



普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术系列教材

数字逻辑原理与工程设计(第2版)

刘真 杨乾明 刘芸 文梅 编著



高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划
高等学校计算机科学与技术系列教材

数字逻辑原理与工程设计

Shuzi Luoji Yuanli yu Gongcheng Sheji

(第2版)

刘真 杨乾明 刘芸 文梅 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书系统地阐述了数字逻辑电路的分析、设计方法以及数字系统的工程实现技术,主要内容包括数制与编码,布尔代数基础,逻辑门电路,组合逻辑电路的分析和设计,触发器,时序电路,数字系统设计基础,可编程逻辑器件,Verilog HDL 语言和基于 FPGA 的数字系统设计实例。

本书采用实例教学的组织形式,内容由浅入深,易于理解和掌握,可作为高等学校计算机类、电子类和自动化类等有关专业的教材和参考书,也可供有关专业工程技术人员参考。

图书在版编目(CIP)数据

数字逻辑原理与工程设计/刘真等编著. --2 版.

--北京:高等教育出版社,2013.4

ISBN 978-7-04-032635-2

I. ①数… II. ①刘… III. ①数字逻辑-高等学校-教材 IV. ①TP331.2

中国版本图书馆 CIP 数据核字(2013)第 027233 号

策划编辑 倪文慧

责任编辑 倪文慧

封面设计 于文燕

版式设计 童 丹

插图绘制 尹 莉

责任校对 刘 莉

责任印制 毛斯璐

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 三河市春园印刷有限公司
开 本 787mm × 1092mm 1/16
印 张 27.25
字 数 620 千字
购书热线 010-58581118
咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>

版 次 2003 年 11 月第 1 版
2013 年 4 月第 2 版
印 次 2013 年 4 月第 1 次印刷
定 价 38.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 32635-00

前 言

“数字逻辑”是计算机科学与技术学科本科生的专业基础课,是“计算机组成原理”的先导课程之一。由于其专业基础课的性质,往往被误认为是一门辅助或从属性质的课程,因而过分强调其专业基础理论知识的特质,忽视了其工程应用的属性。随着数字技术的快速发展和数字系统的广泛应用,为了满足人才培养对创新能力和工程实践综合能力的需求,我们在对国内外相关教材进行分析比较的基础上,结合当前数字技术发展趋势和实际工作经验,通过实例教学的方式组织编写了本书,期望将读者快速引入数字技术工程领域。

全书分为两部分,共 10 章。前一部分为第一至第六章,较详细地介绍了数字逻辑的基本原理。其中:第一、第二章介绍数字逻辑的基本理论,包括数制与编码、布尔代数的公式、定理和规则、布尔函数的基本形式和布尔函数的化简;第三、第四章介绍逻辑门电路和组合逻辑电路的分析和设计方法;第五、第六章介绍触发器和时序逻辑电路的分析和设计方法,时序电路包括同步时序电路和异步时序电路。后一部分为第七至第十章,主要通过实例教学的组织方式,着重介绍复杂数字系统工程设计的基础知识。其中:第七章介绍数字系统的设计方法与设计步骤;第八章介绍可编程逻辑器件的结构原理及其在数字逻辑电路设计中的应用,重点介绍了 Altera、Xilinx 和 Actel 公司的主流可编程逻辑器件 FPGA/CPLD 的结构特点与工作原理;第九章通过实例逐步引入对 Verilog HDL 语言的介绍,并就测试环境设计的一般方法以及有关 Verilog HDL 设计的一些高级话题进行了讨论;第十章通过实例介绍如何基于 FPGA 完成数字系统的设计。

