数字逻辑

谭玉梅 著

哈尔滨地图出版社

数字逻辑

SHUZI LUOJI

谭淑梅　著

哈尔滨地图出版社

·哈尔滨·

图书在版编目(CIP)数据

数字逻辑/谭淑梅著.—哈尔滨:哈尔滨地图出版社,
2005.12
ISBN 7-80717-225-8

Ⅰ.数... Ⅱ.谭... Ⅲ.数字逻辑 Ⅳ.TP302.2

中国版本图书馆 CIP 数据核字(2005)第 154718 号

哈尔滨地图出版社出版、发行
(地址:哈尔滨市南岗区测绘路 2 号　邮政编码:150086)
哈尔滨太平洋彩印有限公司印刷
开本:850 mm×1 168 mm　1/32　印张:8.5　字数:245 千字
2005 年 12 月第 1 版　2005 年 12 月第 1 次印刷
印数:1～1 000　定价:22.00 元

内 容 简 介

本书为计算机专业的广口径教材，它适合各类计算机专业本科教学，经适当剪裁亦可作计算机专业大专参考教材。本教材第一篇为数字逻辑理论，第 1 章至第 3 章主要介绍数字系统逻辑设计的基本知识、基本理论和基本逻辑器件；第 4 章至第 7 章详细讨论了组合逻辑电路和时序逻辑电路的分析与设计方法；第 8 章至第 9 章重点介绍了常用中规模通用集成电路、大规模可编程逻辑器件及其在逻辑设计中的应用。第二篇为数字逻辑实验，共选择了 10 个实验题目。

本书可用作计算机学科应用型本科及高职高专专业教材，也可用作其他非电专业学习数字逻辑电路的基本教材，还可用作专业技术人员学习参考资料。

目　录

第 1 篇　数字逻辑理论

第 1 章　数字逻辑基础

数字逻辑电路研究的主要内容是各种数字逻辑电路的分析与综合。数字逻辑电路采用的计数制是二进制，本章将重点阐述二进制数的表示，此外还介绍几种常见的二—十进制代码及可靠性编码和字符代码。

1.1　概述

1.1.1　数字系统

数字系统是一个能对数字信号进行加工、传递和存储的实体，它由实现各种功能的数字逻辑电路相互连接而成。例如，数字计算机就是一种最具代表性的数字系统。

客观世界存在的各种信号，按其变化规律可以分为两种类型：一类是连续信号，另一类是离散信号。连续信号是指在时间上和数值上均作连续变化的信号，又称为模拟信号。例如，温度、压力等。直接对模拟信号进行处理的电子线路称为模拟电路。若信号的变化在时间上和数值上都是离散的，则称为离散信号。例如，电路开关的状态、脉冲信号等。数字系统讨论的是特殊的离散信号，即数字信号。

1. 数字信号

在两个稳定状态之间作阶跃式变化的信号称为数字信号。有两种表示形式：

电平型：用信号的电位高低表示数字的 0 和 1。

脉冲型：用脉冲的有无表示数字的 0 和 1。

2. 数字逻辑电路

数字逻辑电路是用来处理数字信号的电子线路，对数字信号进行传递、变换、运算、存储及显示等。

(1)功能:对信号进行数值运算,逻辑运算,逻辑判断。

(2) 特点:

①电路的基本工作信号是二值信号;

②电路中的半导体器件一般都工作在开,关状态,对电路进行研究时,主要关心输出和输入之间的逻辑关系;

③电路结构简单,功耗低,便于集成制造和系列化生产;产品价格低廉,使用方便,通用性好;

④数字系统工作速度快,精度高,功能强,可靠性好。

1.1.2 数字逻辑电路的类型和研究方法

1. 数字逻辑电路的类型

根据电路的记忆功能分类:数字电路分为组合逻辑电路和时序逻辑电路。

如果一个逻辑电路在任何时刻的稳定输出仅取决于该时刻的输入,而与电路过去的输入无关,则称为组合逻辑(Combinational Logic)电路。由于这类电路的输出与过去的输入信号无关,所以不需要有记忆功能。

