

# 一维离散小波/小波包 变换的 VLSI 结构\*

吴晓冬 李永明 陈弘毅

(清华大学微电子学研究所 北京 100084)

**摘要** 小波/小波包变换作为强有力的信号处理手段,正在越来越多的领域中得到了应用,因而其硬件实现也日益受到重视.本文针对小波/小波包变换在语音编码中的应用,给出了一维离散小波/小波包变换的 VLSI 结构.和现有的一些实现方案不同,该结构可用于不同支集长度小波、不同长度数据段、不同变换阶数,具有较大的通用性和可编程性,可作为多种处理系统的片上变换单元,亦可单片实现.

EEACC: 0230, 2570, 1265F

## 1 引言

小波变换是一种新的数学理论和方法,它是在傅氏变换的基础上发展起来的,其基本思想是将信号在小波函数系拓展成的空间上进行分解,从而得到信号在不同时间-频率空间(严格地说是时间-尺度空间)上的投影,它克服了傅氏变换时域分辨率差的缺点,在时域和频域同时具有较好的局部化特性,因而特别适于处理时变信号<sup>[1~3]</sup>.小波变换目前已开始广泛地应用于信号分析检测、图象处理、语音编码等领域的研究<sup>[4~6,13]</sup>.

随着小波变换应用的日益广泛,其硬件实现的研究也越来越受到重视. Knowles 最早给出了小波变换的实现结构<sup>[7]</sup>,他以一对并行滤波器 G、H(分别对应小波函数和尺度函数)来实现所有的计算,计算的中间结果由串入并出的寄存器队列来存储.后来, Lewis 和 Knowles 还给出了一种基于四系数小波函数的无乘法器的小波变换结构<sup>[8]</sup>,但这一结构过于依赖所用的小波函数,无任何灵活性. Parhi 等人则给出了基于格形结构的折叠式结构和位串行结构<sup>[9]</sup>,上述这些结构基本上都属于并行滤波器结构形式.后来又出现了 SMD (Single Instruction Multiple Data) 线性阵列和 SMD 网格结构<sup>[10]</sup>, Fridman 等人还从系统综合的角度给出了具有分布式存储和控制特点的脉动阵列结构<sup>[11]</sup>.但这些小波变换结构是从高层次计算的角度来考虑的,其基本单元很大,都是一个完整的滤波器,很不利于单片实现. Grzeszczak 等人给出了 8 位字长的 6 个系数的小波变换芯片结构设计<sup>[12]</sup>,但其面积为  $10\text{mm} \times 7\text{mm}$ ,只可用于阶数较低的滤波器,因而使用范围较窄,且无法作为片上处理单元.

\* 国家自然科学基金重点基金项目

吴晓冬 男,博士研究生,从事小波变换等数字信号处理技术及其 VLSI 实现的研究

李永明 男,副教授,从事大规模模拟集成电路设计和教学、语音信号分析合成集成电路研究设计

陈弘毅 男,教授,博士生导师,从事 ASIC 设计、信号处理的 VLSI 实现及设计方法学研究

1997-11-26 收到, 1998-02-12 定稿

最早的小波变换芯片是 AWARE 公司的小波变换处理器 (WTP), 它使用 4 个系数的滤波器, 可以计算一维小波变换的前向和后向变换, 并具备级联能力以应用于高阶滤波器。但它是完全用户控制方式的, 使用时需大量的用户编程, 因而很不方便。Analog Devices 的 ADV 601 是基于小波变换的图象压缩芯片, 但其内部结构未见公开。

综上, 目前小波变换的硬件实现研究仍处于起步阶段, 现有的这些方案都有很强的针对性, 其具体结构大都受所用小波函数的限制, 往往都只适用于阶数较低的滤波器和较低的分解级数, 不适用于阶数较高的滤波器和较深的分解级数, 而且, 由于小波包变换较较小波变换更为复杂, 至今尚未有硬件实现的报道。而在基于小波变换的图象和声音处理算法中, 通常都使用阶数较高的小波滤波器和较深的分解级数, 而且小波包的使用越来越广泛<sup>[4~6, 13]</sup>。因此, 迫切需要一种适应性和灵活性较高的小波/小波包变换 VLSI 结构。

