International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-1, Issue-4, October 2012

VLSI Implementation of DWT Using Systolic Array Architecture

M. Nireesh Kumar, J. Hemanth, K. Durga Prasad

Abstract*: *This work presents an implementation of Discrete Wavelet Transform (DWT)using Systolic architecture in VLSI .This architecture consist of Input delay unit, filter, register bank and control unit. This performs the calculation of high pass and low pass coefficients by using only one multiplier. This architecture has been simulated and implemented in VLSI. The hardware utilization efficiency is more compared to the referred due to FBRA Scheme. The systolic nature of this architecture corresponding to a clock speed of 115.9 MHz has its advantage in Optimizing area, time and power. The architecture is simple, modular, and cascadable for computation of one, or multi-dimensional DWT.

Index Terms*: *DWT, Six tap FIR Filter, Systolic Array Architecture, Decomposition, FBRA.

I. INTRODUCTION

In recent years, there has been increasing important requirement to address the bandwidth limitations over communication networks. The advent of broadband networks (ISDN, ATM, etc) as well as compression standards such as JPEG, MPEG, etc is an attempt to overcome that's limitations. With the use of more and more digital stationary and moving images, huge amount of disk space is required for storage and manipulation purpose. Image compression is very important in order to reduce storage need.

Redundancies in video sequence can be removed by using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). DCT suffers from the negative effects of blackness and Mosquito noise resulting in poor subjective quality of reconstructed images at high compression.

Wavelet techniques represents real life non stationary signal which is powerful technique for achieving compression. In order to meet the real time requirements, in many applications, design and implementation of DWT is required. For the implementation, Systolic array (DWT-SA) architecture is used. The proposed VLSI architecture computes both high pass and low pass frequency coefficients in clock cycle and thus has efficient hardware utilization. Here, the user is required to input only the data stream and the high-pass and low-pass filter coefficients.

II. DISCRETE WAVELET TRANSFORM

Wavelet is a small wave whose energy is concentrated in time. Properties of wavelets allow both time and frequency analysis of signals. The Discrete Wavelet Transform (DWT), which is based on sub-band coding, is fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required.

A schematic of three stage DWT decomposition is shown in Fig. 1. In figure 1, the signal is denoted by the sequence a[n], where n is an integer. The low pass filter is denoted by L1 while the high pass filter is denoted by H1. At each level, the high pass filter produces detail information; b[n], while the low pass filter associated with scaling function produces coarse approximations, c[n and so on].

The filtering and decimation process is continued until the desired level is reached.

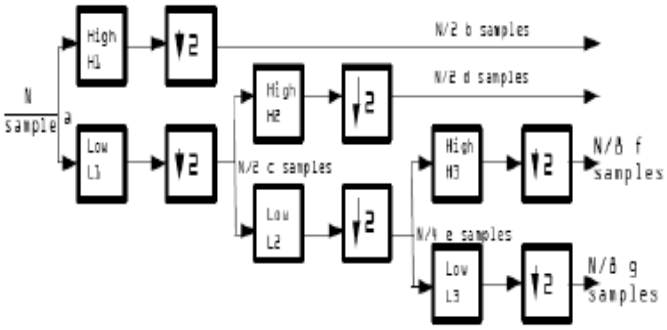

Figure1. Three stage DWT decomposition using pyramid algorithm.

The maximum number of levels depends on the length of the signal. The DWT of the original signal is then obtained by concatenating all the coefficients, starting from the last level of decomposition.

