# 数字电路

孙凯生　主编

郭静萍　宋应生　欧阳琼　等编
孙凯生　关晓毅　王　贵



北京航空航天大学出版社

# 数 字 电 路

孙凯生 主 编

郭静萍、宋应生、欧阳琼 孙凯生、关晓毅 王 贵 等编

北京航空航天大学出版社

（京）新登字166号

## 内 容 简 介

主要内容有：数制与逻辑代数基础；晶体管开关特性及门电路；组合逻辑电路；时序逻辑电路；脉冲产生与变换电路；数字/模拟变换和模拟/数字变换电路等。每章均附思考练习题。本书在阐述基本原理和基本概念的基础上，对一些常用数字集成电路的典型产品进行了分析，可供实际应用时参考。

本书可做机械制造工程各专业本科、专科教材，也可供学习或从事数字电子技术、计算机应用、控制与检测技术等方面的人员参考。



**数 字 电 路**

**SHUZHI DIANLU**

孙凯生 主编

郭静萍、宋应生、欧阳琼 等编
孙凯生、关晓毅、王 贵

责任编辑 许传安

北京航空航天大学出版社出版
新华书店总店科技发行所发行 各地新华书店经销
北京农业工程大学印刷厂印装

787×1092 1/16 印张：12.25 字数：313千字
1992年9月第1版 1992年9月第1次印刷 印数：3000册
ISBN 7-81012-323-8/TN·017 定价：7.40元

# 前　言

本教材是根据机械制造工程各专业教学大纲的要求编写的。教学计划中电子技术类课程共有三门：《电路基础》、《模拟电路》和《数字电路》。《数字电路》参考学时为52学时，其中可安排10学时实验教学。

本书的主要内容为：数制与逻辑代数基础、晶体管开关特性及门电路、组合逻辑电路和时序逻辑电路的分析与设计、脉冲产生与变换电路、数字/模拟变换和模拟/数字变换电路等。

本教材的特点是：在较全面地介绍集成数字电路基本知识和基本概念的基础上，突出实用性、针对性、实践性。书中介绍了一些实用的电路及应用组配方法。在编写过程中注重理论联系实际，对一些常用数字集成电路的典型产品进行了分析、介绍，突出外部特性和应用方法。在每一章后均附有思考与练习题，便于组织教学，提高读者分析和解决问题的能力。

本书参考了国内各重点高等工业院校“数字电路”方面的教材，份量适中，供机械制造工程各专业本科教学使用；也可供相近专业专科和培训班及其他非电子技术类专业教学使用；还可供从事数字电子技术、计算机应用工作的读者参考使用。

本教材由北京航空航天大学制造工程系孙凯生主编。各章编写人如下：第一章郭静萍；第二章郭静萍、王贵，第三章欧阳琼、关晓毅；第四章宋应生、孙凯生；第五章关晓毅、宋应生；第六章孙凯生、宋应生。邬学礼教授、汪一彭教授、韩云台副教授、赵莹琳副教授等参加了本教材大纲的制定，并对教材提出宝贵意见，谨致谢意！

欢迎读者提出改进意见。

编　者

一九九一年十二月

# 目　录

# 第一章　数制及逻辑代数基础

## §1-1　数　　制

### 1-1-1　数字体制

数字体制可分为“进位制”记数和“非进位制”记数。进位制就是一种按进位方式实现计数的制度，常用的进位制有十进制、二进制、八进制、十六进制等。在十进制中，如256的2代表$2\times10^2$，5代表$5\times10^1$，6代表$6\times10^0$。其中任何一个数码所表示的值不只是决定于数码本身，还决定于它所在的位置，我们把$10^2$，$10^1$和$10^0$称为各位的“权重”，所以进位制记数又称为有“权”记数。

每种进位制都有一个基本特征数，称为进位制的“基数”。基数表示了进位制所具有的数字符号的个数及进位规则，显然，十进制的基数为“十”，二进制的基数为“二”，R进制的基数为“R”。基数和权是进位制的两个要素。

所谓“非进位制”记数就是数码所表示的值只决定于数码本身，与它所在的位置无关，所以“非进位制”记数又称为无“权”记数。在古希腊和罗马通用的罗马数字就是一种无“权”记数。如十一不是Ⅰ Ⅰ而是Ⅺ，五十是L而不是ⅤⅩ等等。此种记数法很不方便，如今只在个别地方使用。

