高等院校教学用书

数字电路与微计算机基础

武汉水利电力学院
刘　勇　张立宏
陈邕怀　肖咏华　合编

湖北科学技术出版社

高等院校教学用书

数字电路与
微计算机基础

封面设计：卢国荣

高等院校教学用书

数字电路与微计算机基础

武汉水利电力学院

刘　勇　张立宏
张邕怀　肖咏华　合编

湖北科学技术出版社出版发行

武汉水利电力学院印刷厂印刷

787×1092毫米32开本　12.25印张　283,000字

1985年5月第1版　1985年5月第1次印刷

印数：1—15,000

统一书号：15304·80　　定价：2.50元

目　录

前　言

微计算机自七十年代初期以崭新的姿态进入电子技术的历史舞台后，就以其功耗低、体积小、功能强、可靠性高以及成本低廉一系列优异的性能，强烈地冲击着科研、国防、工农业生产和人们生活等各个方面。它的应用领域还处在不断开拓之中，以致人们难以预料它将来的应用范围。它被称之为世界第四次工业革命的推动者，极大地扩展了人类的智能。目前，迅速在各技术领域中推广使用微计算机已是当务之急，特别对于高等工科院校的学生，应具有在所学专业的学术领域内利用微机实现自动测量和控制的水平。因此，本书与《电工技术基础》、《模拟电子电路》所构成的全套教材可作为非电专业的必修课程。

本书先由一个简单的模型机入手，由浅入深地阐明了计算机的概念和原理，然后以Z－80八位机为对象，讨论机器的指令系统，并以汇编语言程序设计为中心，介绍软件的基础知识以及侧重微机应用介绍接口技术。同时，本书还较详细地讲解了数字电路基础，它是深入学习和应用微机必不可少的知识。本书的编写力求通俗易懂，注重实际，能用较少的学时完成上述内容。

本书共分八章，第一章由肖泳华同志编写，第二章由陈邕怀同志编写，第三章至第六章由刘勇同志编写，第七章、第八章由张立宏同志编写。由于我们对先进的电子计算技术了解不够，又缺乏足够的教学实践，书中必然存在不少缺点和错误，殷切希望各方面的读者能给以批评和指正。

本书初稿经葛州坝水电工程学院副教授陈跃宗同志及蔡品璐、余良春同志，华中工学院副教授邓星钟同志，武汉水利电

力学院周洞汶同志审阅，并在武汉电工理论学会电工应用委员会主持召开的审稿会上交换了修改意见，经编者修改后，又进行了复审。全书由陈跃宗、邓星钟同志任主审。另外，特邀靳希同志担任本书编辑及统稿工作。由于以上全体人员的辛勤劳动，使本书质量得以提高。在此，谨向他们致以诚挚的谢意。

编者

1985年2月

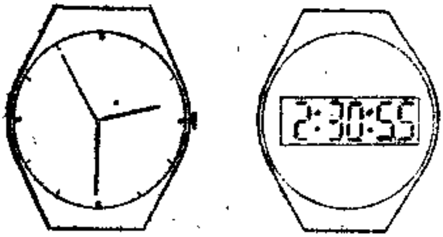
第一章　数字电路基础

引　言

在生产实践和人们生活中所遇到的物理量大部分具有连续变化的特点，如时间、温度、压力等。在模拟电子电路中所采用的就是连续变量，即电路中接受和传递的信号（电压或电流）均用以模拟这些连续变量。但是，在人们会话、数据处理、观测量的记录以及人类活动的许多场合中，并不能使用这些连续变量的准确值，而必须用一些数字量近似地代表它们的实际值。例如，测得的室内温度为20℃，而真正的室温可能为20℃到21℃之间的某一值，实际上这种测量过程就是变量的数字化过程。也就是说在某一指定的时刻，用表示变量大小的数字量代替原始的连续变量，而当原始变量转换成数字形式后，可以进行处理或者存贮，而不会进一步降低其精度。因此，数字式电子系统是处理数字（或数值）形式信息的系统。它是用处理离散量的方法来进行工作的。离散量为整数、纯小数或者带小数的数。

人们对这种连续变量到数字量的变换过程并不陌生。举例来说，当被问到，"你的车现在开得多快"时，驾驶员看着测速计（一种连续的测量仪表）回答道，"每小时六十公里"。这很可能不是准确的车速，但为了用数字形式说出来，他必须将测速计连续的指示值变换成相应的数值。又例如用模拟形式表示时间已经有数百年的历史了，图1－1(a)所表示的时间是二点三十分五十五秒。而近几年来，人们大量使用一种测量和显示时间的全新方法，它是以石英晶体振荡器为基准，把电

