脉冲电路与数字电路

下册 数字电路

赵仕忠 宋玉炎 周文彬 秦国荣 合编

高等教育出版社

本书为《脉冲电路与数字电路》一书的下册(数字电路)。鉴于中、大规模集成电路的应用日益广泛,本书着重对由不同类型、不同集成度数字元件组成的数字系统,进行逻辑功能的分析和逻辑系统的设计。

本书内容包括: 数字电路概述;逻辑函数分析;双极型集成电路;组合电路;触发器; 同步时序电路;异步时序电路;MOS 数字集成电路;数模转换与模数转换。

本书可作为综合大学无线电专业的教材或参考书,也可供其他科技工作者参考。

本书责任编辑: 郭玉凤

脉冲电路与数字电路

下册 数字电路

赵仕忠 宋玉炎 周文彬 秦国荣 合编

*

高等教育出版社出版

新华书店北京发行所发行

二二〇七工厂印装

*

开本 787×1092 1/16 印张 18.75 字数 430 000

1987 年 4 月第 1 版 1987 年 4 月第 1 次印刷

印数 00,001— 3,130

书号 15010·0819 定价 2.75 元

目 录

第一章　数字电路概述

1-1　什么是数字电路

什么是数字电路？要回答这个问题，必须先明确“数字”这个术语的含义。

大家知道，在我们生活的这个世界上，存在许许多多的物理量，如温度、时间和距离等等。这些物理量是随时间作连续变化的。但是在工程应用中，我们往往采用另一种易于传递或处理的物理量（例如电压）来代表原有的温度、时间和距离。这就有了比拟或模拟的概念，因此，在现代的工程技术中，已经习惯用模拟量这一术语来称呼随时间连续变化的物理量。应明确，模拟量是随时间作连续变化的。但是，在会话、数据处理等场合中，遇到的是离散量，离散是指在时间上不连续。相对于模拟量，离散量也有个习惯的名称叫数字量。模拟量和数字量可以相互转换，这就是第九章所要讲的模-数转换和数-模转换的内容。

在了解模拟量和数字量的基础上，下面我们再介绍一下电子电路中的工作信号的概念。

信号（这里指电信号）是运载消息的工具，它的基本形式是随时间变化的电流和电压，所有电子电路中的工作信号，其形状是多样的，但它主要有两种结构：连续时间信号和离散时间信号。

连续时间信号是指在连续时间范围内所定义的信号，如图 1-1 所示。需要注意的是，这里“连续”是指时间，至于信号的幅值可以是连续的，也可以不是连续的，连续时间信号也简称为连续信号或模拟信号，象电话或广播的语言信号以及电视的图像信号等，都是连续信号或模拟信号。我们把工作在模拟信号下的电子电路，统称为模拟电子电路。简称为模拟电路。从这个角度上讲，我们已经学过的脉冲电路，属于模拟电路的范畴。

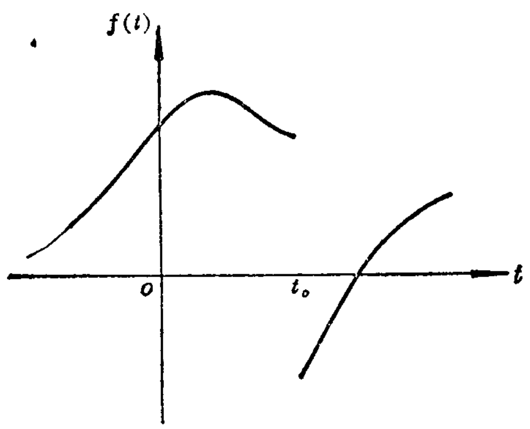

图 1-1　连续信号

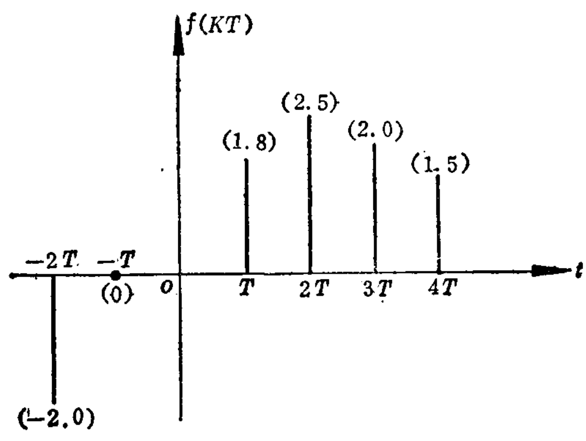

图 1-2　离散信号

离散时间信号是指在一些离散的瞬间才有定义的信号。如图 1-2 所示，信号 $f(KT)$ 仅在 $t=0, \pm T, \pm 2T$, ……才有确定的幅值（图中括号内的数值表示信号的幅值），而在其余时间，信