本书第一、第二和第四章由文梅编写,第三、第五章由刘芸编写,第六章由刘真编写,第七至第十章由杨乾明在陆洪毅老师原来编写的基础上进行了重编,全书由刘真统稿。

与第一版比较,本次修订主要有三个大的更改。一是考虑很多学校数字逻辑与数字电路是一门课程,有的学校甚至没有模拟电路课程,为了知识的相对完整性,增加了第三章“逻辑门电路”和第五章“触发器”;二是对第二部分内容进行了重组,并将第一版第一部分的“简单可编程逻辑器件及其应用”归并到了第二部分,使结构更清晰合理;三是对第二部分内容进行了扩充,去除了部分比较旧的知识点,以较新的器件产品替代了早期器件产品,增加了对典型数字系统辅助设计工具的介绍,最后以实例说明如何利用现代 EDA 工具(针对 FPGA)来设计一个完整的数字系统。

书中给出了大量例题,以便于读者对理论知识的理解;对于个别内容较专的章节标注了星号,以供不同需求的读者选用;同时提供第一部分各章的部分习题答案,以便读者自学参考使用。本书适合作为高等院校计算机类、电子类和自动化类等有关专业的教材,也可供有关专业工程技术人员参考。建议讲授 60 学时左右,并配合适当的实验学时。

本书是编者在多年教学经验基础上一次新的尝试,由于水平和经验有限,疏误之处在所难免,恳请读者和专家批评指正。

编 者
2012 年 12 月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010)58581897 58582371 58581879

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第一章 数制与编码	1	2.3.1 函数的“积之和”与“和之积”表示形式	45
引言	1	2.3.2 函数的“标准积之和”与“标准和之积”形式	46
1.1 进位计数制与数制转换	1	2.4 不完全确定的布尔函数	50
1.1.1 进位计数制及其表示	1	2.5 布尔函数的化简	51
1.1.2 数制转换	4	2.5.1 代数化简法	52
1.2 带符号二进制数的代码表示	12	2.5.2 卡诺图化简法	54
1.2.1 原码	12	*2.5.3 列表化简法	62
1.2.2 反码	14	小结	71
1.2.3 补码	15	习题 2	71
1.2.4 原码、反码和补码之间的转换	17		
1.2.5 溢出的判断和变形码	19	第三章 逻辑门电路	76
1.3 其他常用编码	21	引言	76
1.3.1 十进制数的二进制编码	21	3.1 半导体器件的开关特性	76
1.3.2 字符代码	23	3.1.1 二极管的开关特性	76
1.3.3 可靠性编码	25	3.1.2 三极管的开关特性	79
小结	33	3.1.3 MOS 管的开关特性	81
习题 1	34	3.2 基本逻辑门电路	83
		3.2.1 “与”门电路	83
第二章 布尔代数基础	36	3.2.2 “或”门电路	83
引言	36	3.2.3 “非”门电路	84
2.1 布尔代数的基本概念	36	3.3 TTL 门电路	87
2.1.1 布尔变量及其基本运算	36	3.3.1 TTL“与非”门	87
2.1.2 布尔函数及其表示方法	38	3.3.2 集电极开路门和三态门	94
2.1.3 布尔函数的“相等”概念	39	3.3.3 其他逻辑功能的 TTL 门电路	98
2.2 布尔代数的公式、定理和规则	40	3.4 CMOS 门电路	101
2.2.1 布尔代数的基本公式	40	3.4.1 CMOS 反相器	101
2.2.2 布尔代数的主要定理	42	3.4.2 其他逻辑功能的 CMOS 门	
2.2.3 布尔代数的重要规则	43		
2.3 布尔函数的基本形式	45		