如果一个逻辑电路在任何时刻的稳定输出不仅取决于该时刻的输入,而且与过去的输入相关,则称为时序逻辑(Sequential Logic)电路。由于这类电路的输出与过去的输入相关,所以要有记忆功能,要用电路中记忆元件的状态来反映过去的输入信号。例如,计数器是一个时序逻辑电路。时序逻辑电路按照是否有统一的时钟信号进行控制,又可进一步分为同步时序逻辑电路和异步时序逻辑电路。

2. 数字逻辑电路的研究方法

对数字系统中逻辑电路的研究任务有两个:一是分析,二是设计。对一个现成的数字逻辑电路,研究它的工作性能和逻辑功能称为分析;根据已知的逻辑功能,在给定条件下构造出实现预定功能的逻辑电路称为逻辑设计。

传统方法为用逻辑代数作为基本理论对逻辑电路进行分析和设计,是建立在小规模集成电路基础之上的,它以技术经济指标作为评价一个设计方案优劣的主要性能指标,设计时追求的是如何使一个电路达到最简。因此,在组合逻辑电路设计时,通过对逻辑函数化简,尽可能使电路中的逻辑门和连线数目达到最少。而在时序逻辑电路设计

中，则通过对状态和逻辑函数化简，尽可能使电路中的触发器、逻辑门和连线数目达到最少。但一个最简的方案并不等于一个最佳的方案，最佳的方案应能满足全面性能指标和实际应用要求。所以，对用传统方法设计的电路，往往要根据实际情况进行相应调整。

随着中、大规模集成电路的不断发展，使芯片内容容纳的逻辑器件越来越多，对逻辑电路的设计方法带来了新的方案，如何用各种廉价的中、大规模集成组件去构造满足各种功能的经济合理的电路。要适应这种要求就必须充分了解各种器件的逻辑结构和外部特性，做到合理选择器件，充分利用每一个已选器件的功能，用灵活多变的方法完成各类电路或功能模块的设计。此外，各类或编程逻辑器件(PLD)的出现，给逻辑设计带来了一种全新的方法。人们不再用常规硬线连接的方法去构造电路，而是借助丰富的计算机软件对器件进行编程烧录来实现各种逻辑功能，给逻辑设计带来了极大的方便。

1.2　各种不同的数字系统

数字系统的不同在于每一个数本身基底的不同，基底又称为底数。基底的意义代表着此数为几进制的意思。进制又称为进位。例如 79 的基底为 10。就是指 79 的数为进制。同理 $1011_{(2)}$ 的基底为 2，就是指 1011 的数为二进制，$1011_{(2)}$ 也可写成 $(1011)_2$ 或 10112。数字系统到底有多少种呢？在自然界中数字系统有多种，但严格说起来还是有其限制的，即自然界中的 1 是不能拿来当数字系统的基底，倘若自然数有 N 个，数字系统就有 $N-1$ 种，即 $2,3,4,5,\cdots,N-1,N$。

每一个进制的组成元素到底有几个？又有哪些元素可以组成某进制的任一个数呢？若以 N 进制而言，其组成元素就有 N 个，因为是由 0 开始计数，所以最大一个元素必定是 $N-1$，即 $0,1,2,3,\cdots,N-2,N-1$。例如八进制的组成元素必定有 8 个，即 0,1,2,3,4,5,6,7，因此，八进制的数并不包括 8 这个元素。同理，二进制的组成元素必须只有 2 个，即 0 与 1，也不包括 2 这个元素。

各种不同的数字系统要如何分辨呢？一般均会在数字的下方标示一个较小的数字，以表示几进制的意思。例如 $101101100_{(2)}$ 即表示为

二进制的一个数、$3312_{(4)}$即表示为四进制的一个数、$77251_{(8)}$即表示为八进的一个数、$4975_{(10)}$即表示为十进制的一个数、$15AF_{(16)}$即表示为十六进制的一个数等。此处标示上去的小字即称为基底或底数。若是没有标示通常被视为十进制，因为惟独只有十进制可以容许不标示基底。例如 7629 即表示为十进制的一个数。

底数的标示方法还有另外一种形式，即也可将整个数字用括号括起来，然后在括号外的右下方标示底数。例如$(1011000)_2$，$(3312)_4$，$(77251)_8$，$(4975)_{10}$，$(15AF)_{16}$等。

