

二维 5/3 小波变换在并行计算单元中的设计实现

温琳卉, 谢 憬, 王国兴

(上海交通大学 微电子学院, 上海 200240)

摘 要: 本文提出了一种针对整数二维 5/3 小波提升算法的并行计算设计方案, 其整体结构具有行变换与列变换之间并行计算、数据分组输入、不同行变换(列变换)之间并行计算的特点。文中重点介绍了该小波提升算法的取整处理模式、算法改进和硬件设计实现等方面。本文结构平均每周期输出 2 个变换结果, 完成对 $N \times N$ 大小图像的处理需花费大约 $N^2/2$ 个时钟周期, 同时在 FPGA 中实现最高同步时钟频率 394.415MHz。

关键词: 5/3 整数小波提升变换; 并行计算单元; 硬件设计; FPGA

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2013)07-0047-04

Design and Implementation for 2-D IWT using Parallel Computing Units

WEN Lin-hui, XIE Jing, WANG Guo-xing

(School of Microelectronics, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: In this paper, a novel design and hardware implementation using parallel computing for 2-D integer lifting wavelet transform (IWT) is proposed of which input datas are divided into groups, its row and column transform module use parallel computing technology. This design is capable to produce 2 results every cycle, which takes $N^2/2$ time units to finish a $N \times N$ 2D IWT. And also, we implemented it in FPGA, the estimated frequency of operation is 394.415MHz.

Key words: 5/3 integer lifting-base scheme; parallel computing units; hardware design; FPGA

1 引言

小波变换是分析非平稳信号的强有力的工具, 其具有良好的空间-频率局部化特性及多分辨力特征等。是目前分析非平稳信号的强有力的工具^[1]。

Mallat 算法使得小波变换从理论走向应用研究。而后出现的提升算法大大简化了小波变换的运算过程, 整数小波提升算法更是能够将整数序列映射成整数小波变换系数, 有利于无损数据处理的实现。但是在硬件实现中, 整数小波提升算法在运算速度及硬件消耗等方面还存在较大的提升空间。

本文主要研究整数二维 5/3 小波提升算法的并行设计及优化实现, 针对算法的并行性进行分析, 提出一种行与行、列与列、行与列完全并行的硬件实现

方案, 使运算速度有较大提升。

2 提升算法的并行优化处理思路

2.1 整数 5/3 小波提升算法

本文研究对象整数 5/3 提升小波变换的基本算法公式为^[2]:

$$c(2n+1) = x(2n+1) - \left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor \quad (1)$$

$$d(2n) = x(2n) + \left\lfloor \frac{c(2n-1) + c(2n+1) + 2}{4} \right\rfloor \quad (2)$$

在此基础上, 将整数 5/3 小波提升算法进行展

收稿日期: 2012-10-17; 修回日期: 2012-11-22

基金项目: 国家“八六三”计划项目(2009AA011705)

开,即(其中 $\lfloor \cdot \rfloor$ 表示取整):

$$c(2n+1) = x(2n+1) - \left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor \quad (3)$$

$$d(2n) = \left\lfloor \frac{4 * x(2n) + x(2n-1) + x(2n+1) + 2 - \left\lfloor \frac{2(2n+2) + x(2n)}{2} \right\rfloor + \left\lfloor \frac{x(2n-2) + x(2n)}{2} \right\rfloor}{4} \right\rfloor \quad (4)$$

由上式分析可知算法具有以下特点:

(1) 将 $d(2n)$ 进行展开并改变取整操作, 消除 $d(2n)$ 与 $c(2n-1)$ 、 $c(2n+1)$ 的关联性, 高频系数 $c(2n+1)$ 与低频系数 $d(2n)$ 可并行计算。

(2) 算法可重构, 其重构算法为:

$$x(2n) = \left\lfloor \frac{4d(2n) - c(2n+1) - c(2n-1)}{4} \right\rfloor \quad (5)$$

$$x(2n+1) = c(2n+1) + \left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor \quad (6)$$

(3) 计算 $d(2n)$ 可以考虑使用变形算法, 即直接根据原始数据进行计算, 也可在计算出 $c(2n+1)$ 和 $c(2n-1)$ 的结果后再进行计算。

