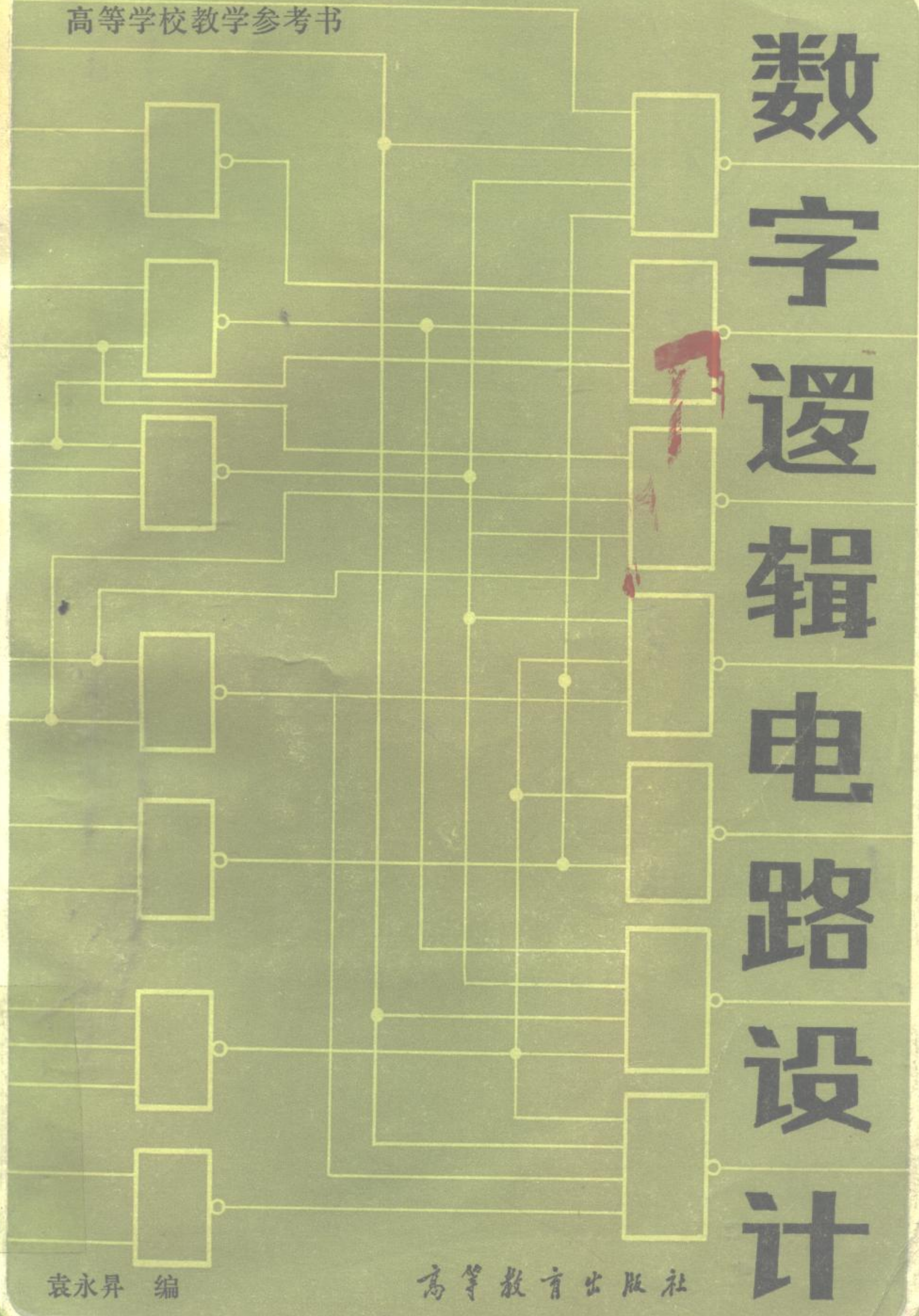
高等学校教学参考书

数字逻辑电路设计



袁永昇 编

高等教育出版社

高等学校教学参考书

数字逻辑电路设计

袁永昇 编

高等教育出版社

本书是编者在多年教学讲稿的基础上加以充实和完善而写成的，内容以中小规模集成电路的逻辑分析和综合为主，同时介绍了大规模集成电路在逻辑设计中的应用。本书从逻辑代数、逻辑器件到集成逻辑部件，从逻辑设计到电路设计，形成了一个较完整的整体。最后，用微处理机系统对“布线逻辑”和“程序逻辑”进行了对比，从而指出了逻辑设计的新方向。

本书取材较新，系统性较强，既注意到了逻辑推理又联系了一定的实际应用，可作为理科和师范院校无线电专业的试用教材或物理专业的选修课教材，[illegible]也可供从事电子技术的工程技术人员参考。

本书责任编辑：郭玉凤

高等学校教学参考书

数字逻辑电路设计

袁永昇 编

*

高等教育出版社出版

新华书店北京发行所发行

二二〇七印刷厂印装

*

开本 787×1092 1/16 印张22 字数500,000

1984年1月第1版 1984年8月第1次印刷

印数 [illegible],001—8,340

书号 13010·0969 定价 3.55 元

前　言

近些年来很多院校的数字技术课程在逐渐加强或相继开设，一些工科院校也先后编写了“脉冲技术与数字电路”或“数字逻辑”等有关教材。我们根据理科院校的教学特点和要求，草拟了一份“数字逻辑电路设计”的编写大纲。在这个大纲的基础上，经过了多次教学实践，逐步充实和完善，最后形成了“数字逻辑电路设计”一书。全书共分五个部分：

第一部分，绪论：主要介绍了数字技术的概貌以及数制和编码，为学习本书提供了必要的预备知识。