一般而言，常用数字系统有二进制、八进制、十进制及十六进制等。

在任意一个进制的任意一个数字中，其有效数字最左边的一位为最高位数，它所代表的乘幂值最高。而有效数字最右边的一位为最低位数，它所代表的乘幂值最低。

二进制常被用于电脑内部的数值运算，十进制是人类惯用的数字系统，十六进制则是人类与电脑沟通二进制的简化表示法，即为了辨认的方便而将 4 个二进制数改写成十六进制的一个元素，即二进制数 1001111101100011 可改写成十六进制数为 9F63。至于八进制则是常用作将十进制转换成二进制时的快速桥梁，在十进制转换成二进制时，整数部分需要使用极为耗时的不断除以 2 的运算，而小数部分则使用不断乘以 2 的运算，即是使用冗长的长除长乘转换法来加以转换。若是先将其转换为八进制，然后再以位增减法转换为二进制，可增快其转换速度。

如表 1－1－1 所示为二进制至十八进制间，各种不同进制所拥有的基本元素。基本元素即是指单个位数中有可能出现的所有数字或代表符号。

表 1－1－1　二进制至十八进制间各种不同进制的基本元素

进　制	代表各种不同进制的基本元素																	
二进制	0	1																
三进制	0	1	2															
四进制	0	1	2	3														
五进制	0	1	2	3	4													
六进制	0	1	2	3	4	5												

续表

进制	代表各种不同进制的基本元素																	
七进制	0	1	2	3	4	5	6											
八进制	0	1	2	3	4	5	6	7										
九进制	0	1	2	3	4	5	6	7	8									
十进制	0	1	2	3	4	5	6	7	8	9								
十一进制	0	1	2	3	4	5	6	7	8	9	A							
十二进制	0	1	2	3	4	5	6	7	8	9	A	B						
十三进制	0	1	2	3	4	5	6	7	8	9	A	B	C					
十四进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D				
十五进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E			
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
十七进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	
十八进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H

例题 1.1　1101 属于:

(A)二进制数字系统　　(B)四进制数字系统

(C)十六进制数字系统　　(D)以上都不对

解　因为 1101 的底数省略,所以可能为十进制,但所有答案中并无十进制,所以选(D)以上都不对的答案。

例题 1.2　以下哪一个进制表示法是错误的?

(A)$(4359)_{10}$　　(B)$(2375)_7$

(C)$(4312)_5$　　(D)$(5F9H)_{18}$

解　参考表 1101 二进制至十八进制间各种不同进制的基本元素,可以发现七进制的基本元素中并没有 7,因此只有答(B)$(2375)_7$是错误的,所以答案选(B)。

1.3　二进制的运算

人类不可能利用二进制来做运算,因为二进制的运算对于人类而言实在是太耗时了。由于电脑使用二进制,因此使用二进制来做运算有其方便性,二进制运算的硬件电路极为简单。硬件电路越简单系统

越不容易出问题，倘若系统一旦出了问题，要排除故障也是很简单的。电脑没有人脑般灵活变通，因此为了使运算的结果不易发生错误，使用了最小的一个进制，即二进制做运算，是为求运算结果的正确性。虽然，人类在做二进制运算时极为耗时，但电脑做二进制运算却非常快。

1.3.1 二进制的加法运算

电脑内部的主要运算其实只是加法运算而已，其余的减法、乘法、除法等各种运算都只是加法运算的应用罢了。加法运算的原则如下所示：

例题 1.3 试计算二进制数 01011+10101 的值

解

$$\begin{array}{r} 01011 \\ +10101 \\ \hline 100000 \end{array}$$

例题 1.4 三个二进制数分别为 11011，10011 与 11，那么它们的和是多少？

解

$$\begin{array}{r} 11011 \\ +10011 \\ \hline 101110 \end{array} \longrightarrow \begin{array}{r} 101110 \\ +\quad 11 \\ \hline 110001 \end{array}$$

1.3.2 二进制的减法运算

其实在电脑中二进制的减法运算只不过是二进制加法运算的应用罢了，例如我们可以将 A－B 改成 A＋(－B)来运算，这样就成为加法运算了，但有一点需要特别留意的是，A＋B 的 B 与 A＋(－B)的(－B)是必须加以区分的，否则电脑会将其视为是相同的。因此我们必须将－B 改变成另一种表示方式，即取补数的方式来与 A 做加法运算。