电路	103	小结	151
*3.5 集成门电路的使用	105	习题 4	151
小结	106		
习题 3	106		
第四章 组合逻辑电路的分析和设计	110	第五章 触发器	154
引言	110	引言	154
4.1 常用逻辑门的图形符号	110	5.1 基本 RS 触发器	154
4.2 布尔函数的实现	111	5.1.1 电路结构和工作原理	154
4.2.1 用“与非”门实现布尔		5.1.2 逻辑功能描述方法	155
函数	112	5.2 同步 RS 触发器	157
4.2.2 用“或非”门实现布尔		5.2.1 电路结构和工作原理	157
函数	112	5.2.2 同步触发器的动作特点	158
4.2.3 用“与或非”门实现布尔		5.3 主从触发器	159
函数	113	5.3.1 主从 JK 触发器	159
4.3 组合电路的分析	114	5.3.2 主从触发器的动作特点	161
4.4 组合电路的设计	116	5.4 边沿触发器	162
4.5 常用组合电路	120	5.4.1 边沿 JK 触发器	162
4.5.1 加法器	120	5.4.2 边沿 D 触发器	166
4.5.2 十进制数字的 7 段显示	128	5.4.3 边沿 T 触发器	168
4.5.3 代码转换电路	132	*5.5 触发器的脉冲工作特性	169
4.5.4 二进制比较器	134	小结	171
4.5.5 阵列乘法器	137	习题 5	171
4.6 二进制译码器	138		
4.6.1 二进制译码器的功能和		第六章 时序电路	175
组成	138	引言	175
4.6.2 用中规模集成译码器进行		6.1 时序电路与时序机	175
设计	139	6.1.1 时序电路的结构和特点	176
4.7 多路选择器	141	6.1.2 时序机的定义	177
4.7.1 多路选择器的逻辑功能和		6.1.3 时序机的状态表和状态图	177
组成	141	6.1.4 完全定义机和不完全定	
4.7.2 用多路选择器进行逻辑		义机	179
设计	142	6.2 同步时序电路的分析与设计	180
4.8 多路分配器	146	6.2.1 建立原始状态表	181
4.9 组合电路中的险态	148	6.2.2 状态表的化简	183
		6.2.3 状态分配	194
		6.2.4 确定激励函数和输出函数	197

6.2.5 分析与设计举例	200	设计	262
6.3 常用的同步时序电路	207	7.3.2 基于集成电路的设计	265
6.3.1 寄存器	207	7.3.3 基于可编程逻辑器件的 设计	267
6.3.2 计数器	208	7.3.4 基于 ASIC 的设计	268
6.3.3 节拍信号发生器	212	7.4 数字系统辅助设计工具	268
6.4 脉冲异步时序电路的分析与 设计	216	7.4.1 印制电路板设计工具	269
6.4.1 异步时序电路的结构特点 和分类	216	7.4.2 ASIC 设计工具	271
6.4.2 脉冲异步时序电路的分析 与设计	217	7.4.3 FPGA 设计工具	272
6.5 电平异步时序电路的分析与 设计	221	7.5 数字系统的发展趋势	273
6.5.1 电平异步时序电路的描述 方法	221	小结	274
6.5.2 建立原始流程表	222	习题 7	274
6.5.3 流程表的简化	225	第八章 可编程逻辑器件	275
6.5.4 流程表的状态分配	226	引言	275
*6.5.5 电平异步时序电路的险态 ..	231	8.1 概述	275
6.5.6 分析与设计举例	232	8.1.1 可编程逻辑器件的发展 历史	275
小结	238	8.1.2 可编程逻辑器件的原理	276
习题 6	239	8.1.3 可编程逻辑器件的分类及 比较	278
第七章 数字系统设计基础	245	8.1.4 PLD 的逻辑表示法	279
引言	245	8.2 可编程存储器	280
7.1 概述	245	8.2.1 可编程只读存储器	280
7.1.1 数字系统的定义和特点	245	8.2.2 可擦除的可编程只读存 储器	282
7.1.2 数字系统的历史	247	8.2.3 采用 PROM 的组合逻辑 设计	283
7.1.3 现代数字系统的一般结构 ..	248	8.3 复杂可编程逻辑器件	284
7.2 数字系统的设计过程	249	8.3.1 概述	284
7.2.1 需求分析	249	8.3.2 Altera 的 CPLD	285
7.2.2 系统设计和实现	251	8.3.3 Xilinx 的 CPLD	290
7.2.3 系统测试	254	8.4 现场可编程门阵列	296
7.3 数字系统设计	262	8.4.1 概述	296
7.3.1 基于通用微处理器/DSP 的		8.4.2 Altera 的 FPGA	297

