

高等院校信息技术规划教材

数字逻辑基础 与Verilog硬件描述语言

贾熹滨 王秀娟 魏坚华 编著
彭建朝 主审

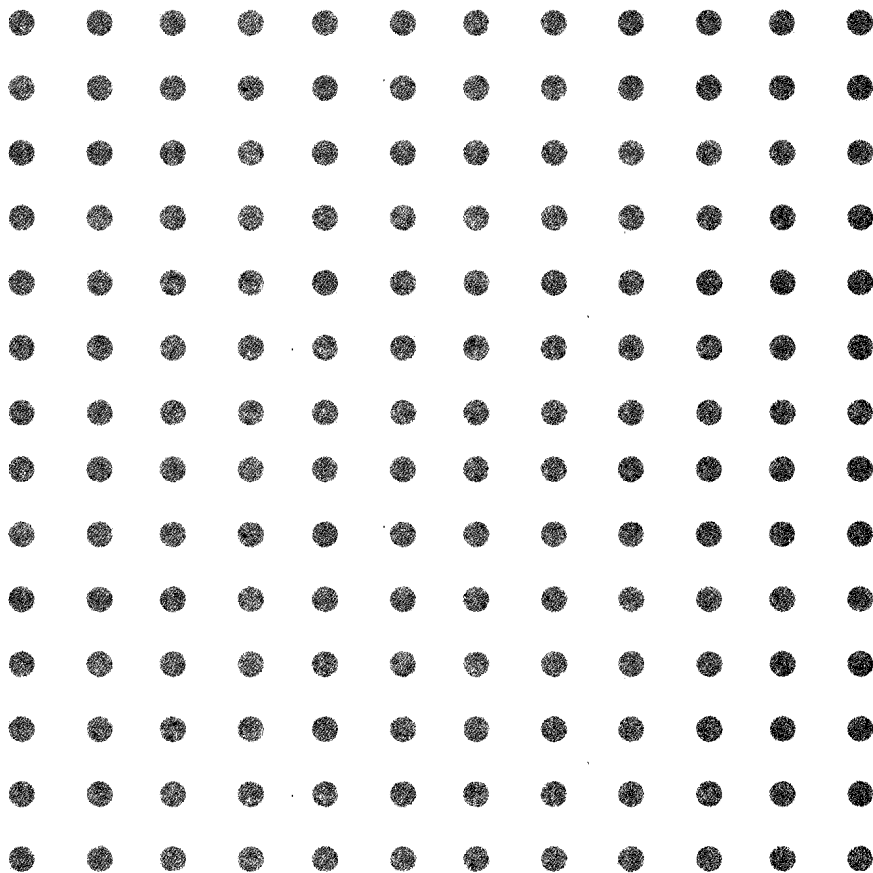


清华大学出版社

高等院校信息技术规划教材

数字逻辑基础 与Verilog硬件描述语言

贾熹滨 王秀娟 魏坚华 编著



清华大学出版社
北京

内 容 简 介

本书在介绍数字逻辑基本概念和知识基础上,系统介绍逻辑电路的分析和设计方法,特别结合现代数字系统设计技术的发展,介绍基于硬件描述语言 Verilog HDL 的逻辑电路建模方法,并给出了所举实例代码及仿真结果。

全书内容分为3部分:第1~3章介绍数字逻辑的理论基础,包括数制、码制、逻辑代数基础以及硬件描述语言基础等;第4章介绍组合电路的分析方法、常用逻辑功能电路的 Verilog HDL 建模方法以及典型功能模块的应用;第5~8章在分析锁存器/触发器工作原理和逻辑特性基础上,介绍同步时序电路的分析方法,分别讨论了典型和一般同步时序电路的 Verilog HDL 建模方法,并介绍了典型同步时序模块的应用方法。

本书可作为计算机、物联网、自动控制、电子信息等专业的本科生教材,也可作为数字系统设计相关技术人员学习 Verilog HDL 建模方法的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数字逻辑基础与 Verilog 硬件描述语言/贾熹滨等编著. —北京:清华大学出版社,2012.8

(高等院校信息技术规划教材)

