

基于FPGA的FFT 处理系统的研究与应用

杨 军 丁洪伟 著

赵东风 主审



科学出版社

基于 FPGA 的 FFT 处理系统的研究与应用

杨 军 丁洪伟 著
赵东风 主审

云南大学“211 工程”项目资助出版

科学出版社
北 京

内 容 简 介

本书主要介绍基于 FPGA 的 FFT 处理系统的设计原理、开发方法和技术细节,总结了作者多年工程项目的开发经验。全书共 7 章,第 1~2 章为基础知识部分,介绍 FFT 算法原理和项目开发环境;第 3~7 章为应用实例部分,分别详细讲解了基于 FPGA 的一维单核 FFT 处理器、多核 FFT 处理器的设计方法,利用一维 FFT 设计基于 FPGA 的 OFDM 系统基带数据传输部分,基于 SOPC 实现二维 FFT 处理器,利用二维 FFT 实现基于 FPGA 的盲图像复原。书中硬件和软件设计分别采用了硬件描述语言和 C 语言,读者需要具有一定的编程开发基础。本书配有实例源代码,读者可从 www.sciencep.com 下载。

本书适合高等院校计算机科学与技术、电子工程、通信工程等相关专业高年级本科生、研究生学习,也可供从事 FPGA 设计与开发的科研人员参考。

图书在版编目(CIP)数据

基于 FPGA 的 FFT 处理系统的研究与应用/杨军,丁洪伟著. —北京:科学出版社,2012

ISBN 978-7-03-035590-4

I. ①基… II. ①杨…②丁… III. ①傅里叶变换-研究 IV. ①O174.22

中国版本图书馆 CIP 数据核字(2012)第 222112 号

责任编辑:张 濮 马晓晓 / 责任校对:林青梅

责任印制:张 倩 / 封面设计:迷底书装

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>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2012 年 10 月第 一 版 开本:B5(720×1000)

2012 年 10 月第一次印刷 印张:15 1/2

字数:312 000

定价:60.00 元

(如有印装质量问题,我社负责调换)

前 言

在数字信号处理中,FFT 算法有着举足轻重的地位。许多算法,如相关、滤波、谱估计、卷积等都可以用 FFT 算法来实现,在雷达、观测、跟踪、高速图像处理、保密无线通信、数字通信、生物工程、图像处理等领域,FFT 算法得到了广泛的应用。FPGA 融合了 DSP 和 ASIC 的优点,具有可配置性强、速度快、密度高、功耗低的特点,其资源丰富,易于实现流水和并行结构。使用 FPGA 实现 FFT 算法,可以提高处理速度,并且具有灵活性高,开发费用低、开发周期短、升级简单、易编程等特点。用 FPGA 实现的 FFT 处理器必将在数字信号处理领域得到更广泛的应用。

作者根据国内外采用 FPGA 开发实现 FFT 处理器的现状,结合当前 FFT 处理器的应用需求,以实用为基本原则,立足于工程实践,详细讲解了基于 FPGA 的 FFT 处理系统的设计技术细节、实际应用开发方法与思路,总结了作者实践项目的开发经验、技巧和开发过程中遇到的问题。全书内容从 FFT 算法的基本原理到 FFT 核心技术的 FPGA 实现,从简单一维 FFT 系统过渡到复杂二维 FFT 系统,从单核 FFT 系统的设计到多核 FFT 系统的实现,以及 FFT 应用实例的开发,以循序渐进的方式带领读者深入学习和研究,帮助读者快速掌握并提高使用 FPGA 技术实现 FFT 算法的能力。

本书系统设计所使用的基本软件工具包括:

Quartus II——用于完成硬件描述语言代码的编写、综合、硬件优化、适配、编程下载和硬件系统调试等;

SOPC Builder——Altera Nios II 嵌入式处理器开发软件包,用于实现 Nios II 系统的配置、生成;

ModelSim——用于对 SOPC 生成的 Nios II 系统的 HDL 描述进行系统功能仿真;

MATLAB——用于对处理结果进行仿真,与 DSP Builder 联合在 Simulink 下开发模型文件,并将其转换为 VHDL 硬件描述语言;

