

doi: 10.3969/j.issn.1006-1576.2012.08.023

基于提升算法的 3 阶 Daubechies 离散小波变换的 FPGA 实现

张敬明¹, 万俊², 梁瑞涛¹, 胡清平²

(1. 中国人民解放军 92060 部队, 辽宁 大连 116041; 2. 海军工程大学兵器工程系, 武汉 430033)

摘要: 为提高小波变换的计算效率, 研究基于提升算法的 3 阶 Daubechies 离散小波变换及其逆变换的 FPGA 实现。简要介绍提升算法的基本原理, 给出 3 阶 Daubechies 小波变换及其逆变换的提升算法过程, 对正变换与逆变换的硬件实现结构进行设计, 该结构无需附加内存, 且采用流水线技术实现小波系数的快速并行输出, 大大节省了传统变换所需的存储空间并提高了计算速度。在 Quartus 设计软件中对提升算法结构进行仿真, 验证了提升结构的正确性。分别使用传统的基于卷积的 DB3 小波滤波器和设计的 DB3 提升结构对包含噪声的模拟信号进行小波阈值滤波处理。结果表明: 提升结构算法计算复杂度小, 在可接受的信噪比范围内, 能够快速实现信号的小波变换处理。

关键词: 提升算法; DB3 小波; 小波变换; FPGA 实现**中图分类号:** TJ02 **文献标志码:** A

FPGA Implementation of DB3 Discrete Wavelet Transform Based on Lifting Algorithm

Zhang Jingming¹, Wan Jun², Liang Ruitao¹, Hu Qingping²

(1. No. 92060 Unit of PLA, Dalian 116041, China;

2. Dept. of Weaponry Engineering, Naval University of Engineering, Wuhan 430033, China)

Abstract: The FPGA implementation of DB3 discrete wavelet transform and inverse transform based on lifting algorithm is studied to improve the calculation efficiency of wavelet transform. Based on the introduction of lifting algorithm principle, the calculation process of DB3 wavelet transform and its inverse transform is presented. The hardware implementation architecture of transform is designed. The wavelet coefficient can replace the original signal data in this architecture, which doesn't need any more memory to store the calculation data. The pipelining technology is used to speedup the calculation and output of wavelet coefficient. The lifting algorithm architecture is simulated in Quartus software and simulation result proves the correctness of the architecture. The traditional DB3 wavelet filter and the lifting algorithm architecture are used to process simulated signal by wavelet threshold de-noising. The results show that the signal noise ratio of method in this thesis is less than the traditional method, while it's still in a sustainable range and the calculation speed is higher than the traditional method.

Key words: lifting algorithm; DB3 wavelet; wavelet transform; FPGA implementation

0 引言

小波变换是 20 世纪 80 年代后期发展起来的应用数学分支, 并在近十几年里得到了快速的发展, 由于其具有良好的时频局部特性和多分辨分析特性, 小波变换在语音识别、图像处理、信号去噪、数据压缩、特征分析等领域都有广泛的应用^[1-3]。长期以来, 离散小波变换的工程实现一直使用 Mallat 快速算法^[4]。这种基于卷积的算法计算复杂, 运算量大, 对存储空间的要求高, 不太利于硬件的实时实现, 制约了它在速度较高或数据量较大的信号处理场合的应用。

Daubechies 等人在 20 世纪 90 年代末提出了小波变换的提升算法^[5], 被誉为第 2 代小波变换。它既继承了第 1 代小波良好的时频局部特性和多分辨分析特性, 又不依赖于傅立叶变换, 因此计算速度

快, 计算时无需额外的存储开销, 非常适合硬件实现。目前使用提升算法的离散小波变换硬件设计研究都集中于 JPEG 标准中的 5_3 小波和 9_7 小波上^[6-7], 以完成二维图像的实时处理。对于信号处理中其他常用的小波变换研究较少; 因此, 笔者研究基于提升算法的 3 阶 Daubechies 小波变换和逆变换的硬件实现, 以方便其在高速信号处理或实时图像处理等领域的应用。

1 提升算法的基本原理

Daubechies 在文献[5]中提出了小波变换的提升算法, 并且证明了任何离散小波变换都可以被分解成为一系列简单的提升步骤, 可以用提升算法来实现。提升算法中的离散小波变换是通过将小波滤波器的多相矩阵(polyphase matrix)分解成为一系列的上三角阵、下三角阵和一个对角阵来实现的。多相