ISBN 978-7-302-29097-1

I. ①数… II. ①贾… III. ①数字逻辑—高等学校—教材 ②硬件描述语言—程序设计—高等学校—教材 IV. ①TP302.2 ②TP312

中国版本图书馆 CIP 数据核字(2012)第 130756 号

责任编辑:焦虹 徐跃进

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.75

字 数:435千字

版 次:2012年8月第1版

印 次:2012年8月第1次印刷

印 数:1~3000

定 价:31.00元

产品编号:039340-01

前言

foreword

数字逻辑是信息、电子等学科重要的基础课程。作为一门经典课程,该课程有着坚实的理论和实践基础,同时随着现代数字技术的发展,该课程又增添了许多新的内容。本书旨在介绍经典理论和方法基础上,介绍面向现代主流的基于硬件描述语言数字电路设计方法,并选用 Verilog HDL 为硬件描述语言。作为被 IEEE 采纳的标准语言之一,Verilog HDL 相对 VHDL 语言具有简洁、高效、易学易用的特点,有助于学生将精力放在数字电路的建模方法,而不是语言学习上,在掌握基于硬件描述语言设计方法基础上,可进一步学习其他设计语言,根据工程需要完成数字系统的设计。

本书在数字逻辑经典方法理论介绍基础上,进一步加强了基于硬件描述语言的电路设计的部分,精简了基于特定功能集成电路器件的“搭积木”式的传统设计方法,同时配合“数字逻辑”精品课程建设,在内容安排上加大具有工程意义的实例介绍,进一步培养学生的工程意识和素质,为学生从事计算机硬件工程任务奠定良好的基础,也为后续数字系统设计、计算机组成原理、微机原理及嵌入式工程方法等硬件课程打下坚实的基础。

本书的内容安排如下:

第 1 章 介绍数字系统中对信息的表示方法,重点阐述进制与码制、带符号数的表示方法,即原码、反码和补码;介绍几种常用的编码方法。

第 2 章 介绍数字逻辑的数学基础,以举重裁判的裁决过程为线索,阐述逻辑代数与逻辑电路之间的关系,逻辑代数的基本概念、基本定理和规则,逻辑函数的基本表达形式以及逻辑函数的卡诺图化简法等。

第 3 章 介绍硬件描述语言的基础,以 Verilog HDL 为硬件描述语言。介绍它的模块结构、语法特点和 3 种建模方法等。

第 4 章 介绍基于逻辑门、典型组合电路的电路分析方法以及典型组合逻辑电路的设计,包括加法器、译码器、编码器、数据选择器、数据分配器、比较器等;重点阐述 Verilog HDL 对组合电路的建

模方法;讨论组合电路中的竞争险象问题。

第5章 介绍时序电路的双稳态元件——锁存器与触发器。从问题需求角度出发,引出具有反馈结构的基本RS锁存器,简单介绍以RS锁存器为基础D锁存器/触发器、JK锁存器/触发器的内部结构,重点探讨边沿触发器的外部逻辑功能以及Verilog HDL模型、锁存器与触发器的区别。

第6章 在时序逻辑概要基础上,重点讲述同步时序电路的分析。从时序电路组成结构和特点出发,分析描述时序逻辑的逻辑函数类型,介绍不同的时序逻辑描述方法。给出同步时序电路的分析方法,并对基于触发器的同步时序电路进行实例分析,同时讨论时序电路中“挂起”现象,说明该现象对电路的影响。

第7章 介绍计数器、寄存器、移位寄存器、移位型计数器、节拍分配器和序列信号发生器等典型同步时序电路的功能,重点探讨基于状态转移图(STG)、行为描述等功能描述基础上的Verilog HDL建模方法,探讨以典型功能单位为核心模块的应用。

第8章 介绍一般同步时序电路的设计方法,重点阐述原始状态图的建立、状态化简以及状态分配,并给出几个完整的设计实例。

每章的最后都有一定数量的习题,以便加深对基本知识、基本理论、基本分析方法和基于Verilog HDL设计方法的理解,有些习题具有一定难度,为学生提供了不同层次的训练。附录提供了Quartus下载地址信息,基于Quartus进行实例分析、设计、仿真的详细说明,供学生参考并鼓励利用Quartus平台完成相关习题的设计与仿真。

数字逻辑课程近年来得到了北京工业大学各级领导的广泛支持,2009年入选校级精品课程,推动了教育教学的稳步进行,不但为课程组创造了充分的研究、实验条件,而且在实验中心建立了先进的EDA实验室,开设了独立的数字逻辑课程实验。系统结构系的诸位教师亲自组织、指导数字逻辑课程建设的各个环节,多次修订和完善大纲、优化教学内容、丰富了教学课件等。

