A 3-Stage Pipeline VLSI Architecture for Fast Computation of the 2-D Discrete Wavelet Transform

Harshitha S[1], Gopalakrishna K[2]

[1]M.Tech student (SP&VLSI), Dept. of ECE., School of Engineering and Technology, Jain University
Bangalore, India

[2]Associate Professor, Dept. of ECE, School of Engineering and Technology, Jain University
Bangalore, India

Abstract: ***The discrete wavelet transform is used extensively in many fields, such as pattern recognition, image compression, speech analysis etc. because of its capability of decomposing a signal at multiple resolution levels. The 2-D Discrete Wavelet Transform is an operation through which a 2-D signal is successively decomposed in a spatial multi resolution domain by using low pass and high pass FIR filters along each of the dimensions. In this paper, we design and propose a scheme for high speed pipeline VLSI architecture for the computation of the 2-D discrete wavelet transform. Here the main focus is to develop an architecture that provides a high operating frequency and a small number of clock cycles along with efficient hardware utilization. This is done by maximizing the inter-stage and intra-stage computational parallelism for the pipeline. The inter-stage parallelism is improved by optimally mapping the computational task of multi decomposition levels to the stages of the pipeline and then synchronizing their operations. The intra-stage parallelism is improved by dividing the 2-D filtering operation into four subtasks that can be performed independently in parallel. In order validate the proposed scheme, the code is written for 8x8 input values, simulated, and implemented in FPGA for the 2-D DWT computation.***

Keywords: Discrete wavelet transform, pattern recognition, FIR filter, parallel architecture, field programmable gate array.

1. Introduction

The discrete wavelet transform has been used in many applications and in fields such as science, computer science, mathematics and engineering. The DWT decomposes a signal into components in different octaves or frequency bands by selecting appropriate scaling and shifting factors. The small scaling factor gives the fine details of the signal and the large scaling factor gives the coarse details. The shifting factor represents the time or space localization of the signal. In other transforms, such as Fourier transforms or cosine transforms where the signal is represented in frequency domain only. But DWT decomposes a signal such that it is represented more efficiently and localized in both time (space) and frequency domains. This means that in the DWT, the time (space) information of the transformed signal is not lost. This property is very useful for the analysis of signals, especially for signals with non-stationary or transitory characteristics. Since DWT involves multiple-level decomposition of a signal, the computation of the DWT can be done by repeating a process where a fully scalable window is shifted along the dimensions of the signal and making the window size shorter in each repetition. This computing process of the DWT can be done by recursively executing a set of instructions developed in software programs such as Wavelet toolbox in MATLAB, SimuWave in Simulink, or WavBox in Toolsmiths. Even though effort has been made to design software algorithms and optimized codes for the implementation of DWT, there are no general-purpose or DSP processor that can provide a performance in terms of the computation speed and resource optimization that can be achieved by a hardware implementation.

In hardware implementations, computation of the DWT is performed by using a custom hardware circuit. Hence it is possible to address the requirements of specific applications such as speed, power and size of the circuit. In literature, there exist a number of architectures for the DWT computation that stress upon such requirements of the applications. But, many applications of the DWT computation involve large volume data such as image or video. The multiple resolution level operation of the DWT adds even more to the vastness of the data to be processed, which adversely affects the requirements of speed, power and area of the architectures. Thus, it is a challenging task to design a high-speed, low-power and area efficient VLSI architecture to implement DWT computation for real-time applications.

2. Literature Survey

In the past, much architecture has been proposed for the DWT computation. All these architectures aim at giving high performance, in terms of the speed, area, throughput, latency and power consumption. We categorize our study of these existing architectures, into single-processor, parallel-processor and pipeline architectures, depending on the number of processors used by them as well as their configuration. In the single-processor category, the architecture proposed by A. S. Lewis *et al.* [1] is an example of a multiplier less architecture, but it is restricted only to certain types of wavelets such as Daubechies. Also, the architecture is not scalable which means that it is not capable of being expanded or upgraded. Another example in the single-processor category is the one that is introduced by Movva and Srinivasan [2]. This architecture uses separable approach and the 2-D DWT is obtained by performing a row-wise computation which is then followed by a column-wise computation using a single lifting-scheme based processor. The drawback of this architecture is low-speed and it requires a large memory space. Hung *et al.* [3] proposed a single-processor architecture which uses non-separable approach. Here an $L\times L$-tap filtering operation is carried out by a processor consisting of a cascade of three blocks: parallel multipliers, L accumulators along the row direction and one accumulator along the column direction. This architecture has low computational speed, since the samples of the four decomposed components are computed sequentially. The architecture of Uzun and Amira [4] is another example of a single-processor architecture, in which the processor consists of $L/2$ adders and $L/2$ parallel processing blocks, where L is the filter length, followed by an accumulator. This architecture requires large storage (delay units) to store the low pass-low pass output components of various resolution levels. In the category of parallel-processor architectures, Chakrabarti and Mumford [5] have given a four-processor architecture, in which one filter performs the horizontal filtering operation of the first resolution level, two filters perform the vertical low pass and high pass filtering operations respectively of all the resolution levels and another filter performs the horizontal filtering operations of the second and subsequent resolution levels. This architecture has a drawback of having low hardware utilization, since the amount of computations assigned to the four processors are not proportional to the amount of hardware employed by them. Wu and Chen [6] proposed six processor architecture for the 2-D DWT computation. Here two processors employ a poly phase decomposition technique for row-wise filtering operation, and four other processors employ a filter coefficient folding technique for column-wise filtering operation. But the design complexity of this architecture is high, since the parallel processors have different structures. In the pipeline-processor category Jou *et al.* [7] gave a pipelined architecture using four processors to form a pipeline. In this architecture, the first two processors are used to perform the row-wise and column-wise operations respectively of first level and remaining two processors are used to perform, the row-wise and column-wise operations respectively, of the remaining levels. The disadvantage of this architecture is that it has a large latency and requires large storage space. In the architecture given by Mihic [8], a J-level 2-D DWT is performed using a pipeline of $2J$ processors. Each processor uses a semi-systolic array of MAC cells for row-wise or column-wise filtering operation of a resolution level. The architecture is not practical for the computation of the DWT with large number of resolution levels and also it has a very large latency. Marino [9] proposed a pipeline of two stages, one for the computation of the first resolution level and the other for the computation of all the remaining levels. Here each stage employs L parallel processing blocks. Since the structures of the processing blocks are different the design complexity of this architecture is high. Also, since each processing block has large number of MAC cells it has a high hardware resource complexity. Based on this study, a scheme for the design of three stage pipeline VLSI architecture for the fast computation of the 2-D discrete wavelet transform has been proposed in this paper.

