A Review of VLSI Architectures for Discrete Wavelet Transform

Binay Chandra
PG Student
Department of Electronics & Telecommunication Engineering
D. Y. Patil College of Engineering Akurdi

Dr. Manish Sharma
Assistant Professor
Department of Electronics & Telecommunication Engineering
D. Y. Patil College of Engineering Akurdi

Abstract

Wavelet transform has attracted attention among scientists for signal and image analysis and with its Multiresolution analysis and compression capabilities. VLSI implementation of such system provides single chip platform which serves as a solution for domains like biomedical imaging, numerical analysis and development. This paper surveys different VLSI architectures for implementation of one and two dimensional wavelet transforms. Lifting and convolution based approaches makes the hardware implementation suitable for wavelet transform in terms of area and speed.

Keywords: Very Large Scale Integration, Wavelet Transform, Pyramid Algorithm, Lifting Algorithms, Folding Architectures

I. INTRODUCTION

The Discrete wavelet transform provide various capabilities over other transforms like Fourier transform. Wavelet transform provides the suitable platform for signal and image processing which has the features like Multiresolution analysis and time and frequency localization. Wavelet transform based image compression technique like JPEG2000 shows enhanced image quality and better resolution over legacy systems like JPEG. Based on different wavelets a number of signal and image processing application can be developed.

The implementation of discrete wavelet transform can be done using convolution and FIR filter bank structure. These implementations require large number of computation which are not suitable when hardware SOC and high speed image and video processing applications. Recently lifting based algorithms has been developed which provides better hardware implementation feature. The lifting based algorithms break the signal or image into different high and low frequency component filters and convert it into matrix multiplication filter implementation.

A. DWT and Lifting

The convolution based approach divides the input signal into low and high pass signals, and then the signal is subsampled and decimated by two. The produced outputs would be low and high pass components of the signal form an analysis filter bank (h, g) and the same signal can be reconstructed back by synthesis filter bank (h, g) starting from Y_h and Y_l.

\[ Y_l(n) = \sum_{i=0}^{T_l-1} h(i)x(2n - i) \]  
\[ Y_H(n) = \sum_{i=0}^{T_H-1} g(i)x(2n - i) \]

When inverse transform is computed both the high and low pass components are up sampled to produce the original input.

Wavelet transform has the feature of Multiresolution analysis where low level component i.e. y_l is further decomposed and down sampled to produce another set of high and low pass components. Same can be applied to an image where the filters are applied row wise then column wise to get 2D Discrete wavelet transform which produces four components LL, LH, HL, HH.

Advantages of lifting based DWT over Convolution based
- Computation complexity is less in case of Lifting based DWT.
Hardware implementation is simple in lifting based technique due to in-place computation, as memory buffers are not required to store the temporary data.

- There is no need to store boundary extension data and reconstruction is easy in case of Lifting based approach.
- Lossless image compression is simple in case of lifting based scheme.

## II. Architectures for 1D DWT

Implementation of 1-D Discrete wavelet transform can be done using Pyramid algorithm proposed by Mallet [1,2]. As shown in below diagram, the pyramid algorithm decomposes the input image into low and high pass components and down sample the data, low pass component is sampled again for further decomposition levels.

![1D DWT filter bank structure for 3 octaves](image)

The implementation of Pyramid algorithm can be done on single processor or multiple processor environments, where each processor computes Low and high pass components. But the drawback of the implementation is that each processor sits idle after computing the part of algorithm and full utilization of hardware is not achieved. The filters can also be implemented using Brent and Kung systolic arrays [3]. The computational load is not uniformly distributed in this case as well.

In order to increase processor utilization, folding operation needs to be performed and computations are done in pair of high/low pass filter. The following subsections explains serial and parallel architectures of performing DWT where in serial the input data is supplied in a serial manner in order to produce the output. And in parallel the input data is passed all at once in order to get the output.

### A. Folded Architecture

Figure 3 shows the folded architecture proposed by Lian, et al. [17] for performing DWT where a single filter pair is reused to compute the octaves of next level. Advantages of this process include efficient hardware implementation and low cost. Though it has some disadvantages (1) The data for each level has to be stored in a RAM or any other memory (2) Also due to serial operation, the latency is increased during computation.