本书的第2、5、6、7章由贾熹滨编写,第1、4、8章由王秀娟编写,第3章和附录由魏坚华编写。全书由彭建朝老师主审。在本书编写过程中,得到了课程组游周密、孙丽君等各位老师的大力支持,他们的教学实践与经验为作者提供了极大的帮助。在此一并表示衷心的感谢。

限于作者的水平与经验,书中疏漏之处敬请广大读者批评指正。

作 者

2012年6月于北京工业大学

目录

Contents

第 1 章 信息表示	1
1.1 数制	1
1.1.1 基本概念	1
1.1.2 常用数制的表示	2
1.2 不同数制间的转换	4
1.2.1 其他进制数转换为十进制数	4
1.2.2 十进制数转换为其他进制数	4
1.2.3 二、八、十六进制数间的转换	6
1.3 带符号二进制数的表示	8
1.3.1 真值与机器数	8
1.3.2 定点数与浮点数	8
1.3.3 原码	9
1.3.4 反码	11
1.3.5 补码	12
1.3.6 真值、原码、反码、补码之间的关系	15
1.4 编码	17
1.4.1 数值数据编码	17
1.4.2 非数值数据编码	23
本章小结	25
思考题 1	25
习题 1	26
第 2 章 逻辑代数基础	28
2.1 概述	28
2.2 逻辑代数中的基本概念	30
2.3 逻辑代数的基本运算	34
2.3.1 与运算	34

2.3.2	或运算	35
2.3.3	非运算	36
2.4	逻辑代数的基本定理及规则	37
2.4.1	逻辑代数的基本公理	37
2.4.2	逻辑代数的基本定理	38
2.4.3	逻辑代数的 3 个基本规则	39
2.5	逻辑函数的性质	43
2.5.1	复合逻辑	43
2.5.2	逻辑函数的基本表达式	47
2.5.3	逻辑函数的标准表达式	48
2.6	逻辑函数的化简	55
2.6.1	逻辑函数的代数化简法	56
2.6.2	逻辑函数的卡诺图化简法	58
2.6.3	具有无关项的逻辑函数及其化简	69
	本章小结	71
	思考题 2	73
	习题 2	73
第 3 章	硬件描述语言 (Verilog HDL) 基础	77
3.1	概述	77
3.1.1	发展历程	77
3.1.2	Verilog HDL 的特点	78
3.1.3	Verilog HDL 模块化设计理念	79
3.2	Verilog HDL 基础知识	79
3.2.1	Verilog HDL 模块结构	79
3.2.2	Verilog HDL 中的词法表示	84
3.2.3	Verilog HDL 的数据类型	85
3.2.4	Verilog HDL 的运算符	88
3.3	Verilog HDL 模块的 3 种建模方式	93
3.3.1	Verilog HDL 模块的结构描述方式	94
3.3.2	Verilog HDL 模块的数据流描述方式	98
3.3.3	Verilog HDL 模块的行为描述方式	100
	本章小结	110
	思考题 3	110
	习题 3	111

第 4 章 组合电路的逻辑分析与设计	113
4.1 概述	113
4.2 组合电路的逻辑分析	119
4.3 组合电路的设计	123
4.4 典型组合逻辑电路	127
4.4.1 编码器	128
4.4.2 译码器	132
4.4.3 数据分配器	142
4.4.4 数据选择器	144
4.4.5 三态缓冲器	150
4.4.6 数值比较电路	152
4.4.7 加法器	155
4.4.8 奇偶校验电路	158
4.5 组合电路中的竞争与险象	160
4.5.1 竞争与险象	161
4.5.2 险象的分类	162
4.5.3 逻辑险象的判断	164
4.5.4 逻辑险象的消除	165
本章小结	166
思考题 4	167
习题 4	167
第 5 章 锁存器与触发器	172
5.1 概述	172
5.2 基本 R-S 锁存器	173
5.3 D 锁存器及 D 触发器	175
5.3.1 D 锁存器	175
5.3.2 正边沿 D 触发器	176
5.3.3 D 触发器的 Verilog HDL 模型	177
5.4 J-K 锁存器及触发器	179
5.4.1 J-K 锁存器	179
5.4.2 负边沿 J-K 触发器	180
5.4.3 J-K 触发器的 Verilog HDL 模型	181
5.5 T 触发器和 T' 触发器	183
5.6 锁存器和触发器的区别	184
5.7 不同类型触发器之间的转换	185

