文章编号:1001 - 2486(2005)06 - 0048 - 05

基于提升算法的二维 DWT 高效 VLSI 实现结构*

康志伟[1],颜福权[1],何怡刚[2]
(1.湖南大学 计算机与通信学院,湖南 长沙　410082; 2.湖南大学 电气与信息工程学院,湖南 长沙　410082)

摘　要:以 CDF9/7 小波为例构造了一种二维 DWT 变换的高效 VLSI 结构。采用改进的提升算法,减少了关键路径上的延时。把乘法器系数表示为 CSD 形式,将乘法优化为最少的移位加操作。提出了一种行变换和列变换同时进行的方法和实现结构,并且整个结构采用流水线处理。通过 VHDL 的行为级仿真,得到的数据和软件仿真的结果相同,证明了该结构的正确性。和其它结构相比,该结构处理速度更快,并且硬件利用率可达 100%。

关键词:提升算法;小波变换;二维 DWT;VLSI;并行结构
中图分类号:TN911.7　　文献标识码:A

An Efficient VLSI Architecture for Two-dimensional DWT Based on the Lifting Scheme

KANG Zhi-wei[1], YAN Fu-quan[1], HE Yi-gang[2]
(1. College of Computer and Communication, Hunan Univ., Changsha 410082, China;
2. College of Electrical and Information Engineering, Hunan Univ., Changsha, 410082, China)

Abstract: An efficient VLSI architecture for two-dimension DWT is proposed and illustrated in detail for the CDF9/7 wavelet transform. The improved lifting scheme is adopted to reduce the critical path delay. Coefficients of the multipliers are transformed into CSD forms and then the multiplications are substituted by shift-add operations. The row transform and column transform are running simultaneously and pipeline design is used to optimize the architecture. This architecture is implemented through behavioral VHDL. The results are identical with those of the software simulation, and thus the validity of this architecture is proved. Compared with other architectures, this one has the advantages of faster computation time and almost 100% hardware utilization.

Key words: lifting scheme; wavelet transform; two-dimensional DWT; VLSI; parallel architecture

离散小波变换(DWT)在语音、图像等信号处理中有着广泛的应用,在 JPEG2000 标准中就推荐采用 5/3 和 9/7 小波来分别进行无损和有损图像压缩,取代基于 DCT 变换的图像压缩,并且还推荐采用提升算法来实现。小波变换的优点是能够对信号进行由粗到精的多尺度分析,但是小波变换的计算量很大,因此对于很多需要实时计算 DWT 的应用,寻求一种高效的 VLSI 实现结构是非常重要的。Mallat 算法是计算 DWT 的传统的方法,1996 年 Sweldens 提出了提升算法(lifting scheme)[1],提升算法提出的初衷是用来构造第二代小波变换,但是第一代小波变换也能够通过提升算法来实现[2],相比于传统的卷积方法,提升算法有计算量小,原位运算,易于实现整数小波变换等优点。可分离的二维 DWT 的计算一般是先沿行(列)的方向进行一维 DWT,对得到的结果再沿列(行)的方向进行一维 DWT。现在已经提出了不少实现 DWT 的 VLSI 结构[3~5],这些结构都是当第一级的行变换完成后,再进行第一级的列变换,当完成第一级的变换之后再进行第二级、第三级的变换,因为行变换和列变换的时间是相同的,但是各级变换所需的时间随着分解级数的提高而以 1/4 的指数规律递减,本文提出了一种在适当增加硬件的基础上行列变换同时进行的结构,这种结构硬件利用率可达 100%,大大缩短了处理的时间。另外还采用了文献[5]中提出的改进的提升算法,减少了关键路径上的延时。针对 CDF9/7 小波变换,本文还提出把提

* 收稿日期:2005 - 04 - 20
基金项目:国家自然科学基金资助项目(50277010);高校博士点基金资助项目(20020532016)
作者简介:康志伟(1962—),男,副教授。