### 1-1-2　十进制记数

十进制记数总共有十个数码：0、1、2、3、4、5、6、7、8、9，外加一个小数点“•”。它的记数规则是以“十”为“基数”，逢十进一。对于某一个十进制数N可以按位置记为：

$$(N)_{10}=(a_{n-1}a_{n-2}\cdots a_1a_0\cdot a_{-1}a_{-2}\cdots a_{-m})_{10}$$

其中：$0\leqslant a_i\leqslant9$：代表数码

$n$：代表整数位数

$m$：代表小数位数

这种表示方法称为“并列表示法”。

如果将并列表示的数N按权展开可以写成：

$$(N)_{10}=a_{n-1}(10)^{n-1}+a_{n-2}(10)^{n-2}+\cdots+a_1(10)^1+$$
$$+a_0(10)^0+a_{-1}(10)^{-1}+a_{-2}(10)^{-2}+\cdots+a_{-m}(10)^{-m}$$

展开后是一个多项式，所以又称它为“多项式表示法”。多项式表示法也可以简记为：

$$(N)_{10}=\sum_{i=-m}^{n-1}a_i(10)^i$$

如十进制数256.25这种并列表示可以展开成多项式的形式：

$$(256.25)_{10}=2\times(10)^2+5\times(10)^1+6\times(10)^0+2\times(10)^{-1}+5\times(10)^{-2}$$

### 1-1-3 二进制记数

和十进制记数类似，二进制记数总共有两个数码：0、1和一个小数点“·”。它的记数规则是以二为基数，逢二进一。同样，对某一个二进制数N可以并列表示为：

$$(N)_2 = a_{n-1}a_{n-2}\cdots a_1a_0\cdot a_{-1}\cdots a_{-m})_2$$

其中：$0 \leqslant a_i \leqslant 1$：代表数码

n：代表整数位数

m：代表小数位数

按“权”展开为多项式：

$$(N)_2 = a_{n-1}(2)^{n-1} + a_{n-2}(2)^{n-2} + \cdots + a_1(2)^1 + a_0(2)^0 + a_{-1}(2)^{-1} + a_{-1}(2)^{-2} + \cdots + a_{-m}(2)^{-m}$$

一般可简记为：

$$(N)_2 = \sum_{i=-m}^{n-1} a_i(2)^i$$

应当特别指出，在这里基数$(10)_2$为了防止和十进制的10相混淆，借用了十进制数码2来代替，指数也借用了十进制代码。

〔例1-1〕 $(1011.01)_2$这种并列表示可以展开为如下的多项式形式

$$(1011.01)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$$

二进制记数的特点是数码少，运算规则简单，例如加法运算规则是：

$$0+0=0$$

$$0+1=1+0=1$$

$$1+1=10$$

乘法运算规则是：

$$0\times0=0$$

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

$$1\times1=1$$

数码少容易由电子器件状态来对应。0、1两个数码可以选用有两个状态的电子器件来和它相对应，如开关的通和断、晶体管的导通和截止等等；运算规则简单可以使运算电路和控制电路简单，所以在数字技术中普遍采用二进制记数。二进制记数各位的“权”见表1-1

**表1-1 二进制的权$2^n$**

| $2^n$ | n | $2^{-n}$ | $2^n$ | n | $2^{-n}$ |
|---|---|---|---|---|---|
| 1 | 0 | 1.0 | 256 | 8 | 0.00390625 |
| 2 | 1 | 0.5 | 512 | 9 | 0.001953125 |
| 4 | 2 | 0.25 | 1024 | 10 | 0.0009765625 |
| 8 | 3 | 0.125 | 2048 | 11 | 0.00048828125 |
| 16 | 4 | 0.0625 | 4096 | 12 | 0.000244140625 |
| 32 | 5 | 0.03125 | 8192 | 13 | 0.0001220703125 |
| 64 | 6 | 0.015625 | 16384 | 14 | 0.00006103515625 |
| 128 | 7 | 0.0078125 | | | |

