INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

A Novel Poly-Phase Architecture For High Performance Discrete Wavelet Transform In FPGA

Rakhee Sanjay Bhojakar
M.Tech Student, W.C.E., Sangli, Maharashtra, India
Patil B.G
Associate Professor, W.C.E., Sangli, Maharashtra, India
Tanaji Umaji Kamble
M.Tech Student, W.C.E., Sangli, Maharashtra, India

Abstract:
Wavelet Transforms are used in number of application. They are applied in different fields such as signal processing, speech and image compression, biometrics, and so on. Design of discrete wavelet transforms is complex due to large number of arithmetic operations involved. In this Paper, pipeline VLSI architecture for the computation of the 1-D discrete wavelet transform (DWT) using poly-phase filter bank structure is proposed. The main focus of the scheme is on reducing the number and period of clock cycles and reducing the complexity of hardware resources needed for the computation. For hardware implementation, the choice of filter bank structure determines the efficiency and accuracy of computation of the DWT. here poly-phase filter bank structure is proposed for DWT computation which increases performance of computation as in poly-phase implementation units can operate in parallel, and therefore the filtering operation have less delay and however, pipelining can be used in this scheme to reduce the delay.

Key words*: Discrete wavelet transform, poly-phase filter bank structure, VLSI architecture, computation of DWT, poly-phase implementation units, parallel, pipeline*

1.Introduction

The wavelet transform becomes an emerging signal processing technique and it is used to decompose and reconstruct non-stationary signals efficiently.

Discrete wavelet transform (DWT) is based on time-scale rather than time-frequency representation. Wavelets are represented by scaling or dilation equations as opposed to difference equations. The discrete wavelet transform provides time-scale representations of discrete-time signals. This multigrid representation is computed using dilation equations rather than traditional difference equations. The DWT decomposes the approximation of the signal at each level of resolution into approximation and detail of the signal at the next level of resolution. Each level can add more detail to the information content.

The discrete wavelet transform uses translated and scaled mother wavelets as a set of basis functions to represent a signal. The translation factor shifts the original signal in the time domain and the scale factor determines the frequency band. As a result, the discrete wavelet transform gives time-frequency joint representations of the original signal. The discrete wavelet transform is a mathematical function that involves multiplication and addition operation to convolve signal of interest with predefined wavelet expansion coefficients.

The DWT consists of banks of low pass filters, high pass filters and down sampling units. Half of the filter convolution results are discarded because of the down sampling at each DWT decomposition stage. Only the approximation part of the DWT wavelet results is kept so that the number of samples is reduced by half. The level of decomposition is limited by the distortion tolerable from the resulting signal.

Several VLSI designs of the discrete wavelet transform have been reported for the discrete wavelet transform implementations. The DSP device is usually a general-purpose, fixed-architecture technology. The DSP processor is not capable of configuring the data path width and it may waste hardware resources. Besides, it needs long development cycle, and it may consume more power. As a low cost fast prototyping tool, FPGA design offers flexibility of re-programmability. The FPGA device allows easy configuration of the design data path, easy modification of the design architecture and also provides high performance at the same time.

The Discrete wavelet transform (DWT) has been developed as an efficient DSP tool for signal analysis and synthesis, speech compression, image compression, and even video compression. Due to its growing number of applications in various areas, it is necessary to explore the hardware implementation options of the Discrete Wavelet Transform (DWT).

The DWT is computationally intensive because of multiple levels of decomposition involved in the computation of the DWT. Design of discrete wavelet transforms is complex due to large number of arithmetic operations involved. It is therefore a challenging problem to design an efficient VLSI architecture to implement the DWT computation for real-time applications, particularly those requiring processing of high frequency or broadband signals.

The Discrete wavelet transform (DWT) has gained the reputation of being a very effective signal analysis tool for many practical applications. This dissertation presents an approach towards VLSI implementation of the Discrete Wavelet Transform. In Discrete Wavelet Transform, the filter implementation plays the key role. The poly phase structure is proposed for the filter implementation, which uses multiply accumulate technique.

The implementation architecture is characterized by several parameters such as sample speed, system latency, and hardware area and power consumption. This dissertation describes several algorithm-architecture trade-offs associated with the implementation of the DWT.

The main focus of the scheme is on reducing the number and period of clock cycles for the DWT computation with little or no overhead on the hardware resources.

2.Poly-Phase Structure

There are various architectures for implementing a two channel filter bank for DWT. A filter bank basically consists of a low pass filter, a high pass filter, decimators or expanders and delay elements. Commonly used structures for hardware implementations of filter bank architectures are: direct form, polyphase and lattice, lifting. In this dissertation poly-phase structure for filter bank implementation is used.

For hardware implementation, the choice of filter bank structure determines the efficiency and accuracy of computation of the DWT. All structures have some advantages and drawbacks which have to be carefully considered and based on the application, the most suitable implementation can be selected.

