量的电子，一个被充电的电容器如果认为是逻辑上的 1，而“空”的电容器则可以认为是逻辑上的 0。

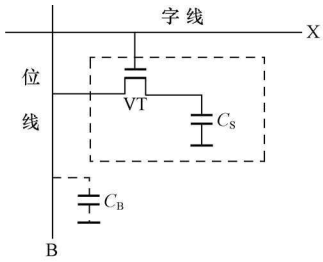


图 7-46　单管动态 MOS 存储单元

但是电容器被利用的这个特性也是它的缺点，因为电容器不能长久地保持存储的电子，所以内存需要不断定时刷新，才能保持暂存的数据。电容器可以由电流来充电（当然这个电流是有一定限制的，否则会把电容击穿）；同时电容的充放电需要一定的时间，虽然对于内存基本单元中的电容来说这个时间很短，只有大约 0.2～0.18 μs，但是这个期间内存是不能执行存取操作的。DRAM 制造商的一些资料显示，内存至少要每 64ms 刷新一次，这也意味着内存有 1%的时间要用来刷新。内存的自动刷新对于内存厂商来说不是一个难题，而关键在于，当对内存单元进行读取操作时要保持内存的内容不变。所以 DRAM 单元每次读取操作之后都要进行刷新，也就是执行一次“回写”操作，因为读写操作会破坏内存中的电荷。因此，内存不但要每 64ms 刷新一次，而且每次读操作之后还要刷新一次，这样就增加了存取操作的周期。

DRAM 的存取速度没有 SRAM 快，但是 DRAM 更容易做成大容量的 RAM。一般的嵌入式系统中，主存储器采用 DRAM，而快速存储器（Cache Memory）则采用 SRAM。DRAM 的优点是制造成本要比 SRAM 低得多。

同步动态 RAM（SDRAM）的存储单元的基本原理同 DRAM 基本一样，但是这些存储单元的组织和控制与 DRAM 就具有相当大的区别了。

SDRAM 是多 bank 结构，例如在一个具有两个 bank 的 SDRAM 的模组中，其中一个 bank 在进行预充电期间，另一个 bank 却马上可以被读取，这样当进行一次读取后，又马上去读取已经预充电 bank 的数据时，就无需等待而是可以直接读取了，这也就大大提高了存储器的访问速度。为了实现这个功能，SDRAM 需要增加对多个 bank 的管理，实现控制其中的 bank 进行预充电。在一个具有两个以上 bank 的 SDRAM 中，一般会多一根叫做 BA_n 的引脚，用来实现在多个 bank 的选择。

三、常用 RAM 芯片

表 7-4 列出了几种常见的 RAM 芯片。

表 7-4　　常用 RAM 芯片列表

SRAM		DRAM		SRAM		DRAM	
型号	存储容量（字×位）	型号	存储容量（字×位）	型号	存储容量（字×位）	型号	存储容量（字×位）
MB2114	1K×4	MB814101	4K×1	HM62256	32K×8	MB81464	64K×4
HM6116	2K×8	MB2118	16K×1	HM628128	128K×8	MB81C4256	256K×4
HM6264	8K×8	MB81416	16K×4				

图 7-47 所示是 2K×8 位静态 CMOS RAM HM6116 的引脚排列图。图中，$A_0 \sim A_{10}$ 是地址码输入端，$D_0 \sim D_7$ 是数据输出端，$\overline{CS}$是选片端，$\overline{OE}$是输出使能端，$\overline{WE}$是写入控制端。

表 7-5 所列是 HM6116 的工作方式与控制信号之间的关系，读出和写入线是分开的，而且写入优先。

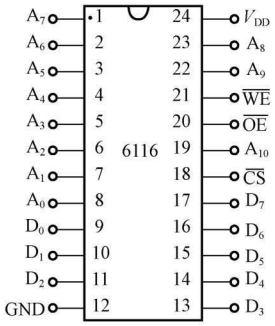


图 7-47　HM6116 静态 RAM 引脚排列图

表 7-5　　HM6116 静态 RAM 工作方式与控制信号之间的关系

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	$A_0 \sim A_{10}$	$D_0 \sim D_7$	工作状态
1	×	×	×	高阻态	低功耗维持
0	0	1	稳定	输出	读
0	×	0	稳定	输入	写

四、RAM 容量的扩展

在实际应用中，经常需要大容量的 RAM。在单片 RAM 芯片容量不能满足要求时，就需要进行扩展，扩展的方法与 ROM 芯片十分类似。

（一）扩展字长

如果一片 ROM 芯片的字数已经够用，而每字的位数不够时，可以用扩展字长的方法来扩展存储容量。如图 7-48 所示，用 8 片 1024（1K）×1 位的 RAM 构成的 1024×8 位的 RAM。

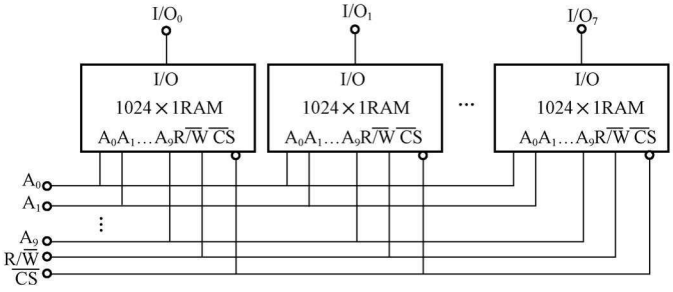


图 7-48　1K×1 位 RAM 扩展成 1K×8 位 RAM

（二）扩展字数

如果一片 RAM 芯片的位数已经够用，但总的字数不够时，可以用扩展字数的方法来扩展存储容量。如图 7-49 所示，用 8 片 1K×8 位 RAM 构成的 8K×8 位 RAM。图中，输入/输出线，读/写线和地址线 A_0～A_9 是并联起来的，高位地址码 A_{10}、A_{11} 和 A_{12} 经 74138 译码器 8 个输出端分别控制 8 片 1K×8 位 RAM 的片选端，以实现字数扩展。

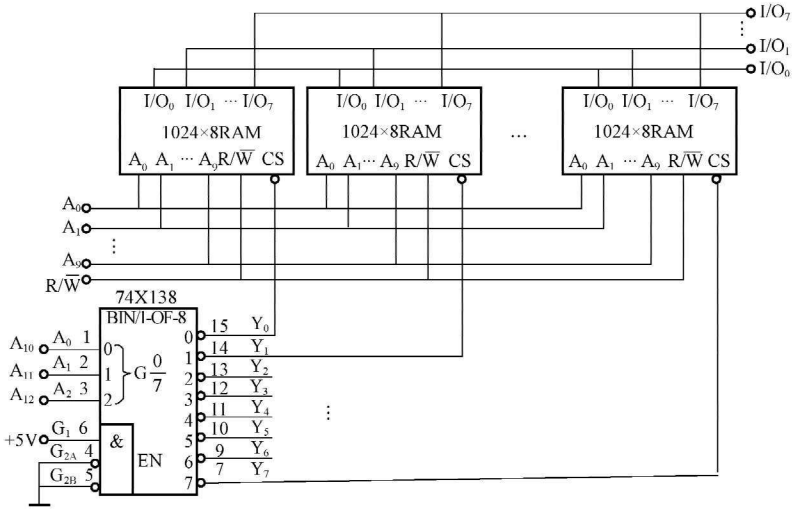


图 7-49　1K×8 位 RAM 扩展成 8K×8 位 RAM

如果一片 RAM 芯片的字数和位数都不够用，可以用同时扩展字数和位数的方法来扩展存储容量。

小　　结

半导体存储器是现代数字系统特别是计算机系统中的重要组成部件，可分为 RAM 和 ROM 两大类，绝大多数属于 MOS 工艺制成的大规模数字集成电路。

ROM 是一种非易失性的存储器，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 又可分成固定 ROM 和可编程 ROM。后者又可细分为 PROM、EPROM、EEPROM 和快闪存储器等。特别是 EEPROM 和快闪存储器可以进行电擦写，已兼有了 RAM 的特性。

从逻辑电路构成的角度看，ROM 是由与门阵列和或门阵列构成的组合逻辑电路。ROM 的输出是输入最小项的组合，因此采用 ROM 可方便地实现各种逻辑函数。随着大规模集成电路成本的不断下降，利用 ROM 构成各种组合、时序电路，愈来愈具有吸引力。

RAM 是一种时序逻辑电路，具有记忆功能，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。

习　　题

7-1　某存储器具有6条地址线和8条双向数据线，具有单个地址译码器，问：

(1) 该存储器字线有多少根？

(2) 该存储器字长是多少？

(3) 该存储器字数是多少？

(4) 该存储器存储容量有多少位？

7-2　有下列存储系统：

(1) 64K×4；

(2) 256K×8；

(3) 1M×3；

(4) 128K×8。

指出这些存储系统各具有多少个存储单元，至少需要几条地址线和数据线。

7-3　有下列存储系统：

(1) 2K×1；

(2) 32K×4；

(3) 256K×32。

设存储器的起始地址全为0，试指出这些存储系统的最高地址为多少？

7-4　分析图7-50中，$A_1A_0=00$、$A_1A_0=01$、$A_1A_0=10$、$A_1A_0=11$，当EN=0时$D_3D_2D_1D_0$的值是多少？EN=1时$D_3D_2D_1D_0$的值是多少？

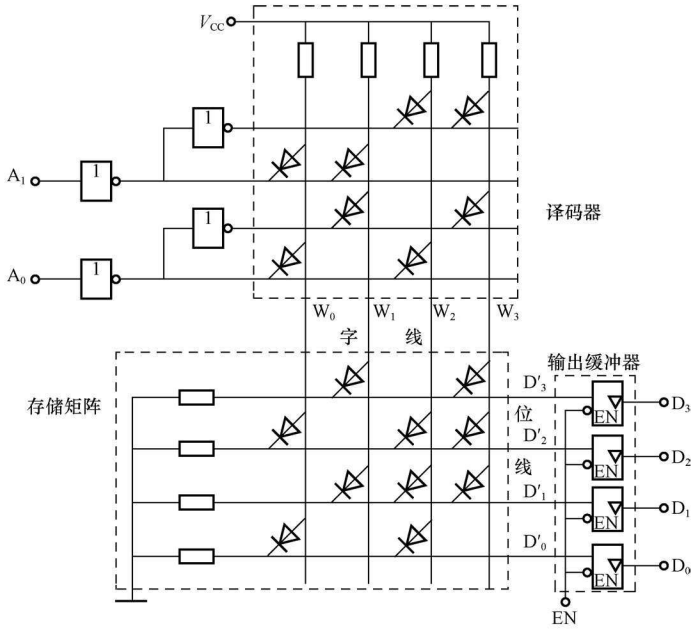


图7-50　题7-4图

7-5　图 7-51 中，MOS 管均为 N 沟道增强型 MOS 管。图中最上面第一行管子为负载管，各管栅极与漏极连接，并与 V_{DD} 连接，总是处于导通状态，可以等效为一个电阻。地址译码器输出高电平有效，试分析该 ROM 存储的信息。

7-6　分析图 7-52 所示的 ROM 阵列图，列表给出 ROM 中所存储的数据。

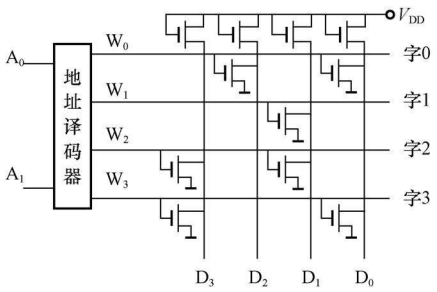


图 7-51　题 7-5 图

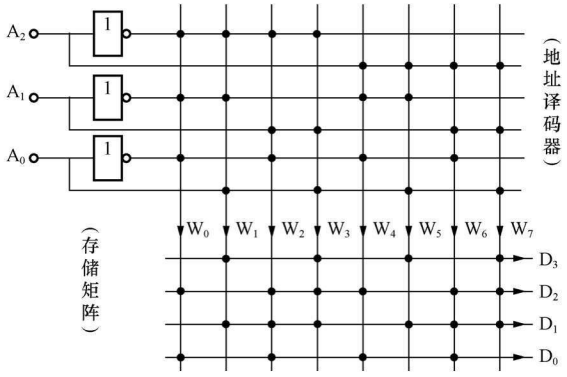


图 7-52　题 7-6 图

7-7　分析图 7-53 中，当 $A_2A_1A_0=000$ 和 $A_2A_1A_0=111$ 时，$D_3D_2D_1D_0$ 的值是多少？

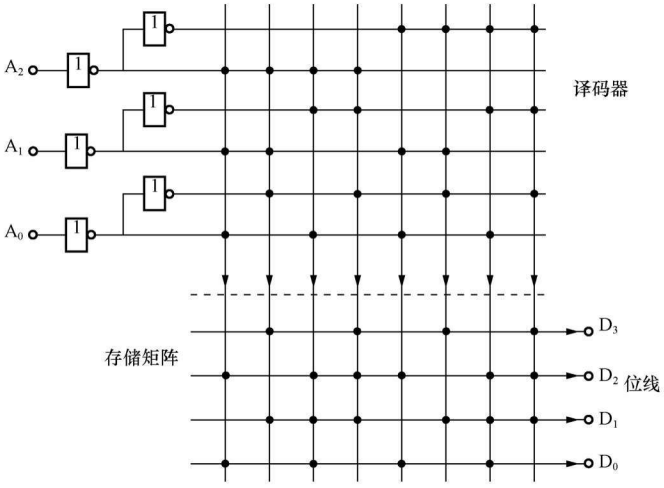


图 7-53　题 7-7 图

7-8　用 ROM 实现 4 位二进制数的平方运算，要求画出 ROM 阵列图。

7-9　由 MOS 管构成的 ROM 如图 7-54 所示。其上面第一行管子为负载管，为 N 沟道增强型 MOS 管，各管栅极与漏极连接，并与 $+V_{DD}$ 连接，总是处于导通状态，可以等效为一个电阻。在基本耦合单元中，以有或无 MOS 管表示存储信息 1 或 0。这些也是 N 沟道增强型 MOS 管。图中所示的地址线为 $A_9\sim A_0$ 共 10 根。为了减少地址译码器的输出线，将地址线分为两组：$A_5\sim A_0$ 作为行地址译码器的输入线，读出行选择线为 $X_0\sim X_{31}$，输出为高电平有效；$A_9\sim A_5$ 作为列地址译码器的输入线，读出列选择线为 $Y_0\sim Y_{31}$，输出为高电平有效，两者总和仅 64 根选择线，使译码器输出线大为减少。由图中所示可知，行选择线和列选择线的交点共为 $32\times32=1024$，每一个交点为一个基本耦合单元，在交点上有 MOS

管的表示信息 0，在交点上无 MOS 管的表示信息 1，输出数码只是 1024 中的某一位，故该存储器容量为 1K×1 位。问当地址为 $A_9 \sim A_0 = 0000100001$ 时 D_{OUT} 的值是多少？

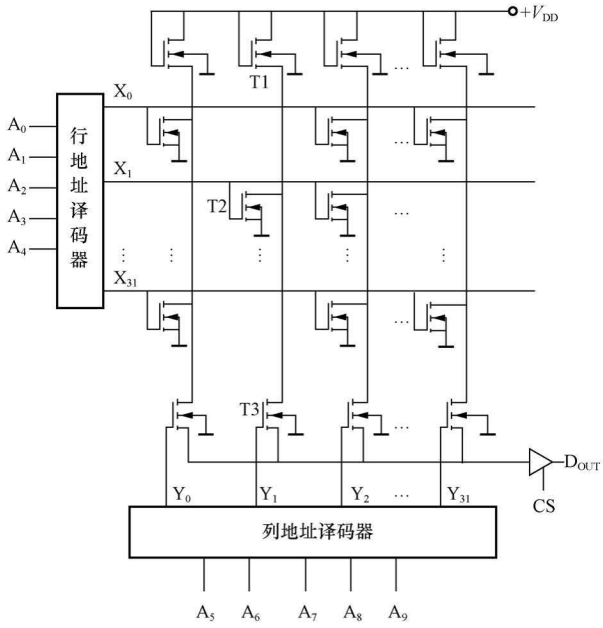


图 7-54　题 7-9 图

7-10　某存储器的信息如表 7-6 所示，具有单个地址译码器，问：

(1) 该存储器字线有多少根？

(2) 该存储器字长是多少？

(3) 该存储器字数是多少？

(4) 该存储器存储容量有多少位？

表 7-6　　题 7-10 表

存储器的地址		存储器的内容	存储器的地址		存储器的内容
二进制形式的地址	十进制形式的地址		二进制形式的地址	十进制形式的地址	
0000000000	0	01010101	1111111101	1021	00000000
0000000001	1	10101010	1111111110	1022	11111111
0000000010	2	11110000	1111111111	1023	01110001
…	…	…			

7-11　用 16K×4 位 EPROM 实现下列各逻辑函数，画出 ROM 阵列图：

(1) $Y_1 = ABC + \overline{A}(B+C)$；

(2) $Y_2 = A\overline{B} + \overline{A}B$；

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

(4) $Y_4 = ABC + \overline{ABC}$。

7-12　试用组合逻辑 ROM 实现从 4 位格雷码到自然二进制代码的转换。

7-13　用 16K×6 位 EEPROM 实现四位二进制数转换成两位 BCD 码的电路，画出 ROM 阵列图。

7-14　利用 ROM 构成的任意波形发生器［如图 7-55（a）所示］，改变 ROM 的内容，即可改变输出波形。当 ROM 的内容如表 7-7 所示时，画出输出端电压随 CP 脉冲变化的波形。二进制递减计数器的状态转移图如图 7-55（b）所示。

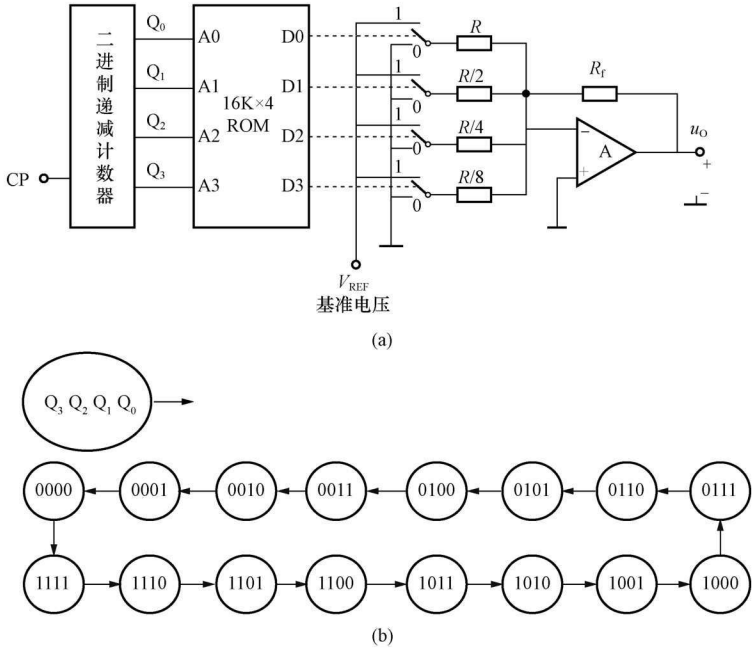


图 7-55　题 7-14 图

表 7-7　　**题 7-14 表**

地址输入信号				ROM 的内容				地址输入信号				ROM 的内容			
A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0	A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0
0	0	0	1	1	1	0	1	1	0	0	1	0	0	1	1
0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0
0	0	1	1	1	1	1	1	1	0	1	1	0	1	0	1
0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0
0	1	0	1	0	0	1	1	1	1	0	1	0	1	0	1
0	1	1	0	0	0	1	0	1	1	1	0	1	0	1	0
0	1	1	1	0	1	0	1	1	1	1	1	0	0	1	1

7-15　用 16K×8 位 EPROM 实现下列各逻辑函数，画出阵列图：

(1) $Y_1(A,B,C,D)=\sum_m(0,2,3,4,6,8,10,12)$；

(2) $Y_2(A,B,C,D)=\sum_m(0,2,4,5,6,7,8,9,14,15)$；

(3) $Y_3(A,B,C,D)=\sum_m(0,1,5,7)$；

(4) $Y_4(A,B,C,D)=\sum_m(0,2,5,7,14,15)$；

(5) $Y_5(A,B,C,D)=\sum_m(3,7,8,9,13,14)$；

(6) $Y_6=A\overline{B}+B\overline{C}+C\overline{D}+\overline{D}A$；

(7) $Y_7=\overline{B}\,\overline{D}+\overline{C}D+\overline{A}\,\overline{C}$；

(8) $Y_8=\overline{A}B+\overline{B}\,\overline{C}+AC\overline{D}$。

7-16　MCM6264 是 MOTOROLA 公司生产的 8K×8 位 SRAM，该芯片采用 28 脚塑料双列直插式封装，单电源+5V 供电。图 7-56 给出了该芯片的引脚排列图。图中 A_0～A_{12} 为地址输入，DQ_0～DQ_7 为数据输入/输出，W 为写允许，E_1、E_2 为片选，NC 为空引脚。查找 MCM6264 芯片的资料，熟悉其功能，并试用 MCM6264 SRAM 芯片设计一个 16K×16 位的存储器系统，画出其逻辑图。

7-17　一个 CMOS 存储单元如图 7-57 所示，其中的 MOS 管均为增强型，行选择线和列选择线均为高电平有效。试分析其工作原理。

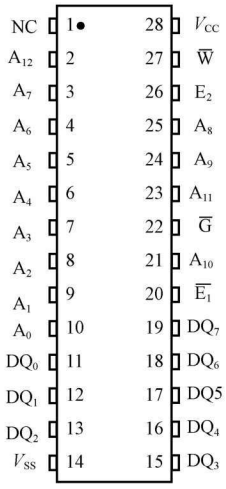


图 7-56　题 7-16 图

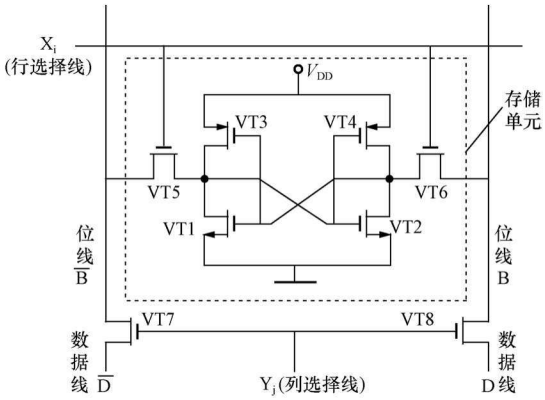


图 7-57　题 7-17 图

第八章　可编程逻辑器件与 VHDL 语言

本　章　提　要

本章介绍可编程逻辑器件的发展历史、分类和电路结构，详细介绍 Lattice 公司的 GAL16V8 器件的结构，并在此基础上简单介绍了 Xilinx 公司的 FPGA 器件和 Altera 公司的 CPLD 器件；介绍 VHDL 语言的基本语法和几个基本数字电路模块的 VHDL 描述。

第一节　概　　述

一、可编程逻辑器件的发展历史

20 世纪 70 年代，熔丝工艺编程的 PROM 和 PLA（Programmable Logic Array）是最早出现的可编程逻辑器件（Programmable Logic Device，PLD）。20 世纪 70 年代末，AMD 公司推出了 PAL（Programmable Array Logic）器件。20 世纪 80 年代初，Lattice 公司首先生产出了可电擦写的、比 PAL 器件使用更灵活的 GAL（Generic Array Logic）器件。20 世纪 80 年代中期，Xilinx 公司提出了现场可编程的概念，同时生产出了世界上第一片 FPGA（Field Programmable Gate Array）器件。同一时期，Altera 公司推出了 EPLD（Erasable Programmable Logic Device），它比 GAL 具有更高的集成度，可以用紫外线或电擦除。20 世纪 80 年代末，Lattice 公司又提出了在系统可编程 ISP（In System Programmability）的概念，并推出了一系列具有在系统可编程能力的 CPLD（Complex PLD）器件。此后，其他 PLD 生产厂家都相继采用了 ISP 技术。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刷电路板（Printed Circuit Board，PCB）上，然后通过计算机的并行口和专用的编程电缆对焊接在 PCB 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。ISP 技术使得调试过程不需要反复插拔芯片，从而不会产生引脚弯曲变形的情况。目前，随着 Internet 技术的发展，可以通过网络远程对 ISP 器件进行编程，以实现不同的逻辑功能。进入 20 世纪 90 年代后，可编程逻辑器件的发展十分迅速，主要表现为三个方面：一是规模越来越大；二是速度越来越快；三是电路结构越来越灵活，电路资源更加丰富。在有些可编程逻辑器件中还集成了微处理器、数字信号处理单元和存储器等，这样，一个完整的数字系统甚至仅用一片可编程逻辑器件就可实现，即所谓的片上系统 SOC（System On Chip）。

各个厂家生产的大容量可编程逻辑器件不尽相同，由于逻辑模块的规模不同，逻辑模块的结构不同，布线资源的结构也不同，导致了众多的品种竞相争艳。用户可根据自己设计的特点选择适当的 PLD 器件，在器件厂家提供的开发软件的配合下把用户的设计合理地映射到某个大容量可编程逻辑器件的特定结构中去。如果没有良好的开发软件的支持，对于用户来说，大容量可编程逻辑器件是无法应用的。

二、可编程逻辑器件的分类

（一）按集成度分类

集成度是集成电路的一项很重要的指标，按照集成度可以将可编程逻辑器件分为两类：

低密度可编程逻辑器件 LDPLD（Low Density PLD）和高密度可编程逻辑器件 HDPLD（High Density PLD）。

一般说来，PROM、PLA、PAL、GAL 器件属于 LDPLD，而 EPLD、CPLD、FPGA 器件则属于 HDPLD。目前，随着高密度可编程逻辑器件的价格不断下降，低密度可编程逻辑器件在实用电子系统设计中已经基本上不再使用。但是高密度可编程逻辑器件是在低密度可编程逻辑器件的基础上逐渐发展起来的。早先的一片低密度可编程逻辑器件成为了高密度可编程逻辑器件的一个基本电路单元，一片高密度可编程逻辑器件中可能集成了很多这样的基本电路单元。所以要学习高密度可编程逻辑器件的结构、资源和使用方法，必须从低密度可编程逻辑器件开始。

（二）按基本结构分类

目前常用的可编程逻辑器件都是从与或阵列和门阵列两种基本结构发展起来的，所以可以从结构上将其分成两大类器件：PLD 器件和 FPGA 器件。

这种分类方法将基本结构为与—或阵列的器件称为 PLD 器件，将基本结构为门阵列的器件称为 FPGA 器件。LDPLD（PROM、PLA、PAL、GAL）、EPLD、CPLD 器件的基本结构都是与—或阵列，而 FPGA 器件则是一种门阵列结构。

（三）按编程工艺分类

所谓编程工艺，是指在可编程逻辑器件中可编程元件的类型。可编程逻辑器件之所以能够让用户在自己的实验室就可以编程，来实现自己特定的逻辑功能，是因为在器件的内部除了有基本的逻辑功能单元外，还有一些资源是用来改变基本的逻辑功能单元之间的连接关系的。从这个角度出发，可编程逻辑器件可分成五类。

1. 熔丝（Fuse）或反熔丝（Anti-Fuse）编程工艺的器件

早期的一些产品如 PROM、Xilinx 公司的 XC8100 系列 FPGA 和 Actel 公司的 FPGA 等采用熔丝或反熔丝作为编程元件，目前已经基本上不再使用。

2. UVEPROM 编程工艺的器件

该器件是用紫外线擦除信息、用电信号编程的器件。Altera 公司的 Classic 系列和 MAX5000 系列 EPLD 采用的就是这种编程工艺。

3. EEPROM 编程工艺的器件

该器件是用电擦除信息、用电编程的器件。Altera 公司的 MAX7000 系列、MAX9000 系列和 Lattice 公司的 GAL 器件、ispLSI 系列 CPLD 都属于这一类器件。

4. FLASH（闪速存储器）编程工艺的器件

Atmel 公司的部分低密度 PLD、Xilinx 公司的 XC9500 系列 CPLD 采用这种编程工艺。

5. SRAM 编程工艺的器件

Xilinx 公司的 FPGA（除 XC8100 系列）和 Altera 公司的 FPGA（FLEX 系列、APEX 系列）均采用这种编程工艺。

对于前四类可编程逻辑器件，它们在编程后，编程数据就保持在器件上，系统失掉电源后所存储的编程数据不丢失，系统再次加上电源后器件立即进入正常工作状态，故将这类器件称为非易失性器件。这类器件的编程原理可参考第七章半导体存储器中的相关内容，方法完全相同，此处不再赘述。而对于第五类 SRAM 编程工艺的可编程逻辑器件，存储在 SRAM 中的配置数据在掉电后会丢失，在每次上电后都要重新进行配置，因此将这类器件

称为易失性器件。由于熔丝或反熔丝编程器件只能编程一次，所以又将这类器件称为一次性可编程逻辑器件，即 OTP（One Time Programmable）器件，其他各类器件均可以多次编程。

（四）按照制造工艺分为双极型和 MOS 型

按制造工艺分，可编程逻辑器件有双极型和 MOS 型两种。使用双极型三极管作为基础电路元件的可编程逻辑器件为双极型制造工艺，使用 MOS 三极管作为基础电路元件的可编程逻辑器件为 MOS 制造工艺。目前大容量的可编程逻辑器件都是 MOS 制造工艺的。

（五）其他分类方法

还有人把可编程逻辑器件分为简单可编程逻辑器件 SPLD（Simple PLD）和复杂可编程逻辑器件 CPLD，将 FPGA 也归于 CPLD 中。

原则上，各种分类方法之间是相互联系、并行不悖的。在各类可编程逻辑器件中，目前大量生产和广泛应用的是以 CPLD 和 FPGA 为代表的 HDPLD，它们都采用 CMOS 制造工艺，编程工艺大多采用 SRAM、EEPROM、FLASH。

三、可编程逻辑器件中信号连接关系的表示和门电路的惯用画法

由于可编程逻辑器件中的资源很丰富，门电路的输入端个数较多，与门、或门等电路单元的表示方法不再使用前面几章中的表示方法，而改用一种比较紧凑的表示方法。由于用户自己可以在实验室改变信号的连接关系，所以在可编程逻辑器件中信号线之间连接关系的表示方法也有所改变，如图 8-1 所示。图 8-1（a）中的圆黑点表示两条信号线之间是芯片生产厂家已经做好的固定连接，用户不能在自己的实验室再通过编程改变它们之间的连接关系。图 8-1（b）中的×表示两条信号线之间的连接关系是用户在自己的实验室通过编程实现的，言外之意是用户如果需要还可以通过编程断开这两条信号线之间的连接关系。总之，图 8-1（a）和图 8-1（b）都表示这两条信号线之间是连通的。图 8-1（c）表示两条信号线之间没有任何连接关系。

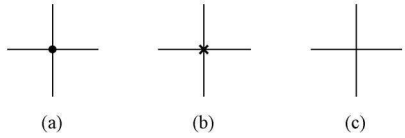


图 8-1　可编程逻辑器件中连接的表示方法

可编程逻辑器件中门电路的新的表示方法如图 8-2 所示。其中，在门的输入端只画一根线，门的输入信号画成与这根线相垂直的线，有几个输入端画几根垂直线，这样可以大大减小门符号的面积。

图 8-2（a）表示的是一个三输入的与门，根据连接关系可知，与门输出 $P=ABD$，其中 B 是固定连接，而 A、D 是可编程连接。图 8-2（b）表示输出恒等于 0 的与门，当一个与门的所有输入变量都连接时，可以在门的符号中画一个叉，而门的所有输入端的可编程连接可以省去不画。与门输出 $P=\overline{A}A\,\overline{B}B=0$。图 8-2（c）表示或门 $Y=P_1+P_3+P_4$，其中 P_1 是固定连接，而 P_3、P_4 是可编程连接。图 8-2（d）表示互补输出的缓冲器。图 8-2（e）表示三态输出的缓冲器。

四、与或阵列图

与或阵列是用多个与门和或门构成的一种阵列结构，任意组合逻辑函数都可以表示成标准与或式和一般与或式。标准与或式是唯一的，而一般与或式则可以有多种形式。图 8-3 表明了一个不可编程的与阵列和一个可编程的或阵列，这就是可编程逻辑器件的与或阵列图。不难写出图 8-3 中输出变量的逻辑表达式为

$$F_1(A,B)=\sum_m(0,1,3)$$
$$F_2(A,B)=\sum_m(0,2,3)$$

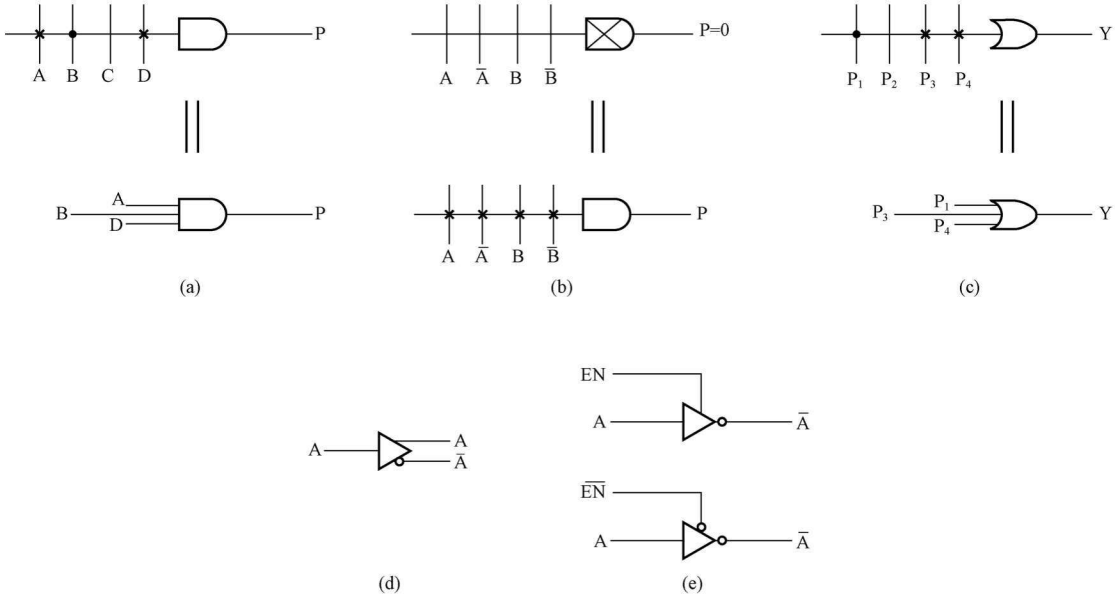


图 8-2 可编程逻辑器件中门电路的惯用画法

有时为了方便，将阵列中的逻辑门省略掉，简化成图 8-4 的形式。

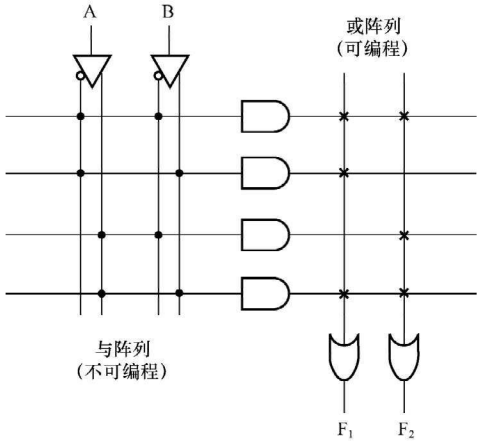


图 8-3 与或阵列图

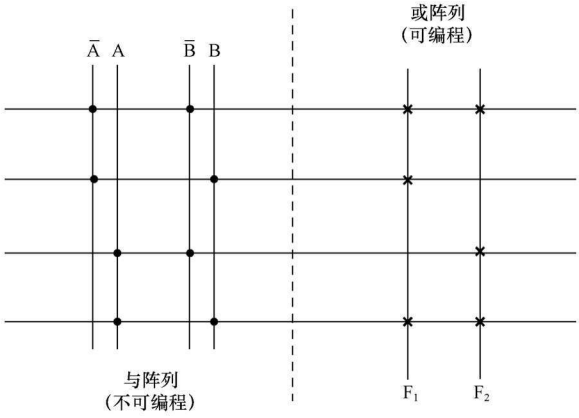


图 8-4 与图 8-3 对应的与或阵列图的简化形式

第二节 可编程逻辑器件

一般说来，PROM、PLA、PAL、GAL 器件属于低密度可编程逻辑器件（LDPLD），目前这种低密度可编程逻辑器件使用较少，学习这一部分知识的目的是为了更好地理解高密度可编程逻辑器件的结构和工作原理，因为这种低密度可编程逻辑器件是构成高密度可编程逻辑器件的基本单元。