### 1-1-4 其它几种进位制记数

在数字技术中除普遍采用二进制记数外，也常用“八进制”记数和“十六进制”记数。

八进制记数共有八个数码：0、1、2、3、4、5、6、7 和一个小数点“•”。记数规则是以八为基数，逢八进一。它的并列表示为

$$(N)_8=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}\cdots a_{-m})_8$$

展开成多项式形式为：

$$(N)_8=a_{n-1}\times(8)^{n-1}+a_{n-2}\times(8)^{n-2}+\cdots+a_1\times(8)^1+a_0\times(8)^0$$
$$+a_{-1}\times(8)^{-1}+a_{-2}\times(8)^{-2}+\cdots+a_{-m}\times(8)^{-m}$$

一般可写为： $(N)_8=\sum\limits_{i=-m}^{n-1}a_i(8)^i$

同样这里基数也是借用十进制的数码，如：

$$(246.3)_8=2\times8^2+4\times8^1+6\times8^0+3\times8^{-1}$$

十六进制记数总共有十六个数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F和一个小数点“•”。它的记数规则是以十六为基数，逢十六进一。

并列表示为 $(N)_{16}=(a_{n-1}a_{n-2}\cdots a_1a_0\cdot a_{-1}a_{-2}\cdots a_{-m})_{16}$

多项式为 $(N)_{16}=a_{n-1}\times(16)^{n-1}+a_{n-2}\times(16)^{n-2}+\cdots+a_1\times(16)^1$

$$+a_0\times(16)^0+a_{-1}\times(16)^{-1}+a_{-2}\times(16)^{-2}+\cdots+a_{-m}\times(16)^{-m}$$

一般写为 $(N)_{16}=\sum\limits_{i=-m}^{n-1}a_i(16)^i$

〔例1-2〕 $(BC08A.F)_{16}=B\times(16)^4+C\times(16)^3+0\times(16)^2+8\times(16)^1+A\times(16)^0+F\times(16)^{-1}$

把上述几种进位制记数及其规则推广到任意进制（如R进制）可表达如下：

“R进制记数”总共有R个数码：0、1、…、R－1，和一个小数点“•”。它的记数规 则是以R为基数，逢R进一。

并列表示为 $(N)_R=(a_{n-1}a_{n-2}\cdots a_1a_0\cdot a_{-1}a_{-2}\cdots a_{-m})_R$

多项式为 $(N)_R=a_{n-1}(R)^{n-1}+a_{n-2}(R)_{n-2}+\cdots+a_1(R)^1+a_0(R)^0$

$$+a_{-1}(R)^{-1}+a_{-2}(R)^{-2}+\cdots+a_{-m}(R)^{-m}$$

一般写为 $(N)_R=\sum\limits_{i=-m}^{n-1}a_i(R)^i$

## §1-2 各进位制记数间的转换

如前节所述，同一数值可以用不同进位制的数表示，如$(12)_{10}=(1100)_2=(14)_8=(c)_{16}$，所以，不同进位制只是描述数值的不同“手段”，因此它们可以相互转换，转换的前提是保证所表示的数值相等。

### 1-2-1 多项式替代法

如果将二进制数 1011.01 转换成十进制数，根据上节多项式表示法可首先将二进制数按

权展开，经过数码代换后再在十进制数中进行计算，所得的值就是该二进制数相对应的十进制数。

〔例1-3〕

$$(1011.01)_2=[1\times(10)^{11}+0\times(10)^{10}+1\times(10)^{1}+1\times(10)^{0}$$
$$+0\times(10)^{-1}+1\times(10)^{-10}]_2$$

经过代码代换 $=[1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}]_{10}$

在十进制中相加 $=(11.25)_{10}$

具体做法可以归纳为如下几个步骤：

(1) 将二进制数按权展开。

(2) 再把二进制的各个数码用十进制数相对应的数码来替代，包括各位权的基数和指数（见表1-2）。

**表1-2　十进制、二进制、八进制、十六进制对照表**

| 十进制 | 二 进 制 | 八进制 | 十六进制 |
|---|---|---|---|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