本章小结	185
思考题 5	186
习题 5	186
第 6 章 时序电路概要和同步时序电路分析	188
6.1 概述	188
6.1.1 时序电路的基本结构	189
6.1.2 时序电路的逻辑函数表达式	189
6.1.3 时序电路的分类	190
6.1.4 时序电路的描述方法	190
6.2 同步时序电路的分析方法与步骤	193
6.3 同步时序电路分析举例	194
6.4 同步时序电路中的“挂起”现象	199
本章小结	201
思考题 6	201
习题 6	202
第 7 章 典型同步时序电路的设计与应用	204
7.1 概述	204
7.2 计数器	205
7.2.1 基于触发器的二进制同步计数器设计	205
7.2.2 同步二进制计数器的 Verilog HDL 描述	208
7.2.3 多种编码十进制计数器的 Verilog HDL 参数化设计模型	211
7.2.4 多功能 4 位二进制加法计数器模块及应用电路分析	215
7.2.5 任意模数加 1 计数器的 Verilog HDL 参数化设计模型	222
7.3 寄存器及其 Verilog HDL 模型	224
7.4 移位寄存器	226
7.4.1 串入-串出结构的移位寄存器	226
7.4.2 串入-并出结构的移位寄存器	227
7.4.3 并入-串出结构的移位寄存器	228
7.4.4 多功能移位寄存器	229
7.5 移位寄存器型计数器	232
7.5.1 环形计数器	232
7.5.2 扭环形计数器	237
7.5.3 最大长度移位型计数器	240
7.6 节拍分配器	240
7.7 序列信号发生器	242

本章小结	244
思考题 7	244
习题 7	245
第 8 章 一般同步时序电路的设计	248
8.1 原始状态图(表)的建立	249
8.2 状态化简	252
8.3 状态分配	257
8.4 一般同步时序电路设计举例	258
8.5 Verilog HDL 综合设计举例	263
本章小结	271
思考题 8	271
习题 8	272
附录 A 基于 Quartus 环境和 Verilog HDL 的电路设计与仿真实例	275
参考文献	290

信息表示

【本章内容】 本章首先介绍数值数据在数字系统中的表示,包括现实世界中常用的几种进位记数制及其相互转换;带符号二进制数在数字系统中的定点表示方法(原码、反码和补码);最后介绍数字系统中常用的几种编码方案。

1.1 数 制

1.1.1 基本概念

数据信息是计算机加工和处理的对象,它包括两种数据信息,一种称为数值数据,用于表示数量的多少,可以带有符号位以表示数值正负;另一种称为字符数据,或者非数值数据,包括图像、视频、英文字母、汉字等。在计算机中实现对数据信息的操作,首先要解决的问题就是如何在计算机(或数字系统)中表示运算的数据。

数制即计数的规则,指用一组固定的符号和统一的规则来表示数值的方法。如在计数过程中采用进位的方法,则称为进位记数制,简称进位制。在进位记数制中,表示数的符号在不同的位置上所代表的数的值是不同的。首先介绍下面几个概念。

- **数码:** 数制中表示基本数值大小的不同数字符号。
- **基数:** 数制所使用数码的个数。
- **位权:** 数制中某一位上的数所表示数值的大小(所处位置的价值)。

平时人们熟悉使用的十进制有 10 个数码: 0、1、2、3、4、5、6、7、8、9;基数为 10;位权为 10 的幂。

相应地,任意 R 进制数的基数为 R ,有 R 个数码 $0 \sim R-1$,位权为 R 的幂。

数字系统中常用的数制有十进制、二进制、八进制和十六进制。它们的数码如表 1-1 所示。

需要特别说明,因为数字符号只能占用一位,因此在十六进制中用 A、B、C、D、E、F 分别表示 10~15。

表 1-1 常用数制的数码

← R →	数码															
R=2	0	1														
R=8	0	1	2	3	4	5	6	7								
R=10	0	1	2	3	4	5	6	7	8	9						
R=16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

