

文章编号: 1007-2780(2011)03-0404-05

二维提升小波的 VLSI 结构设计及 FPGA 验证

孟 伟^{1,2}, 金龙旭^{1*}, 韩双丽¹

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033, E-mail: mengw999@126.com;

2. 中国科学院 研究生院, 北京 100039)

摘 要: 针对 JPEG2000 中小波变换的硬件实现占用资源量大、速度慢等问题, 提出了一种有效的二维小波硬件实现模型。该模型采用流水线并行结构, 即对图像中各行像素进行流水线处理的同时, 对小波分解的各级采用并行结构处理。这样的结构提高了小波变换的处理速度, 实现了实时处理, 节省了硬件的片上存储及外部存储资源。用 FPGA 对此模型进行验证。验证实验采用 Xilinx 公司的 SPARTEN-3 系列芯片, 对 1024×2048 的大图像进行处理, 图像处理速度达到 80 Mpixels/s, 满足实时性要求。

关 键 词: JPEG2000; 小波变换; 并行结构; 提升算法

中图分类号: TN911.73

文献标识码: A

DOI: 10.3788/YJYXS20112603.0404

Design of VLSI Architecture of 2-D Lifting Wavelet and FPGA Verification

MENG Wei^{1,2}, JIN Long-xu^{1*}, HAN Shuang-li¹(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences,
Changchun 130033, China, E-mail: mengw999@126.com;

2. Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to save the resource and improve the speed of wavelet transform's hardware realization, an effective 2-dimension wavelet transform hardware structure has been proposed. This structure used parallel pipeline architecture, each layer of wavelet transform processed by parallel while pixels processed by pipeline. And the structure can improve the speed of wavelet transform, save the on-chip and external memory resource. It had been tested on Xilinx SPARTEN-3 series FPGA by processing large image with size 1024×2048 . Results showed the image processing speed could reach 80 Mpixel/s, meeting the real-time requirement.

Key words: JPEG2000; wavelet transform; parallel architecture; lifting computation

1 引 言

经典的小波分析是从傅立叶的基础发展起来的, 主要建立在二进平移和伸缩的思想基础上, 它在一定程度上受到傅立叶分析的限制, 算法复杂, 占用存储空间大, 已经不能很好地满足图像处理

高效率、低能耗的要求以及视频处理的要求。1996 年, Swedens 提出了不依赖于傅立叶变换的小波提升算法。提升算法提供了一种小波的快速实现方法, 与经典的小波算法相比, 运算量大大减少, 并且能够实现原位计算, 硬件实现过程中可节省存储单元, 因此更适合于硬件实现。

收稿日期: 2010-08-19; 修订日期: 2010-10-15

作者简介: 孟伟(1985—), 女, 山东莱芜人, 博士研究生, 主要从事图像处理算法研究及其 FPGA 实现。

* 通讯联系人, E-mail: jinlx@ciomp.ac.cn

本文采用 FPGA 验证了提升算法的实时变换,对 $1\,024\times 2\,048$ 的大图像的处理速度达到 80 Mpixels/s,满足实时性要求。

2 小波提升算法

2.1 小波提升算法

小波提升算法分为 3 个步骤:分裂、预测和更新,如图 1 所示。首先通过 LAZY 变换将输入的离散信号分为奇数集合和偶数集合;然后交替使用对偶提升和提升,经过若干次后,偶数样本相当于低通小波系数,奇数样本相当于高通小波系数;最后,经过以 K 为比例因子的伸缩变换,完成整个运算。