路及显示技术结合在一起，将时间信息用数字表示出来，如图1—1（b）所示。



（a） 模拟方式　　　（b） 数字方式

图1—1 时间的表示方式

信息的数字处理在近十几年内应用得更加广泛，除了数字仪表、数字钟和计算器外，大量用于微计算机中。

数字式电子系统中的信息是用二元数表示的，即用二元数“1”表示“真”，而用二元数“0”表示“伪”。只用二个数字的运算系统称为二元系统。所以数字电路中的晶体管，大部工作在特性曲线的放大区以外，进入接近开路（OFF）或短路（ON）状态。复杂的数字电路及微计算机的硬件实际上是由大量的基本逻辑门电路、锁存器、寄存器、计数器等以适当的方式组成。逻辑门本身只完成所要求的“0”和“1”两种状态的逻辑运算。在寄存器和计数器中也只用“0”和“1”两个数码，并按二进制数制存贮和计数。因此，人们把工作在数字信号下的电子电路简称为数字电路。

本章将比较系统地介绍各种基本数字电路（包括逻辑门、触发器、数字部件等）的工作原理，重点放在电路状态间逻辑关系的分析上。采用逻辑代数、真值表、卡诺图等作为分析方法。由于数字电路不仅能对信号进行算术运算，而且具有逻辑思

维(逻辑推算、逻辑判断)能力，因而使计算机的制造成为可能。通过本章的学习，将打下学好微计算机所必须具备的基础知识。

1—1 数 制

十进制是常用的数制，而在数字电路及微计算中使用的数制有二进制、八进制、十六进制等。本节首先介绍这些数制，然后说明它们之间的转换及编码方法。

一、进位计数制

1．十进制数

人们最熟悉的数制是十进制，它有十个不同的数字（0 1 2 3 4 5 6 7 8 9）和一个小数点符号。数字的个数称为**基数**，那么十进制的基数就是10。每个数字所代表的值取决于它在数中的位置，称作**位置值**。例如十进制数567.89中每个数字的位置值是：

5 6 7 . 8 9

9	位置值是 $9\times10^{-2}=$	0.09
8	位置值是 $8\times10^{-1}=$	0.80
7	位置值是 $7\times10^{0}=$	7.00
6	位置值是 $6\times10^{1}=$	60.00
5	位置值是 $5\times10^{2}=$	500.00
	总计	567.89

所以这个数的多项式表示形式为：

$$5\times10^{2}+6\times10^{1}+7\times10^{0}+8\times10^{-1}+9\times10^{-2}=567.89$$