图 8-5 是低密度可编程逻辑器件的结构示意图。图中电路由输入电路、与阵列、或阵列和输出电路四部分组成。其中，与阵列和或阵列是主体部分，逻辑函数主要靠它们来实现。与阵列的每一个输入端（包括内部反馈输入）都有输入缓冲电路，从而使输入信号具有足够的驱动能力，并且产生原变量和反变量两个互补信号。有些器件的输入电路还含有锁存器，甚至是宏单元（Macro Cell），可以实现对输入信号的预处理。低密度可编程逻辑器件的电路有多种输出方式，可以由或阵列直接输出（组合方式），也可以通过寄存器输出（时序方式）。输出可以是高电平有效，也可以是低电平有效。无论采用哪种输出方式，输出信号一般最后都是经过三态结构或集电极开路结构的输出缓冲器送到芯片的输出引脚。输出信号还可以通过内部通路反馈到与阵列的输入端。

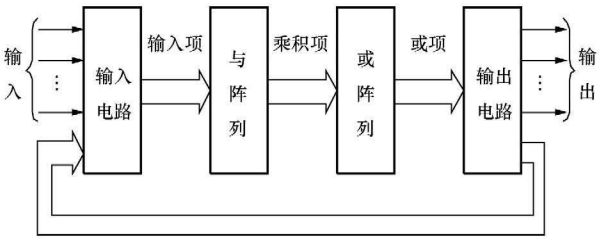


图 8-5　低密度可编程逻辑器件的结构示意图

通过前面的学习，已经知道任何组合逻辑函数都可以写成与或表达式，从而用与门—或门这种二级电路来实现。而任何时序电路又都是由组合电路加上存储器件构成的，因此这种结构对实现数字电路具有普遍意义。

表 8-1 列出了几种低密度可编程逻辑器件的结构特点。

表 8-1　　低密度可编程逻辑器件的结构特点对比

器件名	与阵列	或阵列	输出电路	器件名	与阵列	或阵列	输出电路
PROM	固定	可编程	固定	PAL	可编程	固定	固定
PLA	可编程	可编程	固定	GAL	可编程	固定	可编程

一、PROM 器件

PROM 器件的结构和使用方法在第七章中已详细讲述，此处不再赘述。PROM 的与阵列固定，与阵列输出的乘积项全部是最小项，或阵列可编程。用 PROM 实现组合逻辑函数时不用化简，直接应用最小项表达式即可。下面举例说明 PROM 在实现多输出组合逻辑函数中的应用。

【例 8-1】 用适当容量的 PROM 实现两位二进制数快速乘法器，要求画出与或阵列图。

解　两位二进制数快速乘法器的输入是两个 2 位二进制数，输出的结果是 4 位二进制数。可以设被乘数为$(A_1A_0)_2$，乘数为$(B_1B_0)_2$，则$(A_1A_0)_2\times(B_1B_0)_2=(D_3D_2D_1D_0)_2$。只要将 $A_1A_0B_1B_0$ 按顺序作为 PROM 的地址，把它们的乘积存放在相应的存储单元，即可实现两个 2 位二进制数的快速乘法。该快速乘法器的真值表如表 8-2 所示。用 PROM 实现多输出组合逻辑函数时不用化简，直接使用真值表。

该快速乘法器的 PROM 与或阵列图如图 8-6 所示，它的容量为 16K×4 位。如果要实现 $m\times n$ 快速乘法器，PROM 的容量至少为 $2^{m+n}\times(m+n)$位。用 PROM 实现组合逻辑函数时不需要化简，直接使用标准与或式。

表 8-2 快速乘法器的真值表

输入信号				输出信号			
A_1	A_0	B_1	B_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

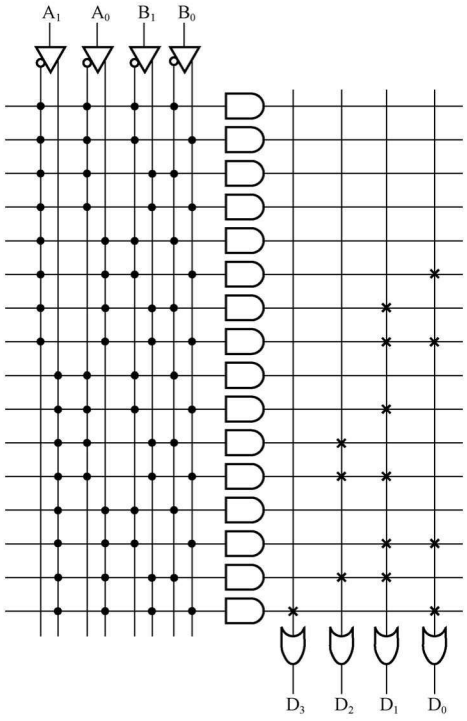


图 8-6 快速乘法器的 PROM 与或阵列图

二、可编程逻辑阵列 PLA 器件

为了提高对芯片的利用率，人们在 PROM 器件的基础上又开发出了一种与阵列、或阵列都可以编程的 PLD，即可编程逻辑阵列 PLA。PLA 与 PROM 相似，为节省逻辑资源，PROM 中的可产生全部最小项的与阵列用可编程的与阵列代替。PLA 的与阵列不提供全部最小项，只提供输入变量的有限个最小项。PLA 的与阵列、或阵列均可编程以决定是否连接。这样，与阵列输出的乘积项不必像 PROM 那样一定是最小项，在采用 PLA 实现组合逻辑函数时可以运用逻辑函数经过化简后的最简与或式，而且与阵列输出的乘积项的个数也可以小于 2^n（n 为输入变量的个数），从而减小了阵列的规模。

按照输出方式，PLA 可以分成两类：一类 PLA 以时序方式输出，在这类 PLA 的输出电路中除了输出缓冲器以外还有触发器，适用于实现时序逻辑电路，称为时序逻辑 PLA；另一类 PLA 以组合方式输出，在这类 PLA 中不含有触发器，适用于实现组合逻辑电路，称为组合逻辑 PLA。

【例 8-2】 试用组合逻辑 PLA 实现从 4 位自然二进制代码到格雷码的转换。

解 4 位自然二进制代码转换为格雷码的真值表如表 8-3 所示。采用 PLA 实现组合逻辑函数时，必须先对逻辑函数进行化简，以提高对芯片的利用率。对多输出逻辑函数进行化简时要注意合理使用逻辑函数之间的公共项，从而共享某些乘积项，使整个设计中的乘积项

的总数最少。用卡诺图进行逻辑函数化简，可以得到最简与或式为

表 8-3　［例 8-2］的真值表

输入信号				输出信号			
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

$$G_3 = B_3$$
$$G_2 = B_3\overline{B}_2 + \overline{B}_3 B_2$$
$$G_1 = B_2\overline{B}_1 + \overline{B}_2 B_1$$
$$G_0 = B_1\overline{B}_0 + \overline{B}_1 B_0$$

本例中简化形式的与或阵列图如图 8-7 所示。

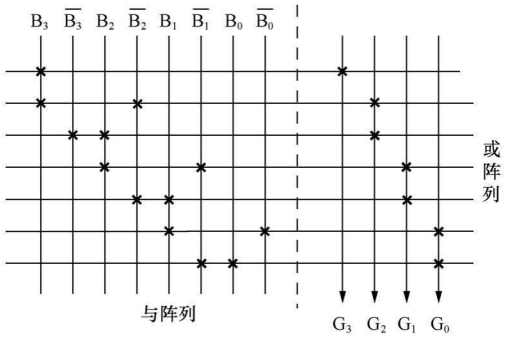


图 8-7　［例 8-2］的简化形式的与或阵列图

三、可编程阵列逻辑 PAL 器件

可编程阵列逻辑 PAL 的主要部分仍然是与或阵列，其中与阵列可根据需要进行编程，而或阵列是固定的。与阵列的可编程性保证了与门输入变量的灵活性，而或阵列固定使器件得以简化，进一步提高了对芯片的利用率。与 PLA 相比，PAL 是一种更加有效的 PLD 结构，被后来发展起来的许多 PLD 所采用。

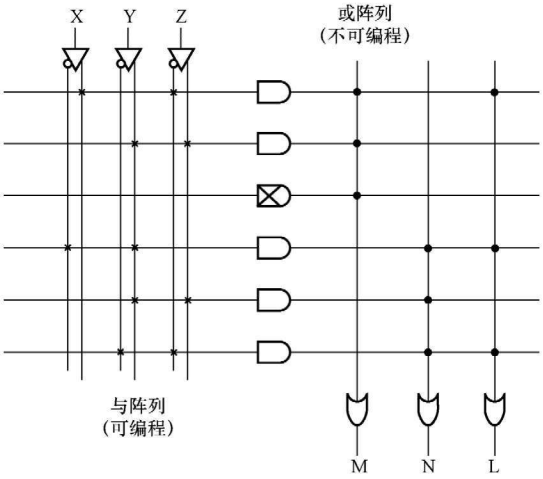


图 8-8　［例 8-3］的与或阵列图

【例 8-3】　用 PAL 实现以下三输出组合逻辑函数：

$$\begin{cases} M(X,Y,Z) = \sum_m(3,4,6,7) \\ N(X,Y,Z) = \sum_m(0,2,3,4,7) \\ N(X,Y,Z) = \sum_m(0,2,3,4,6) \end{cases}$$

解　由题意化简得

$$\begin{cases} M = X\overline{Z} + YZ \\ N = \overline{X}Y + YZ + \overline{Y}\,\overline{Z} \\ L = X\overline{Z} + \overline{X}Y + \overline{Y}\,\overline{Z} \end{cases}$$

由此可画出相应的与或阵列图，如图 8-8 所示。

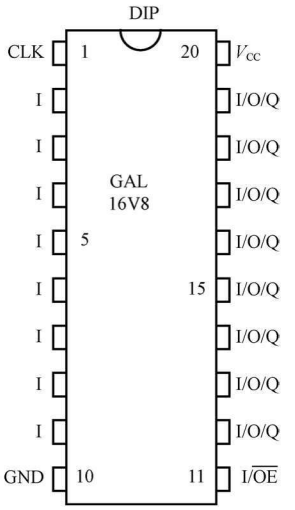


图 8-9 GAL16V8 DIP 封装引脚图

四、通用阵列逻辑 GAL 器件

通用阵列逻辑 GAL 是在 PAL 的基础上发展起来的，继承了 PAL 的与或阵列结构，与 PAL 完全兼容。GAL 的输出逻辑采用宏单元（Output Logic Macro Cell，OLMC），可以通过编程将 OLMC 设定成多种输出结构，大大增强了芯片的通用性和灵活性。另外，GAL 采用 EEPROM 编程工艺，可以用电擦除并重复编程。

典型的 GAL 器件主要有 Lattice 公司的 GAL16V8，其中，“16”表示与阵列的输入变量数；“8”表示输出端的个数，“V”则是输出方式可以改变的意思。下面以 GAL16V8 为例详细讲解 GAL 这一类器件的内部结构。图 8-9 是 GAL16V8 的双列直插封装（Dual In Line，DIP）引脚图。

（一）GAL 的基本结构

图 8-10 是 GAL16V8 的内部结构示意图。它主要由五部分组成：

（1）8 个输入缓冲器（引脚 2～9 作为固定输入端口）；

（2）8 个三态结构的输出缓冲器（引脚 12～19 作为 I/O 端口）；

（3）8 个 OLMC（OLMC12～OLMC19）；

（4）与阵列和 OLMC 之间的 8 个反馈缓冲器；

（5）一个规模为 64K×32 位的可编程与阵列，它共有 32 个输入和 64 个乘积项，这 64 个乘积项平均分配给 8 个 OLMC。

除了以上五个部分以外，GAL16V8 还有一个专用时钟输入端 CLK（引脚 1）、全局输出使能信号 OE 输入端（引脚 11）、一个工作电源端 V_{CC}（引脚 20，一般 $V_{CC}=5V$）和一个接地端 GND（引脚 10）。

在对 GAL16V8 进行编程时，需要用到以下几个引脚：引脚 1 为编程时钟输入端 S_{CLK}，引脚 11 为编程电压输入端 PRLD，引脚 9 被作为编程数据串行输入端 S_{DI}，引脚 12 为编程数据串行输出端 S_{DO}，电源端 V_{CC}（引脚 20）和接地端 GND（引脚 10）。

（二）GAL 编程单元的行地址映射图

图 8-11 是 GAL16V8 的编程单元行地址映射图，表明了在 GAL16V8 中编程单元的地址分配和功能划分。编程是逐行进行

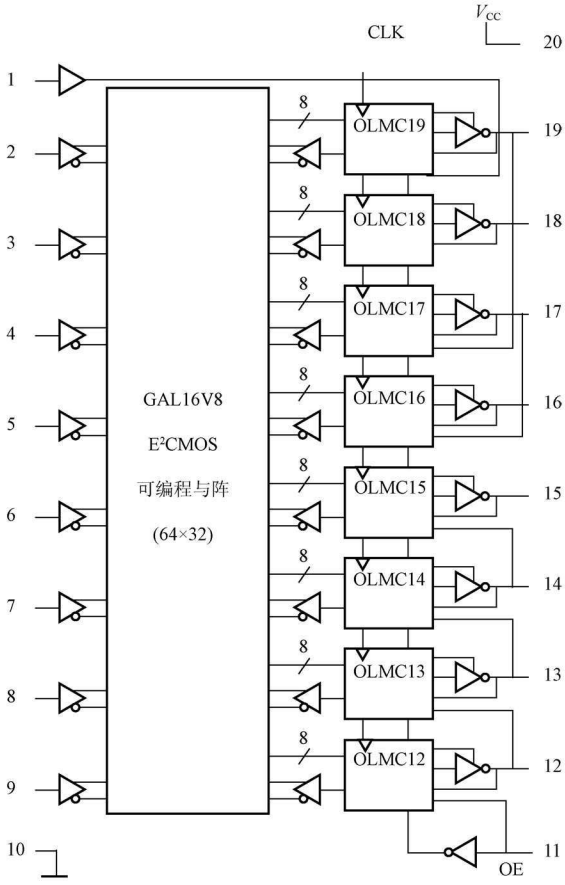


图 8-10 GAL16V8 的内部结构示意图

的，编程数据在相应开发软件的控制下串行输入到 64 位移位寄存器中，每装满一次就向编程单元写入一行数据。各编程单元行的功能如下：

第 0～31 行是与阵列的编程单元，每行有 64 位，编程后可以产生 64 个乘积项。

第 32 行为芯片的电子标签，也有 64 位。用户可以在这里存放器件的编号、电路编号、编程日期、版本号等信息，以备查询。

第 33～59 行是生产厂家保留的空间，用户不能使用。

第 60 行是一个 82 位的结构控制字，用于控制 OLMC 的工作模式和乘积项的禁止。

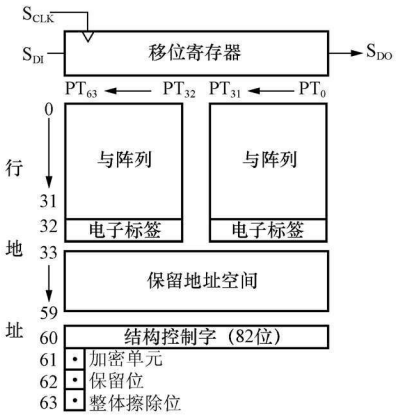


图 8-11　GAL16V8 的编程单元行地址映射图

第 61 行是一位加密单元，加密单元被编程后，与阵列中的编程数据不能被更改或读出，从而使设计成果得以保护。只有当整个芯片的编程数据被擦除时，加密单元才同时被擦除。但是电子标签不受加密单元的保护。

第 62 行是一位保留位。

第 63 行是一个整体擦除位，编程系统对这一位进行擦除将导致整个芯片中所有的编程单元都被擦掉。

（三）GAL16V8 的输出逻辑宏单元 OLMC

GAL16V8 的每一个输出端都有一个 OLMC。OLMC 被设定成哪一种输出结构取决于对结构控制字的编程。图 8-12 给出了 GAL16V8 的控制字示意图。图 8-13 给出了 GAL16V8 的 OLMC 结构图。

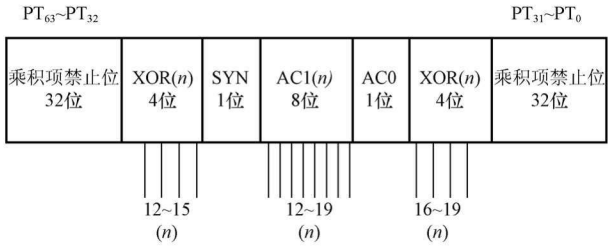


图 8-12　GAL16V8 的控制字示意图

从图 8-13 可看出，OLMC 主要包括以下四个部分。

1. 一个 8 输入的或门

或门的 7 个输入是直接来自于与阵列输出的乘积项，1 个输入来自于乘积项数据选择器的输出。

2. 一个可编程的异或门

通过对控制位 XOR(n)（括号中的 n 是 OLMC 的编号）的编程，可改变输出信号的极性。当 XOR(n)＝0 时，异或门的输出与另一个输入端相等；当 XOR(n)＝1 时，异或门的输出与另一个输入端相反。

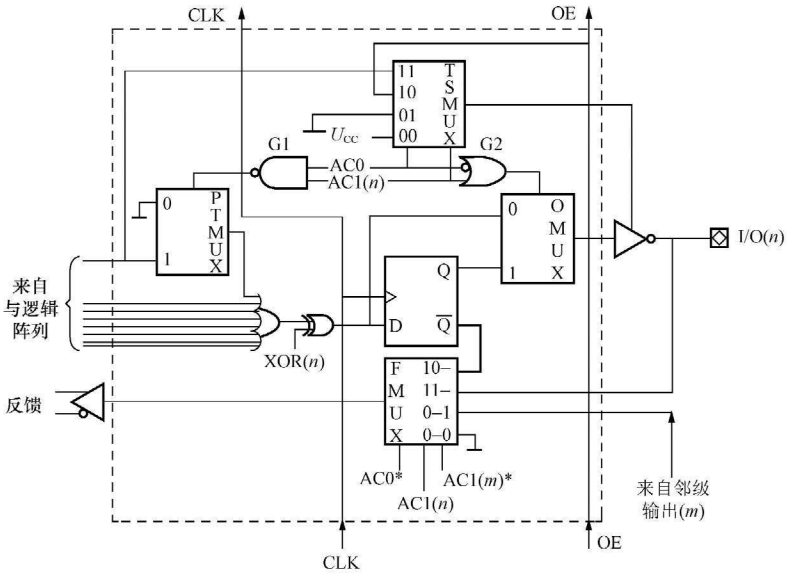


图 8-13 GAL16V8 的 OLMC 结构图

3. 一个 D 触发器

D 触发器用于实现时序逻辑的场合。

4. 四个数据选择器

(1) 乘积项数据选择器 (PTMUX)。它是一个二选一数据选择器，受控制位 AC0 和 AC1(n)控制 (AC0 是所有 OLMC 公用的控制位)。

当 AC0=0 或 AC1(n)=0 时，来自于与阵列的第 8 个乘积项被接入到或门的第 8 个输入端；当 AC0=AC1(n)=1 时，接入到或门的第 8 个输入端的信号为 0。

(2) 输出数据选择器 (OMUX)。它也是一个受控制位 AC0 和 AC1(n)控制的二选一数据选择器。当 AC0=0 或 AC1(n)=1 时，该 OLMC 采用组合输出方式；当 AC0=1 且 AC1(n)=0 时，该 OLMC 为寄存器同步输出。

(3) 三态数据选择器 (TSMUX)。它是一个受控制位 AC0 和 AC1(n)控制的四选一数据选择器，用于选择输出三态缓冲器的使能信号。

当 AC0=AC1(n)=0 时，选择 V_{CC} 作为使能信号，输出三态缓冲器处于常通状态；当 AC0=0 且 AC1(n)=1 时，选择地电平作为使能信号，输出三态缓冲器处于高阻状态，此时器件的引脚作为输入引脚使用；当 AC0=1 且 AC1(n)=0 时，输出三态缓冲器受全局输出使能信号 OE 控制；当 AC0=1 且 AC1(n)=1 时，选择来自于与阵列的第 8 个乘积项作为使能信号。

(4) 反馈数据选择器 (FMUX)。它是一个受本单元控制位 AC0、AC1(n)和相邻单元控制位 AC1(m)控制的四选一数据选择器，用于选择由 OLMC 反馈回与阵列的信号。

当 AC0=AC1(m)=0 时，反馈信号为 0；当 AC0=0 且 AC1(m)=1 时，反馈信号为相邻 OLMC 的输出；当 AC0=1 且 AC1(n)=0 时，反馈信号取自本单元寄存器的 $\overline{Q}$端；当 AC0=1 且 AC1(n)=1 时，反馈信号取自本单元的输出端。

除了以上提到的控制位外，在 GAL16V8 中还有一个同步位 SYN 和 64 个乘积项禁止

位。同步位 SYN 用于控制 GAL 是否有寄存器输出能力：当 SYN＝1 时，GAL 不具备寄存器输出能力；当 SYN＝0 时，GAL 具备寄存器输出能力。另外，在 GAL16V8 的 OLMC19 和 OLMC12 中，AC0 和 AC1(m)分别被$\overline{\text{SYN}}$和 SYN 所代替。64 个乘积项禁止位分别用于控制与阵列输出的 64 个乘积项。当某一个禁止位为 0 时，则相应的乘积项恒为 0，表明在逻辑中不需要这个乘积项。

根据以上所述，不难归纳出 OLMC 的四种工作模式：

（1）当 AC0＝0、AC1(n)＝0 时，OLMC 为专用组合输出模式，如图 8-14（a）所示；

（2）当 AC0＝0、AC1(n)＝1 时，OLMC 为专用输入模式，如图 8-14（b）所示；

（3）当 AC0＝1、AC1(n)＝0 且 SYN＝0 时，OLMC 为寄存器输出模式，如图 8-14（c）所示；

（4）当 AC0＝1 且 AC1(n)＝1 时，OLMC 为组合输入/输出模式，如图 8-14（d）所示。

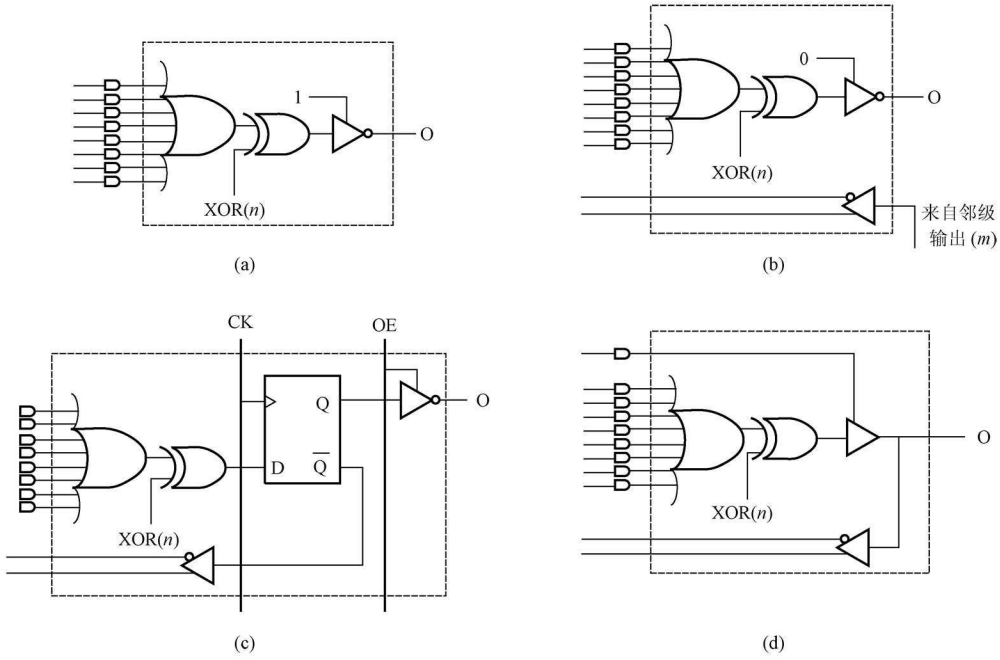


图 8-14　OLMC 的四种工作模式

（a）专用组合输出模式；（b）专用输入模式；（c）寄存器输出模式；（d）组合输入/输出模式

五、高密度可编程逻辑器件（HDPLD）

HDPLD 包括 EPLD、CPLD 和 FPGA 三种。这三种器件大致可以分为两类：一类是与标准门阵列结构类似的单元型 HDPLD，即 FPGA；另一类是基于与或阵列结构（或称为乘积项结构）的阵列扩展型 HDPLD，即 EPLD 和 CPLD，其中 CPLD 是 EPLD 的改进型器件。

经过十几年的发展，目前市场上 HDPLD 产品型号繁多，电路结构也千差万别，其中最具有代表性的是 Xilinx 公司的 FPGA 器件和 Altera 公司的 CPLD 器件。它们开发得较早，占据了大部分的 PLD 市场。目前 CPLD、FPGA 器件的界限越来越模糊。

HDPLD 器件的结构都很复杂，不可能手工来配置这些器件，必须依赖开发软件的配合，所以从用户的角度来说，只需大致了解这些器件中的资源，以帮助用户选择合适的器件，不必对器件的结构做特别深入的了解，其余的工作由相应的开发软件来完成。

（一）现场可编程门阵列（FPGA）

在前面所讲的几种 PLD 电路中，都采用了与或逻辑阵列加上输出逻辑单元的结构形式。而 FPGA 的电路结构则完全不同，它由若干独立的可编程逻辑模块组成，用户可以通过在自己的实验室的编程工作（现场，而不是芯片厂商的工厂）将这些模块连接成所需要的数字系统。因为这些模块的排列形式和门阵列中单元的排列形式相似，所以沿用了门阵列这个名称。

FPGA 基本结构由三种可编程单元和一个用于存放编程数据的静态存储器组成。三种可编程单元分别是可配置的逻辑模块（Configurable Logic Block，CLB），输入/输出模块（Input Output Block，IOB）和可编程开关矩阵（Switch Matrix）及连线资源。它们的工作状态全部由编程数据存储器中的数据设定。

FPGA 之所以能实现更复杂的逻辑，是因为可以通过可编程的互连线实现逻辑模块之间的相互连接，并使最后形成的逻辑层次数不受限制。这种连接线段是可增加的，连接线段的数目事前不可预估，因而总延迟时间也不可预估，这是 FPGA 电路结构的一个缺点。

（二）复杂可编程逻辑器件（CPLD）

CPLD 是在 GAL 的基础上扩展、改进而成的。尽管它的规模比 GAL 大得多，功能也强得多，但它的主体部分，即可编程逻辑阵列块，仍然是基于与或阵列的结构。

CPLD 采用了分区结构，即将整个芯片划分成多个逻辑阵列块（Logic Array Block，LAB）和输入/输出块（I/O）。每个逻辑阵列块都有各自的与或阵列、逻辑宏单元、输入和输出等，相当于一个独立的 SPLD。再通过一定方式的全局性可编程布线通道将这些 SPLD 和输入/输出连接起来，构成更大规模的 CPLD。简单地讲，CPLD 就是将多个 SPLD 集成到一块芯片上，并通过全局性可编程布线通道实现它们之间的连接。这种全局性可编程布线通道是遍布整个器件的连续布线通道（不同于 FPGA 器件中使用的分段走线结构），其延迟时间可以预估。

就编程工艺而言，多数的 CPLD 采用 EEPROM 编程工艺，也有采用 FLASH 编程工艺的。

目前，各大逻辑器件芯片厂商生产的 FPGA 和 CPLD 的界限已经模糊，芯片融合了 FPGA 和 CPLD 各自的优势，更便于用户的使用。就用户而言，只要大概了解芯片的内部结构，在 EDA 工具的支持下，就可以使用可编程逻辑器件进行数字系统的设计了。读者可以从厂商获得芯片的数据手册。

第三节　硬件描述语言 VHDL 的基本语法

一、硬件描述语言概述

硬件描述语言（Hardware Description Language，HDL）是硬件设计者和电子设计自动化（Electronic Design Automation，EDA）工具之间的界面。设计者使用 HDL 来描述自己的设计方案（或设计要求、设计意图），并把这个描述告诉 EDA 工具，最后在 EDA 工具的

帮助下进行详细设计和验证。

EDA 工具主要包括模拟（仿真）软件和综合软件。行为描述文件和输入信号激励作为模拟（仿真）软件的输入，待模拟（仿真）软件处理后，得到输出信号的波形图。行为描述文件和约束条件文件作为综合软件的输入，待综合软件处理后，得到网表和报告文件。

（一）成为 IEEE 标准的两种 HDL

1. VHDL

VHDL（Very High Speed Integrated Circuit HDL，VHSIC HDL）是在美国国防部支持下开发的一种硬件描述语言。美国国防部要求承担其项目的各公司都必须使用 VHDL 来描述设计。VHDL 在 1987 年成为 IEEE（Institute of Electrical and Electronic Engineers）Std 1076－1987 标准，1993 年更新为 IEEE Std 1076－1993 标准，2000 年更新为 IEEE Std 1076－2000 标准，2002 年更新为 IEEE Std 1076－2002 标准。

2. Verilog-HDL

Verilog-HDL 在 1995 年成为 IEEE Std 1364－1995 标准。2001 年发布了新标准 Std 1364－2001，在这个标准中，加入了 Verilog-HDL－A 标准，使 Verilog 有了模拟电路设计的能力。

在底层的 VHDL 设计环境是由 Verilog-HDL 描述的器件库支持的，因此它们之间的互操作性十分重要。目前，Verilog 和 VHDL 的两个国际组织 OVI 和 VI 正在筹划这一工作，准备成立专门的工作组来协调 VHDL 和 Verilog-HDL 语言的互操作性。

（二）VHDL 简介

VHDL 是在 20 世纪七八十年代，由美国国防部资助的 VHSIC 项目开发的产品。1988 年美国国防部规定所有官方的 ASIC 设计都必须以 VHDL 为设计描述语言，所以 VHDL 渐渐成为工业界的标准。VHDL 主要应用在三个方面。

1. 描述

在这个语言首次开发出来时，其目标仅是一个使电路文本化的一种标准，为了使人们采用文本方式描述的设计能够被其他人没有二意性地所理解，因为用自然语言描述电路会产生二意性。这个模型是让人来阅读的。

2. 模拟的模型

作为模型语言，用于采用模拟软件进行模拟。这个模型是让仿真软件来阅读的。

3. 综合的模型

在自动设计系统中，作为设计输入。这个模型是让综合软件来阅读的。

（三）VHDL 语言的特点

（1）像汉语、英语一样 VHDL 是一种高级语言，只不过在语法、词汇方面要简单得多，专用于描述硬件。

（2）可读性好。因为 VHDL 是高级语言，所以比逻辑符号图更易理解。

（3）VHDL 语言本身生命周期长。因为 VHDL 与工艺无关，不会因工艺的过时而使电路描述过时。

（4）支持大规模设计的分解和已有设计的再利用。

（5）VHDL 已成为标准，得到众多 EDA 公司的支持，可适用于多种工作平台。而其他输入方式与特定环境有关，不能重复使用。

（四）学习 VHDL 语言应注意的几个问题

（1）了解 VHDL 语言模拟器是如何模拟代码的过程，有助于弄清一些 VHDL 语句的语义。而对语义有一个清楚地理解，可使你能够精练准确地进行 VHDL 代码编写。目前常用的 VHDL 模拟软件有 Active HDL 和 Modelsim。

（2）VHDL 语言的一些构造，较多的是专用于模拟和验证而不是综合，综合软件也许会忽略掉这样的构造和规则。VHDL 是基于模拟的语言，它所提供的行为描述的一切方便手段实际上都是为建立模拟模型的。

（3）用于模拟的模型和用于综合的模型有差别。

（4）为综合而写的代码可以进行模拟，但不是所有为模拟而写的代码可以用来综合。

（5）应大致了解综合软件的工作原理。目前常用的综合软件有 Synplicity 公司的 Synplify 和 Synplify Pro 软件、Synopsys 公司的 FPGA Express 软件、Mentor 公司的 Leonardo Spectrum 软件、Xilinx 公司的 XST（Xilinx Synthesis Technology）软件。

（6）将 VHDL 和 CPLD、FPGA 的学习结合起来。

（7）应基本熟悉 CPLD、FPGA 器件的逻辑资源。

二、基本的 VHDL 模型

设计实体（design entity）是 VHDL 中的基本单元和最重要的抽象。它可以代表整个系统、一块电路板、一个芯片、一个单元或一个门电路，也可以代表如微处理器那样复杂的电路，还可以代表如单个逻辑门那样简单的电路，对于设计实体可以代表什么几乎没有什么限制。设计实体包括两部分，即实体说明（entity declaration）和结构体（architecture body）。实体说明相当于一个电路的符号，表明电路模块的对外接口信号。结构体相当于一个电路的电路原理图，对功能进行描述。

（一）实体说明（entity declaration）

实体说明给设计实体命名并给设计实体定义一个接口。

```
ENTITY andg IS
  GENERIC(tpd_hl,tpd_lh:Time:=1 ns;n:Positive:=2);
  PORT(input:IN Bit_Vector(1 to n);output:OUT Bit);
END andg;
```

在端口中使用的类型只能在程序包中提前定义。端口的次序可任意排列，输入端口不一定放在输出端口的前面，具有相同类型和模式的端口可以写在一起。

1. 类属说明（generic declaration）

```
GENERIC(tpd_hl,tpd_lh:Time:=1ns;n:Positive:=2);
```

（1）为设计实体和其外部环境通信的静态信息提供通道。静态是指一旦模型连接到仿真器，这些对象的值在整个仿真过程中保持不变。

（2）VHDL 本身与工艺无关，当工艺改变时，只需修改相应程序中的类属参数即可。

（3）隐含说明了常量对象。

（4）类属可以有任意数据类型，但综合工具所能接受的数据类型有所限制。

（5）类属与常数不同，常数只能从设计实体的内部得到赋值，且不能再改变。而类属的值可以由设计实体外部提供，因此，设计者可以从外面通过类属参量的重新设定而容易地改变一个设计实体，或一个元件的内部电路结构和规模。

（6）类属有许多用途，类属的用途只受模块作者创造性的限制。

2. 端口说明（port declaration）

PORT(input:IN Bit_Vector(1 to n);output:OUT Bit);

（1）为设计实体和其外部环境通信的动态信息提供通道。

（2）隐含说明了信号对象。

3. 端口模式（port mode）

（1）IN（流入设计实体）：仅允许数据流入端口。主要用于时钟输入、控制输入（复位、载入和使能）、单向的数据输入。

（2）OUT（流出设计实体）：允许对应多个信号（允许和其他设计实体的多个输入端相连），构造体内部不能再使用（不能用于内部反馈）。常常用于终端计数输出等。

（3）INOUT（双向端口，即可流入，又可流出，当然不是同一时刻），允许用于内部反馈，适用于纯双向信号，比如数据总线。

（4）BUFFER（流出设计实体）：只允许对应一个信号，构造体内部可再使用，可用于内部反馈，当然就不能对应多个信号。与其相连的另一个设计实体的对应端口也要用 BUFFER 说明。可用于计数器的输出（计数器的现态一定被用来决定其次态）。

（二）结构体（architecture body）

一个设计实体只有一个实体说明，但可有多个不同描述风格的结构体，分别代表该器件的不同实现方案。描述风格有行为描述（behavioural description）、数据流描述（dataflow description）、结构描述（structural description）和混合描述共四种。

结构体的基本结构如下：

ARCHITECTURE 结构体的名字 OF 实体说明的名字 IS

说明语句；

BEGIN

并行语句；

END 结构体的名字；

其中，结构体的名字和实体说明的名字可以相同，也可以不同。说明语句可以包括对象说明、类型说明、子程序说明、属性说明等，但是不能说明变量，在此处说明的这些信息只适用于本结构体。下面简略说明并行的概念。

一个实际硬件电路，信号从输入端向输出端经过不同的路径同时传播。并行 VHDL 语言应能表示出这种关系，模拟器在仿真时应能反映出这种并行关系。实际系统中，同一时刻可能有多个动作同时发生，相当于在 VHDL 模拟时钟的某个时刻有多个进程被激活，这就是行为的并行性。在模拟过程中，尤其是在非并行机上，并发的进程实际是串行处理的，这就要求模拟时钟在每一模拟时刻停下，直到在该时刻被激活的所有进程都处理完毕，模拟时钟才向前推进到下一时刻。这就保证了并发行为效果的同时性和并行性。

在 VHDL 标准中定义了变量 NOW，它表示模拟器内部绝对时间。

三、词法元素

词法元素（lexical elements）主要包括标识符、保留字、文字、注释、特殊符号等。

（一）标识符（identifiers）

标识符是用来给 VHDL 模型中的项目起名字的。起名字时应表明使用这个项目的目的，

名字的长度任意，不要使用 VHDL 已经预定义的标识符（比如 Bit、Time）和语言的保留字。

1. 短标识符（short identifiers）

短标识符又名基本标识符（basic identifiers）。它适用于 VHDL－1987、VHDL－1993、VHDL－2000 和 VHDL－2002 标准。其规定如下：