8.4.3 Xilinx 的 FPGA	304	9.8.2 多时钟域	361
8.4.4 Actel 的 FPGA	310	9.8.3 同步复位和异步复位	363
小结	317	9.8.4 可配置电路设计	365
习题 8	317	9.8.5 Verilog HDL 语言的设计 风格	367
第九章 Verilog HDL 语言	318	小结	369
引言	318	习题 9	369
9.1 概述	318	第十章 基于 FPGA 的数字系统设计 实例	370
9.2 Verilog HDL 程序基本结构	319	引言	370
9.3 信号	322	10.1 设计任务描述	370
9.4 模块体的描述	325	10.2 需求分析	371
9.5 行为级建模	335	10.3 系统设计与实现	373
9.5.1 数据类型	335	10.3.1 算法设计	373
9.5.2 运算符及表达式	338	10.3.2 系统实现	374
9.5.3 结构说明语句	340	10.4 系统仿真与测试	400
9.5.4 复杂语句	342	10.5 综合与布局布线	404
9.5.5 时序控制	346	10.6 板上 FPGA 验证	407
9.5.6 阻塞赋值和非阻塞赋值	348	小结	408
9.5.7 任务与函数	351	习题 10	409
9.5.8 Verilog HDL 预编译指令	354	部分习题参考答案	410
9.6 可综合性设计	355	参考文献	425
9.7 测试环境设计	357		
9.8 Verilog HDL 设计高级专题	358		
9.8.1 寄存器	359		

第一章

数制与编码

引言

本章主要讨论数字系统中数的基本表示方法。首先讨论不同的进位计数制及其相互间如何转换;然后讨论二进制数在计算机中的表示方法,包括数的符号、数值以及小数点等如何表示。另外,本章还介绍计算机中常用的几种编码,包括十进制数的常用编码、字符编码以及可靠性编码等。

1.1 进位计数制与数制转换

1.1.1 进位计数制及其表示

所谓进位计数制,就是按进位方式实现计数的一种规则,简称进位制。在日常生活中常见这种按进位制计数的情况,如十进制、十二进制、六十进制等。

对于任何一个数,都可以用不同的进位制来表示。下面先从熟悉的十进制开始,分析各种进位制的特点和表示方法。

1. 十进制

十进制有10个数字符号,即0,1,2,⋯,9。将若干个这样的数字符号并列在一起可以表示一个十进制数,每位不超过“9”,由低位向高位进位的原则是“逢十进一”。这是十进制的特点。

这里要引用两个术语:

(1) “基数”,它表示某种进位制所具有的数字符号的个数,如十进制的基数为“10”;

(2) “位权”或“权”,它表示某种进位制的数处于不同位置时所表示的单位数值,如十进制数135.79,百位(1代表100)上的权为 10^2 ;十位(3代表30)上的权为 10^1 ;个位(5代表5)上的权为 10^0 ;十分位(7代表7/10)上的权为 10^{-1} ;百分位(9代表9/100)上的权为 10^{-2} 。

基数和权是进位制的两个要素,根据基数和权的概念,可以将任何一个数表示成多项式的形式。例如:

$$135.79 = 1 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 7 \times 10^{-1} + 9 \times 10^{-2}$$

对于一个一般的十进制数 N ,它可以表示成

$$(N)_{10} = (d_{n-1}d_{n-2}\cdots d_1d_0 \cdot d_{-1}d_{-2}\cdots d_{-m})_{10} \quad (1.1)$$

或

$$\begin{aligned} (N)_{10} &= d_{n-1}(10)^{n-1} + d_{n-2}(10)^{n-2} + \cdots + d_1(10)^1 + d_0(10)^0 + d_{-1}(10)^{-1} + \\ &\quad d_{-2}(10)^{-2} + \cdots + d_{-m}(10)^{-m} = \sum_{i=-m}^{n-1} d_i (10)^i \end{aligned} \quad (1.2)$$

