·中国仪器仪表学会·

数字电路及报警系统

〔美〕T.G.菲塞尔

徐文立　朱善君
严继昌　董登武　　译
郑　维　敏　　校

机械工业出版社

内 容 简 介

本书共分十三章：第一、二章论述了各种数制和码制；第三、四章介绍了数字电路中常用的各种逻辑器件以及组成这些器件的电子元件；第五章到第七章讲解了继电器逻辑、固态逻辑、可编程序控制器；第八、九章分别讨论了组合逻辑和时序逻辑电路的设计原则和方法；第十章、十一章讲解信号器步序和报警联锁系统设计问题；第十二章归纳了系统文件的要求和意义；最后一章介绍了用可编程序控制器和计算机系统的通信接口和规程。

本书内容丰富，深入浅出，可供大专院校师生、工厂技术人员阅读，也可作为有关专业的参考教材及有关工程技术人员的知识开拓读物。

Alarm and Interlock System
Thmas G·Fisher
1984

数字电路及报警系统

〔美〕T.G.菲塞尔 著
徐文立 朱善君 严继昌 董登武 译
郑维敏校
责任编辑 马健华

●

机械工业出版社出版(北京阜成门外百万庄南里一号)
(北京市书刊出版业营业许可证出字第117号)
通县曙光印刷厂印刷
新华书店北京发行所发行 · 新华书店经售

●

开本787×1092 1/16 10·¹/₄ 印张·字数 255 千字
1985年12月北京第一版·1985年12月北京第一次印刷
印数 00,001—5000 定价2.60元

●

统一书号：15033·6283H

前　言

中国仪器仪表学会最近组织选译了一批美国仪器仪表方面的科学技术图书，这批书是为从事仪器仪表事业的科技人员和管理干部而编写的。它简明地叙述了仪器仪表的结构、原理和应用，总结了科研、生产、实践的经验。文字通顺易懂，实用性强，是仪器仪表工程技术人员很好的参考书。

这批书有下列五种：

1．《销售工程》　(美)小H·N鲁塞尔著

2．《数字器件的原理和应用》　(美)R·A基伯特著

3．《危险场所的电气安全与仪器防护》　(美)W·卡德尔著

4．《数字电路及报警系统》　(美)T·G菲塞尔著

5．《测试基础》　(美)R·L摩尔著

这批书将由机械工业出版社陆续出版，我们深信这批书的出版将会受到从事仪器仪表工程的技术人员和干部的欢迎，也会促进仪器仪表工业的发展。

《数字电路及报警系统》是由徐文立，朱善君、董登武、严继昌译，郑维敏校订。

由于水平所限，错误与不恰当之处在所难免，欢迎读者提供宝贵意见。

中国仪器仪表学会

1985年12月1日

目　录

第一章 数 制

任何用于计数和完成算术运算的符号和字符的集合都可以认为是一种数制。数制由表达数量所需要的不同数字的个数来表征。对十进制来说，需要十个数字：0、1、2、3、4、5、6、7、8和9。为了表明一个给定的量，任何一个数字都可以根据需要重复若干次，例如3333。

1-1 十进制

什么是十进制数？考察十进制数2315。这个数能够分解成：

$$2000+300+10+5=2315$$

而
$$2000=2\times10^3=2\times1000=2000$$
$$300=3\times10^2=3\times100=300$$
$$10=1\times10^1=1\times10=10$$
$$5=5\times10^0=5\times1=5$$

2315

（注意：$10^0=1$，因为任何数的零次幂等于1）。

数10被叫做这个数制的基（或称为底）。这个基等于该数制里不同字符的个数。因为字符本身从0开始，所以基总是比该数制里所用的最大字符多1个单位。数本身并不一定能表明它的基是什么。例如符号1010可能表示一个十进制数，也可能表示一个二进制数、八进制数或者十六进制数。为了避免混淆，不是用十进制写的数应该用基作为下标。例如在八进制中应写成$(1010)_8$。