本文结合小波/小波包变换在语音编码中的应用, 给出了一种可编程的、适用性较广的一维小波/小波包变换的 VLSI 实时实现结构。

## 2 小波变换和小波包变换的基本原理

小波变换的基本原理是这样的, 给定信号  $f(x) \in L^2(R)$  及小波函数  $\Psi(x)$ , 有如下变换和反变换:

$$w_{j,k} = \int_{-\infty}^{+\infty} f(x) 2^{j/2} \Psi(2^j x - k) dx \quad (1a)$$

$$f(x) = \sum_{j,k} w_{j,k} 2^{j/2} \Psi(2^j x - k) \quad (1b)$$

其中  $w_{j,k}$  为信号在  $2^j$  尺度下  $k$  时刻的小波变换值。在本文中我们只考虑离散正交小波变换, 其输入也为离散信号。目前最通用的正交小波变换快速算法是 Mallat 算法, 即金字塔式算法, 它是一种迭代算法, 即先计算出第一级小波变换, 然后在此基础上计算下一级小波变换, 如此重复下去, 见式(2)

$$c_{j+1,m} = \sum_k h_{k-2m} c_{j,k} \quad (2a)$$

$$d_{j+1,m} = \sum_k g_{k-2m} c_{j,k} \quad (2b)$$

其中  $c_j, d_j$  分别是信号在  $2^j$  尺度下分解得到的近似分量和细节分量;  $c_0$  表示输入的离散信号,  $h_k, g_k$  分别为尺度函数和小波函数对应的滤波器系数。目前的一些硬件实现方案都是基于这一算法的, 每级变换都只做一次  $G, H$  滤波, 其结果的一半 ( $H$  的输出) 做为下一级滤波的输入, 另一半 ( $G$  的输出) 即为小波变换最终结果的部分, 这样滤波运算量每级都比前级减半, 总运算量受分解级数  $J$  的影响随  $J$  的增大而减小, 中间数据存储量随分解级数线性增长。

小波包变换则是小波变换的进一步推广, 它不仅对小波变换中的低频部分进一步分解, 而且对高频部分也做进一步分解, 计算公式见式(3), 其中表示方法略有不同,  $c$  的上标  $d$  表示小波包分解级数, 下标第一部分表示分解得到的频段编号, 其取值范围为 0 至  $2^d - 1$ , 第二部分为离散时间变量

$$C_{2^d j, m}^{d+1} = \sum_k h_{k-2m} C_{j, k}^d \quad (3a)$$

$$C_{2^d j+1, m}^{d+1} = \sum_k g_{k-2m} C_{j, k}^d \quad (3b)$$

因而, 和小波变换不同, 小波包变换每次滤波的输出结果全部作为下级滤波的输入继续进行变换运算, 所以每级变换运算量相同, 总运算量与分解级数为级性关系, 而中间数据存储量随分解级数指数增长

### 3 小波/小波包变换的 VLSI 实现方案

我们在考虑小波/小波包变换的硬件实现结构时, 提出了以下一些要求:

- (1) 由于不同场合可能使用不同的小波函数, 故要求系统能对不同的小波函数均适用, 尤其是支集较长的小波函数, 设小波支集长度不超过 20;
- (2) 在信号处理、语音压缩等应用中, 信号的处理都是分帧进行的, 故要求系统能对一定长度范围内的数据段进行变换, 设长度为 2 的整数次方, 设数据段的最大长度不超过 512;
- (3) 由于应用的不同, 进行变换时的分解级数会有所不同, 故要求系统能按所要求的变换级数进行变换, 设变换级数最大不超过 7;
- (4) 由于希望能作为片上处理单元, 故要求系统面积尽可能小

鉴于上述要求, 对已有的一些小波变换实现方案进行分析可知, 它们大都是受具体滤波器限制, 且只能用于小波变换, 无法满足上述要求

现有的离散二进小波变换实现结构, 基本上都是利用了小波变换在各倍频程输出数据量减半的特性, 在时间上进行插空和按一定要求重新排序, 从而完成实时运算的, 这就是 Vishwanath 的所谓的迭代金字塔算法 (RPA)<sup>[10]</sup>, 见图 1(a), 设数据表示为  $X_{k,i}^j$ , 其中  $j$  表示分解级数;  $j=0$  表示原始数据,  $k$  表示数据所处频段的序号 (从低到高),  $i$  表示时间序号值