例题 1.5 试以 2 的补数编码做二进制减法运算 $01001110_2-00111111_2$ 的值？

解 $01001110_2-00111111_2$ MSB=0 为正数，所以题意为正数－正数。

$=01001110_2+(-00111111_2)$ 电脑内部会将其改成正数＋负数的形式来计算。

$=01001110_2+11000001_2$ 电脑内部会将负数以 2 的补数方式来表示。

$$\begin{array}{r} 01001110 \\ +11000001 \\ \hline 100001111 \end{array}$$

有进位 1 产生时表示结果为正数，而进位以外的数 00001111 即为所要的结果。

$=00001111_2$

例题 1.6　试以 2 的补数编码做二进制减法运算 $00101110_2-00110111_2$的值？

解　$00101110_2-00110111_2$　MSB=0 为正数，所以题意为正数一正数。

$=00101110_2+(-00110111_2)$　电脑内部会将减法改成加法的形式来计算。

$=00101110_2+11001001_2$　电脑内部会将负数以 2 的补数来表示。

$$\begin{array}{r}0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\\ +)\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\\ \hline 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\end{array}$$

没有进位产生时表示结果为负数，而 11101111 是负数表示法的形式。

$=11101111_2$　负数表示法的形式可用 2 的补数还原成正数的形

$=-(00010001)_2$　式，但须加上负号。

例题 1.7　请以 2 的补数编码的八位数做减法运算，$88_{16}-AB_{16}$的值？

解　$88_{16}-AB_{16}=10001000_2-10101011_2$　MSB=1 为负数表示法，所以题意为负数一负数。

$=10001000_2-(10101011)_2$　负数表示法的形式可用 2 的补数

$=10001000_2-(-01010101)_2$　还原成正数的形式，但须加上负号

$=10001000_2+01010101_2$

$$\begin{array}{r}1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\\ +)\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\\ \hline 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\end{array}$$

没有进位产生时表示结果为负数，而 11011101 是负数表示法的形式。

$=11011101_2$

$=-(00100011)_2$

$=-3510$

1.3.3　二进制的乘法运算

电脑内部是如何完成二进制乘法运算的呢?其实在电脑的内部,二进制乘法运算也只是二进加法运算的扩充而已,只不过借助了一些移位寄存器来协助完成。移位寄存器有左移位与右移位之分,左移位一次等于原值乘以 21,左移位两次等于原值乘以 22,左移位三次等于原值乘以 23,依次类推,左移位 N 次等于原值乘以 $2N$。当然左移位后所遗留出来的空位不管有几个均要补上 0,例如$(11)_2$经过左移位寄存器一次移位之后变为$(110)_2$,左移位一次后的$(110)_2$是 6 而原值$(11)_2$是 3,正好是原值乘以 21。当$(11)_2$经过左移位寄存器二次移位之后变为$(1100)_2$,左移位二次后的$(1100)_2$是 12 而原值$(11)_2$是 3,正好是原值乘以 22。当$(110)_2$经过左移位寄存器三次移位之后变为$(1100)_2$,左移位三次后的$(11000)_2$是 24 而原值的$(11)_2$是 3,正好是原值乘以 23,依次类推,左移位 N 次是原值乘以 $2N$。

当两个二进制的数相乘时,被乘数必须依乘数的每一位所代表的含义分别执行左移位的操作,然后再将每一个左移位后的值相加起来就完成乘法运算了。

例题 1.8　电脑内部如何执行 $111011_{(2)}\times1010_{(2)}$ 的运算?

解　$111011_{(2)}\times1010_{(2)}$

$=111011_{(2)}\times(1000_{(2)}+10_{(2)})$

$=111011_{(2)}\times1000_{(2)}+111011_{(2)}\times10_{(2)}$

$=111011_{(2)}\times(2^3)_{10}+111011_{(2)}\times(2^1)_{10}$

$\downarrow$　　　　　　　　$\downarrow$

左移位三次　　　　左移位一次

后面补三个 0　　　后面补一个 0

$=111011000_{(2)}+1110110_{(2)}$

$=1001001110_{(2)}$

例题 1.9　电脑内部如何运行 9×7 的运算?