为了得到十进制里的0，应记住：零乘以任何数结果还是等于零。考察数106。

$$106=(1\times10^2)+(0\times10^1)+(6\times10^0)$$
$$=100+0+6=106$$

在上述的数2315中，根据两个方面的因素，给每一位数字赋值。这两个因素是：(1)数字基本值，即它本身作为个位数所表达的值。(2)权值，它由该数字在数中的位置决定。在十进制中，其权值是1，10，100，1000等。不管用什么数制，具有最高权值的位在左边，而具有最低权值的位在右边。小数点（例：十进制中的十进制小数点）是决定权值的起始点。不管所用的是什么数制，紧挨小数点左边的位置，是个位位置，其权值为1。每向左移一位，其权值按照该数制的基的幂增加。反之，如果向小数点右边移动，每移一位其权值按照基的负次幂减少。例如：十进制数378.62是：

$$(3\times10^2)+(7\times10^1)+(8\times10^0)+(6\times10^{-1})+(2\times10^{-2})$$
$$=300+70+8+0.6+0.02=378.62$$

每一个数有最高有效位(MSD)和最低有效位(LSD)。最高有效位乘以最大权值，它对任何算术运算的结果有最大影响。最低有效位乘以最小权值，它对任何算术运算的结果影响最小。在上面的数378.62中，3是MSD，2是LSD。下面是决定MSD和LSD的规律：

整数：最左边非0的位是MSD，而最右边的位(不管是1还是0)是LSD。例如：

```
01 0 0 0 1 1 1 0 0 0
 MSD             LSD
```

小数：MSD处在紧挨小数点右边的位置。而LSD是最右边非零的位。例如

```
.0 1 0 1 0
  |   |
MSD LSD
```

混合数：MSD是最左边非零的位，LSD是最右边非零的位。例如：

```
01101.0110
   |     |
  MSD   LSD
```

数字a的反称为$\overline{a}$，它等于(b－1)－a。这里b是这个数制的基(或称底)。对于十进制，b－1＝10－1＝9，这是十进制中最大的数。例如6的反是9－6，即3。

1-2 二进制

二进制的基为2。它只需两个不同的数字，0和1，来表示任何数量。二进制广泛地用于计算机和开关电路中。在开关电路中只有二个状态：通或断。在开关电路中用二进制是十分方便的。例如：你可以用二进制的1表示通，用二进制的0表示断。在计算机中，应用二进制有许多优点，这是因为它只用二个状态来表示任何数，且进行二进制算术运算十分容易。

二进制数1001 0000 1011能通过在十进制数的讨论中所用的类似步骤转换成十进制数。例如：

$$
\begin{aligned}
1\,001\ \ 0000\ \ 1011 &= 1\times 2^{11} = 1\times 2048 = \quad 2048\\
&+0\times 2^{10} = 0\times 1024 = +\ \ 0\\
&+0\times 2^{9} = 0\times\ 512 = \ \ 0\\
&+1\times 2^{8} = 1\times\ \ 256 = +256\\
&+0\times 2^{7} = 0\times\ \ 128 = +\ \ 0\\
&+0\times 2^{6} = 0\times\ \ \ 64 = +\ \ 0\\
&+0\times 2^{5} = 0\times\ \ \ 32 = +\ \ 0\\
&+0\times 2^{4} = 0\times\ \ \ 16 = +\ \ 0\\
&+1\times 2^{3} = 1\times\ \ \ \ 8 = +\ \ 8\\
&+0\times 2^{2} = 0\times\ \ \ \ 4 = +\ \ 0\\
&+1\times 2^{1} = 1\times\ \ \ \ 2 = +\ \ 2\\
&+1\times 2^{0} = 1\times\ \ \ \ 1 = +\ \ 1\\
&\qquad\qquad\qquad\qquad\qquad\qquad 2315
\end{aligned}
$$

所以，二进制数1001 0000 1011等于十进制数2315。即：

$(1001\ 0000\ 1011)_2=(2315)_{10}$

a的反$\overline{a}$，在二进制数制中等于$(2-1)-a=1-a$。所以0的反是1。1的反是0。

组成一个二进制数的一组 0和1 叫做字。每个 0或1 称为一个位。例如：前面的二进制数1001 0000 1011是一个12位的二进制字。而二进制数1101是一个4位的二进制字。

1-3 八进制

八进制的基是8。表达数量需要8个数字：0、1、2、3、4、5、6、7。在十进制中权值为10^3，10^2等，而八进制涉及的权值为8^3，8^2等。因为8是2的整数次幂$(8=2^3)$，因此从二进制到八进制以及从八进制到二进制的转换，是个很简单的过程。如下所示：

八进制到二进制	二进制到八进制
$(4\ \ 5\ \ 7)_8$	$(100\ 101\ 111)_2$
$(100\ 101\ 111)_2$	$(4\ \ 5\ \ 7)_8$