式中10^{2}、10^{1}、10^{0}、10^{-1}、10^{-2}称为各位数的**权**(Weight)，它们

都是10的幂，小数点右边各位数的权具有负指数。乘上了系数（即0～9十个数字符号）的权叫做加权系数，十进制数的数值即为各加权系数之和。

十进制数的另一个特征是，当两数相应位的和大于9（即最大单个数值符号的值）时，可按“逢十进一”的原则表示这个数。例如8.7+2.4=11.1

显而易见，如果数中的各数字符号左移一位，数的值增大10倍，而右移一位则减小10倍。十进制数567.89左移一位为5678.9，右移一位则为56.789。

2．其它进制数

其它进制（如十六进制、八进制、二进制等）的数，具有与十进制数相类似的特征，可归纳为如下几点：

（1）M进制数具有M个数字和一个小数点符号，M即为该进制数的基数。

M=16亦为十六进制数，它的数字符号除0～9外，还需加上A、B、C、D、E、F六个符号。

M=8亦为八进制数，它的数字符号是0～7。

M=2亦为二进制数，它的数字符号只有两个，即0与1。

（2）M进制数中最大数字符号为基数M减1。则

二进制数最大数字符号是1

十进制数最大数字符号是9

八进制数最大数字符号是7。

十六进制数最大数字符号是15，即为F。

（3）M进制数的加法规则是“逢M进一”，减法规则是“借一当M”。例如

二进制数：　$10_{(2)}^{*}+11_{(2)}=101_{(2)}$　（逢二进一）

$10_{(2)}-01_{(2)}=01_{(2)}$　（借一当二）

八进制数：　$54_{(8)}^{*}+76_{(8)}=152_{(8)}$　　　（逢八进一）

$62_{(8)}-37_{(8)}=23_{(8)}$　　　（借一当八）

十六进制数：$5A_{(16)}^{*}+FB_{(16)}=155_{(16)}$　　（逢十六进一）

$C4_{(16)}-6B_{(16)}=59_{(16)}$　　（借一当十六）

（4）将M进制数中的任何数按权展开，可得到相应的十进制数。例如二进制数：

1 0 1 1 1 . 1 0 1

位置值是 $1\times2^{-3}=0.125_{(10)}^{*}$

位置值是 $0\times2^{-2}=0.0_{(10)}$

位置值是 $1\times2^{-1}=0.5_{(10)}$

位置值是 $1\times2^{0}=1.0_{(10)}$

位置值是 $1\times2^{1}=2.0_{(10)}$

位置值是 $1\times2^{2}=4.0_{(10)}$

位置值是 $0\times2^{3}=0.0_{(10)}$

位置值是 $1\times2^{4}=16.0_{(10)}$

总计　　23.625

即

$$10111.101=1\times2^{4}+0\times2^{3}+1\times2^{2}+1\times2^{1}+1\times2^{0}+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=23.625$$

式中2^{4}、2^{3}、2^{2}、2^{1}、2^{0}、2^{-1}、2^{-2}、2^{-3}是二进制数各位数字的权。

又例如八进制数：

$$6752.43_{(8)}=6\times8^{3}+7\times8^{2}+5\times8^{1}+2\times8^{0}+4\times8^{-1}+3\times8^{-2}=3562.15625$$

*：注脚(2)、(8)、(10)、(16)表示二进制、八进制、十进制、十六进制。

式中8^3、8^2、8^1、8^0、8^{-1}、8^{-2}是八进制数各位数字的权。

又例如十六进制数：

$$5DC3.AF_{(16)}=5\times16^3+D\times16^2+C\times16^1$$
$$+3\times16^0+A\times16^{-1}+F\times16^{-2}=24003.68359$$

式中16^3、16^2、16^1、16^0、16^{-1}、16^{-2}是十六进制数各位数字的权。

表1—1　用不同的数制表示十进制数0至20

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	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

二、两种进位制数的转换

用十进制数非常方便，但要区别出十个明显的电平或状态所要求的电路结构却十分复杂。因此在数字电路及微计算机中，广泛采用二进制数运算。这是因为它只利用“0”和“1”两个数字，这在电路中很容易实现，即可以用0伏表示“0”，用另一个电压值（例如5伏）表示“1”。那么在计算机的输入端就需要把十进制数转换为二进制数，而在输出端再把二进制数转换成十进制数。然而二进制数也有不便之处，当一个给定量所需要的位数很多时，不仅书写冗长，而且读起来也相当麻烦，容易出错。如果把多位二进制数分成三位或四位一组，并表示为相应的八进制或十六进制数，则便于读写、记忆。另外计算机的输入指令和数据常采用八进制或十六进制，因此需要熟悉不同进位制数间的转换。两种进位制数间的转换可以采用不同的方法进行，这里仅介绍两种常用的方法。

1．先将A进制数转换为十进制数，再将十进制数转换为B进制数，从而实现A进制数对B进制数的转换。

第一步：将A进制数转换为十进制数

采用多项式运算法即可将任意进制数转换为十进制数。

例如二进制数$101101_{(2)}$

$$101101_{(2)}=1\times2^5+0\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0=45_{(10)}$$

又例如八进制数$15.5_{(8)}$

$$15.5_{(8)}=1\times8^1\times5\times8^0+5\times8^{-1}=13.625_{(10)}$$

第二步：将十进制数转换成B进制数

采用的方法是：整数部分除以待转换数的基数取

其余数，小数部分乘以待转换数的基数取其整数。最后把所得结果相加即得。

例如将上述十进制数$13.625_{(10)}$转换为二进制数。整数部分的转换：

$$
\begin{array}{r|l}
2 & 13 \quad \text{—— 余 } 1 \quad \text{低位} \\
\hline
2 & 6 \quad \text{—— 余 } 0 \quad \uparrow \\
\hline
2 & 3 \quad \text{—— 余 } 1 \\
\hline
2 & 1 \quad \text{—— 余 } 1 \quad \text{高位} \\
\hline
 & 0
\end{array}
$$