2.2 数据的边界处理和有限精度

基于对上述分析, 本文拟采用对称拓展来对边缘数据进行处理。其延拓方法如图 1 所示, 白色圆圈代表对边缘数据进行延拓之后所得数据^[3]。

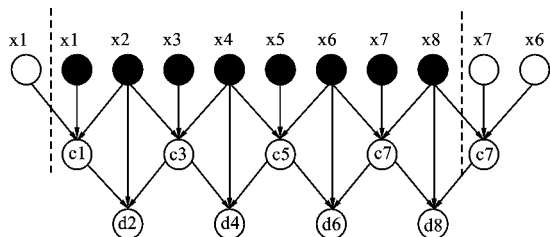


图 1 小波变换边缘数据延拓

在小波变换的过程中, 计算中每一级变换都可能导致结果字长有 1~2 位的增加, 本文设定的处理对象为静止图像 lena, 考虑到算法计算中各种最坏情况, 在此设定变换过程中数据字长为 12 位。

3 硬件设计实现

3.1 系统整体框架

本次设计将初始数据 ($N \times N$ 大小图像经过边缘扩展处理之后的数据) 以层为变换单位将数据按照特定规律不断输入到系统中, 同时, 为提高行列变换并行度, 将采用行变换与列变换并行执行的设计理念, 即在行列变换单元之间插入中间变换结果存

储器, 当行变换结果满足列变换输入数据要求数目时, 则列变换启动。

系统基本框架见图 2。其经过 MUX 模块选择之后进入行缓冲单元 (以下简称 RRF), 后进入行变换单元 (简称 RTU) 进行行变换, 行变换结果写入行变换存储器 (简称 RTRM), 当行变换结果数目满足条件时, 数据将从 RTRM 进入到列缓冲单元 (简称 CRF), 进而在列变换单元 (简称 CTU) 进行列变换, 将最终结果写到存储器 TRM 中。

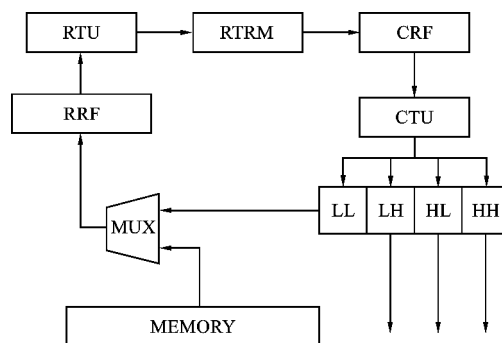


图 2 系统基本框架图

TRM 分为 4 部分: LL、LH、HL 和 HH, 其中 LL 存储 LL 数据, LH, HL, HH 亦同理。LL 数据可作为下一层变换源数据, LH, HL, HH 则传输到片外。整个系统的各读写、选通信号均为控制单元 CU (control unit) 生成。

3.2 行(列)变换硬件实现

行变换与列变换的实现主要由缓存器与运算单元组成, 两者实现过程相同, 故以行变换为代表进行说明。

(1) 行变换并行运算单元 (RTU)

行变换硬件说明见图 3。行变换硬件实现由 4 个 RTU、1 个 RRF 和 1 个 RTRM 组成。假设加法和移位的运算时间为 1 个时钟周期, 4 个 RTU 能在一次计算中花费 4 个周期能得到 8 个结果, $c(m)$ 、 $d(n)$ 代表在本次并行计算中得出的第 m 、 n 个高频 (低频) 系数。其主要计算过程表述如图 4。

以 $c(01)$ 、 $d(02)$ 、 $c(03)$ 的计算过程为例, 运算流程见图 4。其中 $X1 \sim X5$ 是运算所需的 5 个数据,

它们需经过 7 次加法和 1 次移位运算得出结果. X_1 与 X_3 、 X_3 与 X_5 相加后的结果右移 1 位(即除 2 取整运算)在硬件中用数据截取实现,所以不需要花费时钟周期即可完成.