第二部分，逻辑代数基础：从数字逻辑的基本观点出发，联系常见的逻辑命题，由浅入深地介绍了逻辑代数的基本概念、运算，以及公理和定理等。它是逻辑分析和综合的数学工具。

第三部分，逻辑器件简介：以 TTL 集成电路为主，并联系到逻辑器件的新发展，从开关器件、开关信号和开关电路的角度介绍了常用逻辑器件的外特性。它是逻辑分析和综合的物理基础。

第四部分，逻辑电路的分析和综合：这部分系统地介绍了组合逻辑、同步时序逻辑和异步时序逻辑电路的分析和综合方法，以及使用物理器件构成逻辑电路时的过渡特性。同时也介绍了常用中规模集成组合逻辑电路和时序逻辑电路的构成原理、特点和应用。

第五部分，用大规模集成电路实现逻辑设计：简要地介绍了一些大规模器件及其在逻辑设计中的应用。最后通过 MPU 系统结束了布线逻辑并指出逻辑设计的新方向—程序逻辑。

这本书在内容取舍、安排和处理上有以下几个特点：

1. 舍弃了脉冲技术中与数字电路关系不大的部分，选取其一部分以“逻辑器件简介”作为本书的一章。这样做的结果使电路和器件得到了有机的联系，设计理论得到充实；使数字技术从理论到实践，从分析到综合成为一个整体。

2. 书的开头没有把“逻辑代数”作为“集代数”的特例来引入，而是从数理逻辑的角度由浅入深地过渡到逻辑代数。这样，对于学习本书的学生不必要求更多的数学预备知识，同时便于读者自学。

3. 为了培养学生的逻辑推理能力，打好较充实的专业基础知识，尽量注意学科知识的系统性和逻辑推理的严密性。同时又力求叙述简明并通过练习和实验联系一定的工程实际。

4. 内容以讲述中小规模集成电路的逻辑分析和综合为主，由小规模集成电路逻辑部件的设计引伸到中规模集成部件，同时也介绍了一些大规模集成电路在逻辑设计中的应用。实验部分则以中、小规模集成电路为主，同时可根据学校条件适当开设大规模集成电路的应用实验。