（1）有效字符：a～z、A～Z、0～9、_（下划线），即只能是 26 个小写的英文字母、26 个大写的英文字母、十进制的 10 个数码和一个下划线符号；

（2）必须以英文字母开头；

（3）不能以数字或下划线开头；

（4）下划线的前后都必须有英文字母或数字，即不能有两个连续的下划线字符；

（5）不区分大小写。

例如，正确的标识符：Memory1 Adder _ Module Bus _ 16 _ Bit。

例如，错误的标识符：9Memory1 ——数字开头；

Adder __ Module ——两个连续的下划线；

_ qwer ——以下划线开头；

mqw # qw ——无效的字符 #；

Bus _ 16 _ Bit _ ——以下划线结束。

2. 扩展标识符（extended identifier）

扩展标识符仅适用于 VHDL－1993、VHDL－2000 和 VHDL－2002 标准，规定如下：

（1）用反斜杠 \ \ 来界定；

（2）区分大小写；

（3）扩展标识符永远与短标识符不同；

（4）可以包含任何字符序列。例如，\ Counter \，\ counter \，\ counter # \，\ cou nter，\，\ data bus \，\ global. clock \，\ 923 \，\ d # 1 \，\ start __ \，\ A：\\ name \ 都是合法的扩展标识符。

特别注意，\ Counter \、Counter 和 \ counter \ 是三个不同的标识符。

（二）保留字（reserved words，key words）

一些标识符又称为保留字，在 VHDL 中有特殊的应用。它们用来表示模型的特定结构，这些标识符不能用做用户定义的标识符。有如下这些保留字：

ABS ACCESS AFTER ALIAS ALL AND ARCHITECTURE ARRAY ASSERT ATTRIBUTE

BEGIN BLOCK BODY BUFFER BUS

CASE COMPONENT CONFIGURATION CONSTANT

DISCONNECT DOWNTO

ELSE ELSIF END ENTITY EXIT

FILE FOR FUNCTION

GENERATE GENERIC GROUP GUARDED

IF IMPURE IN INERTIAL INOUT IS

LABEL LIBRARY LINKAGE LITERAL LOOP

MAP MOD
NAND NEW NEXT NOR NOT NULL
OF ON OPEN OR OTHERS OUT
PACKAGE PORT POSTPONED PROCEDURE PROCESS PROTECTED PURE
RANGE RECORD REGISTER REJECT REM REPORT
RETURN ROL ROR
SELECT SEVERITY SHARED SIGNAL SLA SLL SRA SRL SUBTYPE
THEN TO TRANSPORT TYPE
UNAFFECTED UNITS UNTIL USE
VARIABLE
WAIT WHEN WHILE WITH
XNOR XOR

（三）文字（literals）

文字有标量类型的文字和复合类型的文字（聚集）两种。

1. 标量类型的文字

（1）整数文字、浮点文字。

整数文字和浮点文字的相同点为：

1）均可用 2 至 16 进制中的任一进制表示；

2）基数和指数用十进制表示；

3）十进制数不用标明基数，其他进制数需在最左端标明基，以＃分隔；

4）不是十进制则需用＃将基数和指数隔开；

5）均有一般计数法和科学计数法两种；

6）为增加可读性可用下划线划分。

整数文字和浮点文字的不同点为：

1）整数文字无小数点，若有指数，其值只能为正；

2）浮点文字有小数点，若有指数，其值可正可负；

例如，2，0，77459102，10E4，16＃D2＃，8＃720＃，2＃11010010＃均为整数文字；65971.333333，65_971.333_333，8＃43.6＃e＋4，8＃43.6＃e－4 均为浮点文字。为增加可读性，浮点数可以用下划线分隔。浮点数的指数部分可为正数也可为负数。

十进制数 253 的几种表示形式：2＃11111101＃、16＃FD＃、16＃0fd＃、8＃0357＃、253。

十进制数 0.5 的几种表示形式：2＃0.100＃、8＃0.4＃、12＃0.6＃、0.5。

十进制数 1024 的几种表示形式：2＃1＃E10、16＃4＃e2、10＃1024＃E＋00、1024。

（2）字符文字。字符文字是用单引号括起来的单个可打印的 ISO 8859 Latin－1 eight－bit 字符，区别大小写，即‘A’与‘a’不同。VHDL－1987 标准只使用 ASCII 字符集。

（3）字符串文字。字符串文字是括在双引号中的一串 ISO 8859 Latin－1 eight－bit 字符，区别大小写，常用于程序的提示和说明。VHDL－1987 标准只使用 ASCII 字符集。字符串文字必须在一行内写完，若要换行，在下一行开始可用“&”连接。

（4）位串文字。位串文字是括在双引号中的一串 bits（binary digits），这些字符仅限于

0～9，a～f，A～F，并且串前还有一个用以标明基数的字母（大小写均可），可用下划线分隔，以增加可读性。适用于 Bit 向量。下面是几个位串文字：

B“1111_0101” ——若使用了下划线，则 B 一定要显式写出；

“11110101” ——若省略了基，则一定是二进制，此位串的长度为 8；

“1111_0101” ——错误，若使用下划线，则基一定不能省略；

X“F0F” ——等效于 B“111100001111”，此位串的长度为 12；

O“70_7” ——等效于 O“707”或 B“111000111”，此位串的长度为 9；

“c3_3c” ——错误。

（5）物理文字。物理文字由整数文字或浮点文字加上表示测量单位或比例因子的标识符组成。表示测量单位的标识符本身也是物理文字，可认为前面有一整数文字 1。数字和单位之间必须包含空格。下面列举的都为物理文字：

15 ft、10 Kohm、2.3 sec、sec、1 sec

2. 复合类型的文字

复合类型的文字又称为聚集，在后面有专门的例子讲解。

（四）注释

当用 VHDL 语言描述数字电路时，一定要用注释对代码进行说明。因为这些代码以后有可能还要被代码设计者本人或者其他工程师重复使用多次。为了使用方便，在编写代码时注意同时尽可能清楚地说明作者的意图。注释用双引号（“”）开头。

（五）特殊符号

VHDL 使用如下一些符号用做运算符、语言结构的分割符和标点符号：

" # & ' ( ) * + − , . / : ; < = > [ ] =>

** := /= >= <= <>

四、对象

（一）对象（objects）的概念

对象是一些存放值的容器，从模拟的角度理解就是在内存中开辟的一些区域。对象包括信号（SIGNAL）、变量（VARIABLE）、常量（CONSTANT）和文件（FILE）四种。

（二）说明一个对象的方法（object declaration）

1. 显式对象说明

说明对象的时候必须同时为其说明一个数据类型。例如

VHDL的保留字，不一定非用大写字母，是为便于阅读、清晰起见才这样写的。

数据类型，对于VHDL预定义的数据类型起始字母习惯用大写，自定义的习惯用小写字母，实际上并不区分大小写。

```
SIGNAL      clock:          Bit;
VARIABLE    sum:            Real;
CONSTANT delay1，delay2:    Time: =10 ns;
```

用户定义的标识符，是对象的名字，习惯用小写字母，并不区分大小写。

2. 隐式对象说明

（1）在端口说明语句中，隐式说明一个或几个信号类型的对象；

（2）在类属说明语句中，隐式说明一个或几个常量类型的对象；

（3）子程序（函数和过程）中的参量；

（4）循环语句和生成语句的下标。

（三）如何正确理解对象

1. 从模拟器的角度理解

（1）变量只有当前值，对变量的赋值没有延迟，立即成为当前值，内存开销小，用于硬件特性的高层次建模所需要的计算中。

（2）信号除当前值外还有许多相关的信息，如历史信息（保存在预定义属性中）和投影输出波形（保存在驱动源中），信号赋值至少有“δ”延迟，不能立即成为其当前值，内存开销大，占用内存多，以便存贮一些相关的内容。信号是硬件中连线的抽象描述。

（3）常量是为特定的数据类型值所赋予的名称。

2. 从综合器的角度理解

（1）对信号和变量的认识单从行为仿真和语法的角度去认识是不完整的。事实上，在许多情况下，综合后所对应的硬件电路结构中信号和变量没什么区别。例如，在满足一定条件的进程中，综合后它们都能引入寄存器，其关键在于它们都具有能够接受赋值这一重要特性，而 VHDL 综合器并不理会它们在接受赋值时存在的延迟特性。

（2）建议只有在临时保存数值时才使用变量。

五、数据类型

（一）数据类型（data type）的定义

在 VHDL 模型中数据类型的概念非常重要，VHDL 是一种强类型的语言。数据类型给定了一个值域，同时还定义了可用于这些值上的操作。具有这个数据类型的对象只能从中取值，说明对象的时候必须同时为其说明一个数据类型。

（二）数据类型的分类

- 标量类型（scalar type）
  - 整数类型（integer type）
  - 实数类型（real type）　综合工具不支持
  - 枚举类型（enumeration type）
  - 物理类型（physical type）
- 复合类型（composite type）
  - 数组类型（array type）
  - 记录类型（record type）　综合工具不支持
- 存取类型（access type）　综合工具不支持
- 文件类型（file type）　综合工具不支持

（三）数据类型的说明方法

1. 整数类型、实数类型

整数类型、实数类型说明的格式如下：

TYPE 标识符 IS 类型定义；

其中，类型定义即范围限制，由保留字 RANGE 和范围说明两部分组成。范围说明包括两个表达式和一个方向（TO 、DOWNTO）；使用预定义属性 RANGE。

以下是 VHDL 在 Standard 包中预定义的两个数据类型：

TYPE Real IS RANGE －1.0E308 TO ＋1.0E308；

TYPE Integer IS RANGE－（2 ** 31－1）TO（2 ** 31－1）；

以下是用户自定义的数据类型：

TYPE twos _ type IS RANGE－32768 TO＋32768；

TYPE voltage IS RANGE 0.0 TO 10.0；

TYPE word _ index IS RANGE 15 DOWNTO 0；

2. 枚举类型

枚举类型说明的格式如下：

TYPE 标识符 IS（类型定义）；

其中，括号中的类型定义是枚举文字表，可以是标识符或字符文字。

以下是 VHDL 在 Standard 包中预定义的三个数据类型：

TYPE Boolean IS(False,True)；

TYPE Bit IS(‘0’,‘1’)；

TYPE Severity_level IS(Note,Warning,Error,Failure)；

以下是 VHDL 用户自定义的两个数据类型：

TYPE traffic_lights IS(red,yellow,green)；

TYPE two_level_logic IS(‘0’,‘1’)；

在逻辑电路中，信号值一般用布尔值（0，1）表示。除此之外，为了表示信号波形的各种状态，在模拟系统中经常增加一些特殊用途的值。除了信号值本身外，有时还要区分相同信号值的不同驱动负载能力（称为信号强度），选取不同信号值和信号强度的组合，就可以构成各种功能的模拟器信号模型，例如，二值模型、三值模型、四值模型、九值模型等。

IEEE 制定了一个多值逻辑系统标准包，称为 Std-logic－1164，用 VHDL 语言 package 形式描述。它可以编译成名为 IEEE 的 VHDL 库（library）。Std-logic－1164 定义了一个九值模型，每个值为逻辑电平（0 、1 、未知）与强度（强制、弱、高阻、未定、无关）的组合。

IEEE 标准九值模型的定义如下：

```
TYPE Std_Ulogic IS(‘U’,        ——未定、未知
                   ‘X’,        ——强制、未知
                   ‘0’,        ——强制、0
                   ‘1’,        ——强制、1
                   ‘Z’,        ——高阻、未知
                   ‘W’,        ——弱、未知
                   ‘L’,        ——弱、0
                   ‘H’,        ——弱、1
                   ‘－’        ——无关、未知
                   )；
```

Std _ Ulogic 为九值模型（nine-valued data type）。该模型信号强度分五种：强制、弱、高阻、未定、无关，信号值分三种：0 、1 、未知。

强度用来表示信号的驱动能力，其中，强制表示该信号连接电源或地，或者是输入激励；弱表示经过较大的电阻与电源或地相连；高阻表示晶体管截止，信号与电源或地相隔

离。在九值模型中，高阻、未定、无关这三种信号强度只对应“未知”一个逻辑电平。

‘U’表示没有初始化的状态。这一状态用来检查时序逻辑电路是否被用适当的方式初始化。带存储功能的器件，如触发器，在仿真开始时首先被初始化为‘U’，如果电路初始化的逻辑功能正确，所有‘U’应在有限仿真时间内变为确定态‘0’、‘1’、‘H’、‘L’、‘X’。

3. 物理类型

物理类型的定义与整数类型的定义相似，只是多了一个单位。其说明格式如下：

```
TYPE Time RANGE－2147483647 TO ＋2147483647
  UNITS
  fs;
  ps＝1000 fs;
  ns＝1000 ps;
  us＝1000 ns;
  ms＝1000 us;
  sec＝1000 ms;
  min＝60 sec;
  hr＝60 min;
END UNITS;
TYPE resistance IS RANGE 1 TO 10E9
  UNITS
    ohm;
    Kohm＝1000 ohm;
END UNITS;
```

注意：fs、ps、ns、us、ms、sec、min、hr 是 VHDL 预定义的标识符，不能再用作用户自定义的标识符。

Note warning error failure false true 是 VHDL 预定义的标识符，不能再用作用户自定义的标识符。所有 VHDL 预定义的标识符都不能再用作用户自定义的标识符。

4. 数组类型

(1) 数组类型的说明格式为：

TYPE 标识符 IS ARRAY（下标）OF 元素的数据类型名；

其中，“()”中表示“维（下标 index)”的数据类型和“维”的范围。“维”的数据类型可以是整数类型或枚举类型。其范围可以是简单的范围说明或子类型标记。子类型标记又可以是只含有数据类型名和跟有范围限制的数据类型名。关键字 OF 后面的元素（element type）可为任何数据类型。

IEEE 1076.6　1999 综合标准定义了综合工具能支持的数据类型。

(2) 数组类型的说明举例。VHDL 语言中预定义的几个数组数据类型：

1)TYPE String IS ARRAY(Positive RANGE <>)OF Character;

元素的数据类型为 Character

一维数组，非限定数组

整数类型，子类型标记，跟有范围限制的类型名

其中，“<>”是下标范围的占位符，在今后用到该数组类型时再填入具体的数值范围。

2）TYPE Bit_Vector IS ARRAY（Natural RANGE <>）OF Bit；

VHDL 语言中用户自定义的几个数组数据类型的例子：

1）TYPE word IS ARRAY(31 DOWNTO 0)OF Bit；

一维数组，限定数组　　元素的数据类型为 Bit

整数类型，简单的范围说明

2）TYPE severity_level_status IS ARRAY(Note TO Failure)OF Integer；

枚举类型，简单的范围说明

3）TYPE row IS RANGE 1 TO 24；

4）TYPE column IS RANGE 1 TO 80；

5）TYPE matrix IS ARRAY(row，column)OF Boolean；

二维数组，限定数组

整数类型，子类型标记只含有类型名

6）TYPE reduced_matrix IS ARRAY（row RANGE 1 TO 10，
column RANGE 1 TO 40)OF Boolean；

二维数组，限定数组

整数类型，跟有范围限制的类型名

7）TYPE clock_level IS（low，rising，high，falling）；

8）TYPE conversion_array IS ARRAY（clock_level）OF Bit；

数组数据类型从维数还可分为一维、二维、多维，从元素数目是否固定还可分为非限定性数组、限定性数组。

建议整个设计保持风格一致，永远选用保留字 DOWNTO 来说明向量。其理由是向量最高位的下标值最大，并且将处于向量的最左边。反之，如果使用保留字 TO 来说明向量，则向量最高位的下标值将为 0。

（3）具有数组类型的对象的说明方法为：

TYPE lpc_coef IS ARRAY(Natural RANGE <>)OF Real；

VARIABLE item:lpc_coef(7 DOWNTO 0)；

其中，lpc_coef（7 DOWNTO 0）是变量对象的数据类型，后面跟一个括号就可知这是一个非限定性数组，此时填入具体的范围。

上述变量对象的说明与如下说明方法等价：

SUBTYPE mxy IS lpc_coef（7 DOWNTO 0）；

VARIABLE item：mxy；

VHDL 可以用一种特殊的表示，即字符串文字，用以表示其元素类型是字符文字（而不是标识符文字）的枚举类型的一维数组的值。

```
SIGNAL abc:word;
                 ↓
              数据类型，对于限定性数组只写出其标识符即可。
CONSTANT two:conversion_array:=('0','1','1','0');   ——位置关联
                              :=(low=>'0',rising=>'1',high=>'1',fall-
                                ing=>'0');
                                                   ——名字关联
                              :=(low=>'0',high=>'1',rising=>'1',fall-
                                ing=>'0');
                              :=(low|falling=>'0',high|rising=>'1');
                              :=(high|rising=>'1',others=>'0');
                              :="0110";              ——字符串文字，无括号
```

说明常数对象的时候，一般情况下，同时给定其取值。括号中的部分称为“聚集”（aggregate），它是用“()”括起来的元素关联表，位置关联和名字关联不可混用，others 除外。

说明方式如下：

```
TYPE symbol IS('a',' t',' d',' h',digit,cr,error);
TYPE state IS RANGE 0 to 6;
TYPE transition_matrix IS ARRAY(state,symbol)of state;
CONSTANT next_state:transition_matrix:=
                                  (0=>('a'=> 1,others=>6),
                                   1=>('t'=> 2,others=>6),
                                   2=>('d'=> 3,'h'=> 5,others=>6),
                                   3=>(digit=> 4,others=> 6),
                                   4=>(digit=> 4,cr=> 0,others=> 6),
                                   5=>(cr=> 0,others=> 6),
                                   6=>(cr=> 0,others=> 6));
```

下面的写法与上面的常数说明相同：

```
CONSTANT next_state:transition_matrix:=((1,6,6,6,6,6,6),
                                        (6,2,6,6,6,6,6),
                                        (6,6,3,5,6,6,6),
                                        (6,6,6,6,4,6,6),
                                        (6,6,6,6,4,0,6),
                                        (6,6,6,6,6,0,6),
                                        (6,6,6,6,6,0,6)
                                       );
```

（4）访问具有数组类型的对象的方法。①对其整体访问。访问整个数组时只需引用它的名字；②访问数组中的一个元素时，则要引用加下标的名字；③使用片名可以对一维数组中元素的相邻子集进行访问（对多维数组不构成片名）。

片名就是一维数组名后附加上一个括在括号中的离散范围，片名的类型与基数组的类型

相同。

```
TYPE Bit_Vector IS ARRAY(Natural RANGE <>)OF Bit;
SIGNAL stress:Bit_Vector(4 DOWNTO 0);
```

访问一个对象的整体：

```
stress<=('0','1','0','0','0');
stress<=(4=>'0',3=> '1',2=>'0',1=>'0',0=>'0');
stress<=(4=>'0',3=> '1',OTHERS=>'0');
stress<="01000";
```

访问单个元素：

```
stress(4)<='0';
```

用片名的方式访问：

```
stress(4 DOWNTO 2)<=('0','1','0');   ——片的方向必须与说明的方向相同
stress(4)<='0';                      —— 一个元素的片
```

说明一个数据类型为 conversion_array 的 three 变量，并给出了几种访问方法。

```
VARIABLE three:conversion_array;
three:=('0','1','0','0');          ——整体访问
three(rising):='1';                ——单个元素
three(rising TO high):="10";       ——片名
```

5. 记录类型

记录类型数据用于描述 SCSI 总线及通信协议，适用于仿真。在生成逻辑电路时，应将它们分解开来。

说明 rational 为一个记录数据类型。

```
TYPE rational IS RECORD
  numerator:Integer;
  denominator:Integer;
END RECORD;
```

上述数据类型的说明中，numerator 和 denominator 称为记录的元素名（域名）。

下面首先说明一个枚举数据类型 month _ name，然后说明了一个记录数据类型 date，在 date 的说明过程中使用了枚举数据类型 month _ name。最后说明了一个数据类型为 date 的变量 today，并给出了几种访问变量的方法。

```
TYPE month_name IS(Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec);
TYPE date IS RECORD
  day:Integer RANGE 1 TO 31;
  month:month_name;
  year:Integer RANGE 0 TO 3000;
END RECORD;
VARIABLE today:date;
today:=(15,May,1995);          ——聚集,整体赋值,位置关联,位置不可随便改动
today.day:=15;                 ——分别赋值,指定域名
```

```
today.month:=May;
today.year:=1995;
today:=(month=>May,day=>15,year=>1995);      ——聚集,名字关联,与位置无关
```

第四节　基本的 VHDL 的并行语句和串行语句

一、基本的 VHDL 并行语句

（一）进程语句（process statement）

进程语句是最基本的 VHDL 并行语句。下面是一个描述 3 进制计数器的进程。

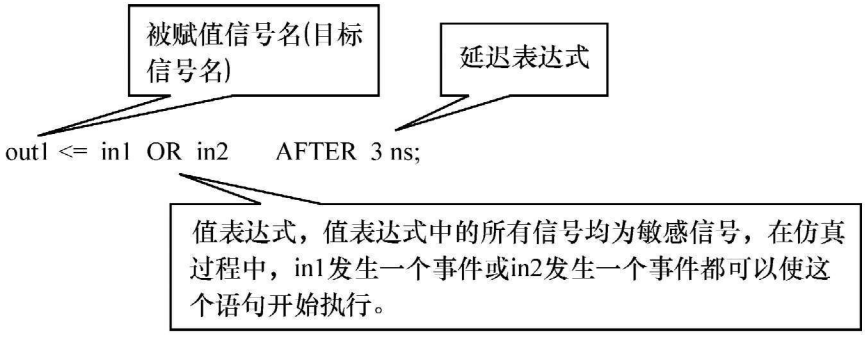
```
PROCESS (clk,clr)
            ——说明区，定义该进程所需的局部数据环境，只适用于该进程。可以说明
数据类型、子程序、变量和常量，但是不能说明信号。
    BEGIN
        ——以下为顺序语句
    IF clr='1' THEN
        q <="000";
    ELSIF clk='1' AND clk'Event THEN
        IF q="010" THEN
            q <="000";
        ELSE
            q <=q+ "001";——必须打开 STD_LOGIC_UNSIGNED 包才行。因为
                          ——此包中定义了对 STD_LOGIC_VECTOR 类型的加法，
                          ——其他的包中没有定义。
        END IF;
    END IF;
END PROCESS;
```

（二）并行信号赋值语句（concurrent signal assignment）

1. 并行信号赋值语句的基本形式

并行信号赋值语句有基本形式的并行信号赋值语句、条件形式的并行信号赋值语句和选择形式的并行信号赋值语句三种形式。

目标信号<=波形表达式



上述语句可以等效于：

```
PROCESS(in1,in2)
    BEGIN
        out1 <=in1 OR in2 AFTER 3 ns;——注意：串行信号赋值语句
                    ——赋值语句的右边都是敏感信号，而且只有一个赋值语句，
                    ——这时进程语句可以写成简单的并行信号赋值语句。
END PROCESS;
```

也等效于如下描述：

```
PROCESS
    BEGIN
        out1 <=in1 OR in2 AFTER 3 ns; ——注意：串行信号赋值语句。
    WAIT ON in1, in2;
END PROCESS;
```

注意，“<=”为信号赋值符号，信号赋值必须有延迟，即便在信号赋值语句中不写入延迟表达式，信号赋值仍然有延迟，此时信号赋值隐含一个无限小的延迟，即“δ”延迟。注意信号与变量赋值区别。

2. 条件信号赋值语句

条件信号赋值语句（conditional signal assignment statement）的形式为

```
ARCHITECTURE data_flow OF xor_gate IS
    BEGIN
      c <='0' WHEN a='0' AND b='0'ELSE
            '1' WHEN a='0' AND b='1'ELSE
            '1' WHEN a='1' AND b='0'ELSE
            '0' WHEN a='1' AND b='1'ELSE
            '0';
END ARCHITECTURE data_flow;
```

以下是使用此语句时需要注意的几个问题：

(1) 条件表达式中的信号为敏感信号，此例中，当 a 或 b 有事件发生时，语句被执行；

(2) WHEN 条件可以是一个简单表达式，也可以是一个值；

(3) 前面的条件满足后执行对应的语句，整个语句结束，后面的条件不再判断；

(4) 前面的条件不满足，接着判断后面的条件是否满足；

(5) 条件有优先级别问题，前面的条件优先级别高；

(6) 没有必要列出所有的信号取值组合；

(7) WHEN 条件可以指定任何简单的表达式，不一定是互不相关值；

(8) IEEE VHDL—1987 标准必须以 ELSE 结束，IEEE L—1993 标准没有此限制；

(9) 条件表达式中可以使用几个不同的信号，使得该命令在设计时非常灵活和有用。

3. 选择信号赋值语句

选择信号赋值语句（selected signal assignment statement）的形式为

```
WITH sel SELECT
```

```
C<=‘0’AFTER 5 ns WHEN “00”|“11”,——注意为逗号
      ‘1’AFTER 5 ns WHEN “01”|“10”;
```

必须是常数，在仿真的过程中必须有确定的值，不能发生变化。

其中，sel 为此选择信号赋值语句的敏感信号，当 sel 有事件发生时，此语句被仿真软件执行。以下是使用此语句时需要注意的几个问题：

(1) WHEN 条件列出所有的信号取值组合；

(2) 无优先级别之分；

(3) 选择一个合适的值，整个语句结束；

(4) WHEN 条件列举的必须是选择信号的互不相关的值；

(5) WHEN 条件只能为一个常数值，而不能是表达式。

(三) 元件例化语句 (component instantiation statement)

图 8-15 是一个由两个 2 输入与门构成一个 3 输入与门的逻辑电路。

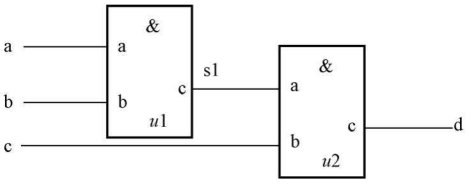


图 8-15　两个 2 输入与门构成一个 3 输入与门的逻辑电路

下面的 VHDL 程序用元件例化的方法描述了这个电路。

```
ENTITY and2 IS
  GENERIC(tpd:Time:=5 ns);
  PORT(a,b:IN Bit;c:OUT Bit);
END ENTITY and2;
ARCHITECTURE only OF and2 IS
    BEGIN
      c<=a AND b AFTER tpd;
END ARCHITECTURE only;
ENTITY and3 IS
    PORT(a,b,c:IN Bit;d:OUT Bit);
END ENTITY and3;
ARCHITECTURE struct OF and3 IS
——从此处开始，以下为结构体的说明区
COMPONENT and2 IS          —— 模板元件说明语句
                 ↓
                 模板元件名,可以与实体名不同
  GENERIC(tpd:Time:=5 ns);
  PORT(a,b:IN Bit;c:OUT Bit);          ——a,b,c 称局部端口
END COMPONENT and2;
FOR u1,u2:and2 USE ENTITY Work.and2(only);   —— 配置指定语句
      ↓    ↓                ↓    ↓
      例元 模板元件名        库   实体名
                                  结构体名,如果此处不指定结构体,则
                                  为默认的最新编译的结构体
```

```
SIGNAL s1:Bit;      ——信号 s1 无模式，只在本结构体内部使用，为局部信号
                    隐含了一个初始值，数据类型的左端值
——以上为结构体的说明区
BEGIN
u1:and2 GENERIC MAP(6 ns);      ——元件例化语句，使例元 u1，u2 与模板元
          ↓                          件 and2 发生了对应关系。
        类属映射
         PORT MAP(a,b,s1);      ——a,b,s1 称为实参
          ↓
        端口映射,位置关联,也可以使用名字关联
        PORT MAP(a=>a,b=>b,c=>s1)   =>左边为虚参,右边为实参
u2:and2 PORT MAP(s1,c,d);       ——无类属映射,取其默认值
END ARCHITECTURE struct;
```

注意：模板元件说明语句可以在实体说明的说明区、结构体说明区和程序包说明区来进行，为后面的元件例化做准备。在 VHDL 语言中，不能直接引用一个设计实体，而应将其说明为一个模板元件，然后引用模板元件。它不是实际可执行的语句，只是一个说明语句。

配置指定语句在设计过程的某些步骤上，设计者可能希望进一步把模板元件明确化，即指明该模板元件与哪一个设计库中的哪一个设计实体以及该设计实体所选择的哪一个结构体相对应。配置指定语句是用来完成此任务的方法之一，另外还有默认连接、配置说明、直接例化三种方式。配置较多用于模拟，较少用于综合。

元件例化语句使例元 u1、u2 与模板元件 and2 发生了对应关系。

二、基本的 VHDL 串行语句

（一）变量赋值语句

（1）:=为变量赋值符号，给任何对象赋初值也使用此符号。

（2）变量说明及其赋值只能在顺序域（进程、函数和过程）中进行。

（3）从模拟的角度看，立即得到其新值，不同于信号赋值。

（4）进程中的变量和函数、过程中的变量有所区别。进程中的变量只在模拟器初始化时初始化一次，而函数、过程中的变量在每次调用都被重新初始化。

（5）变量赋值只是一种纯软件效应，不可能产生与之对应的硬件结构。变量赋值的特性是 VHDL 语法的要求，是行为仿真流程的规定。实际上，在某些条件下变量赋值行为与信号赋值行为所产生的硬件结果是相同的。

（二）串行信号赋值语句

在 PROCESS 进程和子程序中的信号赋值语句为串行信号赋值语句。下面语句中的信号赋值语句 y<=a AND b 是串行信号赋值语句。

```
PROCESS(a,b)
   BEGIN
      y<=a AND b;
END PROCESS and2_behavior;
```

（三）流程控制语句

流程控制语句通过条件控制开关决定是否执行一条或几条语句，或重复执行一条或几条

语句，或跳过一条或几条语句。流程控制语句共有五种，此处只介绍 IF 语句和 CASE 语句两种。

IF 语句中至少应有一个条件句，条件句必须由 BOOLEAN 表达式构成。IF 语句根据条件句产生的判断结果 TRUE 或 FALSE，有条件的选择执行其后的顺序语句。

条件句有优先级别的区别，前面的优先级别高，后面的优先级别低，特别适合于描述优先编码器。

1. IF 语句

IF 语句是一种条件语句，根据语句中所设置的一种或多种条件，有选择地执行指定的顺序语句。IF 语句的语句结构有以下三种：

```
(1) IF 条件句 THEN
    顺序语句;
    END IF;
(2) IF 条件句 THEN
    顺序语句;
    ELSE
    顺序语句;
    END IF;
(3) IF 条件句 THEN
    顺序语句;
    ELSIF 条件句 THEN
    顺序语句;
        …
          ELSE
              顺序语句;
          END IF;
```

2. CASE 语句

CASE 语句的结构为：

```
CASE 表达式 IS
    WHEN 选择值=> 顺序语句;
    WHEN 选择值=> 顺序语句;
      ……
END CASE;
```

根据满足的条件直接选择多项顺序语句中一项执行。注意，口述句中的“=>”不是操作符，只是表示对应关系。当执行到 CASE 语句时，首先计算表达式的值，然后根据条件句中与之相同的选择值，执行对应的顺序语句，最后结束 CASE 语句。表达式可以是一个整数类型或枚举类型的值，也可以由这些数据类型的值构成的数组。顺序语句可以为较复杂的串行语句。

选择值可以有四种不同的表达形式：①单个普通值，如 4 ；②数值选择范围，如（2 TO 4），表示取值为 2、3 或 4；③并列数值，如 3 | 4，表示取值为 3 或者 4；④混合方式，

以上三种方式的混合。

注意：条件句中的选择值必须在表达式的取值范围内。除非所有条件句中的选择值能完全覆盖 CASE 语句中表达式的取值，否则最后一个条件句的选择必须用 OTHERS 表示，它代表已给的所有条件句中未能列出的其他可能的取值。OTHERS 只能出现一次，且只能作为最后一个条件取值。使用 OTHERS 的目的是为了使条件句中的所有选择值能涵盖表达式的所有取值，以免综合器会插入不必要的锁存器。这一点对于定义为 STD _ LOGIC 和 STD _ LOGIC _ VECTOR 的数据类型的值尤为重要，因为这些数据类型的值除了‘1’和‘0’以外，还可能有其他的值。

CASE 语句中的每一条件句的选择值只能出现一次，不能有相同选择值的条件语句出现。CASE 语句执行中必须选中，且只能选中所有条件句中的一条，这表明 CASE 语句中至少要包含一个条件语句。

第五节　基本硬件电路模块的 VHDL 模型

1. 二输入与门的 VHDL 描述

```
ENTITY and2_gate IS
    PORT(a,b:IN Bit;y:OUT Bit);
END and2_gate;
ARCHITECTURE basic OF and2_gate IS
    BEGIN
      PROCESS(a,b)
          BEGIN
              y <=a AND b;
      END PROCESS and2_behavior;
END basic;
```

2. 8 线—3 线优先编码器的 VHDL 描述

```
LIBRARY IEEE;
USE IEEE. Std_logic_1164. ALL;
ENTITY priority_encoder IS
  PORT (input:IN Std_logic_vector(0 TO 7);
         output:OUT Std_logic_vector(0 TO 2)      ——(2 DOWNTO 0)也可以。
        );
END priority_encoder;
ARCHITECTURE priority_encode_arch OF priority_encoder IS
    BEGIN
     PROCESS(input)
      BEGIN
        IF(input(7)='0')THEN
          output <="000";
```

```
        ELSIF(input(6)='0')THEN
          output <="100";
        ELSIF(input(5)='0')THEN
          output <="010";
        ELSIF(input(4)='0')THEN
          output <="110";
        ELSIF(input(3)='0')THEN
          output <="001";
        ELSIF(input(2)='0')THEN
          output <="101";
        ELSIF(input(1)='0')THEN
          output <="011";
      ELSE
        output <="111";
    END IF;
  END PROCESS;
END priority_encode_arch;
```

3. 三进制计数器的 VHDL 描述

```
LIBRARY IEEE;
USE IEEE. std_logic_1164. ALL;
USE IEEE. Std_logic_arith. ALL;
USE IEEE. Std_logic_misc. ALL;
USE IEEE. Std_logic_unsigned. ALL;
ENTITY cnt3 IS
  PORT(clk:IN Std_logic;clr:IN Std_logic;
        q:BUFFER Std_logic_vector(2 DOWNTO 0));
END cnt3;
ARCHITECTURE cnt3_arch OF cnt3 IS
 BEGIN
  PROCESS(clk,clr)
```

——此处为说明区，定义该进程所需的局部数据环境，只适用于该进程。可以说明数据类型、子程序、变量和常量，但是不能说明信号。

```
        BEGIN
                ——以下为顺序语句
          IF clr= '1' THEN
            q <= "000";
          ELSIF clk= '1' AND clk'Event THEN
            IF q = "010" THEN
                q <= "000";
```

```
            ELSE
            q<=q+ "001";    ——必须打开 Std _ logic _ unsigned 包才行。因为此包中定义了对 Std _ logic _ vector 类型的加法，其他的包中没有定义。
            END IF;
          END IF;
    END PROCESS;
  END cnt3 _ arch;
```

4. 七段译码器的 VHDL 描述

```
LIBRARY IEEE;
USE IEEE. std _ logic _ 1164. ALL;
ENTITY hex2led IS
 PORT (hex: in Std _ logic _ vector (3 downto 0);
        led: out Std _ logic _ vector (6 downto 0));
END hex2led;
ARCHITECTURE hex2led _ arch OF hex2led IS
   BEGIN
-- <<enter your statements here>>
--HEX-to-seven-segment decoder
--  hex: in Std _ logic _ vector (3 downto 0);
--  led: out Std _ logic _ vector (6 downto 0);
--
-- segment encoding
--      0
--      —
--  5 |   | 1
--      —     <- 6
--  4 |   | 2
--      —
--      3
  WITH hex SELECT
led<="0000110" WHEN "0001", --1 gfedcba, high level lighting,
      "1011011" WHEN "0010", --2
      "1001111" WHEN "0011", --3
      "1100110" WHEN "0100", --4
      "1101101" WHEN "0101", --5
      "1111101" WHEN "0110", --6
      "0000111" WHEN "0111", --7
      "1111111" WHEN "1000", --8
      "1101111" WHEN "1001", --9
```

```
        "1110111" WHEN "1010", --a
        "1111100" WHEN "1011", --b
        "0111001" WHEN "1100", --c
        "1011110" WHEN "1101", --d
        "1111001" WHEN "1110", --e
        "1110001" WHEN "1111", --f
        "0111111" WHEN OTHERS; --0
END hex2led _ arch;
```