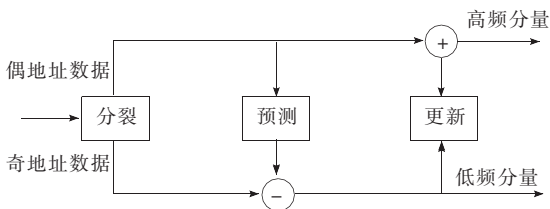


图 1 提升小波分解步骤

Fig. 1 Lifting wavelet decomposition

计算公式如下:

$$Y_{i+1} = a(x_{2i} + x_{2i+2}) + x_{2i+1} \quad (1)$$

$$Y_{2i} = b(Y_{2i+1} + Y_{2i+3}) + x_{2i} \quad (2)$$

其中, x_{2i} 代表 x 的偶数序列, x_{2i+1} 代表 x 的奇数序列, Y_{2i+1} 为分解后的高频部分, Y_{2i} 为分解后的低频部分。

2.2 提升算法用硬件实现的优点

小波提升算法的计算过程可以用图 2 表示。 $X_0 \sim X_8$ 为一组图像数据的一部分,经过如图 2 的计算把原图像分成高频和低频。图 2 为 (5, 3) 小波变换^[1]。

提升小波更适用于硬件实现,其优点主要表现为以下几方面:

(1) 计算量比基于滤波器组卷积的方法降低

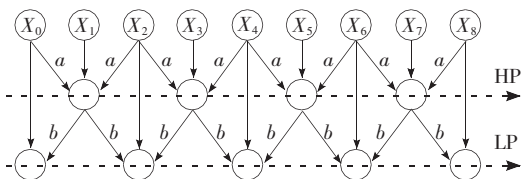


图 2 提升算法的数据关系图

Fig. 2 Data dependency diagram for lifting computation

了一半,在提升算法中只用到相乘和累加,又因为算法中的乘法是 2 倍数的乘法,硬件实现中的乘法器可以用移位代替。硬件实现算法的瓶颈就是算法的复杂度,而提升算法的出现为小波变化的硬件实现提供了基础,从而能使整个图像的处理速度得到提高。

(2) 提升小波的实现过程是原位计算,在运行过程的中间结果不用存储,直接参与下一次的运算,因此不需要额外的缓冲,可以节省硬件的片上存储。

(3) 由于提升算法的运算只涉及到相邻的两个像素,使得小波实现可以逐行处理,每处理完 3 行进行列处理,使得小波变化速度很快,可达到实时。相对与传统小波算法,需要先对整幅图像进行行变化,再进行列变化,不仅要占用大量存储单元,更限制了处理速度。特别是对处理大幅图像或者无边界的遥感图像,提升小波算法的优点更是突出。

(4) 提升小波的过程是整数到整数的变换,易于硬件实现,且能满足图像无损压缩的要求。

(5) 没有边界问题,不需要边界扩展,为硬件实现节省资源。

3 提升小波硬件实现结构设计

随着开发工具的日渐完善和优良性能日益显露,利用 FPGA 来实现各种信号处理算法已成为一种普遍现象和发展趋势。用 FPGA 实现图像处理算法有很多优点:

(1) 高度并行性。用户根据算法特点在内部采用多个算术单元的并行结构,提高其运算能力。

(2) 高效运算。许多图像处理算法如滤波,包含大量与常数相乘的操作,可以在内部采用查找表的方式高效实现,而不需要采用复杂的乘法阵列结构,从而可节省大量的硅片空间。