其中, n 表示整数部分的位数; m 表示小数部分的位数; 10 表示基数, $(10)^i$ 为第 i 位的权; d_i 表示各个数字符号, 在十进制中有

$$d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

通常称式(1.1)为并列表示法, 式(1.2)为多项式表示法或按权展开式。

在数字系统中使用的进位制并不限于十进制。广义地, 一个 R 进制的数 N 可以表示成

$$\begin{aligned} (N)_R &= (r_{n-1}r_{n-2}\cdots r_1r_0 \cdot r_{-1}r_{-2}\cdots r_{-m})_R \\ &= r_{n-1}R^{n-1} + r_{n-2}R^{n-2} + \cdots + r_1R^1 + r_0R^0 + r_{-1}R^{-1} + r_{-2}R^{-2} + \cdots + r_{-m}R^{-m} \\ &= \sum_{i=-m}^{n-1} r_i R^i \end{aligned}$$

其中, n 表示整数的位数, m 表示小数的位数; R 为基数, 在十进制中 R 应写成“10”; r_i 是 R 进制中各个数字符号, 即有

$$r_i \in \{0, 1, 2, \cdots, R-1\}$$

2. 二进制

数制是人类在实践中创造的。对于一个数, 原则上讲可以用任何一种进位制来计数或进行算术运算。但是, 不同的进位制的运算方法及难易程度各不相同。因此, 选择什么样的进位制来表示数, 对数字系统的性能影响很大。在数字系统中, 常用二进制来表示数和进行运算。这是因为二进制只有 0 和 1 两个数字符号, 容易用物理状态来表示; 二进制运算规则简单, 便于进行算术运算; 此外, 采用二进制来表示数可以节省设备, 其运算逻辑电路的设计也比较方便。

二进制算术运算十分简单, 规则如下:

加法规则 $0+0=0, 0+1=1+0=1, 1+1=10$

乘法规则 $0 \times 0=0, 0 \times 1=1 \times 0=0, 1 \times 1=1$

下面举几个二进制数四则运算的例子, 熟悉它的运算规则。

例 1 两个二进制数相加, 采用“逢二进一”的法则。

$$\begin{array}{r} 1101 \\ +) 1001 \\ \hline 10110 \end{array}$$

例 2 两个二进制数相减, 采用“借一当二”的法则。

$$\begin{array}{r}
 1101 \\
 -) 0110 \\
 \hline
 0111
 \end{array}$$

例 3 两个二进制数相乘,其方法与十进制乘法运算相似,但采用二进制运算规则。

$$\begin{array}{r}
 1011 \\
 \times) 1101 \\
 \hline
 1011 \\
 0000 \\
 1011 \\
 1011 \\
 \hline
 10001111
 \end{array}$$

例 4 两个二进制数相除,其方法与十进制除法运算相似,但采用二进制运算规则。

$$\begin{array}{r}
 1010 \quad \cdots \text{商} \\
 1101 \overline{)10001001} \\
 \underline{1101} \\
 10000 \\
 \underline{1101} \\
 111 \quad \cdots \text{余数}
 \end{array}$$

3. 八进制和十六进制

虽然数字系统广泛采用二进制,但当二进制数的位数很多时,书写和阅读很不方便,容易出错。为此,人们通常采用二进制的缩写形式——八进制和十六进制。

八进制的基数 $R=8$,每位可取 8 个不同的数字符号(即 0,1,2,⋯,7),其进位规则是“逢八进一”。

十六进制的基数 $R=16$,每位可取 16 个不同的数字符号(即 0,1,2,⋯,9,A,B,C,D,E,F),其进位规则是“逢十六进一”。

表 1.1 列出了当基数 R 为 10、2、8 和 16 时,数值 0 到 20 的不同进位制数。

表 1.1 不同基数的进位制数

$R=10$	$R=2$	$R=8$	$R=16$
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3

续表

$R = 10$	$R = 2$	$R = 8$	$R = 16$
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