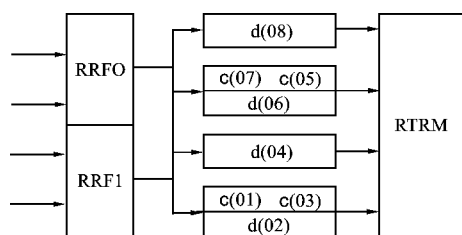


图3 行变换硬件框架

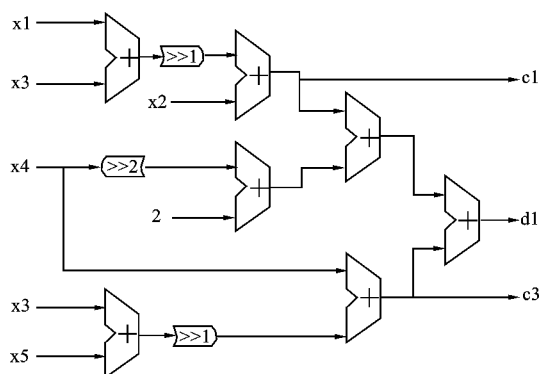


图4 数据运算流程

经过分析可知某个时刻最多只需 2 个加法器和 1 个移位器,为了节省面积,RTU 中将采用加法器复用技术,使每个 RTU 只需 2 个加法器和 1 个移位器即可.

同理, $c(05) \sim c(07)$ 与 $c(01) \sim c(03)$ 运算过程完全相同; $d(04)$ 与 $d(08)$ 的运算也与其大同小异,其最终输出结果只有 1 个而非 3 个.

(2) 行变换缓存单元(RRF)

设定外部输入数据传输位宽为 32 位,每次传送 4 个 8 位二进制初始图像数据. 对运算单元而言,每次运算需要 4 个时钟周期和 14 个输入数据,传输 14 个数据至少需要 4 个时钟周期. 由此,一次运算的数据传输时间和数据计算时间相同.

因此,可以将 RRF 设计为双端口模式,这样在并行计算单元计算的同时,下一次需要运算的数据也在不断进入缓存器,在本次计算完成的同时,下一次待运算的数据也已准备好. 以每次计算的数据块为单位,数据实现流水线操作.

3.3 行列变换并行实现

(1) 数据输入模式

由之前分析可知每次行变换能得到 8 个结果,为了提高行列处理并行性,行变行将每次输入两行数据,每行各 8 个数据,每次行变换能得到两行且每

行 4 个的输出结果.

从图 5 可知,初始数据经过边缘处理之后以两行为单位输入行缓存(RRF)中,行扫描数据按照列的方向进行,这样能更快速得到列变换所需数据,提高行列并行度.

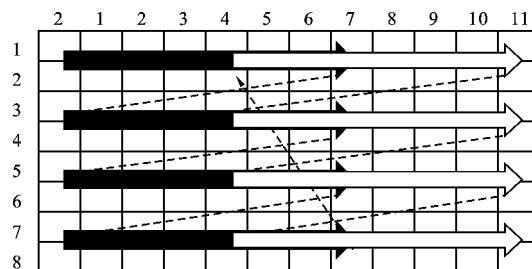


图5 数据扫描示意图

(1) 行变换结果存储(RTRM)

将行变换结果存储在 RTRM 中, RTRM 也设计为双端口模式. 当行变换结果累积到一定数目,则开始将 RTRM 中的行变换结果输入到列缓存.

(2) 行变换结果存储(RTRM)

将行变换结果存储在 RTRM 中, RTRM 也设计为双端口模式. 当行变换结果累积到一定数目,则开始将 RTRM 中的行变换结果输入到列缓存(CRF)中,从而进行列变换. 而后的行变换结果还将不断存储在 RTRM 里;当本次列变换完成时,下一次运算的数据已经在 RTRM 中准备好,行列变换可以并行实现. 数据流程如附表所示,可看出在经过前期短暂无列变换结果之后,每 4 个时钟周期能产生 8 个二维变换结果.

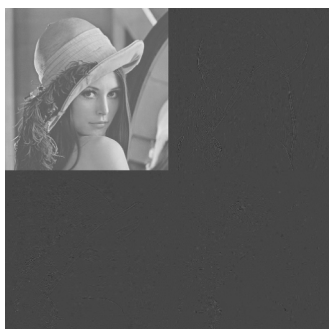
由此可知本文设计通过对 4 个行变换并行运算单元和 4 个列变换并行运算单元的有效控制实现了不同行(列)变换之间的完全并行、行变换与列变换的完全并行.