号 $f(KT)$ 没有定义，这里离散是指时间，至于离散瞬间的间隔，则可以是均匀的，也可以不是均匀的。离散时间信号简称为离散信号，象雷达数据，遥控指令等都是离散信号。至于一般所称的数字信号是指那些在时间上和数值上都是离散的信号，一方面，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；另一方面，它们的数值大小和增减变化，都采取数字的形式。我们把工作在数字信号下的电子电路，称为数字电子电路，或简称为数字电路。

用数字信号表示自然界中的物理量，这是我们早已习惯的事情，如测得室温是25°C。不过，在数字电路中所使用的不是通常的十进制，而是二进制以及由二进制构成的其他进制。这是由于电子电路很容易产生两种稳定状态(如"开"和"关"，高电平和低电平等)，而两种稳定状态只能代表两个数码，在二进制计数中，每一位只有 0 和 1 两个数码，所以正好可以用电路的两种稳定状态来表示。例如用低电平表示 0，高电平表示 1，或者反过来，用高电平表示 0，低电平表示 1。所以，在数字电路中，通常采用二进制计数法。因此，数字电路主要是研究 0 和 1 两种逻辑状态的产生和各种逻辑运算等，数字电路主要是通过逻辑运算完成其它运算功能的，所以它又叫逻辑电路，或数字逻辑电路。

现在，我们可以比较完整地回答开头提出的问题，即数字电路是处理数字信号并能完成数字运算的电路。但应注意，它不能直接对模拟信号进行处理，必须转换成数字信号以后方可。模拟信号与它们的数字信号之间进行转换的电路，也算作数字电路。

1-2 数字电路的特点和本书的研究范围

随着半导体集成电路的迅速发展和数字电子计算机的广泛应用，对数字系统的研究已形成一门新的学科。相对于模拟电路，数字电路有它自己的特点和研究范围。

在模拟电路中，主要研究的是微弱信号的放大以及各种形式信号的产生、交换和反馈等，采用的数学工具是符号法，重点放在稳态特性的分析上。而对于脉冲电路，其重点是研究波形(非正弦波形的产生、变换和传输等)，所用的数学工具是经典法和拉普拉斯变换法等，关心的是波形的瞬态特性。数字电路则不同了，它研究的主要问题是 0 和 1 两种状态的逻辑关系和逻辑设计。应该明确，这里的 0 和 1，并不是通常所表示的数量大小，而是代表完全对立的两个方面，如电位的高和低，信号的有和无等两种截然不同的状态。若用 1 表示高电位，用 0 表示低电位的叫正逻辑，而用 0 表示高电位，用 1 表示低电位的叫负逻辑。以后若无特殊声明，都是对正逻辑而言的。数字电路使用的数学工具是逻辑(布尔)代数、真值表和卡诺图，也就是本书第二章所要讲的内容，它是数字电路课程的基本理论部分，是学习以后各章的基础。

在电路结构方面，数字电路中最基本的单元是逻辑门(第三章的内容)，由逻辑门组成的触发器(第五章的内容)也是数字电路中的常用电路。相比于线性集成电路的基本单元，逻辑门的电路结构比较简单，而且对元件的要求不太严格，允许元件有较大的分散性，允许电源的参数有较大的误差。这是因为逻辑门这种基本单元，只有两个状态，只要能区分开 0 和 1 就足够了，而且可以通过增加二进制数位数的方法，达到很高的精度。而在模拟电路中，为了提高放大电路的精度，不仅需要提高元件的精度和电源的稳定度，往往还需要选用比较复杂的电路。