(3) 按照十进制数的运算规则计算多项式的值。

上述方法可推广为一般的形式。将 $\alpha$ 进制的数转换为 $\beta$ 进制的数时，可按以下规则进行：

(1) 将$\alpha$进制的数按权展开。

(2) 再将$\alpha$进制各位数码$a_i$和权$(10)^i$用相对应的$\beta$进制的数码$b_i$和$(\alpha)^i$来替代。

(3) 最后按$\beta$进制的运算规则计算多项式的值。

具体表示为：

$$(N)_\alpha=(a_{n-1}a_{n-2}\cdots a_1a_0\cdot a_{-1}a_{-2}\cdots a_{-m})_\alpha$$
$$=[a_{n-1}(10)^{n-1}+a_{n-2}(10)^{n-2}+\cdots a_1(10)^1$$
$$+a_0(10)^0+a_{-1}(10)^{-1}+a_{-2}(10)^{-2}+\cdots+a_{-m}(10)^{-m}]_\alpha$$
$$=[b_{n-1}\alpha^{n-1}+b_{n-2}\alpha^{n-2}+\cdots+b_1\alpha^1+b_0\alpha^0$$

$$+b_{-1}\alpha^{-1}+b_{-2}\alpha^{-2}+\cdots+b_{-m}\alpha^{-m}]_\beta$$
$$=(b_{k-1}b_{k-2}\cdots b_1b_0\cdot b_{-1}b_{-2}\cdots b_{-1})_\beta$$

〔例1-4〕 试把八进制数357.4转换为十进制数。

**解**：
$$(357.4)_8=[3\times(10)^2+5\times(10)^1+7\times(10)^0+4\times(10)^{-1}]_8$$
$$=(3\times8^2+5\times8^1+7\times8_0+4\times8^{-1})_{10}$$
$$=(192+40+7+0.5)_{10}$$
$$=(239.5)_{10}$$

反之，如把十进制数239.5转换成八进制数就难以完成，因为对八进制的运算规则我们不熟悉。

所以，当把"$\alpha$进制"数转换为"$\beta$进制"数时必需在掌握$\beta$进制数运算规则的条件下才能使用"多项式替代法"。

### 1-2-2 基数乘除法

基数乘除法和多项式替代法不同，它必需将整数和小数分开进行转换。整数转换使用基数除法，小数转换用基数乘法，转换后再合并。

**1. 基数除法**

若将$\alpha$进制的整数$(N)_\alpha$转换为$\beta$进制的整数$(M)_\beta$可采用基数除法。如将十进制整数$(N)_{10}$转换为二进制数$(M)_2$。将$(M)_2$展开为多项式

$$(M)_2=a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\cdots+a_12^1+a_02^0$$

因为转换后的$(M)_2$与$(N)_{10}$等值，所以

$$(N)_{10}=a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\cdots+a_12^1+a_02^0$$

上式两边同时除以2，则两边的商和余数对应相等。

$$\frac{(N)_{10}}{2}=\underbrace{(a_{n-1}2^{n-2}+a_{n-2}2^{n-3}+\cdots+a_12^0)}_{\text{商 }(N_1)_{10}}+\underbrace{a_0}_{\text{余数}}$$

余数$a_0$明显为二进制数$(M)_2$最小位的系数。

$$\frac{(N_1)_{10}}{2}=\underbrace{(a_{n-1}2^{n-3}+a_{n-2}2^{n-4}+\cdots+a_22^0)}_{\text{商 }(N_2)_{10}}+\underbrace{a_1}_{\text{余数}}$$

余数$a_1$为二进制数$(M)_2$次小位的系数。依次类推，便可求出二进制数每一位的系数。

推而广之，将$\alpha$进制的整数转换为$\beta$进制的整数步骤如下：

(1) 将$\alpha$进制$(N)_\alpha$，在$\alpha$进制中反复除以$\beta$，直至商为0。

(2) 每次除后所得余数为$(M)_\beta$由低位到高位的系数。将所得余数以$\beta$进制数表示，则为$(M)_\beta$并列表示时由低位到高位之值。

此方法概括为"除基取余"。

〔例1-5〕 用基数除法把$(712)_{10}$转换为二进制数。

二进制的基数在十进制数中是 2，即$\beta=2$。

将$(712)_{10}$反复由 2 来除，可以采用简化除式来做：

| | |
|---|---|
| $2\,\underline{\mid 712}$ | 余数 |
| $2\,\underline{\mid 356}$ | 0 |
| $2\,\underline{\mid 178}$ | 0 |
| $2\,\underline{\mid 89}$ | 0 |
| $2\,\underline{\mid 44}$ | 1 |
| $2\,\underline{\mid 22}$ | 0 |
| $2\,\underline{\mid 11}$ | 0 |
| $2\,\underline{\mid 5}$ | 1 |
| $2\,\underline{\mid 2}$ | 1 |
| $2\,\underline{\mid 1}$ | 0 |
| $0$ | 1 |