A. DATA DEPENDANCIES WITHIN DWT

The wavelet decomposition of a 1-D input signal for three stages is shown in Fig. 1. The transfer functions of the sixth order high pass (g (n)) and low pass (h (n)) FIR filter can be expressed in equations 1a and 1b:

Revised Manuscript Received on 30 October 2012.
* Correspondence Author
Mr. M.Nireesh Kumar*, M.Tech Student, Dept of ECE, JPNCE, Mahbubnagar, India
Mr.J.Hemanth, M.Tech , Asst Prof Dept of ECE,Vemu IT, Chittoor, India
Mr. K.Durga Prasad, M.Tech, Assoc Prof Dept of ECE,JPNCE, Mahbubnagar, India

For clarity, the intermediate and final DWT coefficients in Fig. 1, are denoted by a, b, c, d, e, f and g. The DWT computation is complex because of the data dependencies at different octaves. Eq. 3a-3n shows the relationship among a, b, c, d, e, f and g. We note that the data dependencies at various octaves are represented by identical symbols used in more than one octave equation.

$$High\ (z) = g0 + g1z\text{-}1 + g2z\text{-}2 + g3z\text{-}3 + g4z\text{-}4 + g5z\text{-}5 \quad (1a)$$

$$Low\ (z) = h + h0\ 1z\text{-}1 + hz2\text{-}2 + hz3\text{-}3 + h4z\text{-}4 + hz5\text{-}5 \quad (1b)$$

1st octave:

$$b(0) = g(0)a(0) + g(1)a(-1) + g(2)a(-2) + g(3)a(-3) + g(4)a(-4) + g(5)a(-5) \quad (3a)$$
$$b(2) = g(0)a(2) + g(1)a(1) + g(2)a(0) + g(3)a(-1) + g(4)a(-2) + g(5)a(-3) \quad (3b)$$
$$b(4) = g(0)a(4) + g(1)a(3) + g(2)a(2) + g(3)a(1) + g(4)a(0) + g(5)a(-1) \quad (3c)$$
$$b(6) = g(0)a(6) + g(1)a(5) + g(2)a(4) + g(3)a(3) + g(4)a(2) + g(5)a(1) \quad (3d)$$
$$c(0) = h(0)a(0) + h(1)a(-1) + h(2)a(-2) + h(3)a(-3) + h(4)a(-4) + h(5)a(-5) \quad (3e)$$
$$c(2) = h(0)a(2) + h(1)a(1) + h(2)a(0) + h(3)a(-1) + h(4)a(-2) + h(5)a(-3) \quad (3f)$$
$$c(4) = h(0)a(4) + h(1)a(3) + h(2)a(2) + h(3)a(1) + h(4)a(0) + h(5)a(-1) \quad (3g)$$
$$c(6) = h(0)a(6) + h(1)a(5) + h(2)a(4) + h(3)a(3) + h(4)a(2) + h(5)a(1) \quad (3h)$$

2nd octave:

$$d(0) = g(0)c(0) + g(1)c(-2) + g(2)c(-4) + g(3)c(-6) + g(4)c(-8) + g(5)c(-10) \quad (3i)$$
$$d(4) = g(0)c(4) + g(1)c(2) + g(2)c(0) + g(3)c(-2) + g(4)c(-4) + g(5)c(-6) \quad (3j)$$
$$e(0) = h(0)c(0) + h(1)c(-2) + h(2)c(-4) + h(3)c(-6) + h(4)c(-8) + h(5)c(-10) \quad (3k)$$
$$e(4) = h(0)c(4) + h(1)c(2) + h(2)c(0) + h(3)c(-2) + h(4)c(-4) + h(5)c(-6) \quad (3l)$$

3rd octave:

$$f(0) = g(0)e(0) + g(1)e(-4) + g(2)e(-8) + g(3)e(-12) + g(4)e(-16) + g(5)e(-20) \quad (3m)$$
$$g(0) = h(0)e(0) + h(1)e(-4) + h(2)e(-8) + h(3)e(-12) + h(4)e(-16) + h(5)e(-20) \quad (3n)$$

III. SYSTOLIC ARRAY

A systolic array is an arrangement of processors in an array where data flows synchronously across the array between neighbors, usually with different data flowing in different directions. Each processor at each step takes in data from one or more neighbors (e.g. North and West), processes it and, in the next step, outputs results in the opposite direction (South and East).

Systolic arrays are specialized form of parallel computing, where processors connected by short wires. An example of two-dimensional systolic array is given in the Fig 2 given below.