(3) 流水线处理。在复杂的组合逻辑之间插入寄存器,保证流水处理方式有很高的数据吞吐量并且插入的寄存器不会占用额外的芯片资源。

(4) 在线动态重构。用户可根据需要加载不同的设计程序使其实现特定的功能,即用一套硬件系统可以根据不同的需求完成不同的任务。

(5) 高速数据传输和存储。高性能的 FPGA 内部包含大量的片内存储器,可以在数百兆的数

据率下实现数据的传输和存储操作。与 DSP 相比,用户可以根据算法特点用 FPGA 设计合适的硬件实现结构,提高芯片内集成电路的有效性。

3.1 整体结构框架

本文以(5,3)小波为例来说明提升小波的模型设计。整体设计采用流水线并行结构,结构图如图 3 所示。

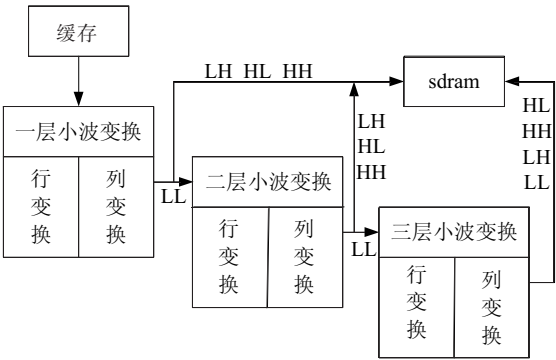


图 3 提升小波总体结构图
Fig. 3 Structure of lifting wavelet

3.1.1 输入

由于主程序是 80 MHz 频率,而 CMOS 工作频率为 48 MHz,并且是连续输出数据,所以整体结构需要对输入进行缓冲。本设计中用两个缓冲,缓冲采用乒乓结构,每一缓冲存储一行数据,在主程序处理缓冲一中的数据时,下一行数据存入缓冲二。

3.1.2 多级并行模块

这个模块是设计的核心模块。首先进行行变换,每 3 行为一组,第一个 3 行的行处理结束后进行列处理,以后每两行的行处理结束后,与前一行的结果构成 3 行进行列处理,从而构成流水线结构。在以往文献^[2-5]中,采取的结构是先处理完一层小波变换后存储,再进行第二层的小波变换,这样的处理模型虽然可以使资源重复利用,节省硬件资源,但需要大容量的内部存储和外部存储,存储单元的增加又限制了芯片的处理速度,所以这样的模型并不是最优的。随着集成电路的发展,芯片所含逻辑量大大增加,为提升小波速度的进一步提高提供了基础。本文采用并行处理的结构,在处理第一层小波的同时,处理第二和第三层小波。虽然所用的硬件逻辑资源有所增加,但这样的并行结构可实现实时的小波变换而且可以大大节省存储单元,使小波算法的硬件实现不再受

FPGA 片上存储资源的限制。

小波变换用 4 个内部 RAM 以流水线形式存储行变换结果,当前 3 行的变换结束后,在存储第 4 行变换结果的同时取前 3 行数据进行列变换。行变换计算需要 2 个周期,列变换需要 4 个周期,列变换的时间是行变换的两倍,即列变换要占用 2 行的行变换时间,每次列变换的后一半进行时,行变换得暂停,等待列变换运算结束后继续下一行的行变换和存储。CMOS 输入 4 行的时间 T 为 $4N/48$ (N 为一行的数据量),程序完成 4 行行变换的时间 t 为 $6N/80$ 。若 $T > t$,则表明在输入 4 行的时间内可以完成 4 行的变换,所以本文结构可以完成实时变换。本文结构用 4 个内部存储,能最高效率地完成行列变换流水线,而不产生数据冲突。第二级和第三级小波变换所要处理的像素数量分别为上一级变换的一半,所以也能在第一级小波变换处理一行的时间内可同时完成,因此该结构能高效地完成小波变换。

3.1.3 输出存储

输出结果存入 SDRAM,由于整个过程采用并行处理,而结果只能串行输出到 SDRAM 存储,所以每层小波变换后的有效数据需要在下一层小波变换的同时存储完成。每层小波变换分解出 4 组数据,分别为 LL、LH、HL、HH,其中低频(LL)数据继续下一层小波变换的分解,LL、LH、HH 分量存储到 SDRAM 的不同地址。由于 LL、LH、HH 为并行输出,SDRAM 为串行存储,并且分解后输出数据为 8 位,SDRAM 为 32 位,所以在存储前加缓冲以防止数据丢失同时满足 32 位存储要求。

3.2 二维提升小波的行列变换结构

为提高效率实现流水线并行结构,本文对提升小波算法的公式进行了改进,使其更适合 FPGA 实现。所设计的提升小波算法硬件实现模型在进行提升运算时可直接计算出行或列变换的高频和低频。