一位八进制数字等效于三位二进制数字。这个关系用在数字机里有许多便利之处，因为八进制能代替较麻烦的二进制，它是数字机中的实际代码。

数2660(八进制)，在十进制中是多少？

$$(2660)_8=(2\times8^3)+(6\times8^2)+(6\times8^1)+(0\times8^0)$$
$$=(2\times512)+(6\times64)+(6\times8)+(0\times1)$$
$$=1024+384+48+0=(1456)_{10}$$

所以在八进制中的2660等于十进制中的1456。

在八进制中，a的反为$\overline{a}=(b-1)-a=7-a$。例如八进制中3的反等于7-3，即4。

1-4 十六进制

十六进制的基为16。用了十进制中的十个数字和英文字母中前六个字母（见表2-1）。16也是2的整数次幂$(16=2^4)$，所以一个十六进制的数字其值和4位二进制数字等效：

十六进制到二进制	二进制到十六进制
$(D\ \ \ \ 7)_{16}$	$(1101\ 1000\ 0111)_2$
$(1101\ 1000\ 0111)_2$	$(D\ \ 8\ \ 7)_{16}$

十六进制越来越多地被用在现代数字机里。

表1-1数制的比较

十进制	二进制	八进制	十六进制
0	0	0	0
1	01	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20

1-5 计数

表1-1对十进制数 0 到 32 以及对应的二进制、八进制和十六进制数进行了比较。注意，在其他数制中的计数同在十进制数制中一样。

以八进制计数为例：

1. 从0开始，加1到最低有效位，直到所有基本字符都被用到：0、1、2、3、4、5、6、7。

2. 因为 7是最大字符，所以下一个数将需要2个数位，把LSD变为0，在这个0的左边添1：…6，7，10。

3. 继续加1到LSD，直到再次达到最大字符：…6，7，10，11，12，13，14，15，16，17。

4. 再次把LSD变为0，并且在它的前一位上加1：…16，17，20。

5. 用这种方式继续加1，直到这两位数字均达到最大：…26，27，30…36，37，40…46，47，50…56，57，60…66，67，70…76，77。

6. 把这两位变为0，并且在它们的前一位上添1：…76，77，100。以此类推：…776，777，1000等。

1-6 基的转换

把一个数从基b_1转换成基b_2时，有两种情况需要考虑。第一种情况，b_1小于b_2，这时可以按前面用过的将八进制或二进制转换成十进制的方法。这个方法是将该数用b_1的幂来表达，然后根据b_2的算术运算相加。第二种情况，b_1大于b_2，例如从十进制转换到二进制或八进制。则采用下面的步骤：用基b_2除该数，得出余数。然后再用基b_2除上面的商，再次得出余数。这个步骤一直继续到商为0。这些余数就是所求的基b_2下的数。十进制数2315转换成八进制和二进制的例子如下所示：

十进制到八进制

2315/8 = 289，余数3 ↑
289/8 = 36，余数1
36/8 = 4，余数4
4/8 = 0，余数4

$(2315)_{10} = (4413)_8$

十进制到二进制

2315/2 = 1157，余数1 ↑
1157/2 = 578，余数1
578/2 = 289，余数0
289/2 = 144，余数1
144/2 = 72，余数0
72/2 = 36，余数0
36/2 = 18，余数0
18/2 = 9，余数0
9/2 = 4，余数1
4/2 = 2，余数0
2/2 = 1，余数0
1/2 = 0，余数1

