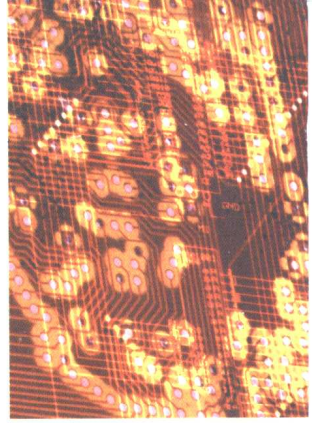
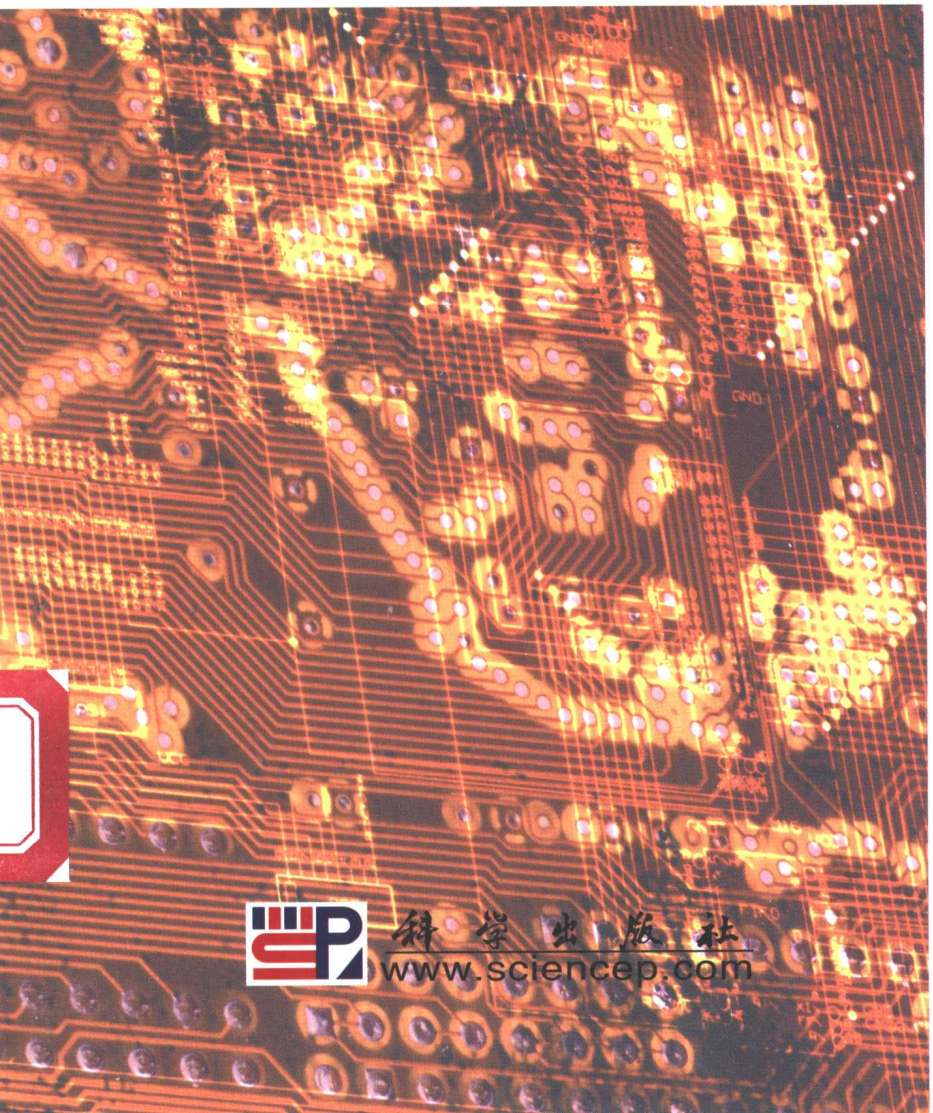
现代超大规模集成电路设计丛书



VLSI数字信号处理
——设计与实现

张 欣 编著



科学出版社
www.sciencep.com

现代超大规模集成电路设计丛书

VLSI 数字信号处理

——设计与实现

张　欣　编著

科学出版社

北　京

内 容 简 介

本书比较详细地讨论了超大规模集成电路(VLSI)上数字信号处理的基础理论知识和实现方法。全书共分十章,内容包括:VLSI 基本算术运算单元的设计,VLSI 并行处理结构和微流水线设计技术,并行 FIR 滤波器,数论变换,多速率采样处理技术,FFT 和 DCT/IDCT 变换,Galois 域上的算术运算等。书中给出了大量用 VHDL 或 Verilog 语言来描述其算法实现的例子。