收稿日期: 2012-03-20; 修回日期: 2012-04-23

作者简介: 张敬明(1976—), 男, 广西人, 学士, 助理工程师, 从事装备检测与信号处理研究。

矩阵 $P(z)$ 可以使用 Euclidean 算法进行分解得到式 (1), 此即为小波变换的提升方案。一个提升方案可以划分成分解、预测、更新和缩放 4 个步骤, 图 1 是提升算法前 3 个主要步骤示意图。

$$P(z) = \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \quad (1)$$

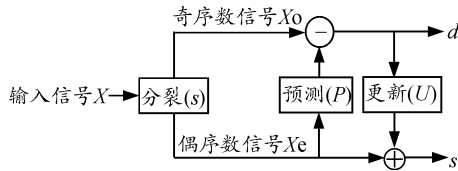


图 1 提升算法示意图

分解: 通常将输入信号分为偶数序列和奇数序列。预测: 在基于原始数据相关性的基础上, 用偶数序列 X_e 与一个预测算子 P 去预测奇数序列 X_o , 然后使用预测值与奇数序列的差值来代替奇数序列, 即 $d=X_o-P(X_e)$ 。这样就能使用更小的子集来代替原始信号集。更新: 为了使原始信号的某些全局特性在偶数信号中继续保持, 需要进行更新。可以利用第二步计算得到的 d 进行更新, 构造一个算子 U 去更新偶数信号, $s=X_e+U(d)$ 。 d 和 s 代表最终输出的高频和低频信号。可以看出, 提升算法能够在计算过程中不需要除了前级步骤的输出之外的数据, 即实现原位运算。这样在每个点都可以用新的数据流替换旧的数据流, 大大节省了运算过程中间变量的存储空间。

2 DB3 离散小波变换的提升算法

Daubechies 在文献[8]中给出了 DB3 小波变换的低通和高通滤波器, 分别为:

$$h(z) = h_2 z^2 + h_1 z + h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} \quad (2)$$

$$g(z) = -h_3 z^2 + h_2 z - h_1 + h_0 z^{-1} - h_1 z^{-2} + h_2 z^{-3} \quad (3)$$

式中各系数为: $h_2=0.333, h_1=0.807, h_0=0.459, h_1=-0.135, h_2=-0.085, h_3=0.035$ 。设 $P(z)$ 是 DB3 小波正变换的多相矩阵, 笔者使用 Euclidean 算法对 $P(z)$ 进行分解, 转化为一系列的上三角阵、下三角阵和一个对角阵相乘的形式, 得到式 (4):

$$P(z) = \begin{bmatrix} h_e(z) & g_o(z) \\ h_o(z) & g_e(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta z^{-1} + \beta' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma + \gamma' z & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \quad (4)$$

式中各系数为: $\alpha \approx -0.4123, \beta \approx -1.5651, \beta' \approx 0.3524, \gamma \approx 0.0285, \gamma' \approx 0.4922, \delta \approx -0.3896, \zeta \approx 1.9182$ 。将输入信号数据 x_n 分解为偶数序列信号 x_{2n} 和奇数序列信号 x_{2n+1} , DB3 小波正变换就可转化为以下步骤:

$$s'_n = x_{2n} + \alpha x_{2n+1} \quad (5)$$

$$d'_n = \beta s'_{n-1} + \beta' s'_n + x_{2n+1} \quad (6)$$

$$s''_n = s'_n + \gamma d'_n + \gamma' d'_{n-1} \quad (7)$$

$$d''_n = \delta s''_n + d'_n \quad (8)$$

$$s_n = \zeta s''_n \quad (9)$$

$$d_n = d''_n / \zeta \quad (10)$$

式中 s'_n, d'_n, s''_n, d''_n 都是提升算法的中间结果, s_n 和 d_n 分别为输出的低频系数和高频系数。由于式 (6) 中出现了 s'_n 前一个时刻的信号 s'_{n-1} , 这里将式 (6)~(10) 中信号都延迟一个时刻, 改写为:

$$d'_{n+1} = \beta s'_n + \beta' s'_{n+1} + x_{2n+3} \quad (11)$$

$$s''_{n+1} = s'_{n+1} + \gamma d'_{n+1} + \gamma' d'_{n+2} \quad (12)$$

$$d''_{n+1} = \delta s''_{n+1} + d'_{n+1} \quad (13)$$

$$s_{n+1} = \zeta s''_{n+1} \quad (14)$$

$$d_{n+1} = d''_{n+1} / \zeta \quad (15)$$

根据式 (4) 还可以得到 DB3 小波逆变换的滤波器组的多相矩阵 $P(z)^{-1}$, 式中各系数的值与式 (4) 中相同, 小波逆变换也可类似的转化为一系列乘法与加法运算, 这里不再赘述。

$$P(z)^{-1} = \begin{bmatrix} 1/\zeta & 0 \\ 0 & \zeta \end{bmatrix} \begin{bmatrix} 1 & -\delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\gamma - \gamma' z & 1 \end{bmatrix} \begin{bmatrix} 1 & -\beta z^{-1} - \beta' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\alpha & 1 \end{bmatrix} \quad (16)$$