3. Computation of the 2-D discrete wavelet transform

The 2-D DWT is an operation through which a 2-D signal is successively decomposed in a spatial multi-resolution domain by using low pass and high pass FIR filters along each of the two dimensions. At a given resolution level, the four FIR filters, denoted as high pass high pass (HH), high pass low pass (HL), low pass high pass (LH) and low pass low pass (LL) filters produce the HH, HL, LH and LL sub band data respectively of the decomposed signal. This is illustrated in Figure 1.

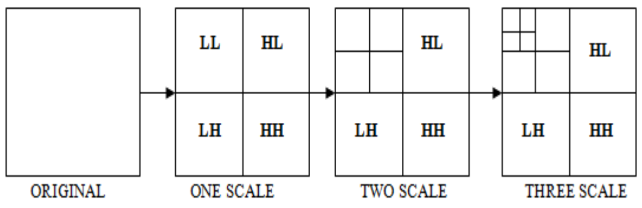


Figure 1: Illustration of 2-D DWT decomposition

At each level the samples of the four sub bands of the decomposed signal are decimated by a factor of two in each of the two dimensions. For the first level of decomposition the 2-D signal is given as input and for the succeeding levels of decomposition the decimated LL sub band signal from the previous resolution level is given as input.

3.1 Computation of the four sub bands

Let us consider a 2-D signal that is represented by an N_0 x N_0 matrix $\mathbf{S^{(0)}}$. Let its $(m,n)^{th}$ element be denoted by $S^{(0)}$ (m,n) for $(0 \le m,n \le N_0 - 1)$. Here N_0 is chosen to be 2^J where J is an integer. Let us denote the coefficients of a 2-D FIR filter P (P=HH, HL, LH, LL) by an L x M matrix $\mathbf{H^{(P)}}$. Let the $(k,i)^{th}$ coefficient of the filter P be denoted by $H^{(P)}$ (k,i) for $(0 \le k \le L-1, 0 \le i \le M-1)$. The decomposition of the 2-D signal at a given level j = 1, 2, 3, . . . , J is given by

$$A(j)\,(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H(HH)\,(k,i)\;.\;S(j-1)\,(2m-k,\,2n-i) \tag{1a}$$

$$B(j)\,(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H(HL)\,(k,i)\;.\;S(j-1)\,(2m-k,\,2n-i) \tag{1b}$$

$$C(j)\,(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H(LH)\,(k,i)\;.\;S(j-1)\,(2m-k,\,2n-i) \tag{1c}$$

$$S(j)\,(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H(LL)\,(k,i)\;.\;S(j-1)\,(2m-k,\,2n-i) \tag{1d}$$

Here A(j)(m,n), B(j)(m,n), C(j)(m,n) and S(j)(m,n) for $(0 \le m,n \le N_j - 1)$ are the $(m,n)^{th}$ elements of the four N_j x N_j matrices, where $N_j = N_0/2^j$. A(j), B(j), C(j) and S(j) are the High pass High pass (HH), High pass Low pass (HL), Low pass High pass (LH) and Low pass Low pass (LL) sub bands respectively of the 2-D input signal at the j^{th} level. From

(1a), (1b), (1c) and (1d) we see that the four decomposed sub bands at a given level are obtained by performing four 2-D convolutions. Each 2-D convolution can be viewed as a sum of the products of the L x M filter coefficients and the elements contained in an L x M window sliding on a 2-D data. The decimation by a factor of two in the horizontal and vertical dimensions can be obtained by sliding the L x M window by two positions horizontally and vertically for the computation of two successive samples. Further only the Low pass Low pass (LL) sub band data of decomposition is used as input for the decomposition at the next level. After J decompositions, the 2-D signal $\mathbf{S^{(0)}}$ is transformed into J resolution levels, with HH, HL and LH sub bands from each of the first J─1 levels and HH, HL, LH and LL sub bands from the last (J^{th}) level. Since $N_j = N_0/2^j$, the number of samples that needs to be processed at each level j is one fourth of that at the preceding level.