因为十进制的数码 0 和 1 也就是二进制的数码 0 和 1，所以结果为：

$$(712)_{10} = (1011001000)_2$$

**2. 基数乘法**

若将$\alpha$进制的小数$(N)_\alpha$转换为$\beta$进制的小数$(M)_\beta$，可以采用基数乘法。如将十进制的小数$(N)_{10}$转换为二进制数$(M)_2$。将$(M)_2$展开为多项式：

$$(M)_2 = a_{-1}2^{-1} + a_{-2}2^{-2} + \cdots + a_{-m}2^{-m}$$

因为转换后的$(M)_2$与$(N)_{10}$等值，所以

$$(N)_{10} = a_{-1}2^{-1} + a_{-2}2^{-2} + \cdots + a_{-m}2^{-m}$$

上式两边同时乘以2

$$2(N)_{10} = a_{-1} + a_{-2}2^{-1} + \cdots\cdots + a_{-m}2^{-m+1}$$

$a_{-1}$为积的整数部分，其余为积的小数部分，而等式两边积的整数部分和小数部分对应相等，所以$a_{-1}$为$(M)_2$最高位的系数。

$$2(N)_{10} - a_1 = a_{-2}2^{-1} + \cdots\cdots + a_{-m}2^{-m+1}$$

将上式两边再同时乘以 2 则得：

$$2[(N)_{10} - a_1] = a_{-2} + a_{-3}2^{-1} + \cdots\cdots + a_{-m}2^{-m+2}$$

同理，$a_{-2}$为$(M)_2$次高位的系数。依次类推，便可求出二进制小数$(M)_2$的各位系数。

推而广之，基数乘法的基本规则为：

(1) 将$\alpha$进制小数$(N)_\alpha$在$\alpha$进制中反复乘以$\beta$，直到乘积的小数部分为0或达到要求的精度为止。

(2) 每次乘完所得积的整数部分用$\beta$进制数码代换便得$(M)_\beta$。各次乘积的整数即为$(M)_\beta$并列表示时由高位到低位之值。

此方法概括为“乘基取整”。

〔**例1-6**〕 用基数乘法将$(0.71875)_{10}$转换为二进制数。

$\beta = 2$　　$\underbrace{\text{整数部分}}$

$0.71875 \times 2 = 1.4375$

$0.4375 \times 2 = 0.875$

$0.875 \quad \times 2 = 1.75$

$0.75 \quad \times 2 = 1.5$

$0.5 \quad \times 2 = 1.0$

所得结果为：

$$(0.71875)_{10} = (0.10111)_2$$

如果$\alpha$进制数$(N)_\alpha$既有整数又有小数，则可以把它分开来转换，对整数部分用“除基取余”；对小数部分用“乘基取整”，最后再把所得整数和小数合并起来。

〔例1-7〕 将十进制数 $(712.71875)_{10}$转换为二进制数。

分两步做：首先对整数部分按“除基取余”进行转换

$$(712)_{10} = (1011001000)_2$$

再对小数部分按“乘基取整”进行转换

$$(0.71875)_{10} = (0.10111)_2$$

然后把转换后的整数和小数合并起来，便得到等值的二进制数。

$$(712.71875) = (1011001000.10111)_2$$

### 1-2-3 数码直接代换法

在由$\alpha$进制数转换为$\beta$进制数时，如果$\alpha$和$\beta$都是 $2^i(i = 1,2,\cdots n)$，则可以采用数码直接代换法进行转换，具体作法如下：

**1. $2^i$进制数与二进制数的转换**

如果把$2^i$进制数按权展开为：

$$\begin{aligned}(N)_{2^i} &= [a_{n-1}(2^i)^{n-1} + a_{n-2}(2^i)^{n-2} + \cdots \\ &\quad + a_1(2^i)^1 + a_0(2^i)^0 + a_{-1}(2^i)^{-1} \\ &\quad + a_{-2}(2^i)^{-2} + \cdots + a_{-m}(2^i)^{-m}] \\ &= a_{n-1}2^{i(n-1)} + a_{n-2}2^{i(n-2)} + \cdots + \\ &\quad + a_1 2^i + a_0 2^0 + a_{-1}2^{-i} + a_{-2}2^{-2i} + \cdots + a_{-m}2^{-mi}\end{aligned}$$