3 基于提升算法的离散小波变换 FPGA 实现

3.1 DB3 离散小波变换与逆变换的硬件结构

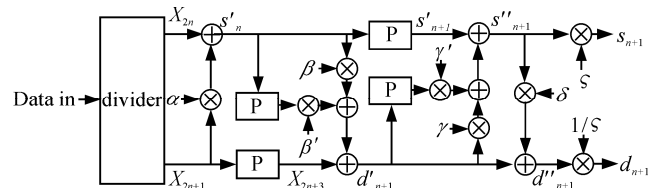


图 2 DB3 小波变换硬件实现结构

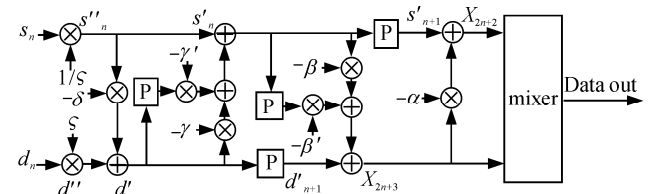


图 3 DB3 小波逆变换硬件实现结构

根据上述的提升算法计算步骤, 设计了 DB3 小波变换与逆变换的硬件实现结构, 如图 2 和图 3 所示。图 2 中 “divider” 表示信号分解器, 将信号分解为奇偶序列; “mixer” 是信号合成器, 将输入的两路信号按奇偶秩序合成为一个信号; “P” 是寄存器, 表示信号延迟一个时间单位; “+” 和 “×” 分

别表示加法器和乘法器，乘法器的系数值就是提升算法计算步骤公式中的系数值。根据设计的小波变换与逆变换硬件实现结构，就能够方便地在 FPGA 中构建 DB3 小波变换与逆变换的功能模块。

3.2 DB3 离散小波变换与逆变换的 FPGA 设计

现场可编程门阵列 (field programmable gate array, FPGA) 是近年来广泛应用的超大规模、超高速的可编程逻辑器件，由于其具有高集成度、低功耗、高运算速度等优点，在超高速信号处理方面有非常广泛的应用。笔者使用 Altera 公司的 Cyclone 系列 FPGA 来实现 DB3 离散小波变换，FPGA 型号为 EP2C8Q208C8，使用 Altera 公司的 Quartus II 软件进行 FPGA 功能设计。首先在原理图设计模式下构建 DB3 小波变换的各步骤运算模块，它们分别对应式 (5)~(15)。对各步骤运算模块创建各自设计文件，使用 VHDL 语言进行模块的功能描述。分别将前一步模块的输出作为后一步模块的输入，各模块使用统一系统时钟信号，连接各模块之后构成 DB3 离散小波变换与逆变换的整体模块。

3.3 软件仿真

在 Quartus II 软件中完成设计后，对工程文件进行编译，并建立矢量波形文件进行仿真测试，在仿真波形文件中添加输入信号、输出信号和各级运算模块中的寄存器以检验提升算法各步骤的运算结果。生成 Quartus 仿真网表后进行系统工程仿真。输入信号使用常见的正弦信号。通过对比得到的仿真结果和式 (5)~(15) 的公式的计算结果，可验证设计的工程文件实现了 DB3 离散小波正变换和逆变换。但是各级运算步骤的中间量存在微小的误差，这主要是由于使用 FPGA 进行乘法除法运算时都是采用二进制数据移位的方式进行的，因此对滤波器系数有略微的近似。另外，提升算法每一步骤模块执行都会延时一个时钟周期，这是因为 FPGA 内部寄存器之间的赋值需要在下一个周期时钟上升沿才能触发完成。将正变换得到的输出结果 (即高频输出信号和低频输出信号) 作为逆变换的输入信号进行仿真验算，将结果对比原始输入的信号发现丢失了末尾几个数据，这是由于在提升算法步骤中，有部分公式需要将此时刻 s 值与下一时刻 d 值共同计算，这使最后的一个 s 值要作舍去处理，经过正变换与逆变换 2 次提升算法，就会丢失几个数据。但对于数据量较大的输入信号的影响可以忽略不计。