3.2.1 行变换

行变换计算公式如公式(3)、(4)所示,其中低频变换不依靠高频变换的值,而是直接由原数列的值计算出来。其中, x_{2i} 代表 x 的偶数序列, x_{2i+1} 代表 x 的奇数序列, Y_{2i+1} 为分解后的高频部分, Y_{2i} 为分解后的低频部分。

$$Y_{2i+1} = -\frac{1}{2}(x_{2i} + x_{2i+2}) + x_{2i+1} \quad (3)$$

$$Y_{2i} = \frac{1}{2}(7x_{2i} + 2x_{2i+1} - 2x_{2i+2} + 2x_{2i+3} - x_{2i+4}) \quad (4)$$

计算结构图如图 4 所示,系数如公式 (3)、(4)所示。其中输出的边界如 Y_0 等的处理如图 4 所示, Y_0 的计算结果只由前 3 个像素决定,系数由公式(5)给出。

$$Y_0 = \frac{1}{8}(7x_0 + 2x_1 - 2x_2) \quad (5)$$

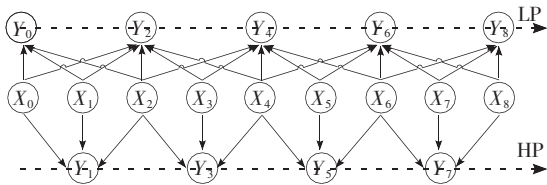


图 4 行计算的数据关系图

Fig. 4 Data dependency diagram for row computation

3.2.2 列变换

列变换公式如式(6)、(7)所示,低频变换由行变换的结果和上一列低频变换的值计算而来。这样的结构不仅节省了中间结果的存储量,而且易于实现流水线结构。

$$Y_{2i+1} = \frac{1}{2}(x_{2i} + x_{2i+2}) + x_{2i+1} \quad (6)$$

$$Y_{2i} = \frac{1}{8}(7x_{2i} - x_{2i+2}) + \frac{1}{4}(x_{2i+1} - Y_{2i-1}) \quad (7)$$

每 3 行变换运算结束后,进行列变换,低频和高频同时计算输出,其中低频计算依赖行变换结果和上一行列变换的高频结果。这样的设计可以

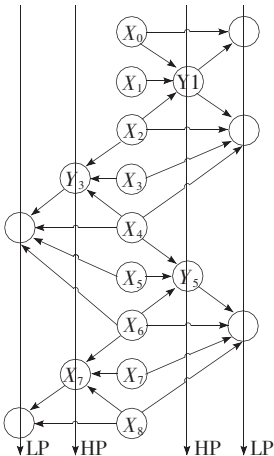


图 5 列计算的数据关系图

Fig. 5 Data dependency diagram for column computation

使得新一行列变换的高频结果存储在前一行计算结果的存储单元中,而新一行列变换的低频计算不受影响。其中 Y_0 为边界, Y_0 的值由前 3 个像素根据公式(5)计算得出。

计算过程中的除法用移位代替。输入像素有负值存在,而负数是以补码形式存储,所以在进行计算之前需要对其进行补码处理,对负值进行取绝对运算然后移位,计算结束后,存储时再进行补码运算将其恢复。

3.2.3 小波变换的字长

由于提升小波变换算法在计算高低频数据时采用了加减法,所以可能会导致结果的溢出,从而破坏数据的正确性。在硬件实现上,增加位会增加硬件资源和存储器,因此需要对小波变换的字长进行合理选择。计算时,每一级变换都可致结果字长有 1~2 位的增长。如果设计一个 J 级分解,为了能一致地表示所有计算中的系数字长,内部存储器字的宽度最好能满足最坏的情况,也就是第 J 级的宽度。而第 J 级输出系数的最大可能宽度为 $8+2J$ 位^[6]。对于 5\3 滤波器,当整数位取到 14 时,可得到完全无损的定点表示。然而,在权衡精度与硬件资源的情况下,本文设计的提升小波算法结构采用 10 位宽度。