3.2 Four channel filtering operation

For parallel processing of the 2-D DWT computation, the L x M filtering operation should be be divided into multi-channel operations where each channel can process one part of the 2-D data. We can see from (1a), (1b), (1c), (1d) that the even and odd indexed elements always operate on the even and odd indexed filter coefficients respectively. We can therefore divide the LL sub band matrix $\mathbf{S^{(j)}}$ at the j^{th} level into four ($N_j/2 + L/2$) x ($N_j/2 + M/2$) sub matrices. These sub matrices are represented by $S_{ee}^{(j)}$, $S_{oe}^{(j)}$, $S_{eo}^{(j)}$ and $S_{oo}^{(j)}$. The $(m,n)^{th}$ elements of the sub matrices for ($0 \le m \le N_j/2 + L/2 - 1$, $0 \le n \le N_j/2 + M/2 - 1$) are given by

$$S_{ee}^{(j)}(m,n) = S^{(j)}(2m,2n) \qquad (2a)$$
$$S_{oe}^{(j)}(m,n) = S^{(j)}(2m+1,2n) \qquad (2b)$$
$$S_{eo}^{(j)}(m,n) = S^{(j)}(2m,2n+1) \qquad (2c)$$
$$S_{oo}^{(j)}(m,n) = S^{(j)}(2m+1,2n+1) \qquad (2d)$$

From (2a), (2b), (2c) and (2d) we can see that the data at any resolution level is divided into four channels for processing by first separating the even and odd indexed rows of $\mathbf{S^{(j)}}$, and then separating the even and odd indexed columns of the resulting two sub-matrices. The data in each channel is then computed by an ($L/2$ x $M/2$) tap filtering operation. For such a four channel filtering operation the filter coefficients also need to be decomposed properly. Accordingly, the matrix $\mathbf{H^{(P)}}$ should be decomposed into four ($L/2$ x $M/2$) sub-matrices. These sub matrices are represented by $H_{ee}^{(P)}$, $H_{oe}^{(P)}$, $H_{eo}^{(P)}$ and $H_{oo}^{(P)}$. The $(k,i)^{th}$ elements of the sub matrices for ($0 \le k \le L/2 - 1$, $0 \le i \le M/2 - 1$) are given by

$$H_{ee}^{(P)}(k,i) = H^{(P)}(2k,2i) \qquad (3a)$$
$$H_{oe}^{(P)}(k,i) = H^{(P)}(2k+1,2i) \qquad (3b)$$
$$H_{eo}^{(P)}(k,i) = H^{(P)}(2k,2i+1) \qquad (3c)$$
$$H_{oo}^{(P)}(k,i) = H^{(P)}(2k+1,2i+1) \qquad (3d)$$

At the j^{th} resolution level any of the four sub band signals A(j), B(j), C(j) and S(j) can be computed as a sum of four convolutions by using ($L/2$ x $M/2$) tap filters. Hence we can now express the LL sub band as

$$S^{(j)}(m,n) = \sum_{k=0}^{L/2-1}\sum_{i-0}^{M/2-1} H_{ee}^{(LL)}(k,i)\, S_{ee}^{(j-1)}(m+k,\ n+i)$$
$$+ \sum_{k=0}^{L/2-1}\sum_{i-0}^{M/2-1} H_{oe}^{(LL)}(k,i)\, S_{oe}^{(j-1)}(m+k,\ n+i)$$
$$+ \sum_{k=0}^{L/2-1}\sum_{i-0}^{M/2-1} H_{eo}^{(LL)}(k,i)\, S_{eo}^{(j-1)}(m+k,\ n+i)$$
$$+ \sum_{k=0}^{L/2-1}\sum_{i-0}^{M/2-1} H_{oo}^{(LL)}(k,i)\, S_{oo}^{(j-1)}(m+k,\ n+i) \qquad (4)$$

We can see from (4) that the filtering operations in the four channels are independent and identical, which can be exploited in the design of efficient pipeline architecture for the 2-D DWT computation.