Nios II IDE——用于进行软件开发、调试和向目标开发板进行 Flash 下载。

全书共分 7 章。前两章为基础知识部分,详细介绍了 FFT 算法原理,对项目开发所需的各个工具软件进行了简单的介绍,以帮助读者快速熟悉开发环境。第 3~7 章为实战应用部分,从 FPGA 实现单核 FFT 处理器的三种方法开始,过渡到双核 FFT 处理器的 FPGA 实现;然后以一个单核 FFT 的经典应用实例——基于

FPGA的OFDM系统基带数据传输部分的设计与实现来结束一维FFT处理器的讲解;接着详细讲述了2维FFT处理器的SOPC设计与实现;最后以一个二维FFT在图像处理领域的应用实例——基于FPGA的盲图像复原来结束全书。在实例讲解过程中,详细介绍了FPGA项目开发的技术和技巧,深入地讲解了开发方案与设计思路,并对设计过程中的重点和难点进行了详细讲解和注释。实例具有代表性,有助于读者巩固知识点、掌握利用FPGA实现FFT处理系统的技术。

本书语言简洁,结构清晰,内容系统全面,注重基础知识和实际工程结合。在实例的讲解上,既介绍了设计原理、结构框图、基本步骤和流程,提供了开发方案和设计思路,也穿插了一些经验技巧和注意事项,以同时提高读者的理论知识和实践能力。

本书第2、4、6、7章由杨军教授编写,第1、3、5章由丁洪伟副教授编写,王小军、赵嘎、陈成参与了相关编写工作。作为本书的主审,赵东风教授在本书的框架设计、内容安排和材料组织等方面给予了极大的帮助和支持,并提出了很多宝贵意见。另外,为本书的顺利完成做出贡献的人员还有舒平平、张伟平、李剑、孔延兵、董寅、张凯、杜琛、皇甫正茂、刘明清、周庆芳,他们在资料的收集、整理,源代码的设计、分析、仿真,硬件平台的验证,书稿的录入、排版、绘图等方面做了大量的工作。在此一并向他们表示最诚挚的谢意!

本书涉及的知识范围广,硬件设计和软件设计中分别采用了硬件描述语言和C语言,读者需要具有一定的硬件语言编程基础(Verilog HDL或VHDL)、C语言编程能力和FPGA开发基础。本书适合高等院校计算机科学与技术、电子工程、通信工程等相关专业的本科高年级学生、研究生学习、也可供从事FPGA开发与应用的科研人员参考。

由于作者水平有限,加之编写时间仓促,书中难免有错误和不足之处,诚望读者批评指正。

作 者

2012年4月

目 录

前言

第 1 章 FFT 算法原理与实现	1
1.1 引言	1
1.2 FFT 算法原理	2
1.2.1 DFT 算法简介	2
1.2.2 离散傅里叶变换的性质	5
1.2.3 频率域抽样理论	9
1.2.4 DFT 的应用举例	11
1.2.5 FFT/IFFT 算法原理	18
1.2.6 进一步减少运算量的措施	30
1.2.7 二维 FFT/IFFT 算法原理	31
1.2.8 二维傅里叶变换的基本性质	32
1.3 FFT 硬件实现现状	34
第 2 章 项目开发环境介绍	36
2.1 软件平台	36
2.1.1 硬件开发工具 Quartus II 8.0	36
2.1.2 仿真工具 ModelSim	40
2.1.3 集成开发环境 Nios II IDE 8.0	45
2.1.4 数值计算与仿真测试工具 MATLAB	49
2.2 硬件平台	52
2.2.1 DE2 平台简介	52
2.2.2 DE2 平台的原理	53
2.2.3 DE2 平台的开发环境	57
2.2.4 DE2 开发板测试说明	58
第 3 章 基于 FPGA 的一维单核 FFT 处理器的设计与实现	60
3.1 基于 CORDIC 算法实现 FFT 处理器	60
3.1.1 CORDIC 算法原理	60
3.1.2 FFT 处理器的设计与实现	62
3.1.3 系统综合与仿真测试	68
3.2 使用 DSP Builder 设计 FFT 处理器	74