5. 通过“大规模集成电路及其在数字逻辑设计中的应用”结束布线逻辑；由“MPU 系统”引入了程序逻辑的基本概念，从而起到了承上启下的作用。通过逻辑处理中的布线逻辑方式和程

序逻辑方式的比较，可使读者对数字逻辑电路设计和程序设计得到较全面的认识，并对进一步学习程序逻辑方式有所启发。

6. 书中内容取材较广。为了帮助读者理解书中主要内容除配备各种类型的例题外，还搜集了大量的习题和思考题以便读者练习。

7. 本书授课时数伸缩性较大(40～120学时均可使用)，前五章是必讲内容，第六章以后各章均可作为选讲部分。凡画有*号的章、节、段及习题均可适当取舍。

本书从制定大纲到最后定稿，曾得到北京大学无线电系褚天鹏副教授和王洪涛等同志以及东北师范大学物理系姜兴波和郝会新副教授的帮助，特别是褚天鹏副教授和王洪涛同志审阅了本书的原稿，此外华东师大和北师大的有关同志还协助审阅了部分修改稿，提出了很多宝贵意见和详细的修改方案，在此谨致以诚挚的谢意！

本书虽然经过了几次教学实践，但由于笔者水平有限，编写的时间又很仓促，难免有许多缺点和错误之处，恳切希望批评指正。

编　者

1983年12月

目　录

第一章 绪 论

§1. 模拟系统和数字系统

人们生活在物质的世界中。在人们的生活和生产活动中，总是和各种各样的“信息”打交道。信息的传递、加工和处理方式，随着社会生产力的不断前进具有不同的含意。

我国古代为了监视外敌的入侵，从边疆到京城修筑了很多烽火台，用烽火的“有”、“无”来表示是否有敌人入侵。这虽然是传递信息的一种原始方式，但却具有简单而准确的特点。类似于这种传递信息的方式至今有些领域还在延用着，如海军使用的“旗语”、交通警察使用的“手势”等等。这是一种用简单的符号来表达一定含意的一种传递信息的方式。

随着社会生产和科学技术的不断发展，信息的传递、加工和处理方式也在不断地改进。继这种符号式的传递方式之后，出现了信息的传递、加工和处理的模拟方式。“信息”是物理量数值特征的表征，所谓“模拟”就是对这些物理量的“仿真”。自然界中的物理量绝大多数都是连续变化的，这里的连续是指：

(1) 时间上连续——这些量随时间连续变化。

(2) 幅值上连续——这些量的幅值也是连续变化的。

随着“电学”和“电子学”的出现，人们往往把其它物理量先通过“换能器”（又叫传感器）转换为“**电模拟量**”，然后再对“电模拟量”进行处理。我们把对电模拟量进行传递、加工和处理的实际工程系统称作“电模拟系统”，简称为“**模拟系统**”。模拟通讯（如有线电话、无线电广播和电视等）、模拟控制、模拟电子计算机等都是这种模拟系统。

近年来，由于数字技术的发展，采用“数字技术”来传递、加工和处理信息的实际工程系统不断出现。我们把这样的实际工程系统称作“数字逻辑系统”，简称为“**数字系统**”，PCM 通讯系统、数字控制系统及数字电子计算机等都是数字系统。用“数字系统”来处理物理量，必须首先把“模拟量”在时间上和幅值上进行离散化，使“模拟量”变成“**数字量**”。

“数字系统”和“模拟系统”相比具有以下几个优点：

(1) 精度高：只要设备量允许就可以得到很高的精度，如数字电子计算机字长若是 64 位就相当于十进制 19 位，精度可达 10^{-19}，这是模拟系统难以达到的。