可见 $2^i$ 进制数按权展开后各位的权也都是 2 的乘幂，而且它的指数与二进制数各位权的指数是倍数关系。这个倍数 i 就是 $2^i$ 进制数一位对应于二进制数的位数，所以由 $2^i$ 进制数转换为二进制数时，可直接把 $2^i$ 进制各位数码用等值的 i 位二进制数码进行代换便可。

〔例1-8〕 将八进制数$(67.721)_8$转换为二进制数。

因为$8 = 2^3$，即八进制一位对应于二进制三位，

$$\overbrace{110}^{6}\quad \overbrace{111.}^{7.}\quad \overbrace{111}^{7}\quad \overbrace{010}^{2}\quad \overbrace{001}^{1}$$

所以， $(67.721)_8 = 110111.111010001)_2$

反之，如果将二进制数转换为 $2^i$ 进制数，对于整数部分从小数点向左数，每 i 位对应于 $2^i$进制数一位，高位不足时添“0”补齐；小数部分从小数点向右数，每 i 位对应于 $2^i$ 进制数一位，低位不足时添“0”补齐。

〔例1-9〕 将二进制数$(11111101.01001111)_2$转换为八进制数。

因为$8 = 2^3$

$$\underbrace{011}_{3}\quad\underbrace{111}_{7}\quad\underbrace{101.}_{5.}\quad\underbrace{010}_{2}\quad\underbrace{011}_{3}\quad\underbrace{110}_{6}$$

所以 $(11111101.01001111)_2=(375.236)_8$

**2. $2^i$进制数与$2^j$进制数的转换($i \neq j \neq 1$)**

$2^i$ 进制数与$2^j$ 进制数的转换可先将 $2^i$ 进制数转换为二进制数，然后再将二进制数转换为$2^j$进制数。

〔**例1-10**〕 将八进制数$(375.236)_8$转换为十六进制数。

先将八进制转换为二进制数

$$(375.239)_8=(11111101.01001111)_2$$

再将二进制数转换为十六进制数

$$\underbrace{1111}_{F}\quad\underbrace{1101.}_{D.}\quad\underbrace{0100}_{4}\quad\underbrace{1111}_{F}$$

所以 $(375.236)_8=(FD.4F)_{16}$

### 1-2-4 进位数制转换中的几个问题

**1. 小数位制的确定**

在前述进位数制的转换中都是小数可准确地实现转换的情形，即转换后小数位数是有限的。在有些小数的转换中转换后小数的位数是无限的，如：

$$(21.3)_{10}=(10101.010011001\cdots)_2$$

或虽然有限但位数很多，这时可以按着一定的精度进行小数的近似转换。

如果$\alpha$进制数的小数是k位，转换为$\beta$进制数的小数有j位，那么要保持同样的精度一定有：

$$(0.1)_\alpha^k=(0.1)_\beta^j$$

在十进制中应写成：

$$\left(\frac{1}{\alpha}\right)_{10}^k=\left(\frac{1}{\beta}\right)_{10}^j$$

对等式两边取以$\alpha$为底的对数，则得：

$$k\log_\alpha\left(\frac{1}{\alpha}\right)=j\log_\alpha\left(\frac{1}{\beta}\right)$$

即 $$k\log_\alpha(\alpha)=j\log_\alpha(\beta)$$

因为 $$\log_\alpha(\alpha)=1$$

所以 $$k=j\log_\alpha(\beta)=j\frac{\log_{10}(\beta)}{\log_{10}(\alpha)}=j\frac{\lg(\beta)}{\lg(\alpha)}$$

或 $$j=k\frac{\lg(\alpha)}{\lg(\beta)}$$

取j为整数，故j应满足

$$k\frac{\lg(\alpha)}{\lg(\beta)}\leqslant j\leqslant k\frac{\lg(\alpha)}{\lg(\beta)}+1$$

〔**例1-11**〕 将$(0.43215)_{10}$转换为十六进制小数，要求保持$\pm(0.1)_{10}^5$的精度。

因为k = 5

$$5\frac{\lg(10)}{\lg(16)}\leqslant j\leqslant 5\frac{\lg(10)}{\lg(16)}+1;\quad 4.15\leqslant j\leqslant 5.15$$

