



计算机系统结构系列教材

数字逻辑原理 与工程设计

刘真 李宗伯 文梅 陆洪毅 编

高等教育出版社

计算机系统结构系列教材

数字逻辑原理与工程设计

刘 真 李宗伯 文 梅 陆洪毅 编

高等教育出版社

内 容 提 要

本书系统阐述了数字逻辑电路的分析和设计方法以及基本的数字系统工程实现技术。内容包括: 数制与编码, 布尔代数, 布尔函数的基本形式和化简, 组合电路、同步时序电路和异步时序电路的分析与设计, 采用中、大规模集成电路实现设计的具体实例, 数字系统设计方法, 数字电路设计中的测试问题和可测性设计技术, 可编程逻辑器件的结构原理, VerilogHDL 语言及综合设计实例等。

本书取材较新, 采用实例教学的组织形式, 内容由浅入深, 引人入胜。书中给出了大量例题, 书后还附有部分习题答案。本书可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. —北京: 高等教育出版社, 2003. 11

(计算机系统结构系列教材)

ISBN 7-04-013321-0

I. 数… II. 刘… III. ①数字逻辑—理论—高等学校—教材②数字系统—系统设计—高等学校—教材
IV. TP302.2

中国版本图书馆 CIP 数据核字(2003)第 092117 号

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 100011

总 机 010-82028899

购书热线 010-64054588

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京人卫印刷厂

开 本 787×1092 1/16

印 张 25.5

字 数 500 000

版 次 2003 年 11 月第 1 版

印 次 2003 年 11 月第 1 次印刷

定 价 29.50 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

计算机系统结构系列教材编委会

主任：周兴铭

委员：卢锡城 陈火旺 齐治昌 杨晓东
李思昆 杨学军 王志英 窦文华
谢伦国 王凤学 张俊科 宋焕章
张民选 王怀民 张晨曦 宁 洪
戴 葵

出版前言

20世纪中期开始的以知识和信息的产生、处理和应用为基础的信息化革命极大地推动了世界经济的发展,人类社会的进步和国家的繁荣现阶段比任何时候都依赖于信息技术的进步和应用。科教兴国、知识创新已经成为我国溶入全球经济、跻身世界经济强国的基本方针。

计算机技术是信息社会的主要技术之一。面对信息社会和知识经济对高素质计算机人才培养的需求,面对信息技术对高素质计算机人才培养所带来的深刻变革,我们一直都在探索和思考:计算机科学与技术专业的人才应具备怎样的知识结构?如何培养适应学科发展的创新人才?如何逐步缩小与国外高素质计算机人才培养之间的差距?

国防科学技术大学计算机学院的“计算机系统结构”学科1988年首批列入国家级重点学科之一,1985年首批设立博士后科研流动站。经过30多年的建设和发展,形成了以高性能计算机系统、分布式计算机系统、计算机仿真系统、计算机网络与通信、移动计算、嵌入式系统和计算机性能评测等为主的学科发展方向,形成了本科、硕士和博士等层次的完整科学系统的人才培养体系,获得了“银河”系列计算机、核心路由器、高性能微处理器等多项高水平科研成果,为国家和国防现代化建设做出了突出的贡献,培养了大批优秀的计算机人才,积累了丰富的理论和实践经验,学科建设整体水平保持全国领先地位。

近年来,结合我院多年的人才培养和科学研究工作实践和经验,我们逐步确立了“加强核心基础知识教育,分阶段、分层次突出创新能力和工程实践综合能力”高素质计算机人才培养的总体思路,并制定了2002年计算机科学与技术专业人才培养计划。

人才培养计划的落实和实施离不开优秀的教材体系作为支撑。在计算机系统结构专业教材体系建设上,我们曾经撰写出版了多部在国内有影响的专著和教材,受到了国内计算机教育工作者的高度评价,并被许多高等院校采用。为进一步适应计算机科学技术的飞速发展和对计算机创新人才培养的需要,发挥我院计算机系统结构专业的学科优势,在高等教育出版社的大力支持下,我们决定面向国内大专院校计算机系统结构专业方向的人才培养要求,结合我院多年来在计算机系统结构专业方向学科建设的成果,重新撰写出版一套计算机系统结构专业方向的系列教材。