但对于小波包变换而言, RPA 算法不再有效 由于小波包变换对频带是均匀划分的, 各频段输出的数据量是相同的, 而且各输出数据在时间上是并行的, 即各数据是同时计算得到的, 不存在时间先后问题, 见图 1(b), 无法象 RPA 算法那样进行插空和重排 而且, 进行小波包变换时, 中间数据的量非常大 对此, 解决的办法如果只是与变换结构相对应, 即增加硬件资源 (存储单元和处理单元) 的数量, 是行不通的 因为对于小波变换而言, 资源的增加是随分解级数线性增长的, 而对小波包变换而言, 增加则是随分解级数而指数增长, 设分解级数为  $J$ , 小波对应的滤波器阶数为  $L$ , 处理帧长为  $N$ , 则对小波变换而言, 所需滤波器数量为  $2 \times J$ , 存储量为  $2 \times J \times L$  个字, 而对小波包变换而言, 所需滤波器数量为  $2^{J+1} - 2$ , 存储量为  $(2^{J+1} - 2) \times L$  个字 所以必须从系统结构设计的角度来寻找解决这些问题的途径

#### 3.1 存储系统的设计

传统的折叠结构解决了处理单元数量增长的问题, 但存储量大的问题一直未能解决, 无论是 Knowles 的存储队列, 还是 Chakrabarti 等人的存储阵列 SMD 线性阵列和 SMD 网格结构, 以及 Fridman 等的脉动阵列结构则无论从存储单元数量还是从处理单元数量的角度都未能解决问题, 这些结构只是从高层次算法和并行计算的角度来处理问题 尽管其结构都是可 scalable 的, 但不适于甚至不可能以芯片方式实现, 尤其是分解级数和滤波器阶数都比较大的时候

基于小波/小波包变换的声音编码算法中<sup>[4~6,13]</sup>, 使用的小波滤波器阶数通常接近 20, 分解级数通常可达 6 或 7, 在这种情况下现有各种小波变换实现方案都很难满足要求

考虑到上述问题, 并结合对变换系统的要求, 我们在存储系统设计中采用了两组缓冲区

原始数据(j=0)	...	$X^0_{0,0}$	$X^0_{0,1}$	$X^0_{0,2}$	$X^0_{0,3}$	$X^0_{0,4}$	$X^0_{0,5}$	$X^0_{0,6}$	$X^0_{0,7}$	...
第一级分解(j=1)		$X^1_{0,0}$	$X^1_{0,1}$	$X^1_{0,2}$	$X^1_{0,3}$					
		$X^1_{1,0}$	$X^1_{1,1}$	$X^1_{1,2}$	$X^1_{1,3}$					
第二级分解(j=2)				$X^2_{0,0}$			$X^2_{0,1}$			
				$X^2_{1,0}$			$X^2_{1,1}$			
第三级分解(j=3)									$X^3_{0,0}$	
									$X^3_{1,0}$	

(a)

原始数据(j=0)	...	$X^0_{0,0}$	$X^0_{0,1}$	$X^0_{0,2}$	$X^0_{0,3}$	$X^0_{0,4}$	$X^0_{0,5}$	$X^0_{0,6}$	$X^0_{0,7}$	...
第一级分解(j=1)		$X^1_{0,0}$	$X^1_{0,1}$	$X^1_{0,2}$	$X^1_{0,3}$					
		$X^1_{1,0}$	$X^1_{1,1}$	$X^1_{1,2}$	$X^1_{1,3}$					
第二级分解(j=2)				$X^2_{0,0}$			$X^2_{0,1}$			
				$X^2_{1,0}$			$X^2_{1,1}$			
				$X^2_{2,0}$			$X^2_{2,1}$			
				$X^2_{3,0}$			$X^2_{3,1}$			
第三级分解(j=3)									$X^3_{0,0}$	
									$X^3_{1,0}$	
									$X^3_{2,0}$	
									$X^3_{3,0}$	
									$X^3_{4,0}$	
									$X^3_{5,0}$	
									$X^3_{6,0}$	
								$X^3_{7,0}$		

(b)

图 1 小波及小波包变换的结果输出顺序

其中斜体部分表示要输出的结果: (a)小波变换RPA 算法, (b) 小波包变换

交替工作的乒乓式结构, 即一组缓冲区用于数据的输入和输出, 另一组用于进行变换, 两组缓冲区轮换工作, 见图 2, 只要完成一帧数据变换所需时间小于其输入输出时间, 即可保证对外部系统的实时处理