(2) 抗干扰性强：传递、加工和处理的信息只有“0”和“1”两个电平，与模拟系统相比既可靠又准确。

(3) 保密性好：在数字通讯中可以进行加密处理，这一点尤其适于军用。

(4) 有处理本领：可以对信息进行存贮和判断。

(5) 通用性强：可以采用标准化部件来构成各种各样的数字系统。

缺点是设备量大、设备费用高，但是由于集成电路的飞速发展，新型电子器件的不断出现，集成度愈来愈高，价格则愈来愈便宜，因此“数字系统”对“模拟系统”的竞争力也愈来愈强，在某些领域有逐渐取代“模拟系统”的趋势。

§2. 数字逻辑发展简史

数字逻辑的理论基础起源于 19 世纪。1847 年英国数学家布尔(George Boole)在研究逻辑问题时企图用数学表述而发展起数学领域中的一个新的分支——“布尔代数”，又称为“逻辑代数”。它是用来表示二值性函数关系的，是数字逻辑电路设计的数学基础。

到了 20 世纪 30 年代，山农(C. E. Shannon)用布尔代数分析继电器电路，开始用于工业及通讯(如拨号式自动交换系统)，后来经过人们逐渐充实和丰富建立了“开关理论”，奠定了数字逻辑电路设计的理论基础和初步形成了一套分析和综合的方法。

30 年代为了解决通讯中的干扰问题，有人提出了数字通讯的脉码调制(PCM)方案。到了 40 年代，随着科学技术的发展，由于高速度的、大量和精确的计算工作的需要，计算器械开始由机械的、电磁的、进而转向“电子数字技术”。1946 年美国科学界研制出了世界第一台电子管式计算机，开创了电子计算机的第一代。

1955 年到 1962年，在计算机科学中开始以晶体管取代电子管，进入了电子计算机的第二代。

1962 年至 1970 年，由于半导体器件的发展，开始出现了小规模集成电路(SSI)，使电子计算机转入了第三代产品，同时数字技术也开始进入了通讯、测量及自动控制等各个领域。

1970 年由于中、大规模集成电路(MSI, LSI)的出现，使电子计算机进一步缩小了体积、降低了功耗并提高了可靠性，为应用于实际生产现场创造了条件。

1971 年美国 Intel 公司研制出第一台“微处理器”4004，这是一种超大规模集成电路(GSI)，使电子计算机由第三代开始向第四代发展。超大规模集成电路的出现，使计算机趋向微型化，而且成本愈来愈低，并逐渐深入到国防、经济、生产、科研及人民生活的各个领域。

现在的趋势是把这种微处理器装入到各种各样的专用设备，使它成为设备总体的一个组成部分，这将为今后的生产和人们生活开创一个新局面。数字技术的普及，计算机的全面应用，其历史意义不亚于第一次工业革命，所以有人称它为第二次工业革命。

§3. 数字逻辑电路的分类和研究方法

1. 数字逻辑电路的分类

一个功能完备的“数字系统”是由若干个功能相对独立的“数字逻辑部件”所组成的，如电子计算机是一个数字系统，而构成计算机的运算器、控制器、移位寄存器、存贮器、译码器等等则是数字逻辑部件。数字逻辑部件又称为“数字逻辑电路”，它是由单元电路(各种逻辑门电路和触发器)所组成，或是把各单元电路集成在一个芯片上(即中、大规模集成电路)所构成。根据单元电路构成数字逻辑部件的规则不同，可将它分为两大类，即“组合逻辑电路”和“时序逻辑电路”。

如果一个数字逻辑电路的输出只和当时的输入状态有关，而和该逻辑电路的历史状态无关

则称它为“组合逻辑电路”，如半加、译器码器等都是“组合逻辑电路”如果一个数字逻辑电路的输出不只和当时的输入状态有关，而且和该逻辑电路的历史状态有关则称它为“时序逻辑电路”，如计数器、移位寄存器等都是时序逻辑电路。