数字电路除了具有算术运算的功能外，还可以方便地进行各种逻辑推演和逻辑判断。在传统的模拟电路中，要完成准确的运算和逻辑判断的功能，必须使用较为复杂的设备才能办到，这是数字电路的又一特点。逻辑电路可分为两大类，一类是电路输出状态完全由当前的输入状态来决定，而与此电路过去的历史无关，意思是输出只与某些输入的组合有关，所以这类电路叫组合逻辑电路(第四章的内容)。在日常生活中，这种例子很多。拿普通的对字锁来说，它有三个码盘，只有当三位正确的数字组合在一起时，锁才能打开，否则，只要有一位数字不正确，就不能打开。这就是说，只有三个正确的数字同时出现在码盘上时，才有可能打开，而与码盘过去曾经放过的数字无关。另一类是电路的输出状态不仅与当前的输入状态有关，而且与以前各时刻的输入状态有关，意思是输出与输入值的过去和现在的顺序都有关系。所以这类电路叫时序逻辑电路。同样，时序逻辑电路也可以用一种对字锁来比拟，不过这种对字锁只有一个码盘，假设它需要正确的连拨动三次才能打开，那么，第三次拨动的数码输入时，必须要求前两次都送入正确的数码时，锁才能打开。这就是说，最后输出不但和当时的输入(第三个数码)有关，而且还和当时的内部状态(前两次输入的数码)有关。时序逻辑电路又可分为两类，一类叫同步时序逻辑电路(第六章内容)，在这类电路中，有一个统一的同步信号，也叫时钟脉冲，只有在时钟脉冲到来时，电路的状态才能发生改变，而且只改变一次，如果时钟脉冲没有到来，即使输入信号有了改变，它可能会影响到输出，但也决不会改变电路的内部状态；另一类叫异步时序逻辑电路(第七章的内容)，在这类电路中，没有统一的时钟脉冲来同步，电路状态的改变是由输入信号直接引起的。

上面我们从电路结构、电路功能和使用的数学工具三个方面，介绍了数字电路的主要内容和特点，下面我们再从所用的元器件这个角度，作一介绍：

随着集成电路的迅速发展，数字电路所使用的元件，已由使用分立元件发展到广泛使用半导体集成电路了。按电路中晶体管的导电类型，半导体集成电路可分为双极型集成电路和单极型集成电路两种。本书第八章 MOS 集成电路属于单极型集成电路，其余各章的电路，主要是以双极型集成电路为例，但从目前发展的情况来看，MOS 集成电路日益得到广泛的应用。

半导体集成电路，根据在一定大小的一块硅片上包含元件(指晶体管、二极管、电阻或电容)或逻辑门的数量，通常分为以下四种：

小规模集成电路(SSI)，1960 年出现，在一块硅片上包含 10～100 个元件或 1～10 个逻辑门。如集成逻辑门和集成触发器等。

中规模集成电路(MSI)，1966 年出现，在一块硅片上包含 100～1, 000 个元件或 10～100 个逻辑门。如集成计数器、集成寄存器和集成译码器等逻辑部件。

大规模集成电路(LSI)，1970 年出现，在一块硅片上包含 1,000～10,000 个元件或 100～1,000 个逻辑门。如半导体存贮器、某些设备的控制器等逻辑系统。

超大规模集成电路(VLSI)，1975 年出现，在一块硅片上包含 10,000 个以上的元件。如在一块硅片上能集成一个完整的微型计算机。

本书以小规模集成电路为主说明各种功能的数字电路的基本原理和设计方法，在此基础上，也适当地介绍一些中规模和大规模数字集成电路。这里说明一点，提请大家注意，在进行逻辑设

计的时候，其结果不是唯一的，但我们要找的是一个最佳设计，要满足全面的性能指标。例如在完成需要的逻辑功能的前提下，还要求满足价格低，时间省，功耗小，可靠性高等等。就技术经济指标来看，原来要求是所用的逻辑门的个数最少，为了这一目标，已经研究出多种求得最简逻辑结构的方法，可是在今天，集成电路的规模越来越大、可靠性提高、价格降低的新条件下，一味地追求逻辑门的个数最少，并不一定是最佳设计了。现在作为评价设计的指标常常变为追求集成封装(集成块)的数目最少和引出线的总数最少。因为这是决定数字系统可靠性和价格的主要条件。本书的重点，还是以寻求逻辑门数目最少为目标，因为这些方法仍是数字电路的基础。对于在新的工艺条件下，如何寻求最佳设计，只作一些简单的介绍。

目前，数字电路的应用已极为广泛，几乎渗透到了国民经济和人民生活的一切领域之中，如数字通讯、电子数字计算机、自动控制、数字仪表、遥控遥测、导航、人造卫星等。可以相信，随着集成电路技术的进一步发展和完善，数字电路的应用必将得到更快的发展和普及。

1-3　二进制数及数制间的转换

在以后各章的学习中，经常要用到二进制和其它有关数制，所以，我们先简略地介绍二进制数的概念、它的基本运算规则以及数制间的转换关系。

一、二进制数的表示

1. 十进制数

人们最熟悉的是十进制数，它有 0，1，2，……9 十个不同的数码。任何一个数都可由这十个数码中的某几个组合而成，而每一个数码因它在数中的位置不同而代表不同的数量。

