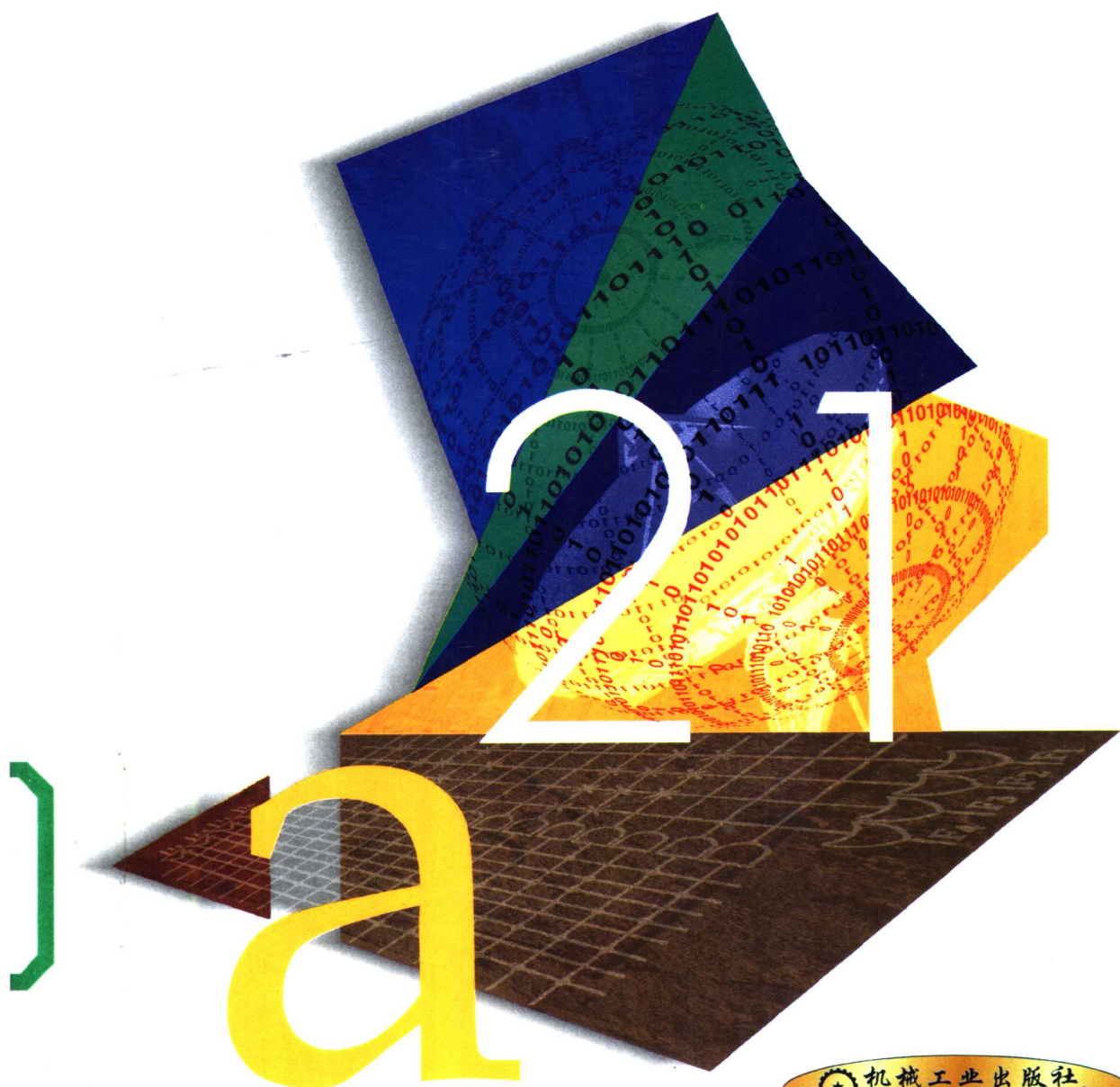




● 徐惠民 安德宁 编著

数字逻辑设计与VHDL描述

21世纪高等院校通信与信息专业系列教材



 机械工业出版社
China Machine Press

21 世纪高等院校通信与信息专业系列教材

数字逻辑设计与 VHDL 描述

徐惠民 安德宁 编著



机械工业出版社

本书是适应 21 世纪需要的“数字逻辑设计与 VHDL 描述”教材。本书在保留“数字电路与逻辑设计”的系统性和完整性基础上,详细介绍了用 VHDL 硬件描述语言对数字电路和系统进行描述和设计的方法。全书包括数字逻辑设计基础,以 CMOS 为主的数字集成电路,组合电路的分析、设计和描述,时序电路的分析、设计和描述,可编程逻辑器件,数字系统的描述和设计。全书强调基本概念和基本方法,每章都有相当数量的习题和思考题。