![Folded architecture for 1D DWT](image)

### 1) Serial Architecture

Different architectures have been proposed for reducing storage and latency which compute wavelet transform in an interlaced manner. Interlacing is the process of scheduling the output instance at earliest when the computation of the data is required. One such algorithm is Recursive pyramid algorithm (RPA) which generates its outputs in interlaced manner [4, 5]. RPA works on an
assumption that the computation of multiply and accumulate function can be executed in a single cycle. The implementation using RPA technique is shown in below figure, where routing network is used to pass the data to different MAC units which forms the low and high pass filtering regions and then low pass filtering data is again processed through routing network to produce higher level decompositions.

2) Parallel Architecture

The implementation of parallel architecture is presented in this section. The computational unit of this filter can be implemented with K multipliers and (K-1) adders to add the products. The block diagram of parallel architecture is shown in figure.

![Parallel filter implementation of 1D DWT](image)

The computational delay in case of parallel filter architecture is $T_m + T_a$, where $T_m$ is the time taken to perform multiply operation and $T_a$ is the time taken to compute add operation. If it happens in single cycle then RPA algorithm can be applied otherwise modified RPA (MRPA) algorithm is used for computation. MRPA is only the shifted version of RPA where the shift is constant in order to not have conflicts in computation of wavelet levels.

B. Direct Mapped Architecture

The direct mapped architecture [11] for computing wavelet transform is shown in figure 6, where it uses 8 adders, 4 multipliers, 6 delay elements, 8 pipeline register (R). It has two input lines, one inputs even samples, and the other inputs odd samples,

![Direct mapped architecture](image)

The direct mapped architecture has 4 pipeline stages, Adder A1 computes the addition of a sample and one delayed sample, and it is then provided to the multiplier, after processing from the multiplier, the data is given to adder A2 block which adds the next input sample with the output of the multiplier, adder A6 computes the high-pass output samples, adder A8 shown in figure produces low-pass output.

The hardware utilization can be reduced by having only two lifting steps. (5,3) filter can be used in order to reduce the hardware utilization by 50%.

C. Programmable Multiply Accumulate Architecture

The implementation of data dependencies is done using four MACs and nine registers proposed by Chang [6]. The algorithm is executed in two phases where the input is provided to one of the registers and data flow happens with the computations in the blocks. The computation is done in the following manner,

R0 ← $x_{2i-1}$; R2 ← $x_{2i}$;
R3 ← R0 + a(R1 + R2);
R4 ← R1 + b(R5 + R3);
R8 ← R5 + c(R6 + R4);

Output$_{LP}$ ← R6 + d(R7 + R8); Output$_{HP}$ ← R8

Similarly, the computation and register allocation in phase 2 are done in the following order

R0 ← $x_{2i+1}$; R1 ← $x_{2i+2}$;
R5 ← R0 + a(R2 + R1);
R6 ← R2 + b(R3 + R5);
R7 ← R3 + c(R4 + R6);

Output$_{LP}$ ← R4 + d(R8 + R7); Output$_{HP}$ ← R7
The operations performed with this takes two input arguments and two output arguments as High and Low pass.

![Programmable Multiply accumulate architecture](image)

**Fig. 7:** Programmable Multiply accumulate architecture

### D. Dual Scan Architecture

In [7], Liao et al. presented dual scan architecture for evaluating 1D DWT that achieves full hardware utilization. By using interleaving process, two streams of input data are processed using shared functional resources. The block diagram of Dual Scan Architecture is shown in figure 8.

![The dual scan architecture](image)

**Fig. 8:** The dual scan architecture

The Dual scan architecture consists of the processing element and a memory block with two switches for input and output. The proposed architecture basically implements the convolution based lifting scheme. The input switches connects to Processing element when first stage of lifting is computed, and it’s connected to Memory element when other stages are being computed. The switch SW0 is used to separate the low frequency coefficients of input signals. When the final stage of computation is completed, then SW1 is connected to the output terminal. The calculation of DWT is done in processing element and low frequency components are stored in memory and next cycle of processing happens with the data present in memory. Dual scan architecture is capable of processing data in every cycle, so hardware utilization is 100% in this case.

### III. Architectures for 2D DWT

2D DWT can be obtained by performing 1D DWT in horizontal then followed by vertical direction. The resulting decomposition produces 4 subbands. The decomposition process is shown in figure 9.