小　　结

可编程逻辑器件的发展十分迅速，主要表现为三个方面：一是规模越来越大；二是速度越来越快；三是电路结构越来越灵活，电路资源更加丰富。在有些可编程逻辑器件中还集成了微处理器、数字信号处理单元和存储器等，这样，一个完整的数字系统甚至仅用一片可编程逻辑器件就可实现，即所谓的片上系统。

如果没有良好的开发软件的支持，对于用户来说，大容量可编程逻辑器件是无法应用的。

可编程逻辑器件按集成度分类，有低密度可编程逻辑器件和高密度可编程逻辑器件两大类；按结构特点，可分成 PLD 器件和 FPGA 器件两大类；按编程工艺，可分成熔丝或反熔丝编程工艺、UVEPROM 编程工艺、EEPROM 编程工艺、FLASH 编程工艺、SRAM 编程工艺等五种；按照制造工艺，可分为双极型和 MOS 型。

可编程逻辑器件中的资源很丰富，门电路的输入个数较多。与门、或门等电路单元的表示方法不再使用前面几章中的表示方法，而改用一种称为与或阵列图的比较紧凑的表示方法。

低密度可编程逻辑器件包括 PROM、PLA、PAL 和 GAL 器件，目前这种低密度可编程逻辑器件使用较少。高密度可编程逻辑器件包括 EPLD、CPLD 和 FPGA 三种，大致可以分为两类：一类是与标准门阵列结构类似的单元型 HDPLD，即 FPGA；另一类是基于与或阵列结构（或称为乘积项结构）的阵列扩展型 HDPLD，即 EPLD 和 CPLD，其中 CPLD 是 EPLD 的改进型器件。

硬件描述语言是硬件设计者和电子设计自动化工具之间的界面。设计者使用 HDL 来描述自己的设计方案（或设计要求、设计意图），并把这个描述告诉 EDA 工具，最后在 EDA 工具的帮助下进行详细设计和验证。VHDL 主要应用在描述、模拟的模型、综合的模型三个方面。通过一些常用的 VHDL 电路模型来了解 VHDL 的结构。

习　　题

8-1　分析图 8-16 所示的逻辑电路，写出输出逻辑函数表达式。

8-2　写出图 8-17 中 L 的逻辑表达式。

8-3　写出图 8-18 中 F_1、F_2 的逻辑表达式。

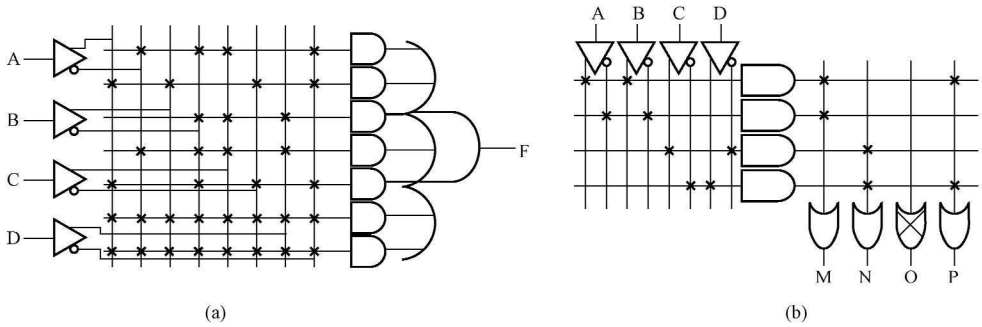


图 8-16 题 8-1 图

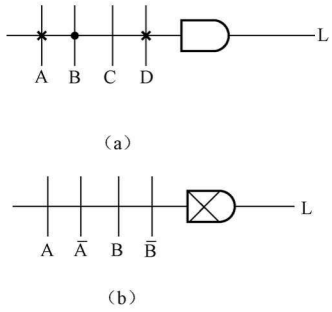


图 8-17 题 8-2 图

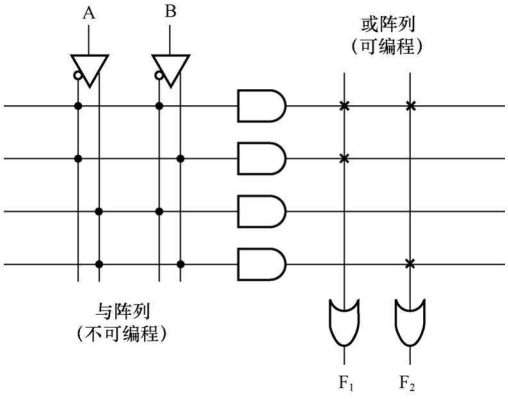


图 8-18 题 8-3 图

8-4 写出图 8-19 中 F_1、F_2、F_3、F_4、F_5、F_6 的逻辑表达式。

8-5 写出图 8-20 输出 F、P 的逻辑表达式。

8-6 用 PROM 阵列实现 1 位二进制全加器的逻辑功能，要求画出与或阵列图。

8-7 用 PLA 阵列实现 1 位二进制全加器的逻辑功能，要求画出与或阵列图。

8-8 用 PLA 和 D 触发器构成余 3 码十进制计数器，要求画出逻辑电路图。

8-9 用 PAL 和 JK 触发器构成 8421BCD 码十进制计数器，要求画出逻辑电路图。

8-10 用 PAL 实现 8421BCD 码转换成 Gray 码的电路，要求画出与或阵列图。

8-11 试写出 1 位二进制全加器的 VHDL 模型。

8-12 设计一个 3 线—8 线译码电路的 VHDL 模型，要求输出低电平有效。

8-13 有 8 个数码管依次排列，点亮方式为扫描方式，数码管高电平被点亮。当 $L_2L_1L_0=000$ 点亮最左边的数码管，当 $L_2L_1L_0=001$ 点亮从左边数第二个数码管。依此类推，当 $L_2L_1L_0=111$ 点亮最右边的数码管。试用 VHDL 设计此数码管扫描显示电路。提示：设计一个八进制计数器，状态从 000、001、010、011、100、101、110、111 依次循环。八进制计数器的输出送给 $L_2L_1L_0$，设计一个七段译码器，将 4 位二进制数转换成七段信号，依次在 8 个数码管上点亮。

8-14 用 VHDL 设计一个秒表电路，对外接口信号如图 8-21 所示。可以提供 1kHz 时钟信号、两个低电平有效的按钮、8 个扫描方式的高电平被点亮的数码管。要求如下：

(1) 最大记时时间为 59.9s，记时精度为 100ms；

（2）设置复位按钮和启停按钮，要求低电平有效；

（3）用 3 个数码管显示时间，显示方式为扫描方式，任一时刻只有一个数码管点亮；

（4）使用按钮去抖动电路，以消除抖动影响。

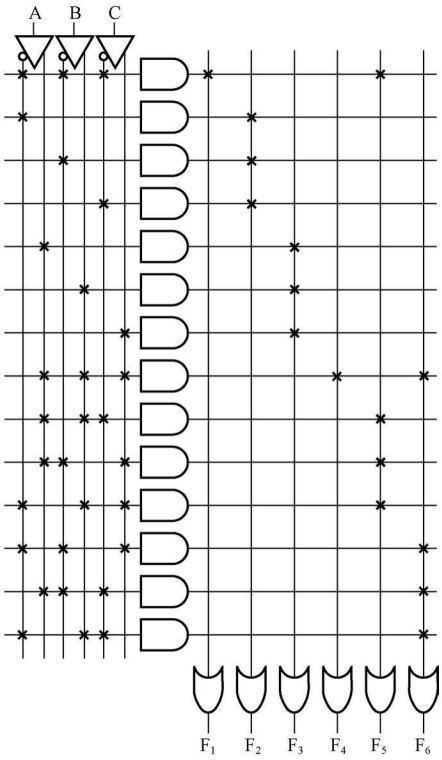


图 8-19　题 8-4 图

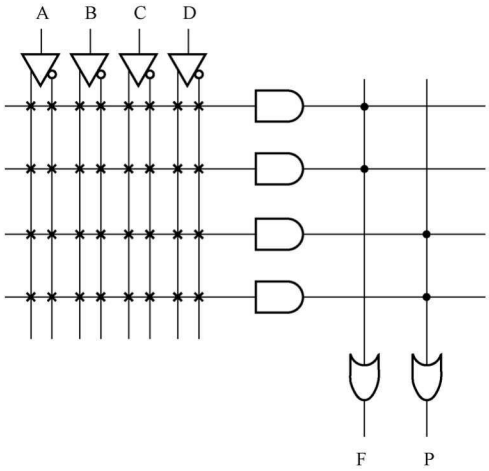


图 8-20　题 8-5 图

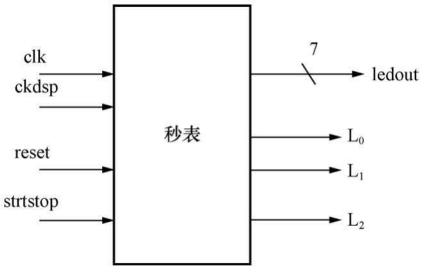


图 8-21　题 8-14 图

第九章　数模与模数转换电路

本　章　提　要

本章系统讲授数/模转换器和模/数转换器的基本原理和常见的典型电路。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 转换器；能将数字信号转换成模拟信号的电路称为数/模转换器，简称 D/A 转换器。A/D 转换器和 D/A 转换器已经成为计算机系统中不可缺少的接口电路。在数/模转换电路中，分别介绍了权电阻网络 D/A 转换、倒 T 型电阻网络 D/A 转换、权电流型 D/A 转换等几种转换原理；在模/数转换电路中，分别介绍了并行 A/D 转换、逐次逼近式 A/D 转换、双积分式 A/D 转换等几种转换原理。本章除介绍几种常用 A/D 转换器与 D/A 转换器的电路结构、工作原理外，还讨论了转换器的主要性能指标，介绍了常见集成芯片的功能、引脚及应用实例。

第一节　概　　述

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、通信和检测领域中，对信号的处理广泛采用了数字计算机技术。在工程实际应用中，需要处理的各种物理量（如温度、压力、流量、位移、图像、语音等），都是通过各种传感器转换得到连续的模拟电信号，只有将这些连续的模拟信号转换成数字信号才能使数字系统识别、传输、分析、处理。能将模拟信号转换成数字信号的电路，称为模/数转换器，简记为 ADC（Analog to Digital Converter）。而经计算机分析、处理后输出的数字量往往也需要将其转换成为相应的模拟信号才能被执行机构所接收。能将数字信号转换成模拟信号的电路称为数/模转换器，简记 DAC（Digital to Analog Converter）。

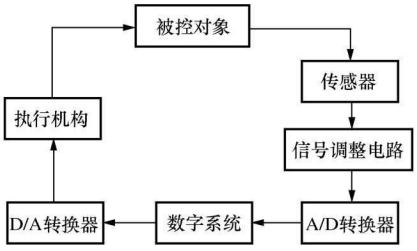


图 9-1　典型数控系统结构框图

典型的数控系统结构框图如图 9-1 所示。图中被控对象可以是温度、压力、流量、速度、位移、图像的亮度等。这些连续变化的模拟量通过各种传感器变成电压、电流等电量，经信号调整电路把电信号调整到合适的电信号范围，满足后面电路输入的要求。再通过 A/D 转换变成数字信号，送到数字系统进行加工处理，处理后的数字信息再经 D/A 转换变成电信号模拟量，送到执行机构，对被控对象进行控制。可以看出，D/A 转换和 A/D 转换在数字系统中有着十分重要的地位。

D/A 转换器和 A/D 转换器是现代数字系统中不可缺少的数字电路与模拟电路的接口电路，它们的种类很多，并且 A/D 转换器中要用到 D/A 转换器，所以下面先讨论 D/A 转换器。

第二节　D/A　转　换　器

一、D/A 转换器的基本原理

数字量是用二进制数码按位数组合起来表示的，对于有权码，每位数码都有一定的权。为了将数字量转换成模拟量，必须将每位数码按其权的大小转换成相应的模拟量，然后将这些模拟量相加，即可得到与数字量成正比的总模拟量，从而实现了数/模转换。这就是构成 D/A 转换器的基本思路。

D/A 转换器的输入、输出关系框图如图 9-2 所示。图中 $D_0 \sim D_{n-1}$ 是输入的 n 位二进制数，u_O 是与输入二进制数成比例的输出电压。

一个输入为 3 位二进制数时 D/A 转换器的转换关系如图 9-3 所示，它具体而形象地反映了 D/A 转换器的基本功能。

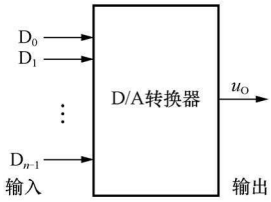


图 9-2　D/A 转换器的输入、输出关系框图

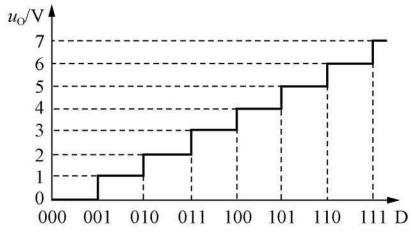


图 9-3　3 位 D/A 转换器的转换关系

D/A 转换器的原理框图如图 9-4 所示，它主要由五部分组成，即数码寄存器、n 位模拟开关、解码网络、求和电路和参考电压。数码寄存器用来暂时存放输入的数字量 D，寄存器的输入可以是并行输入也可以是串行输入，但输出只能是并行输出。寄存器的输出分别控制几个模拟开关的接通和断开，每个模拟开关相当于一个单刀双掷开关，它们分别与解码网络相连。解码网络中的几个支路权重不同，当输入数字量为 1 时，相应的支路接通，如果输入数字量为 0，相应支路断开，通过求和电路，从而实现二进制数的加权求和，转换成相应的十进制数。

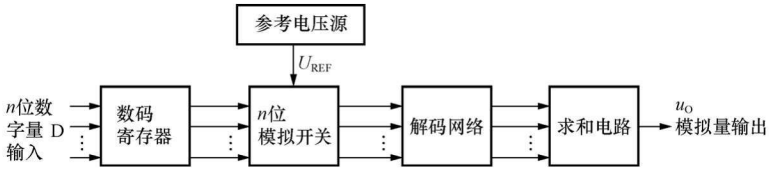


图 9-4　D/A 转换器的原理框图

二、几种常见的 D/A 转换器

根据解码网络的不同，D/A 转换器可分为不同类型，常见的有权电阻网络 D/A 转换器、倒 T 型电阻网络 D/A 转换器、权电流型 D/A 转换器等。

（一）权电阻网络 D/A 转换器

4 位二进制权电阻网络 D/A 转换器的原理如图 9-5 所示。由图中可知，该电路由四部分组成：基准电压源 U_{REF}，权电阻网络，模拟开关，求和放大电路。权电阻网络中各位权

电阻的阻值与权值相对应，电子开关受输入的各位数字信号控制。当某位数字量为 1 时，对应的开关 S 接通 U_{REF}；当数字量为 0 时，开关 S 接通地电位。求和放大器 A 把各个支路电流相加，通过反馈电阻 R_f 转换为输出模拟电压 u_O，显然 u_O 与输入数字量有关。

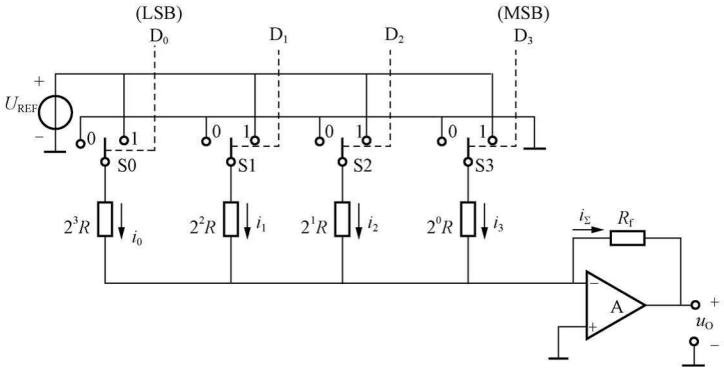


图 9-5 权电阻 D/A 转换器电路原理图

运放工作在线性状态，所以有虚短、虚断的特点，流入运放输入端的电流为 0，反相输入端有虚地。根据 KCL 电流定律可以写出

$$
\begin{aligned}
i_{\Sigma} &= i_0 + i_1 + i_2 + i_3 = D_0 \frac{U_{REF}}{2^3 R} + D_1 \frac{U_{REF}}{2^2 R} + D_2 \frac{U_{REF}}{2^1 R} + D_3 \frac{U_{REF}}{2^0 R} \\
&= \frac{U_{REF}}{R}\left(\frac{D_0}{2^3} + \frac{D_1}{2^2} + \frac{D_2}{2^1} + \frac{D_3}{2^0}\right) \qquad (9-1) \\
&= \frac{U_{REF}}{2^3 \times R}\sum_{i=0}^{3}(D_i \times 2^i)
\end{aligned}
$$

$$
\begin{aligned}
u_O &= -i_{\Sigma} R_f \\
&= -\frac{R_f}{R}\frac{U_{REF}}{2^3}\sum_{i=0}^{3}(D_i \times 2^i) \qquad (9-2)
\end{aligned}
$$

n 位数字量就有 n 个模拟开关和 n 个权电阻与其对应，构成 n 位二进制权电阻 DAC，其权电阻分别为 R，$2R$，$4R$，$8R$，…，$2^{n-1}R$，则有

$$u_O = -\frac{R_f}{R}\frac{U_{REF}}{2^{n-1}}\sum_{i=0}^{n-1}(D_i \times 2^i) \qquad (9-3)$$

当反馈电阻 $R_F = R/2$，则输出电压的计算公式可写成

$$u_O = -\frac{U_{REF}}{2^n}\sum_{i=0}^{n-1}(D_i \times 2^i) \qquad (9-4)$$

上式表明，输出的模拟电压正比于输入的数字量 D_n，从而实现了数字量到模拟量的转换。

权电阻 D/A 转换器电路十分简单，但是当数字量的位数增多时，权电阻数目增多，阻值范围越来越大，权电阻阻值的种类太多，集成电路制造、权电阻的匹配都比较困难，使权电阻 D/A 转换器的转换精度受到了限制。

（二）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，又称 $R-2R$

网络型 D/A 转换器。

倒 T 形电阻网络 D/A 转换器的原理图如图 9-6 所示。S0～S3 为模拟开关，$R-2R$ 电阻解码网络呈倒 T 形，运算放大器 A 构成求和电路，外接 R_f 引入负反馈，根据运放电路的“虚短”、“虚断”概念，反相输入端为地电位点（“虚地”点）。Si 由输入数码 D_i 控制，当 $D_i=1$ 时，Si 接运放反相输入端（“虚地”），I_i 流入求和电路；当 $D_i=0$ 时，Si 将电阻 $2R$ 接地，I_i 不流入求和电路。

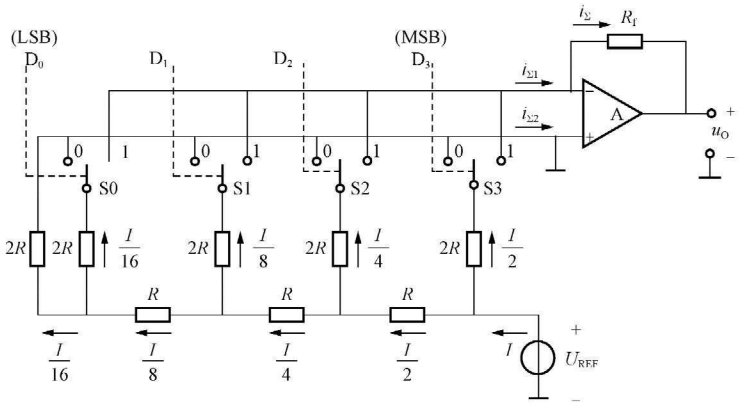


图 9-6　倒 T 形电阻网络 D/A 转换器电路原理图

无论模拟开关 Si 处于何种位置，与 Si 相连的 $2R$ 电阻均等效接“地”（地或虚地）。这样流经某一 $2R$ 电阻的电流与开关位置无关，为确定值。

分析 $R-2R$ 电阻解码网络不难发现，从每个接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 的整倍数递减。设由基准电压源提供的总电流为 $I(I=U_{REF}/R)$，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 $I/16$。

运放工作在线性状态，所以有虚短、虚断的特点，流入运放输入端的电流为 0，反相输入端为虚地。根据 KCL 电流定律可以得到总电流的公式为

$$i_{\Sigma}=i_{\Sigma 1}=\frac{U_{REF}}{R}\left(\frac{D_0}{2^4}+\frac{D_1}{2^3}+\frac{D_2}{2^2}+\frac{D_3}{2^1}\right)$$

$$=\frac{U_{REF}}{2^4\times R}\sum_{i=0}^{3}(D_i\times 2^i) \tag{9-5}$$

则输出电压为

$$u_O=-i_{\Sigma}R_f$$

$$=-\frac{R_f}{R}\frac{U_{REF}}{2^4}\sum_{i=0}^{3}(D_i\times 2^i) \tag{9-6}$$

将输入数字量扩展到 n 位，可得 n 位倒 T 型电阻网络 D/A 转换器输出模拟量与输入数字量之间的一般关系式为

$$u_O=-\frac{R_f}{R}\frac{U_{REF}}{2^n}\left[\sum_{i=0}^{n-1}(D_i\times 2^i)\right] \tag{9-7}$$

设　$K=\frac{R_f}{R}\frac{U_{REF}}{2^n}$，$N_B$ 表示括号中的 n 位二进制数，则

$$u_O = -KN_B \tag{9-8}$$

电路中的模拟开关通常有 CMOS 开关和双极性开关两种形式，下面简单介绍一下 CMOS 开关的基本工作原理。

简化的 CMOS 开关电路如图 9-7 所示。VT1～VT7 构成两个互为倒相的 CMOS 反相器，两个反相器的输出分别控制 VT8 和 VT9 的栅极，VT8 和 VT9 的漏极同时接电阻网络中的一个 $2R$ 电阻，而源极分别接两个电流输出端 $I_{\Sigma 1}$ 和 $I_{\Sigma 2}$。当输入端 D_i 为高电平时，VT4 和 VT5 组成的反相器输出高电平，VT6 和 VT7 组成的反相器输出低电平，结果使 VT9 导通，VT8 截止，由 VT9 将电流引向 $I_{\Sigma 1}$，在图 9-6 中开关接到运放的反相输入端。而当 D_i 为低电平时，则 VT8 导通，VT9 截止，将电流引向 $I_{\Sigma 2}$，在图 9-6 中开关接到运放的同相输入端。

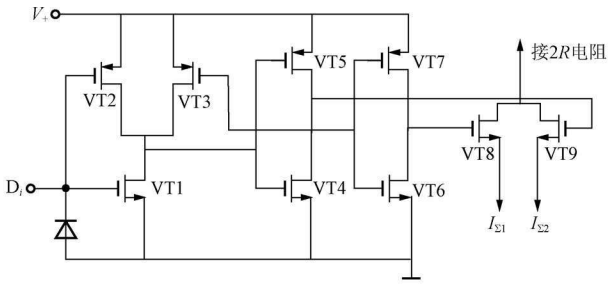


图 9-7　CMOS 开关电路

为了保证 D/A 转换的精度，电阻网络中的 R 和 $2R$ 电阻之比的精度要高，同时，每个开关上的电压降要尽量相等。由于从高位到低位的电流按 2 的整数倍递减，这就要求对应的开关导通电阻要按 2 的整数倍递减。如果图 9-6 中 $U_{REF}=10V$，流过开关 S3 的电流为 0.5mA，流过开关 S2 的电流为 0.25mA。若要求每个开关的电压降为 10mV，则 S3 的导通电阻为 20Ω，S2 的导通电阻为 40Ω，其他开关的导通电阻可依此类推。

总之，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 $2R$ 电阻的比值精度要高；③每个模拟开关的开关电压降要相等；④为实现电流从高位到低位按 2 的整倍数递减，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。这一特点不仅提高了转换速度，而且也减少了动态过程中输出端可能出现的尖脉冲。倒 T 形电阻网络 D/A 转换器是目前广泛使用的 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）等。

（三）权电流型 D/A 转换器

尽管倒 T 型电阻网络 D/A 转换器具有较高的转换速度，实际上电路中的模拟开关有一定的导通电阻和导通压降，而且每个开关的情况又不完全相同，它们的存在无疑将引起转换误差，影响转换精度。为进一步提高 D/A 转换器的转换精度，可采用权电流型 D/A 转换器。

权电流型 D/A 转换器的原理图见图 9-8。与图 9-6 相比，倒 T 型电阻网络被一组恒

流源取代，每个恒流源电流的大小从高位到低位依次为 $I/2$、$I/4$、$I/8$、$I/16$，与倒 T 型电阻网络每一个节点电流减一半的效果相同，与二进制对应的位权成正比。由于采用了恒流源，每个支路电流的大小不再受开关内阻和导通压降的影响，从而降低了对开关电路的要求。

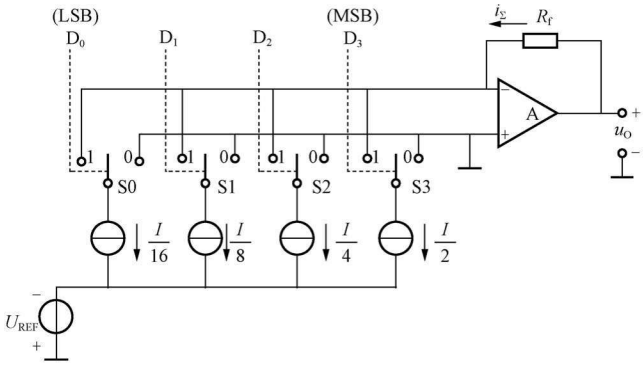


图 9-8　权电流型 D/A 转换器电路原理图

当输入数字量的某一位代码 $D_i=1$ 时，开关 Si 接运算放大器的反相输入端，相应的权电流流入求和电路；当 $D_i=0$ 时，开关 Si 接地。分析该电路可得出

$$
\begin{aligned}
u_O &= i_\Sigma R_f \\
&= R_f\left(\frac{I}{2}D_3+\frac{I}{4}D_2+\frac{I}{8}D_1+\frac{I}{16}D_0\right) \\
&= \frac{I}{2^4}R_f(D_3\times 2^3+D_2\times 2^2+D_1\times 2^1+D_0\times 2^0) \\
&= \frac{I}{2^4}R_f\sum_{i=0}^{3}D_i\times 2^i
\end{aligned}
\tag{9-9}
$$

可见，输出电压正比于输入的数字量。采用了恒流源电路之后，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。

具有电流负反馈的 BJT 恒流源电路的权电流 D/A 转换器的实际电路如图 9-9 所示。由图可见，VT3～VT0 的基极是接在一起的，而三极管的发射结压降 U_{BE} 相同，则它们的发射极具有相同的电位，在计算各支路的电流时，可以认为所有 $2R$ 电阻的上端都接到了同一个电位上，因而电路的工作状态与倒 T 型电阻网络的工作状态一样。这时流过每个 $2R$ 电阻的电流自左而右依次减少 1/2。为了保证所有三极管的发射结压降相等，在发射极电流较大的三极管中按比例加大了发射结的面积，即 VT3～VT0 发射结面积之比为 8∶4∶2∶1。在图中 VT3～VT0 均采用了多发射极晶体管，其发射极个数分别是 8、4、2、1。这样，在各 BJT 电流比值为 8∶4∶2∶1的情况下，VT3～VT0 的发射极电流密度相等，结面积的比值8∶4∶2∶1，可使各发射结电压 U_{BE} 相同。由于 VT3～VT0 的基极电位相同，所以它们的发射极 e3、e2、e1、e0 就为等电位点。在计算各支路电流时将它们等效连接后，流入每个 $2R$ 电阻的电流从高位到低位依次减少 1/2，各支路中电流分配比例满足 8∶4∶2∶1 的要求。

参考电压源 U_{REF}、运算放大器 A2 和 R_1、VTr、R、V_{EE} 组成基准电流 I_{REF} 产生电路。A2 和 R_1、VTr 的集电结组成电压并联负反馈电路，以稳定输出电压（即 VTr 的基极电

压)。VTr 的集电结、电阻 R 到 $-V_{EE}$ 为反馈电路的负载。由于电路处于深度负反馈，根据虚短的原理，其基准电流为

$$I = I_{REF} = \frac{U_{REF}}{R_1} = 2I_{E3}$$

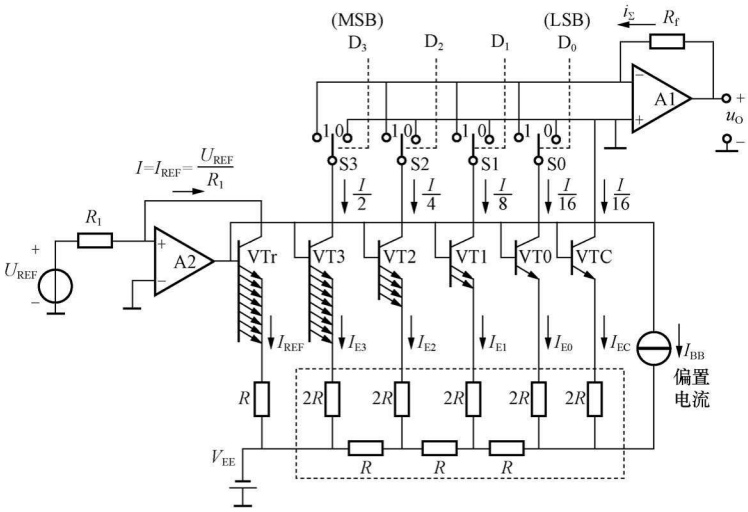


图 9-9 权电流型 D/A 转换器的实际电路

由倒 T 形电阻网络分析可知，$I_{E3}=I/2$，$I_{E2}=I/4$，$I_{E1}=I/8$，$I_{E0}=I/16$，于是可得输出电压为

$$\begin{aligned} u_O &= i_\Sigma R_f \\ &= \frac{R_f U_{REF}}{2^4 R_1}(D_3 \times 2^3 + D_2 \times 2^2 + D_1 \times 2^1 + D_0 \times 2^0) \end{aligned} \tag{9-10}$$

可推得 n 位倒 T 形权电流 D/A 转换器的输出电压为

$$u_O = \frac{U_{REF}}{R_1}\frac{R_f}{2^n}\sum_{i=0}^{n-1} D_i \times 2^i \tag{9-11}$$

该电路特点为，基准电流仅与基准电压 U_{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。

由于在这种权电流型 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。采用这种权电流型 D/A 转换电路生产的单片集成 D/A 转换器有 AD1408、DAC0806、DAC0808 型等。这些器件都采用双极型工艺制作，工作速度较高。

三、D/A 转换器的输出方式

D/A 转换器的输出方式有单极性和双极性两种。单极性输出时，D/A 转换器输入的数字量只是数值，没有符号，即无符号二进制码，输出的电压只有正或者只有负。在实际应用中，D/A 转换器输入的数字量有正数也有负数，这就要求 D/A 转换器能将正数、负数分别对应转换成正、负极性的模拟电压，即工作于双极性方式。

（一）单极性输出方式

D/A 转换器采用单极性输出方式时，假设输入数字量采用的是无符号二进制码，其中 MSB 是最高位，LSB 是最低位，输入的数字量与模拟电压输出量之间的关系见表 9-1。

表 9-1　　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	$\pm U_{REF}$ (255/256)
			⋮					⋮
1	0	0	0	0	0	0	1	$\pm U_{REF}$ (129/256)
1	0	0	0	0	0	0	0	$\pm U_{REF}$ (128/256)
0	1	1	1	1	1	1	1	$\pm U_{REF}$ (127/256)
			⋮					⋮
0	0	0	0	0	0	0	1	$\pm U_{REF}$ (1/256)
0	0	0	0	0	0	0	0	$\pm U_{REF}$ (0/256)

倒 T 型电阻网络 D/A 转换器单极性输出电路如图 9-10 所示。如果要输出正的电压，将参考电压接为负值，或者将上面的电路再接一级电压反相放大即可。

（二）双极性电压输出

因为在数字系统中通常把带符号的数值表示为补码的形式，因此要求 D/A 转换器能够把补码形式的输入数字量转换成正负极性的模拟电压输出。

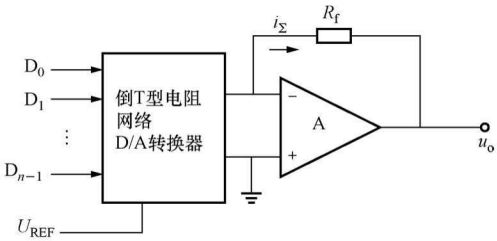


图 9-10　倒 T 型电阻网络 D/A 转换器的单极性电压输出电路

以输入为 3 位二进制补码的情况为例，说明转换的原理。3 位二进制补码（最高位是符号位）可以表示从＋3 到－4 之间的任何整数，它们与十进制数的对应关系及希望得到的输出模拟电压见表 9-2。

偏移二进制码是 D/A 转换过程中用道的一类编码。偏移二进制码本身没有符号，其形式上与无符号二进制码相同，但是对应的输出电压有所不同，假设无符号二进制码对应的输出的电压为正，则其与偏移二进制码的对应关系见表 9-3。

表 9-2　　输入为 3 位二进制补码时要求 D/A 转换器的输出

补码输入			对应的十进制数	要求的输出电压（V）
D_2	D_1	D_0		
0	1	1	＋3	＋3
0	1	0	＋2	＋2
0	0	1	＋1	＋1
0	0	0	0	0
1	1	1	－1	－1
1	1	0	－2	－2
1	0	1	－3	－3
1	0	0	－4	－4

表 9-3　　无符号二进制码与偏移二进制码的对应关系

3 位无符号二进制码		3 位偏移二进制码	
$D_2D_1D_0$	对应的电压（V）	$D_2D_1D_0$	对应的电压（V）
111	＋7	111	＋3
110	＋6	110	＋2
101	＋5	101	＋1
100	＋4	100	0
011	＋3	011	－1
010	＋2	010	－2
001	＋1	001	－3
000	0	000	－4

从表 9-3 可以看出，将 3 位无符号二进制码对应的电压减去 4V 后的电压就是 3 位偏移二进制码对应的电压。比较表 9-2 和表 9-3 可知，3 位二进制补码和 3 位偏移二进制码的关系是最高位相反。

用倒 T 型电阻网络 D/A 转换实现表 9-2 补码转换关系的电路如图 9-11 所示。在图 9-11 中，U_{REF}取 8V，$D_2D_1D_0$ 为 3 位二进制补码，D_2 反相后为 D'_2，$D'_2D_1D_0$ 为 3 位偏移二进制码，$D'_2D_1D_0$ 被倒 T 型电阻网络和运放 A1 转换成对应的电压 u_{O1}，此电压必须减去 4V 才是作为 $D'_2D_1D_0$ 作为偏移二进制码对应的输出，运放 A2 构成的反相加法电路正好能满足此要求。

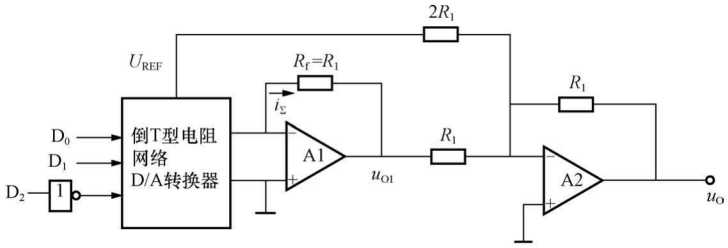


图 9-11 具有双极性输出电压的 D/A 转换器

假设输入 $D_2D_1D_0=010$，由图 9-11 可知 D/A 转换器的输入为 110，转换结果

$$u_{O1}=-\frac{R_f}{R}\cdot\frac{U_{REF}}{2^n}\left[\sum_{i=0}^{n-1}(D_i\cdot 2^i)\right]=-\frac{8}{2^3}\left[\sum_{i=0}^{2}(D_i\cdot 2^i)\right]=-6(V)$$

运放 A2 构成反相加法，即

$$u_O=-\left(\frac{R_1}{2R_1}U_{REF}+\frac{R_1}{R_1}u_{O1}\right)=-\left(u_{O1}+\frac{U_{REF}}{2}\right)=-u_{O1}-\frac{U_{REF}}{2}$$

$$=-u_{O1}-4=6-4=2(V)$$

实现表 9-2 中输入 010（补码形式）对应的输出电压是+2V。

假设输入 $D_2D_1D_0=111$，由图 9-11 可知 D/A 转换器的输入为 011，转换结果

$$u_{O1}=-\frac{R_f}{R}\cdot\frac{U_{REF}}{2^n}\left[\sum_{i=0}^{n-1}(D_i\cdot 2^i)\right]=-\frac{8}{2^3}\left[\sum_{i=0}^{2}(D_i\cdot 2^i)\right]=-3(V)$$

$$u_O=-\left(u_{O1}+\frac{U_{REF}}{2}\right)=3-4=-1(V)$$

实现表 9-2 中输入 111（补码形式）对应的输出电压是−1V。

依此类推，可以得到如表 9-2 对应的数字输入量和输出量的关系。

四、D/A 转换器的主要性能指标

D/A 转换器的特性参数很多，主要技术指标有转换精度和转换速度。

（一）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。

1. 分辨率

分辨率用于表征 D/A 转换器对输入微小量变化的敏感程度，是 D/A 转换器在理论上可以达到的精度。

输入数字量位数越多，输出电压可分离的等级越多，即分辨率越高。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的最小输出电压（此时输入的数字代码只有最低有效位为 1，其余各位都是 0）与最大输出电压（此时输入的数字代码各有效位全为 1）之比给出。n 位 D/A 转换器的分辨率为$\frac{1}{2^n-1}$，表示