为此,我们成立了由多名院士、专家和教授组成的计算机系统结构系列教材编委会,一批长期工作在一线、从事过多年教学和科研实践的中青年教师作为该系列教材的作者。编

委会委员和该系列教材作者在广泛调研国内外的相关最新教材的基础上,结合计算机系统结构专业方向人才培养的特点和规律,统筹规划,力求完整,经过反复讨论,确定了该系列教材的内容体系和知识点的衔接。

该套系列教材主要包括:《电路与电子技术基础》、《数字逻辑原理与工程设计》、《计算机原理与设计》、《汇编语言程序设计》、《计算机体系结构》、《嵌入式系统设计》和《数字系统设计工程》等。其内容全面涵盖了计算机系统结构专业方向的基础理论、基本原理和工程实践等专业知识,既保持了该专业方向基础知识的稳定性,又紧随计算机技术的发展前沿,反映了该专业方向的最新研究内容和研究成果。

该系列教材内容既相互关联,又相对独立,教师可方便地根据具体情况选取相关的部分作为教学内容。该系列教材既可以作为计算机专业本科生的必修课或选修课教材使用,也可以作为有关研究人员的参考书使用。我们十分希望该系列教材的出版能够为培养我国高素质计算机人才贡献一份力量。该系列教材的出版也难免有许多不足之处,希望能够得到国内同行的批评指正,让我们一起交流、共同研究、共同完善。

计算机系统结构系列教材编委会

2003年9月

作者前言

“数字逻辑”是计算机科学与技术学科大学本科学生的专业基础课,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,在计算机系统结构系列教材编委会的指导下,确定了立足于数字逻辑基本原理,分层次、分阶段介绍数字系统工程设计的编写方针。

全书被组织成两部分,共 11 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和简单数字系统的工程设计。其中第一至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;第三至第五章介绍数字逻辑电路的分析和设计方法,包括组合电路、同步时序电路和异步时序电路的分析与设计以及采用中、大规模集成电路实现设计的具体实例;第六章介绍简单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。后一部分为第七至第十一章,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、第八章介绍数字系统的设计方法与设计步骤,数字电路设计中的测试问题和可测性设计技术;第九章重点介绍 Xilinx 和 Altera 公司的两种主流可编程逻辑器件 FPGA/CPLD 的结构特点与工作原理;第十章通过实例逐步引入对 VerilogHDL 语言的介绍,并就测试环境设计的一般方法以及有关 VerilogHDL 设计的一些高级话题进行了讨论;第十一章通过实例介绍如何使用 VerilogHDL 语言设计组合逻辑电路和时序逻辑电路,并讨论了一个完整的 8 位 CPU 的综合实例,介绍利用 VerilogHDL 设计 FPGA 应用和 ASIC 芯片的过程。

本书第一至第三章由文梅编写,第四至第六章由刘真编写,第七至第九章由李宗伯编写,第十、第十一章及附录由陆洪毅编写,刘真统稿。周南良和汪诗林两位老师无偿地提供了使用多年的教学讲稿电子版,并在成书过程中提出了许多宝贵意见,在此表示感谢。北京航空航天大学夏宇闻教授审阅了本书,并提出了许多建设性的意见,编者尽力进行了修改。

本书取材较新,叙述清楚,注重实用。书中给出了大量例题,部分习题答案可以从高等教育出版社网站 cs.hep.com.cn 上下载以便于自学。对于个别内容较深的章节,标注了星