本书可作高等院校通信与信息专业的教材,也可作为相关技术人员参考和培训教材。

图书在版编目(CIP)数据

数字逻辑设计与 VHDL 描述/徐惠民, 安德宁编著. —北京: 机械工业出版社, 2002.3

21 世纪高等院校通信与信息专业系列教材

ISBN 7-111-09955-9

I. 数… II. ①徐…②安… III. ①数字逻辑—逻辑设计—高等学校—教材②硬件描述语言, VHDL—程序设计—高等学校—教材 IV. TP302.2

中国版本图书馆 CIP 数据核字 (2002) 第 013049 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚 责任编辑: 周艳娟 王琼先 版式设计: 霍永明

责任校对: 韩 晶 责任印制: 路 琳

中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行

2002 年 4 月第 1 版·第 1 次印刷

1000mm×1400mmB5·10.75 印张·442 千字

0 001—5000 册

定价: 26.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、68326677-2527

21 世纪高等院校

通信与信息专业教材编委会名单

(按姓氏笔画排序)

编委会主任	乐光新		
编委会副主任	张文军	张思东	杨海平
	陈瑞藻	徐澄圻	
编委会委员	王金龙	冯正和	刘增基
	李少洪	邹家禄	吴镇扬
	赵尔沅	徐惠民	彭启琮
	解月珍		
秘书长	胡毓坚		

出版说明

为了培养 21 世纪国家和社会急需的通信与信息领域的高级科技人才,为了配合高等院校通信与信息专业的教学改革和教材建设,机械工业出版社会同全国在通信与信息领域具有雄厚师资和技术力量的高等院校,组成阵容强大的编委会,组织长期从事教学的骨干教师编写了这套面向普通高等院校的通信与信息专业系列教材,并且将陆续出版。

这套教材将力求做到:专业基础课教材概念清晰、理论准确、深度合理,并注意与专业课教学的衔接;专业课教材覆盖面广、深度适中,不仅体现相关领域的最新进展,而且注重理论联系实际。

这套教材的选题是开放式的。随着现代通信与信息技术日新月异地发展,我们将不断更新和补充选题,使这套教材及时反映通信与信息领域的新发展和新技术。我们也欢迎在教学第一线有丰富教学经验的教师及通信与信息领域的科技人员积极参与这项工作。

由于通信与信息技术发展迅速而且涉及领域非常宽,这套教材的选题和编审中如有缺点和不足之处,诚恳希望各位老师和同学提出宝贵意见,以利于今后不断地改进。

机械工业出版社

高等院校通信与信息专业系列教材编委会

前 言

21 世纪是信息化的时代，数字化是人类进入信息化的必要条件。“数字逻辑设计与 VHDL 描述”是数字化的基础，因此“数字逻辑设计与 VHDL 描述”课程是通信工程、计算机、自动控制、电子工程等电类专业和机电一体化非电专业的一门专业基础课。也是一门发展最快、应用最广的学科。

随着计算技术和数字技术的发展，在现代电子设备中，单纯用模拟电路实现的已经很少见，通常只在微弱信号放大、高频数据采集和大功率输出等局部电路采用模拟电路，其余部分广泛采用数字电路。因此，对大多数电子设备而言，其整体部分是数字系统。为此，对数字电路的分析与设计，就成为电子工程技术人员必备的专业基础知识。

在数字电路与数字系统中，所用逻辑组件已由 20 世纪 60 年代的小规模集成 (SSI)、20 世纪 70 年代后的中规模集成 (MSI) 标准逻辑部件，发展到目前的大规模集成 (LSI)、超大规模集成 (VLSI)、专用集成电路 (ASIC)。集成电路工艺已由 TTL 为主变为以 CMOS 为主。相应地，数字逻辑电路的设计方法，也在不停地演变和发展，使数字系统的设计从传统的单纯硬件设计方法，变为计算机软硬件协同设计的方法。它使电子设计自动化 (EDA) 和电子系统设计自动化 (ESDA) 成为现代电子系统设计和制造中的主要技术手段。