D/A 转换器在理论上可以达到的精度。

2. 转换误差

D/A 转换器在稳态工作时，实际的输出电流或电压值与理想数值之差定义为转换误差。

由于转换器中各元件参数值存在误差，基准电源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 转换器的实际输出与理想输出值有一些误差。转换误差常用满量程 FSR（Full Scale Range）的百分数表示，有时也用最低有效位 LSB（Least Significant Bit）的倍数来表示。

D/A 转换误差是零点（失调）误差、增益误差、线性误差、微分线性误差等的综合。

零点（失调）误差是由运算放大器的零点漂移引起的，其大小与输入数字量无关，该误差使转移特性曲线发生平移，如图 9-12（a）所示。实际输出特性曲线上每一点都与理想数值有一个固定误差，零点误差可以通过调零的方法消除，但环境温度改变时，会产生新的零点误差。

增益误差是输出特性曲线的斜率与理想输出特性曲线的斜率之比，如图 9-12（b）所示。图中的实际输出特性曲线上的每一个点与理想输出特性曲线均相差同一百分数，输出数值越大，误差的绝对值也越大。增益误差可以通过初始增益调节误差，但环境温度变化会引起新的增益误差。

线性误差表示的是实际的 D/A 转换特性和理想转换特性之间的最大偏差，如图 9-12（c）所示。没有零点误差和增益误差的 D/A 转换器，其输出数值在一条直线上，转换关系是线性的。如果 D/A 的输出数值不在一条直线上，则称其与理想输出特性曲线的误差值为线性误差。理想的 D/A 转换特性是连接坐标原点和满量程输出（输入全为 1）理论值的一条直线。与实际可能的 D/A 转换特性相比较，线性误差反映的是实际输出特性的整体线性度，即与理想输出特性的偏离程度。

微分线性误差反映了线性误差在整个输出特性中的分布。任意两个相邻输入数据所对应的输出差值与 1LSB 之差称为该点的微分线性误差。在图 9-13 中，a、b 两点为实际输出值，该处的微分线性误差为 3.5LSB－1.5LSB－1LSB＝1LSB。确定 D/A 的最大微分线性误差，只需分别求出 n 个进位处的微分线性误差，然后找出最大值即可。

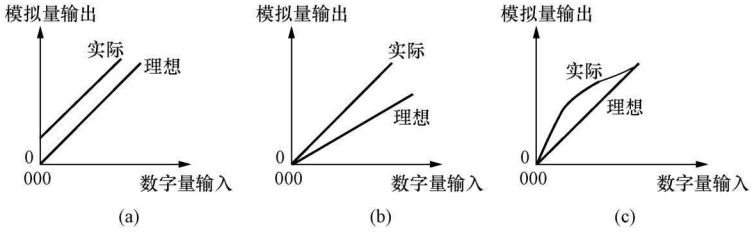


图 9-12　几种转换误差的示意图

（a）零点（失调）误差；（b）增益误差；（c）线性误差

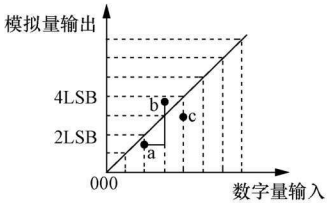


图 9-13　D/A 的微分线性误差

当输入数据单调增加时，输出电压或电流增加或不变，则 D/A 转换器是单调的，否则是非单调的。如果微分线性误差小于 1LSB，认为转换器具有单调性。在自动控制系统中使用的 D/A 转换器必须具有单调性，否则可能使系统在 D/A 转换器的非单调区间内来回摆

动，形成振荡，不能稳定工作。

D/A 转换器的绝对误差（或绝对精度）是指输入端加入最大数字量（全 1）时，D/A 转换器的理论值与实际值之差。该误差值应低于 LSB/2。

例如，一个 8 位的 D/A 转换器，对应最大数字量 $(FF)_H$ 的模拟理论输出值为$\frac{255}{256}U_{REF}$，$\frac{1}{2}LSB=\frac{1}{512}U_{REF}$，所以实际值不应超过$\left(\frac{255}{256}\pm\frac{1}{512}\right)U_{REF}$。

（二）转换速度

当 D/A 转换器输入的数字量发生变化时，输出的模拟量并不立即达到所对应的数值，需要一段时间。通常用建立时间来描述 D/A 转换器的转换速度。

建立时间（t_{set}）是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从全 0 变为全 1 时，输出电压达到规定的误差范围(±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 0.1μs 以内。

五、集成 D/A 转换器及其应用

D/A 转换器在实际电路中应用很广，不仅常作为接口电路用于微机系统，还可以利用其电路结构特征和输入、输出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。

单片集成 D/A 转换器产品的种类繁多，性能指标各异，下面以 AD7520 为例介绍集成 D/A 转换器的结构及其应用。

（一）AD7520 引脚及使用说明

AD7520 是 10 位 CMOS 电流开关型 D/A 转换器，芯片内含有倒 T 型电阻网络、CMOS 电流开关和反馈电阻（$R=10k\Omega$）。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。其内部电阻网络组成的 D/A 转换电路如图 9-14 所示，AD7520 的引脚图如图 9-15 所示，其 CMOS 开关电路如图 9-7 所示。

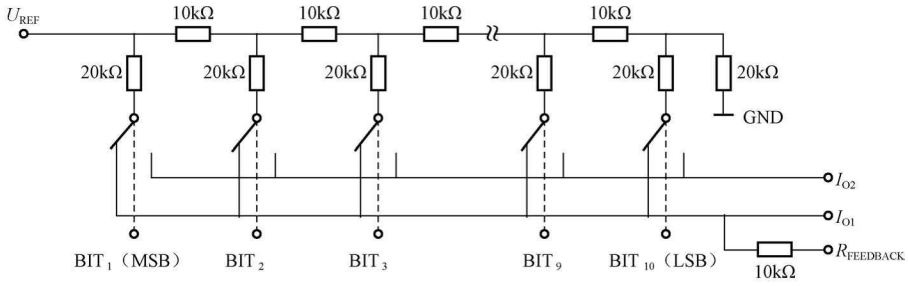


图 9-14 AD7520 的内部倒 T 型电阻网络转换电路

（二）AD7520 的输出方式

AD7520 的输出量是电流，如果要实现电压输出，需要接输出电路将电流转换成电压。图 9-16 所示电路是 AD7520 的单极性输出方式，输入的数字量与输出模拟量之间的关系如表 9-4 所示，其中 $N=10$，$LSB=2^{-N}U_{REF}$。

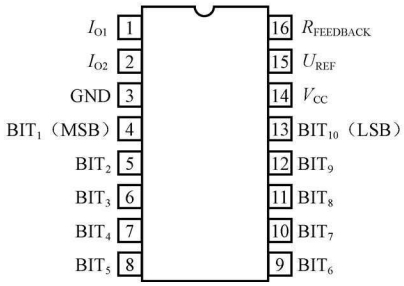


图 9-15　AD7520 的引脚图

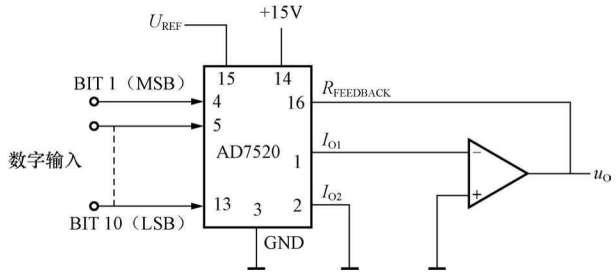


图 9-16　AD7520 的单极性输出方式的电路图

表 9-4　AD7520 的单极性输出时的输入/输出关系

数字输入	模拟输出	数字输入	模拟输出	数字输入	模拟输出
1111111111	$-U_{REF}(1-2^{-N})$	1000000000	$-U_{REF}/2$	0000000001	$-U_{REF}(2^{-N})$
1000000001	$-U_{REF}\left(\frac{1}{2}+2^{-N}\right)$	0111111111	$-U_{REF}\left(\frac{1}{2}-2^{-N}\right)$	0000000000	0

图 9-17 为 AD7520 的双极性输出方式的电路图。输入输出关系如表 9-5 所示，其中，$N=10$，$LSB=2^{-(N-1)}U_{REF}$。

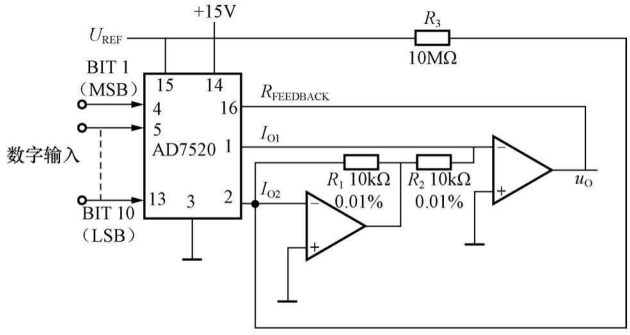


图 9-17　AD7520 的双极性输出方式的电路图

表 9-5　AD7520 双极性输出时输入/输出的关系

数字输入	模拟输出
1111111111	$-U_{REF}(1-2^{-(N-1)})$
1000000001	$-U_{REF}(2^{-(N-1)})$
1000000000	0
0111111111	$U_{REF}(2^{-(N-1)})$
0000000001	$U_{REF}(1-2^{-(N-1)})$
0000000000	U_{REF}

（三）AD7520 的应用

1. 数字式可编程增益控制电路

数字式可编程控制电路如图 9-18 所示。电路中运算放大器接成普通的反相比例放大形式，输出为单极性形式，u_I 接入 AD7520 的反馈电阻的输入端，输出接到参考电压源 U_{REF}。AD7520 属于倒 T 型电阻网络 A/D 转换器，根据运算放大器虚地原理，可以得到

$$u_I/R=-i_{\Sigma}$$
$$=\frac{-U_{REF}}{2^{10}R}(D_9\times 2^9+\cdots+D_2\times 2^2+D_1\times 2^1+D_0\times 2^0)$$

其中，$U_{REF}=u_O$，所以

$$A_u=u_O/u_I$$
$$=-2^{10}/(D_9\times 2^9+\cdots+D_2\times 2^2+D_1\times 2^1+D_0\times 2^0) \qquad (9-12)$$

通过改变输入的数字量而获得不同的输出电压，即实现了数字式可编程增益控制。

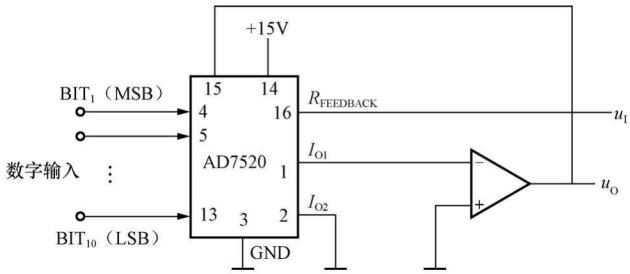


图 9-18 数字式可编程控制电路

2. 波形产生电路

由 D/A 转换器 AD7520、10 位可逆计数器和加减控制电路组成的波形产生电路如图 9-19 所示。

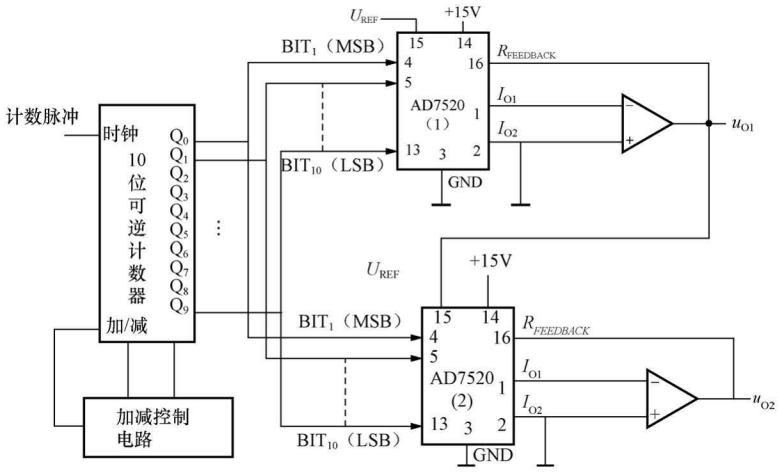


图 9-19 AD7520、10 位可逆计数器和加减控制电路组成的波形产生电路

加减控制电路与 10 位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；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。D/A 转换器的输出电压为

$$u_{O1} = -\frac{U_{REF}}{2^{10}}\sum_{i=0}^{9} D_i \times 2^i \tag{9-13}$$

可以看出，u_{O1}是一个近似的三角波。将这个三角波作为 D/A 转换器（2）的参考电压，可得第二级 D/A 转换器输出的模拟电压为

$$u_{O2} = U_{REF}\left(\sum_{i=0}^{9} D_i \times 2^i / 2^{10}\right)^2 \tag{9-14}$$

显然，该电压波形是一个抛物线波。

第三节 A/D 转 换 器

一、A/D 转换的一般工作过程

在 A/D 转换器中，因为输入的模拟信号在时间上是连续的量，而输出的数字信号代码

是离散量，所以进行转换时必须在一系列选定的瞬间（亦即时间坐标轴上的一些规定点上）对输入的模拟信号取样，然后再把这些取样值转换为输出的数字量。因此，一般的 A/D 转换过程是通过取样—保持和量化编码两部分电路完成的，如图 9-20 所示。

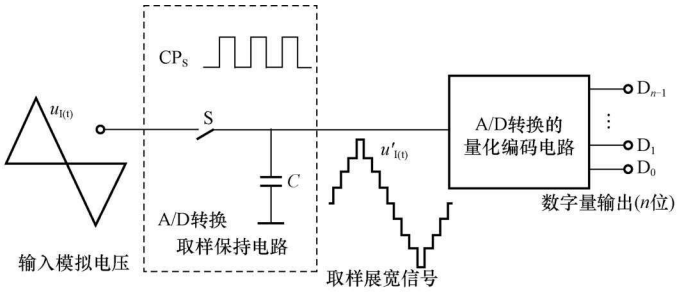


图 9-20　模拟量到数字量的转换过程

取样保持电路是由受控的理想模拟开关 S 与存储电容 C 组成，模拟开关由周期性的取样脉冲 CP_S 进行功能逻辑控制，将模拟量 $u_I(t)$ 转换成时间上离散的模拟量 $u'_I(t)$，然后将取样值暂时存储在电容器上，即将 $u_I(t)$ 转换成阶梯状的样值展宽信号 $u'_I(t)$，并将取样值保持到下一个取样脉冲到来之前，在这段保持的时间里，取样值通过模数转换电路数字化。

（一）取样定理

如图 9-21 所示，为了能正确无误地用取样信号 u_S 表示模拟输入信号 u_I，保证能从取样信号不失真地将原来的模拟信号 u_I 恢复出来，取样信号必须有足够高的频率。

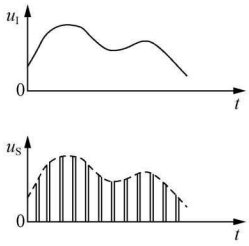


图 9-21　对输入模拟信号的取样

可以证明，开关的取样频率必须满足

$$f_S \geqslant 2f_{Imax} \tag{9-15}$$

式中：f_S 为取样频率；f_{Imax} 为输入模拟信号 u_I 的最高频率分量的频率。

式（9-15）就是所谓的取样定理。

因为每次把取样电压转换为相应的数字量都需要一定的时间，所以在每次取样以后，必须把取样电压保持一段时间。可见，进行 A/D 转换时所用的输入电压，实际上是每次取样结束时的 u_I 值。取样保持往往是通过取样保持电路完成的。

（二）取样保持电路

取样保持电路种类很多，在此只介绍一种最基本最简单的形式，其原理图如图 9-22（a）所示。电路由输入放大器 A1、输出放大器 A2、保持电容 C_H 和开关及驱动电路组成。电路中要求 A1 有很高的输入阻抗，以减少对输入信号的影响，并且输出电阻低，使电容充电时间常数小。为使保持阶段 C_H 上所存电荷不易泄放，A2 应具有较高的输入阻抗，还应具有低输出阻抗以提高带负载的能力，A1、A2 均工作在单位增益的电压跟随器状态，一般还要求 $A_{u1}A_{u2}=1$。

分析取样保持电路的工作原理。由图 9-22（b）图所示，在 $t=t_0$ 时刻，u_C 信号有效，开关 S 闭合，电容被迅速充电，由于 $A_{u1}A_{u2}=1$，所以 $u_O=u_I$，在 $t_0\sim t_1$ 时刻是取样阶段，

u_O 跟随 u_I。当 $t=t_1$ 时刻，S 断开，因为 A2 的输入阻抗很高，如果是无穷大，则电容上的电压无放电回路，虽然 S 断开了，但由于 C_H 上的电压不变，所以输出电压 u_O 的数值得以保持下来，即图中的 $t_1 \sim t_2$ 的平坦部分，就是保持阶段，保持的是取样最后时刻开关断开前一时刻的输入信号的值。

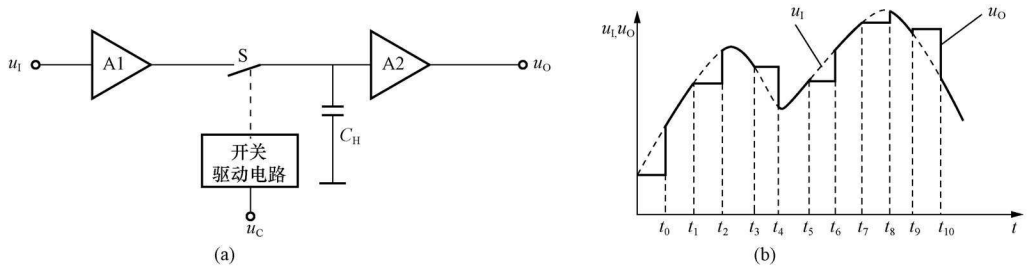


图 9-22 取样—保持电路的基本形式

（a）原理图；（b）波形图

（三）量化和编码

取样保持电路的输出虽然已成为阶梯状，但是其阶梯幅值仍连续可变、有无限多个数值，无法与 n 位有限的 2^n 个数字量输出 X 相对应，因此，必须将取样后的值只限于在某些规定个数的离散的电平上。凡介于两个离散电平之间的取样值就要用某种方式整理归并到这两个离散电平之一上。这种将幅值取整归并的方式及过程称为“量化”。

将量化后的有限个整量值用 n 位一组的某种二进制代码来描述以形成数字量。这种用数字代码表示量化幅值的过程称作“编码”。这个二进制代码就是 A/D 转换的输出信号。

规定的最小数量单位叫做量化单位，用 Δ 表示。显然，数字信号最低有效位中的 1 表示的数量大小，就等于 Δ。既然模拟电压是连续的，那么它就不一定能被 Δ 整除，因而不可避免地会引入误差，这种误差称为量化误差。在把模拟信号划分为不同的量化等级时，用不同的划分方法可以得到不同的量化误差。

对取样值进行量化的方法，一般有两种：舍尾取整法和四舍五入法。

假定需要把 0～+1V 的模拟电压信号转换成 3 位二进制代码，这时便可以取 Δ=1/8V，并规定凡是数值在 0～1/8V 之间的模拟电压都当作 0×Δ 看待，用二进制的 000 表示；凡数值在 1/8～2/8V 之间的模拟电压都当作 1×Δ 看待，用二进制的 001 表示，等等。模拟输入对应的二进制代码如图 9-23（a）所示。不难看出，最大的量化误差可达 Δ，即 1/8V。这就是舍尾取整法。

为了减少量化误差，通常采用图 9-23（b）所示的划分方法，取量化单位 Δ=2/15V，并将 000 代码所对应的模拟电压规定为 0～1/15V，即 0～Δ/2。这时，最大量化误差将减少为 Δ/2=1/15V。这个道理不难理解，因为现在把每个二进制代码所代表的模拟电压值规定为它所对应的模拟电压范围的中点，所以最大的量化误差自然就缩小为 Δ/2 了。这就是四舍五入法。

无论采用哪种量化方法，量化过程必然使被测输入值与量化值之间有误差，这二值之差称为量化误差。显然，四舍五入法比舍尾取整法的量化误差小。量化误差除了与量化方法有

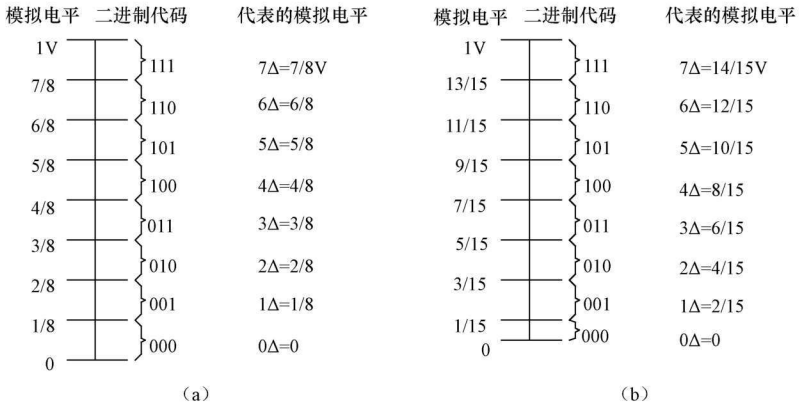


图 9-23　两种量化方法

(a) 舍尾取整法；(b) 四舍五入法

关，还与数字量的位数有关，位数越多，量化误差越小，转换精度越高。

二、A/D 转换器的基本工作原理

下面介绍并行比较型 A/D 转换、逐次比较型 A/D 转换、双积分型 A/D 转换等几种转换原理。

（一）并行比较型 A/D 转换器

3 位并行比较型 A/D 转换的原理电路如图 9-24 所示，它由电阻分压器、电压比较器、寄存器和优先编码器四部分组成。优先编码器输入为高电平有效。I_7、I_6、I_5、I_4、I_3、I_2、I_1、I_0 优先级别依次降低。当所有输入都无效时，输出 $D_3D_2D_1=000$。对 I_1 的编码为 $D_3D_2D_1=001$，对 I_2 的编码为 $D_3D_2D_1=010$，对 I_3 的编码为 $D_3D_2D_1=011$，对 I_4 的编码为 $D_3D_2D_1=100$，对 I_5 的编码为 $D_3D_2D_1=101$，对 I_6 的编码为 $D_3D_2D_1=110$，对 I_7 的编码为 $D_3D_2D_1=111$。

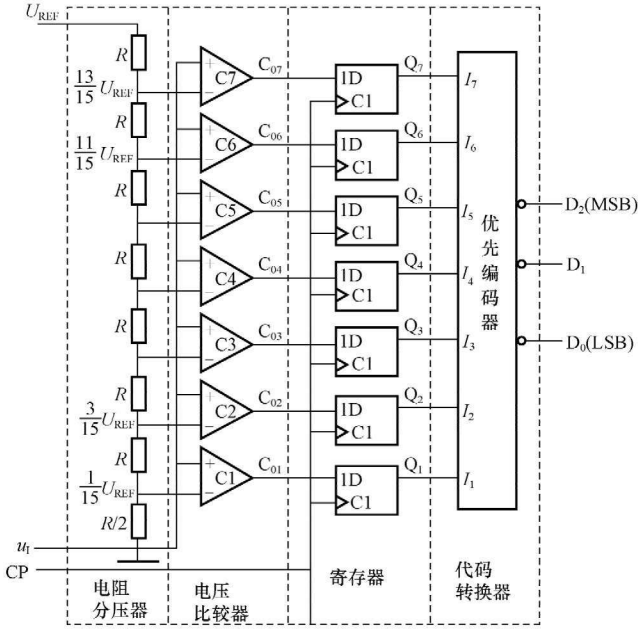


图 9-24　3 位并行比较型 A/D 转换器原理电路图

若输入模拟电压范围 $u_I=0\sim8V$，$U_{REF}=8V$。由图 9-24 可知，输出为 3 位二进制代码（$n=3$），采用的是图 9-23（b）所示的四舍五入法，利用电阻分压器把参考电压分成 8 段，得到从$\frac{1}{15}U_{REF}\sim\frac{13}{15}U_{REF}$之间 7 个比较电平，量化单位 $\Delta=\frac{2}{15}V$；然后，把这 7 个比较电平分别接到 7 个比较器 C1～C7 的输入端作为比较基准电压，将输入的模拟电压同时加到每个比较器的另一个输入端上，与这 7 个比较基准电压进行比较，当输入信号小于基准电压时，比较器输出逻辑 0，输入信号大于基准电压时，比较器输出逻辑 1，3 位 A/D 转换器需要 7 个比较器。若 A/D 转换器输出为 n 位二进制数，则需要 2^n-1 个比较器。

若输入电压 $u_I<(1/15)U_{REF}$，则所有比较器的输出全为低电平，CP 上升沿到来后寄存器中所有的触发器都被置为 0 状态。优先编码的输出结果为 000。

若输入电压$(1/15)U_{REF}<u_I<(3/15)U_{REF}$，则只有 C1 输出为高电平，CP 上升沿到来后，$Q_1=1$，寄存器中其余的触发器都被置为 0 状态。优先编码的输出结果为 001。

若输入电压$(3/15)U_{REF}<u_I<(5/15)U_{REF}$，则只有 C1、C2 输出为高电平，CP 上升沿到来后，$Q_1=Q_2=1$，寄存器中其余的触发器都被置为 0 状态。优先编码器对 Q_2 进行编码，其输出结果为 010。

依次类推，可得到 3 位并行 A/D 转换器输入与输出转换关系对照表，如表 9-6 所示。

表 9-6　　3 位并行 A/D 转换器输入与输出转换关系对照表

输入模拟电压 u_I	寄存器状态（代码转换器输入）							数字量输出（代码转换器输出）		
	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	D_2	D_1	D_0
$\left(0\sim\frac{1}{15}\right)U_{REF}$	0	0	0	0	0	0	0	0	0	0
$\left(\frac{1}{15}\sim\frac{3}{15}\right)U_{REF}$	0	0	0	0	0	0	1	0	0	1
$\left(\frac{3}{15}\sim\frac{5}{15}\right)U_{REF}$	0	0	0	0	0	1	1	0	1	0
$\left(\frac{5}{15}\sim\frac{7}{15}\right)U_{REF}$	0	0	0	0	1	1	1	0	1	1
$\left(\frac{7}{15}\sim\frac{9}{15}\right)U_{REF}$	0	0	0	1	1	1	1	1	0	0
$\left(\frac{9}{15}\sim\frac{11}{15}\right)U_{REF}$	0	0	1	1	1	1	1	1	0	1
$\left(\frac{11}{15}\sim\frac{13}{15}\right)U_{REF}$	0	1	1	1	1	1	1	1	1	0
$\left(\frac{13}{15}\sim1\right)U_{REF}$	1	1	1	1	1	1	1	1	1	1

单片集成并行比较型 A/D 转换器的产品较多，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，所用的比较器个数为 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）使用这种含有寄存器的并行 A/D 转换电路时，可以不用附加取样—保持电路，因为比较器和寄存器这两部分也兼有取样—保持功能。这也是该电路的一个优点。

（二）逐次比较型 A/D 转换器

逐次比较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，与被称物体进行比较，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。照此方法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 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的对应值。

N 位逐次比较型 A/D 转换器框图如图 9-25 所示。

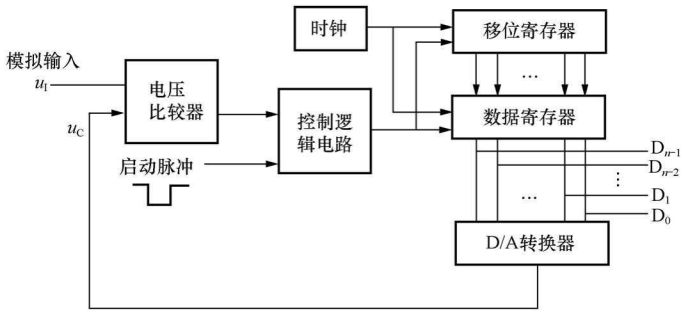


图 9-25　N 位逐次比较型 A/D 转换器框图

N 位逐次比较型 A/D 转换器由控制逻辑电路、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 转换器进行转换得到对应的模拟电压 u_C。然后把 u_C 与输入模拟电压进行比较，如果 $u_C<u_I$，说明这个数太小，应该保留这个 1，则最高位为 1；如果 $u_C>u_I$，说明这个数太大，应该去掉这个 1，则最高位为 0，比较结果存于数据寄存器的 D_{n-1} 位。在第二个 CP 作用下，移位寄存器的次高位置 1，其他位置为 0。如最高位已存 1，则此时将 1100…0 送入 A/D 转换器进行转换，转换结果与 u_I 进行比较，确定 D_{n-2} 位是 1 还是 0。依次类推，直到最低位的数值被确定，就完成了一次转换，这时寄存器输出的数码就是输入的模拟信号所对应得数字量。

为进一步理解逐次比较型 A/D 转换器的工作原理及转换过程，下面用实例加以说明。

【例 9-1】 有一个逐次比较型 8 位 A/D 转换器，时钟频率为 200kHz。设模拟输入 $u_I=4.115$V 和 8 位 D/A 转换器的单位量化电压 $\Delta V=0.022$V，并假定在顺序脉冲作用之前，已将寄存器的各位清零，所以电压比较器的输出为 1。分析 A/D 转换器的输出为多少？

解　电路工作过程如下：

第一步，当启动脉冲有效，顺序脉冲发生器中第一个脉冲上升沿到达后，将 8 位寄存器的最高位 Q_7 置 1，其他各位保持为 0，寄存器的状态为 10000000。这时 D/A 转换器输出 $u_C=128\Delta V=2.816$V，u_C 与 u_I 进行比较，由于 $u_C<u_I$，比较器输出为高电平，于是 Q_7 保留为 1。

第二步，顺序脉冲发生器中第二个脉冲的上升沿到达后，将 8 位寄存器的次高位 Q_6 置

1，寄存器的状态为 11000000，D/A 转换器输出 $u_C=192\Delta V=4.224V$，u_C 与 u_I 进行比较，由于 $u_C>u_I$，比较器输出为低电平，于是 Q_6 为 0。

第三步，顺序脉冲发生器中第三个脉冲的上升沿到达后，将 8 位寄存器的 Q_5 置 1，寄存器的状态为 10100000，D/A 转换器输出 $u_C=3.520V$，u_C 与 u_I 进行比较。由于 $u_C<u_I$，比较器输出为高电平，于是 Q_5 保留为 1。A/D 转换器逐位进行试探，直至最低位。

在 A/D 转换的过程中，各位寄存器的状态、D/A 转换器的输出电压 u_C、比较器的输出状态可分别由表 9-7 表示。D/A 转换器的输出波形如图 9-26 所示。

表 9-7　　逐次比较型 A/D 转换器的工作过程

顺序脉冲序数	寄存器状态 $Q_7Q_6Q_5Q_4Q_3Q_2Q_1Q_0$	D/A 转换输出 u_C(V)	比较器输出状态	该位留与否
1	10000000	2.816	1	留
2	11000000	4.224	0	舍
3	10100000	3.520	1	留
4	10110000	3.872	1	留
5	10111000	4.048	1	留
6	10111100	4.136	0	舍
7	10111010	4.092	1	留
8	10111011	4.114	1	留

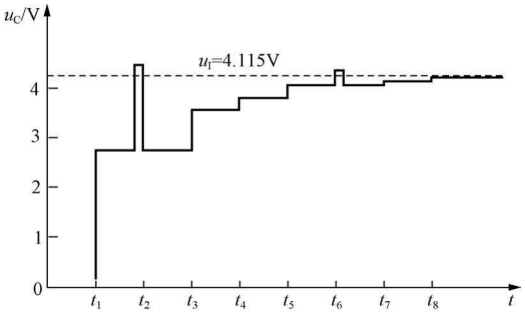


图 9-26　D/A 转换器输出电压 u_C 的波形

由表 9-7 和图 9-26 可以看出，A/D 转换器逐位比较的过程，就是 D/A 转换器输出电压逐渐逼近输入模拟电压的过程，也是寄存器数码逐步建立的过程。A/D 转换结束时，D/A 转换的输出电压与输入电压近似相等，其误差取决于转换器的位数。而寄存器状态所对应的数值与 D/A 转换器的输出电压成正比，也就是与输入电压成正比，从而实现了 A/D 转换。

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，位数越少，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，转换精度高的特点。

常用的集成逐次比较型 A/D 转换器有 ADC0808/0809 系列（8 位）、AD575（10 位）、AD574A（12 位）等。

（三）双积分型 A/D 转换器

双积分型 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转换，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。

双积分型 A/D 转换器的原理电路如图 9-27。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）和定时器/计数器（FF0～FFn）等几部分组成。

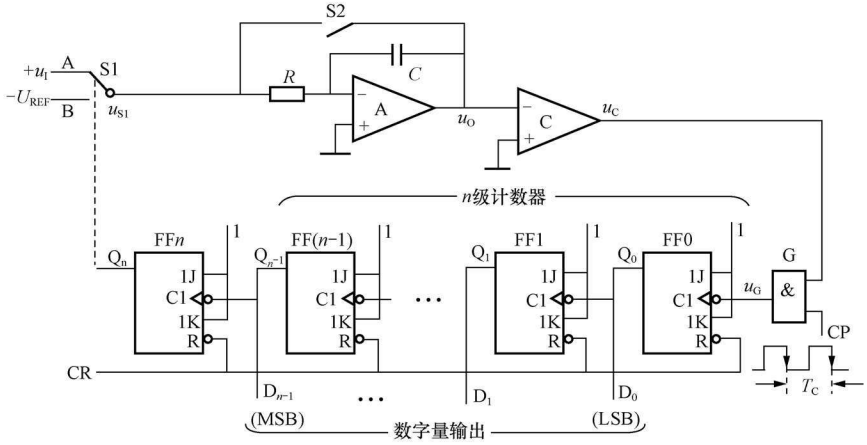


图 9-27　双积分型 A/D 转换器原理电路图

积分器是转换器的核心部分，它的输入端所接开关 S1 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 u_I 和参考电压 U_{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 $\tau=RC$。

过零比较器用来确定积分器输出电压 u_O 的过零时刻。当 $u_O\geqslant 0$ 时，比较器输出 u_C 为低电平；当 $u_O<0$ 时，u_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：由 $n+1$ 个接成计数型的触发器 FF0～FFn 串联组成。触发器FF0～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，FF0～FF（$n-1$）均回到 0 状态，而 FFn 反转为 1 态，$Q_n=1$ 后，开关 S1 从位置 A 转接到位置 B。

时钟脉冲控制门：时钟脉冲源标准周期为 T_C，作为测量时间间隔的标准时间。当 $u_C=1$ 时，与门打开，时钟脉冲通过与门加到触发器 FF0 的输入端。

下面以输入正极性的直流电压 u_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，工作波形如图 9-28 所示。

（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 S2 闭合，待积分电容放电完毕，再将 S2 断开。

（2）第一次积分阶段。在转换过程开始时（$t=0$），开关 S1 与 A 端接通，正的输入电压 u_I 加到积分器的输入端。积分器从 0V 开始对 u_I 积分，输出一个负斜波电压为