![The 2-D DWT filter bank structure for J = 2 stages.](image)

**Fig. 9:** The 2-D DWT filter bank structure for $J = 2$ stages.

In this implementation, the input data is filtered along rows to produce high and low pass components and the image is divided into $N*N/2$ regions for high and low pass components. Both the regions are further filtered in vertical direction to produce the subbands LL, LH, HL and HH each having the area of $N/2*N/2$. The LL subband is further decomposed to next level because it contains maximum information. Different architectures for the implementation of 2D DWT has been provided below,
A. Folded Architecture

The folded architecture for the implementation of 2D DWT has been shown in figure, where a single low/high pass filter pair is used iteratively in order to obtain the decomposition levels.

In direct implementation a memory like RAM is required of size $N^2$ which increases the hardware complexity. An address generator is also required in order to enable accessing of data for row and column processing. Folded architecture has major drawbacks like hardware complexity and latency. Different architectures have to be proposed in order to overcome these drawbacks, 1) Serial-Parallel Architecture

Serial parallel filter consist of serial and parallel filters arranged in a manner that row computation is done by 2 serial filters and column computation is done by two parallel filters. Each serial and parallel filter composed of high pass and low pass filters. The block diagram of serial-Parallel system is shown in the figure 3.3 below.

Input data is fed to two serial filters at the same time. The filter $S_1$ computes the major low and high pass components and the serial filter $S_2$ computes the high and low pass components of all the stages (H,L,LL,HHL etc.). The outputs of serial filters are stored in a storage unit 1. And it is applied to parallel filters $P_1$ and $P_2$. These parallel filters also consists of low and high pass filters, and reads data column wise in order to output the components. The output of blocks $P_1$ are stored in the Storage unit 2, where it is further processed if more decomposition level is required. Storage units can be implemented using shift registers and RAMs.

2) Parallel Filters

Parallel filters are the slight modification to the serial parallel filter architecture where instead of serial filters parallel filters are used to compute the components, also input data is fed only to the parallel filter P1, which compute high and low pass component. The hardware complexity is less in case of Parallel filters as compared to Serial filters. The parallel filter P1 computes H and L component and parallel filter P2 computes other components like LL, LLLH, HH, HHHL. P1 and P2 computes the components along row wise and stores the data P3 and P4 computes the components along column wise and stores the data in the storage unit P2 which is again fed to parallel filter for further decomposition levels. The storage unit can be implemented using shift registers and RAMs.

Other versions of parallel architectures include the number of parallel filters to be used with storage unit size is increased accordingly.

B. (4,2) Filter Architecture

Ferratti and Rizzo [8] presented a dedicated architecture for evaluating 2D DWT using (4,2) filter. The block diagram of the (4,2) filter shown in figure 20, it has two parallel filters which predict and update the values along Rows and two parallel filters which predict and update the values along columns. Buffers are added in order to store the intermediate data generated by parallel filters. Dual ported buffers are used in four data words are accessed at a time. The filter parameters taken for Pred-row is $L_g=4$. $P_{red}$ row is used to compute the ‘H’ values. Upd-row uses filter parameter $L_h=2$ and it also requires ‘H’ value to evaluate ‘L’ value. Pred-col filters do the same kind of operation. If only the first level is in consideration, it utilizes only 50% architecture. When computing higher level, RPA algorithm takes care of the processing and utilization could increase in this case.

C. 2D Recursive Architecture

The recursive based architecture [9] is based on 1D architecture where the computation of hardware utilization is increased by the process of interleaving. The block diagram of 2D recursive architecture is shown in figure 14. It consists of row and column processor for operating the row and column data. Image data is inputted to the row processor and it reads data in the raster scan format. When even rows data are processed, the odd row data are shifted to first in first out (FIFO) registers, two FIFO registers are used to store high and low frequency components separately.

When the processing of row data happens, the current line DWT coefficients and the previous line components are taken to column processors. The Column processor calculates the DWT column wise in zig zag fashion. The computation of DWT is done in the interleaved manner. Data arrangement switches are incorporated in order to input the data to row and column processor.
Jung et al. [16] adopted a technique which was a mix of recursive pyramid algorithm and folded architecture for evaluating lifting based DWT. The row processor perform the row wise computation and column processor is responsible for performing filtering operation along columns in first level and along both row and columns at higher levels. Hardware utilization for row processor is almost 100% but of column processor is somewhat less.

IV. CONCLUSION

Different architectures for computing one dimensional and two dimensional wavelet transform have been presented in this paper. Lifting and folding based architectures have been presented and the performance in terms of hardware utilization and efficiency has been discussed. Parallel architectures for computing discrete wavelet transform are better in terms of timing requirements.

REFERENCES