EDA 和 ESDA 技术是现代电子工程师进行电子系统和电子工程设计所必须掌握的技术。

为了适应电子系统设计技术的发展，培养面向 21 世纪、参与国内外市场竞争的电子技术人材，本书在保留“数字电路与逻辑设计”的系统性和完整性基础上，对中小规模的内容作了适当精简，在门电路部分以介绍 CMOS 为主，在组合和时序电路方面，加强了大规模组件方面的内容，特别是在可编程逻辑器件 (PLD) 方面的编程 (用 VHDL 语言) 和使用，作了较详细的介绍。为读者提供了独立分析和设计数字电路和数字系统的工具，帮助读者建立规范有序的思维习惯，提高分析和解决实际问题的能力。

本书既重视基本的逻辑设计概念和方法的介绍，也重视对于用硬件描述语言描述和设计数字电路的介绍。为了使读者能够较早的接触和充分掌握 VHDL 语言的使用，我们采用一边介绍语言，一边介绍语言的应用，最后介绍对于数字系统的描述。

本书是作者依据多年教学的经验 and 科研的经验，参考国内外优秀教材编写而成。全书共分为 9 章，第 1~3 章是预备知识。内容包括数制与编码、逻辑代数基

础和门电路。主要介绍了二进制数和其他进制数之间的互换，二进制数的算术运算，几种常用的二十进制代码及其加法运算，格雷码和差错检测码。对于逻辑函数的化简，只介绍代数法和卡诺图法。在门电路部分主要介绍 CMOS 集成电路的结构和外特性等。对 TTL 和 ECL 电路只作简要介绍。

第 4 章组合逻辑电路。结合实际例子介绍使用中、小规模集成电路进行逻辑设计的方法。

第 5 章开始引入 VHDL 语言及其描述。先介绍最基本的描述方法和语句，并结合对于组合电路的描述，使读者开始熟悉和运用这种方法。

第 6 章介绍触发器，也介绍用 VHDL 语言对于时序电路进行描述的基本方法。

第 7 章介绍时序电路的分析和设计。包括使用中、小规模集成电路的设计，以及用 VHDL 语言对时序电路和系统进行描述的方法。

第 8 章是可编程逻辑器件，介绍了可编程逻辑器件的工作原理，PAL、GAL 和 CPLD 芯片的结构和原理，对于门阵列只作了简单的介绍。

第 9 章是数字系统设计，希望读者通过这一章的学习能够开始自己设计数字系统。

本书在每一章后面有一定数量的习题，其中有些题有一定难度。作者的意图在于引起读者思考，读者不要求做完全部习题，习题数量比较多，为教师和同学提供了选择的机会。我们将考虑编写和出版有关的习题解答和实验指导。