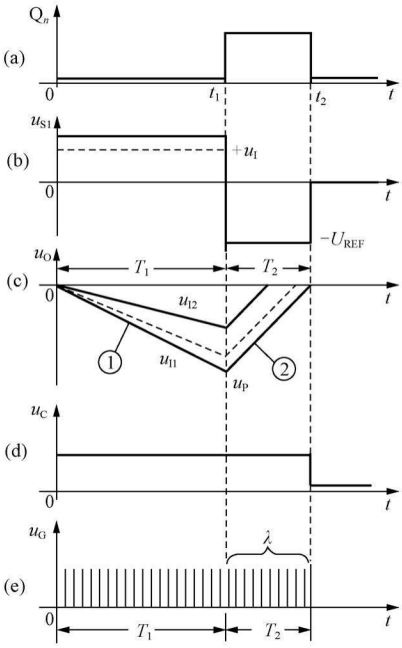


图 9-28　双积分型 A/D 转换器各点工作波形

$$u_O = -\frac{1}{\tau}\int_0^t u_I \mathrm{d}t \tag{9-16}$$

由于 $u_O<0V$，过零比较器输出端 u_C 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。经过 2^n 个时钟脉冲后，触发器 FF0～FF（$n-1$）都翻转到 0 态，而 $Q_n=1$，开关 S1 由 A 点转到 B 点，第一次积分结束。第一次积分时间为

$$t = T_1 = 2^n T_C \tag{9-17}$$

在第一次积分结束时积分器的输出电压 u_P 为

$$u_P = -\frac{T_1}{\tau}u_I = -\frac{2^n T_C}{\tau}u_I \tag{9-18}$$

（3）第二次积分阶段。当 $t=t_1$ 时，S1 转接到 B 点，具有与 u_I 相反极性的基准电压 $-U_{REF}$ 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 $u_O>0V$，比较器输出 $u_C=0$，时钟脉冲控制门 G 被关闭，计数停止。在此阶段结束时 u_O 的表达式可写为

$$u_O(t_2) = V_P - \frac{1}{\tau}\int_{t_1}^{t_2}(-U_{REF})\mathrm{d}t = 0 \tag{9-19}$$

设 $T_2=t_2-t_1$，于是有

$$\frac{U_{REF}T_2}{\tau} = \frac{2^n T_C}{\tau}u_I \tag{9-20}$$

设在此期间计数器所累计的时钟脉冲个数为 λ，则

$$T_2 = \lambda T_C$$

$$T_2 = \frac{2^n T_C}{U_{REF}}u_I \tag{9-21}$$

可见，T_2 与 u_I 成正比，T_2 就是双积分 A/D 转换过程的中间变量，所以

$$\lambda = \frac{T_2}{T_C} = \frac{2^n}{U_{REF}}u_I \tag{9-22}$$

上式表明，在计数器中所计得的数 $\lambda(\lambda=Q_{n-1}\cdots Q_1Q_0)$，与在取样时间 T_1 内输入电压的平均值 u_I 成正比。只要 $u_I<U_{REF}$，转换器就能将输入电压转换为数字量，并能从计数器读取转换结果。如果取 $U_{REF}=2^n$，则 $\lambda=u_I$，计数器所计的数在数值上就等于被测电压。

在第二次积分阶段结束后，控制电路又使开关 S2 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。

由图 9-28 可以看到，当 u_I 取不同值时，因为均需要 2^n 个时钟脉冲（计数器计满为止），所以反相积分的时间相同，只是反相积分结束时的输出电压不同。由于第二次积分的输入电压相同，都是 U_{REF}，所以充电电流相同，积分的斜率相同，但积分的时间不同。

双积分 A/D 转换器最突出的优点是工作性能比较稳定。由于转换过程中先后进行了两次积分，而且积分时间常数相同，转换结果与时间常数无关。因此 R、C 参数的缓慢变化不影响电路的转换精度，而且不要求 R、C 参数十分精确。此外，转换结果与时钟信号的周期无关，只要每次转换过程中 T_C 不变，那么时钟周期在长时间里发生缓慢的变化也不会带来转换误差。因此，完全可能用精度比较低的元器件制成精度很高的双积分型 A/D 转换器。

由于双积分 A/D 转换器在 T_1 时间内采集的是输入电压的平均值，因此具有很强的抗工频干扰能力。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，使输入信号正负变化时，仍有良好的抑制能力。在工业系统中经常碰到的是工频或工频的倍频干扰，故通常选定取样时间 T_1 总是等于工频电源周期的倍数，如 20ms 或 40ms 等。另一方面，由于在转换过程中，前后两次积分所采用的是同一积分器，因此，在两次积分期间（一般在几十毫秒至数百毫秒之间），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。

双积分型 A/D 转换器的主要缺点是工作速度低。如果采用图 9-27 所给出的方案，那么每完成一次转换的时间应取在 $2T_1$ 以上，即不应小于 $2^{n+1}T_C$。如果再加上转换前的准备时间和输出转换结果的时间，则完成一次转换所需要的时间还要长一些。尽管如此，但由于它的优点十分突出，所以在对转换速度要求不高的场合，双积分型 A/D 转换器用得非常广泛。

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$\left(3\frac{1}{2}\text{位，BCD 码}\right)$等。

三、A/D 转换的主要技术指标

（一）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

分辨率说明 A/D 转换器对输入信号的分辨能力。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。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 5V，那么这个转换器应能区分输入信号的最小电压为 19.53mV。

2. 转换误差

转换误差表示 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，给出相对误差不大于±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有效位的半个字。有时也用满量程输出的百分数给出转换误差。

A/D 转换器的转换误差已经综合反映了电路内部各个元件及单元电路偏差对转换精度的影响，所以不再分别讨论这些因素各自对转换精度的影响了。另外，工程手册上给出的转换精度都是在一定的电源电压和环境温度下得到的数据，如果这些条件变了，将引起附加的转换误差。因此为了获得较高的转换精度，必须保证供电电源有很好的稳定度，并限制环境温度的变化。对于那些需外加参考电压的 A/D 转换器，尤其需要保证参考电压应有的稳定度。

（二）转换速度

转换速度常用转换时间来表示。转换时间指 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。

不同类型的转换器转换速度相差甚远。并行比较型 A/D 转换器转换速度最高，8 位二进制输出的单片集成 A/D 转换器转换时间可达 50ns 以内。逐次比较型 A/D 转换器次之，它们多数转换时间在 10～50μs 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 9-2】 某信号采集系统要求用一片 A/D 转换集成芯片在 1s 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 0～0.025V（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，信号电压范围为 0～0.025V，分辨的温度为 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 16 次/s，取样时间为 62.5ms。对于这样慢的取样，任何一个 A/D 转换器都可以达到。选用带有取样—保持（S/H）的逐次比较型 A/D 转换器或不带 S/H 的双积分式 A/D 转换器均可。

四、集成 A/D 转换器

在单片集成 A/D 转换器中，逐次比较型使用较多，下面以 ADC0809 为例介绍集成A/D 转换器。

AD0809 是 CMOS 集成工艺制成的逐次比较型 A/D 转换器芯片。其分辨率为 8 位，转换时间为 100μs，输入电压范围为由参考电压 U_{REF}（+）和 U_{REF}（−）决定，即输入电压范围必须位于 U_{REF}（+）和 U_{REF}（−）之间。转换器主要有 8 路模拟开关、地址译码、256 个电阻组成的电阻阶梯和开关树、8 位逐次比较寄存器 SAR 及比较器等电路组成，其原理框图如图 9-29 所示。8 路模拟信号由 AD0809 的 IN0～IN7 端输入，由 ADDA～ADDC 的状态确定模拟输入通道。START 和 ALE 有效，A/D 转换启动，256 个电阻和开关树组成 DAC 电路，模拟开关受 8 位逐次比较寄存器的输出状态控制，D/A 的转换结果与输入模拟电压进行比较，比较结果再送入 8 位逐次比较寄存器，逐次比较的结果再控制开关树。如此不断进行比较，直到转换完最低位为止，A/D 转换完成，输出 EOC 转换结束信号有效，将 A/D 转换好的数字量输出。

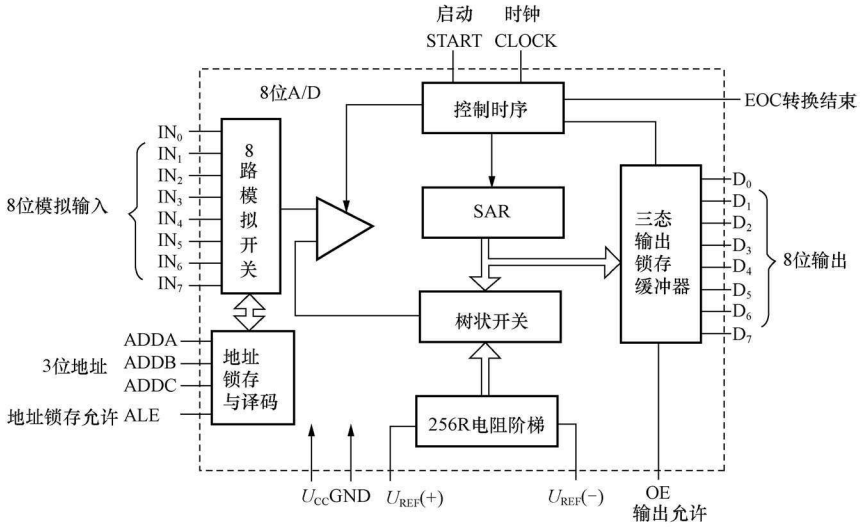


图 9-29 ADC0809 原理框图

ADC0809 采用 28 脚双列直插封装，引脚图如图 9-30 所示。ADC0809 引脚名称及意义如下：

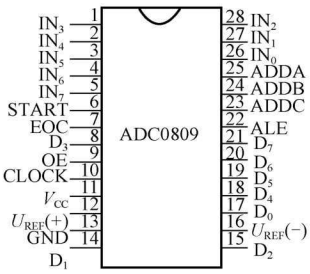


图 9-30　AD0809 引脚图

$IN_0 \sim IN_7$：ADC0809 的 8 路模拟信号输入端，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，能与微机总线相连接。

GND：地。

CLOCK：外电路提供时钟脉冲输入端，典型值为 640kHz，对应的转换时间为 100μs。

OE：数据输出允许端，高电平有效。它控制 ADC 内部三态输出锁存缓冲器。当 OE=0 时，各数字输出端均呈高阻态；当 OE=1 时，允许缓冲器中的数据输出。

EOC：转换结束信号，由 ADC 内部的控制逻辑电路产生。EOC=0 表示转换正在进行；EOC=1 表示转换已经结束。因此 EOC 可以作为微机的中断请求或查询信号。显然，只有当 EOC=1 以后，才可让 OE 上升为高电平，这时读出的数据才是正确的转换结果。

START：启动输入端，输入启动脉冲的下降沿使 ADC 开始转换，脉冲宽度要求大于 100ns。

ALE：通道地址锁存输入端。输入 ALE 脉冲上升沿使地址锁存器锁存地址信号。为了稳定锁存地址，即在 ADC 转换周期内 8 路模拟开关稳定地接通某一指定的模拟通道。ALE 脉宽应大于 100ns。下一个 ALE 脉冲上升沿允许通道地址更新锁存。实际使用中，要求 ADC 开始转换之前地址就应锁存，所以通常把 ALE 端与 START 端连在一起，施加同一脉冲信号。上升沿锁存地址，下降沿启动转换。

ADDA、ADDB、ADDC：它们分别为模拟通道的地址选择线，其中，ADDA 是最低位，ADDC 是最高位。它们的译码状态分别对应 $IN_0 \sim IN_7$ 8 个通道。

小　　结

A/D、D/A 转换器是现代数字系统的重要部件，应用日益广泛。A/D、D/A 转换器的种类十分繁杂，不可能一一举例。因此首先应着重基本类型的原理、概念的理解。

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。

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下，可采用双积分 A/D 转换器，当然也可选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有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、D/A 转换器的发展趋势是高速度、高分辨率及易于与微型计算机接口，用以满足各个应用领域对信号处理的要求。

习　　题

9-1　8 位 D/A 转换器的分辨率是多少？

9-2　某一控制系统中有一个 D/A 转换器，若系统要求该 D/A 转换器的精度要小于 0.25。试问应选多少位的 D/A 转换器？

9-3　对于一个 8 位 D/A 转换器，试回答：

（1）若最小输出电压增量为 0.02V，问当输入代码为 01001100 时，输出电压 u_O 为多少伏？

（2）其分辨率用百分数表示是多少？

（3）某一系统中要求 D/A 转换器的精度小于 0.12%，问这一 D/A 转换器能否应用？

9-4　倒 T 形电阻网络 D/A 转换器的电路如图 9-6 所示。已知 $U_{REF}=10V$，开关接运放反相输入端。试求：

（1）输出模拟电压 u_O 的输出范围。

（2）当输入数字量 $D_3D_2D_1D_0=0110$ 时，对应的 u_O 值。

9-5　电路如图 9-5 所示。试问：

（1）若 $U_{REF}=10V$，输入信号 $D_3D_2D_1D_0=0011$，输出的模拟电压的值为多少？

（2）电路的分辨率为多少？

9-6　有一个逐次比较型 4 位 A/D 转换器，输入电压 u_I 和 D/A 转换器的输出电压 u_O 的波形如图 9-31（a）、（b）所示。

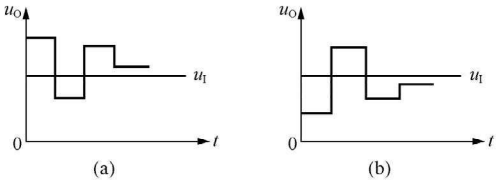


图 9-31　题 9-6 图

（1）转换结束时，图 9-31（a）、（b）的输出数字量各为多少？

（2）4 位 D/A 转换器的最大输出电压为 $U_{Omax}=30V$，估计图 9-31（a）、（b）两种情况下的输入电压范围各是多少？

9-7　模拟信号最高频率分量 $f=20kHz$，问对该信号取样的最低取样频率应为多少？

9-8　为什么 A/D 转换一定要量化？选用哪种量化方法误差比较小？

9-9　双积分 A/D 转换器中，计数器最大计数容量为 $(3000)_D$。已知计数时钟频率 $f_{CP}=30kHz$，积分器中 $R=100k\Omega$，$C=1\mu F$，输入电压 u_I 的变化范围为 0～5V。试求：

（1）第一次积分时间 T_1。

（2）求积分器的最大输出电压 $|U_{Omax}|$。

（3）当 $U_{REF}=10V$，第二次积分计数器计数值 $\lambda=(1500)_{10}$ 时，输入电压的平均值 u_I 为多少？输出数字量又是多少？

9-10　计数式 A/D 转换器的原理框图如图 9-32 所示，其中 D/A 的最大输出电压为 5V，CS 为开始转换控制信号。令开始转换控制信号 CS 为 0 时，计数器清零；当 $u_I=1.2V$ 后，CS=1，开始转换，问：

（1）转换结束后，输出数字量？

（2）转换误差是多少？

（3）叙述转换过程。

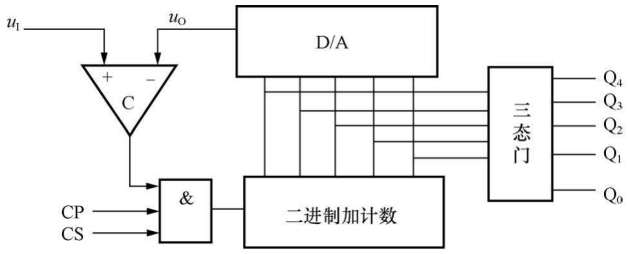


图 9-32　题 9-10 图

9-11　试用 D/A 转换器 AD7520 和计数器 74X161 组成阶梯形波发生器，输出的阶梯波波形如图 9-33 所示，要求画出完整的逻辑电路图。

9-12　在如图 9-27 所示的双积分型 A/D 转换中，若计数器为 10 位二进制计数器，时钟信号频率为 50kHz，试计算完成一次转换需最小转换时间是多少？

9-13　电路如图 9-34 所示。试回答：

（1）运放工作在线性区还是非线性区？

（2）这是哪种类型的解码网络？

（3）求当 $D_4D_3D_2D_1D_0=11100$ 时，输出电压 u_O 的值。

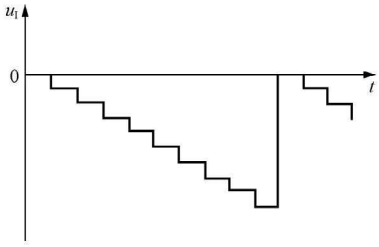


图 9-33　题 9-11 图

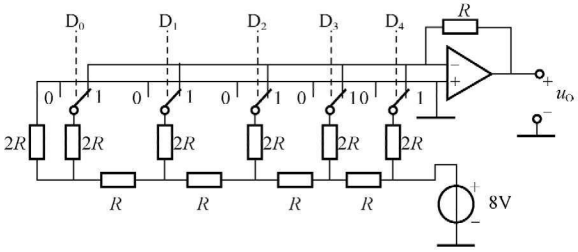


图 9-34　题 9-13 图

第十章　数 字 系 统 设 计

本　章　提　要

本章首先给出数字系统的定义、算法状态机和数字系统设计的一般流程，然后用四个具体实例详细介绍了数字系统的设计方法。

第一节　概　　述

一、数字系统的定义

前面几章介绍的编码器、译码器、数据选择器、数据分配器、比较器、加法器、计数器、寄存器、存储器等电路，都只能实现某一特定的功能，因此称为功能部件级电路。由若干这样的功能部件和一个控制电路一起构成的系统可以完成更复杂的功能，这就是数字系统。数字系统是一个能完成一系列复杂操作的逻辑单元，它可以是一台数字计算机，一个自动控制系统，一个数据采集系统，或者是日常生活中用的一些电子仪器，如电子秤等。

用前面学过的状态表的方法设计大型复杂的数字系统，有时是十分困难的，甚至是不可能的，原因是状态数大得惊人。

二、数字系统的设计方法

数字系统的设计方法有两种：自上而下的设计方法和自下而上的设计方法。现代数字系统的设计常采用自上而下的设计方法。自上而下的设计方法的设计步骤如下。

1. 明确设计任务

仔细阅读技术说明书，将叙述不明确的地方加以明确。

2. 确定系统方案和逻辑划分

将待设计的数字系统划分为控制器和数据通路两部分，给出 ASM 图（ASM 图将在第二节中予以介绍）。

图 10-1 给出了数字系统的控制器和数据通路划分示意图。数据通路主要完成数据的采集、存储、运算处理和传输。它与外界进行数据交换，而它所有的存取、运算等操作都是在控制器的控制下进行的。它与控制器的联系是接收控制器发出的控制信号，向控制器发出自身的状态信号。

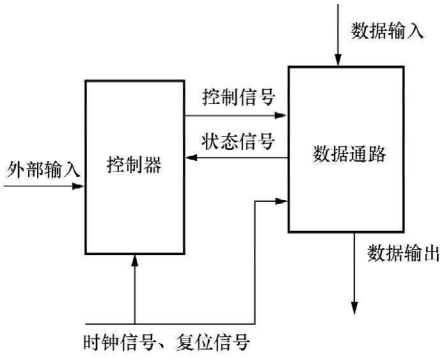


图 10-1　数字系统的控制器和数据通路划分示意图

控制器根据外部输入信号和数据通路提供的状态信号，决定下一步要完成的操作，并向数据通路发出控制信号以控制数据通路完成该操作。通常以是否具有控制器作为区别功能部件和数字系统的标志。凡是含有控制器且能按顺序进行操作的系统，不论规模大小，都可以称为数字系统。否则，不论规模多么大，比如大规模存储电路，也不能称为数字系统。

3. 数据通路的设计

数据通路由存储器、运算器、数据选择器等部件组成，这些部件主要由通用的逻辑器件组成。组成数据通路的器件选用标准有两条：一是易于控制，即器件的控制方式、控制信号以及产生这些控制信号的逻辑应尽可能简单，以简化控制单元的设计；二是力求模块数少，以减少电路体积、功耗，降低成本。

4. 控制器的设计

控制器是执行算法的核心。控制器的设计可采用前面学过的同步时序逻辑电路的设计方法。

三、数字系统的实现方法

随着集成电路技术的发展，数字系统的设计方法也经历了分立元件、小规模、中规模、大规模、超大规模、专用集成电路这样一个发展历程。在目前的技术水平上，数字系统的实现方法大概有以下几种方法：①采用通用的集成电路实现；②采用可编程逻辑器件和硬件描述语言实现；③采用单片微处理器作为核心实现；④采用片上系统实现。

1. 采用通用的集成电路实现

这是一种比较传统的设计方法，已经历了比较长的发展历史，积累了一定的经验，实际应用比较广泛，目前仍被设计者使用，是学习其他更先进的设计方法的基础。后面几节中的四个设计实例就采用了这种方法。

2. 采用可编程逻辑器件和硬件描述语言实现

用可编程逻辑器件实现数字系统时，要用 VHDL 或 Verilog-HDL 语言来描述数字系统的功能，然后在相应开发软件的支持下，对 VHDL 或 Verilog-HDL 程序进行仿真、综合后下载到可编程逻辑芯片中。由于目前流行的 VHDL 或 Verilog-HDL 综合软件主要针对寄存器传输级进行综合，所以应该在寄存器传输级建立数字系统的模型。尽管可以在越来越抽象的级别上描述数字系统，用中规模集成电路实现数字系统的方法还是必须要掌握的。

3. 采用单片微处理器作为核心实现

这种方法使用器件少，功能灵活，但是速度较低，比较典型的例子是单片机系统和嵌入式系统，应用也比较广泛。

4. 采用片上系统实现

随着大规模集成电路技术和数字系统设计自动化软件技术的飞速发展，目前可以将一个完整的数字系统集成在一个芯片中，即所谓的片上系统，从而使得电子设备的体积大大缩小。

本章主要介绍第一种和第二种实现方法。

第二节 算法状态机

算法状态机（Algorithmic State Machine，ASM）是一种描述时钟驱动的控制器的工作流程的方法。ASM 图的元素包括状态框、判决框、条件输出框、输入路径和输出路径。

一、状态框

状态框是一个矩形块，如图 10-2 所示。它表示控制器的一个状态，该状态的名字放

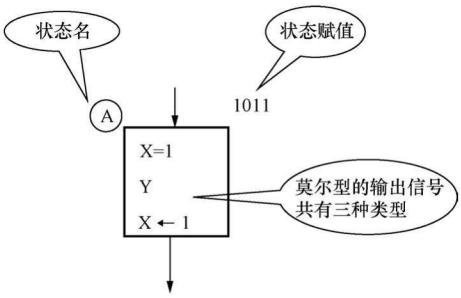


图 10-2 ASM 图的状态框示意图

在左上角，右上角是此状态的赋值。矩形块内是控制器的莫尔型输出信号，共有三种不同的输出形式。“X=1”表示输出信号 X 输出一个时钟周期的 1 信号，其他的时间输出 0。“Y”表示输出信号 Y 在电路的 A 状态持续期间一直输出 1。“X ←—1”表示在状态 A 结束时即状态 A 的次态开始时将 1 赋给 X。

二、判决框

判决框是一个菱形块，如图 10-3 所示。块内是控制器的输入信号，每一个输入信号都对应一个判决框。

三、条件输出框

条件输出框是一个椭圆状或两端为圆弧线的块，如图 10-4 所示。块内是控制器的米里型的输出。

输入、输出路径是一些带箭头的有向线段，由它们把状态框、判别框和条件输出框有机地连接起来。

电路状态的转换是在系统时钟脉冲 CP 的控制下进行的。当无 CP 脉冲到来时，系统将维持现在的状态不变。

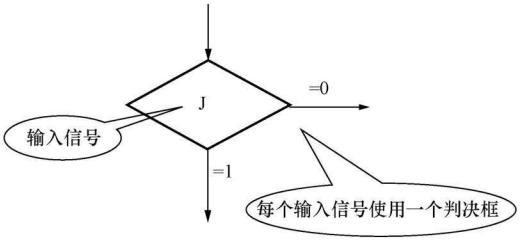


图 10-3 ASM 图的判决框示意图

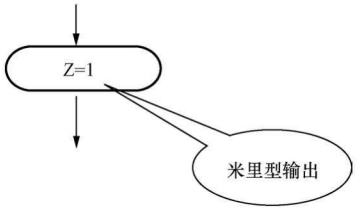


图 10-4 ASM 图的条件输出框示意图

第三节 数字系统设计举例之一

设计一个数字系统，要求能完成如下功能：① 3 位二进制数据的装载；② 数据的左移功能；③ 数据的普通的右移功能；④ 兼有运算作用的数据的右移功能。

设计步骤：

(1) 明确设计要求。题目中给出的要求不是十分明确，所以设计者需要再做进一步的约定。约定系统输入的三位二进制数据为 $X_2X_1X_0$，系统输出的三位二进制数据为 $B_2B_1B_0$，输入信号 A 控制数据的左移操作和右移操作，当 A=0 时进行左移操作，当 A=1 时进行右移操作。当 A=0 时进行左移操作，最低位补 0，最高位被丢弃，即 3 位二进制数据 $B_2B_1B_0$ 变为 $B_1B_0 0$。当 A=1 时进行右移操作，在进行右移操作之前，对输入信号的最低位 X_0 进行检测，若 $X_0=1$，则执行 $B_0=B_0\oplus B_1$、$B_1=B_1\oplus B_2$、$B_2=B_2B_1B_0$ 操作；若 $X_0=0$，则执行普通的右移操作，最高位补 0，最低位被丢弃，即 3 位二进制数据 $B_2B_1B_0$ 变为 $0B_2B_1$。待设计数字系统的对外接口信号示意图如图 10-5 所示。

（2）确定系统方案和逻辑划分，分为控制器和数据通路两部分。将待设计的数字系统划分为数据通路和控制器两部分，如图 10-6 所示。

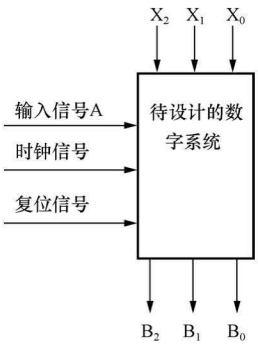


图 10-5　待设计数字系统的对外接口信号示意图

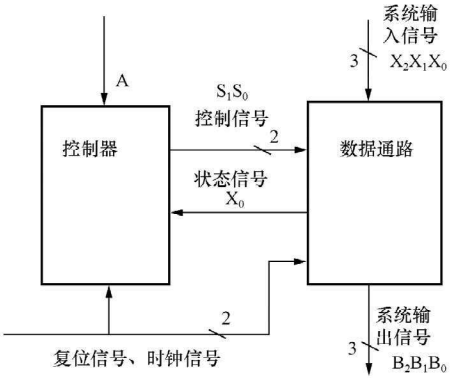


图 10-6　待设计数字系统的数据通路和控制器功能划分示意图

（3）给出待设计数字系统的 ASM 图，如图 10-7 所示。

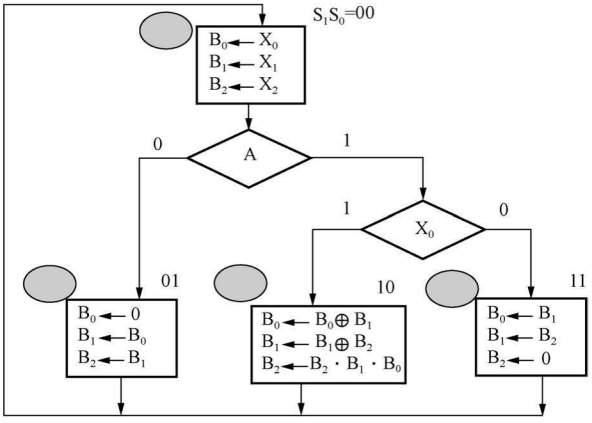


图 10-7　待设计数字系统的 ASM 图

（4）控制器的设计。控制器的对外接口信号示意图如图 10-8 所示。控制器的输入信号是 A、X_0，输出信号是 S_1、S_0。用莫尔状态机实现控制器电路。因为要实现四种功能，所以可以设计电路共有四个状态，在每一种状态下实现一种功能。电路的状态可以用两个 D 触发器实现，两个触发器的状态分别用 Q_1 和 Q_0 表示。

当 $Q_1Q_0=00$ 时，进行数据的装载，即 $B_2=X_2$，$B_1=X_1$，$B_0=X_0$；

当 $Q_1Q_0=01$ 时，完成数据的左移，即 $B_2=B_1$，$B_1=B_0$，$B_0=0$；

当 $Q_1Q_0=10$ 时，完成数据的右移功能并执行 $B_2=B_2 \cdot B_1 \cdot B_0$，$B_1=B_1 \oplus B_2$，$B_0=B_0 \oplus B_1$；

当 $Q_1Q_0=11$ 时，完成右移操作，$B_2=0$，$B_1=B_2$，$B_0=B_1$。

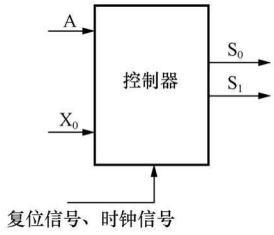


图 10-8　控制器部分的对外接口信号示意图

由前面的分析可知，输入信号 A 控制数据的左移和右移操作，当 A=0 时进行左移操作，最低位补 0，即 3 位二进制数据 $B_2B_1B_0$

变为 $B_1B_0 0$。但是此处还有需要说明的地方。当电路正处在 $Q_1Q_0=01$（数据的左移状态）、$Q_1Q_0=10$ 或者 $Q_1Q_0=11$ 状态时，若 $A=0$，那么电路是继续处在数据的左移状态、$Q_1Q_0=10$ 或者 $Q_1Q_0=11$ 状态，还是回到数据的装载状态（$Q_1Q_0=00$）呢？在此处认为电路应该回到数据的装载状态，数据不能无限制地左移或者右移，因为这不符合一般运算的要求。

当 $A=1$ 时进行右移操作，在进行右移操作之前，对输入信号的最低位 X_0 进行检测，若 $X_0=0$，则执行右移操作，最高位补 0，即 3 位二进制数据 $B_2B_1B_0$ 变为 $0B_2B_1$。此处还有需要明确的地方。当电路正处在 $Q_1Q_0=01$（数据的左移状态）、$Q_1Q_0=10$ 或者 $Q_1Q_0=11$ 状态时，若 $A=1$ 和 $X_0=0$ 成立，那么电路是继续处在数据的左移状态或者 $Q_1Q_0=10$ 或者 $Q_1Q_0=11$ 状态，还是回到数据的装载状态（$Q_1Q_0=00$）呢？在此处认为也应该回到数据的装载状态，数据不能无限制地右移，因为这不符合一般运算的要求。

同理，当 $A=1$ 时进行右移操作，在进行右移操作之前，对输入信号的最低位 X_0 进行检测，若 $X_0=1$，则执行 $B_2=B_2\cdot B_1\cdot B_0$、$B_1=B_1\oplus B_2$、$B_0=B_0\oplus B_1$ 操作。当电路正处在 $Q_1Q_0=01$（数据的左移状态）、$Q_1Q_0=10$ 或者 $Q_1Q_0=11$ 状态时，若 $A=1$ 和 $X_0=1$ 成立，那么电路回到数据的装载状态（$Q_1Q_0=00$），因为数据不能无限制地右移。

后面将用前面学过的同步时序电路的设计方法设计此控制器。

（5）数据通路的设计。数据通路部分的对外接口信号示意图如图 10-9 所示。

数据通路可采用数据选择器、D 触发器和门电路组成。数据通路的输入信号 $X_2X_1X_0$ 是整个数字系统待处理的数据，数据处理完成后传送给 $B_2B_1B_0$，还有两个输入信号 S_1S_0 来自控制器的输出信号，决定进行何种运算。

当控制器的输出 $S_1S_0=00$ 时进行数据的装载，即 $B_2=X_2$、$B_1=X_1$、$B_0=X_0$；当控制器的输出 $S_1S_0=01$ 时完成数据的左移，即 $B_2=B_1$、$B_1=B_0$、$B_0=0$；当控制器的输出 $S_1S_0=10$ 时完成数据的右移功能并执行 $B_2=B_2\cdot B_1\cdot B_0$、$B_1=B_1\oplus B_2$、$B_0=B_0\oplus B_1$ 操作；当控制器的输出 $S_1S_0=11$ 时完成右移操作，即 $B_2=0$、$B_1=B_2$、$B_0=B_1$。

根据上述描述，得到数据通路部分的电路图如图 10-10 所示。该电路使用了三个 4 选 1

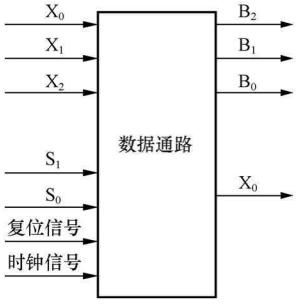


图 10-9 数据通路部分的对外接口信号示意图

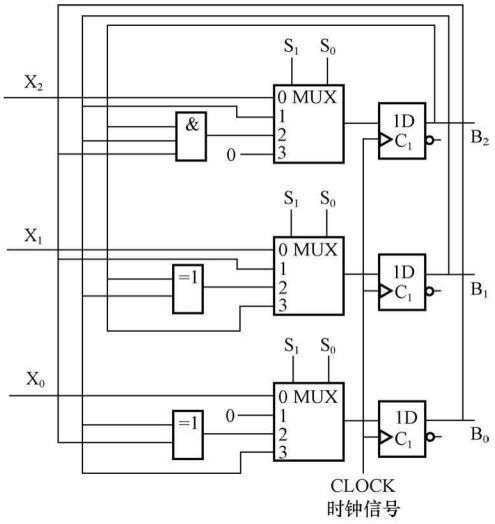


图 10-10 数据通路部分的电路图

数据选择器 MUX，地址选择端由 S_1S_0 控制，数据输入端分别接对应的信号。

（6）下面用同步时序电路的设计方法设计控制器。画出控制器的状态图，如图 10-11 所示。采用莫尔设计，共有 4 个状态，输入信号是 AX_0，输出信号是 S_1S_0。

此控制器的状态表如表 10-1 所示，驱动信号真值表如表 10-2 所示。

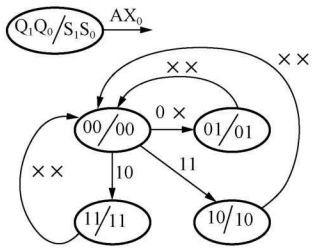


图 10-11　控制器的状态图

表 10-1　控 制 器 的 状 态 表

输入信号		现态信号		次态信号		输出信号	
A	X_0	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	S_1	S_0
0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	1	1	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	0	0	1	0	1	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

表 10-2　控制器的驱动信号真值表

输入信号		现态信号		触发器的输入信号	
A	X_0	Q_1^n	Q_0^n	D_1	D_0
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

用卡诺图化简得到 D_1、D_0 的逻辑表达式为

$$D_1 = \overline{Q_1^n} \cdot \overline{Q_0^n} \cdot A,\quad D_0 = \overline{Q_1^n} \cdot \overline{Q_0^n} \cdot (\overline{A} + \overline{X_0})$$

可以将触发器的状态信号直接作为输出信号，将表 10-1 中的 Q_1^{n+1}、Q_0^{n+1} 两列删去，又因为 S_1、S_0 与 A、X_0 无关，所以将 A、X_0 两列也删去，得到表 10-3。由表 10-3 写出输出信号逻辑函数表达式，即

$$S_1 = Q_1^n$$

$$S_0 = Q_0^n$$

表 10-3　　控制器输出信号的真值表

现态信号		输出信号		现态信号		输出信号		现态信号		输出信号	
Q_1^n	Q_0^n	S_1	S_0	Q_1^n	Q_0^n	S_1	S_0	Q_1^n	Q_0^n	S_1	S_0
0	0	0	0	1	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1	0	1	0	1
1	0	1	0	0	0	0	0	1	0	1	0
1	1	1	1	0	1	0	1	1	1	1	1
0	0	0	0	1	0	1	0				
0	1	0	1	1	1	1	1				

根据上面表达式得到控制器的逻辑电路图如图 10-12 所示。

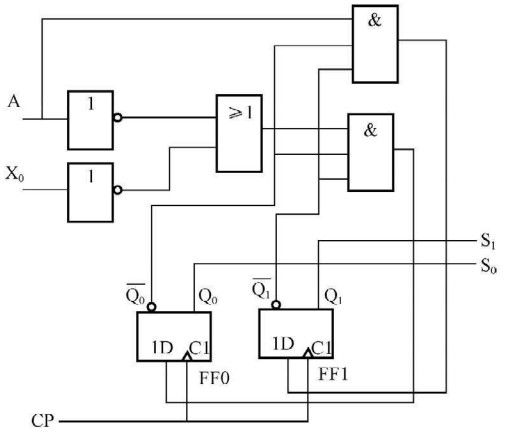


图 10-12 待设计数字系统控制器的电路图

将上述的分析综合起来，不难得到待设计数字系统的完整逻辑电路图。

第四节 数字系统设计举例之二

设计一个四位无符号二进制数相乘的乘法运算单元。

设计步骤：

(1) 明确设计要求。Reset 信号为乘法运算单元的启动信号，正脉冲启动。Halt 信号为乘法运算单元的运算完成指示信号，高电平有效，即当为高电平时表示运算完成。待设计电路的对外接口信号如图 10-13 所示。

图 10-14 给出了手工进行二进制数 0111 和 1010 相乘的过程。乘数从右到左各位依次被检测，如果为 1，部分积就是被乘数；如果为 0，部分积就是 0。每一个新的部分积在加入到“部分积之和”以前要左移。也可以这样想，保持部分积位置固定，将“部分积之和”右移，此处使用后一种思路进行电路的设计。