4 实验结果与讨论

对于流水线并行结构,在 Xilinx8.2 下进行编程,并用 Modelsim XE 进行功能仿真和布线后仿真。布线后仿真时钟频率达 80 MHz,相当于 80 Mpixel/s,图像大小为 1 024×2 048,仿真波形图如图 6 所示,资源占用率如表 1 所示。从仿真波形中看出,该结构很好地实现了小波变换的快速处理,三层小波流水线并行输出,第一层小波变换每三行计算完毕,第二层小波计算一行,同样



表 1 资源利用
Table 1 Device utilization summary

Logic utilization	Used	Available	Utilization/ %
Slices	7 534	13 312	56
slice flip flops	2 524	26 624	9
4 input LUTs	14 373	26 624	53
bonded IOBs	58	487	11
BRAMs	29	32	90
MUTRT18×18s	6	32	18
GCLKs	8	8	100

第二层小波每 3 行计算完毕,第三层计算一行,而从整体看,三层小波不同行又是同时计算完成。图 4 中的信号是并行的计算结果串行存储的仿真

波形,从波形可以看出每一行数据计算完毕后,开始串行存储,由于计算一个像素的行列变换需要两个时钟周期,而存储只需要一个,所以在并行运算的同时实现了串行输出。与文献[7-10]相比,资源利用率也得到充分的提高。由于采用的是大图像的仿真测试,受逻辑长度影响,速度比小图像有所降低;但大图像更贴近实际需要,能更好地模拟实际情况。

5 结 论

以 JPEG2000 中的(5,3)小波变换为例,提出了二维提升小波的 FPGA 结构设计,整体结构主要采用分时复用和流水线并行处理的办法,实现了小波变换的实时处理,为图像的进一步处理奠定了基础。

参 考 文 献:

[1] Tinku Acharya, Tsai Ping-Sing. *JPEG2000 Standard for Image Compression: concept, algorithms and VLSI architectures* [M]. Canada: John Wiley & Sons, Inc., 2004:96-104.

[2] 马伍新,崔占忠,代方震. 用 FPGA 实现(5,3)小波变换 [J]. 计算机工程与设计,2003, 24(1): 47-49.

[3] Lian C J, Chen K F. Lifting based discrete wavelet transform architecture for JPEG2000 [C]// *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems*, New Jersey, USA:IEEE, 2001:445-448.

[4] Movva S, Srinivasan S. A novel architecture for lifting based discrete wavelet transform for JPEG2000 standard suitable for VLSI implementation [C]// *Proceedings of 16th International Conference on VLSI Design*, India: IEEE, 2003: 202-207.

[5] Xiong C Y, Tian J W. The improved lifting scheme and novel reconfigurable VLSI architecture for the 5/3 and 9/7 wavelet filters [C]//*Proceedings of 2004 International Conference on Communications, Circuits and Systems*, China: Chengdu:IEEE, 2004:728.

[6] 田宝凤,徐抒岩,孙荣春,等. 一种适合星上应用的遥感图像有损压缩算法 [J]. 光学 精密工程, 2006, 14(4): 725-730.

[7] 王丽荣,申铨国,王延杰. 基于提升方法的整数小波在 DSP 上的实现 [J]. 光学 精密工程, 2004,12(2):187-188.

[8] 朱珂,华林. 应用于 JPEG 2000 的高性能离散小波变换 VLSI 结构设计 [J]. 计算机辅助设计与图形学学报,2004, 16(7):1010-1015.

[9] 王承君,熊承义. 二维 5/3 小波变换的 FPGA 结构设计与实现 [J]. 中南民族大学学报(自然科学版), 2006, 25(4): 50-53.

[10] 刘军伟,饶妮妮. 提升小波变换的 FPGA 设计与实现 [J]. 微计算机信息, 2005, 21(10):132-134.