本书可供从事 VLSI(或 FPGA)芯片数字信号处理算法映射工作的研究和开发人员参考,还可以作为电子工程、计算机科学与工程等专业的研究生和高年级本科生的参考教材。

图书在版编目(CIP)数据

VLSI 数字信号处理设计与实现/张欣编著 .—北京:科学出版社,2003

(现代超大规模集成电路设计丛书)

ISBN 7-03-011300-4

Ⅰ.V… Ⅱ.张… Ⅲ.超大规模集成电路-数字信号-信号处理 Ⅳ.TN47

中国版本图书馆 CIP 数据核字(2003)第 021268 号

责任编辑:李 宁 钟 诣/责任校对:朱光光

责任印制:刘秀平/封面设计:陈 敬

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003 年 7 月第 一 版 开本:B5(720×1000)

2003 年 7 月第一次印刷 印张:14

印数:1—3 000 字数:270 000

定价:30.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

前　言

自从 1959 年 Robert Noyce 和 Jack Kelby 各自独立发明了平面集成电路以来，全世界的技术和经济发生了根本性的改观和变革。40 多年来，集成电路本身已发生了惊人的变化，它历经了小规模(SSI)、中规模(MSI)、大规模(LSI)、超大规模(VLSI)等发展阶段，目前正步入系统芯片(SOC)的成长阶段；集成电路的设计工艺尺寸也已经从微米、亚微米发展到深亚微米。Mead 和 Conway 等人在 VLSI 设计方法学方面开展的具有远见的卓越的实际工作，使得我们学习集成电路设计的基本原理变得非常容易，设计人员在经过不太长时间的学习后，就能设计 VLSI。

VLSI 数字信号处理在集成电路的高端设计中有着十分重要的地位，随着 VLSI 设计方法学的不断发展，人们将会更加注意研究数字信号处理的算法到 VLSI 芯片体系结构的映射方法学等问题。当然，把经典的数字信号处理算法映射到 VLSI 芯片中并非是一件很容易的事情，目前，也还没有一门很普遍适用的设计方法学。本书所介绍的内容主要是基于 VLSI(或 FPGAs)芯片上的数字信号处理技术，希望对提高我国 IC 设计工程师的芯片设计创新能力能有一定帮助。

本书讨论的问题很多，涉及的知识面也比较广泛，我们介绍了一般数字信号处理理论教科书中很少讲述的诸如剩余数系统、分布式算法、并行 FIR 数字滤波器以及数论变换和多速率采样变换等知识，这些知识对于目前在校的电子工程类专业的本科生或研究生未来成为合格的集成电路设计工程师相当地重要。我们还认为，诸如如何设计数据通道里高速乘法器和加法器等算术运算单元的问题，对于一般的 IC(或 FPGA)设计工程师而言，也是必须掌握的基本技能，为此，本书将专门用一章来进行介绍。但在本书中，我们没有用专门的章节来系统地对算法至 VLSI 芯片体系结构映射的方法进行深入的介绍和讨论，尽管这个问题是集成电路设计中极其重要的研究课题之一。即便如此，本书对于想深入从事集成电路设计领域研究的人来说，仍然具有很强的实际的指导意义和实用性。由于数字信号处理工程师大都不是数学家，所以我们在书中采用描述性方法来介绍相关的数学理论基础知识，而不是给出其严格意义上的数学推理和论证。

全书共分十章，各章自成系统，同时又相互联系。书中的近 100 幅插图和用 VHDL 或 Verilog 硬件描述语言描述其算法实现的例子，为读者提供了较多的设计框架和实现方法；同时，本书还给出了部分设计例子的综合结果、仿真和少量研究实例可供读者研究。在每个章节中，我们还提供了大量的参考文献，为想进一步深入理解某些算法的读者提供了查阅文献的线索。

本书可供从事集成电路设计或用 FPGA 芯片开发专用算法的工程师使用，也

可作为电子工程、计算机科学与工程等专业高年级本科生、硕士研究生的教学参考书，同时它还为国内从事数字信号处理研究的科研人员提供某些深入研究课题的素材。

在编写本书的过程中，我得到很多朋友的关心支持和帮助，并得到很多有益的建议。特别要提及的是：电子科技大学的博士研究生刘震昆、硕士生李晓林、黄庆波参与了文献资料的整理工作；书中的部分插图由袁结全和汪涌两位共同制作。在本书将要出版之际，龙军和迟伟两位先生给予了我极大的帮助和鼓励，在此一并表示最热忱的感谢。