3.2.1 DSP Builder 简介	74
3.2.2 FFT 模型的建立	75
3.2.3 FFT 模型的实现	81
3.2.4 系统综合与仿真测试	82
3.3 基于 Altera FFT 兆核函数的设计与实现	84
3.3.1 FFT 兆核函数的设计	84
3.3.2 FFT 兆核函数 8.0 仿真	87
3.4 小结	90
第 4 章 基于 FPGA 的一维多核 FFT 处理器的设计与实现	91
4.1 设计思路与原理	91
4.1.1 设计背景	91
4.1.2 实现途径	92
4.1.3 总体结构设计	94
4.2 详细设计	96
4.2.1 系统工作流程	96
4.2.2 系统时钟分析	107
4.2.3 地址发生器	109
4.2.4 地址流水线	112
4.2.5 辐角发生器	114
4.2.6 RAM 存储器	117
4.2.7 数据转换	121
4.2.8 其他	127
4.3 系统综合与仿真测试	128
4.4 小结	130
第 5 章 一维 FFT 应用实例——基于 FPGA 的 OFDM 系统	
基带数据传输部分的设计与实现	132
5.1 OFDM 技术简介	133
5.1.1 OFDM 技术的发展	134
5.1.2 OFDM 技术的特点	135
5.2 OFDM 系统基本原理	137
5.2.1 OFDM 子载波的正交性	137
5.2.2 基于 DFT 的 OFDM 系统数学模型	138
5.2.3 OFDM 系统的基本模型	139
5.2.4 OFDM 系统的相关技术	141
5.3 OFDM 系统整体设计	146

5.3.1 扰码模块	146
5.3.2 RS 码模块	147
5.3.3 卷积码模块	148
5.3.4 QAM 映射模块	148
5.3.5 IFFT/FFT 模块	149
5.3.6 加循环前缀模块	151
5.3.7 基带成形滤波器模块	151
5.4 OFDM 基带数据传输主要模块的 FPGA 实现	159
5.4.1 FFT IP 核生成与配置	159
5.4.2 信道编解码模块	161
5.4.3 发送端平方根升余弦滚降滤波器的实现	165
5.4.4 交织和解交织模块	166
5.4.5 Viterbi 译码的 FPGA 实现	168
5.4.6 星座映射与解映射	170
5.5 OFDM 系统的仿真与验证	172
5.6 小结	173
第 6 章 基于 SOPC 的二维 FFT 处理器的设计与实现	174
6.1 设计思路与原理	174
6.1.1 原理分析	174
6.1.2 SOPC 简介	175
6.1.3 Avalon 总线 ST 模式	178
6.2 硬件设计	185
6.2.1 总体结构设计	185
6.2.2 硬件系统的 SOPC 设计	186
6.3 软件设计与综合测试	192
6.3.1 系统软件设计	192
6.3.2 系统综合与仿真测试	199
6.4 小结	200
第 7 章 二维 FFT 应用实例——基于 FPGA 的盲图像复原	201
7.1 图像复原数学模型	201
7.1.1 图像退化	202
7.1.2 点扩展函数	203
7.1.3 噪声模型	205
7.1.4 图像的主要组成部分	205
7.1.5 图像复原处理算法要解决的主要问题	206

7.2	图像复原方法	207
7.2.1	常见的图像复原方法	207
7.2.2	算法简介	209
7.2.3	图像复原算法的电路实现技术现状	210
7.3	盲图像复原系统整体设计	215
7.3.1	3×3 方形窗结构	215
7.3.2	中值滤波器模块	216
7.3.3	幂运算模块	217
7.3.4	二维FFT模块	218
7.4	盲图像复原系统的FPGA实现	218
7.4.1	信号输入/输出系统	218
7.4.2	3×3 方形窗模板的实现与仿真	223
7.4.3	中值滤波的实现	223
7.4.4	FPGA 例化双口RAM	226
7.5	仿真与验证	229
7.5.1	中值滤波分析	230
7.5.2	盲图像复原处理	231
附录		232
参考文献		237

第 1 章 FFT 算法原理与实现

1.1 引言

随着计算机技术的快速发展,数字信号处理已深入到各个学科领域。在数字信号处理中,许多算法如相关、滤波、谱估计、卷积等都可转化为离散傅里叶变换(DFT)来实现,从而为离散信号分析在理论上提供了变换工具。但是,DFT 计算量大,实现困难,难以应用于实际。