It is observed that the direct form is a very inefficient method for DWT implementation. This method is almost never used for DWT computation. The polyphase structure appears to be an efficient method for DWT computation. But the lattice and lifting implementations require fewer computations than the polyphase implementation and therefore are more efficient in terms of number of computations. However, the polyphase implementation can be made more efficient than the lattice and lifting schemes in case of long filters by incorporating techniques like MAC Arithmetic.

In the case of the lattice and lifting schemes, the filtering units cannot operate in parallel as each filtering unit depends on results from the previous filtering unit. In the case of convolutional polyphase implementation, the units can operate in parallel, and therefore the filtering operations have less delay. However, pipelining can be used in the other schemes to reduce the delay. The nonpolyphase structure is inefficient: in analysis, downsampling discards half the samples just computed by the filters; in synthesis, upsampling before filtering means that half of the filter multiplication operations are multiplying by zero.

In short, half of all mathematical computations are wasted. In contrast, a polyphase structure does not perform the wasted operations: downsampling precedes filtering and upsampling follows filtering. The polyphase filter structure is computationally equivalent to the original structure, but much more efficient. The result is a theoretical doubling in throughput compared to the nonpolyphase structure.

$$\begin{bmatrix} G_{0even} G_{0odd} \\ H_{0even} H_{0odd} \end{bmatrix} \times \begin{bmatrix} X_{even} \\ Z^{-1} X_{odd} \end{bmatrix} = \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix}$$

The input signal is split into odd and even samples, similarly, the filter coefficients are also split into even and odd components so that convolves with of the filter &

Convolve with of the filter. The two phases are added together in the end to produce the low pass output. Similar to the high pass filter where the high pass filter is split into even and odd phases H0even and H0odd.

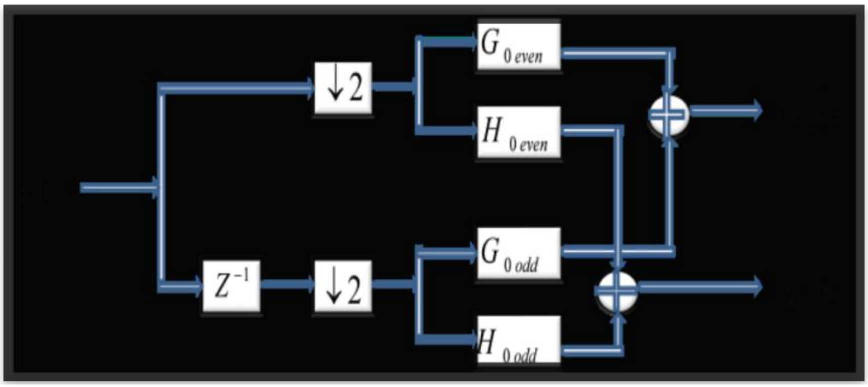


Figure 1: Block Diagram Of The Poly-Phase Structure

3.Choice Of Pipeline Structure For Dwt Computation

In a pipeline structure for the DWT computation, multiple stages are used to carry out the computations of the various decomposition levels of the transform. Thus, the computation corresponding to each decomposition level needs to be mapped to a stage or stages of the pipeline. In order to maximize the hardware utilization of a pipeline, the hardware resource of a stage should be proportional to the amount of the computation assigned to the stage.

In this two-stage structure, stage 2 performs by operating on the data produced by stage 1 and on those produced by it, and therefore, the operations of the two stages need to be synchronized in a best possible manner.

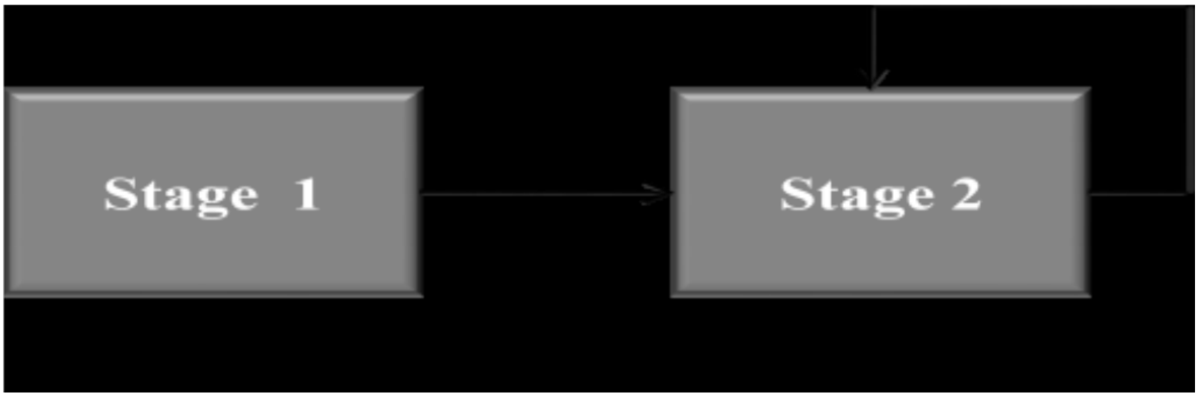
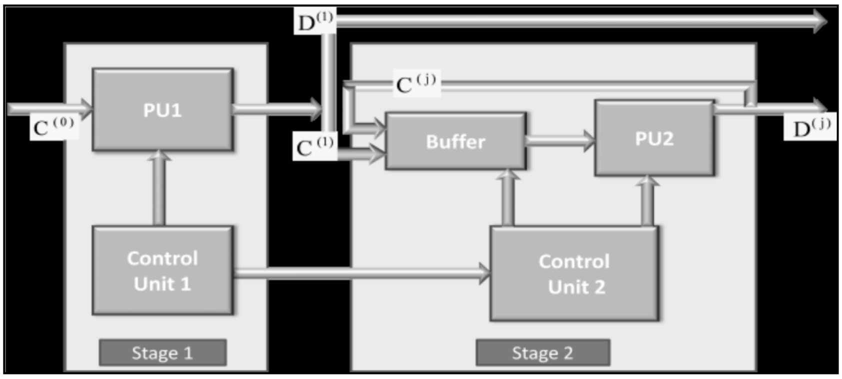