4 小波阈值去噪与结果分析

滤波去噪是小波变换的一个重要应用领域，D.L.Donoho 和 I.M.Johnstone 在小波变换的基础上提出了小波阈值去噪的概念，该方法在信号处理方面得到了广泛的应用。Donoho 提出的小波阈值去噪方法分为以下 3 个步骤：

- 1) 对输入信号作小波变换，得到一组小波系数；
- 2) 对小波系数进行阈值处理，得到修正后的系数值；
- 3) 利用修正后的系数值进行小波逆变换，重构信号作为滤波去噪后的信号。

第 2 步的阈值处理是小波阈值去噪的重要步骤。Donoho 小波阈值处理的基本思想是：当小波系数小于某个临界阈值时，认为这时的小波系数主要是由噪声引起的，予以舍弃；当小波系数大于这个临界阈值时，认为这时的小波系数主要是由信号引起，但仍含有噪声的影响，于是把这一部分系数进行压缩保留，即软阈值方法。为了验证文中小波提升结构的实际应用有效性，这里将提升算法应用于小波阈值去噪的验算。在正弦波中添加高斯噪声构成输入信号，原始信号与输入信号如图 4 所示。

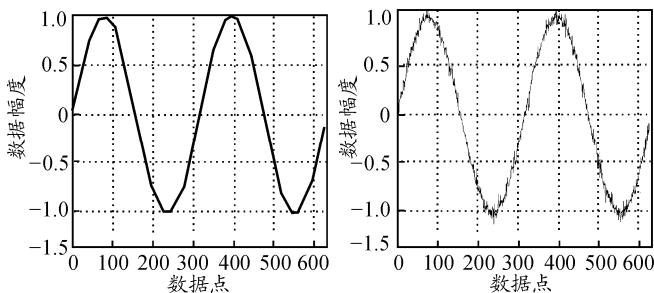
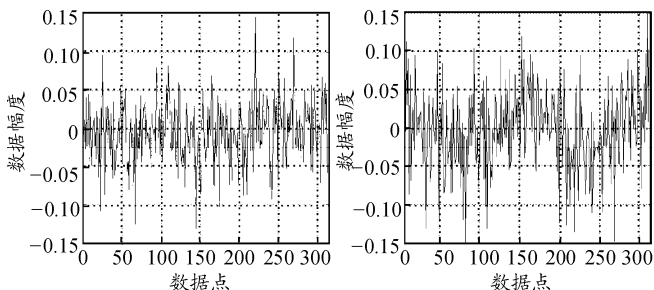


图 4 原始信号与添加噪声后的信号

分别使用文中的提升结构和传统的基于卷积的 3 阶 Daubechies 滤波器完成该信号的 3 层小波变换，得到的结果如图 5 所示。图 5 中左侧为基于卷积 DB3 小波分解得到的系数，右侧是基于提升方案的 DB3 小波分解得到的系数。从图中的结果曲线可以看出小波变换后得到的高频系数主要是噪声引起的，低频系数则包含了原信号的信息。