快速傅里叶变换(FFT)是 1965 年由 J. W. Cooley 和 T. W. Tukey 提出的,其思想是利用系数的对称性和周期性,将大点数的 DFT 分解成若干个小点数的 DFT。采用这种算法能使计算机计算离散傅里叶变换所需要的乘法次数大为减少,特别是被变换的抽样点数 N 越多,FFT 算法计算量的节省就越显著,可以使 DFT 的运算时间缩短几个数量级。快速傅里叶变换已成为一门重要的学科,广泛应用在数字信号处理(DSP)和科学计算领域。特别是二维快速傅里叶变换(2D-FFT)在成像技术的光谱和频域分析中有广泛的应用,如图像数字水印、指纹识别、合成孔径雷达(SAR)的成像处理和医学成像等。随着图像尺寸的增加,能处理大数据量的 2D-FFT 的作用也越来越突出。在合成孔径雷达的成像处理、数字全息成像和医学成像等方面,需要处理的数据量已达到 2048×2048 像素甚至更高。因此,需要采用新的技术和硬件体系结构来实现 FFT,使其满足大数据量、高精度和实时性的要求。

目前 FFT 算法主要有软件、硬件两种实现方法。软件实现以数学函数方式调用,不具备高速性、实时性的特征。硬件实现的主要方案有:通用数字信号处理器(DSP)、专用芯片(ASIC)、现场可编程门阵列(FPGA)。DSP 具有纯软件实现的灵活性,适用于流程复杂的算法,如通信系统中信道的编译码、QAM 映射等算法。但 DSP 运算速度较慢、接口不灵活,而且没有 FFT 运算所需的大容量存储器,需要外置特定的接口、控制芯片和 RAM。DSP 完成 FFT 运算需占用大量 DSP 的运算时间,使整个系统的数据吞吐率降低,同时也无法发挥 DSP 软件实现的灵活性。采用 FFT 专用器件,速度虽能够达到要求,但其外围电路复杂,可扩展性差,成本昂贵。

近几年,随着 FPGA 技术和计算机并行技术的迅速发展,采用速度更快、灵活性更高的 FPGA 来实现具有并行特征的 FFT 算法,已成为国内外研究的热点。

FPGA 融合了 DSP 和 ASIC 的优点,并具有可配置性强、速度快、密度高、功耗低的特点,其资源丰富,易于实现流水和并行结构。使用 FPGA 实现 FFT 算法,可以提高处理速度,并且具有灵活性高,开发费用低、开发周期短、升级简单、易编程等特点。

2010 年 10 月在杭州召开的 CCF 中国计算机大会上,北京理工大学专用处理器研究所所长刘大可教授针对“面向 2015 年的并行多核嵌入式计算平台”这一课题进行了演讲,重点讲解了多核并行专用处理器(主要是并行数字信号处理器)的发展现状和相关技术,并强调该技术“是一个很有前途的分支”。

多核处理器从通用性角度分为通用多核处理器和专用多核处理器两类。全球 90% 的多核处理器的设计面向专用多核处理器,专用多核处理器的未来市场份额约占全部多核市场的 50%。本书涉及的多核 FFT 处理器属于多核并行专用处理器中的并行数字信号处理器(多核流处理器)。

从近年的人才需求市场的情况来看,对 FPGA 电子工程师的需求一路走高,尤其是 SOPC 嵌入式方面的开发人才甚至出现了供不应求的状况。一方面,这是由于工程技术的电子化、集成化和系统化促进了电子工程技术的发展,也促进了 FPGA 技术在社会各行各业中的广泛应用,人才需求不断增加;另一方面,则是因为 FPGA 电子工程师的门槛相对较高,这个高门槛来自于工程师的经验和实践能力。

1.2 FFT 算法原理

1.2.1 DFT 算法简介

有限长序列在数字信号处理中是很重要的一种序列,可以用 z 变换和傅里叶变换来研究它,但是使用离散傅里叶变换(DFT)更能反映“有限长”序列的特点。离散傅里叶变换作为有限长序列的一种傅里叶表示法在理论上相当重要,而且由于存在着计算离散傅里叶变换的有效快速算法,它在各种数字信号处理的算法中起着核心的作用。

有限长序列的离散傅里叶变换(DFT)和周期序列的离散傅里叶级数(DFS)在本质上是一样的。为了更好地理解 DFT,需要先讨论周期序列的 DFS。而为了讨论离散傅里叶级数和离散傅里叶变换,首先来回顾傅里叶变换的几种可能形式。