Figure 2: Pipeline Structure With Two-Stage[21]

4. Architecture Design

In the stage-equalized architectures, the two stages together perform the DWT computation, with the amount and type of computations of the individual stages being the same, each of the two stages can use identical processing units. However, the control units to be employed by the stages have to be different.



.

Figure 3: Block Diagram Of The Two-Stage Architecture [21]

5.Processing Unit For Filtering Computation

In each stage, the processing unit by employing an - MAC cell network performs an -tap filtering operation and, at each clock cycle, generates a total

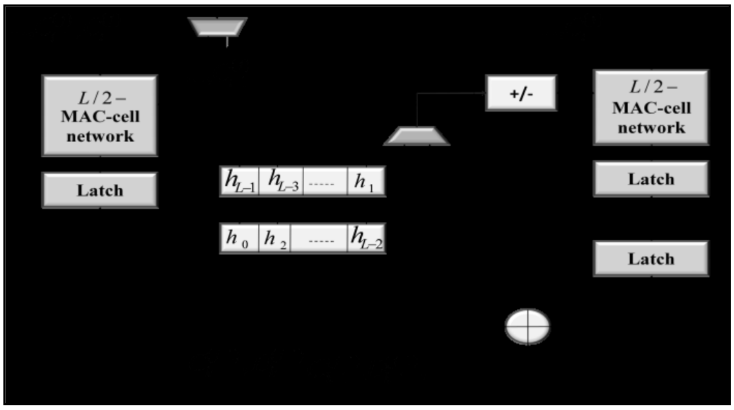


Figure 4: Block Diagram Of The Processing Unit For Filtering Computation[21]

6. Buffer Unit

All the output data must be synchronized. This synchronization process is facilitated by introducing in stage 2 a buffer, which stores output data from the two stages and provides input data to stage 2**.**

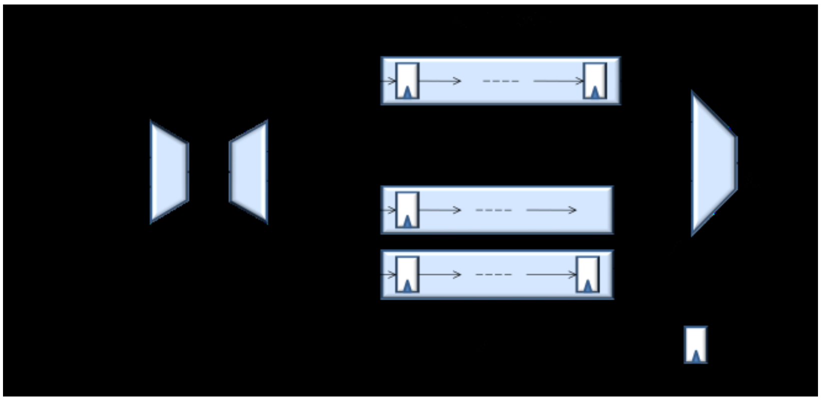


Figure 5: Structure Of The Buffer [21]

7.Mac (Multiply–Accumulate) Unit

It is crucial to aim at achieving the shortest critical data path when designing an -MAC-cell network for this architecture.