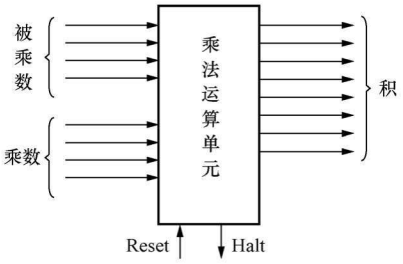
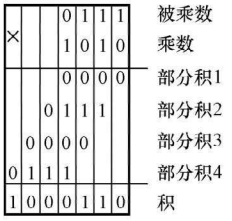


图 10-13 待设计数字系统的对外接口信号示意图



			0	1	1	1	被乘数
×			1	0	1	0	乘数
			0	0	0	0	部分积1
		0	1	1	1		部分积2
	0	0	0	0			部分积3
0	1	1	1				部分积4
1	0	0	0	1	1	0	积

图 10-14 手工进行二进制数相乘运算过程

(2) 确定系统方案和逻辑划分，将待设计的乘法器划分为控制器和数据通路两部分。数据通路部分的对外接口信号如图 10-15 所示。控制器部分的对外接口信号如图 10-16 所示。

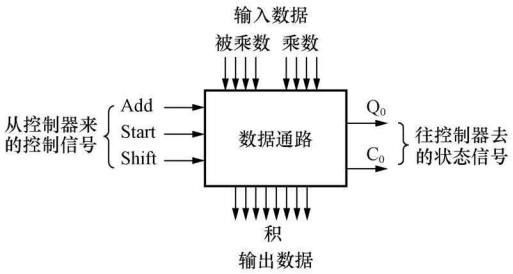


图 10-15　数据通路部分的对外接口信号示意图

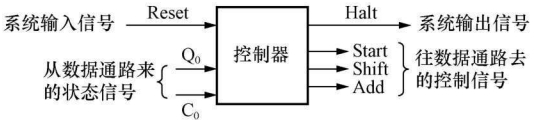


图 10-16　控制器部分的对外接口信号示意图

（3）数据通路的设计。数据通路使用了 M、Q、A 三个寄存器、一个二进制加法单元和一个 2 位二进制计数器单元 CNT。M、Q 和 A 三个寄存器共用同一个时钟信号。2 位二进制计数器单元 CNT 有 00、01、10、11 共四个状态，当为 00 状态时 C_0 信号被设置为 1，表明这是第 4 个周期，运算已经完成。2 位二进制计数器单元 CNT 在控制器的 Shift 输出信号控制下完成状态的转换。

寄存器 M 存储 4 位被乘数，没有移位功能。寄存器 Q 有移位功能，在运算开始时，存储 4 位乘数；在运算过程中，存放“部分积之和”的低位部分；当运算结束时，其中存放有乘积的低 4 位。Q_0 是寄存器 Q 的最低位。寄存器 A 有移位功能，可以存储 5 位数据，在运算过程中，其中最高位存放二进制加法单元来的进位，低 4 位存放“部分积之和”的高位部分；当运算结束时，其中存放有乘积的高 5 位。

寄存器 M 选用一片集成芯片 74X194，只使用其中的并行输入和存储功能。寄存器 Q 选用一片集成芯片 74X194，使用其中的并行输入、存储功能和右移功能。寄存器 A 必须存储 5 位数据，最高位由一个单独的 D 触发器构成，选用一片 74X74。低 4 位选用一片集成芯片 74X194，使用其中的并行输入、存储功能和右移功能。

二进制加法单元选用一片集成芯片 74X283，它是 4 位二进制加法器。

2 位二进制计数器的状态用来记忆加法的迭代次数，因为是 4 位二进制数相加，所以要有 4 个状态，可以用一片集成芯片 74X161 和门电路构成。

（4）控制器的设计。控制器的输入信号是 Reset、Q_0 和 C_0。Q_0 和 C_0 是从数据通路而来的状态信号。当 Q_0 是 0 时，不进行加计算，跳过这一步。

控制器的输出信号有 Start、Shift、Add 和 Halt。这些是往数据通路去的控制信号。这四个信号均设定为高电平有效。

Start 信号有效使数据通路中的寄存器 A 和 2 位二进制计数器单元 CNT 清零，寄存器 M 和 Q 置数，其中寄存器 M 存放有被乘数，寄存器 Q 存放有乘数。Add 信号有效使数据通路中的寄存器 A 进行置数。Shift 信号有效使数据通路中的寄存器 A 和寄存器 Q 进行移位，2 位二进制计数器单元状态进入下一状态。Halt 信号有效表明运算完成。

用莫尔状态机实现，电路共有四个状态，分别用 Start、Add、Shift 和 Halt 表示，四种状态下的输出信号就是各自状态的名字。四个状态最少可以用两个触发器实现，触发器的状态用 Y_1Y_0 表示。Start 状态对应 $Y_1Y_0=00$，Add 状态对应 $Y_1Y_0=01$，Shift 状态对应 $Y_1Y_0=10$，Halt 状态对应 $Y_1Y_0=11$。

当 $Y_1Y_0=00$ 时，Start 信号有效使数据通路中的寄存器 A 和 2 位二进制计数器单元清零，寄存器 M 和 Q 置数，其中寄存器 M 存放有被乘数，寄存器 Q 存放有乘数。

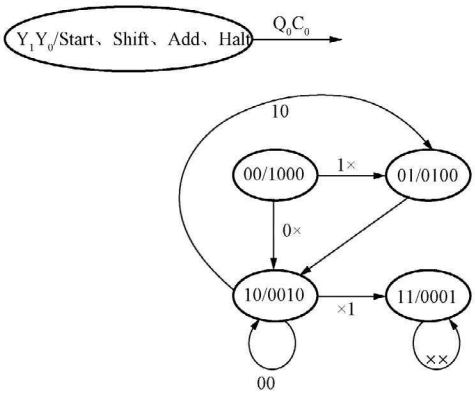


图 10-17 控制器的状态图

当 $Y_1Y_0=01$ 时，Add 信号有效使数据通路中的寄存器 A 进行置数，其中存放有“部分积之和”。

当 $Y_1Y_0=10$ 时，Shift 信号有效使数据通路中的寄存器 A 和寄存器 Q 进行移位，2 位二进制计数器单元 CNT 的状态进入下一状态。也就是说，CNT 在每完成一次移位后状态加 1，而不是在 CP 时钟的有效沿之后加 1。

当 $Y_1Y_0=11$ 时，Halt 信号有效表明整个运算过程已经完成。

控制器的状态图如图 10-17 所示，状态表如表 10-4 所示。

表 10-4　　控制器的状态表

输入信号		现态信号		次态信号		输出信号			
Q_0	C_0	Y_1^n	Y_0^n	Y_1^{n+1}	Y_0^{n+1}	Start	Add	Shift	Halt
0	0	0	0	1	0	1	0	0	0
0	0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	1	1	1	0	0	0	1
0	1	0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	1	0
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	0	0	0
1	0	0	1	1	0	0	1	0	0
1	0	1	0	0	1	0	0	1	0
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	1	1	0	0	0
1	1	0	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0	1	0
1	1	1	1	1	1	0	0	0	1

用 D 触发器实现上述功能。列出输出信号和驱动信号的真值表如表 10-5 所示。

表 10-5　　输出信号和驱动信号的真值表

输入信号		现态信号		驱动信号		输出信号			
Q_0	C_0	Y_1^n	Y_0^n	D_1	D_0	Start	Add	Shift	Halt
0	0	0	0	1	0	1	0	0	0
0	0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	0	0	1	0

续表

输入信号		现态信号		驱动信号		输出信号			
Q_0	C_0	Y_1^n	Y_0^n	D_1	D_0	Start	Add	Shift	Halt
0	0	1	1	1	1	0	0	0	1
0	1	0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0	1	0
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	0	0	0
1	0	0	1	1	0	0	1	0	0
1	0	1	0	0	1	0	0	1	0
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	1	1	0	0	0
1	1	0	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0	1	0
1	1	1	1	1	1	0	0	0	1

用卡诺图化简方法可得到触发器驱动函数的逻辑表达式为

$$D_1 = \overline{Q_0} + \overline{Y_0^n} + C_0 Y_1^n$$

$$D_0 = Q_0 \overline{Y_0^n} + Y_1^n Y_0^n + C_0 Y_1^n$$

根据上面表达式得到控制器的逻辑电路图，如图 10 - 18 所示。

（5）画出待设计数字系统的 ASM 图，如图 10 - 19 所示。

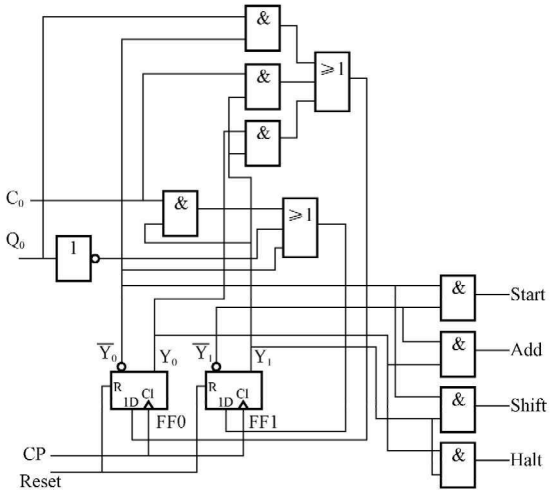


图 10 - 18　控制器的逻辑电路图

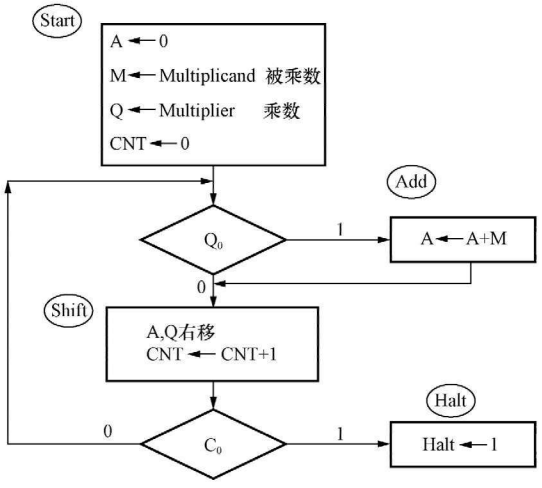


图 10 - 19　待设计数字系统的 ASM 图

（6）乘法器电路框图。进一步细化的乘法器电路框图如图 10 - 20 所示。图中，显示出了控制器和数据通路部分。读者可以根据此框图得到更细化的逻辑电路。

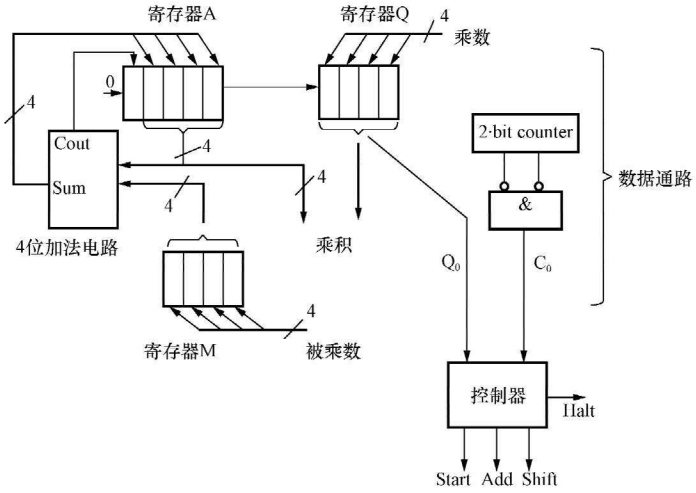


图 10-20 乘法器电路框图

第五节 数字系统设计举例之三

设计一个彩灯控制器电路。要求该电路 8 路彩灯 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$ 的排列如图 10-21所示，彩灯控制器可以自动控制多路彩灯，按不同的节拍循环显示各种灯光变换花型。要求变换的三种花型是：

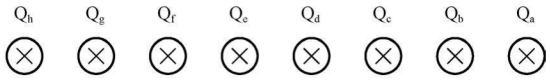


图 10-21 8 路彩灯的排列图

花型 1：8 路彩灯从左到右顺次点亮，全亮后逆序渐灭。循环两次。

花型 2：8 路彩灯从中间到两边对称地逐次渐亮，全亮后仍由中间到两边对称地逐次渐灭。循环两次。

花型 3：8 路彩灯分两半，从左到右顺次渐亮，全亮后则灭。循环三次。

三种花型先以慢节拍显示一遍，再以快节拍显示一遍，然后往复循环。

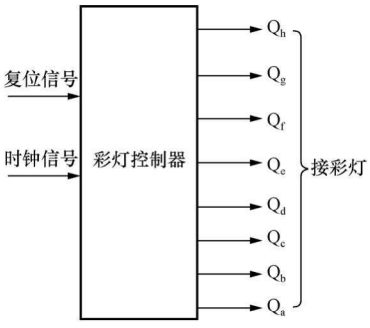


图 10-22 彩灯控制器的对外接口信号示意图

设计步骤：

(1) 明确设计要求。彩灯控制器的对外接口信号示意图如图 10-22 所示。

设彩灯控制器输出为 1 时可以点亮彩灯，为 0 时不能点亮彩灯。根据原题要求列出彩灯控制器的各花型的状态列表，花型 1 的状态列表如表 10-6 所示，花型 2 的状态列表如表 10-7 所示，花型 3 的状态列表如表 10-8 所示。表 10-9 为一次总循环所需要的状态列表，其中加入了第 64 个状态并规定为清零，是为了使状态总数等于 2^6，以简化控制器电路。

表 10-6　花 型 1 的 状 态 列 表

时钟周期的序号	状态 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$
1	00000000
2	10000000
3	11000000
4	11100000
5	11110000
6	11111000
7	11111100
8	11111110
9	11111111
10	11111110
11	11111100
12	11111000
13	11110000
14	11100000
15	11000000
16	10000000

表 10-7　花 型 2 的 状 态 列 表

时钟周期的序号	状态 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$
1	00000000
2	00011000
3	00111100
4	01111110
5	11111111
6	11100111
7	11000011
8	10000001

表 10-8　花 型 3 的 状 态 列 表

时钟周期的序号	状态 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$
1	00000000
2	10001000
3	11001100
4	11101110
5	11111111

表 10-9　　彩灯控制器一次总循环的状态列表

时钟周期的序号	状态 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$	时钟周期的序号	状态 $Q_hQ_gQ_fQ_eQ_dQ_cQ_bQ_a$
初始状态	00000000	33	00011000
1	10000000	34	00111100
2	11000000	35	01111110
3	11100000	36	11111111
4	11110000	37	11100111
5	11111000	38	11000011
6	11111100	39	10000001
7	11111110	40	00000000
8	11111111	41	00011000
9	11111110	42	00111100
10	11111100	43	01111110
11	11111000	44	11111111
12	11110000	45	11100111
13	11100000	46	11000011
14	11000000	47	10000001
15	10000000	48	00000000
16	00000000	49	10001000
17	10000000	50	11001100
18	11000000	51	11101110
19	11100000	52	11111111
20	11110000	53	00000000
21	11111000	54	10001000
22	11111100	55	11001100
23	11111110	56	11101110
24	11111111	57	11111111
25	11111110	58	00000000
26	11111100	59	10001000
27	11111000	60	11001100
28	11110000	61	11101110
29	11100000	62	11111111
30	11000000	63	00000000
31	10000000	64	00000000
32	00000000		

花型1, 2遍

花型2, 2遍

花型3, 3遍

（2）确定系统方案和逻辑划分，将待设计数字系统分为控制器和数据通路两部分。

（3）数据通路的设计。数据通路的组成：

1）一个时钟信号产生电路，其框图如图 10-23 所示。它能同时输出某频率的时钟信号和它的二分频信号。

2）一个二选一数据选择器，其框图如图 10-24 所示。当 CLKCTRL＝0 时，CLK＝CLKA；当 CLKCTRL＝1 时，CLK＝CLKB。

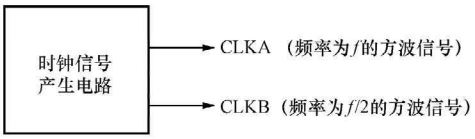


图 10-23　时钟信号产生电路的接口信号

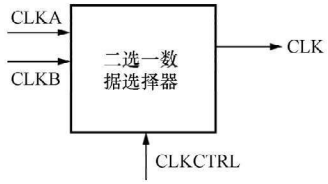


图 10-24　二选一数据选择器的接口信号

3）两片 74X194 移位寄存器，如图 10-25 所示。74X194 能左移、右移、清零或并行置零，可满足题目中的要求，两片 74X194 共有 8 个输出信号，分别控制 8 路彩灯。

总结图 10-23、图 10-24、图 10-25，可得到数据通路的对外接口信号如图 10-26 所示。

（4）控制器的设计。控制器的对外接口信号如图 10-27 所示。

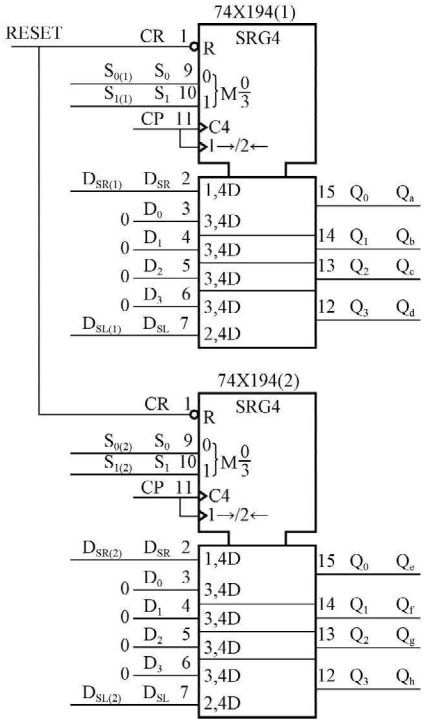


图 10-25　两片 74X194 移位寄存器的接口信号

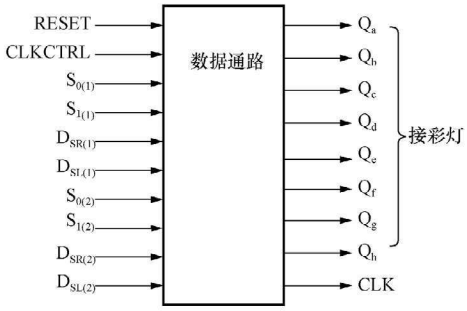


图 10-26　数据通路对外接口信号示意图

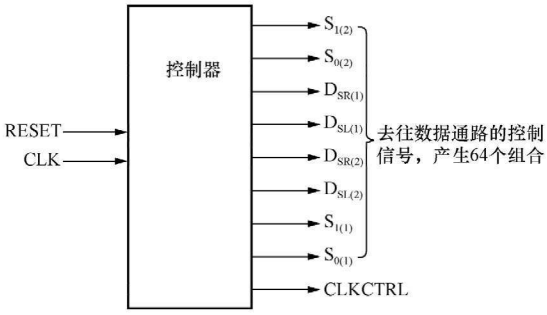


图 10-27　控制器的对外接口信号示意图

由于一次花型循环实为 63 拍，再加一拍清零共 64 拍，此外，设计还要求慢节拍和快节拍各循环一次，所以总共需要 128 拍。7 位二进制计数器共有 128 个状态，用 $Q_6Q_5Q_4Q_3Q_2Q_1Q_0$ 表示，从 0000000 到 1111111 循环变化。

设计此电路模块的方案有多种，图 10-28 是其中的一种电路设计方案。图中 FF0～FF5

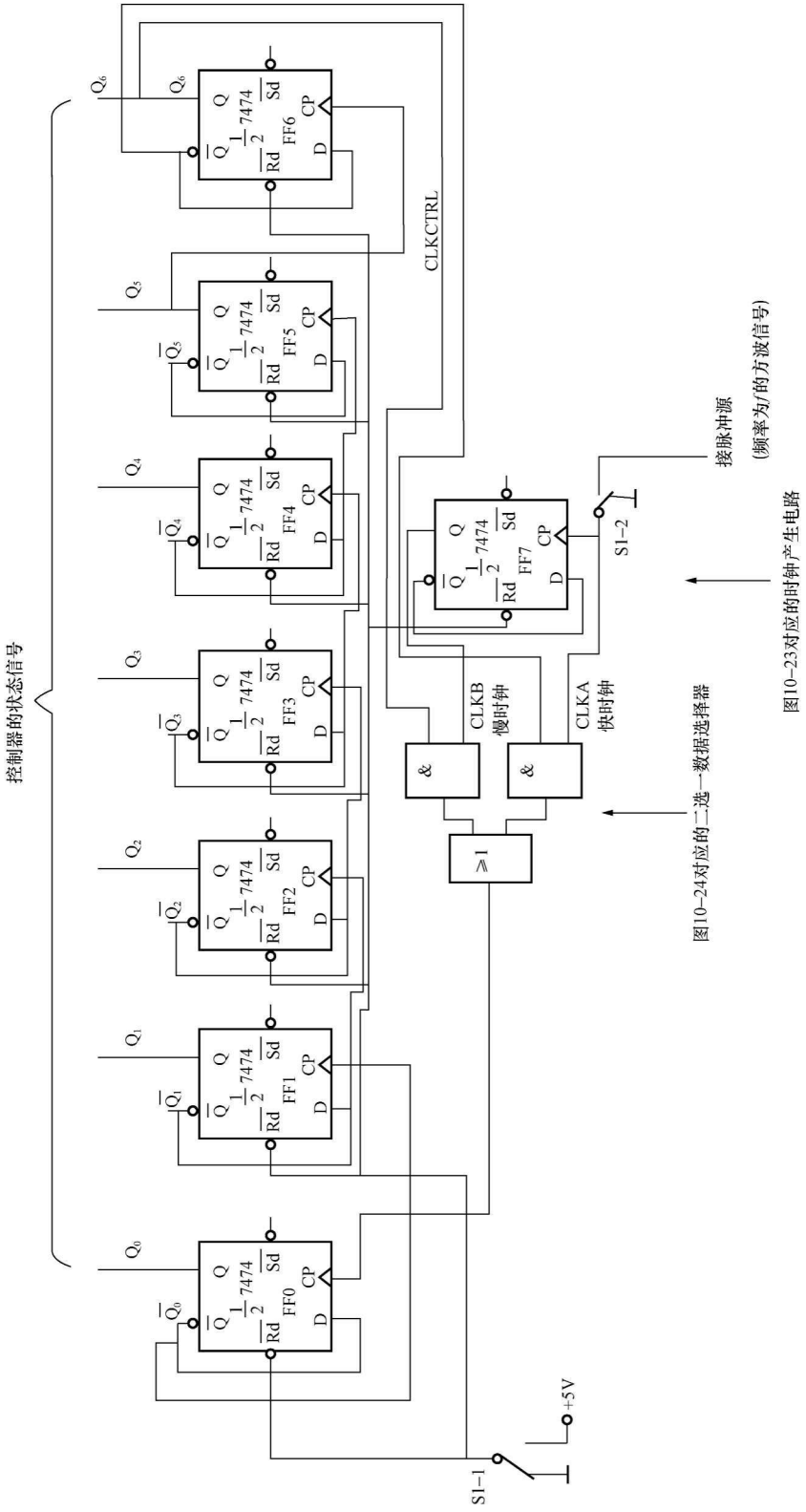


图10-28　128个状态的产生电路

构成 6 位二进制计数器，用于产生 64 节拍。FF6 实现二分频，FF7 和与门、或门构成二选一数据选择器，和 FF6 共同完成快、慢节拍自动控制。电路共需 4 片 7474 集成触发器。开关 S1-1 和 S1-2 应同步动作，完成节拍发生器的清零和启动。为了运行可靠，可将 FF0～FF7 的 Sd 端接到＋5V 上。

表 10 - 10 是控制器的慢节拍或快节拍的 64 个状态与 74X194 芯片的控制信号对应关系。

为便于设计，74X194 采用同步置零方式实现清零。再次列出 74X194 的功能表如表 10 - 11 所示。控制器的输出信号真值表如表 10 - 12 所示。

表 10 - 10　　控制器的状态与 74X194 芯片的控制信号对应关系

节拍	$(Q_5Q_4Q_3Q_2Q_1Q_0)^n$	移位寄存器 74X194（2）	移位寄存器 74X194（1）
1～4	000000～000011	左移，$D_{SL}=1$	左移，$D_{SL}=0$
5～8	000100～000111	左移，$D_{SL}=1$	左移，$D_{SL}=1$
9～12	001000～001011	右移，$D_{SR}=1$	右移，$D_{SR}=0$
13～16	001100～001111	右移，$D_{SR}=0$	右移，$D_{SR}=0$
17～20	010000～010011	左移，$D_{SL}=1$	左移，$D_{SL}=0$
21～24	010100～010111	左移，$D_{SL}=1$	左移，$D_{SL}=1$
25～28	011000～011011	右移，$D_{SR}=1$	右移，$D_{SR}=0$
29～32	011100～011111	右移，$D_{SR}=0$	右移，$D_{SR}=0$
33～36	100000～100011	右移，$D_{SR}=1$	左移，$D_{SL}=1$
37～40	100100～100111	右移，$D_{SR}=0$	左移，$D_{SL}=0$
41～44	101000～101011	右移，$D_{SR}=1$	左移，$D_{SL}=1$
45～48	101100～101111	右移，$D_{SR}=0$	左移，$D_{SL}=0$
49～52	110000～110011	左移，$D_{SL}=1$	左移，$D_{SL}=1$
53	110100	清零	清零
54～57	110101～111000	左移，$D_{SL}=1$	左移，$D_{SL}=1$
58	111001	清零	清零
59～62	111010～111101	左移，$D_{SL}=1$	左移，$D_{SL}=1$
63～64	111110～111111	清零	清零

表 10 - 11　　74X194 的功能表

输入										输出				工作模式
清零	控制		串行输入		时钟	并行输入								
CR	S_1	S_0	D_{SL}	D_{SR}	CP	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3	
0	×	×	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	0	0	×	×	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n	保持
1	0	1	×	1	↑	×	×	×	×	1	Q_0^n	Q_1^n	Q_2^n	右移 D_{SR}为串行输入， Q_3 为串行输出
			×	0	↑	×	×	×	×	0	Q_0^n	Q_1^n	Q_2^n	
1	1	0	1	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	1	左移 D_{SL}为串行输入， Q_0 为串行输出
			0	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	0	
1	1	1	×	×	↑	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*	并行置数

表 10 - 12　　控制器的输出信号真值表

节拍	控制器的状态信号							控制器的输出信号								
	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	CLKCTRL	$S_{1(2)}$	$S_{0(2)}$	$D_{SL(2)}$	$D_{SR(2)}$	$S_{1(1)}$	$S_{0(1)}$	$D_{SL(1)}$	$D_{SR(1)}$
1	0	0	0	0	0	0	0	0	1	0	1	×	1	0	0	×
2	0	0	0	0	0	0	1	0	1	0	1	×	1	0	0	×
3	0	0	0	0	0	1	0	0	1	0	1	×	1	0	0	×
4	0	0	0	0	0	1	1	0	1	0	1	×	1	0	0	×
5	0	0	0	0	1	0	0	0	1	0	1	×	1	0	1	×
6	0	0	0	0	1	0	1	0	1	0	1	×	1	0	1	×
7	0	0	0	0	1	1	0	0	1	0	1	×	1	0	1	×
8	0	0	0	0	1	1	1	0	1	0	1	×	1	0	1	×
9	0	0	0	1	0	0	0	0	0	1	×	1	0	1	×	0
10	0	0	0	1	0	0	1	0	0	1	×	1	0	1	×	0
11	0	0	0	1	0	1	0	0	0	1	×	1	0	1	×	0
12	0	0	0	1	0	1	1	0	0	1	×	1	0	1	×	0
13	0	0	0	1	1	0	0	0	0	1	×	0	0	1	×	0
14	0	0	0	1	1	0	1	0	0	1	×	0	0	1	×	0
15	0	0	0	1	1	1	0	0	0	1	×	0	0	1	×	0
16	0	0	0	1	1	1	1	0	0	1	×	0	0	1	×	0
17	0	0	1	0	0	0	0	0	1	0	1	×	1	0	0	×
18	0	0	1	0	0	0	1	0	1	0	1	×	1	0	0	×
19	0	0	1	0	0	1	0	0	1	0	1	×	1	0	0	×
20	0	0	1	0	0	1	1	0	1	0	1	×	1	0	0	×
21	0	0	1	0	1	0	0	0	1	0	1	×	1	0	1	×
22	0	0	1	0	1	0	1	0	1	0	1	×	1	0	1	×
23	0	0	1	0	1	1	0	0	1	0	1	×	1	0	1	×
24	0	0	1	0	1	1	1	0	1	0	1	×	1	0	1	×
25	0	0	1	1	0	0	0	0	0	1	×	1	0	1	×	0
26	0	0	1	1	0	0	1	0	0	1	×	1	0	1	×	0
27	0	0	1	1	0	1	0	0	0	1	×	1	0	1	×	0
28	0	0	1	1	0	1	1	0	0	1	×	1	0	1	×	0
29	0	0	1	1	1	0	0	0	0	1	×	0	0	1	×	0
30	0	0	1	1	1	0	1	0	0	1	×	0	0	1	×	0
31	0	0	1	1	1	1	0	0	0	1	×	0	0	1	×	0
32	0	0	1	1	1	1	1	0	0	1	×	0	0	1	×	0
33	0	1	0	0	0	0	0	0	0	1	×	1	1	0	1	×

续表

节拍	控制器的状态信号							控制器的输出信号								
	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	CLKCTRL	$S_{1(2)}$	$S_{0(2)}$	$D_{SL(2)}$	$D_{SR(2)}$	$S_{1(1)}$	$S_{0(1)}$	$D_{SL(1)}$	$D_{SR(1)}$
34	0	1	0	0	0	0	1	0	0	1	×	1	1	0	1	×
35	0	1	0	0	0	1	0	0	0	1	×	1	1	0	1	×
36	0	1	0	0	0	1	1	0	0	1	×	1	1	0	1	×
37	0	1	0	0	1	0	0	0	0	1	×	0	1	0	0	×
38	0	1	0	0	1	0	1	0	0	1	×	0	1	0	0	×
39	0	1	0	0	1	1	0	0	0	1	×	0	1	0	0	×
40	0	1	0	0	1	1	1	0	0	1	×	0	1	0	0	×
41	0	1	0	1	0	0	0	0	0	1	×	1	1	0	1	×
42	0	1	0	1	0	0	1	0	0	1	×	1	1	0	1	×
43	0	1	0	1	0	1	0	0	0	1	×	1	1	0	1	×
44	0	1	0	1	0	1	1	0	0	1	×	1	1	0	1	×
45	0	1	0	1	1	0	0	0	0	1	×	0	1	0	0	×
46	0	1	0	1	1	0	1	0	0	1	×	0	1	0	0	×
47	0	1	0	1	1	1	0	0	0	1	×	0	1	0	0	×
48	0	1	0	1	1	1	1	0	0	1	×	0	1	0	0	×
49	0	1	1	0	0	0	0	0	1	0	1	×	1	0	1	×
50	0	1	1	0	0	0	1	0	1	0	1	×	1	0	1	×
51	0	1	1	0	0	1	0	0	1	0	1	×	1	0	1	×
52	0	1	1	0	0	1	1	0	1	0	1	×	1	0	1	×
53	0	1	1	0	1	0	0	0	1	1	×	×	1	1	×	×
54	0	1	1	0	1	0	1	0	1	0	1	×	1	0	1	×
55	0	1	1	0	1	1	0	0	1	0	1	×	1	0	1	×
56	0	1	1	0	1	1	1	0	1	0	1	×	1	0	1	×
57	0	1	1	1	0	0	0	0	1	0	1	×	1	0	1	×
58	0	1	1	1	0	0	1	0	1	1	×	×	1	1	×	×
59	0	1	1	1	0	1	0	0	1	0	1	×	1	0	1	×
60	0	1	1	1	0	1	1	0	1	0	1	×	1	0	1	×
61	0	1	1	1	1	0	0	0	1	0	1	×	1	0	1	×
62	0	1	1	1	1	0	1	0	1	0	1	×	1	0	1	×
63	0	1	1	1	1	1	0	0	1	1	×	×	1	1	×	×
64	0	1	1	1	1	1	1	0	1	1	×	×	1	1	×	×
65	1	0	0	0	0	0	0	1	1	0	1	×	1	0	0	×
66	1	0	0	0	0	0	1	1	1	0	1	×	1	0	0	×

续表

节拍	控制器的状态信号							控制器的输出信号								
	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	CLKCTRL	$S_{1(2)}$	$S_{0(2)}$	$D_{SL(2)}$	$D_{SR(2)}$	$S_{1(1)}$	$S_{0(1)}$	$D_{SL(1)}$	$D_{SR(1)}$
67	1	0	0	0	0	1	0	1	1	0	1	×	1	0	0	×
68	1	0	0	0	0	1	1	1	1	0	1	×	1	0	0	×
69	1	0	0	0	1	0	0	1	1	0	1	×	1	0	1	×
70	1	0	0	0	1	0	1	1	1	0	1	×	1	0	1	×
71	1	0	0	0	1	1	0	1	1	0	1	×	1	0	1	×
72	1	0	0	0	1	1	1	1	1	0	1	×	1	0	1	×
73	1	0	0	1	0	0	0	1	0	1	×	1	0	1	×	0
74	1	0	0	1	0	0	1	1	0	1	×	1	0	1	×	0
75	1	0	0	1	0	1	0	1	0	1	×	1	0	1	×	0
76	1	0	0	1	0	1	1	1	0	1	×	1	0	1	×	0
77	1	0	0	1	1	0	0	1	0	1	×	0	0	1	×	0
78	1	0	0	1	1	0	1	1	0	1	×	0	0	1	×	0
79	1	0	0	1	1	1	0	1	0	1	×	0	0	1	×	0
80	1	0	0	1	1	1	1	1	0	1	×	0	0	1	×	0
81	1	0	1	0	0	0	0	1	1	0	1	×	1	0	0	×
82	1	0	1	0	0	0	1	1	1	0	1	×	1	0	0	×
83	1	0	1	0	0	1	0	1	1	0	1	×	1	0	0	×
84	1	0	1	0	0	1	1	1	1	0	1	×	1	0	0	×
85	1	0	1	0	1	0	0	1	1	0	1	×	1	0	1	×
86	1	0	1	0	1	0	1	1	1	0	1	×	1	0	1	×
87	1	0	1	0	1	1	0	1	1	0	1	×	1	0	1	×
88	1	0	1	0	1	1	1	1	1	0	1	×	1	0	1	×
89	1	0	1	1	0	0	0	1	0	1	×	1	0	1	×	0
90	1	0	1	1	0	0	1	1	0	1	×	1	0	1	×	0
91	1	0	1	1	0	1	0	1	0	1	×	1	0	1	×	0
92	1	0	1	1	0	1	1	1	0	1	×	1	0	1	×	0
93	1	0	1	1	1	0	0	1	0	1	×	0	0	1	×	0
94	1	0	1	1	1	0	1	1	0	1	×	0	0	1	×	0
95	1	0	1	1	1	1	0	1	0	1	×	0	0	1	×	0
96	1	0	1	1	1	1	1	1	0	1	×	0	0	1	×	0
97	1	1	0	0	0	0	0	1	0	1	×	1	1	0	1	×
98	1	1	0	0	0	0	1	1	0	1	×	1	1	0	1	×
99	1	1	0	0	0	1	0	1	0	1	×	1	1	0	1	×

续表

节拍	控制器的状态信号							控制器的输出信号								
	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	CLKCTRL	$S_{1(2)}$	$S_{0(2)}$	$D_{SL(2)}$	$D_{SR(2)}$	$S_{1(1)}$	$S_{0(1)}$	$D_{SL(1)}$	$D_{SR(1)}$
100	1	1	0	0	0	1	1	1	0	1	×	1	1	0	1	×
101	1	1	0	0	1	0	0	1	0	1	×	0	1	0	0	×
102	1	1	0	0	1	0	1	1	0	1	×	0	1	0	0	×
103	1	1	0	0	1	1	0	1	0	1	×	0	1	0	0	×
104	1	1	0	0	1	1	1	1	0	1	×	0	1	0	0	×
105	1	1	0	1	0	0	0	1	0	1	×	1	1	0	1	×
106	1	1	0	1	0	0	1	1	0	1	×	1	1	0	1	×
107	1	1	0	1	0	1	0	1	0	1	×	1	1	0	1	×
108	1	1	0	1	0	1	1	1	0	1	×	1	1	0	1	×
109	1	1	0	1	1	0	0	1	0	1	×	0	1	0	0	×
110	1	1	0	1	1	0	1	1	0	1	×	0	1	0	0	×
111	1	1	0	1	1	1	0	1	0	1	×	0	1	0	0	×
112	1	1	0	1	1	1	1	1	0	1	×	0	1	0	0	×
113	1	1	1	0	0	0	0	1	1	0	1	×	1	0	1	×
114	1	1	1	0	0	0	1	1	1	0	1	×	1	0	1	×
115	1	1	1	0	0	1	0	1	1	0	1	×	1	0	1	×
116	1	1	1	0	0	1	1	1	1	0	1	×	1	0	1	×
117	1	1	1	0	1	0	0	1	1	1	×	×	1	1	×	×
118	1	1	1	0	1	0	1	1	1	0	1	×	1	0	1	×
119	1	1	1	0	1	1	0	1	1	0	1	×	1	0	1	×
120	1	1	1	0	1	1	1	1	1	0	1	×	1	0	1	×
121	1	1	1	1	0	0	0	1	1	0	1	×	1	0	1	×
122	1	1	1	1	0	0	1	1	1	1	×	×	1	1	×	×
123	1	1	1	1	0	1	0	1	1	0	1	×	1	0	1	×
124	1	1	1	1	0	1	1	1	1	0	1	×	1	0	1	×
125	1	1	1	1	1	0	0	1	1	0	1	×	1	0	1	×
126	1	1	1	1	1	0	1	1	1	0	1	×	1	0	1	×
127	1	1	1	1	1	1	0	1	1	1	×	×	1	1	×	×
128	1	1	1	1	1	1	1	1	1	1	×	×	1	1	×	×