在“时序逻辑电路”中根据是否受时钟脉冲控制又进一步分为“同步时序逻辑电路”和“异步时序逻辑电路”，前者受时钟脉冲控制，后者不受时钟脉冲控制。

2. 数字逻辑电路的研究方法

组成“数字逻辑电路”的单元电路通俗地说就是各种各样的“开关”，所以单元电路又称为“开关电路”。因为目前绝大部分开关电路都已集成化，所以开关电路又可以称为“开关器件”。开关器件本身虽然简单，但它却可以组成一个结构复杂功能完善的数字系统（如电子计算机），这正如砖瓦本身虽然简单，但它却可以构成一个富丽堂皇的宏伟建筑。一个伟大的建筑不是砖瓦的简单堆积，而是靠好的总体方案和结构设计；同样一个完善的“数字系统”也不是开关器件的简单连接，而是依赖于完整的“开关理论”和一套完善的逻辑设计方法，这正是“数字逻辑电路设计”这一课程所要达到的目的。

数字逻辑电路的研究可以分为“分析”和“综合”两个方面。所谓“逻辑分析”是对一个现成的数字逻辑电路，研究它的逻辑功能；而“逻辑综合”则是根据给定的逻辑功能去设计一个最佳的数字逻辑电路。这里的“最佳”是指可靠性高、成本低和其它具体条件。对于小规模集成电路最佳的标准是器件省、接线少，而对于中、大规模集成电路则是集成度高、外部引线少等等。虽然今后随着集成电路的发展最佳标准也将相应改变，但做为器件省、接线少这一经典的设计方法仍不失去做为数字逻辑电路设计的基础，所以我们在全书中仍按器件省、接线少这一经典设计方法来处理。

组合逻辑的分析和综合已经建立了一套完整的理论和形成了一套完善的处理方法。时序逻辑的理论和方法还有待进一步健全和完善。

最后还要指出一点，无论逻辑分析或综合我们都是把数字系统作为“实时系统”来处理的。所谓“实时系统”是指信息在该系统内传递和加工所需要的时间相对于整个物理系统运动过程所需要的时间可以忽略不记（开关器件的时延只有 ns 数量级，所以不难做到）。这样在处理上就和“模拟系统”一样，可以不做时间配合上的考虑，从而给数字逻辑电路的分析和综合带来很大的方便，只是在构成电路时才考虑它的过渡过程，这点将在有关章节中做专门论述。

§4. 数　　制

1. 数字体制

数字体制可分为“进位制”记数和“非进位制”记数，我们通常使用阿拉伯数码进行记数的方式就是一种“进位制”记数，如 256 的 2 代表 2×10^2，5 代表 5×10^1，6 代表 6×10^0。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。

上述这种记数方法是以十为基数，逢十进一，称为“十进制记数”。在进位制记数中除了“十

进制计数”，还有“二进制记数”、“八进制记数”、“十二进制记数”、“十六进制记数”和“六十进制记数”等等。

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I、II、III、IV、V、VI、VII、VIII、IX、X、XI、XII，这里十一不是II，十二不是III，五十是“L”而不是V X等等。这种记数法当数字再大一些就很不方便了，所以至今除了个别地方使用外已基本不用。

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：

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

其中 $0\leqslant a_i\leqslant 9$：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的形式，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：

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

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：

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

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”展开为多项式的形式则为：

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

一般可简记为：

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

特别应当指出，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

例如：$(1101.1)_2$这种并列表示可以展开为如下的多项式形式

$$(1101.1)_2 = 1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}$$

二进制记数除了它的数码少外还有其运算规则简单，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：

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

数码少可以使与它对应的电子器件状态数少，0、1两个数码就可以选用只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都具有两个状态；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			

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，为了书写上的方便也常用"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：