号,以供不同层次的读者选用,适合作为高等学校本科计算机专业和相关专业的教材,也可供科技人员参考。

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
2003年9月

目 录

第一章 数制与编码	(1)	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”表示形式	(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	(52)
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(Q-M法)	(62)
1.2.2 反码	(14)	小结	(69)
1.2.3 补码	(15)	习题	(70)
1.2.4 原码、反码和补码 之间的转换	(18)	第三章 组合逻辑电路的分析和设计	(74)
1.2.5 溢出的判断和变形码	(20)	引言	(74)
1.3 常用的其他编码	(22)	3.1 常用逻辑门的图形符号	(74)
1.3.1 十进制数的二进制编码	(22)	3.2 布尔函数的实现	(75)
1.3.2 字符代码	(24)	3.2.1 用与非门实现布尔函数	(76)
1.3.3 可靠性编码	(26)	3.2.2 用或非门实现布尔函数	(76)
小结	(34)	3.2.3 用与或非门实现布尔函数	(77)
习题	(34)	3.3 组合逻辑电路的分析	(78)
第二章 布尔代数基础	(37)	3.4 组合逻辑电路的设计	(80)
引言	(37)	3.5 常用组合逻辑电路	(84)
2.1 布尔代数的基本概念	(37)	3.5.1 加法器	(84)
2.1.1 布尔变量及其基本运算	(37)	3.5.2 十进制数字的七段显示	(92)
2.1.2 布尔函数及其表示方法	(39)	3.5.3 代码转换电路	(96)
2.1.3 布尔函数的“相等”概念	(40)	3.6 二进制译码器	(98)
2.2 布尔代数的公式、定理和规则	(41)	3.6.1 二进制译码器的功能和组成	(98)
2.2.1 布尔代数的基本公式	(41)	3.6.2 用中规模集成译码器 进行设计	(100)
2.2.2 布尔代数的主要定理	(42)	3.7 多路选择器	(102)
2.2.3 布尔代数的重要规则	(44)	3.7.1 多路选择器的逻辑 功能和组成	(102)
2.3 布尔函数的基本形式	(45)		
2.3.1 函数的“积之和”与“和之积”			

3.7.2 用多路选择器进行逻辑设计 … (103)	5.1.3 异步时序电路的类型 …… (166)
3.8 多路分配器 …… (107)	5.2 脉冲异步时序电路的分析与设计 … (167)
3.9 组合逻辑电路中的险态 …… (109)	5.3 电平异步时序电路的分析与设计 … (170)
小结 …… (112)	5.3.1 建立原始流程表 …… (171)
习题 …… (112)	5.3.2 流程表的简化 …… (174)
第四章 同步时序电路 …… (115)	5.3.3 流程表的状态分配 …… (175)
引言 …… (115)	* 5.3.4 电平异步时序电路的险态 …… (181)
4.1 时序电路与时序机 …… (115)	5.3.5 分析与设计举例 …… (182)
4.1.1 时序电路的结构和特点 …… (116)	小结 …… (189)
4.1.2 时序机的定义 …… (117)	习题 …… (189)
4.1.3 时序机的状态表和状态图 …… (117)	第六章 简单可编程逻辑器件及其应用 … (192)
4.1.4 完全定义机和不完全定义机 … (119)	引言 …… (192)
4.2 触发器 …… (119)	6.1 概述 …… (193)
4.2.1 RS 触发器 …… (120)	6.1.1 可编程逻辑器件的发展历史 … (193)
4.2.2 JK 触发器 …… (121)	6.1.2 可编程逻辑器件中的编程 元件 …… (194)
4.2.3 T 触发器 …… (122)	6.1.3 可编程逻辑器件的分类 …… (195)
4.2.4 D 触发器 …… (122)	6.1.4 PLD 的基本结构 …… (196)
4.3 同步时序电路的分析与设计 …… (123)	6.1.5 PLD 的逻辑表示法 …… (197)
4.3.1 建立原始状态表 …… (124)	6.1.6 常用 PLD 器件的比较 …… (198)
4.3.2 状态表的化简 …… (126)	6.2 可编程只读存储器 PROM …… (198)
4.3.3 状态分配 …… (136)	6.2.1 PROM 基本结构和工作原理 … (199)
4.3.4 确定激励函数和输出函数 …… (139)	6.2.2 采用 PROM 的组合逻辑设计 … (201)
4.3.5 分析与设计举例 …… (142)	6.3 可编程逻辑阵列 PLA …… (202)
4.4 常用的同步时序电路 …… (148)	6.4 可编程阵列逻辑 PAL …… (207)
4.4.1 寄存器 …… (148)	6.5 通用阵列逻辑 GAL …… (212)
4.4.2 计数器 …… (150)	6.6 简单 PLD 设计方法及步骤 …… (219)
4.4.3 节拍信号发生器 …… (153)	6.6.1 PLD 器件的设计步骤 …… (219)
小结 …… (158)	6.6.2 可编程器件设计软件简介 …… (221)
习题 …… (158)	6.6.3 可编程逻辑器件设计举例 …… (224)
第五章 异步时序电路 …… (163)	小结 …… (230)
引言 …… (163)	习题 …… (231)
5.1 异步时序电路的结构和描述 …… (164)	第七章 数字系统设计基础 …… (235)
5.1.1 异步时序电路的结构模型 和特点 …… (164)	引言 …… (235)
5.1.2 异步时序电路的描述方法 ——流程表 …… (165)	7.1 基本概念 …… (235)
	7.1.1 数字系统的定义 …… (235)