所以取j = 5。

**2. 数制转换方法的选定**

任意两种数制间的转换，选取哪种方法要根据具体情况来定。

如果是$2^i$进位制数之间的转换，当然选用"数码直接代换法"较简单。

如果是非$2^i$进位制数之间的转换，如由$\alpha$进位制数转换为$\beta$进位制数，这就要看你熟悉哪种进位制数的运算规则。

当熟悉$\alpha$进位制数的运算规则时，就采用"基数乘除法"；当熟悉$\beta$进位制数的运算规则时，就采用"多项式替代法"；若$\alpha$进位制数和$\beta$进位制数的运算规则都不熟悉，那就只好用十进制来搭桥，先把$\alpha$进制数转换为十进制数，再把十进制数转换为$\beta$进制数。

〔例1-12〕 将八进制数$(357.4)_8$转换为五进制数，并保留4位小数。

首先用"多项式替代法"，将$(357.4)_8$转换成十进制数。

$$\begin{aligned}(357.4) &= 3\times 8^2+5\times 8^1+7\times 8^0+4\times 8^{-1}=192+40+7+0.5\\ &=(239.5)_{10}\end{aligned}$$

再用"基数乘除法"将$(239.5)_{10}$转换为五进制数，整数部分用基数除法；小数部分用基数乘法。

```
                                          0.5
5| 239                                  ×   5
  5| 47  ...................... 4         2.5
     5| 9  .................... 2       ×   5
        5| 1  ................. 4         2.5
           0  ................. 1       ×   5
                                          2.5
                                        ×   5
                                          2.5
```

$$(239.5)_{10}=(1424.2222)_5$$

所以$(357.4)_8=(1424.2222)_5$。

## §1-3 十进制数的代码表示

在数字设备中，任何数据和信息都是用若干位"0"和"1"组成的二进制代码来表示的。n位二进制代码可以组成$2^n$个不同的码字，代表$2^n$种不同的信息或数据。但是，十进制是人们感到最方便的数制。有时人们希望数字设备直接用十进制数进行运算和显示运算结果。为了解决这个矛盾，可用特定的二进制码来代表每个十进制数字，这就是二进制编码的十进制。简称二-十进制码（Binary Coded Decimal Codes-BCD码）。

二-十进制码是以10为基数组成的，它属于十进制，但数字是用二进制码来表示的。为了表示十个数字，至少要用四位二进制码来表示。四位二进制码可以组成$2^4=16$种不同状态，从中取出十个状态来表示0～9十个数码，还有六种状态是不用的。根据数码所选状态的不

同，二-十进制码可以有很多种表示方法，最常用的有8421码、2421码、余3码等，在此仅介绍8421码、余3码及余-3格雷码。如表1-3所示。

**表1-3 常用二-十进制码**

| 十进制数 | 8421码 | 余-3码 | 余-3格雷码 |
|---|---|---|---|
| 0 | 0 0 0 0 | 0 0 1 1 | 0 0 1 0 |
| 1 | 0 0 0 1 | 0 1 0 0 | 0 1 1 0 |
| 2 | 0 0 1 0 | 0 1 0 1 | 0 1 1 1 |
| 3 | 0 0 1 1 | 0 1 1 0 | 0 1 0 1 |
| 4 | 0 1 0 0 | 0 1 1 1 | 0 1 0 0 |
| 5 | 0 1 0 1 | 1 0 0 0 | 1 1 0 0 |
| 6 | 0 1 1 0 | 1 0 0 1 | 1 1 0 1 |
| 7 | 0 1 1 1 | 1 0 1 0 | 1 1 1 1 |
| 8 | 1 0 0 0 | 1 0 1 1 | 1 1 1 0 |
| 9 | 1 0 0 1 | 1 1 0 0 | 1 0 1 0 |

8421码是最基本最简单的一种二-十进制码。它的特点是每个十进制数码0～9，分别用四位二进制数的等效值来表示，其编码方法如表1-3第二列的前十种状态所示。二进制数的最后六种状态（1010-1111），在8421码中是不允许出现的。