1. 连续时间、连续频率——连续傅里叶变换

设 $x(t)$ 为连续时间非周期信号,且 $\int_{-\infty}^{\infty} |x(t)| dt < \infty$, 傅里叶变换表示为

$$x(t) \leftrightarrow X(j\Omega)$$

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega$$

图 1.1 所示为连续傅里叶变换示意图,可以看出,时域连续函数在频域是非周期的谱,时域非周期函数在频域是连续的谱。

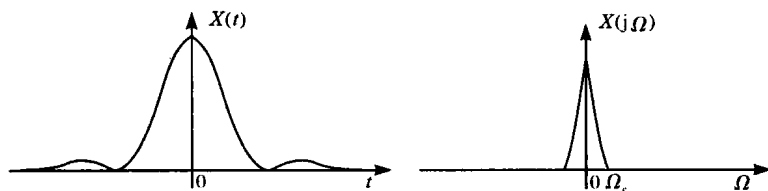


图 1.1 连续傅里叶变换示意图

2. 连续时间、离散频率——傅里叶级数

设 $f(t)$ 代表一个周期为 T_1 的周期性连续时间函数, $f(t)$ 可展开成傅里叶级数,其傅里叶级数的系数为 F_n , $f(t)$ 和 F_n 组成变换对,表示为

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\Omega_1 t}$$

$$F_n = \frac{1}{T_1} \int_{-T_1/2}^{T_1/2} f(t) e^{-jn\Omega_1 t} dt \quad (T_1 = \frac{2\pi}{\Omega_1})$$

图 1.2 所示为傅里叶级数示意图,可以看出,时域连续函数在频域是非周期的谱,而时域周期函数在频域是离散的谱。

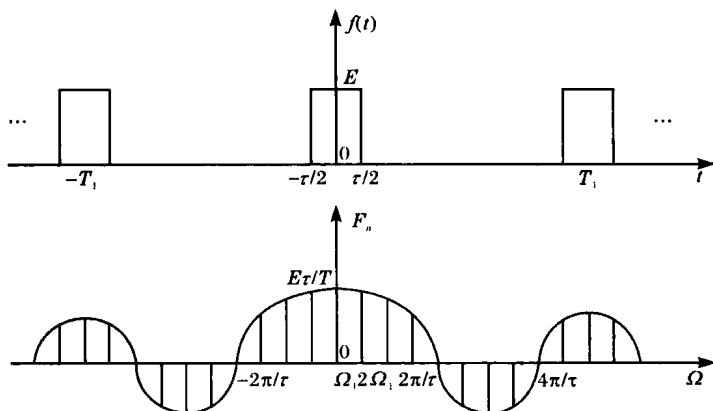


图 1.2 傅里叶级数示意图

3. 离散时间、连续频率——序列的傅里叶变换

$$\text{正变换: DTFT}[x(n)] = X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n}$$

$$\text{逆变换: DTFT}^{-1}[X(e^{j\omega})] = x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$$

$$X(e^{j\omega}) \text{ 级数收敛条件为 } \left| \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \right| = \sum_{n=-\infty}^{\infty} |x(n)| < \infty.$$

图 1.3 所示为序列的傅里叶变换示意图,可以看出,时域离散函数在频域是周期的谱,而时域非周期函数在频域是连续的谱。

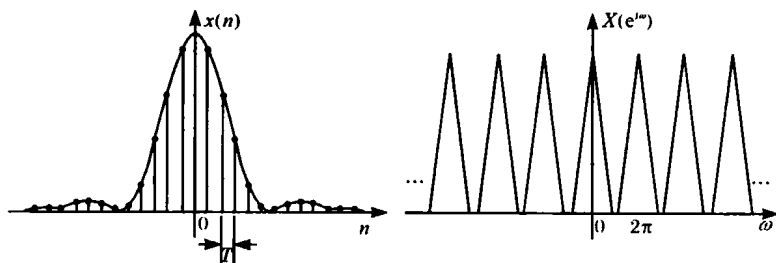


图 1.3 序列的傅里叶变换示意图

4. 离散时间、离散频率——离散傅里叶变换

上面讨论的三种傅里叶变换对,都不适用于在计算机上运算,因为至少在一个域(时域或频域)中,函数是连续的。从数字计算角度,应考虑时域和频域都是离散的情况,这就是下面的离散傅里叶变换。