4. Pipeline structure for the 2-D DWT computation

In the proposed three-stage architecture, the first and second level decompositions are computed by stages 1 and 2 respectively, and the decomposition of the remaining levels are computed by stage 3. The computation blocks in the three stages differ only in the number of similar processing units used by them depending on the amount of the computations assigned to the stages. This is because the basic operation of computing each output sample, regardless of the resolution level or the sub band, is the same. From (4) we can see that an (L x M) tap filtering operation is divided into four independent ($L/2$ x $M/2$) tap filtering operations, where each operates on the 2-D ($L/2$ x $M/2$) data which results from the even or odd indexed rows and even or odd indexed columns of an (L x M) window of the LL sub band data. We can now consider a unit having ($L/2$ x $M/2$) MAC cells as the basic processing unit for ($L/2$ x $M/2$) tap filtering operation. The L x M window of the raw 2-D input data or the L x M window of the LL sub band data must be divided into four $L/2$ x $M/2$ sub windows. This dividing of the data can be done by designing an appropriate data scanning unit (DSU) for each stage. The basis for the DSU is the way in which the raw input or the LL sub band data is scanned. The three stages also require memory space (buffer) to store the raw input data or the LL-sub band data before scanning. Stages 1 and 2 require a buffer of size of O(N) since they store only part of a few rows of raw input or LL sub band data at a time. Stage 3 has a buffer size of O(N^2) as they store the entire LL sub band data of one resolution level. Figure 2 shows the block diagram of the 3-stage pipeline structure. Here only the LL sub band data flow to the next stages after stage 1 which is necessary for subsequent operations and the HH, HL and LH sub band data are taken directly into an external memory.

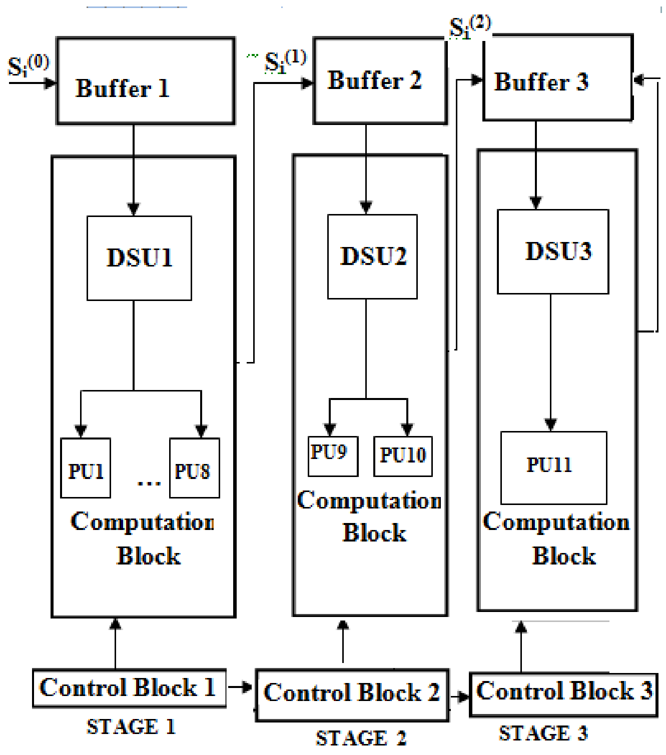


Figure 2: Block Diagram of the 3-Stage Pipeline Structure

4.1 Structure of the data scanning unit

In order to partition the *L* x *M* window into four *L/2* x *M/2* sub-windows we make use of the Data Scanning Unit. The DSU first partitions the samples of the window into two parts depending on whether a sample belongs to an even indexed or odd indexed row. Then the samples in each part are further partitioned into two parts depending on whether a sample belongs to an even indexed or odd indexed column. The first partition is obtained by directing the scanned samples alternatively into two sets of *L/2* shift registers. The second partition is obtained by reorganizing the samples stored in two sets of shift registers by using demultiplexers. Finally the samples of the four sub windows are stored into four units of *L/2* x *M/2* parallel registers respectively.

Figure 3 shows a structure of the DSU used to obtain the sub windows as explained above. The down sampling operation by two in the vertical and horizontal directions which is required by the transform is automatically incorporated by this data scanning scheme. Thus we do not need any additional peripheral circuits and registers for the down sampling operations. Hence when compared to other schemes this data scanning scheme uses less hardware resources and lesser number of registers for the stages.