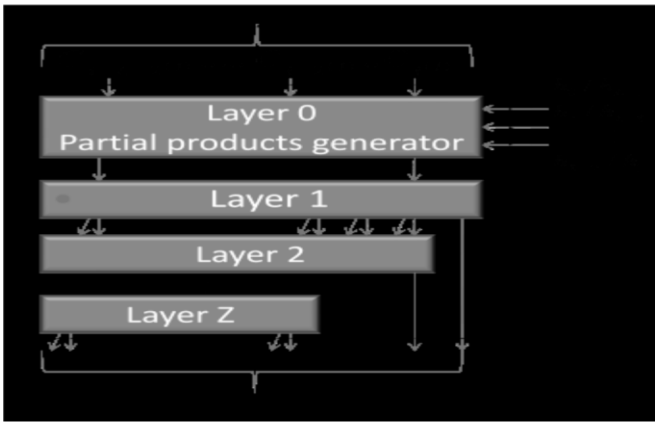


Figure 6: Structure Of The -Mac-Cell Network [21]

8.Mac Logic Imlementation

The goal of this project was to design and VLSI implementation of pipelined MAC for high-speed DSP applications. For designing the MAC, various architectures of multipliers and one bit full adders are considered. The static and dynamic one bit full adder was implemented as the basic block. The total process is coded with VHDL to describe the hardware. Finally, the whole process is implemented on VERTEX 7 Board.

MAC is composed of an adder, multiplier and an accumulator. Usually adders implemented are Carry-Select or Carry-Save adders, as speed is of utmost importance in DSP. One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle is the architecture of the MAC unit.

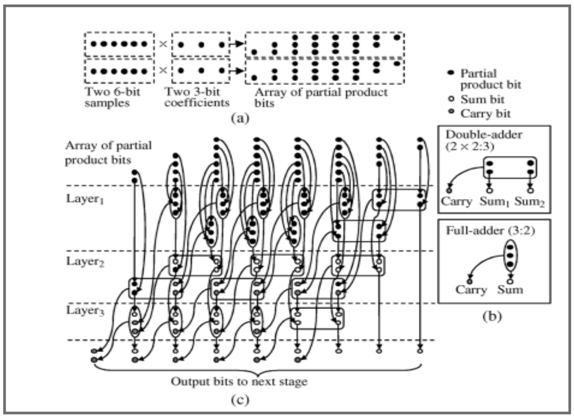


Figure 7: Structure Of The Mac-Cell Network [21]