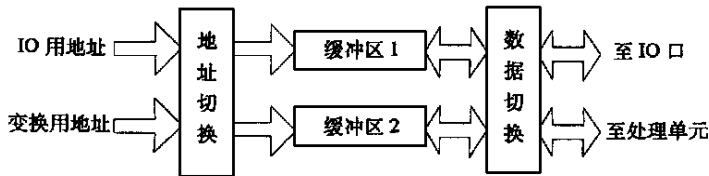


图 2 双缓冲区的乒乓式处理结构

有了上述的乒乓式处理结构, 就为变换处理赢得了时间, 可以用一帧的时间来集中处理一帧的数据 这对于小波包变换这样的数据依赖关系较强的计算来说是有利的, 在得到一帧输入数据后再去考虑如何实现变换就比较灵活了.

由于是分帧处理, 对于存储量的需求也发生了变化, 所需的中间数据存储量不再与滤波器长度和分解级数紧密相关, 而只和帧长有关系 依常规的做法, 变换时将缓冲区再分为两部分, 对输入数据进行一级小波/小波包分解, 结果存于缓冲区的第二部分, 然后对一级分解的结果再进行第二级分解, 结果存入缓冲区的第二部分, 如此用两部分缓冲区交替工作, 即

可实现小波/小波包变换 这种做法与软件编程的思路是相一致的,其结果可以按要求的任意顺序进行排列输出,控制电路也相对简单,且很容易处理边界效应

但由于变换时使用的两缓冲区,其每一部分的大小都与帧长相对应,使得存储量依然可观,以 512 点帧长为例,当字长为 16 时,所需存储量为  $16 \times 512 \times 2 = 16\text{kb}$ ,尽管工艺实现不成问题,但这对于变换器作为片上处理单元是不利的

考虑到进行小波包变换时,每次依时间顺序移动两个数据点,产生两个变换结果,是有可能采用同一个缓冲区来进行处理的,而不必使用两部分缓冲区 但这样的问题在于,一来数据位置原来的规整性被打乱,变换数据的取和存都有严格的位置限制,二来进行变换时边界效应难以考虑,因为每次取出的原始数据都会被得到的结果所替代,而这些数据很可能被再次用到

### 3 2 地址及边界的处理

对于问题一,以 8 点数据的三级小波包变换为例,设数据仍表示为  $X_{k,i}^j$ ,其中  $j$  表示分解级数, $j = 0$  表示原始数据, $k$  表示数据所处频段的序号(从低到高), $i$  表示时间序号值 在各分解级数(即不同尺度)下变换结果的理想

原始数据 ( $j=0$ )  $X_{0,0}^0 X_{0,1}^0 X_{0,2}^0 X_{0,3}^0 X_{0,4}^0 X_{0,5}^0 X_{0,6}^0 X_{0,7}^0$   
 第一级分解( $j=1$ )  $X_{1,0,0}^1 X_{1,0,1}^1 X_{1,0,2}^1 X_{1,0,3}^1 X_{1,1,0}^1 X_{1,1,1}^1 X_{1,1,2}^1 X_{1,1,3}^1$   
 第二级分解( $j=2$ )  $X_{2,0,0}^2 X_{2,0,1}^2 X_{2,1,0}^2 X_{2,1,1}^2 X_{2,2,0}^2 X_{2,2,1}^2 X_{2,3,0}^2 X_{2,3,1}^2$   
 第三级分解( $j=3$ )  $X_{3,0,0}^3 X_{3,1,0}^3 X_{3,2,0}^3 X_{3,3,0}^3 X_{3,4,0}^3 X_{3,5,0}^3 X_{3,6,0}^3 X_{3,7,0}^3$

(a)

原始数据 ( $j=0$ )  $X_{0,0}^0 X_{0,1}^0 X_{0,2}^0 X_{0,3}^0 X_{0,4}^0 X_{0,5}^0 X_{0,6}^0 X_{0,7}^0$   
 第一级分解( $j=1$ )  $X_{1,0,0}^1 X_{1,1,0}^1 X_{1,0,1}^1 X_{1,1,1}^1 X_{1,0,2}^1 X_{1,1,2}^1 X_{1,0,3}^1 X_{1,1,3}^1$   
 第二级分解( $j=2$ )  $X_{2,0,0}^2 X_{2,2,0}^2 X_{2,1,0}^2 X_{2,3,0}^2 X_{2,0,1}^2 X_{2,2,1}^2 X_{2,1,1}^2 X_{2,3,1}^2$   
 第三级分解( $j=3$ )  $X_{3,0,0}^3 X_{3,4,0}^3 X_{3,2,0}^3 X_{3,6,0}^3 X_{3,1,0}^3 X_{3,5,0}^3 X_{3,3,0}^3 X_{3,7,0}^3$