还要感谢我的妻子廖琪女士，感谢她能够长期忍受我为查阅文献牺牲大量的业余时间，感谢她对我编写此书的鼓励与支持。没有她的无私奉献，编著此书简直不可想象。

本书内容相当部分来自作者长期参与众多公司和研究所的研究课题以及开设VLSI数字信号处理设计知识讲座所积累的经验、体会和资料。作为目前国内第一部比较详细地介绍关于VLSI数字信号处理的专业书籍，作者恳切地希望读者对于书中的不妥及错误之处予以批评指正。您可以通过电子邮箱 zhx98@mail.sc.cninfo.net 和 xinzhang@ieee.org 与我们联系。

张　欣

2002年11月于成都

目　录

第一章 绪 论

1.1 引 言

1965 年，Cooley-Tukey 提出了快速傅里叶变换作为数字信号处理(DSP)这一领域的开端。到现在，它已经成长为一门独立的数字信号处理学科，且应用领域十分广泛，诸如雷达、卫星导航、声纳、数字通信、生物医学工程、语音通信、图像处理、多媒体计算机等。数字信号的理论基础涉及高等代数、初等数论、近世代数、随机过程和概率统计等众多的数学知识，而在其自身学科的发展中，它又与最优控制、通信理论、人工智能、模式识别、神经网络等学科紧密结合。在其算法的实现上，大体可以分为软件实现方法和硬件实现方法两种。软件实现方法既可以采用基于通用的微型计算机选择一种语言来编制算法的软件包，也可以采用基于某种可编程 DSP 单芯片方案来实现，如 TI 公司的 TMS320CX 系列，AD 公司的 ADSP21X、ADSP210X 系列等。采用硬件实现的方法一般都比采用软件实现方法要困难得多，通常的实现方法是利用某种半导体工艺来设计专门的 DSP 算法的 VLSI 专用芯片，例如 FFT、FIR 等专用集成电路芯片等。近年来，随着 FPGAs 工艺不断的进步，国内外的 DSP 算法研究者和开发者也更多地采用基于 FPGAs 芯片来实现自己专用 DSP 算法的芯片原型，这也是在未来一定时期内用硬件方法来实现 DSP 算法的主要设计技术之一。总的说来，目前采用软件实现 DSP 算法的方法在其运算效率、执行速度等方面都很差，仅仅在其算法的移植性和灵活性方法好些，再者就是开发周期比较短。而采用硬件实现方法，有执行效率好、速度快、集成度高等很多优点，缺点就是研发周期长，难度也比较大。随着动态可配置技术的发展，用 FPGAs 来实现众多复杂 DSP 算法的灵活性已经有了长足进展。

通常，在 VLSI 上实现某种数字信号处理不像用软件来实现那么简单，如何把一个好的 DSP 算法适配到一个合理的 VLSI 体系结构中去，通常也是一件很困难的事，也没有一个通用法则。也就是说，把一个 DSP 算法映射到某个 VLSI 体系结构中去的过程本身就是一种创新。当然，即便有了这个映射过程，还远没有实现 DSP 算法在 VLSI 芯片设计中的性能，同时，还得结合具体工艺来进一步提高算法的性能。在本书中，我们并不想介绍算法到结构的映射方法，而是通过一些 DSP 算法的详细讨论来体现它们在 VLSI 芯片设计中的重要意义。

1.2 本书各章内容简介

本书试图把数字信号处理的基础理论知识与 VLSI(或 FPGAs)芯片设计结合起来,该工作在国外的某些大学以及 IC 设计公司已经有了深入和广泛地研究成果。在众多这方面文献中,我们将有选择性地介绍 VLSI(或 FPGAs)上数字信号处理理论的有关专题内容,每个专题内容既有一定的独立性,又有其知识上的相关性。全书共分 9 章来介绍,以下简单地介绍各章内容。

第二章,介绍计算机算术运算。数字信号处理是建立在一定的数的系统之上的,常见的二进制系统在一般的教科书均可找到。在本章我们将介绍一般读者不太熟悉的剩余数系统(RNS)和分布式算法(DA),同时,对数论基础知识作了简单地介绍,它是后面第五章数论变换的理论基础,以及对理解 Galois 域上的运算也是很有用的。众所周知,设计一个高速的乘法器和加法器对于提高 VLSI 上的 DSP 算法性能是至关重要的,在这一节里,将介绍一个 8×8 位的 Baugh-Wooley 补码阵列乘法器算法和一个二进制流水线加法器。在本章末,给出了一个剩余数算法电路的实现,在关键运算单元中,我们都给出了 VHDL 源代码,供读者研究。