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

$$+a_{-2}\times(8)^{-2}+\cdots+a_{-m}\times(8)^{-m}$$

一般可写为:

$$(N)_8=\sum_{i=-m}^{n-1}a_i(8)^i$$

同样这里基数 8 也是借用十进制的数码。

如: $(357.4)_8=3\times8^2+5\times8^1+7\times8^0+4\times8^{-1}$

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:

$$(N)_{16}=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_{16}$$

多项式表示为:

$$\begin{aligned}(N)_{16}=&a_{n-1}\times(16)^{n-1}+a_{n-2}\times(16)^{n-2}+\cdots\\&+a_1\times(16)^1+a_0\times(16)^0+a_{-1}\times(16)^{-1}+\\&+a_{-2}\times(16)^{-2}+\cdots+a_{-m}\times(16)^{-m}\end{aligned}$$

一般写为:

$$(N)_{16}=\sum_{i=-m}^{n-1}a_i(16)^i$$

例如: $(\mathrm{BA07C.F})_{16}=\mathrm{B}\times(16)^4+\mathrm{A}\times(16)^3+0\times(16)^2$
$+7\times(16)^1+\mathrm{C}\times(16)^0+\mathrm{F}\times(16)^{-1}$

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:

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

多项式表示为:

$$\begin{aligned}(N)_R=&a_{n-1}(R)^{n-1}+a_{n-2}(R)^{n-2}+\cdots\\&+a_1(R)^1+a_0(R)^0+a_{-1}(R)^{-1}+\\&+a_{-2}(R)^{-2}+\cdots+a_{-m}(R)^{-m}\end{aligned}$$

一般写为:

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

§5. 各进位制记数间的转换

1. 多项式替代法

如果将二进制数 1101.1 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。例如：

$$(1101.1)_2=[1\times(10)^{11}+1\times(10)^{10}$$
$$+0\times(10)^1+1\times(10)^0+1\times(10)^{-1}]_2$$
$$=(1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1})_{10}$$
$$=(8+4+0+1+0.5)_{10}$$
$$=(13.5)_{10}$$

具体作法可以归纳为如下几个步骤：

(1) 将二进制数按权展开。

(2) 再把二进制数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	十进制	二进制	八进制	十六进制
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

(3) 按照十进制数的运算规则计算多项式的值。

反之若将十进制数 13.5 转换为二进制数，具体步骤如下：

$$(13.5)_{10}=[1\times(10)^1+3\times(10)^0+5\times(10)^{-1}]_{10}$$
$$=[1\times(1010)^1+11\times(1010)^0+101\times(1010)^{-1}]_2$$
$$=(1010+11+0.1)_2$$
$$=(1101.1)_2$$

也可归纳为：

(1) 将十进制数按权展开。

(2) 再把十进制数的各位数码(包括各位权的基数和指数) 用相对应的二进制数码替代(见表 1-2)。

(3) 按照二进制数的运算规则计算多项式的值。

上述方法可以推广为一般的形式。将 α 进制的数转换为 β 进制的数时，可按以下规则进行：

(1) 将 α 进制的数按权展开。

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

(3) 最后按照β进制的运算规则计算多项式的值。

具体表示为:

$$
\begin{aligned}
(N)_\alpha &= (a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_\alpha \\
&= [a_{n-1}(10)^{n-1}+a_{n-2}(10)^{n-2}+\cdots+a_1(10)^1 \\
&\quad +a_0(10)^0+a_{-1}(10)^{-1}+a_{-2}(10)^{-2}+\cdots+a_{-m}(10)^{-m}]_\alpha \\
&= [b_{n-1}\alpha^{n-1}+b_{n-2}\alpha^{n-2}+\cdots+b_1\alpha^1+b_0\alpha^0 \\
&\quad +b_{-1}\alpha^{-1}+b_{-2}\alpha^{-2}+\cdots+b_{-m}\alpha^{-m}]_\beta \\
&= (b_{k-1}b_{k-2}\cdots b_1b_0.b_{-1}b_{-2}\cdots b_{-l})_\beta
\end{aligned}
$$