例如 34.25 这个数，最高位的 3 是处在 10^1 位，即十位，它表示 3×10，4 处在 10^0 位，即个位，表示它本身的数值。小数点后的第一位的 2 是处在 10^{-1} 位，它表示 2×0.1，5 处在 10^{-2} 位，它表示 5×0.01，因此这个数可写成

$$34.25=3\times10^1+4\times10^0+2\times10^{-1}+5\times10^{-2}$$

由于任何一个十进数 N，都可以表示成

$$N_{10}=K_n10^n+K_{n-1}10^{n-1}+\cdots+K_010^0$$
$$+K_{-1}10^{-1}+K_{-2}10^{-2}+\cdots+K_{-m}10^{-m}=\sum_{i=-m}^{n}K_i\cdot10^i$$

式中 m 和 n 均为正整数，K_i 可以是 0、1、2……9 十个数码中的一个。

这个 10 就称为计数制的基数，对十进制来说，基数是十，顾名思义，十进制即"逢十进一"。

日常生活中还碰到十二进制(如铅笔 12 支为一打)，十六进制(老秤中的十六两为一斤)，六十进制(钟表上六十分为一小时，六十秒为一分)等。

2. 二进制数

二进制数，即基数为二的进位制，它的每一数位只可能取 0 和 1 两个数码，而且是"逢二进一"。为了熟悉二进制数的表示，我们可先考察表 1-1 中列出的若干代表性的十进制数和二进制数的对应关系：

表 1-1 十进制数与二进制数对照表

十进数	二进数	十进数	二进数	十进数	二进数
0	0	16	10000	0.5	0.1
1	1	$(=2^4)$	4 个 0	$(=2^{-1})$	
2	10	32	100000	0.25	0.01
3	11	$(=2^5)$	5 个 0	$(=2^{-2})$	
4	100	64	1000000	0.125	0.001
5	101	$(=2^6)$	6 个 0	$(=2^{-3})$	
6	110	128	10000000	0.0625	0.0001
7	111	$(=2^7)$	7 个 0	$(=2^{-4})$	
8	1000	256	100000000		
9	1001	$(=2^8)$	8 个 0		
10	1010	512	1000000000		
11	1011	$(=2^9)$	9个 0		
12	1100	1024	1000……0		
13	1101	$(=2^{10})$	10个 0		
14	1110	2048	100 …… 0		
15	1111	$(=2^{11})$	11 个 0		

从表 1-1 可以看出，二进制数有如下几个特点:

1. 逢二进一:如由 11 到 100,111 到 1000 等进位的情况。

2. 每向左移一位，数值增加到原来的两倍:如 11 左移一位变成 110，数值由 3 成为 6，10000 左移一位变成 100000 数值由 16 成为 32。

每向右移一位，数值变为原来的二分之一:如 11 右移二位，变成 0.11，数值由 3 成为 0.75。

3. 一个二进制数 1101.101 是下列展开式的缩写

$$1\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$$

即

$$(1101.101)_2=(13.625)_{10}$$

此等式的含义为:二进制的 1101.101 等于十进制的 13.625。

一般地，任意一个二进制数可展开为

$$N_2=K_n2^n+K_{n-1}2^{n-1}+\cdots+K_02^0+K_{-1}2^{-1}+\cdots+K_{-m}2^{-m}$$
$$=\sum_{i=-m}^{n}K_i2^i$$

其中 K_i 只能取 0 或 1，它由基数决定，m 和 n 为正整数，按通常的记法，记为

$$\underbrace{K_nK_{n-1}\cdots K_1K_0}_{\text{整数部分}}\cdot\underbrace{K_{-1}K_{-2}\cdots K_{-m}}_{\text{小数部分}}$$

4. 一个二进制数，如果是 n 位，它所可能表示的最大数值为 2^n-1，但所能表示的最大状态数为 2^n 个，如一个三位二进制的最大数值为

$$(111)_2=(2^3-1)_{10}=(7)_{10}$$

但它所能表示的最大状态数却是 $2^3=8$，是 8 种状态的组合。

5. 对于最高位为 1，其余皆为 0 的二进制数，其数值为

$$\underbrace{(1000\cdots0)_2}_{n\text{ 个 }0}=2^n$$

如

$$\underbrace{(1000)_2}_{3\text{ 个 }0}=(2^3)_{10}=(8)_{10}$$

二、二进制数的运算及其优点

二进制数的运算是非常简便的，其运算规则为

$0+0=0, 0+1=1, 1+0=1, 1+1=10$

$0\times0=0, 0\times1=0, 1\times0=0, 1\times1=1$

这是加法和乘法的运算规则，至于减法和除法，则分别为加法和乘法的逆运算。在运算过程中需要注意的是：当两数相加时是逢二进一，当两数相减时是借一当二。

下面通过几个例子来领会二进制数的运算规则。

例 1-1 加法

$$\begin{array}{rl}
101.01 & \left(5\frac{1}{4}\right)\\
+\ 110.11 & \left(6\frac{3}{4}\right)\\
\hline
1100.00 & (12)
\end{array}$$