第三章,流水线和 VLSI 并行处理结构。在同步电路的设计中,着重介绍线性流水线原理,以及它在一个实际的数据通道设计中的应用,流水线技术对于提高 VLSI 芯片上的 DSP 算法的局部运算的并行性具有很重要的作用。在异步电路设计中,介绍了微流水线(Micropipelines)技术,它在低功耗、高速 VLSI 芯片设计中占有很重要的位置。Systolic 阵列结构可以更好地解决高度并行性和规则性(大量重复运算)的 DSP 算法在 VLSI 上的实现,本章将结合一个经典的例子——VLSI 矩阵运算处理器来具体讨论它的实现方法。

第四章,数字滤波器。本章主要介绍一类具有线性相位 FIR 数字滤波器的设计,并利用第 2 章介绍的分布式算法来完成 FIR 滤波器的实现。关于并行 FIR 滤波器,可以利用多相分解来实现,基于 Winograd 的方法可以更加快速地实现 FIR 滤波器。

第五章,数论变换。利用第 2 章介绍的数论知识来讨论定义在一维上的数论变换,其中,Fermat 数变换在实际工程设计中很有用。通过一个 32 点 FNT 流水线结构来描述数论变换的具体实现方法。最后,简单地叙述了利用 Fermat 数变换来计算复数卷积。

第六章,多速率信号处理。在图像的放大(或缩小)、数字电视的图像信号以及软件无线电的信号等处理中,采样率的变换方法是十分重要的。本章比较详尽地讨论了各种采样率的变换方法和实现方法,并结合 Hogenauer CIC 滤波器具体讨论了实现方法。在章节末,对滤波器组也作了简单地介绍。

第七章,FFT 变换。主要讨论 Cooley-Tukey、Good-Thomas、Winograd 三种

FFT 算法。其中，给出了 Cooley-Tukey 算法中的复数旋转因子和蝶形运算单元的 VHDL 代码。

第八章，DCT 和 IDCT 变换。在定义 DCT 和 IDCT 的基础上，讨论了一种 DCT 的体系结构实现方法。其中，基于矢量处理方法和分布算法方法分别在 VLSI和 FPGAs 芯片设计中都有广泛的用途。

第九章，信道纠错编码。通过介绍近世代数基础知识，对线性分组码和循环码的编、译方法作了比较详尽的讨论。在 Galois 域上的算术运算实现，对于某些 VLSI芯片设计是十分重要的，给出了 Galois 域上循环码的一种 VHDL 建模方法。该码也是 BCH 码和 RS 码的基础。

第十章，实例研究。主要通过一个具体的 VLSI 芯片设计例子来介绍一些数字信号处理的实现方法，对学习和研究算法到 VLSI 结构映射有一定启发性。

第二章　计算机算术运算及其实现

2.1　引　　言

在计算机算术运算单元里有两个极其重要的基础设计：一个是数的表示，一个是代数运算的实现[1~5]。在本章里，我们首先介绍数的基数表示的一般方法。在介绍剩余数系统的表示方法时，对数论知识作了简略性地介绍，其目的是为了使读者更好地理解剩余数的表示方法，同时，也为本书第五章介绍数论快速变换打下基础。在二进制加法器和乘法器中，着重介绍在 VLSI 设计中用途比较广泛的流水线加法器和补码阵列乘法器的设计。为了便于读者更好地理解阵列乘法器，我们还将给出一个 8×8 位的 Baugh-Wooley 补码阵列乘法器的 VHDL 源代码。由于分布式算法(DA)在数字图像处理中的 DCT/IDCT 有很重要的应用，我们也作了比较深入的讨论。本章末，我们还将对 SD 剩余数算术电路的逻辑实现进行深入地研究。

2.2　算术运算的数的系统

在数字信号处理中，算术运算算法的实现，在很大程度上取决于数值数据在存储器或寄存器里的不同表示方法。由于实现有限精度的算术运算，所有允许的数值表示必须限制在有限的字长范围内。算术运算单元的设计者在解决范围很广的应用问题时，必须注意时间和空间效率结构方面的现实性以及提供充分精度的数值分析时的现实性。在算术运算单元的设计中，一般可分为五类：普通基数、带符号数字、残数(或叫剩余数)、有理数、对数运算数的数的系统。