升结构中的常系数乘法器的系数表示为 CSD 形式,把常系数乘法优化为最少的移位加操作,大大地提高了处理的速度。

1 提升算法及其改进

有限长的滤波器构成的小波滤波器组都可以分解为一系列的提升步骤[2],其基本的过程是把滤波器组的多相矩阵分解为一系列的上三角、下三角矩阵和一个对角矩阵的乘积。CDF9/7 小波有着优秀的图像压缩性能,其分解端低通滤波器 $\tilde{h}$ 长度为 9,重构端低通滤波器 H 的长度为 7,高通滤波器 $\tilde{g}$ 和 g 在 $z=1$ 有四阶零点,即分解和重构小波都有四阶消失矩。其分解端多相矩阵分解形式为:

$$\tilde{P}(z)=\begin{bmatrix}1 & \alpha(1+z^{-1})\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \beta(1+z) & 1\end{bmatrix}\begin{bmatrix}1 & \gamma(1+z^{-1})\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \delta(1+z) & 1\end{bmatrix}\begin{bmatrix}\zeta & 0\\ 0 & 1/\zeta\end{bmatrix} \tag{1}$$

其中,$\alpha=-1.586134342$,$\beta=-0.052980118$,$\gamma=0.882911075$,$\delta=0.443506852$,$\zeta=1.230174105$。从多相矩阵的分解形式可以看出,该小波变换可以分解为两步提升过程和一个尺度变换过程,每一步的提升过程又包括一次提升和一次对偶提升。另外还可以看出在做提升和对偶提升的时候都是一个通道的两个相邻的值相加后再乘以一个常数然后再和另一个通道的对应的值相加,显然该通道的处理时间要大大地多于另外一个通道的时间,为此文献[5]对式(1)进行恒等变换,提出了一种改进的提升形式如下:

$$\tilde{P}(z)=\begin{bmatrix}1 & 1+z^{-1}\\ 0 & a\end{bmatrix}\begin{bmatrix}b & 0\\ 1+z & 1\end{bmatrix}\begin{bmatrix}d & 0\\ 1+2 & 1\end{bmatrix}\begin{bmatrix}k1 & 0\\ 0 & k0\end{bmatrix} \tag{2}$$

其中,$a=1/\alpha=-0.6305$,$b=1/\alpha\beta=11.9000$,$c=1/\beta\gamma=-21.3781$,$d=1/\gamma\delta=2.5538$,$k1=\alpha\beta\gamma\delta\zeta=0.0405$,$k0=\alpha\beta\gamma/\zeta=0.0603$,这样两个通道的处理时间更加均衡,减少了关键路径上的延时,该提升分解可以表示为:

$$\begin{cases}H^{(1)}(n)=a\times x(2n+1)+x(2n)+x(2n+2)\\ H^{(2)}(n)=c\times H^{(1)}(n)+L^{(1)}(n)+L^{(1)}(n+1)\\ H(n)=k0\times H^{(2)}(n)\\ L^{(1)}(n)=b\times x(2n)+H^{(1)}(n)+H^{(1)}(n-1)\\ L^{(2)}(n)=d\times L^{(1)}(n)+H^{(2)}(n)+H^{(2)}(n-1)\\ L(n)=k1\times L^{(2)}(n)\end{cases} \tag{3}$$

2 把乘法优化为最少的移位加操作

在 CDF9/7 小波变换中的乘法都是常系数乘法,因此可以把这些乘法优化为移位加操作,这样可以大大地减少芯片的实现面积并且提高运算的速度。为了减少移位加操作的次数,本文提出把乘法器系数表示为 CSD(canonical signed digit)形式,这种表示形式中把系数 X 表示为:$X=x_{n-1}x_{n-2}\cdots x_1x_0$,$x_i$,($i=0,1,\cdots,n-1$)为 0,1 或者 -1,且 $X=\sum_{i=0}^{n-1}x_i2^i$。对于 CDF9/7 的改进提升算法实现方案中的系数表示为 CSD 形式,并保留 5 位非零位的结果为:$a=0.-0-000-0101$,$b=10-00.00-010-$,$c=-01010.101$,$d=10.100100-00-$,$k1=0.0000101010-00-$,$k0=0.00010000-00-00001001$,其中“-”表示 -1,这样我们就能把每个乘法转换为五个移位之和的形式。例如对于 x 和 a 的乘积就可以等效地表示为:$x\times a=(-x)\gg1+(-x)\gg3+(-x)\gg7+x\gg9+x\gg11$;“$\gg$”表示右移操作,即对于 CSD 表示形式中的为 1 的项做相应的左移或者右移,对于为 -1 的项则把输入的数取相反数后再做相应的移位。由于采用 CSD 形式的表示能在满足给定的精度条件下用最少的非零比特位来表示所需的常数,因此这种方法能把乘法优化位最少的移位加操作。在 Matlab 中利用浮点乘法实现改进的提升算法来对标准图像 Lena 和 Camera 进行多级 CDF9/7 小波分解,得到分解的中间结果和最后的分解结果的最大值在 -8192 ~ 8191 之间,因此本文采用整数部分为 14 位的定点数,另外小数部分采用 12 位的长度[6]。采用 5 位非零位 CSD 表示利用后面提出的结构用 VHDL 仿真得到的结果和利用 Matlab 实现得到的结果的相对误差(表 2)。可见采用五位的移位加实现有足够的精度。