解　9×7

$=1001_2\times111_2$

$=1001_2\times(100_2+10_2+1_2)$

$=1001_2\times100_2+1001_2\times10_2+1001_2\times1_2$

$=1001_2\times(2^2)_{10}+1001_2\times(2^1)_{10}+1001_2\times(2^0)_{10}$

↓ 左移位两次 后面补两个 0　　↓ 左移位一次 后面补一个 0　　↓ 不移位

$=100100_2+10010_2+1001_2$

$=111111_2$

$=6310$

1.3.4　二进制的除法运算

电脑内部只有加法运算，若要完成除法运算就必须利用二进制减法(减法运算也是加法运算搭配补数的应用)，须借助于右移位寄存器。在上一小节中我们提到借助于左移位寄存器的应用，电脑可以执行乘法运算。而右移位寄存器的功能恰巧与左移寄存器相反，左移位一次是乘上 2^1 倍，右移位一次则是除以 2^1 倍，左移位二次是乘上 2^2 倍，右移位二次则是除以 2^2 倍，左移位三次是乘上 2^3 倍，右移位三次则是除以 2^3 倍，依次类推，左移位 N 次则是乘上 2^N 倍，右移位 N 次则是除以 2^N 倍。例如 $11001100_{(2)}$ 是十进制 204，右移位一次后变为 $1100110_{(2)}$，$1100110_{(2)}$ 转换为十进制是 102，正好是原值除以 2^1 倍，当 $11001100_{(2)}$ 右移位二次后则变为 $110011_{(2)}$，$110011_{(2)}$ 转换为十进制是 51，正好是原值除以 2^2 倍，当 $11001100_{(2)}$ 右移位三次后变为 $11001.1_{(2)}$，$11001.1_{(2)}$ 转换为十进制是 25.5，正好是原值除以 2^3 倍。

例题 1.10　电脑内部如何运行 $110110_{(2)}\div100_{(2)}$ 的运算呢？

解　$110110_{(2)}\div100_{(2)}$

$=110110_{(2)}\div(2^2)_{10}$

↓ 右移位两次

$=1101.10_{(2)}$

$=13.5_{(10)}$

例题 1.11　电脑内部如何运行 $39\div8$ 的运算呢？

解　$39\div8$

$=100111_2 \div 1000_2$

$=100111_2 \div (2^3)_{10}$

↓

右移位三次

$=100.111_2$

$=4.875_{10}$

1.4 补数

补数有 1 的补数、2 的补数、3 的补数、4 的补数 、……、$N-1$ 的补数、N 的补数。由于每一个自然数 1,2,3,4,…,N 均可以拿来当作补数,因此一般会有一个错误的概念,以为补数有无限多种,其实对于任何一种进制而言补数只有两种。例如,十进制就只有 10 的补数与 9 的补数,八进制就只有 8 的补数与 7 的补数,五进制就只有 5 的补数与 4 的补数,四进制就只有 4 的补数与 3 的补数,二进制就只有 2 的补数与 1 的补数…… 总之,任何一种进制的补数就只有进制本身的补数与进制本身减 1 的补数两种。因此 R 进制就只有 R 的补数与 $R-1$ 的补数 。

例题 1.12 下面哪一个是错误的?

(A)$1011_{(2)}$的 2 的补数为 0101 (B)$1011_{(2)}$的 1 的补数为 0100

(C)$1011_{(4)}$的 3 的补数为 2322 (D)$1011_{(4)}$的 2 的补数为 1211

解 因为四进制只有 4 的补数与 3 的补数两种,并没有 2 的补数,所以答案选(D)。

例题 1.13 每一种进制各自均有几种补数?

(A)两种 (B)一种 (C)无限多种 (D) $R-1$ 种

解 每一种进制各自均有两种补数,例如,四进制只有 4 的补数与 3 的补数两种。三进制只有 3 的补数与 2 的补数两种。二进制只有 2 的补数与 1 的补数两种。所以答案选(A)。

例题 1.14 下面的说法哪种是错误的?

(A)$1011_{(2)}$可以有 2 的补数与 1 的补数

(B)$1011_{(3)}$可以有 3 的补数与 2 的补数