由此可分别求出控制器的各输出信号的逻辑表达式。图 10－28 中，CLKCTRL＝Q_6，高位移位寄存器 74X194（2）芯片功能控制信号 $S_{1(2)}$、$S_{0(2)}$的设计过程如下：

观察表 10－12 可以看出，$S_{1(2)}$、$S_{0(2)}$与 Q_6 信号无关，因此 $S_{1(2)}$、$S_{0(2)}$是$Q_5Q_4Q_3Q_2Q_1Q_0$的函数。分别画出 $S_{1(2)}$的六变量卡诺图和 $S_{0(2)}$的六变量卡诺图，然后进行化简。图 10－29 是 $S_{1(2)}$的六变量卡诺图。图中用四张四变量卡诺图来表示一个六变量函数，每张卡诺图对应于 Q_5、Q_4 的一组取值，左右或上下两张卡诺图的对应方框可以合并，从而消掉 Q_5 或Q_4 变

量。图 10 - 30 是 $S_{0(2)}$ 的六变量卡诺图。

$S_{1(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=00$

$S_{1(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=01$

$S_{1(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=10$

$S_{1(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=11$

图 10 - 29　高位移位寄存器 74X194（2）号芯片 $S_{1(2)}$ 信号的卡诺图

$S_{0(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=00$

$S_{0(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=01$

$S_{0(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=10$

$S_{0(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	0	1	1
10	0	1	0	0

$Q_5Q_4=11$

图 10 - 30　高位移位寄存器 74X194（2）号芯片 $S_{0(2)}$ 信号的卡诺图

从图 10 - 29 得出 $S_{1(2)}$ 的逻辑表达式为

$$S_{1(2)}=\overline{Q}_5\overline{Q}_3+Q_5Q_4$$

从图 10 - 30 得出 $S_{0(2)}$ 的逻辑表达式为

$$S_{0(2)}=\overline{Q}_5Q_3+Q_5\overline{Q_4}+Q_3\overline{Q_2}\,\overline{Q_1}Q_0+Q_3Q_2Q_1+Q_5\overline{Q_3}Q_2\overline{Q_1}\,\overline{Q_0}$$

观察表 10 - 12 可以看出，$D_{SL(2)}$、$D_{SR(2)}$ 与 Q_6 信号无关，因此 $D_{SL(2)}$、$D_{SR(2)}$ 是 $Q_5Q_4Q_3Q_2Q_1Q_0$ 的函数，分别画出 $D_{SL(2)}$ 的六变量卡诺图和 $D_{SR(2)}$ 的六变量卡诺图，然后进行化简。

图 10-31 是求 $D_{SL(2)}$ 的六变量卡诺图，图中用四张四变量卡诺图来表示一个六变量函数，每张卡诺图对应于 Q_5、Q_4 的一组取值。左右或上下两张卡诺图的对应方框可以合并，从而消掉 Q_5 或 Q_4 变量。图 10-32 是求 $D_{SR(2)}$ 的六变量卡诺图。

$D_{SL(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=00$

$D_{SL(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=01$

$D_{SL(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=10$

$D_{SL(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	×	1	1	1
11	1	1	×	×
10	1	×	1	1

$Q_5Q_4=11$

图 10-31 $D_{SL(2)}$ 信号的卡诺图

从图 10-31 得出 $D_{SL(2)}$ 的逻辑表达式为

$$D_{SL(2)}=1$$

从图 10-32 得出 $D_{SR(2)}$ 的逻辑表达式为

$$D_{SR(2)}=\overline{Q_2}$$

$D_{SR(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	0	0	0	0
10	1	1	1	1

$Q_5Q_4=00$

$D_{SR(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	0	0	0	0
10	1	1	1	1

$Q_5Q_4=01$

$D_{SR(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$Q_5Q_4=10$

$D_{SR(2)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=11$

图 10-32 $D_{SR(2)}$ 信号的卡诺图

观察表 10 - 12 可以看出，$S_{1(1)}$、$S_{0(1)}$ 与 Q_6 信号无关，因此 $S_{1(1)}$、$S_{0(1)}$ 是 $Q_5Q_4Q_3Q_2Q_1Q_0$ 的函数，分别画出 $S_{1(1)}$ 的六变量卡诺图和 $S_{0(1)}$ 的六变量卡诺图，然后进行化简。图 10 - 33 是求 $S_{1(1)}$ 的六变量卡诺图。图 10 - 33 中用四张四变量卡诺图来表示一个六变量函数，每张卡诺图对应于 Q_5、Q_4 的一组取值。左右或上下两张卡诺图的对应方框可以合并，从而消掉 Q_5 或 Q_4 变量。图 10 - 34 是求 $S_{0(1)}$ 的六变量卡诺图。

$S_{1(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=00$

$S_{1(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=01$

$S_{1(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=10$

$S_{1(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=11$

图 10 - 33　$S_{1(1)}$ 信号的卡诺图

从图 10 - 33 得出 $S_{1(1)}$ 的逻辑表达式为

$$S_{1(1)}=\overline{Q_3}+Q_5$$

$S_{0(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=00$

$S_{0(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$Q_5Q_4=01$

$S_{0(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=10$

$S_{0(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	0	1	1
10	0	1	0	0

$Q_5Q_4=11$

图 10 - 34　$S_{0(1)}$ 信号的卡诺图

从图 10-34 得出 $S_{0(1)}$ 的逻辑表达式为

$$S_{0(1)} = \overline{Q}_5 Q_3 + Q_5 Q_4 \overline{Q_3} Q_2 \overline{Q_1}\,\overline{Q_0} + Q_4 Q_3 \overline{Q_2}\,\overline{Q_1} Q_0 + Q_4 Q_3 Q_2 Q_1$$

观察表 10-12 可以看出，$D_{SL(1)}$、$D_{SR(1)}$ 与 Q_6 信号无关，因此 $D_{SL(1)}$、$D_{SR(1)}$ 是 $Q_5 Q_4 Q_3 Q_2 Q_1 Q_0$ 的函数，分别画出 $D_{SL(1)}$ 的六变量卡诺图和 $D_{SR(1)}$ 的六变量卡诺图，然后进行化简，是求 $D_{SL(1)}$ 的六变量卡诺图。图中用四张四变量卡诺图来表示一个六变量函数，每张卡诺图对应于 Q_5、Q_4 的一组取值。左右或上下两张卡诺图的对应方框可以合并，从而消掉 Q_5 或 Q_4 变量。图 10-36 是求 $D_{SR(1)}$ 的六变量卡诺图。

$D_{SL(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=00$

$D_{SL(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=01$

$D_{SL(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$Q_5Q_4=10$

$D_{SL(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	1
01	×	1	1	1
11	1	×	1	×
10	1	1	×	×

$Q_5Q_4=11$

图 10-35 $D_{SL(1)}$ 信号的卡诺图

$D_{SR(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=00$

$D_{SR(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	0	0	0	0
10	0	0	0	0

$Q_5Q_4=01$

$D_{SR(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=10$

$D_{SR(1)}$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10	×	×	×	×

$Q_5Q_4=11$

图 10-36 $D_{SR(1)}$ 信号的卡诺图

从图 10 - 35 得出 $D_{SL(1)}$ 的逻辑表达式为

$$D_{SL(1)}=\overline{Q}_5Q_2+Q_5Q_4+Q_5\overline{Q_2}$$

从图 10 - 36 得出 $D_{SR(1)}$ 的逻辑表达式为

$$D_{SR(1)}=0$$

在理解前面所述内容的基础上，不难画出完整的电路图，此部分内容请读者自行完成。

第六节　数字系统设计举例之四

在进行智力竞赛抢答题比赛时，各参赛者考虑好后都想抢先回答，如果没有合适的设备，主持人将难以分辨出抢答者的先后顺序，为了使比赛能顺利进行，需要有一个能判断抢答者的先后顺序的设备，称为智力竞赛抢答器。本题目要求设计一个四路抢答器，可容纳 4 个队参加比赛。

一、分析题目，明确设计要求

（一）系统组成

系统应包括主持人控制台、选手抢答台、选手计分器等部分。主持人通过主持人控制台上所设按钮向选手抢答台和选手计分器发送抢答允许、加分、减分等信号；抢答选手通过选手抢答台上所设按钮发送抢答信号并封锁竞争对手的抢答器，同时向自己的计分器发出计分允许信号。

（二）功能描述

第一步：主持人预置每组 100 分。

第二步：主持人按准备按钮，宣布准备抢答并述题。

第三步：判断有无违章提前抢答者。如果有违章提前抢答者（选手抢答台红灯亮并发出声音），则宣布本题失败并做违章处理（如减分或口头警告），然后返回第二步。如果没有违章提前抢答者，则进行第四步。

第四步：主持人按开始抢答按钮，宣布开始抢答。

第五步：选手按自己的选手抢答台上所设的抢答按钮抢答。抢到答题权的选手抢答台上所设的绿灯发亮并发出声音，经主持人允许后答题。

第六步：主持人宣布选手得分，并按加 10 分按钮给选手加 10 分或者按减 10 分按钮给选手减 10 分。

第七步：转入第二步进行下一题，或宣布抢答活动结束。

二、主持人控制台功能规划和电路设计

（一）功能规划

1. 主持人控制台状态和所需器件

主持人控制台只有准备状态 S_0 和抢答开始状态 S_1，故只需一个状态触发器 FF0。此外需要预置 100 分按钮（L）、准备按钮（R）、抢答开始按钮（S）、加 10 分按钮（U）、减 10 分按钮（D）。

2. 按钮功能规定和信号形式

L：同时将 4 个选手的计分器置成 100 分，选用带低有效预置端的计数器构成计分器，故信号形式为负脉冲。

R：将本台状态触发器 FF0 复位到 S_0 状态，同时将 4 个选手抢答台置于初始状态，信号形式为负脉冲。

S：将本台状态触发器 FF0 置位到 S_1 状态，再由 FF0 向 4 个选手抢答台输出抢答允许信号。信号形式为负脉冲。

U：给竞答选手加 10 分，信号形式为负脉冲。

D：给竞答选手减 10 分，信号形式为负脉冲。

（二）电路设计

本设计使用 Multisim2001 软件中提供的元件来实现。FF0 选用带低电平预置和清除的 D 触发器，将 D 端和 CP 端接地，作为 RS 锁存器使用。如果没有按钮，也可以改用单刀双掷开关构成，每次操作必须连按两次才可模仿一个按钮。由此得出电路设计如图 10-37 所示。由于 FF0 在上电仿真时会自动进入置位状态 S_1，故在电路中加入 0.1s 的延迟继电器使 FF0 上电复位。

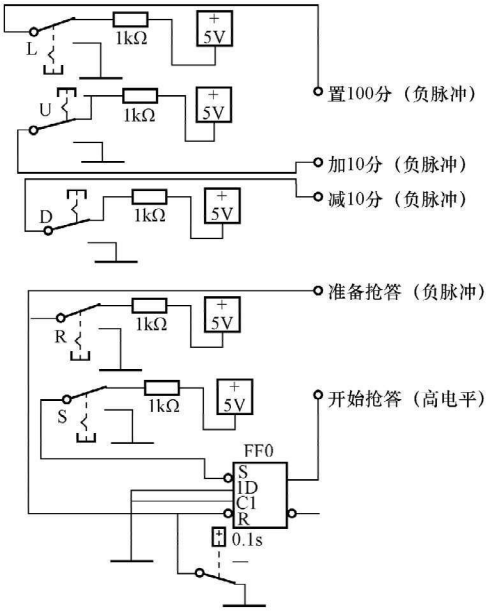


图 10-37 主持人控制台电路

三、选手抢答器功能规划和电路设计

（一）选手抢答器的输入信号和输出信号及其含义

1. 选手抢答器输入信号

（1）主持人控制台送来的准备抢答信号，为负脉冲，它将选手抢答器复位到初始状态；

（2）主持人控制台送来的开始抢答信号，允许竞答选手按抢答按钮，设为 X_1，1 有效；

（3）其他选手抢答器送来的封锁信号，使本选手不能抢答，设其组合后为 X_0，0 封锁；

（4）竞答选手按抢答按钮形成的抢答信号，在电路中作为时钟信号 CP，为负脉冲。

2. 选手抢答器输出信号

（1）违章抢答信号，用于点亮红灯并使蜂鸣器发声，设为 Y_1，1 有效；

（2）抢答有效信号，用于点亮绿灯，设为 Y_2，1 有效；

（3）计分允许信号，使主持人能对本选手加分或减分，设为 Y_3，1 有效；

（4）输出封锁信号，使其他竞答选手按抢答按钮无效，设为 Y_4，1 有效。

（二）选手抢答器的状态规划和原始状态图

选手抢答器应至少具有三个状态：其一为初始状态 S_0；其二为违章抢答后状态 S_1；其三为有效抢答后状态 S_2。采用莫尔设计，根据题意可画出原始状态图如图 10-38 所示。

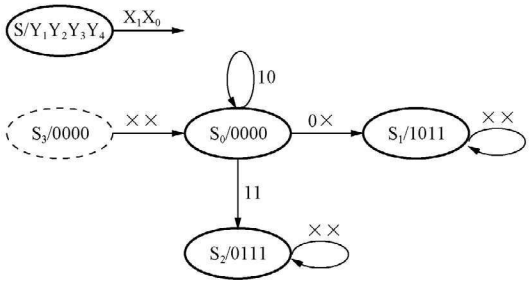


图 10-38 选手抢答器的原始状态图

在准备抢答信号到来后选手抢答器被复位到初始状态 S_0，此时若无抢答允许信号，则选手按抢答按钮为违章抢答，状态转到 S_1 并给出红灯信号、计分允许信号和输出封锁信号。此

后状态停在 S_1 等待被复位。

在状态 S_0 下，若有抢答允许信号并且无输入封锁信号，则选手按抢答按钮有效，状态转到 S_2，同时给出绿灯信号、计分允许信号和输出封锁信号。此后状态停在 S_2 等待被复位；若有抢答允许信号并且有输入封锁信号，则选手按抢答按钮无效，状态停在 S_0 不变，输出也不变。

从图 10-38 可知，计分允许信号和输出封锁信号始终相同，可以合用 Y_3 而取消 Y_4。三个状态的输出各异，已不可化简，需用两个触发器，选用带有低电平有效清除端的负边沿双 JK 触发器 74X112。S_0、S_1、S_2 的编码分别选为 00、01、10，设状态 11 为 S_3，并规定当电路处于 S_3 状态时，在任何输入下其输出全为 0 并且次状态为 S_0。由此得驱动函数和输出函数真值表如表 10-13 所示。

表 10-13　　驱动函数和输出函数真值表

$Q_1^n\ Q_0^n$	X_1X_0	$Q_1^{n+1}Q_0^{n+1}$	$J_1K_1J_0K_0$	$Y_1Y_2Y_3$
0 0	0 ×	0 1	0 × 1 ×	0 0 0
	1 0	0 0	0 × 0 ×	
	1 1	1 0	1 × 0 ×	
0 1	× ×	0 1	0 ×× 0	1 0 1
1 0	× ×	1 0	× 0 0 ×	0 1 1
1 1	× ×	0 0	× 1 × 1	0 0 0

由表 10-13 可知，Y_1、Y_2、Y_3 仅与状态 Q_1、Q_0 有关，可直接写出其表达式，即

$$Y_1=\overline{Q_1^n}\cdot Q_0^n$$

$$Y_2=Q_1^n\cdot\overline{Q_0^n}$$

$$Y_3=Y_1+Y_2=Q_1^n\oplus Q_0^n$$

各驱动函数的卡诺图如图 10-39 所示。

从卡诺图得到逻辑表达式为

$$J_1=\overline{Q_0^n}\cdot X_1\cdot X_0$$

$$K_1=Q_0^n$$

$$J_0=\overline{X_1}\cdot\overline{Q_1^n}$$

$$K_0=Q_1^n$$

输入封锁信号 X_0 为 0 时起封锁作用，其他选手抢答器中任一个输出封锁信号为 1 时，X_0 都应为 0，故可用或非门实现。

根据上述分析设计电路的原理图如图 10-40 所示。图中输入为 1J、1K 的触发器标注为 FF0，图中输入为 2J、2K 的触发器标注为 FF1。由于同样的电路需要 4 套，全画出来将使总图太大，故而可用 Multisim2001 软件所提供的生成子电路的方法将其生成一个小的模块，如图 10-41 所示，以使电路总图简洁并便于仿真。注意生成子电路后一定要记住各引出脚的含义，以便和总图中的其他电路正确连接。

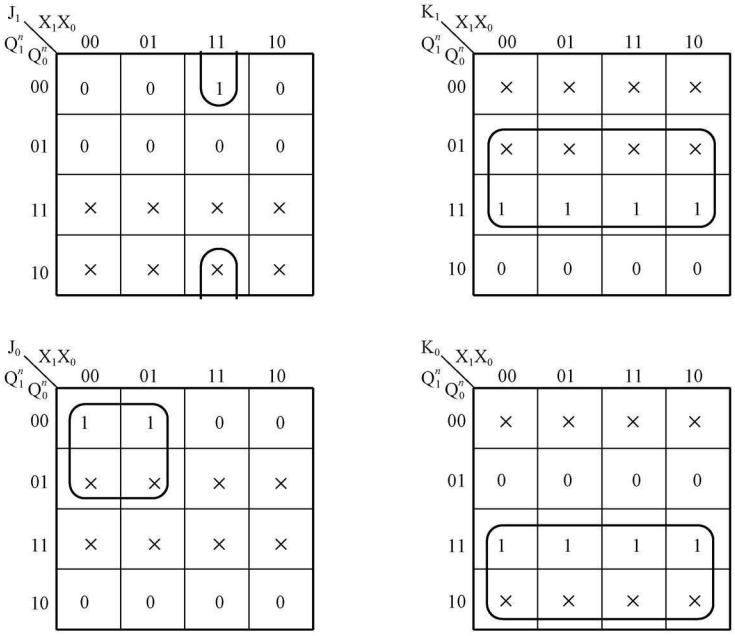


图 10-39 驱动函数的卡诺图

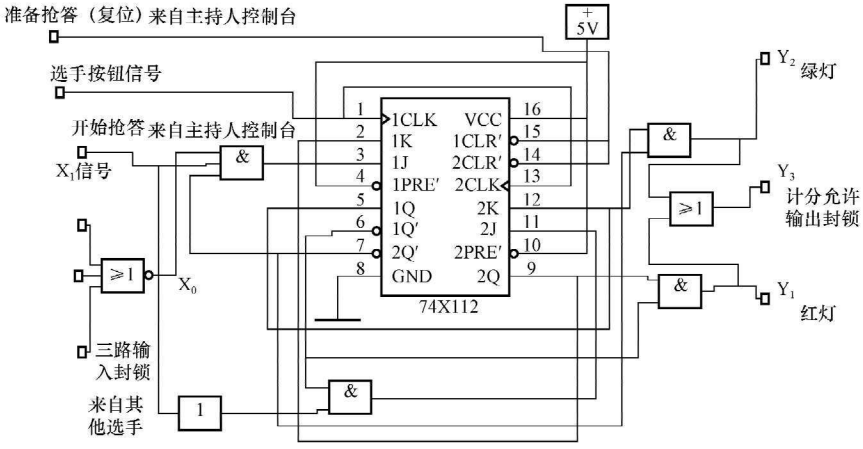


图 10-40 选手抢答器电路原理图

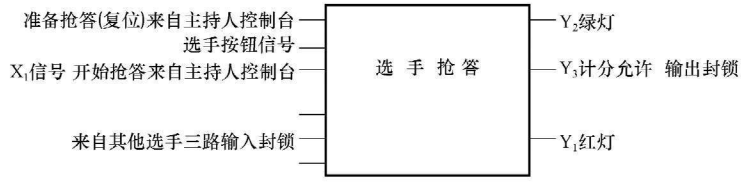


图 10-41 选手抢答器电路接口信号示意图

四、计分器功能和电路设计

（一）计分器应具备的基本功能

（1）计数范围为 000～990，以 10 为最小计数单位；

（2）采用 3 位七段显示数码管显示计分值；

（3）可用预置脉冲将计数值预置成 100 分；

（4）具有计分允许信号输入端，该信号有效时方可对本计分器进行加分或减分操作；

（5）具有加 10 分脉冲信号和减 10 分脉冲信号输入端。

（二）器件选择

根据计分器的功能要求，应选用具有清除和预置功能的十进制可逆计数器，有 74X168 和 74X192 两种芯片都可以选用。但在 Multisim2001 的数字器件库中只有 74X192，因此确定选用 74X192，其逻辑符号如图 10 - 42 所示。

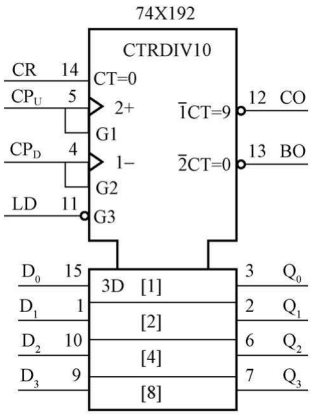


图 10 - 42　74X192 芯片的逻辑符号

根据所选器件可构成计分器电路的逻辑符号连接图，如图 10 - 43 所示，本设计需要 4 套相同电路。为便于读者仿真，计分器电路在 Multisim2001 中的符号连接图如图 10 - 44 所示。

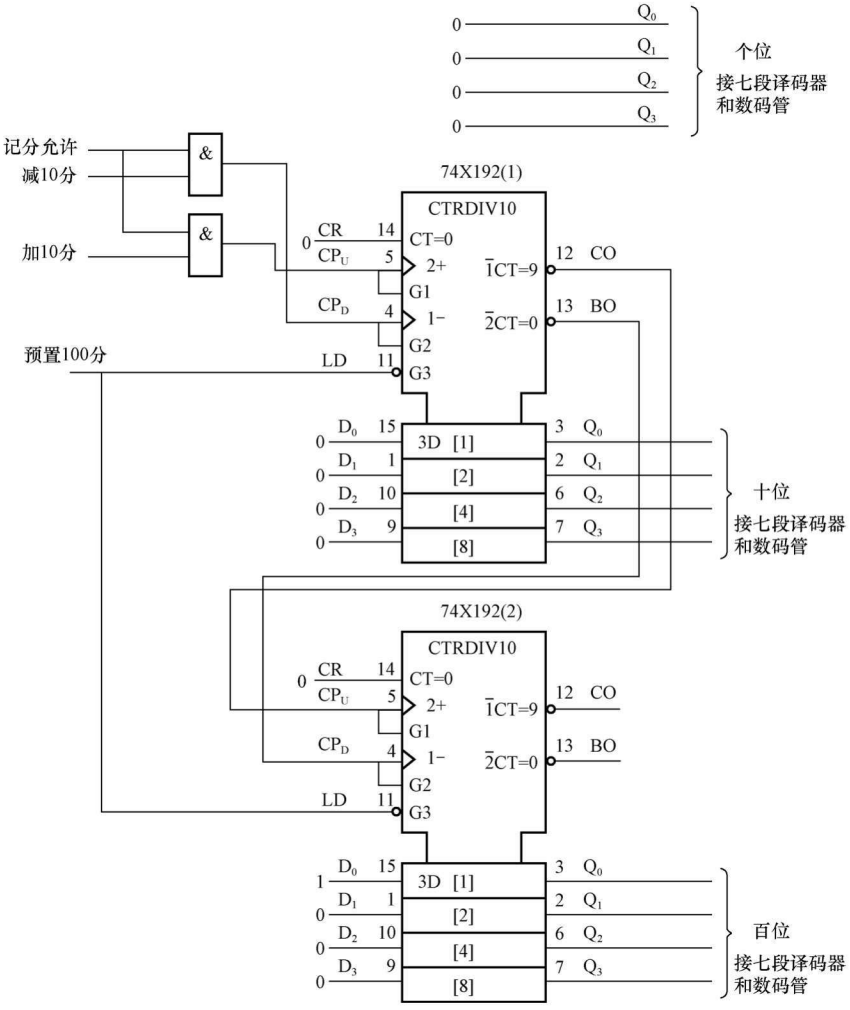


图 10 - 43　计分器电路的逻辑符号连接图

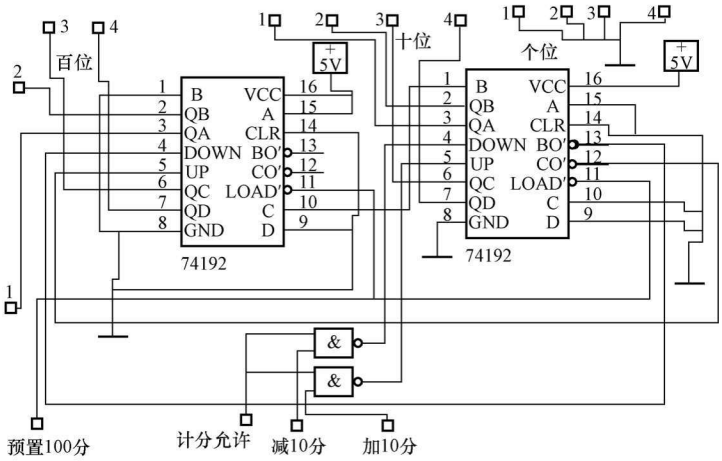


图 10-44 计分器电路在 Multisim2001 中的符号连接图

五、整体连接和调试

在 Multisim2001 环境下将主持人控制台、四路选手抢答器、四路计分器互相正确连接，加入必要的外部元件，如开关、红绿灯、蜂鸣器、数码管等，即可认为基本完成了电路原理图设计，然后便可接通电源进行仿真调试。仿真时应按照设计要求逐条检验，看所设计的电路是否完全具备预想的所有功能。通常一次取得完全成功的可能性不大，这时候需要对原设计做某些调整或改变，直到电路完全满足设计要求。

小　　结

数字系统是一个能完成一系列复杂操作的逻辑单元。数字系统的设计方法有两种，即自上而下的设计方法和自下而上的设计方法。待设计的数字系统划分为控制器和数据通路两部分。数据通路主要完成数据的采集、存储、运算处理和传输，它与外界进行数据交换，而它所有的存取、运算等操作都是在控制器的控制下进行的。数据通路与控制器的联系是接收控制器发出的控制信号，向控制器发出自身的状态信号。控制器是执行算法的核心。控制器的设计可采用前面学过的同步时序逻辑电路的设计方法。算法状态机是一种描述时钟驱动的控制器的工作流程的方法。

本章给出了数字系统设计的四个例子，通过具体实例体会数字系统设计的方法和过程。

习　　题

10-1　说明图 10-45 所示的 ASM 图中有几个状态，几个输入信号，几个输出信号，并说明输出信号是米里型还是莫尔型。

10-2　求二进制数补码电路的 ASM 图如图 10-46 所示，试分析此算法，并分别设计控制器和数据通路。

10-3　设计一个简单的交通灯控制电路，路口示意图如图 10-47 所示，主路没有装配传感器，辅路装配有传感器。要求进行数据通路和控制器的划分，给出 ASM 图，用 VHDL

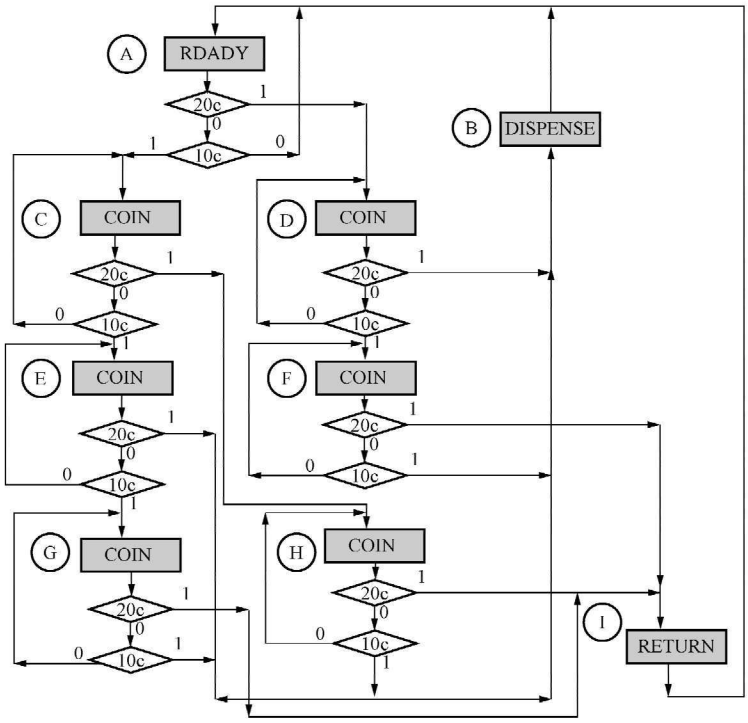


图 10-45　题 10-1 图

描述该电路，并完成该电路的设计。系统要求如下：

（1）交通灯只有红灯和绿灯；

（2）在主路和辅路都没有车辆的情况下，主路一直处于绿灯，辅路一直处于红灯；

（3）辅路上的传感器检测到有车到来时，辅路上的交通灯变成绿灯，并启动一个计时电路，而主路变成红灯；

（4）当规定的时间到达后，不管辅路上是否还有车辆在通行，辅路上的交通灯都将变成红灯，同时主路的交通灯变成绿灯。

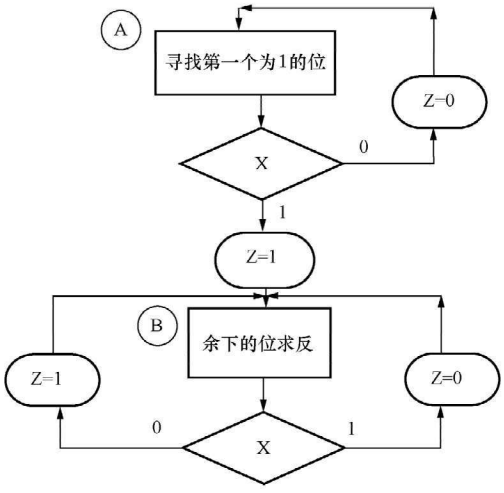


图 10-46　题 10-2 图

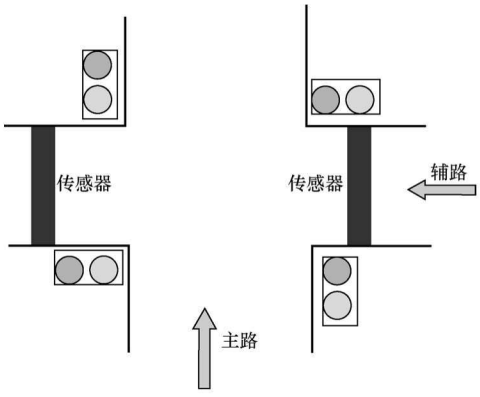


图 10-47　题 10-3 图

10-4 设计一个可以控制计数功能的计数器，当 $M=1$ 时，计数器是 3 位二进制计数器，计数规律为 000，001，010，011，100，101，110，111；当 $M=0$ 时，计数器以 3 位 gray 码顺序计数，计数规律为 000，001，011，010，110，111，101，100。要求：

（1）画出 ASM 图；

（2）用 PLA 和 D 触发器设计该电路。

10-5 设计一个简单的 3 位二进制密码锁，当接收到的密码与预先设定的密码相同时，可以开锁。试回答：

（1）进行数据通路和控制器功能的划分；

（2）画出电路的 ASM 图；

（3）设计该电路；

（4）用 VHDL 语言描述该电路，并仿真验证电路的正确性。

10-6 图 10-48 是某序列检测器的 ASM 图，试分析此 ASM 图的功能，并设计该电路。

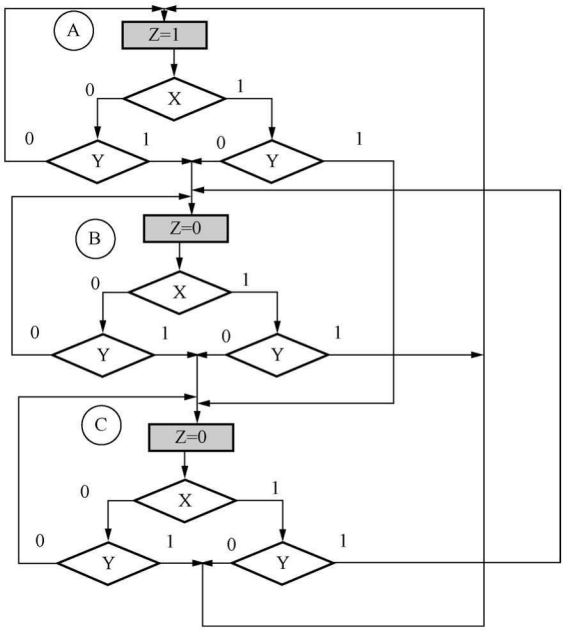


图 10-48 题 10-6 图

10-7 完成本章第四节中无符号数乘法器的数据通路的设计，将整个系统连接起来，并用仿真软件仿真电路的正确性。

参 考 文 献

[1] 康华光，邹寿彬．电子技术基础（数字部分）．4 版．北京：高等教育出版社，2000.
[2] 侯建军，熊华钢，路而红，等．数字电子技术基础．北京：高等教育出版社，2003.
[3] [美] James Bignell 等著．数字电子技术．刘海涛，等译．北京：科学出版社，2005.
[4] [美] Victor P. Nelson 等．数字逻辑电路分析与设计．影印版．北京：清华大学出版社，1997.
[5] 王毓银．数字电路逻辑设计．3 版．北京：高等教育出版社，1999.
[6] 瞿德福．数字电路与模拟电路．北京：中国标准出版社，2004.
[7] 康华光，邹寿彬．电子技术基础（数字部分）．5 版．北京：高等教育出版社，2006.
[8] 边计年，薛宏熙等．数字系统设计自动化．2 版．北京：清华大学出版社，2005.
[9] 薛宏熙，边计年等．数字系统设计自动化．北京：清华大学出版社，1996.
[10] 阎石．数字电子技术基础．2 版．北京：高等教育出版社，1998.
[11] 臧春华，郑步生等．现代电子技术基础（数字部分）．北京：北京航空航天大学出版社，2005.

普通高等教育“十一五”规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUIHUA JIAOCAI

Electrical Engineering

微机原理与接口技术	马　平
走进电世界	李裕能
电路	唐　巍
电路（少学时）	张长富
电工学（上册）	张文生
电工学（下册）	张文生
电工电子技术	李　海
电工电子技术	蔡　旭
电工电子技术（上册）	郑宗亚
电工电子技术（下册）	郑宗亚
电工学实验指导书	娄　娟
数字电子技术基础	王义军
数字电子技术基础	**李月乔**
模拟电子技术基础	韩学军
数字电子技术	朱传琴
模拟电子技术	方舒燕
数字电路实验及课程设计	郁汉琪
工程电磁场（附光盘）	杨宪章
电力电子技术	石新春
电力电子技术	李先允
电力电子技术习题集	李先允
电力市场（第二版）	杜松怀
电力市场营销管理（第二版）	刘秋华
电力工程概论（第二版）	韦　钢
电机与拖动基础（第二版）	刘启新
电机学	曾令全
信号与系统	胡　钋
信号与系统分析	吴　冰
信号分析与处理	杨育霞
数字信号处理	张长森
单片机原理及应用	许　珉
PLC应用技术要点与题解	弭洪涛
电气控制与PLC应用（第二版）	范永胜
电气照明技术（第二版）	谢秀颖
电子信息工程概论	叶树江
电力系统电磁兼容技术	李景禄
电力工程基础	温步瀛
电力系统分析基础	韦　钢

ISBN 978-7-5083-6488-9
9 787508 364889 >
定价：38.00元

[General Information]
SS号=96095950