本书的第 1、2、3、4、8 章由安德宁编写，第 5、6、7、9 章由徐惠民编写。彭家浚、李春宜、韩玉芬、殷福全、王启星等也参与了本书编写的有关工作。

本书对于 VHDL 语言的介绍只是基本的，全面的学习可参见有关的资料。

本书的编写相当仓促，不当之处，希望各位老师和同学提出指正。

编 者

目 录

出版说明

前言

第 1 章 数制与编码 1

1.1 进位计数制 1

1.1.1 基数和权 1

1.1.2 二进制数之间的转换 2

1.1.3 十进制数和非十进制数 之间的转换 3

1.2 二—十进制编码 5

1.2.1 几种二—十进制编码 5

1.2.2 二—十进制代码的加法 6

1.3 格雷 (Gray) 码 9

1.4 差错检测码 10

1.4.1 奇偶校验码 10

1.4.2 五中取二码和六中取二码 11

1.5 习题 12

第 2 章 逻辑代数基础 15

2.1 基本概念 15

2.1.1 逻辑变量和逻辑函数 15

2.1.2 基本逻辑运算 15

2.1.3 导出逻辑运算 17

2.1.4 逻辑函数的表示方法 20

2.2 逻辑代数的定理和规则 22

2.2.1 逻辑代数的基本定律 22

2.2.2 常用公式 23

2.2.3 展开定理 24

2.2.4 逻辑代数的三个规则 25

2.3 逻辑函数的标准表达式 26

2.3.1 标准“与或”式 26

2.3.2 标准“或与”式 28

2.3.3 不完全确定的逻辑函数 30

2.4 逻辑函数的化简方法 31

2.4.1 逻辑函数式的化简目标 31

2.4.2 代数化简法 32

2.4.3 卡诺图法化简逻辑函数 33

2.5 习题 43

第 3 章 集成逻辑门电路 48

3.1 概述 48

3.1.1 电压电平 48

3.1.2 正逻辑和负逻辑 49

3.2 MOS 晶体管 50

3.2.1 MOS 晶体管的分类 50

3.2.2 MOS 管的三个工作区 51

3.2.3 MOS 管的开关时间 52

3.3 CMOS 反相器 52

3.3.1 CMOS 反相器的结构及工作 原理 52

3.3.2 CMOS 反相器的电压传输 特性 53

3.3.3 CMOS 反相器的功耗 55

3.3.4 CMOS 反相器的开关时间 56

3.4 CMOS 其他逻辑门电路 57

3.4.1 CMOS 与非门 57

3.4.2 CMOS 或非门 58

3.4.3 门的输入端数的扩展 58

3.4.4 缓冲门、与门及或门 59

3.4.5 CMOS 与或非门和异或门 59

3.4.6 CMOS 传输门 60

3.5 CMOS 集成电路的输出结构 62

3.5.1 推挽输出 62

3.5.2 三态输出 62

3.5.3 漏极开路输出 63

3.5.4 施密特触发器 65

3.6 CMOS 逻辑系列 66

3.6.1 HC 和 HCT 系列 66

3.6.2 VHC 和 VHCT 68

3.6.3 FAC 和 FACT 68

3.6.4 CMOS 电路使用中应注意

的问题	69
3.7 双极型集成逻辑电路	70
3.7.1 晶体三极管非门	70
3.7.2 肖特基晶体三极管	72
3.8 TTL 逻辑门电路	72
3.8.1 TTL 与非门电路	72
3.8.2 LSTTL 或非门	75
3.8.3 TTL 系列	75
3.8.4 TTL 电路使用中注意 的问题	76
3.9 ECL 逻辑电路	77
3.9.1 基本 ECL 电路	77
3.9.2 ECL 系列	78
3.9.3 ECL 电路使用中应注意 的问题	79
3.10 逻辑门的混合逻辑符号	79
3.10.1 缓冲门的混合逻辑符号	80
3.10.2 与门和与非门的混合 逻辑符号	80
3.10.3 或门及或非门的混合 逻辑符号	80
3.11 习题	81
第 4 章 组合逻辑电路	84
4.1 组合逻辑电路的分析	84
4.1.1 组合逻辑电路的分析步骤	84
4.1.2 分析举例	84
4.2 中规模组合逻辑电路	88
4.2.1 编码器	88
4.2.2 译码器	91
4.2.3 数据选择器	97
4.2.4 加法器	101
4.2.5 数值比较器	103
4.2.6 奇偶校验器	106
4.3 组合逻辑电路的综合	108
4.3.1 组合逻辑电路的综合方法	109
4.3.2 组合逻辑电路设计举例	109
4.4 组合逻辑电路中的竞争与冒险	115
4.4.1 冒险的分类	116
4.4.2 冒险的检查及消除	118

4.5 习题	120
--------------	-----

第 5 章 VHDL 描述组合逻辑

电路	126
5.1 硬件描述语言 VHDL	126
5.1.1 设计过程	126
5.1.2 VHDL 语言的基本特点	127
5.2 VHDL 描述的基本结构	128
5.2.1 实体描述	129
5.2.2 结构体描述	130
5.3 数据类型、运算符和表达式	131
5.3.1 枚举类型	131
5.3.2 数组类型	132
5.3.3 子类型	133
5.3.4 VHDL 运算符	134
5.3.5 常量的定义	135
5.3.6 VHDL 表达式	136
5.4 VHDL 的库和包	137
5.4.1 VHDL 库的种类和使用	137
5.4.2 程序包	138
5.4.3 库和程序包的引用	139
5.4.4 函数和过程	140
5.5 并行处理语句	141
5.5.1 并行赋值语句	141
5.5.2 条件赋值语句	142
5.5.3 选择信号赋值语句	143
5.6 顺序描述语句	144
5.6.1 PROCESS 语句	144
5.6.2 信号和变量赋值语句	145
5.6.3 分支语句	145
5.6.4 循环语句	147
5.7 结构描述语句	148
5.7.1 部件声明语句	148
5.7.2 部件描述语句	149
5.7.3 重复部件的描述	150
5.8 VHDL 描述组合逻辑电路	151
5.8.1 译码电路的描述	151
5.8.2 三态门的描述	153
5.8.3 编码器的描述	154
5.9 习题	156