例 1-2 减法

$$\begin{array}{rl}
1100 & (12)\\
-\ 110.11 & \left(6\frac{3}{4}\right)\\
\hline
101.01 & \left(5\frac{1}{4}\right)
\end{array}$$

例 1-3 乘法

$$\begin{array}{rl}
10.101 & (2.625)\\
\times\ 101 & (5)\\
\hline
10101 & \\
00000 & \\
10101 & \\
\hline
1101.001 & (13.125)
\end{array}$$

例 1-4 除法

$$\begin{array}{r|rl}
 & 10.101 & (2.625)\\
\cline{2-2}
101 & 1101.001 & (13.125)\\
(5) & 101 & \\
\cline{2-2}
 & 110 & \\
 & 101 & \\
\cline{2-2}
 & 101 & \\
 & 101 & \\
\cline{2-2}
 & 0 &
\end{array}$$

二进制有如下优点：

1. 运算简单。如上所述，它只有 0 和 1 两个数字，所以运算非常简单，是最方便的一种计数制，这就使数字系统结构简化。

2. 容易表示。任何具有两个稳定状态的元件，都可以用作二进制数的每一位，而制造两种稳定状态的元件要比制造多种稳定状态的元件容易得多，象氖灯的亮和灭、晶体管的导通和截止、触发器的两种工作状态等，都可以用来表示 0 和 1 两个数字。

3. 可以使用逻辑代数。由于二进制只有 0 和 1，因此可以使用逻辑代数来分析和综合有关的逻辑电路。

4. 节省设备。以与十进制比较为例，要表示 0 至 99 之间的数，用十进制数，需要 2×10 个设备状态，而用二进制数表示，则只需 7×2 个设备状态就足够了。

三、数制间的转换

由于数字电路中通常采用二进制，而人们都习惯于十进制，因此需要找出它们之间的转换关系。

1. 二进制数转换成十进制数

二进制数转换成十进制数的过程简称二化十，记作 2→10。

转换的方法是：首先把二进制数写成它的展开表达式，然后将 2 的方次相加。例如

$$
\begin{aligned}
(10111)_2 &= 1\times2^4+1\times2^2+1\times2^1+1\times2^0 \\
&= 16+4+2+1=(23)_{10} \\
(101.1101)_2 &= 1\times2^2+1\times2^0+1\times2^{-1}+1\times2^{-2}+1\times2^{-4} \\
&= 4+1+0.5+0.25+0.0625 \\
&= (5.8125)_{10}
\end{aligned}
$$

2. 十进制数转换成二进制数

十进制数转换成二进制数的过程，简称十化二，记作 10→2。下面通过几个例子说明转换的方法。

例 1-5　$(29)_{10}=(?)_2$

这是一个十进制整数化为二进制数的例子。方法是将十进制数除以 2，得到一个商和余数，再将商数除以 2，又得到一个新的商数和余数，如此连续进行下去，直到商等于 0 为止，这时得到的各次余数(显然，每次余数只能是 0 或 1)就是所求二进制数的各位数字，其最后余数为最高位数字。运算过程如下：

$$
\begin{array}{r|l l l}
2 & 29 & & \\
2 & 14 & 余\ 1 & (低位) \\
2 & 7 & 余\ 0 & \\
2 & 3 & 余\ 1 & \\
2 & 1 & 余\ 1 & (高位) \\
 & 0 & 余\ 1 &
\end{array}
$$

即　$(29)_{10}=(11101)_2$

这种把十进制整数化为二进制数的方法称为“除 2 取余”法。对于任何一个十进制整数，都可以用有限位的二进制的形式表达出来。

例 1-6　$(0.625)_{10}=(?)_2$

这是一个十进制纯小数化为二进制纯小数的例子，方法是先用 2 乘以十进制纯小数，然后去掉乘积的整数部分，再用 2 去乘剩下的纯小数部分，如此下去，一直到乘积只剩整数或满足所求的精确度为止。把每次乘积的整数部分（不是 0 就是 1），由上而下排列起来（即第一次相乘所得的整数部分为二进制数小数点后的第一位），就是所求的二进制小数。运算过程如下：