相除之商为零，则运算完成。从下至上读各余数，得到

$$13_{(10)} = 1101_{(2)}$$

小数部分的转换：

$$
\begin{array}{lll}
 & & 0.625 \\
 & & \times \quad 2 \\
\hline
\text{高位} & \text{取整数 } 1 & \longleftarrow 1.250 \\
 & & \times \quad 2 \\
\hline
\downarrow & \text{取整数 } 0 & \longleftarrow 0.500 \\
 & & \times \quad 2 \\
\hline
\text{低位} & \text{取整数 } 1 & \longleftarrow 1.000
\end{array}
$$

由于小数部分为零，表明运算完成。从上至下读各小数点左边一位数字，得到

$$0.625_{(10)} = .101_{(2)}$$

则 $\quad 1101_{(2)} + 0.101_{(2)} = 1101.101_{(2)}$

通过以上两步转换，即把八进制数$15.5_{(8)}$转换成二进制数$1101 \cdot 101_{(2)}$。

2．以2^n为基数的进位制之间数的转换

若n=1即为二进制数，n=3即为八进制数，n=4即为十六进制数。这三种进位制之间数的转换，可以先把八进制数（或十六进制数）转换为二进制数，然后再把二进制数转换为十六进制数（或八进制数）。

由于在八进制中有八个数字符号（0～7），若用三位二进制数字表示：即相应为000～111。

【例1—1】将八进制数$37.4_{(8)}$转换成二进制数。

解：八进制数　3　7　.　4

二进制数　011　111　.　100

略去整数前面和小数后面的零，即得到

$$37.4_{(8)}=11111.1_{(2)}$$

【例1—2】将八进制数76435.425转换成二进制数

解：八进制数　7　6　4　3　5　.　4　2　5

二进制数　$\overbrace{111}$ $\overbrace{110}$ $\overbrace{100}$ $\overbrace{011}$ $\overbrace{101}$. $\overbrace{100}$ $\overbrace{010}$ $\overbrace{101}$

即$76435.425_{(8)}=111110100011101.100010101_{(2)}$

如果要把二进制数转换成八进制数，则是上述转换的逆过程，即从二进制数的小数点开始，把二进制数的整数部分和小数部分，分别分成三位一组，每组即表示一位八进制数。

【例1—3】把二进制数111101.1转换成八进制数。

解：二进制数　$\underbrace{111}_{7}$　$\underbrace{101}_{5}$　.　$\underbrace{100}_{4}$

八进制数　7　5　.　4

值得指出是，在二进制数小数部分的尾部或者是整数部分的首部添1个至几个零，并不影响该二进制数的值。

所以　$111101.1_{(2)}=75.4_{(8)}$

同样道理，由于十六进制数是以$(16=2^4)$为基数，要把十六进制数的每位数字表示出来，则需要用到四位二进制数，

即为0000～1111。所以把二进制数转换成十六进制数的方法是：从二进制数的小数点开始，把它的整数部分和小数部分，各分成四位一组，对于两端不足四位者，可以添零凑齐。

【例1—4】把二进制数111100·111转换成十六进制数。

解： 二进制数 0011 1100·1110

十六进制数 3 C · E

即$111100.111_{(2)}=3C.E_{(16)}$

【例1—5】把例[1—2]中的二进制数转换成十六进制数。

解： 0111 1101 0001 1101 · 1000 1010 1000

7 D 1 D · 8 A 8

实际上，此例把八进制数$76435\cdot425_{(8)}$转换成了十六进制数$7D1D\cdot8A8_{(16)}$。

如果把十六进制数转换成二进制数，则是上述转换的逆过程，即从小数点开始，把每位十六进制数的数字都用四位二进制数表示。

【例1—6】将十六进制数E8·7转换成二进制数。

解： 十六进制数 E 8 · 7

二进制数 1110 1000 · 0111

即$E8.7_{(16)}=11101000.0111_{(2)}$

三、原码、反码与补码

数的正负号通常用符号"+"和"-"表示，例如十进制负数345，用符号"-"和数值345表示，即-345。同样，对于二进制数也用符号后面跟着数值（或称绝对值）的方法来表示。由于二进制只有"0"和"1"两个符号，如果用"0"表示"+"，则"1"表示"-"。这个"0"或"1"称为这个数的符号位(Sign bit)，它设在最高位。对于8位的计算机，一个带符号位的数，如果最高位（第八位）是0，表明是正