1.1.2 常用数制的表示

1. 常用数制的区分

一般使用 2、8、10、16 作为下标或者使用 B(Binary)、O(Octal)、D(Decimal)、H(Hexdecimal)作为后缀或下标来分别表示二进制数、八进制数、十进制数和十六进制数。

例如：

$$(246)_{10} = (246)_D = 246D = 246$$

$$(100100)_2 = (100100)_B = 100100B$$

$$(207)_8 = (207)_O = 207O$$

$$(12C)_{16} = (12C)_H = 12CH$$

需要说明,在表示八进制数时,由于字母 O 极容易和数字 0 混淆,因此一般不推荐数值加后缀 O 的表示方法;如果一个数没有给出表示其进制数的下标或者后缀,则默认为十进制数。

2. 十进制数的表示

十进制数的特点:

- 有十个数码,即 0~9;
- 有小数点;
- 逢十进一;
- 具有位权 10^i 。

例如,十进制数 567.38,它是 5 个 100、6 个 10、7 个 1、3 个 0.1、8 个 0.01 的和,因此可以将其展开表示为:

$$(567.38)_{10} = 5 \times 10^2 + 6 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2} \quad (1-1)$$

式(1-1)左侧的表示形式称为位置表示法,右侧的表示形式称为按权展开式。

推广开来,可以得到任意一个十进制数 N 的位置表示法式(1-2)和按权展开式(1-3):

$$(N)_{10} = (K_{n-1}K_{n-2}\cdots K_1K_0 \cdot K_{-1}K_{-2}\cdots K_{-m})_{10} \quad (1-2)$$

$$\begin{aligned} (N)_{10} &= K_{n-1} \cdot 10^{n-1} + K_{n-2} \cdot 10^{n-2} + \cdots + K_1 \cdot 10^1 + K_0 \cdot 10^0 \\ &\quad + K_{-1} \cdot 10^{-1} + K_{-2} \cdot 10^{-2} + \cdots + K_{-m} \cdot 10^{-m} \\ &= \sum_{i=-m}^{n-1} K_i \cdot 10^i \end{aligned} \quad (1-3)$$

其中,10 是基数, 10^i 为位权, 权系数 K_i 为正整数且满足 $K_i \in [0, 9]$, n 和 m 分别表示整数部分和小数部分的位数。

3. 任意 R 进制数的表示

由十进制数的表示可以推断出任意 R 进制数的特点:

- 有 R 个数码, 即 $0 \sim R-1$;
- 有小数点;
- 逢 R 进一;
- 具有位权 R^i 。

因此, 任意一个 R 进制数 N 可以表示为:

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0 \cdot K_{-1}K_{-2}\cdots K_{-m})_R \quad (1-4)$$

$$\begin{aligned} (N)_R &= K_{n-1} \cdot R^{n-1} + K_{n-2} \cdot R^{n-2} + \cdots + K_1 \cdot R^1 + K_0 \cdot R^0 \\ &\quad + K_{-1} \cdot R^{-1} + K_{-2} \cdot R^{-2} + \cdots + K_{-m} \cdot R^{-m} \\ &= \sum_{i=-m}^{n-1} K_i \cdot R^i \end{aligned} \quad (1-5)$$

其中, R 是基数, R^i 为位权, 权系数 K_i 为正整数且满足 $K_i \in [0, R-1]$, n 和 m 分别表示整数部分和小数部分的位数。

例如, 当 R 分别取 2、8、16 时, 可以得到对应数的两种表示法:

$$(100100.1)_B = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$$

$$(207.35)_O = 2 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 5 \times 8^{-2}$$

$$(12C.A8)_H = 1 \times 16^2 + 2 \times 16^1 + C \times 16^0 + A \times 16^{-1} + 8 \times 16^{-2}$$

4. 数字系统中的常用数制

数字系统内部数据都是采用二进制形式表示, 这是因为:

- 二进制形式便于物理元件的实现;
- 二进制数只有 0 和 1 两个数字, 恰好可以用物理元件的两种稳定状态来表示, 例如晶体管的导通和截止, 光盘的凸区和凹槽, 只要规定其中一个状态为 1, 另一个状态为 0, 就可以表示二进制数了;
- 二进制运算规则简单, 实现运算的电路也会相应简单;
- 二进制的乘法运算规则只有 4 个 $0 \times 0 = 0$; $0 \times 1 = 0$; $1 \times 0 = 0$; $1 \times 1 = 1$;
- 二进制数码 0、1 表示真、假逻辑量, 便于计算机方便地进行逻辑运算。