图 1.4 所示为离散傅里叶变换示意图。其中,时域抽样间隔为 T ,频域周期为 $\Omega_s = 2\pi/T$,时域周期为 T_1 ,频域抽样间隔为 $\Omega_1 = 2\pi/T_1$ 。可以看出,时域离散函数在频域是周期的谱,而时域周期函数在频域是离散的谱。

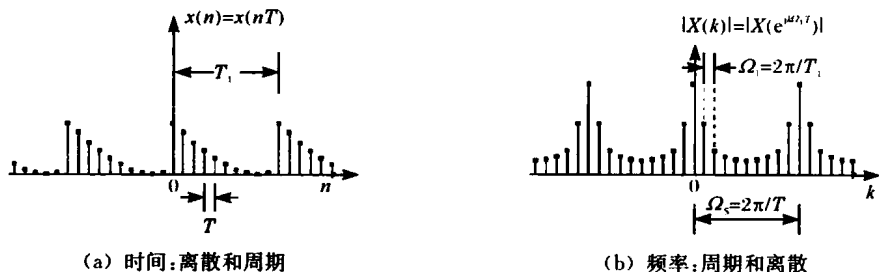


图 1.4 离散傅里叶变换示意图

周期序列实际上只有有限个序列值才有意义,因而它的离散傅里叶级数表达式也适用于有限长序列,这就可以得到有限长序列的傅里叶变换。

设 $x(n)$ 是一个长度为 M 的有限长序列,则

$$\begin{aligned}\text{正变换: } X(k) &= \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \\ &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (k = 0, 1, 2, \dots, N-1)\end{aligned}$$

$$\begin{aligned}\text{逆变换: } x(n) &= \text{IDFT}[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \quad (n = 0, 1, 2, \dots, N-1)\end{aligned}$$

式中, $W_N = e^{-j\frac{2\pi}{N}}$, N 称为 DFT 变换区间长度, $N \geq M$ 。

【例 1.1】 $x(n) = R_4(n)$, 求 $x(n)$ 的 8 点和 16 点 DFT。

解: (1) DFT 变换区间 $N=8$, 则

$$\begin{aligned}X(k) &= \sum_{n=0}^7 x(n) e^{-j\frac{2\pi}{8}kn} = \sum_{n=0}^3 e^{-j\frac{2\pi}{8}kn} \\ &= \frac{1 - e^{-j\frac{2\pi}{8}k \times 4}}{1 - e^{-j\frac{2\pi}{8}k}} = \frac{1 - e^{-jk\pi}}{1 - e^{-j\frac{\pi}{4}k}} \\ &= e^{-j\frac{3\pi}{8}k} \frac{\sin\left(\frac{\pi}{2}k\right)}{\sin\left(\frac{\pi}{8}k\right)} \quad (k = 0, 1, 2, \dots, 7)\end{aligned}$$

(2) DFT 变换区间 $N=16$, 则

$$\begin{aligned}X(k) &= \sum_{n=0}^{15} x(n) e^{-j\frac{2\pi}{16}kn} = \sum_{n=0}^3 e^{-j\frac{2\pi}{16}kn} \\ &= e^{-j\frac{3\pi}{16}k} \frac{\sin\left(\frac{\pi}{4}k\right)}{\sin\left(\frac{\pi}{16}k\right)} \quad (k = 0, 1, \dots, 15)\end{aligned}$$

1.2.2 离散傅里叶变换的性质

1. 线性

设 $x_1(n)$ 、 $x_2(n)$ 是两个有限长序列, 长度分别为 N_1 、 N_2 , 且 $y(n) = ax_1(n) + bx_2(n)$, a, b 为常数。其中 $N = \max[N_1, N_2]$; $x_1(n)$ 为有限长序列, 长度为 N ; $x_2(n)$ 为有限长序列, 长度为 N ; $y(n)$ 有限长序列, 长度为 N 。

$x_1(n)$ 的 N 点 DFT 为

$$X_1(k) = \text{DFT}[x_1(n)] = \sum_{n=0}^{N-1} x_1(n) W_N^{nk} \quad (0 \leq k \leq N-1)$$

$x_2(n)$ 的 N 点 DFT 为

$$X_2(k) = \text{DFT}[x_2(n)] = \sum_{n=0}^{N-1} x_2(n) W_N^{nk} \quad (0 \leq k \leq N-1)$$