2.2.1　普通基数的数的系统

在数字信号处理的算术运算单元中，我们以 r 为基数用 $n+k$ 重数字向量来表示数 X，即

$$\boldsymbol{X}=(x_{n-1},\cdots,x_0,x_{-1},\cdots,x_{-k})_r \tag{2.1}$$

其中，每个分量 $x_i(-k\leqslant i\leqslant n-1)$称为向量 $\boldsymbol{X}$ 的第 i 个数字。每位数字可以有 r 个不同的值

$$\{0,1,\cdots,r-1\} \tag{2.2}$$

其中，r 是数的系统的基数。在一个固定基数的数的系统中，所有各位数字都具有相同的基数值。比如，常用的十进制数的表示法里，$r=10$。

混合基数的数表示在不同位数上具有不同的基数值。在具有权的基数的数的系统中，我们把每个数字向量 $\boldsymbol{X}$ 指定一个单值，并表示为

$$\boldsymbol{X}_{\mathrm{v}} = \sum_{i=-k}^{n-1} x_i \omega_i \tag{2.3}$$

其中，每个 ω_i 称为第 i 位数字的加权因子。$n+k$ 个加权因子形成一个权向量，表示为

$$\boldsymbol{W} = (\omega_{n-1}, \cdots, \omega_0, \omega_{-1}, \cdots, \omega_{-k}) \tag{2.4}$$

数 $\boldsymbol{X}_{\mathrm{v}}$ 的值可以用 $\boldsymbol{X} \cdot \boldsymbol{W}$ 求出，即两个向量的点积。我们可以从权向量 $\boldsymbol{W} = (r^{n-1}, \cdots, r^0, r^{-1}, \cdots, r^{-k})$ 引出一个基数为 r 的数 $\boldsymbol{X}$ 的常用记数表示法，其值为

$$\boldsymbol{X}_{\mathrm{v}} = \boldsymbol{X} \cdot \boldsymbol{W} = \sum_{i=-k}^{n-1} x_i \cdot \omega_i = \sum_{i=-k}^{n-1} x_i \cdot r^i \tag{2.4'}$$

我们经常使用的有四种基数系统，即分别是基数值 $r = 2、8、10、16$ 的二进制、八进制、十进制、十六进制的数系统。通常，当基数愈高时，至少需要 $k = \lceil m \rceil (m = \log_2 r)$ 位二进制位来对基数每个数字进行编码。其中，符号 $\lceil x \rceil$ 表示不小于实数 x 的最小整数。

2.2.2 带符号数字的数的系统

带符号数字(signed-digit, SD)的数的表示法可以有冗余性，在设计高速算术运算时显得很有用。比如，它可以用在并行加法器、再编码乘法器、高基数除法、以及剩余数算术运算等单元电路的设计中。

SD 数的定义如下：

给定一个基数 r，则一个 SD 数的每一个数字可具有以下 $2\alpha + 1$ 个值：

$$\sum\nolimits_r = \{-\alpha, \cdots, -1, 0, 1, \cdots, \alpha\} \tag{2.5}$$

其中最大数字的数值 α 必须在以下范围内：

$$\left\lceil \frac{r-1}{2} \right\rceil \leqslant \alpha \leqslant r - 1 \tag{2.6}$$

为了在对称的数字集合 $\sum_r$ 中得到最小的冗余，可以选择 $\alpha = \left\lfloor \frac{r}{2} \right\rfloor$ 作为最大数值。其中，符号 $\lfloor x \rfloor$ 表示小于或等于实数 x 的最大整数。所以，如果 r_{e} 是偶数，则有 $\alpha = \frac{r_{\mathrm{e}}}{2}$，如果 r_0 是奇数，则有 $\alpha = \frac{r_0 - 1}{2}$。也就是说，相邻奇偶基数可以产生相同的数字集合。相应于这种选择的数字集合可以表示为以下两种不同方式，但在 $r_0 = r_{\mathrm{e}} + 1$ 时，两者是相同的集合

$$\sum\nolimits_{r_0} = \left\{-\frac{r_0 - 1}{2}, \cdots, -1, 0, 1, \cdots, \frac{r_0 - 1}{2}\right\}$$

$$\sum\nolimits_{r_{\mathrm{e}}} = \left\{-\frac{r_{\mathrm{e}}}{2}, \cdots, -1, 0, 1, \cdots, \frac{r_{\mathrm{e}}}{2}\right\} \tag{2.7}$$