9.Simulation Results

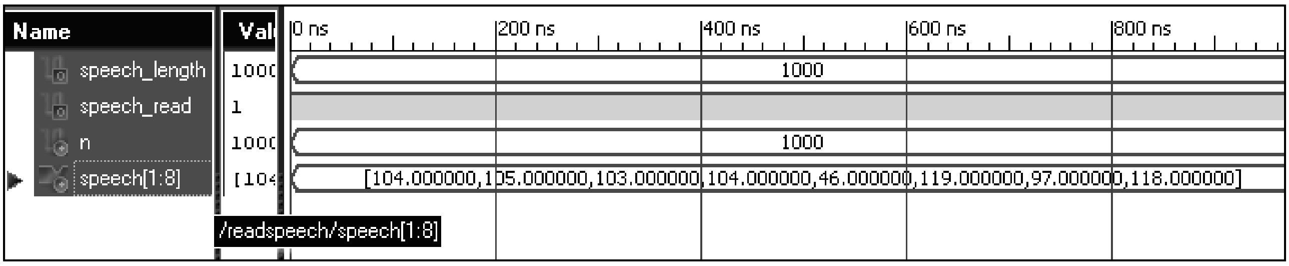


Figure 8: Reading Audio File Consisting Of 8 Samples

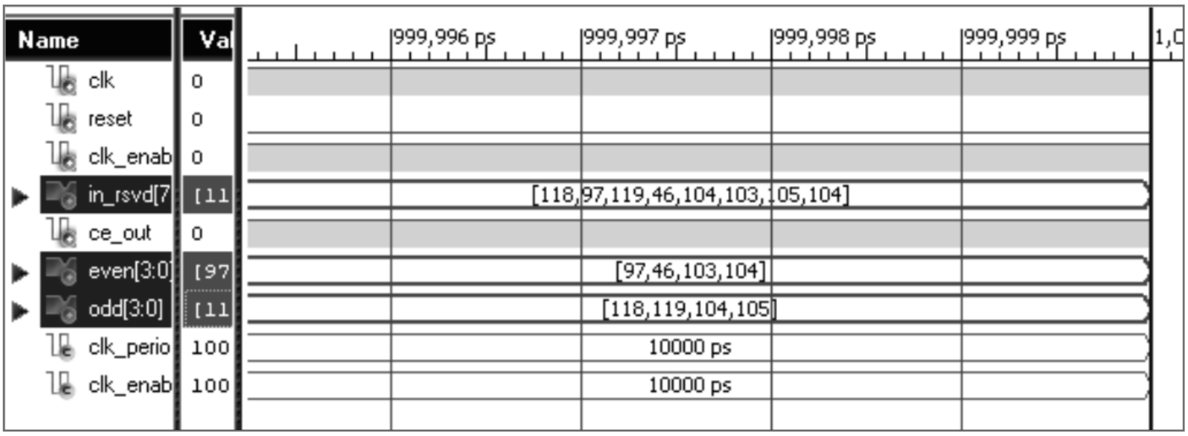


Figure 9: Separating Samples In Even And Odd Component

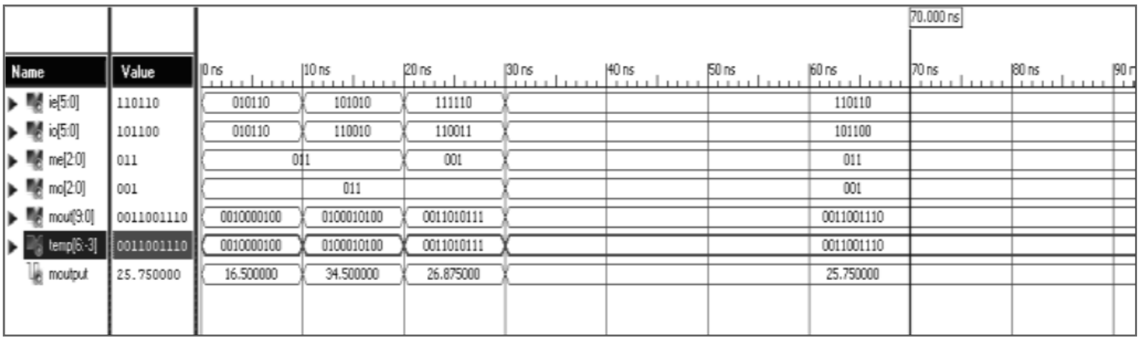


Figure 10: Result Of Mac Logic (Ieven = 5.5,10.5,15.5,13.5, Iodd = 5.5,12.5,12.75,11, Heven = 1.5,1.5,0.5,1.5 , Hodd = 1.5,1.5,1.5,0.5)