7.1.2 数字系统的优点	(237)	引言	(271)
7.1.3 数字系统的模型与结构	(237)	9.1 概述	(271)
7.2 数字系统的设计	(239)	9.2 CPLD 的结构特点	(273)
7.2.1 数字系统的实现途径	(239)	9.2.1 概述	(273)
7.2.2 数字系统的设计过程	(241)	9.2.2 Altera MAX 7000 系列 CPLD 的 结构特点	(274)
7.2.3 数字系统辅助设计工具	(242)	9.2.3 应用实例	(279)
7.3 数字系统设计的发展趋势	(242)	9.3 FPGA 的结构特点	(280)
7.4 后续章节内容与使用建议	(243)	9.3.1 概述	(280)
小结	(243)	9.3.2 Xilinx Spartan II 系列 FPGA 的 结构特点	(281)
习题	(244)	9.3.3 应用实例	(288)
第八章 数字系统的设计	(245)	小结	(288)
引言	(245)	习题	(289)
8.1 基本概念	(245)	第十章 VerilogHDL 语言	(290)
8.1.1 设计的表示	(245)	引言	(290)
8.1.2 设计的抽象层次	(247)	10.1 概述	(290)
8.1.3 结构化设计	(249)	10.2 一个数字系统实例及其 VerilogHDL 语言描述	(292)
8.1.4 设计窗口与设计空间	(251)	10.3 信号	(295)
8.2 需求分析	(251)	10.4 模块体的描述	(299)
8.2.1 需求分析的主要内容	(251)	10.5 行为级描述	(311)
8.2.2 需求分析报告	(252)	10.5.1 变量和参数	(312)
8.2.3 实例	(253)	10.5.2 行为级描述基础	(313)
8.3 系统设计	(254)	10.5.3 复杂语句	(314)
8.3.1 基于通用微处理器/DSP 的 设计过程	(254)	10.5.4 时序控制	(317)
8.3.2 基于集成电路的设计过程	(256)	10.5.5 事件控制	(318)
8.3.3 基于可编程逻辑器件的 设计过程	(259)	10.6 其他的 VerilogHDL 语言结构	(320)
8.3.4 基于 ASIC 的设计过程	(259)	10.6.1 函数和任务	(320)
8.4 系统实现	(260)	10.6.2 VerilogHDL 系统函数和系统 任务	(321)
8.5 数字系统的测试和可测试性设计	(261)	10.6.3 VerilogHDL 预编译指令	(323)
8.5.1 数字系统的测试	(262)	10.7 可综合性设计	(324)
8.5.2 可测性设计	(264)	10.8 测试环境(Testbench)设计	(327)
小结	(269)	* 10.9 VerilogHDL 设计高级专题	(328)
习题	(269)	10.9.1 寄存器	(328)
第九章 复杂可编程逻辑器件 FPGA/CPLD	(271)		

10.9.2 多时钟域·····	(331)	11.2.1 时序逻辑电路的基本特征·····	(344)
10.9.3 同步复位与异步复位·····	(333)	11.2.2 有限状态机的设计·····	(344)
10.9.4 VerilogHDL 语言的描述风格···	(336)	11.2.3 实例 1:寄存器·····	(347)
小结·····	(338)	11.2.4 实例 2:移位器·····	(348)
习题·····	(338)	11.3 一个简单的 8 位 CPU 系统设计···	(349)
第十一章 用 VerilogHDL 语言设计		11.3.1 KD-CPU 简介·····	(349)
数字电路 ·····	(340)	11.3.2 KD-CPU 的 VerilogHDL 描述·····	(352)
引言·····	(340)	11.3.3 KD-CPU 的实现·····	(369)
11.1 组合逻辑电路设计和描述·····	(340)	小结·····	(372)
11.1.1 组合逻辑电路基本特征·····	(340)	习题·····	(372)
11.1.2 实例 1:多路数据选择器·····	(340)	附录 Verilog HDL 语言语法参考 ·····	(374)
11.1.3 实例 2:加法器·····	(341)	参考文献 ·····	(395)
11.1.4 实例 3:译码器·····	(342)		
11.2 时序逻辑电路设计和描述·····	(344)		