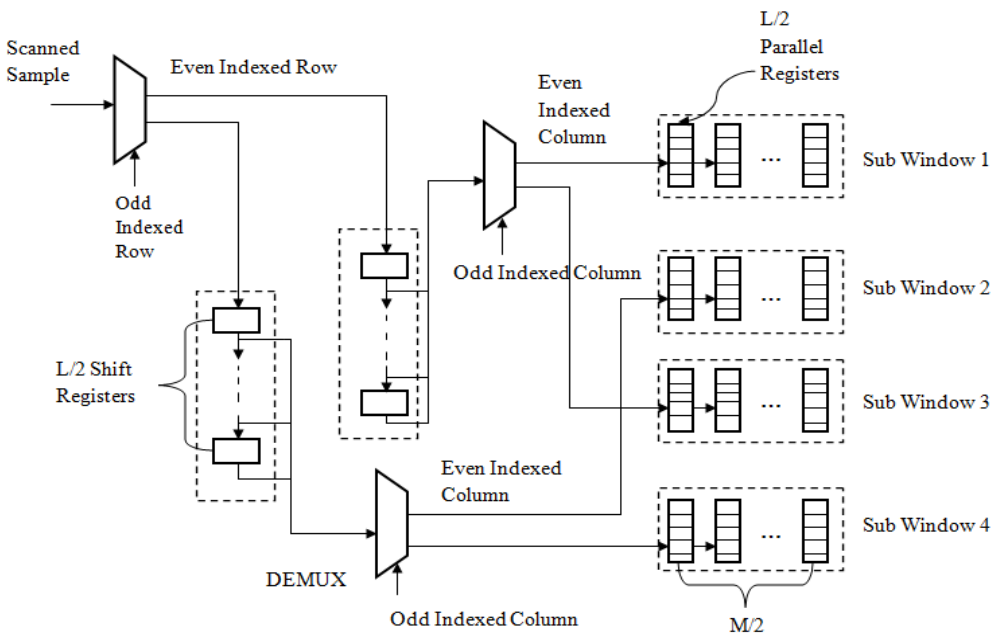


Figure 3: Structure of the Data Scanning Unit

4.2 Structure of the processing units

As explained decomposing the input data into four sub bands needs four *L* x *M* filtering operations and each of the four filtering operations needs four *L/2* x *M/2* tap filtering operations. Thus we need a total of sixteen *L/2* x *M/2* tap filtering operations for the computation of the samples for the four sub bands. Hence for each stage these sixteen types of filtering operations must be assigned appropriately to the processing units available to the stage. In stage 1 we have eight processing units and the processing task is distributed among them so that one processing unit carries out the subtask of *L/2* x *M/2* tap filtering operations using the data of one sub window. Figure 4 shows the structure of eight processing units used by stage 1. Each of the processing units PU1 to PU4 does the LL and LH filtering operations sequentially using the sub windows 1 to 4 respectively. Similarly each of the processing units PU5 to PU8 does the HH and HL filtering operations using the same sub windows. Here the LL and HH sub band samples are produced in parallel in one clock cycle and the LH and HL sub band samples are produced in parallel in the next clock cycle.

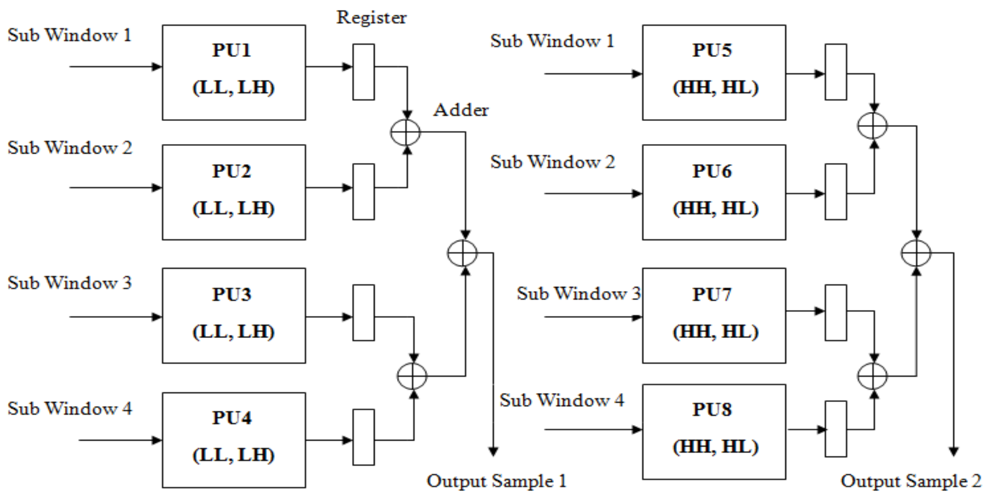


Figure 4: Structure of eight processing units used by stage 1

Stage 2 uses two processing units and the data of the four sub windows 1 to 4 are assigned in a sequential manner to the two processing units. The sub windows 1 and 3 are sequentially given to PU9 and the sub windows 2 and 4 are given to PU10. Figure 5 shows the structure of two processing units used by stage 2. Each of the processing units PU9 and PU10 does the *L/2* x *M/2* tap filtering operations. Here PU9 and PU10 operate in parallel in order to produce the LL, LH, HH and HL sub band samples sequentially in eight consecutive clock cycles.

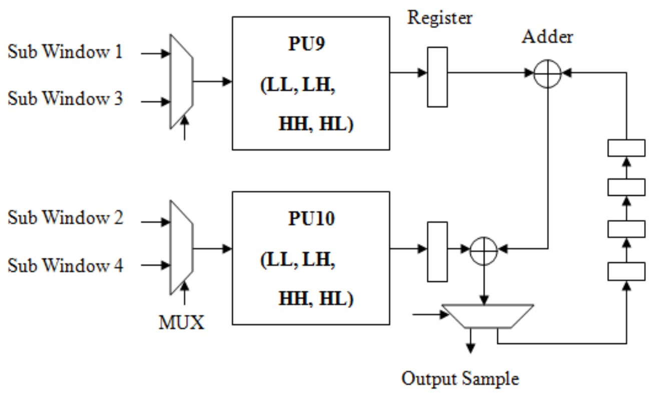


Figure 5: Structure of two processing units used by stage 2

Stage 3 uses only one processing unit, PU11. It carries out all the filtering operations for each of the four sub windows. The four sub windows 1 to 4 are given successively as input to PU11. Figure 6 shows the structure of a processing unit used by stage 3. For each sub window the processing unit PU11 does the *L/2* x *M/2* tap filtering operations. Here PU11 produces the LL, LH, HH and HL sub band samples sequentially in sixteen consecutive clock cycles.