第 6 章 集成触发器	159	7.3.3 序列信号发生器	205
6.1 触发器的基本特性及其记忆作用	160	7.3.4 M 序列发生器	207
6.2 电位型触发器	161	7.4 异步计数器	208
6.2.1 基本 RS 触发器	161	7.4.1 异步计数器的基本形式	209
6.2.2 带使能端的 RS 触发器	163	7.4.2 异步计数器的分析	209
6.2.3 D 触发器	164	7.5 中规模时序集成电路	211
6.2.4 锁存器	165	7.5.1 中规模集成计数器	212
6.3 时钟控制的集成触发器	166	7.5.2 中规模计数器的应用	217
6.3.1 主从触发器	166	7.5.3 中规模移位寄存器	226
6.3.2 T 触发器	170	7.5.4 中规模移位寄存器的应用	229
6.3.3 边沿触发器	171	7.5.5 时序部件的 VHDL 描述	233
6.4 触发器的逻辑符号	173	7.6 一般时序电路的分析和设计	239
6.5 CMOS 触发器	174	7.6.1 一般时序电路的分析	239
6.5.1 带使能端 D 触发器	174	7.6.2 一般时序电路的设计	242
6.5.2 CMOS 主从 D 触发器	175	7.6.3 时序机的 VHDL 描述	248
6.5.3 CMOS JK 触发器	176	7.7 习题	252
6.6 集成触发器的时间参数	177	第 8 章 可编程逻辑器件	261
6.6.1 建立时间和保持时间	178	8.1 概述	261
6.6.2 时钟信号的时间参数	178	8.1.1 专用集成电路的分类	261
6.7 触发器的 VHDL 描述	179	8.1.2 PLD 的基本结构	261
6.7.1 电位型触发器的 VHDL 描述	179	8.1.3 PLD 电路的表示方法	262
6.7.2 钟控型触发器的描述	180	8.1.4 PLD 的分类	263
6.8 习题	183	8.1.5 PLD 的性能特点	265
第 7 章 时序逻辑电路的分析、设计和描述	188	8.2 只读存储器 ROM	265
7.1 时序电路基础	188	8.2.1 ROM 的逻辑结构	265
7.1.1 同步时序电路的分类和描述	188	8.2.2 ROM 的分类	266
7.1.2 常用时序电路	190	8.2.3 ROM 的应用	268
7.2 常用同步时序电路的分析	192	8.3 可编程阵列逻辑 (PAL)	270
7.2.1 同步时序电路分析的步骤	192	8.3.1 PAL 的基本结构	270
7.2.2 同步计数器的分析	192	8.3.2 PAL16L8 的逻辑结构图及应用	272
7.2.3 移位寄存器及其应用电路的分析	195	8.4 通用阵列逻辑 (GAL)	274
7.3 常用时序电路的设计	201	8.4.1 GAL 的性能特点	274
7.3.1 基本的设计步骤	201	8.4.2 GAL 的结构	275
7.3.2 同步计数器的设计	201	8.4.3 输出逻辑宏单元 OLMC	276
		8.5 可擦除可编程的逻辑器件 (EPLD)	279
		8.5.1 MAX7000 系列器件的结构	279

8.5.2 宏单元 (MACROCELL) ...	281	8.8 习题	300
8.5.3 扩展乘积项	282	第 9 章 数字系统设计	303
8.5.4 PIA 和 I/O 控制块	283	9.1 层次化结构设计	303
8.5.5 EPM7128S 应用举例	285	9.1.1 设计的层次	303
8.6 复杂的可编程逻辑器件		9.1.2 系统结构的分解	305
(CPLD)	289	9.2 自顶向下设计方法	306
8.6.1 FLEX 10K 器件的结构	289	9.2.1 自顶向下设计方法的基本	
8.6.2 嵌入阵列块 (EAB)	290	设计过程	306
8.6.3 逻辑阵列块 (LAB)	292	9.2.2 数字系统的基本划分	307
8.6.4 逻辑单元 (LE)	293	9.3 复数运算器的设计	310
8.6.5 快速通道互连	297	9.3.1 设计要求	310
8.6.6 输入输出单元 (IOE)	298	9.3.2 确定系统接口	310
8.7 现场可编程门阵列 (FPGA)		9.3.3 系统划分	311
的特点	299	9.3.4 系统描述	314
8.7.1 FPGA 的基本结构	299	9.4 习题	331
8.7.2 FPGA 的特点	300	参考文献	333