0.625×2	整数部分为 1	纯小数部分为 0.25
0.25×2	整数部分为 0	纯小数部分为 0.5
0.5×2	整数部分为 1 ↓	

（低位）

即
$$(0.625)_{10}=(0.101)_2$$

这种把十进制纯小数化为二进制纯小数的方法称为“乘 2 取整”法。例 1-6 是一个能以有限的二进位（此例为小数点后三位）准确地实现转换的例子，但并不是任何一个十进制小数都能以有限位二进制的小数形式精确表示的。这时的 10→2 转换，只能根据人们对精度的要求，转换到一定位数为止。

例 1-7　$(9.634)_{10}=(?)_2$

这是一个既有整数又有小数的十进制数化为二进制数的例子，要求小数部分取七位有效数字。

整数部分

$$\begin{array}{r|l}
2 & 9 \\ \hline
2 & 4 \quad 余\ 1 \uparrow \\ \hline
2 & 2 \quad 余\ 0 \\ \hline
2 & 1 \quad 余\ 0 \\ \hline
 & 0 \quad 余\ 1
\end{array}$$

即
$$(9)_{10}=(1001)_2$$

小数部分

0.634×2	整数部分	1	小数部分	0.268
0.268×2	整数部分	0	小数部分	0.536
0.536×2	整数部分	1	小数部分	0.072
0.072×2	整数部分	0	小数部分	0.144
0.144×2	整数部分	0	小数部分	0.288
0.288×2	整数部分	0	小数部分	0.576
0.576×2	整数部分	1 ↓	小数部分	0.152

即　$(0.634)_{10}=(0.1010001)_2$

整数部分与小数部分相加，可得最后结果为

$$(9.634)_{10}=(1001.1010001)_2$$

3. 八进制数及其转换

由上述 2⟵⟶10 转换中可以看出,在相同的计数能力下,二进制数位数是十进制数位数的三倍多,书写起来,既费力又易出错,所以人们往往用八进制来书写相应的二进制数码(如数字计算机程序中的操作码)。

所谓八进制就是用 0、1、2、3、4、5、6、7 这八个数码,按"逢八进一"的规则来计数的,基数为八。因此有表 1-2 的对应关系。

表 1-2 十、二、八进数的对应

十进制	0	1	2	3	4	5	6	7	8	9	10
二进制	0	1	10	11	100	101	110	111	1000	1001	1010
八进制	0	1	2	3	4	5	6	7	10	11	12

显然,八进制数要比相应的二进制数简短得多。

八进制数转换成十进制数也是很简单的。例如

$(165)_8=1\times 8^2+6\times 8^1+5\times 8^0=64+48+5=(117)_{10}$

$(77)_8=7\times 8^1+7\times 8^0=56+7=(63)_{10}$

十进制数转换成八进制数,可采用类似于 10→2 转换中的方法,只不过这时用"除 8 取余"(小数时用"乘 8 取整")来运算就可以了。

八进制数的一个更重要的优点是它与二进制数的相互转换很方便,这是因为 $2^3=8$,所以一位八进制数相当于三位二进制数,它们是完全对应的。

例 1-8 $(63.52)_8=(?)_2$

只要把每位八进制数,用三位二进制数表示即可,即

$$\begin{array}{cccc} 6 & 3 & . & 5 & 2 \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ \overbrace{1\,1\,0} & \overbrace{0\,1\,1} & . & \overbrace{1\,0\,1} & \overbrace{0\,1\,0} \end{array}$$

所以

$$(63.52)_8=(110011.101010)_2$$

例 1-9 $(1101111.10011)_2=(?)_8$

只要将每三位二进制数用一个八进制数表示即可,即

$$\begin{array}{ccccc} \underbrace{0\,0\,1} & \underbrace{1\,0\,1} & \underbrace{1\,1\,1}. & \underbrace{1\,0\,0} & \underbrace{1\,1\,0} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 5 & 7\ . & 4 & 6 \end{array}$$

所以

$$(1101111.10011)_2=(157.46)_8$$

这里要注意,分组时要以小数点为界,对整数部分自右向左分组,对小数部分则自左向右

分组。

十六进制在与二进制转换中也有类似的简便性，所以数字计算机上也有用十六进制的。

4. 十六进制数及其转换

十六进制数的特点是逢十六进一。它的基本数字共 16 个，从小到大依次是 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