$y(n)$ 的 N 点 DFT 为

$$\begin{aligned} Y(k) &= \text{DFT}[y(n)] = \sum_{n=0}^{N-1} y(n) W_N^{nk} = \sum_{n=0}^{N-1} [ax_1(n) + bx_2(n)] W_N^{nk} \\ &= aX_1(k) + bX_2(k) \quad (0 \leq k \leq N-1) \end{aligned}$$

2. 循环移位定理

1) 序列的循环移位

设 $x(n)$ 为有限长序列, 长度为 N , 则 $x(n)$ 的循环移位定义为

$$y(n) = x((n+m))_N R_N(n)$$

上式表明, 先将 $x(n)$ 以 N 为周期进行周期延拓得到序列 $\bar{x}(n) = x((n))_N$, 再将 $\bar{x}(n)$ 左移得到 $\bar{x}(n+m)$, 最后取 $\bar{x}(n+m)$ 主值区间 ($n=0, 1, \dots, n-1$) 上的序列值, 得到有限长序列 $x(n)$ 的循环移位序列 $y(n)$ 。过程如图 1.5 所示。

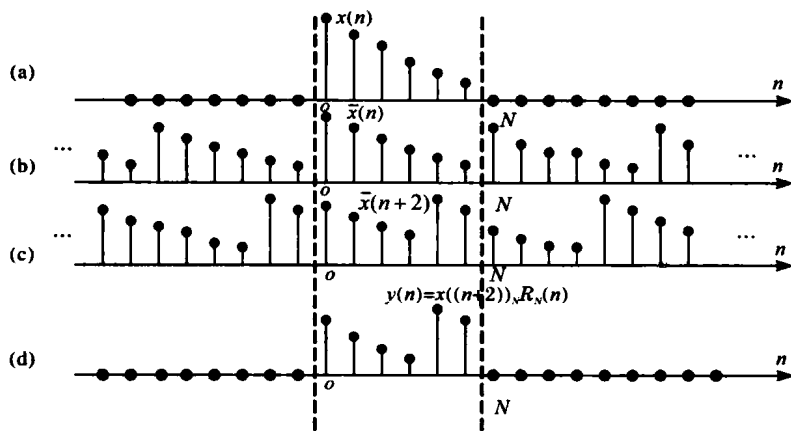


图 1.5 循环移位过程示意图 ($N=6$)

2) 时域循环移位定理

设 $x(n)$ 为有限长序列, 长度为 N , $y(n)$ 为 $x(n)$ 的循环移位序列, 即

$$y(n) = x((n+m))_N R_N(n)$$

则 $Y(k) = \text{DFT}[y(n)] = \text{DFT}[x((n+m))_N R_N(n)] = W_N^{-km} X(k)$, 其中 $X(k) = \text{DFT}[x(n)]$, $0 \leq k \leq N-1$ 。

$$\text{证明: } Y(k) = \text{DFT}[y(n)] = \sum_{n=0}^{N-1} x((n+m))_N R_N(n) W_N^{kn} = \sum_{n=0}^{N-1} x((n+m))_N W_N^{kn}$$

令 $n+m=n'$, 则有

$$Y(k) = \sum_{n'=m}^{N-1+m} x((n'))_N W_N^{k(n'-m)} = W_N^{-km} \sum_{n'=m}^{N-1+m} x((n'))_N W_N^{kn'}$$

由于上式中求和项以 N 为周期, 所以对其在任一周期上的求和结果相同。将上式的求和区间改在主值区间, 则得

$$\begin{aligned} Y(k) &= W_N^{-km} \sum_{n'=0}^{N-1} x((n'))_N W_N^{kn'} \\ &= W_N^{-km} \sum_{n'=0}^{N-1} x(n') W_N^{kn'} = W_N^{-km} X(k) \quad (0 \leq k \leq N-1) \end{aligned}$$

3) 频域循环移位定理

如果 $X(k) = \text{DFT}[x(n)]$, $0 \leq k \leq N-1$, $Y(k) = X((k+l))_N R_N(k)$, 则 $y(n) = \text{IDFT}[Y(k)] = W_N^l x(n)$ 。

$$\begin{aligned} \text{证明: } y(n) &= \text{IDFT}[Y(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X((k+l))_N R_N(k) W_N^{-kn} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X((k+l))_N W_N^{-kn} \end{aligned}$$