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。在日常生活中就是按这种进位制计数的,如十进制、十二进制、六十进制,等等。

对于任何一个数,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。

十进制有十个数字符号,即0、1、2、3、4、5、6、7、8、9。将若干个这样的符号并列在一起,可以表示一个十进制数,每位不超过“9”,由低位向高位进位是“逢十进一”。这是十进制的特点。

这里要引入两个术语:一个叫“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“十”;另一个叫“位权”或“权”,它表示某种进位制的数中不同位置上数字的单位数值,如十进制数135.79,最左位为百位(1代表100),权为 10^2 ;第二位为十位(3代表30),权为 10^1 ;第三位为个位(5代表5),权为 10^0 ;小数点右边第一位为十分位(7代表7/10),权为 10^{-1} ;第二位为百分位(9代表9/100),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 ,它可表示成

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

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

式中, 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, 把式(1.1)称为并列表示法, 把式(1.2)称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 , 它可表示成

$$(N)_R = (r_{n-1}r_{n-2}\cdots r_1r_0.r_{-1}r_{-2}\cdots r_{-m})_R \\ = r_{n-1}R^{n-1} + r_{n-2}R^{n-2} + \cdots + r_1R^1 + r_0R^0 \\ + r_{-1}R^{-1} + r_{-2}R^{-2} + \cdots + r_{-m}R^{-m} \\ = \sum_{i=-m}^{n-1} r_iR^i$$

式中, 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

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

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记数或进行算术运算; 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, 从中领会它的运算规则。

例 1 两个二进制数相加, 采用“逢二进一”的法则。

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ +) \quad 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 1\ 1\ 0 \end{array}$$

例 2 两个二进制数相减, 采用“借一当二”的法则。

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ -) 0\ 1\ 1\ 0 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

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。

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ \times) 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 1 \\ 0\ 0\ 0\ 0 \\ 1\ 0\ 1\ 1 \\ 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \end{array}$$

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。

$$\begin{array}{r} 1\ 0\ 1\ 0 \quad \dots\dots \text{商} \\ 1\ 1\ 0\ 1 \overline{) 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1} \\ \underline{1\ 1\ 0\ 1} \\ 1\ 0\ 0\ 0\ 0 \\ \underline{1\ 1\ 0\ 1} \\ 1\ 1\ 1 \quad \dots\dots \text{余数} \end{array}$$

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、3、4、5、6、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、3、4、5、6、7、8、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 时表示十进制数值 0 ~ 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
4	100	4	4

续表

$R = 10$	$R = 2$	$R = 8$	$R = 16$
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数转换成人们习惯的十进制数。

1. 直接转换法

由于1位八进制的8个数字符号正好相应于3位二进制数的八种不同组合,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,八进制与二进制之间数的转换就极为方便。

例5 将二进制数 11010.1101 转换为八进制数。

$$\begin{array}{ccccccc} \underline{011} & \underline{010} & . & \underline{110} & \underline{100} & & \\ 3 & 2 & . & 6 & 1 & & \end{array}$$

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3位分成一组、头尾不足3位的补0;然后将每组的3位二进制数转换为1位八进制数。

例6 将八进制数357.6转换为二进制数。

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

所以, $(357.6)_8 = (11101111.11)_2$ 。

同理,由于1位十六进制的16个数字符号正好相应于4位二进制数的十六种不同的组合,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,十六进制与二进制之间数的转换也很方便。

例7 将二进制数1010110110.110111转换为十六进制数。

$$\begin{array}{ccccccc} \underline{0010} & \underline{1011} & \underline{0110} & . & \underline{1101} & \underline{1100} & \\ 2 & B & 6 & . & D & C & \end{array}$$

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。

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

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