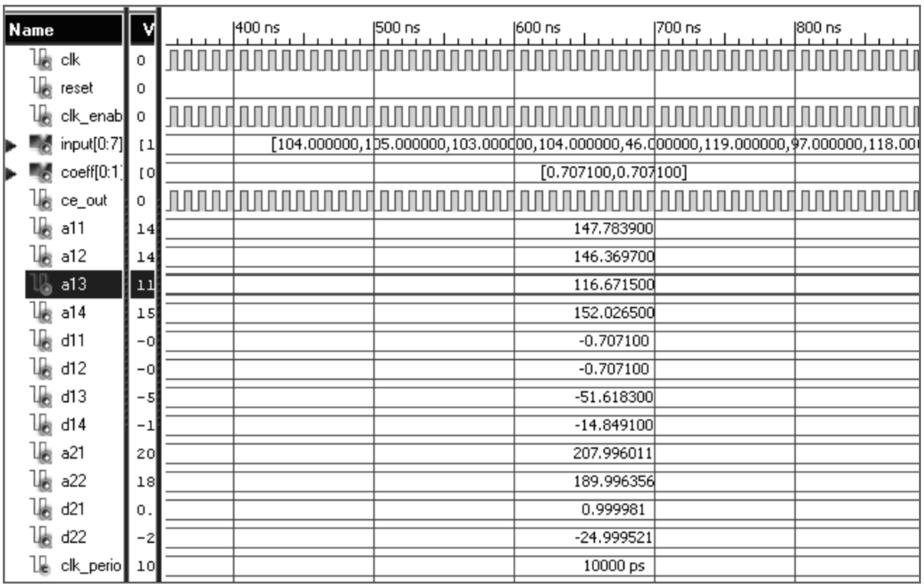


Figure 11: DWT Computation Of Samples Using Haar Wavelet

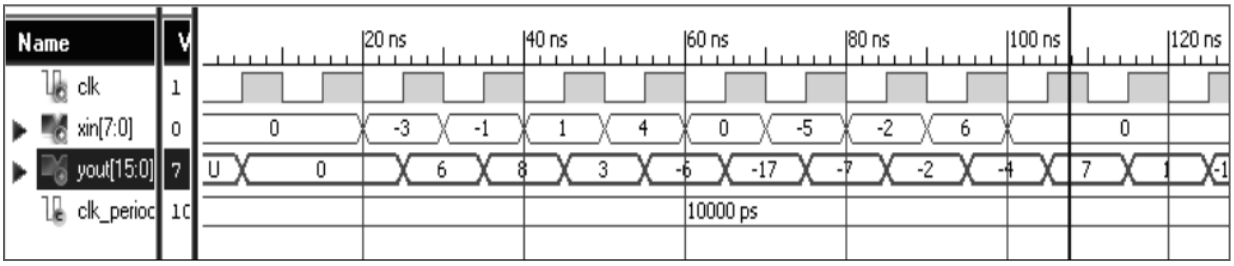


Figure 12: Direct Form Fir Filter In DWT Computation

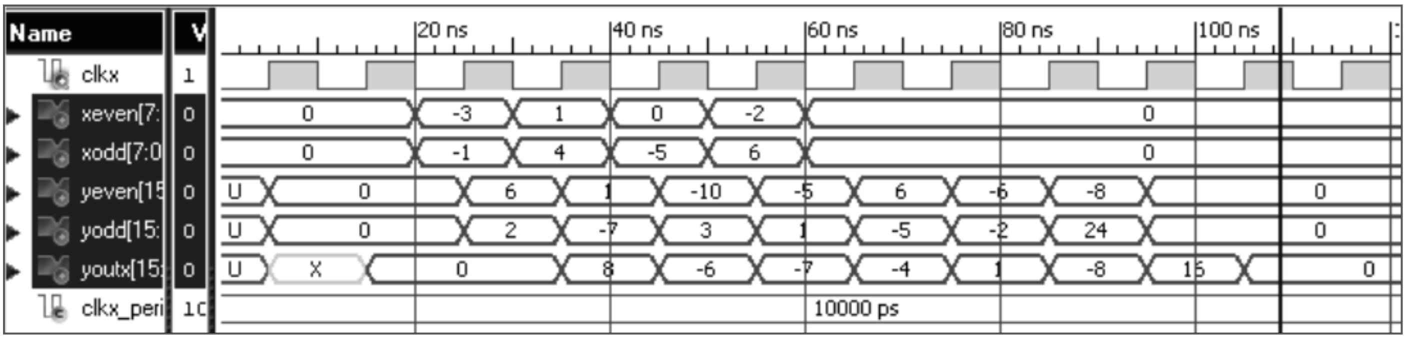
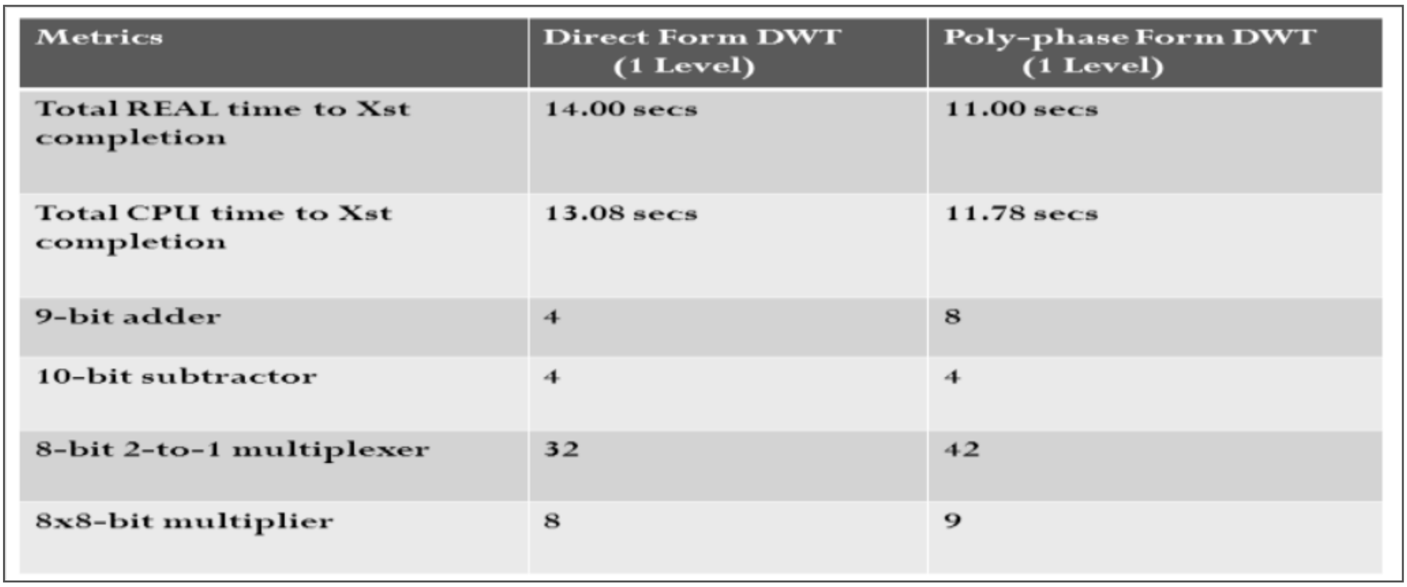