3 高效二维 DWT 结构

二维 DWT 的总体结构如图 1 所示，当进行第一级变换的时候控制模块选择从输入端口接受数据，当进行第二级以后的变换时候，控制模块选择从存储模块读取上一级变换的 LL 子带数据。行变换的结果存入行缓存模块，这样就可以实现行列变换的并行运算，列变换结束后即得到了一级变换的所需结果，并将结果存入存储模块，然后根据需要结束或者进行下一级的变换。

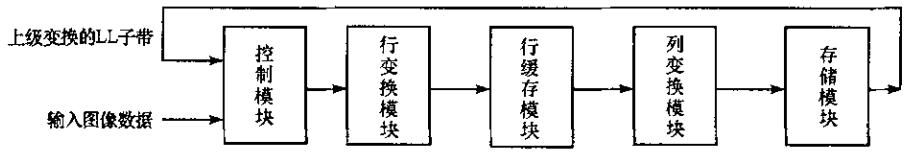

图 1 二维 DWT 总体结构

Fig.1 Overall architecture for 2D – DWT

3.1 行变换模块结构

根据式(3)，采用流水线设计方法对行变换结构进行优化，并且对边界处理采用内嵌扩展算法[7]，得到如图 2 所示的行变换结构。该行变换结构每个时钟输入两个数据，同时输出两个行变换结果数据，表 1 给出了流水线处理时序。

表 1 行变换结构的流水线操作时序

Tab.1 The pipelined time scheduling for the row transform architecture

时钟	输入	$H^{(1)}$	$L^{(1)}$	$H^{(2)}$	$L^{(2)}$	H	L
0	$xe(0), xo(0)$						
1	$xe(1), xo(1)$	$H^{(1)}(0)$					
2	$xe(2), xo(2)$	$H^{(1)}(1)$	$L^{(1)}(0)$				
3	$xe(3), xo(3)$	$H^{(1)}(2)$	$L^{(1)}(1)$				
4	$xe(4), xo(4)$	$H^{(1)}(3)$	$L^{(1)}(2)$	$H^{(2)}(0)$			
5	$xe(5), xo(5)$	$H^{(1)}(4)$	$L^{(1)}(3)$	$H^{(2)}(1)$	$L^{(2)}(0)$		
6	$xe(6), xo(6)$	$H^{(1)}(5)$	$L^{(1)}(4)$	$H^{(2)}(2)$	$L^{(2)}(1)$	$H(0)$	$L(0)$
7	$xe(7), xo(7)$	$H^{(1)}(6)$	$L^{(1)}(5)$	$H^{(2)}(3)$	$L^{(2)}(2)$	$H(1)$	$L(1)$
n	$xe(n), xo(n)$	$H^{(1)}(n-1)$	$L^{(1)}(n-2)$	$H^{(2)}(n-4)$	$L^{(2)}(n-5)$	$H(n-6)$	$L(n-6)$