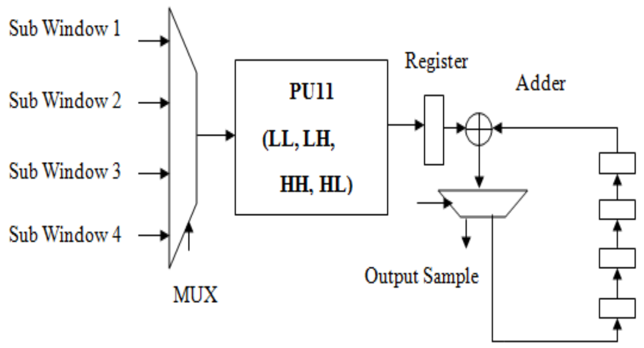


Figure 6: Structure of a processing unit used by stage 3

4.3 Block diagram of the processing unit

We can view the filtering operation carried out by a processing unit as *L/2* x *M/2* parallel multiplications followed by an accumulation of the *L/2* x *M/2* products. Suppose the input samples have a word length of X bits and the filter coefficients have a word length of *Y* bits, then the processing unit will produce an array of (Y * L * *M/4)* x X bits simultaneously in one clock cycle. So to obtain the output sample pertaining to a given sub window the bits of the partial products must be accumulated vertically downward and from right to left by taking into consideration the propagation of the carry bits. Now the partial product bits and/or the carry bits from different rows have to be added. This is done by using bit wise adders in parallel. Then finally we need to add the two rows of bits produced by the accumulation network in order to produce one row of output bits by using a carry propagation adder. Figure 7 shows a block diagram of a processing unit as discussed above.

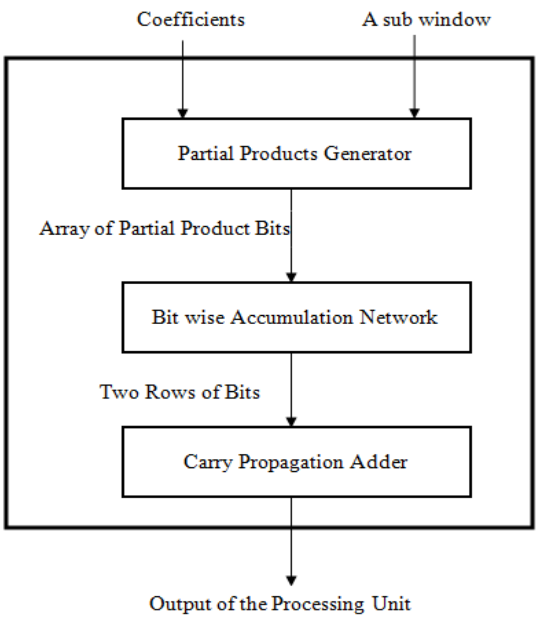

Coefficients
A sub window
Partial Products Generator
Array of Partial Product Bits
Bit wise Accumulation Network
Two Rows of Bits
Carry Propagation Adder
Output of the Processing Unit

Figure 7: Block diagram of a processing unit

5. Results and Discussion

In order evaluate the performance of the proposed architecture, the code is written for 8x8 input values, simulated, and implemented in FPGA for the 2-D DWT computation. The code is simulated by using Xilinx 14.2 version supported with Isim simulator for three levels of decomposition. The simulation results for one, two and three levels of decomposition are shown in Figure 8, Figure 9 and Figure 10 respectively.