(b)

理想排放顺序见图 3(a),而采用单缓冲区后得到的排放顺序见图 3(b),各频带的数据是交叉排放的 这在一定意义上带来了不便,但可以通过在输出端重新调整来恢复原顺序

对新顺序的分析表明,数据排序有以下规律可供取数据时利用:

- (1) 当分解级数为  $j$  时,同一频段内相邻数据的间隔为  $2^j$ ;
- (2) 每一频段第一个样点的位置为对应该频段序号的二进制数的位反转值,例如当位数为 2 时,有 00 00,01 10,等等,当位数为 3 时,有 000 000,001 100,依次类推,位数与变换阶数  $j$  相对应

于是可以按如下方法来从数据的理想排序位置得到实际排序位置,即先确定该数据所在频段,找出该频段的第一个数据的起始位置,然后依该频段数据间的间隔来确定该数据的实际位置,地址转换的电路见图 4 当  $ctl$  为 1 时,表示新频段起始地址计算,控制多路器选择频段号位反转值, $ctl$  为 0 时表示同频段内的地址计算,控制多路器选取上次地址值

对于问题二,仔细分析小波包变换过程中数据被重复利用的情况可知,每一频段中被重复用到两次或以上的数据的数量最多为  $(L - 2)$ ,为此,可以采用一组小的备份区来临时备份这些可能被重复用到的数据,相对于处理帧长而言  $L$  很小,因而这部分备份区的大小是可以忽略的

其工作过程为,当第一次从缓冲区中取出数据到处理单元时,如果需要,则同时向备份区做备份,当再次用到这些数据时从备份区而不是从缓冲区来

图 3 小波包变换的结果排放顺序  
(a) 理想顺序, (b) 实际顺序

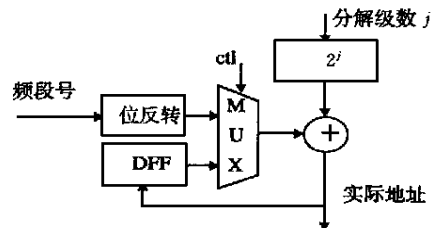


图 4 地址转换电路

取

### 3.3 运算处理单元的设计

小波/小波包变换系统的运算处理单元主要由乘法器和加法器组成, 用于完成滤波计算, 通常的处理单元(PE)为一个或一对完整的滤波器<sup>[7, 9, 10~12]</sup>, 其复杂度和小波函数有关, 小波函数的支集越长, 则对应的滤波器阶数越高, 从而处理单元的复杂度越高

为了解决滤波器数量的问题, 我们在系统中采用了传统的折叠方式来处理, 但这依然不够, 由于希望系统能用于阶数较高的小波滤波器, 如果按通常的结构来设计, 仅一个滤波器就将占去大部分芯片面积, 那么芯片面积将非常大, 这是不现实的, 这也是为什么现有的各种结构不适于单片实现以及不适用于阶数较高的滤波器的原因 由于采用了双缓冲区的乒乓式结构来进行变换, 可以考虑在时间上进一步复用, 即将折叠原理进一步用于滤波器内部, 而不仅仅是用于滤波器之间 这样做降低了并行度, 不利于提高运算速度, 但由于我们采用了乒乓切换的工作方式, 允许以较低的运算速度来完成实时变换, 因而即使采用单乘加单元也是可行的, 见图 5(a). 以声音信号为例, 采样频率为 44.1kHz, 设帧长为 512, 滤波器阶数为 20, 每个变换数据要计算约 20 次乘加, 每级小波包分解要计算  $512 \times 20$  次乘加得出 512 个变换数据, 则 6 级小波包分解的总乘加次数约为  $512 \times 6 \times 20$ , 而一帧数据的变换处理时间不应超过  $(512/44.1) \text{ms}$ , 则每次乘加处理的时间不能超过  $1/(6 \times 20 \times 44.1 \text{k}) \approx 0.2 \mu\text{s}$ , 这一要求容易达到, 以满足实时处理的要求 当要进一步提高速度时, 可以通过提高并行度来改善, 由于算法的并行性, 高通滤波器和低通滤波器作用于同一组输入数据, 其输出是可以同时计算得到的 这可使总计算时间缩短一倍, 其代价是运算单元数量增加一倍, 见图 5(b), 例如对图象进行小波变换时, 设图象大小为  $512 \times 512$ , 帧速率为 25, 则每帧图象的小波变换处理时间不能超过  $1/25 \text{s}$ , 而总的乘加次数约为  $512 \times 512 \times 2 \times L$ , 当滤波器阶数  $L$  为 6 时, 采用有两个乘加器的处理单元时每次乘加处理时间不能超过约 25ns, 用高速乘法器和流水线结构是可以达到这一要求的