令 $k+l=k'$, 则有

$$\begin{aligned} y(n) &= \frac{1}{N} \sum_{k'=l}^{N-1+l} X((k'))_N W_N^{-(k'-l)n} = W_N^l \left[\frac{1}{N} \sum_{k=l}^{N-1+l} X((k'))_N W_N^{-k'n} \right] \\ &= W_N^l \left[\frac{1}{N} \sum_{k=0}^{N-1} X((k'))_N W_N^{-kn} \right] = W_N^l \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k') W_N^{-k'n} \right] = W_N^l x(n) \end{aligned}$$

3. 循环卷积定理

有限长序列 $x_1(n)$ 和 $x_2(n)$, 长度分别为 N_1 和 N_2 , $N = \max[N_1, N_2]$ 。 $x_1(n)$ 和 $x_2(n)$ 的 N 点 DFT 分别为: $X_1(k) = \text{DFT}[x_1(n)]$, $X_2(k) = \text{DFT}[x_2(n)]$ 。如果 $X(k) = X_1(k) X_2(k)$, 则

$$x(n) = \text{IDFT}[X(k)] = \left[\sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N \right] R_N(n)$$

在循环卷积过程中, 要求对循环反转, 循环移位, 特别是两个长度为 N 的序列的循环卷积长度仍为 N 。显然, 这与一般的线性卷积不同, 故称为循环卷积, 记为

$$x(n) = x_1(n) \otimes x_2(n) = \left[\sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N \right] R_N(n)$$

4. 复共轭序列的DFT

设 $x^*(n)$ 是 $x(n)$ 的复共轭序列, 长度为 N , 已知 $X(k) = \text{DFT}[x(n)]$, 则 $\text{DFT}[x^*(n)] = X^*(N-k)$, $0 \leq k \leq N-1$, 且 $X(N) = X(0)$ 。

证明:

$$\begin{aligned} \text{DFT}[x^*(n)] &= \sum_{n=0}^{N-1} x^*(n) W_N^{*k} R_N(k) = \left[\sum_{n=0}^{N-1} x(n) W_N^{*k} \right]^* R_N(k) \\ &= \left[\sum_{n=0}^{N-1} x(n) W_N^{(N-k)n} \right]^* R_N(k) \\ &= X^*((N-k))_N R_N(k) = X^*(N-k) \quad (0 \leq k \leq N-1) \end{aligned}$$

已知 $X(k) = \text{DFT}[x(n)]$, 则 $\text{DFT}[x^*(N-n)] = X^*(k)$ 。

证明: 因为 $X(k) = \text{DFT}[x(n)]$, 即

$$\begin{aligned} x(n) &= \text{IDFT}[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \\ x(N-n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-(N-n)k} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{*k} \end{aligned}$$

所以

$$x^*(N-n) = \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{*k} \right]^* = \frac{1}{N} \sum_{k=0}^{N-1} X^*(k) W_N^{-k} = \text{IDFT}[X^*(k)]$$

即 $\text{DFT}[x^*(N-n)] = X^*(k)$ 。

5. DFT 的共轭对称性

1) 有限长共轭对称序列和共轭反对称序列

用 $x_{\text{ep}}(n)$ 和 $x_{\text{op}}(n)$ 分别表示有限长共轭对称序列和共轭反对称序列。二者的定义为

$$\begin{aligned} x_{\text{ep}}(n) &= x_{\text{ep}}^*(N-n) \quad (0 \leq n \leq N-1) \\ x_{\text{op}}(n) &= -x_{\text{op}}^*(N-n) \quad (0 \leq n \leq N-1) \end{aligned}$$

当 N 为偶数时, 将上式的 n 换成 $N/2 - n$, 得到 $x_{\text{ep}}\left(\frac{N}{2} - n\right) = x_{\text{ep}}^*\left(\frac{N}{2} + n\right)$, $0 \leq n \leq N/2 - 1$ 。

当 N 为奇数时, 将上式的 n 换成 $(N-1)/2 - n$, 得到 $x_{\text{ep}}\left(\frac{N-1}{2} - n\right) = x_{\text{ep}}^*\left(\frac{N+1}{2} + n\right)$, $0 \leq n \leq (N-1)/2 - 1$ 。

任意有限长序列 $x(n)$ 可表示成共轭对称分量和共轭反对称分量之和 $x(n) =$