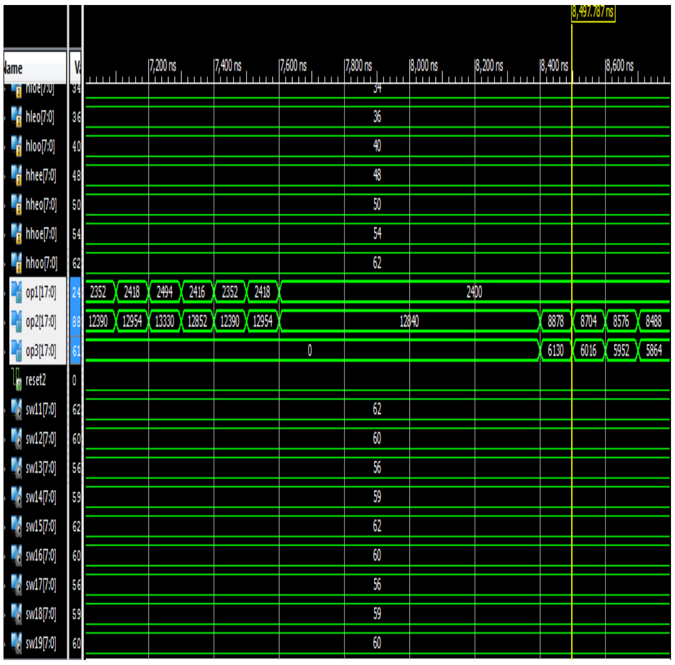


Figure 8: Simulation results for one level of decomposition

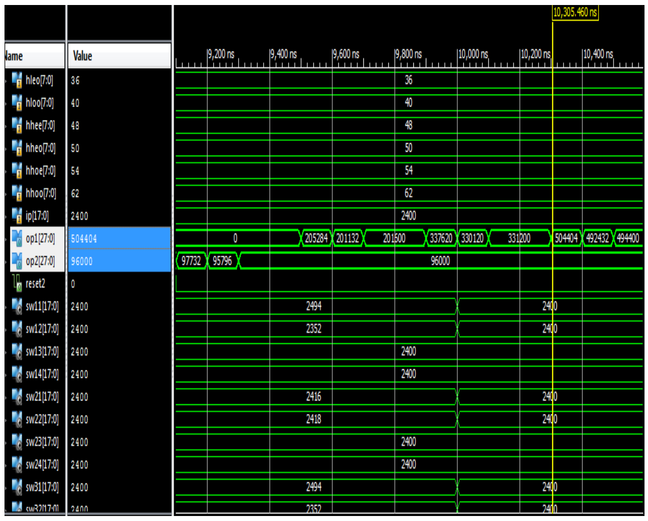


Figure 9: Simulation results for two levels of decomposition

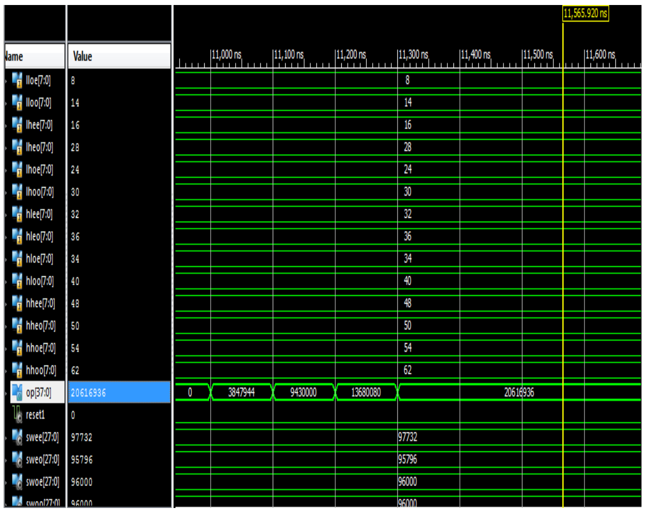
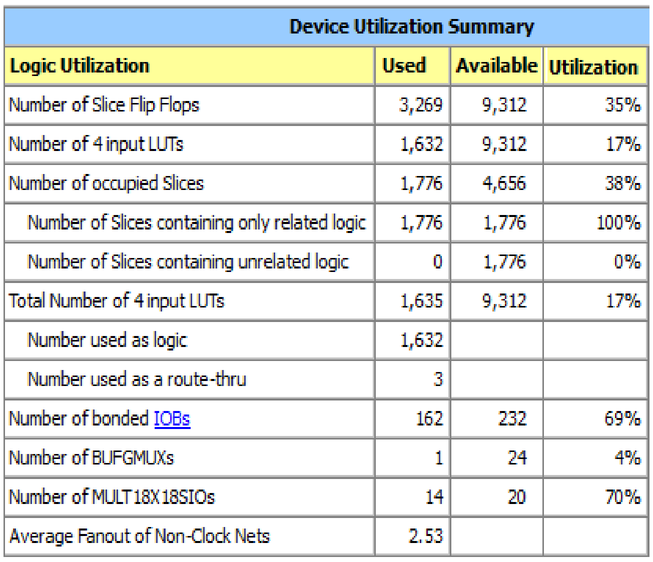


Figure 10: Simulation results for three levels of decomposition

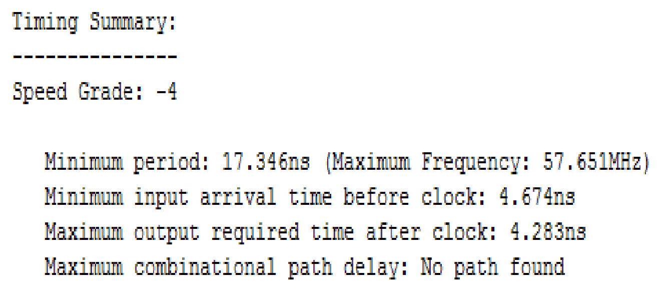
The hardware resources used by the FPGA board Spartan3E XC3S500E-4 in terms of the numbers of configuration logic block (CLB) slices, flip-flop slices, 4-input look-up tables (LUTs) and input/output blocks (IOBs) is shown in Figure 11.



Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	3,269	9,312	35%
Number of 4 input LUTs	1,632	9,312	17%
Number of occupied Slices	1,776	4,656	38%
Number of Slices containing only related logic	1,776	1,776	100%
Number of Slices containing unrelated logic	0	1,776	0%
Total Number of 4 input LUTs	1,635	9,312	17%
Number used as logic	1,632		
Number used as a route-thru	3		
Number of bonded IOBs	162	232	69%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	14	20	70%
Average Fanout of Non-Clock Nets	2.53		

Figure 11: Device utilization summary for 2-D DWT computation

Also the proposed pipeline architecture is found to perform well with a clock period as short as 17.346 ns (i.e. a maximum clock frequency of 57.651 MHz). When compared to parallel processing architecture the proposed scheme computes the 2-D DWT of a 2-D signal at higher speed.



```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 17.346ns (Maximum Frequency: 57.651MHz)
   Minimum input arrival time before clock: 4.674ns
   Maximum output required time after clock: 4.283ns
   Maximum combinational path delay: No path found
```

Figure 12: Timing Summary for 2-D DWT computation

Table 1: Comparison of proposed scheme with parallel architecture

Type of architecture	*T_c in nanoseconds*
Parallel architecture	17.8
Proposed (Pipeline architecture)	17.34

6. Conclusion

In this paper we have proposed a three-stage pipeline architecture for the real-time computation of the 2-D DWT. The main objective is to achieve a short computation time by maximizing the clock frequency and hence minimizing the number of clock cycles required for the DWT computation. This is done by developing a scheme for enhanced inter-stage and intra-stage computational parallelism for the pipeline architecture. The inter-stage parallelism is improved by optimally mapping the computational task of multi decomposition levels to the stages of the pipeline and then synchronizing their operations. The intra-stage parallelism is improved by dividing the 2-D filtering operation into four subtasks that can be performed independently in parallel. The simulation results for the 2-D DWT computation show that the proposed scheme can operate with a minimum clock period of 17.346 nanoseconds, that is at a maximum frequency of 57.651 MHz. The proposed scheme computes the 2-D DWT at a higher speed when compared to parallel architecture.

References

[1] A.S. Lewis and G. Knowles, "VLSI architecture for 2D Daubechies wavelet transform without multipliers," Electron. Lett., vol. 27, no. 2, pp.171−173, Jan. 1991.

[2] S. Movva, S. Srinivasan, "A novel architecture for lifting-based discrete wavelet transform for JPEG2000 standard suitable for VLSI implementation," in Proc. IEEE 16th Int. Conf. VLSI Design, 4−8 Jan. 2003, pp. 202- 207.

[3] K.C. Hung, Y.S. Hung, and Y.J. Huang, "A nonseparable VLSI architecture for the two-dimensional discrete periodized wavelet transform," IEEE Trans. VLSI Systems, vol. 9, no. 5, pp. 565−576, Oct. 2001.

[4] S. Uzun and A. Amira, "Design and FPGA implementation of non-separable 2- D biorthogonal wavelet transforms for image/video coding," in Proc. IEEE Int. Conf. Image Processing (ICIP), 24−27 Oct. 2004, vol.4, pp. 2825−2828.

[5] C. Chakrabarti and C. Mumford, "Efficient realizations of analysis and synthesis filters based on the 2-D discrete wavelet transform," in Proc. IEEE Int. Conf. Audio, Speech, and Signal Processing, May 1996, pp. 3256–3259.

[6] P. Wu and L. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 4, pp. 536 545, Apr. 2001.

[7] J. M. Jou, P. Y. Chen, Y. H. Shiau, and M. S. Liang, "A scalable pipelined architecture for separable 2 D discrete wavelet transform," in Proc. the Asia and South Pacific Design Automation Conf. (ASP-DAC), 18−21 Jan. 1999, vol. 1, pp. 205−208.

[8] K. Mihic, "An efficient semi-systolic architecture for 2-D discrete wavelet transform," in Proc. IEEE European Conf. Circuit Theory and Design (ECCTD), Espoo, Finland, 28−31 Aug. 2001, pp. 333−336.

[9] F. Marino, "Efficient high-speed/low-power pipelined architecture for the direct 2-D discrete wavelet transform," IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 12, pp. 1476−1491, Dec. 2000.

Author Profile

Mr. Gopalakrishna K received his B.E degree in Electronics from Dr. Ambedkar Institute of Technology, Bangalore University, Bangalore in the year 1993 and M.Tech degree in VLSI Design and Embedded Systems from VTU. He is awarded with M. Phil degree in Computer Science from M. S. University, Tirunelveli, Tamil Nadu, in the year 2003. He has 21 years of teaching experience in various colleges in India. He is currently working as Associate Professor, Dept. of Electronics & Communication Engineering, School of Engineering & Technology, Jain University. He is pursuing PhD. from Jain University. His field of interests includes Advance microprocessors, Microcontrollers, Embedded Systems, VLSI Technology and VLSI Designing.

Ms. Harshitha S is a student in the Department of Electronics and Communication Engineering, School of Engineering and Technology, Jain University, Karnataka, India. She received her Bachelor degree in Electronics & Communication Engineering from VTU in 2011. She is pursuing M. Tech (SP and VLSI) in Electronics and Communication Engineering, Jain University, Karnataka, India. Her research interest includes Low power VLSI Design, Analog and Mixed mode VLSI Design, CMOS VLSI Design, Embedded Systems Design, DSP and Adaptive signal processing.