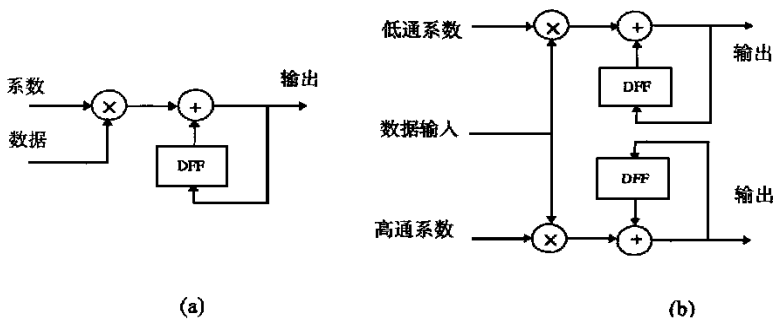


图 5 运算单元

(a) 单乘加器运算单元, (b) 双乘加器运算单元

当采用一组乘加单元时, 数据的馈给要另外处理, 因为低通和高通滤波器用的是相同的数据, 因此计算出的结果不能马上写回缓冲区, 而采用双乘加单元时则不存在这一问题

### 3.4 系统的整体结构

系统的整体结构示意图见图 6 当需要用到每一级变换的中间结果时, 只需在中间结果存回缓冲区的同时将其对外输出即可.

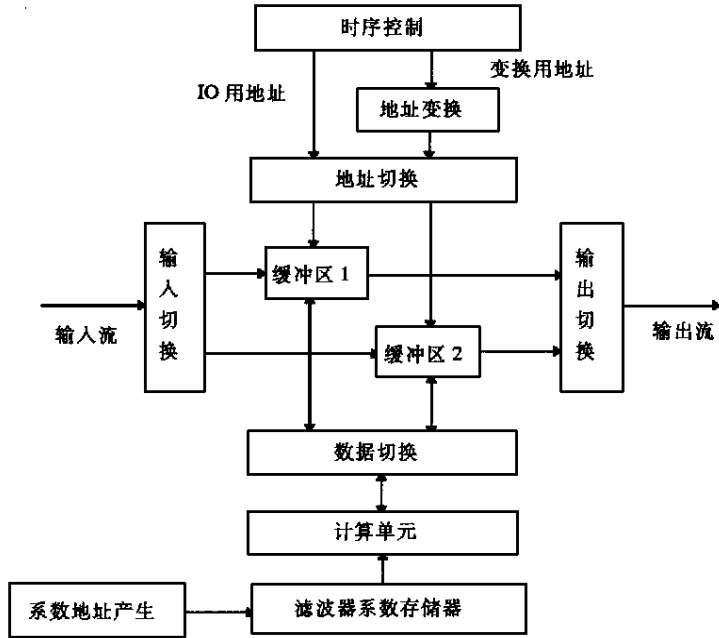


图 6 系统整体结构示意图

和现有的几种小波变换方案相比较, 本结构运算单元数量少, 存储量小, 不仅可实现小波变换, 而且可实现小波包变换, 适于VLSI实现, 见表 1, 其中  $N$  表示变换数据段长度,  $L$  表示滤波器阶数,  $J$  表示分解级数,  $k$  表示字长

表 1 本文结构和现有方案的对比结果

方案	乘法器数目	存储量	备注
Know les 的方案	$2L$	$O(2LJk)$	
AWARER 的WTP	4	$O(Nk)$	
并行滤波器	$2L$	$O(LJk)$	
脉动阵列	$L$	$O(LJk)$	
SMD 线性阵列	$2N$	$O(Nk)$	
SMD 网格结构	$2NJ$	$O(NJk)$	
本文的结构	1 或 2	$O(Nk)$	可实现小波包变换