例如，基数为 2 的 SD 系统具有数字集合 $\sum_2 = \{-1, 0, 1\}$。我们可以推出一个 SD 数

$$Y = (y_{n-1} \cdots y_0 y_{-1} \cdots y_k)_r \tag{2.8}$$

的代数值 Y_v 为

$$Y_v = \sum_{i=-k}^{n-1} y_i \cdot r^i \tag{2.9}$$

该式与前面的式(2.4)有些相似，只不过现在的 Y 本身可以是正的或是负的，而无需要一个显示符号。基数为 2 的 SD 系统可以认为具有一个数字集合 $\{\bar{1}, 0, 1\}$。这里 -1 用 $\bar{1}$ 来表示，假如我们要用 SD 表示法来表示 -3 这个数值，就有五种方法：

$$\begin{aligned} Y &= (00\,\bar{1}\,\bar{1})_2 = -2-1 = -3 \\ &= (0\,\bar{1}01)_2 = -4+1 = -3 \\ &= (\bar{1}101)_2 = -8+4+1 = -3 \\ &= (0\,\bar{1}1\,\bar{1})_2 = -4+2-1 = -3 \\ &= (\bar{1}11\,\bar{1})_2 = -8+4+2-1 = -3 \end{aligned}$$

在一个具有 n 个数字、其值为 Y_v 的 SD 向量中的非零数字的数目被称为权 $\omega(n, Y_v)$。一般地，一个二进制的 n 位数字的 SD 向量，它的权定义如下：

$$\omega(n, Y_v) = \sum_{i=0}^{n-1} |y_i| \tag{2.10}$$

其中，当 $y_i \neq 0$ 时，$|y_i| = 1$。对于给定的 n 值和 Y_v 值，具有最小权的 SD 向量叫做最小 SD 表示法。在上例中，只有这两个向量 $(00\,\bar{1}\,\bar{1})_2$ 和 $(0\,\bar{1}01)_2$ 才是 SD 的最小向量表示法。最小 SD 表示法在乘法再编码方面很有意义。

2.2.3 定点数的表示法

通常，具有 n 个数字位置的数，最左边的数字保留作为符号指示位。我们考虑基数为 r 的数

$$\aleph = (a_{n-1} a_{n-2} \cdots a_1 a_0)_r \tag{2.11}$$

其中，符号数字 a_{n-1} 所取的值为

$$a_{n-1} = \begin{cases} 0 & \text{当 } \aleph \geqslant 0 \\ r-1 & \text{当 } \aleph < 0 \end{cases} \tag{2.12}$$

在 $\aleph$ 中其余的数字表示 $\aleph$ 的真正数值，或者是它的补值中的一种。由于 $\aleph$ 是一个位置表示的数，我们可以选择一个固定位置来放置一个基数点把整数部分和分数部分区分开来，那么我们称这个 $\aleph$ 表示法为定点数。

对于正的定点数，符号位 $a_{n-1} = 0$，其余的数字 $a_{n-2}, \cdots, a_1, a_0$ 表示真正数字。其值为

$$|\aleph| = \sum_{i=0}^{n-1} a_i \cdot r^i \tag{2.13}$$

负的定点数有三种不同的表示法,即

① 带符号数字表示法

$$\overline{\aleph} = ((r-1)a_{n-2}\cdots a_1 a_0)_r \tag{2.14}$$

其中,对于 $n-2 \geqslant i \geqslant 0$ 的 a_i 是真正的数值位,$r-1$ 位为符号位。

② 基数反码表示法

$$\overline{\aleph} = ((r-1)\bar{m}_{n-2}\cdots \bar{m}_1 \bar{m}_0)_r \tag{2.15}$$

其中,对于 $n-2 \geqslant i \geqslant 0$ 的 $\bar{m} = (r-1) - m_i$。在此表示法中,有 $\overline{\aleph} = r^n - 1 - \aleph$。它也称为 $r-1$ 的补码表示法。

③ 基数补码表示法

$$\overline{\aleph} = (((r-1)\bar{m}_{n-2}\cdots \bar{m}_1 \bar{m}_0) + 1)_r \tag{2.16}$$

在此表示法中,有 $\overline{\aleph} = r^n - \aleph$。我们也称它为 r 的补码表示法。

2.2.4 剩余数系统

本小节我们来讨论剩余数系统(residue number system, RNS),也有人把它称为残数系统。我们在介绍初等数论时,只引出相关的定义,并不加证明地给出一些有用的定理。欲认真研究的读者,可自行阅读有关初等数论方面的书籍。