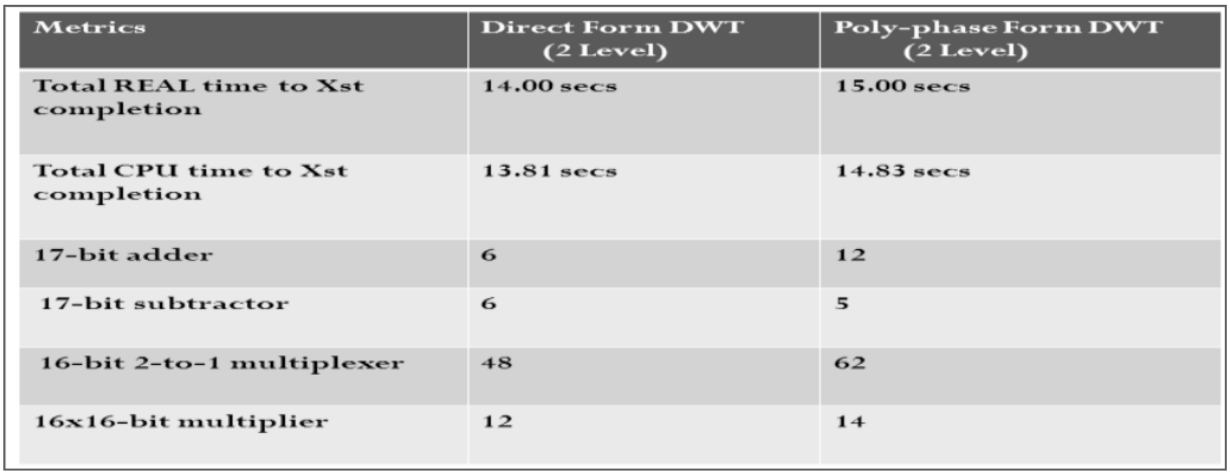
Figure 13: Poly-Phase Form Fir Filter In DWT Computation

10.Performance Evaluation
(Selected Device: 7vx330tffg1157-3)



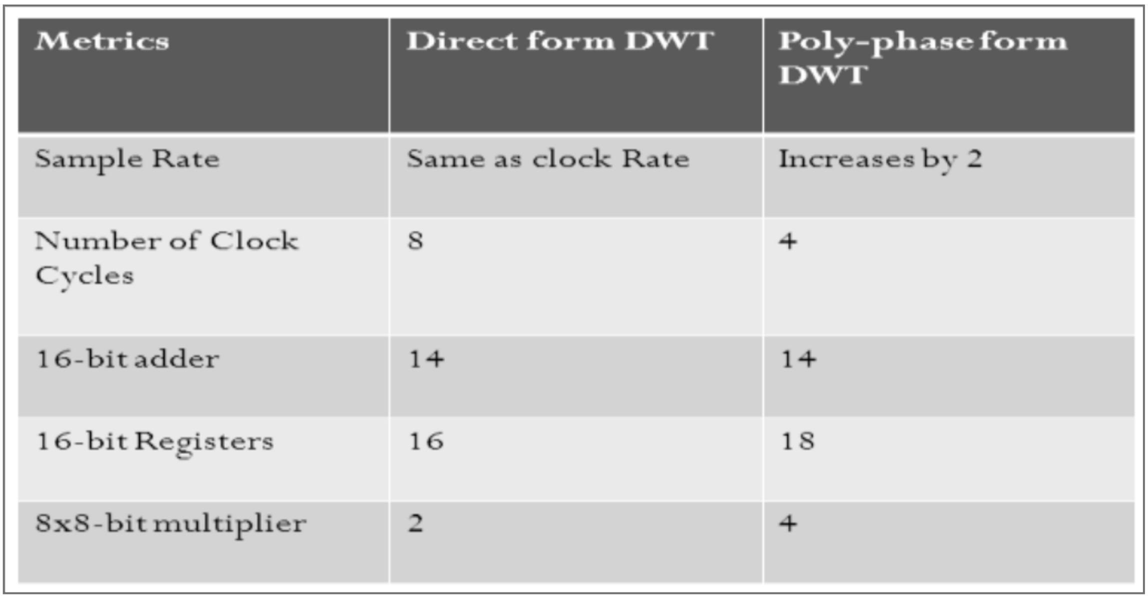
Metrics	Direct Form DWT (1 Level)	Poly-phase Form DWT (1 Level)
Total REAL time to Xst completion	14.00 secs	11.00 secs
Total CPU time to Xst completion	13.08 secs	11.78 secs
9-bit adder	4	8
10-bit subtractor	4	4
8-bit 2-to-1 multiplexer	32	42
8x8-bit multiplier	8	9

Table 1: Comparison Between Direct & Poly-Phase Form For 1st Level DWT



Metrics	Direct Form DWT (2 Level)	Poly-phase Form DWT (2 Level)
Total REAL time to Xst completion	14.00 secs	15.00 secs
Total CPU time to Xst completion	13.81 secs	14.83 secs
17-bit adder	6	12
17-bit subtractor	6	5
16-bit 2-to-1 multiplexer	48	62
16x16-bit multiplier	12	14

Table 2: Comparison Between Direct & Poly-Phase Form For 2nd Level DWT



Metrics	Direct form DWT	Poly-phase form DWT
Sample Rate	Same as clock Rate	Increases by 2
Number of Clock Cycles	8	4
16-bit adder	14	14
16-bit Registers	16	18
8x8-bit multiplier	2	4

Table 3: Performance Comparison With Selected Device

11.Conclusion

This architecture is better than the existing architecture in terms of computation time and efficient use of available resources. In order to assess the effectiveness of the proposed scheme, the design is simulated using XILINX and MATLAB.Poly-phase architecture is proposed to increase performance of DWT Design is targeted on FPGA with combined Pipeline and Parallel architecture to reduce Hardware Resources.