4 实验与分析

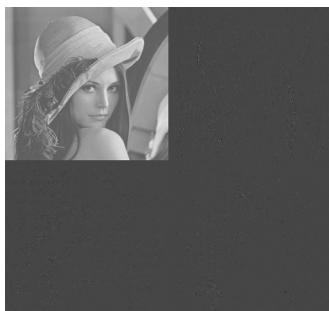
4.1 算法仿真

对本文算法及其优化方案采用 Matlab 进行理论实验分析. 通过对整数 5/3 提升小波基本算法和变形后的算法分别编程对比,变换对象是 256×256 大小的 lena 灰度图像. 变换结果见图 6 所示. 基于一般视觉比较(a)和(b)两图基本无差别. 另外选用图像质量评价算法均方差 MSE 方法来对两幅图进行比较. 对于 MN^2N 大小的两幅图像而言, MSE 的计算公式为:

$$MSE = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (f_1(i,j) - f_2(i,j))^2}{M \times N} \quad (7)$$



(a) 算法变形前



(b) 算法变形后

图 6 算法仿真结果

在 Matlab 中计算结果为 $MSE = 0.0249$, 可看出变换之后两者效果基本一致, 因此可认为改进后

的小波变换具有和原小波变换相同的功能.

4.2 硬件实现结果分析

对描述的二维整数 $5/3$ 提升小波变换实现结构, 进行硬件实现. 使用 Verilog 语言进行编程, 在 Modelsim 下验证系统功能正确, 而后在 Xilinx 公司的 ISE12.2 下选用 Virtex6 XC6VLX75T 器件进行综合和实现, 综合后最高时钟频率达到 394.415Mhz, 硬件使用情况说明见表 1.

表 1 资源使用情况表

资源	使用情况
Number of slice	971 out of 93120 1%
Number of LUTs	1342 out of 46560 3%

将本文中的结构与已有结构进行对比, 结果见表 2. 其中“N”的含义是设定研究对象为 $N \times N$ 大小的图像; “Ls”代表系统延迟; “P”代表运算结构; “R”代表行运算结构 RP; “C”代表列运算结构 CP.

硬件消耗主要集中在运算单元、控制模块和片上存储, 从表中看出, 本设计虽然在硬件消耗上较之其他结构更大一些, 但是却能取得更高的处理速度和工作频率.

表 2 各种结构对比结果

设计方案	移位器	加法器	计算时间	结构	控制复杂度	片上存储/byte	最高频率/MHz
文献[1]	2	4	N^2	P	简单	$N^2 + N$	70.49
文献[4]	2	4	$N^2/4$	P	复杂	N^2	112
文献[5]	1	2	$N^2 + Ls$	RC	简单	N^2	82.61
本文方案	4	8	$N^2/2 + Ls$	RC	简单	$N^2 + 4(N+3)$	394.4

5 结束语

本文提出了 $5/3$ 整数提升小波变换在并行运算单元上的硬件设计新思路, 并详细介绍了算法改进和硬件实现过程. 该方案能够实现算法较大运算速度提升.

参考文献:

- [1] 王承君, 熊承义. 二维 $5/3$ 小波变换的结构设计与 FPGA 实现[J]. 中南民族大学学报, 2006, 25(4): 50—53.
- [2] Al-Sulaifanie A K, Ahmadi A, Zwolinski M. Very large scale integration architecture for integer wavelet transform[J]. Computers & Digital Techniques, IET, 2010, 4(6): 471—483.
- [3] Ibrahim Saeed, Herman Agustawan. High-speed and power efficient lifting-based VLSI architectures for

two-dimensional discrete wavelet transform[C]// proceedings of the 2nd AICMS, Tronoh; Petronas, 2008: 998—1005.

- [4] Marshall P, Kozicki S, Zewail R, et al. Reconfigurable, fully scalable integer wavelet transform unit for JPEG2000 [C]// proceedings of Electrical and Computer Engineer, Canada; Saskatoon, 2005: 798—801.
- [5] 刘军伟, 饶妮妮. 提升小波变换的 FPGA 设计与实现[J]. 微计算机息, 2005, 21(10): 132—134.

作者简介:

温琳卉 女, (1988—), 硕士研究生. 研究方向为集成电路设计.

谢 憬 男, (1980—), 助理研究员. 研究方向为 Soc 设计方法与实现.

王国兴 男, (1974—), 副教授, 硕士生导师. 研究方向为高性能数模混合集成电路设计.