1. 初等数论基础知识

定理 1 设 a、b 是两个整数,其中 $b>0$,则存在两个惟一的整数 q 和 r,使得

$$a = bq + r \qquad (0 \leqslant r < b) \tag{2.17}$$

成立。

在式(2.17)中,我们把其中的 q 叫做 a 被 b 除时得出的不完全商,r 叫做 a 被 b 除时所得到的余数(r 也可称为整数 a 模 b 的最小非负剩余),也叫非负最小剩余,通常记为 $\langle a \rangle_b = r$。在不引起混淆的情况下,$\langle a \rangle_b$ 中的 b 常略去不写。

定理 2 对于整数 a_1、a_2、b,其中 $b>0$,有

$$\langle a_1 + a_2 \rangle_b = \langle \langle a_1 \rangle_b + \langle a_2 \rangle_b \rangle_b \tag{2.18}$$

$$\langle a_1 - a_2 \rangle_b = \langle \langle a_1 \rangle_b - \langle a_2 \rangle_b \rangle_b \tag{2.19}$$

$$\langle a_1 a_2 \rangle_b = \langle \langle a_1 \rangle_b \times \langle a_2 \rangle_b \rangle_b \tag{2.20}$$

定义 1 设 p 是一个素数,形如 $2^p - 1$ 的数叫做梅森数(Mersenne number),并记作 $M_p = 2^p - 1$。

寻找素数 p,使得梅森数 M_p 是素数,仍是近代数论研究的课题之一,不过,它在代数编码方面得到了广泛的应用。

定义 2 我们把形如 $F_n = 2^{2^n} + 1 (n \geqslant 0)$ 的数叫做费马数(Fermat number)。

定理 3 任给两个费马数 F_m、F_n,$m \neq n$,则 $(F_m, F_n) = 1$。

费马数跟现代数字信号处理理论有紧密的联系。例如，用费马数给出的数论变换，可用来计算整数序列的卷积。我们将在第五章中加以介绍。

定义 3 设 n 是一个正整数，若 n 的全部因数的和等于 $2n$，那么我们把 n 叫做完全数。例如，$56=1+2+4+7+14+28$，故 56 是一个完全数。

定义 4 给定一个正整数，把它叫做模。如果用 m 去除任意两个整数 a 与 b 所得的余数相同，我们就称 a、b 对模 m 同余。并记作

$$a \equiv b(\mathrm{mod}\, m)$$

由上面的定义，我们可以得到以下性质：

① $a \equiv a(\mathrm{mod}\, m)$

② 若 $a \equiv b(\mathrm{mod}\, m)$，则 $b \equiv a(\mathrm{mod}\, m)$

③ 若 $a \equiv b(\mathrm{mod}\, m)$，$b \equiv c(\mathrm{mod}\, m)$，则 $a \equiv c(\mathrm{mod}\, m)$

④ 若 $a_1 \equiv b_1(\mathrm{mod}\, m)$，$a_2 \equiv b_2(\mathrm{mod}\, m)$，则 $a_1 + a_2 \equiv b_1 + b_2(\mathrm{mod}\, m)$

⑤ 若 $a_1 \equiv b_1(\mathrm{mod}\, m)$，$a_2 \equiv b_2(\mathrm{mod}\, m)$，则 $a_1 a_2 \equiv b_1 b_2(\mathrm{mod}\, m)$

2. 剩余数系统

我们已经定义了同余的概念，现在就可以对任意给定的一个正整数 m，利用模 m 同余这个关系，把全体整数分成若干类。

定义 5 设 m 是一个给定的正整数，$C_r(r=0、1、\cdots、m-1)$表示所有形如 $qm+r$ 的整数组成的集，其中 $q=0、\pm1、\pm2、\cdots$，则 $C_0、C_1、\cdots、C_{m-1}$叫做模 m 的剩余类。

定义 6 在模 m 的剩余类 $C_0、C_1、\cdots、C_{m-1}$ 中各取一数 $a_j \in C_j$，此 m 个数 $a_0、a_1、\cdots、a_{m-1}$ 称为模 m 的一组完全剩余系。我们从每一类中取出的最小的非负整数，如 $\{0,1,\cdots,m-1\}$ 称为非负最小完全剩余系，并记作 Z_m。在以后的各章节中，Z_m 均指

$$Z_m = \{0,1,2,\cdots,m-1\}$$

根据定义 4 里的性质，我们很容易验证 Z_m 是一个数环，一般称 Z_m 为模的整数环。设 $a \in Z_m(a \neq 0)$，若存在 $b \in Z_m(b \neq 0)$，使得 $ab=1$，则称 b 为 a 的逆元，并记作 a^{-1}（显然，a 也是 b 的逆元）。我们可以证明这样的结论：当 m 为素数时，在 Z_m 中的每一个非零元素都存在其逆元。故在当 m 为素数时，Z_m 被称为域。例如，设 $m=7$，$Z_m=\{0,1,2,3,4,5,6\}$。由于 $2\times4\equiv1(\mathrm{mod}7)$，$3\times5\equiv1(\mathrm{mod}7)$，$6\times6\equiv1(\mathrm{mod}7)$，所以有，2 的逆元是 4，3 的逆元是 5，6 的逆元是 6，4 的逆元是 2 等。而当 m 不为素数时，例如，$m=6$，$Z_m=\{0,1,2,3,4,5\}$，2，3，4 均无逆元，只有 1 和 5 有逆元。