〔例 1〕 试把八进制数 357.4 转换为十进制数。

解:

$$
\begin{aligned}
(357.4)_8 &= [3\times(10)^2+5\times(10)^1+ \\
&\quad +7\times(10)^0+4\times(10)^{-1}]_8 \\
&= (3\times8^2+5\times8^1+7\times8^0+4\times8^{-1})_{10} \\
&= (192+40+7+0.5)_{10} \\
&= (239.5)_{10}
\end{aligned}
$$

反之,如果要我们把十进制数 239.5 转换成八进制数就难以完成了,因为对八进制的运算规则我们并没有掌握。

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

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α进制的整数$(N)_\alpha$转换为β进制的整数$(M)_\beta$可采用基数除法。

β进制数$(M)_\beta$的并列表示为:

$$(M)_\beta=(b_{k-1}b_{k-2}\cdots b_1b_0)_\beta$$

按权展开后为:

$$
\begin{aligned}
(M)_\beta &= [b_{k-1}(10)^{k-1}+b_{k-2}(10)^{k-2}+\cdots \\
&\quad +b_1(10)^1+b_0(10)^0]_\beta
\end{aligned}
$$

β进制数$(M)_\beta$反复被β进制的基数去除

$$\frac{(M)_\beta}{10}=\underbrace{b_{k-1}(10)^{k-2}+\cdots+b_2(10)^1+b_1(10)^0}_{(M_1)_\beta}\cdots b_0$$

$$\frac{(M_1)_\beta}{10}=\underbrace{b_{k-1}(10)^{k-3}+\cdots+b_2(10)^0}_{(M_2)_\beta}\cdots\cdots\cdots\cdots\cdots\cdots b_1$$

$$\vdots$$

$$\frac{(M_{k-2})_\beta}{10}=\underbrace{b_{k-1}(10)^0}_{(M_{k-1})_\beta}\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots b_{k-2}$$

$$\frac{(M_{k-1})_\beta}{10}=0\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots b_{k-1}$$

可见除得的余数就是β进制数$(M)_\beta$的各位数码(由低位到高位排列)。根据这个道理我们把β进制的基数(10)转换为α进制数，相应的数码为β，再用β反复去除$(N)_\alpha$

商　　　　　　余数

$$\frac{(N)_\alpha}{\beta}=(N_1)_\alpha\cdots\cdots\cdots\cdots\cdots a'_0$$

$$\frac{(N_1)_\alpha}{\beta}=(N_2)_\alpha\cdots\cdots\cdots\cdots\cdots a'_1$$

$$\vdots$$

$$\frac{(N_{k-1})}{\beta}=0\cdots\cdots\cdots\cdots\cdots a'_{k-1}$$

所得余数也一定是β进制数$(M)_\beta$各位数码的值，只不过仍然用α进制的数码来表示。我们只要将$a'_0a'_1\cdots\cdots a'_{k-1}$代换为$\beta$进制数码$b_0b_1\cdots b_{k-1}$，则$\alpha$进制数$(N)_\alpha$便转换成为$\beta$进制数

$$(M)_\beta=(b_{k-1}b_{k-2}\cdots b_1b_0)_\beta \qquad 0\leqslant b_j\leqslant\beta-1 \quad (j=0,\ 1,\ \cdots,k-1)$$

由此得出“基数除法”的基本规则：

(1) 将α进制数$(N)_\alpha$在α进制中反复除以β，直到商为“0”。

(2) 每次除完所得余数用β进制数码来代换便得$(M)_\beta$。

〔例 2〕 用基数除法把$(712)_{10}$转换为二进制数。

二进制数基数在十进制数中是 2，即 $\beta=2$。

将$(712)_{10}$反复用 2 来除，可以采用简化除式来做：

除式	余数
2 | 712	
2 | 356	0
2 | 178	0
2 | 89	0
2 | 44	1
2 | 22	0
2 | 11	0
2 | 5	1
2 | 2	1
| 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：