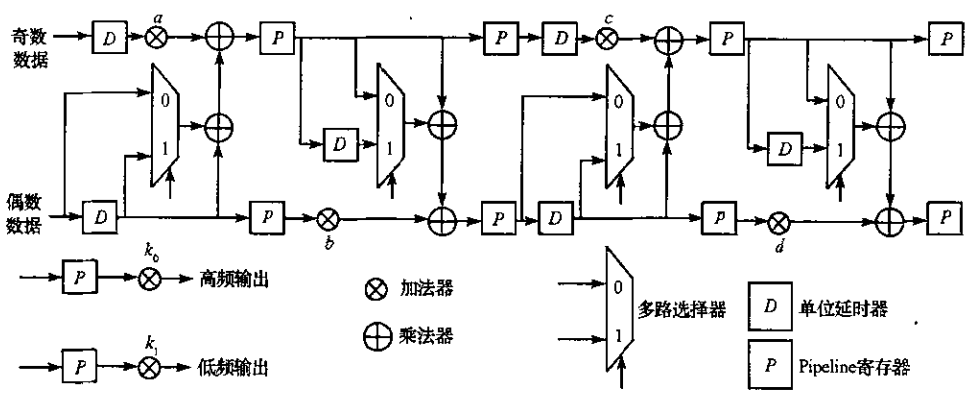

图 2 行变换模块结构

Fig.2 The architecture of row transform module

可见在每个提升和对偶提升及尺度变换之前加入两个 Pipeline 寄存器即可实现流水线操作，并且从

第六个时钟开始每个时钟得到两个输出结果,而在流水线操作情况下时钟的最短周期为每一级流水线的最长时间,这远远小于不采用流水线方式情况下从输入端到输出端的处理时间,因此大大提高了处理的速度。另外该结构的边界处理采用内嵌扩展算法,该方法无需对输入的有限长序列直接进行扩展,而是根据输入序列对称扩展后得到的中间结果 $H^{(1)}(n)$,$L^{(1)}(n)$,$H^{(2)}(n)$在边界上的扩展规律,直接对 $H^{(1)}(n)$,$L^{(1)}(n)$,$H^{(2)}(n)$进行扩展,可用图 2 所示的多路选择器来实现,当进行正常运算时,多路选择器的选择端为 0,当在边界处的时候将多路选择器的选择端置为 1 即可。

3.2 行缓存模块结构

为了行列变换同时进行,因此不能等到一列的数据全部产生后再进行列变换,而是在得到一列的两个输入数据后即开始进行列变换,但是在下一时钟由于没有该列的下两个输入数据因此不能计算该列的变换的下两个结果,而是计算下一行的列变换的结果。因此该列的两个变换结果计算完成后电路中的状态需要保存下来,以便下次再计算该列变换的下两个结果的时候恢复电路的状态。因为列变换和行变换的计算是相同的,只是输入不同而已,因此需要保存的状态为图 2 中每个延时单元和 Pipeline 寄存器的输入,因为再下一次计算的时候这些输入本来应该传输到延时单元和 Pipeline 寄存器的输出端参与计算,从图 2 可以看出共有 14 个这样的单元,但是在两步提升运算中的两个延时单元的输入与其下级的 Pipeline 寄存器的输入相同,因此需要保存的状态个数为 12 个。从表 1 可以看出,当输入是 $xe(n)$和 $xo(n)$的时候,这些需要保存的状态数据为:$xo(n)$,$H^{(1)}(n-1)$,$H^{(1)}(n-2)$,$H^{(1)}(n-3)$,$H^{(2)}(n-4)$,$H^{(2)}(n-5)$,$xe(n)$,$xe(n-1)$,$L^{(1)}(n-2)$,$L^{(1)}(n-3)$,$L^{(1)}(n-4)$,$L^{(2)}(n-5)$。当进行列变换时,除了需要上次处理后的电路的状态外还需要两个输入数据,因此计算某列的两个列变换值需要 2 个输入和 12 个上次的电路状态数据,总共需要 14 个数据,当处理的图像有 N 列的时候就需要 $14\times N$ 大小的存贮空间。另外为了缓冲行变换的结果,在行缓存模块中还需要设置可以容纳两行行变换结果 $2\times N$ 的存贮空间,因此总的行缓存模块的大小为 $16\times N$,其结构如图 3 所示。