第 1 章 数制与编码

本章简单介绍数制和编码。主要是介绍十进制数的二进制编码。这些编码不但在数字系统中使用，在计算机系统中也广泛地使用。

1.1 进位计数制

1.1.1 基数和权

进位计数制是以表示计数符号的个数来命名的。计数符号的个数称为基数，用符号 r 来表示。

十进制数有 0、1、2、3、4、5、6、7、8、9 十个计数符号，故 $r=10$ 。

二进制数有 0、1 两个计数符号，故 $r=2$ 。一位二进制数只可能是 0 或 1，通常就把一位二进制数称为一个比特。

八进制数有 0、1、2、3、4、5、6、7 八个计数符号，故 $r=8$ 。

十六进制数有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个计数符号，故 $r=16$ 。

进位计数制一般用位置计数法来表示，例如十进制数 323，从右向左即从低位向高位，每位都是逢十进一。因此，同一计数符号处在不同数位，代表的数值不同。各个数位的位值，称为进位计数制各位的权。例如十进制数 323 最低位的权是 10^0 ，次低位的权是 10^1 ，最高位的权是 10^2 。

因此，各个数位的权值都是基数的整数次幂。例如整数第 i 位的权是 r^{i-1} ，小数后第 m 位的权是 r^{-m} 。

一个 r 进制数 $(N)_r$ 的按权展开多项式可以表示为

$$\begin{aligned}(N)_r &= a_{n-1}r^{n-1} + a_{n-2}r^{n-2} + \cdots + a_1r^1 + a_0r^0 + a_{-1}r^{-1} + \cdots + a_{-m}r^{-m} \\ &= \sum_{i=-m}^{n-1} a_i r^i\end{aligned}$$

式中 a_i 为计数符号，又称数码，其取值范围为 $0 \leq a_i \leq (r-1)$ 。 n 为整数部分的位数， m 为小数部分的位数。二进制数各位的权如表 1-1 所示。

表 1-1 二进制数各位的权

二进制位数	13	12	11	10	9	8	7	6	5	4	3	2	1
权	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
十进制数表示	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

(续)

二进制位数	-1	-2	-3	-4	-5	-6
权	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
十进制数表示	0.5	0.25	0.125	0.0625	0.03125	0.015625

例如 $(110101.01)_2$ 可展开为

$$(110101.01)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

1.1.2 二进制数之间的转换

1. 二进制数和八进制数之间的转换

由于 $8 = 2^3$, $16 = 2^4$, 因此二进制数、八进制数和十六进制数都属于 2^n 进制数。

根据二进制数的多项展开式可知：用三位二进制数能表示 0~7 八个数，所以三位二进制数可以表示一位八进制数。

因此，二进制数和八进制数之间的转换是非常简便的。二进制数要转换为八进制数时，只要将二进制数整数部分自右向左每三位分为一组，最后不足三位时左边用 0 补足；小数部分则自左向右每三位分为一组，最后不足三位时右边用 0 补足。再把每三位二进制数对应的八进制数码写出即可。

【例 1-1】 将 $(11101.1101)_2$ 转换成八进制数。

解： 二进制数 = 011 101 . 110 100

八进制数 = 3 5 . 6 4

$$(11101.1101)_2 = (35.64)_8$$

若要将八进制数转换为二进制数时，只要写出每位八进制数码所对应的二进制数，依次排好即可。

【例 1-2】 将 $(234.567)_8$ 转换为二进制数。

解： 八进制数 = 2 3 4 . 5 6 7

二进制数 = 010 011 100 . 101 110 111

$$(234.567)_8 = (10011100.101110111)_2$$

2. 二进制数和十六进制数之间的转换

用四位二进制数能表示 0~15 十六个数，所以四位二进制数可以表示一位十六进制数。

对多于四位的二进制数，要转换为十六进制数时，只要将二进制数整数部分自右向左每四位分为一组，最后不足四位时左边用 0 补足；小数部分则自左向右每四位分为一组，最后不足四位时右边用 0 补足。再把每四位二进制数对应的十六进制