十六进制比八进制书写更简便。由于 $2^4=16$，因此每位十六进制数正好与四位二进制数相对应，二者可以很方便地进行换算。

例 1-10 $(3B.8C)_{16}=(?)_2$

只要把每位十六进制数直接用四位二进制数表示出来即可：

$$\overbrace{0\ 0\ 1\ 1}^{\substack{3\\ \downarrow}}\quad\overbrace{1\ 0\ 1\ 1}^{\substack{B\\ \downarrow}}\ .\ \overbrace{1\ 0\ 0\ 0}^{\substack{8\\ \downarrow}}\quad\overbrace{1\ 1\ 0\ 0}^{\substack{C\\ \downarrow}}$$

所以 $(3B.8C)_{16}=(111011.100011)_2$

例 1-11 $(1110101.10111)_2=(?)_{16}$

以小数点为界，整数部分自右向左，小数部分自左向右，按四位二进制数对应一位十六进制数的方式进行：

$$\underbrace{0\ 1\ 1\ 1}_{\substack{\downarrow\\ 7}}\quad\underbrace{0\ 1\ 0\ 1}_{\substack{\downarrow\\ 5}}\ .\ \underbrace{1\ 0\ 1\ 1}_{\substack{\downarrow\\ B}}\quad\underbrace{1\ 0\ 0\ 0}_{\substack{\downarrow\\ 8}}$$

所以 $(1110101.10111)_2=(75.B8)_{16}$

1-4 十进制数的二进制编码(BCD)

在数字系统中，常常使用一种十进制与二进制之间过渡性的计数方式，叫做“二-十进制”，又叫“二进制编码的十进制数”，简称 BCD。在二-十进制中，所表示的仍是十进制数，但每位十进制数都写成四位二进制码的形式。

8421 码是 BCD 码中最直观、最自然的一种编码方式，十进制数与其对应的码如表 1-3 所示。很显然，8421 码的每个码与所代表的十进制数之间符合二进制数与十进制数的换算关系，因此，8421 码也称为简单的二进码。

将十进制数表示成 BCD 码，只要使数与码对应排列即可。例如十进制数 1982 写成8421 码就是 0001 1001 1000 0010。

由于四位二进制码可以表示十六种不同的状态，而一个十进制数的每位最多有十种不同的状态，所以，用四位二进制码表示一位十进制数可以有多种组合方式，而且不管哪种组合方式，均有六个状态是多余的。例如 8421 码的情况下，1010、1011、1100、1101、1110 和 1111 就是多余的。

BCD 码的更详细情况将在第四章中介绍。

表 1-3　BCD 码示例

十进制数	8421码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

1-5　数的传送方式

数在数字系统中的传送方式主要有三种形式。

1. 串行传送

一个数的各位数从低位到高位(或从高位到低位)依次传送，叫串行传送。这时不同的数位是在时间上加以区别的，如图 1-3 所示。

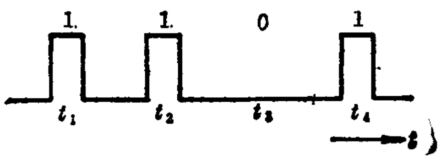

图 1-3　串行传送

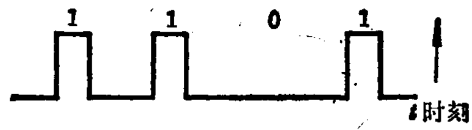

图 1-4　并行传送

2. 并行传送

一个数的所有各位数同时传送，叫并行传送。这时不同的数位是空间上加以区别的，如图 1-4 所示。

3. 串-并行传送

把一个数的各位数分成若干组，组内各位同时传送，组与组之间串行传送，这种方式叫串-并行传送，就传送和完成运算的速度以及所用的设备数量来说，串-并行传送介乎前二者之间。