(a) 1 层 DB3 小波分解与提升小波分解得到的高频系数

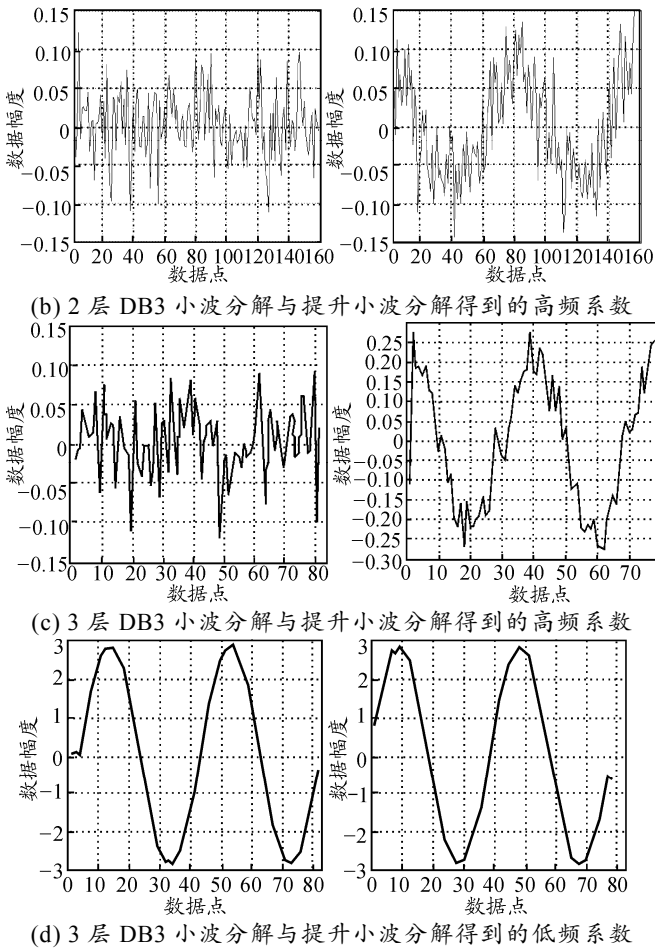


图 5 三层 DB3 小波变换的结果

虽然 2 种方法得到的高频系数存在差异, 但对高频系数进行阈值处理后, 高频系数都被置零, 而 2 种方法得到的低频系数基本相同。再进行信号的小波重构, 得到结果如图 6 所示。

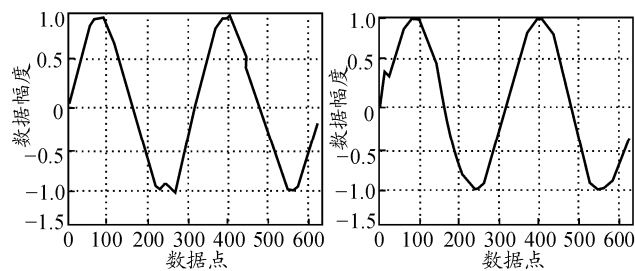


图 6 小波阈值去噪处理的结果

可以看出 2 种小波阈值去噪方法都有效去除了原输入信号中的噪声, 但是使用提升小波实现信号滤波的结果中出现了信号头部的畸变, 这主要还是由前述的提升算法中首尾信号丢失造成的, 对于信号数据量大的情况, 这种影响可以忽略, 而在数据量小的情况下要克服这一不足, 可以使用信号的补零延拓或对称延拓等方法。为了对 2 种方法进行定量比较, 笔者使用均方误差(MSE)和信噪比(SNR)作为比较指标, 它们的定义如下:

$$MSE = \sum_{i=1}^N \frac{(s_i - y_i)^2}{N} \tag{17}$$

$$SNR = 10 \lg \frac{\sum_{i=1}^N s_i^2}{\sum_{i=1}^N (s_i - y_i)^2} \tag{18}$$

式中: s 为原始正弦信号; y 为含噪信号经过滤波处理后的输出信号; N 为信号数据的长度。将输出的 2 种滤波处理后的信号和原始正弦信号分别代入式 (17) 和式 (18), 计算得到基于卷积的 DB3 小波阈值处理结果的均方误差为 2.633×10^{-4} , 信噪比为 75.556 5 dB, 而基于提升小波的阈值去噪处理结果的均方误差为 6.4×10^{-3} , 信噪比为 43.588 8 dB。虽然使用提升小波实现阈值去噪的结果的 2 个性能指标都较基于卷积的处理结果略低, 但是仍在可接受的范围之内, 信号波形没有产生严重畸变, 能够应用于信号或图像的滤波去噪实时硬件实现。

5 结束语

笔者研究了基于提升算法的 3 阶 Daubechies 离散小波变换及其逆变换的硬件实现。由于提升结构能实现小波变换的流水线快速实现, 计算复杂度小, 可以达到实时处理的程度。实例计算和结果对比分析证明了笔者设计的硬件结构的可行性和实用性。

参考文献:

- [1] 陈晓惠, 郑晨, 段汕. 形态小波域多尺度马尔可夫模型在纹理图像分割中的应用[J]. 中国图像图形学报, 2011, 16(5): 761-766.
- [2] 姚诚, 刘广孚, 李忠国. 基于小波系数功率谱的潜油电泵偏磨故障诊断[J]. 仪器仪表学报, 2011, 32(8): 1757-1762.
- [3] 袁飞. 基于平移不变小波的语音信号去噪研究[J]. 自动化技术与应用, 2011, 30(8): 40-42.
- [4] Mallat S G. A theory for multiresolution signal decomposition: the wavelet representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(7): 674-693.
- [5] Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps[J]. Journal of Fourier Analysis and Applications, 1998, 4(3): 247-269.
- [6] 郝燕玲, 刘营. 应用于 JPEG2000 的离散小波变换并行超大规模集成结构[J]. 光学精密工程, 2009, 17(5): 1181-1186.
- [7] 王前, 吕东强, 栗靖. 新型 9/7 小波基构造及快速实现[J]. 电子与信息学报, 2009, 31(5): 1210-1213.
- [8] Daubechies I. Orthonormal bases of compactly supported wavelets[J]. Comm. Pure Appl. Math., 1988, 41(6): 909-996.