1.1.2 数制转换

在计算机和其他数字系统中普遍采用二进制,采用二进制的数字系统只能处理二进制数或用二进制编码形式表示的其他进位制数,而信息本身可能是其他进位制数,因此需要进行不同进位制数之间的转换。例如,人们习惯于使用十进制数,所以在用计算机进行信息处理时,首先必须把十进制数转换成二进制数才能被计算机所接受,然后进行运算,运算结果又必须从二进制转换成人们习惯的十进制数。

本节介绍不同进位制之间相互转换的方法。

1. 直接转换法

由于一位八进制的 8 个数字符号正好相应于三位二进制数的 8 种不同组合,所以八进制与二进制之间有简单的对应关系:

八进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

这样,八进制数与二进制数之间的转换就极为方便。

例 5 将二进制数 11010.1101 转换为八进制数。

$$\begin{array}{cccc} \underbrace{011}_3 & \underbrace{010}_2 & . & \underbrace{110}_6 \underbrace{100}_4 \end{array}$$

所以 $(11010.1101)_2 = (32.64)_8$ 。

由二进制转换成八进制的方法是:以小数点为界,将二进制数的整数部分从低位开始,小数部分从高位开始,每三位分成一组、头尾不足三位的补 0;然后将每组的三位二进制数转换为一位八进制数。

例 6 将八进制数 357.6 转换为二进制数。

$$\begin{array}{cccc} 3 & 5 & 7 & . & 6 \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ 011 & 101 & 111 & . & 110 \end{array}$$

所以 $(357.6)_8 = (11101111.11)_2$ 。

同理,由于一位十六进制的 16 个数字符号正好相应于 4 位二进制数的 16 种不同的组合,所以,十六进制与二进制之间有简单的对应关系:

十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

这样,十六进制数与二进制数之间的转换也很方便。

例 7 将二进制数 1010110110.110111 转换为十六进制数。

$$\begin{array}{cccccc} \underbrace{0010}_2 & \underbrace{1011}_B & \underbrace{0110}_6 & . & \underbrace{1101}_D & \underbrace{1100}_C \end{array}$$

所以 $(1010110110.110111)_2 = (2B6.DC)_{16}$ 。

例 8 将十六进制数 5D.6E 转换为二进制数。

$$\begin{array}{cccc} 5 & D & . & 6 & E \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 0101 & 1101 & . & 0110 & 1110 \end{array}$$

所以 $(5D.6E)_{16} = (1011101.0110111)_2$ 。

由此可见,采用八进制和十六进制要比用二进制书写简短,易读易记,而且转换也很方便,因此,计算机工作者普遍采用八进制或十六进制来书写和表达。

2. 多项式替代法

先来看一个简单例子。

例 9 将二进制数 1101.101 转换成十进制数。

先把二进制数的并列表示法展开成多项式表示法,则有

$$(1101.101)_2 = [1 \times (10)^{11} + 1 \times (10)^{10} + 0 \times (10)^1 + 1 \times (10)^0 + 1 \times (10)^{-1} + 0 \times (10)^{-10} + 1 \times (10)^{-11}]_2$$

再把等式右边的二进制数替代成十进制数,则得

$$(1101.101)_2 = [1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}]_{10}$$

在十进制中计算等式右边之值,得

$$(1101.101)_2 = (8+4+1+0.5+0.125)_{10} = (13.625)_{10}$$

这一方法可以推广到任意两个 α 、 β 进制数之间的转换,其方法是:先将 α 进制的数在 α 进制中按权展开,然后替代成相应 β 进制中的数,最后在 β 进制中计算即可得 β 进制的数。