定义 7 设 $a_1、a_2、\cdots、a_k$ 为整数，$m_1、m_2、\cdots、m_k$ 为正整数，称

$$\begin{aligned} x &\equiv a_1(\mathrm{mod}\, m_1) \\ x &\equiv a_2(\mathrm{mod}\, m_2) \end{aligned} \tag{2.21}$$

$$\cdots\ \cdots\ \cdots\ \cdots$$

$$x \equiv a_k (\mathrm{mod} m_k)$$

为联立一次同余式组。若有一整数 l，使得

$$\begin{aligned} l &\equiv a_1 (\mathrm{mod} m_1) \\ l &\equiv a_2 (\mathrm{mod} m_2) \\ &\cdots\ \cdots\ \cdots\ \cdots \\ l &\equiv a_k (\mathrm{mod} m_k) \end{aligned} \tag{2.22}$$

成立，则称 l 为联立同余式组(2.21)的解(或根)。求同余式组的解，叫做解同余方程。

定理 4（中国剩余定理） 设 $m_1、m_2、\cdots、m_k$ 为两两互素的 k 个正整数，$m = m_1 m_2 \cdots m_k$，$m = m_i M_i (i = 1、2、\cdots、k)$，则同余式组(2.21)有惟一解

$$x \equiv \prod_{i=1}^{k} M_i' M_i a_i (\mathrm{mod} m) \tag{2.23}$$

其中，$M_i' M_i \equiv 1 (\mathrm{mod} m_i) (i = 1、2、\cdots、k)$。在求解同余式组时，要注意，应首先求出 m 来，再求出衍数 $M_i = \frac{m}{m_i} (i = 1、2、\cdots、k)$，通过求解同余方程 $M_i' M_i \equiv 1 (\mathrm{mod} m_i) (i = 1、2、\cdots、k)$而得到 M_i'，最后利用式(2.23)求得式(2.21)的解。由于这种解法由我国古代的孙子所发明，故又称它为孙子定理。该定理是初等数论中很重要的定理之一，它有很多用途。

下面我们来讨论整数的剩余表示，它有很重要的应用场合。在本章的实例研究中将会具体介绍它的一个应用。

定义 8 设 $m_1 > 0, \cdots, m_k > 0, (m_i, m_j) = 1, 0 < i < j \leqslant k$，一个整数 x 对于模 $m_1、m_2、\cdots、m_k$ 的剩余表示是指序列$(\langle x \rangle_{m_1}, \langle x \rangle_{m_2}, \cdots, \langle x \rangle_{m_k})$，并记作

$$x \leftrightarrow (\langle x \rangle_{m_1}, \langle x \rangle_{m_2}, \cdots, \langle x \rangle_{m_k})$$

例如，有 $m_1 = 2, m_2 = 3, m_3 = 5$，则 22 的剩余表示为(0, 1, 2)。很显然，一个数的剩余表示是惟一的，但反过来可以有很多数具有同一个剩余表示。例如，所有形如 $30t + 22$ 的整数，其剩余表示均为(0, 1, 2)。为了避免此种情况发生，我们有如下定理：

定理 5 设 $m_1 > 0, \cdots, m_k > 0, (m_i, m_j) = 1, 0 < i < j \leqslant k$，两个整数 $x、x'$ 对于模 $m_1、m_2、\cdots、m_k$ 的剩余表示相同的充分必要条件是 $x \equiv x' (\mathrm{mod} M)$，其中，$M = \prod_{l=1}^{k} m_l$。如果我们限定 $0 \leqslant x < M = \prod_{l}^{k} m_l$，那么，不同的整数 x 对于模 $m_1、m_2、\cdots、m_k$ 的剩余表示，也是不同的。有兴趣的读者可以验证一下，当 $m_1 = 2, m_2 = 3, m_3 = 5$ 时，$M = 2 \times 3 \times 5 = 30$，即整数 0 到 29 的剩余表示就不会有相同的。