但是当二进制数位过长时, 就会暴露出书写冗长, 阅读、书写不方便的缺点, 因此, 人们在书写和表达时常用八进制和十六进制作为中间过渡进制, 它们既能克服二进制缺点, 又能与二进制直接转换。

1.2 不同数制间的转换

一个数据在不同数制之间转换时,必然要求转换前后数值大小不变,即保证转换前后的两个数的整数部分和小数部分分别相等。按照这一要求,可以总结得出十进制、二进制、八进制和十六进制数之间相互转换的方法。

1.2.1 其他进制数转换为十进制数

将二进制数、八进制数、十六进制数转换成十进制数的方法是:按权展开,对多项式进行算术求和,结果用十进制的位置表示法给出。

下面给出几个二进制、八进制、十六进制数转换成十进制数的例子,结果要求精确到小数点后四位。注意:严格意义上需要计算小数点后位数,保证转换精确,具体方法在后面介绍。

【例 1-1】 $(1101.1001)_B = (?)_D$

$$\begin{aligned} (1101.1001)_B &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= (8 + 4 + 0 + 1 + 0.5 + 0.0625)_D \\ &= (13.5625)_D \end{aligned}$$

【例 1-2】 $(3CF.0D)_H = (?)_D$

$$\begin{aligned} (3CF.0D)_H &= 3 \times 16^2 + 12 \times 16^1 + 15 \times 16^0 + 0 \times 16^{-1} + 13 \times 16^{-2} \\ &= (768 + 192 + 15 + 0 + 13/256)_D \\ &\approx (975.0508)_D \end{aligned}$$

【例 1-3】 $(376.25)_O = (?)_D$

$$\begin{aligned} (376.25)_O &= (3 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2})_D \\ &= (192 + 56 + 6 + 2/8 + 5/64)_D \\ &\approx (254.3281)_D \end{aligned}$$

1.2.2 十进制数转换为其他进制数

转换原则:若两数相等,则它们的整数部分和小数部分分别相等。

转换方法:对整数部分和小数部分分别转换,整数部分的转换采用“除基取余”法,小数部分的转换采用“乘基取整”法,具体操作如下:

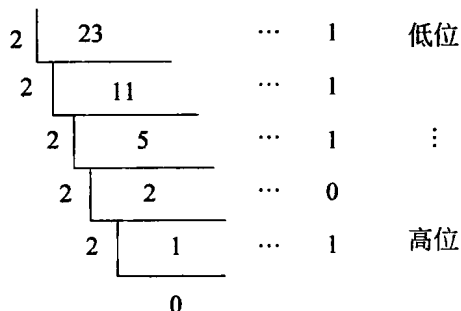
“除基取余”法:用目标数制的基数(R)去除十进制数,第一次相除所得余数是目的数的最低位(K_0),将所得的商再除以该基数,重复该过程直到商等于 0 为止,然后逆序排列余数。

“乘基取整”法:用目标数制的基数(R)去乘十进制小数,第一次相乘结果的整数部分为目的数的最高位(K_{-1}),将其小数部分再乘以该基数,重复该过程,直到乘积的小数部分为 0 或满足要求的精度为止(即根据设备字长限制,取有限位的近似值),然后顺序排列每次乘积的整数部分。

下面以十进制数转换成二进制数为例进行说明。

【例 1-4】 $(23.36)_D = (?)_B$, 精确到小数点后 4 位。

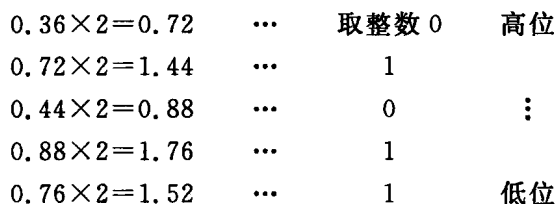
整数部分: 除以 2 取余



即

$$(23)_D = (10111)_B$$

小数部分: 乘 2 取整



即

$$(0.36)_D = (0.01011)_B$$

综上所述可得

$$(23.36)_D = (10111.01011)_B$$

按照上述方法,可以得到十进制数 0~16 在二、八、十六进制中的转换结果,如表 1-2 所示。

表 1-2 0~16 的不同进制表示

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

续表

十进制	二进制	八进制	十六进制
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

小数的转换精度

由例 1-4 可看出,小数不一定能够准确转换,即存在一个转换精度问题。那么,将一个 R 进制的 i 位纯小数转换成 L 进制的小数时,需要至少保留多少位才能保证原来的精度呢?