数码写出即可。

【例 1-3】 将 $(11101.1101)_2$ 转换成十六进制数。

解： 二进制数 = 0001 1101 . 1101
 十六进制数 = 1 D . D
 $(11101.1101)_2 = (1D.D)_{16}$

若要将十六进制数转换为二进制数时，只要写出每位十六进制数码所对应的二进制数，依次排好即可，但整数部分高位的 0 和小数部分低位的 0 不用写出。

【例 1-4】 将 $(6F.A8)_{16}$ 转换为二进制数。

解： 十六进制数 = 6 F . A 8
 二进制数 = 0110 1111 . 1010 1000
 $(6F.A8)_{16} = (1101111.10101)_2$

表 1-2 列出了与十进制数 0~16 相对应的二进制 八进制 十六进制数。

表 1-2 与十进制数相对应的二进制、八进制、十六进制数

十进	二进	八进	十六进	十进	二进	八进	十六进	十进	二进	八进	十六进
0	0000	0	0	6	0110	6	6	12	1100	14	C
1	0001	1	1	7	0111	7	7	13	1101	15	D
2	0010	2	2	8	1000	10	8	14	1110	16	E
3	0011	3	3	9	1001	11	9	15	1111	17	F
4	0100	4	4	10	1010	12	A	16	10000	20	10
5	0101	5	5	11	1011	13	B				

1.1.3 十进制数和非十进制数之间的转换

1. 非十进制数转换为十进制数

任何一个数都可用其按权展开式表示为

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

于是可用按权展开式替代法将一个非十进制数转换为十进制数。具体方法是：将一非十进制数按权展开成一多项式，每项是该位数码与相应权值之积，把此多项式中的数码和权用等值十进制数表示，所得结果就是转换后该数的十进制数。这种方法又称为多项式替代法。

【例 1-5】 将二进制数 10101011.011 转换成十进制数。

解： $(10101011.011)_2 = 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-2} + 1 \times 2^{-3}$
 $= 128 + 32 + 8 + 2 + 1 + 0.25 + 0.125$
 $= (171.375)_{10}$

【例 1-6】 将十六进制数 6FB.A8 转换成十进制数。

$$\begin{aligned}\text{解: } (6\text{FB.A8})_{16} &= 6 \times 16^2 + 15 \times 16^1 + 11 \times 16^0 + 10 \times 16^{-1} + 8 \times 16^{-2} \\ &= 1536 + 240 + 11 + 0.625 + 0.03125 = (1814.65625)_{10}\end{aligned}$$

2. 十进制数转换为非十进制数

十进制数转换为非十进制数时，要将其整数部分和小数部分分别转换，再将结果合并为目的数制形式。

(1) 整数部分的转换

整数部分的转换采用基数除法。所谓基数除法，是用目的数制的基数去除十进制整数，第一次除所得余数为目的数的最低位，所得到的商再除以该基数，所得余数为目的数的次低位，依次类推，继续上面的过程，直到商为 0 时，所得余数为目的数的最高位。此法称为除基取余法。

【例 1-7】 将十进制数 179 转换为二进制数

$$\begin{array}{rcl} \text{解:} & 2 \overline{) 179} & \\ & 2 \overline{) 89} \cdots \cdots 1 & \text{最低位 (LSB)} \\ & 2 \overline{) 44} \cdots \cdots 1 & \\ & 2 \overline{) 22} \cdots \cdots 0 & \\ & 2 \overline{) 11} \cdots \cdots 0 & \\ & 2 \overline{) 5} \cdots \cdots 1 & \\ & 2 \overline{) 2} \cdots \cdots 1 & \\ & 2 \overline{) 1} \cdots \cdots 0 & \\ & 0 \cdots \cdots 1 & \text{最高位 (MSB)} \end{array}$$

$$(179)_{10} = (10110011)_2$$

(2) 小数部分的转换

小数部分的转换是采用基数乘法进行的。所谓基数乘法，是用该小数乘目的数制的基数，第一次乘得结果的整数部分为目的数小数的最高位，其小数部分再乘基数，所得结果的整数部分为目的数小数的次最高位，依此类推，继续上面的过程，直到小数部分为 0 或达到要求精度为止。此法称为乘基取整法。