8421码的每一位二进制数码都有特定的权值，故称为有权码。8421码各位的权值从左至右分别为8、4、2、1，这也就是它名称的由来。对于有权码来说，它所代表的十进制数是：

$$N_{10} = \sum_{i=0}^{3} a_i w_i$$

式中 $a_i$ 是第i位上的二进制数码（0或1），$w_i$ 是i位的权。例如8421码1001所表示的十进制数为：

$$N = 1\times 8 + 0\times 4 + 0\times 2 + 1\times 1 = 9$$

即1001是9的代码。数字5的8421码为0101。故十进制数59用8421码表示时为01011001，凡是十进制数为奇数（1，3，5，7，9）时，它的二进制编码的最低位（即权值为的1位）都是1；凡是十进制数为偶数（0，2，4，6，8）时，它的二进制编码的最低位都是0，故8421码具有奇偶特征。因此，采用8421码的十进制数容易判别它的奇偶性。

余-3码是由8421码加3产生的。如十进制数6的8421码是0110，而用余-3码表示则为：0110+0011=1001。在这种编码方案中，十个码中的0和9、1和8、2和7、3和6、4和5的码字对应位恰好一个是0时，另一个就是1。具有这种特性的BCD码叫自补码。

余-3格雷码是格雷码（循环码）的一种。在某些应用中，用二进制码表示自然数时，希望任何相邻的两个代码之间（如3和4、7和8）仅有一位二进制数码不同，其余各位数码均相同，这种二进制编码叫格雷码（循环码）。它可为纯二进制格式，也可为BCD代码形式，表中余-3格雷码为BCD代码形式，它为无权代码。

## §1-4 逻辑代数的基本概念

逻辑代数是19世纪英国数学家狄·摩根（A. De Morgan，1806～1878年）和布尔（1815～1864年）把数学的形式化的方法应用到逻辑学领域而建立起的一门"应用数学"。它是数学

和逻辑学的互相渗透。

逻辑代数是一个由逻辑变量集K，常量0，1及"或""与""非"三种运算符所构成的代数系统，记为（K，+，·，0，1)。其中逻辑变量集是指逻辑代数中所有可能变量的集合，它可用字母表示，但每个变量的取值只可能为常量0或1。

### 1-4-1 逻辑函数

逻辑代数和普通代数一样，也有函数概念，称逻辑函数或简称函数。在逻辑电路中，如果它的一组输入变量与某一个输出变量间存在确定的对应关系，即当输入变量取某一组值时，输出变量就有一个确定的值与之对应，则该输出变量是此输入变量的逻辑函数。

逻辑电路中最基本的逻辑关系有"与"、"或""非"三种。对应地，逻辑代数中有"与"、"或""非"三个最基本的函数关系，常称为三种基本逻辑运算。

**1. "与"逻辑运算**

所谓"与"逻辑运算就是"仅当决定事件发生的所有条件（A、B、C…）均具备时，事件F才发生"。这种因果关系叫"与"逻辑运算，又叫逻辑乘。

如果条件A、B的取值为1表示条件具备，取值为0表示条件不具备。事件F作为A、B的函数也只有发生（用1表示）和不发生（用0表示）两种取值。自变量A、B和因变量F的所有可能的取值关系，可由表1-4表示。

**表1-4 "与"逻辑运算真值表**

| 输入 | | 输出 |
|---|---|---|
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

此表称为"与"逻辑运算的真值表。由于每个输入变量只有0、1两个取值，即若$A \neq 1$则$A=0$，或$A \neq 0$则$A=1$。因此n个变量的取值只有$2^n$种组合。真值表列出了所有可能的输入组合下逻辑运算的结果，所以它能完全确定逻辑运算的规律。同一个逻辑函数只可能有唯一的真值表。在列表时为了防止遗漏和重复，可采用二进制数的规律来排列输入变量的取值。

真值表1-4中输入和输出之间的四行关系，恰好相当于普通代数乘法所具有的如下规律：

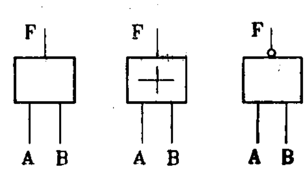
$$0\times0=0;\quad 0\times1=0;\quad 1\times0=0;\quad 1\times1=1$$

因此，"与"运算可写成：

$$F=A\cdot B \quad 或 \quad F=AB \qquad (1\text{-}1)$$

式（1-1）叫做"与"逻辑函数式。虽然表1-4只定义了两个变量的逻辑乘，但"与"运算的概念可以扩大应用于任意多个变量。

在数字电路中，用来完成"与"逻辑运算的电路叫"与"门。可用图1-1(a)所示的符号表示。



(a) (b) (c)

图1-1 "与"、"或"、"非"符号