虽然我们讨论的是正交小波变换, 但本结构对于保持数据长度不变的双正交小波变换也一样有效 对方案的适当调整和简化可以实现对无限长数据流的小波/小波包变换

### 4 结论

本文给出了一维小波/小波包变换的VLSI实现结构 由于利用了双缓冲区乒乓处理技术, 从而降低了对系统运算速度的要求, 能以较低的硬件代价来实现实时变换功能 以VHDL 语言在SYNOPSISYS 软件上综合结果表明, 当字长为 24 位时, 系统规模约为 12000 门, 另加 26kb 的存储量 以  $1\mu\text{m}$  工艺实现时,  $24 \times 24$  位乘法器的延时约为 50ns, 因而系统可用于语音、声音的实时处理, 当采用双乘法器的处理单元和更高水平工艺时, 乘法器延时可进一步降低, 系统可用于单色图象的实时处理 和现有的一些小波变换结构相比, 本结构

不仅可以实现小波变换, 而且可以实现小波包变换, 并适用于不同支集长度小波、不同长度数据、不同阶数变换, 能够处理边界效应, 这些特点是其它方案所不具备的, 因而通用性较强, 适合作为基于小波/小波包变换的各类处理系统的片上变换单元, 亦可单片实现

## 参 考 文 献

- [ 1 ] 秦前清, 杨宗凯, 实用小波分析, 西安: 西安电子科技大学出版社, 1994
- [ 2 ] I Daubechies, Commun Pure Appl Math , 1988, **41**(11): 909~ 996
- [ 3 ] S Mallat, Trans Amer Math Soc , 1989, **315**(11): 69~ 87.
- [ 4 ] D. Sinha, A. H. Tewfik, IEEE Trans Signal Processing, 1993, **41**(12): 3463~ 3479.
- [ 5 ] A. H. Tewfik, D. Sinha and P. Jorgensen, IEEE Trans Information Theory, 1992, **38**: 747~ 765
- [ 6 ] P. E. Kudumakis, M. B. Sandler, On the Performance of Wavelets for Low Bit Rate Coding of Audio Signals, Proceedings of ICASSP, 1995: 3087~ 3090
- [ 7 ] G. Knowles, Electronics Lett , 1990, **26**(15): 184~ 185
- [ 8 ] A. S. Lewis, G. Knowles, Electronics Lett , 1991, **27**(2): 171~ 173
- [ 9 ] K. K. Parhi, T. Nishitani, IEEE Trans VLSI Systems, 1993, **1**(2): 191~ 202
- [ 10 ] C. Chakrabarti, M. Vishwanath, IEEE Trans Signal Processing, 1995, **43**(3): 759~ 771.
- [ 11 ] J. Fridman, E. Manolakos, On the Synthesis of Regular VLSI Architecture for the 1-D Discrete Wavelet Transform, Proc of SPIE Conf on Mathematical Imaging: Wavelet Applications in Signal and Image Processing II, San Diego CA, 1994
- [ 12 ] A. Grzeszczak, M. K. Mandal, S. Panchanathan and T. Yeap, IEEE Trans VLSI Systems, 1996, **4**(4): 421~ 433
- [ 13 ] 吴晓冬, 李永明, 陈弘毅, 通信学报, 1997, **18**(9): 86~ 91.

## VLSI Architecture for 1-D Discrete Wavelet/Wavelet Packet Transform

Wu Xiaodong, Li Yongming, Chen Hongyi

(Institute of Microelectronics, Tsinghua University, Beijing 100084)

Received 26 November 1997, revised manuscript received 12 February 1998

**Abstract** Being a powerful signal processing tool, the wavelet/wavelet packet transform is being used in more and more research areas. Therefore its hardware implementation is being paid more attention to. In this paper, based on its use in speech coding, a VLSI architecture for 1-D discrete wavelet/wavelet packet transform is presented. Unlike the existed solutions, this architecture can be used for wavelets with different support length, for data segments with different length, and for different decomposition level. It has the features of flexibility and programmability, and is suitable to be implemented as a on-chip transform unit for many systems, and can also be implemented as a single chip.