本 章 小 结

数字信号是一种离散信号，它在时间和数值上都是离散的，数字电路是处理数字信号并能进行数字运算和逻辑判断的电路，也叫逻辑电路。

数字电路在电路结构、电路功能、所用的数学工具和元器件等方面，都不同于模拟电路，这些

特点都与二进制数密切相关。

二进制数具有运算简单、容易表示、节省设备和便于使用逻辑代数等优点，所以在数字电路中得到广泛的应用。应熟练掌握二进制与十进制、二-十进制之间的相互转换方法。

思考题和习题

1. 模拟信号和数字信号的含义是什么？

2. 什么是数字电路？它与模拟电路作比较，有哪些特点？

3. 在数字电路中，为什么要采用二进制？在电子数字计算机中，还常常采用八进制和十六进制，为什么？

4. 将下列各数按权展开：

(1) $(11010)_2$；　(2) $(1011.011)_2$；

(3) $(7642)_{10}$；　(4) $(4211.375)_{10}$；

(5) $(56.743)_8$。

5. 把下列二进制数转换成十进制数：

(1) $(11000101)_2$；　(2) $(1010010)_2$；

(3) $(010001)_2$；　(4) $(0.01001)_2$；

(5) $(10101.001)_2$。

6. 二进制数 0000000000～1111111111 可以有多少种状态的组合？它所可能表示的最大数值是多少？

7. 将下列二进制数转换成八进制数、十六进制数：

(1) $(110101001001)_2$；　(2) $(111011)_2$；　(3) $(0.101011)_2$

(4) $(1011111.01101)_2$　(5) $(11110000)_2$。

8. 把下列十进制数转换成二进制数：

(1) 49；(2) 78；(3) 113；(4) 295；(5) 306。

9. 把下列十进制数转换成二进制数(准确到小数点后六位)：

(1) 9.18；(2) 0.765；(3) 6.625；(4) 8.91；

(5) 0.634。

10. 在二-十进制编码中，有多少种组合方式？为什么常用 8421 码？

11. 用 8421 码表示下列十进制数：

(1) $(42.78)_{10}$；　(2) $(106.35)_{10}$；

(3) $(4.09)_{10}$。

12. 用十进制数表示 8421 码 0001100101111001。

第二章　逻辑函数分析

逻辑代数是研究数字电路的工具。逻辑代数起源于推理判断问题，用文字、符号表示条件和结果，用逻辑函数表示它们之间的逻辑关系。本章将从逻辑函数的基本定义出发，导出一系列定律、公式和图形规则，为处理逻辑表达式提供简便的方法，从而为分析和设计数字电路打下基础。

2-1　基本逻辑函数

一、引言

在实际事物中，存在着大量的因果关系，并且条件和结果往往都可区分为两种对立的状态，如是与否，开与关，正与负等。运用推理方法，可以从各条件所处的状态出发，推出结果所处的状态，条件与结果的这种关系就叫做逻辑关系。

首先让我们看一个众所周知的平面几何定理："如果两个三角形的三组对应边都相等，则两个三角形全等。"可以说三组对应边的关系就是三个条件，它们各自存在着两个对立的状态：相等与不等。两个三角形的关系是结果，它也存在着全等与不全等两个对立的状态。上述定理表示出了三个条件与一个结果之间存在的逻辑关系。再举一个例子。在一个电灯的供电电路中，串联有三个开关：S_1、S_2 和 S_3，如图 2-1 所示。开关的两个对立状态是"接"与"断，"而电灯则是"亮"与"灭"。开关与灯之间的逻辑关系是："只有三个开关都接通，灯才亮"。上面两个例子描述的是完全不相干的两种事物，但它们的逻辑关系却有一定的共同点。为了研究方便，在第一个例子中，把对应边相等定义为 I 状态，对应边不等定义为 II 状态；三角形全等定义为 I 状态，不全等定义为 II 状态。在第二个例子中，开关接通定义为 I 状态，开关断开定义为 II 状态； 灯亮定义为 I 状态，灯灭定义为 II 状态。这样，我们就可以把两个例子的逻辑关系统一成：如果三个条件都处在 I 状态，则结果处在 I 状态；而三个条件中，只要有一个处在 II 状态，结果便处在 II 状态。在实际中，我们可以找到许多类似的逻辑关系，它就是下面将要研究的三种基本逻辑关系之一的"与"逻辑。

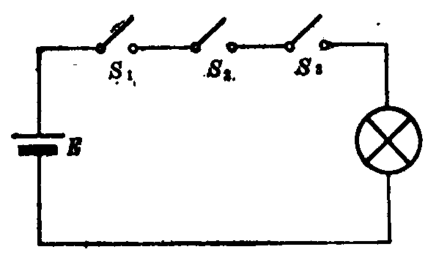

图 2-1　开关串联电路

研究逻辑的数学工具是逻辑代数。在逻辑代数中是用 0 和 1 来表示条件和结果的两种对立状态的。例如在上面两个例子中，可以用 1 表示 I 状态，即对应边相等、三角形全等、开关接通和灯亮；用 0 表示 II 状态，即对应边不等、三角形不全等、开关断开和灯灭。更进一步，如果把条件视为自变量，结果视为因变量，它们的状态都以 0 或 1 的取值来描述，那么，逻辑关系就变成了逻辑函数。必须注意的是，这里的变量都是逻辑变量，它们的取值不是数量，而是状态，与普通代数