12.Future Scope

Synthesis filter banks to compute the inverse DWT can be implemented. The architectures of the filter banks can be further improved and the proposed architecture can also be extended to 2-D DWT computation

13.References

1. S. Mallat, ―A theory for multiresolution signal decomposition: The wavelet representation,‖ IEEE Trans. Pattern Anal. Mach. Intell., vol.11, no. 7, pp. 674–693, Jul. 1989.
2. J. Chilo and T. Lindblad, ―Hardware implementation of 1D wavelet transform on an FPGA for infrasound signal classification,‖ IEEE Trans. Nucl. Sci., vol. 55, no. 1, pp. 9–13, Feb. 2008.
3. S. Cheng, C. Tseng, and M. Cole, ―Efficient and effective VLSI architecture for a wavelet-based broadband sonar signal detection system,‖ in Proc. IEEE 14th ICECS, Marrakech, Morocco, Dec. 2007, pp. 593–596.
4. K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh, and K. E. Thomson, ―A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants,‖ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1266–1278, Jun. 2007.
5. C. Chakrabarti, M. Vishwanath, and R. M. Owens, ―Architectures for wavelet transforms: A survey,‖ J. VLSI Signal Process., vol. 14, no. 2, pp. 171–192, Nov. 1996.
6. C. Huang, P. Tseng, and L. Chen, ―Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform,‖ IEEE Trans. Signal Process., vol. 53, no. 4, pp. 1575–1586, Apr. 2005.
7. M. Martina and G. Masera, ―Multiplierless, folded 9/7–5/3 wavelet VLSI architecture,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 9, pp. 770–774, Sep. 2007.
8. A. Acharyya, K. Maharatna, B. M. Al-Hashimi, and S. R. Gunn, ―Memory reduction methodology for distributed-arithmetic-based DWT/IDWT exploiting data symmetry,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 4, pp. 285–289, Apr. 2009.
9. K. A. Kotteri, S. Barua, A. E. Bell, and J. E. Carletta, ―A comparison of hardware implementations of the biorthogonal 9/7 DWT: Convolution versus lifting,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 52, no. 5, pp. 256–260, May 2006.
10. C. Wang and W. S. Gan, ―Efficient VLSI architecture for lifting-based discrete wavelet packet transform,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 5, pp. 422–426, May 2007.
11. G. Shi, W. Liu, L. Zhang, and F. Li, ―An efficient folded architecture for lifting-based discrete wavelet transform,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 4, pp. 290–294, Apr. 2009.
12. 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.
13. C. Chakrabarti and M. Vishwanath, ―Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mapping on SIMD array computers,‖ IEEE Trans. Signal Process., vol. 43, no. 3, pp. 759–771, Mar. 1995.
14. A. Grzesczak, M. K. Mandal, and S. Panchanathan, ―VLSI implementation of discrete wavelet transform,‖ IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 4, no. 4, pp. 421–433, Dec. 1996.

15. C. Cheng and K. K. Parhi, ―High-speed VLSI implementation of 2-D discrete wavelet transform,‖ IEEE Trans. Signal Process., vol. 56, no.1, pp. 393–403, Jan. 2008.
16. S. S. Nayak, ―Bit-level systolic implementation of 1D and 2D discrete wavelet transform,‖ Proc. Inst. Elect. Eng.Circuits Devices Syst., vol.152, no. 1, pp. 25–32, Feb. 2005.
17. F. Marino, D. Guevorkian, and J. Astola, ―Highly efficient high-speed/low-power architectures for 1-D discrete wavelet transform,‖ IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 47, no. 12, pp.1492–1502, Dec. 2000.
18. T. Park, ―Efficient VLSI architecture for one-dimensional discrete wavelet transform using a scalable data recorder unit,‖ in Proc. ITC-CSCC, Phuket, Thailand, Jul. 2002, pp. 353–356.
19. S. Masud and J. V. McCanny, ―Reusable silicon IP cores for discrete wavelet transform applications,‖ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 6, pp. 1114–1124, Jun. 2004.
20. M. Vishwanath, ―The recursive pyramid algorithm for the discrete wavelet transform,‖ IEEE Trans. Signal Process., vol. 42, no. 3, pp. 673–677, Mar. 1994.
21. C. Zhang, C. Wang, and M. O. Ahmad, ―pipeline VLSI architecture for high-speed computation of the 1-D discrete wavelet transform,‖ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 10, pp. 2729–2740, Oct. 2010