例 10 $(123.4)_8 = (?)_{10}$

$$\begin{aligned} (123.4)_8 &= [1 \times (10)^2 + 2 \times (10)^1 + 3 \times (10)^0 + 4 \times (10)^{-1}]_8 \text{ (展开)} \\ &= (1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1})_{10} \quad \text{(替代)} \\ &= (64 + 16 + 3 + 0.5)_{10} \text{ (在十进制中计算)} \\ &= (83.5)_{10} \end{aligned}$$

例 11 $(201.2)_3 = (?)_2$

$$\begin{aligned} (201.2)_3 &= [2 \times (10)^2 + 0 \times (10)^1 + 1 \times (10)^0 + 2 \times (10)^{-1}]_3 \text{ (展开)} \\ &= (10 \times 11^{10} + 0 \times 11^1 + 1 \times 11^0 + 10 \times 11^{-1})_2 \quad \text{(替代)} \\ &= (10010 + 1 + 0.101010 \cdots)_2 \text{ (在二进制中计算)} \\ &= (10011.101010 \cdots)_2 \end{aligned}$$

由以上两例可看出,多项式替代法由于要在 β 进制中进行计算,当它为十进制时,计算较方便,而当它为其他进制时,计算就很不方便。因此,这种方法适用于 α 进制向十进制的转换。

3. 基数乘/除法

基数乘/除法分为基数乘法和基数除法两种。对于整数的转换,采用基数除法;对于小数的转换,采用基数乘法。下面分别介绍这两种方法。

(1) 基数除法

例 12 将十进制整数 25 转换为二进制数,即

$$(25)_{10} = (?)_2$$

我们来推导转换的方法。设转换结果为

$$\begin{aligned} (25)_{10} &= (k_{n-1}k_{n-2} \cdots k_1k_0)_2 \\ &= (k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_12^1 + k_02^0)_{10} \end{aligned} \quad (1.3)$$

在十进制中计算,将式(1.3)两边除以 2,则得

$$12 + \frac{1}{2} = (k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_12^0) + \frac{k_0}{2}$$

两数相等,则它们整数部分和小数部分必定分别相等,故有

$$12 = k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_12^0 \quad (1.4)$$

$$\frac{1}{2} = \frac{k_0}{2}, \quad k_0 = 1$$

将式(1.4)两边同除以2,可得

$$6 = (k_{n-1}2^{n-3} + k_{n-2}2^{n-4} + \cdots + k_22^0) + \frac{k_1}{2}$$

故有

$$6 = k_{n-1}2^{n-3} + k_{n-2}2^{n-4} + \cdots + k_22^0$$

$$0 = \frac{k_1}{2}, \quad k_1 = 0$$

可见,所要求的二进制数 $(k_{n-1}k_{n-2}\cdots k_1k_0)_2$ 的最低位 k_0 是十进制数25除以2所得余数;次低位 k_1 是所得商12再除以2所得的余数;以此类推,继续用2除,直到商为0为止,于是,各次所得的余数即为要求的二进制数 $k_0 \sim k_{n-1}$ 之值。此法又称除2取余法。

可以将上述过程写成简单算式如下:

2	25	余数	↑ 低位
2	12	$1 = k_0$	
2	6	$0 = k_1$	
2	3	$0 = k_2$	
2	1	$1 = k_3$	
	0	$1 = k_4$	↑ 高位

所以,转换结果为 $(25)_{10} = (11001)_2$ 。

上述将十进制整数转换为二进制整数的方法可以推广到任何两个 α 、 β 进制数之间的转换。其方法是:先将 α 进制的整数在 α 进制中连续除以 β ,求得各次余数 $(k_i)_\alpha$;然后将各余数替代成 β 进制中相应的数字符号 $(k'_i)_\beta$ 。最后按照并列表示法列出即得 β 进制的整数。应该指出,由于要在 α 进制中运算,而人们对十进制运算非常熟悉,所以基数除法适用于十进制向 β 进制的转换。

例 13 $(785)_{10} = (?)_8$

8	785	余数	↑ 低位
8	98	1	
8	12	2	
8	1	4	
	0	1	↑ 高位