$(2315)_{10} = (100100001011)_2$

这个步骤只适用于整数。对小数必须采用不同的步骤。其步骤是：

1. 用基b_2乘以这个小数，如果积小于1，则在基b_2下的第1位数为0。如果积大于1，则在新数制中的第1位数是这个积的整数部分。

2. 用基b_2乘以这个积的小数部分，继续上面的步骤，以确定下一位数。

3. 继续这个步骤，直到以b_2为基的数有了所需的位数为止。因为在很多情况下，上面的过程不会恰好终结。

例：十进制到八进制

转换$(0.5125)_{10}$到以8为基的数。

$0.5125\times8=4.100$　　整数为 4

$0.100\times8=0.800$　　整数为 0

$0.800\times8=6.400$　　整数为 6

$0.400\times8=3.200$　　整数为 3 ↓

所以，$(0.5125)_{10}=(0.4063)_8$

例：十进制到二进制

转换$(59.65)_{10}$到二进制（先转换整数部分，然后转换小数部分。整数部分用除，小数部分用乘）。

59/2 = 29，余数 1 ↑	$0.65\times2=1.30$，整数 1
29/2 = 14，余数 1	$0.30\times2=0.60$，整数 0
14/2 = 7，余数 0	$0.60\times2=1.20$，整数 1
7/2 = 3，余数 1	$0.20\times2=0.40$，整数 0
3/2 = 1，余数 1	$0.40\times2=0.80$，整数 0
1/2 = 0，余数 1	$0.80\times2=1.60$，整数 1 ↓
$(59)_{10}=(111011)_2$	$(0.65)_{10}=(0.101001)_2$

$(59.65)_{10}=(111011.101001)_2$

将这个二进制数重新转换成以10为基的数，得到59.64。所以上面的转换是不精确的。

对于二进制到八进制和八进制到二进制的转换，可以采用一个较简单的转换步骤。因为$8=2^3$，每位八进制数字能由三位二进制数字表示。例如$(5)_8=(101)_2$。为了把一个二进制转换成八进制，把二进制数从小数点开始分成三位一组，然后决定每组对应的八进制数字。

例：二进制到八进制

转换$(100100001011.100)_2$到八进制数。

100 100 001 011.100

4　4　1　3．4

所以，$(100100001011.100)_2=(4413.4)_8$

例：八进制到二进制

转换$(325.65)_8$到二进制数。

3　2　5　·　6　5

011 010 101 .110 101

所以，$(325.65)_8=(011010101.110101)_2$

1-7　二进制算术运算

下面给出基本的算术运算：

位		加		减		乘
a	b	进位	和	借位	差	积
0	0	0	0	0	0	0
0	1	0	1	1	1	0
1	0	0	1	0	1	0
1	1	1	0	0	0	1

二进制的加法和十进制加法类似。对应位相加，如果得到的进位是1，该**进位**被加到左边的二进制位上。例如：1 + 1 在十进制中等于2，但在二进制中是10，得到的和为0，进位为1。

```
  1
 +1
----
 10
 | |
进位 和
```

又例如：

```
    1001←————进位
    1001.01 = (9.25)10
  + 1101.10 = (13.50)10
  ------------------
    10110.11 = (22.75)10
```

二进制减法也用和十进制减法十分相象的方法来实现。为了从0减去1，要从左边的位中借一个1。如果有借位，并且左边下一位的数字是1，则把它变成0以后继续做减法。然而，如果左边下一位的数字为0，则把它变为1，并且左边连着的每一个0都要变为1，直到左边出现第一个为1的位，把它改为0，再继续进行减法。

例如：

```
    1001←————借位
    10110.11 = (22.75)10
  - 01001.01 = (9.25)10
  ------------------
    01101.10 = (13.50)10
```

二进制减法也能用称为补码加的减法形式来实现。

这种方法在计算机里是比较容易实现的，因为它使得计算机里的加法器既可以做加法也可以做减法。用补码加的二进制减法，是通过把减数转换成它的补码，然后和另外一项(被减数)相加来完成的。把二进制数中的每一个1或0改写成它的反码，并在其最低位上加1，便得到该二进制数的补码。

例如：补码加法

（二进制减法）

```
 10110  （被减数）        (22)10
-01001  （减数）          (9)10
-------
 10110  （减数的反码）
+    1
-------
 10111  （减数的补码）
+10110  （被减数）
-------
101101  （差）            (13)10
```

上面最高位的数字1（底部划线的）是一个超出减数和被减数的最高位的进位位。这个进位位表明答案是正的。将其去掉后，留下来的数即是差。如果在这个位置上没有1（意味着是0），它表明答案是负的。为了得到差，必须对它求补。

```
 01101   （被减数）        (13)10
-10110   （减数）          (22)10
-------
 01001   （减数的反码）
+    1
-------
 01010   （减数的补码）
+01101   （被减数）
-------
010111
```

这里没有进位，所以答案是负的，要得到答案，必须对它求补。

```
 01000
+    1
-------
-01001        (-9)10
```

在二进制中乘法是十分简单的，实际上它只涉及到加法。

例如：乘法

```
          1101.10        =(13.50)10
        × 1001.01        =(9.25)10
        ---------
          110110
         000000
        110110
       000000
      000000
     +110110
     ---------
     1111100.1110        =(124.875)10
```

除法用下面的方法实现。