假设至少要保留 j 位,则有:

i 位 R 进制小数的精度为 R^{-i} , j 位 L 进制小数的精度为 L^{-j} ,要保证原有精度,应该满足:

$$L^{-j} \leq R^{-i} \quad (1-6)$$

求解上式可得:

$$j \geq i \lg R / \lg L \quad (1-7)$$

在例 1-4 中,将 0.36 转换成二进制数时, $R=10, L=2, i=2$, 代入式(1-7)得

$$j \geq 2 \lg 10 / \lg 2 \approx 6.6$$

所以,若要保证原有精度,转换后的二进制小数至少应保留 $j=7$ 位。

1.2.3 二、八、十六进制数间的转换

1. 二进制数与八进制数间的转换

转换方法 1: 根据 1.2.1 节中提供的方法,首先将待转换数转换为十进制数,然后将得到的十进制数转换为目标进制数。

【例 1-5】 $(110101.011)_B = (?)_O$

解: $(110101.011)_B = (53.375)_D = (65.3)_O$

【例 1-6】 $(613.24)_O = (?)_B$

解: $(613.24)_O = (395.3125)_D = (110001011.0101)_B$

转换方法 2: 由表 1-3 可以看到,八进制的 8 个数码 0~7 与 3 位二进制的 8 种组合存在对应关系。

表 1-3 八进制数码与 3 位二进制组合的对应关系

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

则可以采用分组对应转换的方法：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，最后一组不足三位时，分别在整数的最高位前面和小数的最低位后面补 0 凑齐三位。然后，将每组的三位二进制数转换成对应的八进制数字符号。

仍以例 1-5 和例 1-6 为例说明：

例 1-5 的解法 2： $(110101.011)_B = (?)_O$

二进制数	<u>110</u>	<u>101</u>	.	<u>011</u>
八进制数	6	5	.	3

所以， $(110101.011)_B = (65.3)_O$

将八进制数转换成二进制数的方法是：按位对应转换。将八进制数中的每一位数字符号表示成对应的 3 位二进制数，去掉整数部分首部和小数部分尾部的 0。

例 1-6 的解法 2： $(613.24)_O = (?)_B$

八进制数	6	1	3	.	2	4
	↓	↓	↓		↓	↓
二进制数	110	001	011	.	010	100

所以， $(613.24)_O = (110001011.0101)_B$

2. 二进制数与十六进制数间的转换

转换方法 1：根据 1.2.1 节提供的方法，首先将待转换数转换为十进制数，然后将得到的十进制数转换为目标进制数。

【例 1-7】 $(101110101.101)_B = (?)_H$

解： $(101110101.101)_B = (373.625)_D = (175.A)_H$

【例 1-8】 $(729.B4)_H = (?)_B$

解： $(729.B4)_H = (1833.703125)_D = (11100101001.101101)_B$

转换方法 2：同样由表 1-2 可以看到，十六进制的 16 个数码 0~F 与 4 位二进制的 16 种组合存在一一对应关系，因此也可以采用分组对应转换的方法，即：

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，每四位分成一组，最后一组不足四位时，分别在整数的最高位前面和小数的最低位后面补 0 凑齐四位。然后，将每组的四位二进制数转换成对应的十六进制数字符号。

仍以例 1-7 和例 1-8 为例说明：

例 1-7 的解法 2： $(101110101.101)_B = (?)_H$

解：	二进制数	<u>0001</u>	<u>0111</u>	<u>0101</u>	.	<u>1010</u>
	十六进制数	1	7	5	.	A

所以， $(101110101.101)_B = (175.A)_H$

将十六进制数转换成二进制数的方法是：按位对应转换。将十六进制数中的每一位数字符号表示成对应的 4 位二进制数，去掉整数部分首部和小数部分尾部的 0。

例 1-8 的解法 2： $(729.B4)_H = (?)_B$

解：	十六进制数	7	2	9	.	B	4
		↓	↓	↓		↓	↓
	二进制数	0111	0010	1001	.	1011	0100