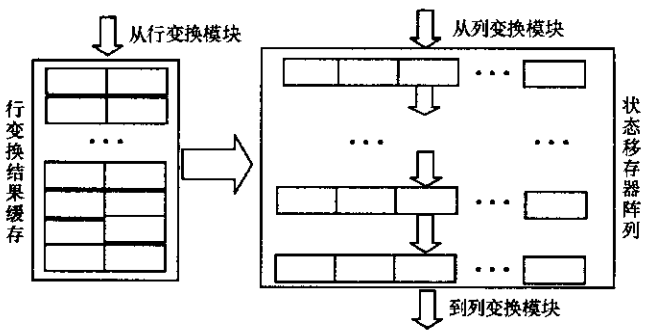

图 3 行缓存模块结构

Fig.3 The architecture of row buffer module

该行缓存模块由行变换结果缓存和状态移存器阵列组成,在完成一个偶数行和一个奇数行的行变换后,行变换结果缓存被充满,此时将这些行变换结果全部载入到状态移存器阵列每行的前两个单元作为进行列变换时的输入。状态移存器阵列的每一行存储了完成某列的两个列变换结果后电路的状态,因此该移存器阵列有 N 行,每行存储了 14 个数据,包括 2 个输入数据和 12 个状态数据。在每个时钟沿把从列变换模块来的电路状态数据存入移存器阵列的最后一行,并从第一行读出下一个列变换所需的状态数据,加载到列变换模块。采用移存器阵列的结构,在每个时钟沿的时候把高一行的状态数据移存到低一行的移存器中,这样每个时刻需要加载的状态数据都从第一行读出,需要保存的状态数据存入到最后一行。

行变换模块每个时钟向行缓存模块输出两个数据,一个低频数据,一个高频数据,而一个列变换模块要计算行变换的低通和高通两个通道的变换结果,一个时钟需要行变换的低通或者高通通道的两个分解结果,这样行变换模块的数据的输出速率和列变换模块要求的数据输入速率是相同的,所有模块可

以达到100%的硬件利用率。

3.3 列变换模块

列变换和行变换的运算是相同的,因此列变换模块和行变换模块的结构也很相似,只是列变换模块在每个时钟沿要把上一次的状态送到行缓存模块,并且从行缓存模块加载该时钟所需的输入和状态数据。反映在电路结构上即是列变换模块比行变换模块多了14根加载输入和状态的数据线和12根输出要保存状态的数据线,在此列变换的结构图从略。

4 试验结果

在Active HDL6.2中利用VHDL对本文提出的二维DWT实现结构进行了行为级的仿真,对256×256×8的Lena图像和Camera图像进行了4级小波分解,得到的结果和利用Matlab计算得到的结果的最大相对误差如表2所示,验证了该结构的正确性。

表2 硬件仿真相对于软件实现结果的相对误差最大值

Tab.2 The maximum of relative error between hardware simulation and Matlab implementation

图像	一级	二级	三级	四级
Camera	0.0975%	0.1928%	0.2884%	0.3800%
Lena	0.0966%	0.1918%	0.2848%	0.3799%

5 结 论

以JEPG2000标准推荐的CDF9/7小波为例,给出了一种二维DWT的高效VLSI结构,该结构和其它结构的不同之处在于行列变换同时进行,采用改进的提升算法,根据乘法器系数的CSD表示利用最少的移位加来实现乘法。行列变化同时进行,虽然增加了实现的硬件开销,但是得到的结构的硬件利用率仍然为100%,改进的提升算法减少了关键路径的延时,用最少的移位加来实现变换中的常系数乘法大大降低了所需的硬件资源,因此该结构具有快速高效的特点。

参考文献:

[1] Sweldens W. The Lifting Scheme: A Custom-design Construction of Biorthogonal Wavelets[J]. Applied and Computational Harmonic Analysis, 1996,3:186-200.

[2] Daubechies I, Sweldens W. Factoring Wavelet Transforms into Lifting Steps[J]. Fourier Analysis and Appli-cations, 1998,4(3):247-269.

[3] Lian C J, Chen K F. Lifting Based Discrete Wavelet Transform Architecture for JPEG2000[A]. Proceedings of the 2001 IEEE International Symposium on Circuits and Systems[C], Piscataway, USA, 2001:445-448.

[4] Movva S, Srinivasan S. A Novel Architecture for Lifting-based Discrete Wavelet Transform for JPEG2000 Standard Suitable for VLSI Implementation[A]. Proceedings of 16th International Conference on VLSI Design[C], New Delhi, India, 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[A]. 2004 International Conference on Communications, Circuits and Systems[C], Chengdu, China, 2004:728~732.

[6] Spiliotopoulos V, Zervas N D. Quantizing the 9/7 Daubechies Filter Coefficients for 2D DWT VLSI Implem-entations[A]. Proceedings of the 14th International Conference on Digital Signal Processing[C], Santorini, Greece, 2002:277~231.

[7] Tan K.C.B, Arslan T. An Embedded Extension Algorithm for the Lifting Based Discrete Wavelet Transform in JPEG2000[A]. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2002[C], 2002,4:3513-3516.