【例 1-8】 将十进制小数 0.6875 转换为二进制数。

$$\begin{array}{rcl} \text{解:} & 0.6875 & 0.75 \\ \times & 2 & \times 2 \\ \hline & 1.3750 \cdots \cdots 1 & \text{高位} \\ \times & 2 & \times 2 \\ \hline & 0.750 \cdots \cdots 0 & \end{array} \quad \begin{array}{rcl} & 1.50 \cdots \cdots 1 & \\ \times & 2 & \\ \hline & 1.0 \cdots \cdots 1 & \text{低位} \end{array}$$

故

$$(0.6875)_{10} = (0.1011)_2$$

【例 1-9】 将十进制数 23.59375 转换成八进制数、二进制数和十六进制数。

解:

$$\begin{array}{r} 8 \overline{) 23} \\ 8 \overline{) 2} \cdots \cdots 7 \\ 0 \cdots \cdots 2 \end{array} \qquad \begin{array}{r} 0.59375 \\ \times \quad 8 \\ \hline 4.75000 \cdots \cdots 4 \\ \times \quad 8 \\ \hline 6.00 \cdots \cdots 6 \end{array}$$

故

$$(23.59375)_{10} = (27.46)_8$$

又

$$(27.46)_8 = (10111.10011)_2 = (17.98)_{16}$$

因此

$$(23.59375)_{10} = (10111.10011)_2$$

$$(23.59375)_{10} = (17.98)_{16}$$

1.2 二 - 十进制编码

所谓编码,就是用若干位特定二进制数码来表示自然数、字母和符号的过程。这些特定二进制数码称为字符的代码。例如 1101 二进制数码是等值十进制数 13 的自然二进制代码。

在计算机中,十进制数除了换算成二进制数外,还可以直接用十进制数进行输入和运算,这种方法就是将十进制数的十个数码,分别用不少于 4 位的特定二进制数码来表示,这种表示方法称为十进制数的二进制编码,简称二 - 十进制代码。这种形式的编码既有二进制的形式,又有十进制的特点,并便于数字设备的识别与转换。

四位二进制数有 0000~1111 十六种组合,要从这十六种组合中选出十种组合作为 0~9 的代码,其方案很多,下面仅介绍几种二 - 十进制代码。

1.2.1 几种二 - 十进制编码

1. 8421BCD 码

8421BCD 码简称 BCD 码,是最常用的二 - 十进制代码。它是用 0000~1001 来分别表示十进制的 0~9。其余六种 1010~1111 是未使用的代码,称为伪码。代码的各位有固定的权,四位二进制数从左到右的权分别为 8、4、2、1。故它是一种有权码,因而根据代码的组成便可知它所代表的十进制数。设 8421BCD 码的各位为 $a_3a_2a_1a_0$,则它所代表的值为

$$N = 8a_3 + 4a_2 + 2a_1 + 1a_0$$

2. 2421 码

2421 码也是一种有权码，四位二进制数从左到右的权分别为 2、4、2、1。由于它有两位的权均为 2，故它的编码方案不止一种，在此仅介绍一种对 9 的自补码，所谓对 9 的自补码，就是 4 的 2421 码为 0100，它正好是 5 的 2421 码 1011 按位取反而得。0 的 2421 码为 0000，按位取反，正好是 9 的 2421 码 1111。这种 2421 码，见表 1-3。2421 码所代表的值为

$$N = 2a_3 + 4a_2 + 2a_1 + 1a_0$$

2421 码的伪码是 0101~1010。

3. 余 3 码

余 3 码也是一种被广泛采用的二-十进制代码。对应于同样的十进制数字，余 3 码比相应的 8421 码多 0011，所以称为余 3 码。如表 1-3 所示。

余 3 码的伪码是 0000、0001、0010、1101、1110、1111。

表 1-3 常用二-十进制代码

十进制数	8421 码	2421 码	余 3 码
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

一个 n 位十进制数，需要用 n 个二-十进制代码来表示。

【例 1-10】 试分别用 8421BCD 码、2421 码和余 3 码表示 $(68.73)_{10}$ 。

解： $(68.73)_{10} = [01101000.01110011]_{\text{BCD}}$
 $(68.73)_{10} = [11001110.11010011]_{2421}$
 $(68.73)_{10} = [10011011.10100110]_{\text{余3码}}$

1.2.2 二-十进制代码的加法

二-十进制代码的加法原则上与二进制数制的情况一样。但也有所不同，四位二进制数的加法是逢 16 进 1，而二-十进制代码的加法是逢 10 进 1。当和数未处