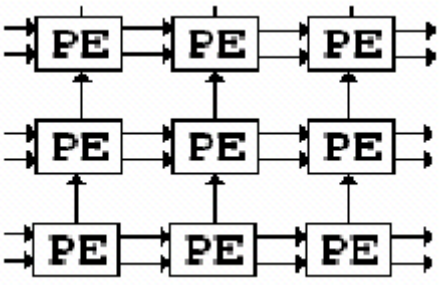

Fig 2: Example of two dimensional systolic array

The array given above takes in inputs parallel performs parallel processing and outputs the result. Systolic arrays do not lost their speed duo to their connection unlike any other parallelism. Cells i.e. Processing Elements (PE), compute data and store it independently of each other. Each cell (PE) is an independent processor and has some registers, Arithmetic, and Logic Units (ALUs).

The cells (Processing Elements) share information with their neighbors, after performing the needed operations on the data.

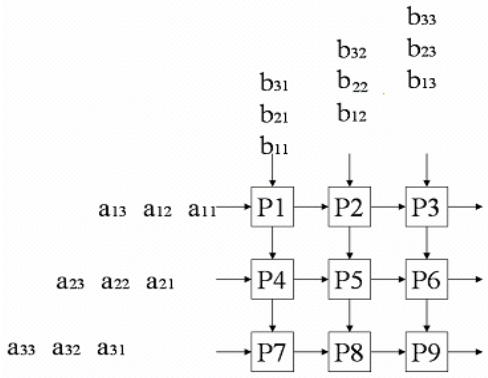

Fig 3: Example of Systolic Array Processing

Example of Systolic Array is shown in the Figure 4 above. Here each cell takes in inputs from top and left, multiplies those two number and stores in the local register which is inside the each Processing Element. After 9 clock pulses the result would be stored in each processing elements. In the full search block matching it needs N^2 subtractions, N^2 magnitude operations and N^2 magnitude accumulations are needed. Hence systolic arrays can be used to perform these operations duo to its advantageous properties like regularity, modularity and local communication.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

SA is extremely fast but expensive and difficult to implement and build. SA has both properties of Processor Pipelining and parallelism (which minimizes the computation time).

Processor Pipelining: Ideally at least one new instruction completes every time cycle.

Parallelism: Multiple jobs are allowed to perform simultaneously.

We need a high-performance, special-purpose computer system to meet specific application. I/O and computation imbalance is a notable problem. The concept of Systolic architecture can map high-level computation into hardware structures. Systolic system works like an automobile assembly line. Systolic system is easy to implement because of its regularity and easy to reconfigure.

The term "systolic" was first used in this context by H.T. Kung, then at CMU; it refers to the "pumping" action of a heart. Systolic architecture can result in cost-effective, high -performance special-purpose systems for wide range of problems.

III. THE PROPOSED SYSTOLIC ARRAY ARCHITETURE

The proposed systolic array (DWT-SA) architecture is an improved architecture. Here, only one set of multipliers and adders has been employed. The multiplier and adder set performs all necessary computations to generate all high pass and low pass coefficients.

A. DWT-SA ARCHITECTURE:

The design of DWT-SA is based on a computation schedule derived from Eq. 3a - 3n which is the result of applying the pyramid algorithm for eight data points ($N = 8$) to the six tap filter. We note that Eq. 1a and 1b represent the high pass and low pass components of the six tap FIR filter. The proposed DWT-SA architecture is shown in Fig. 4. It comprises of four basic units: Input Delay, Filter, Register Bank, and Control unit.

B. FILTER UNIT (FU):

The Filter Unit (FU) proposed for this architecture is a six tap non-recursive FIR digital filter whose transfer function for the high pass and low pass components are shown in Eq. 1 .This feature makes possible systolic implementation of DWT.

B.1. FILTER CELL (FC):

Eq. 1a-1b show that computations of the high pass and low pass DWT coefficients at specific time instants are identical except for different values of the LPF and HPF filter coefficients. The signed-number represents either positive, negative numbers or one positive and other negative numbers. To avoid this problem the proposed filter cell consists of invert and xor operation as shown in Figure as shown in Fig. 6.

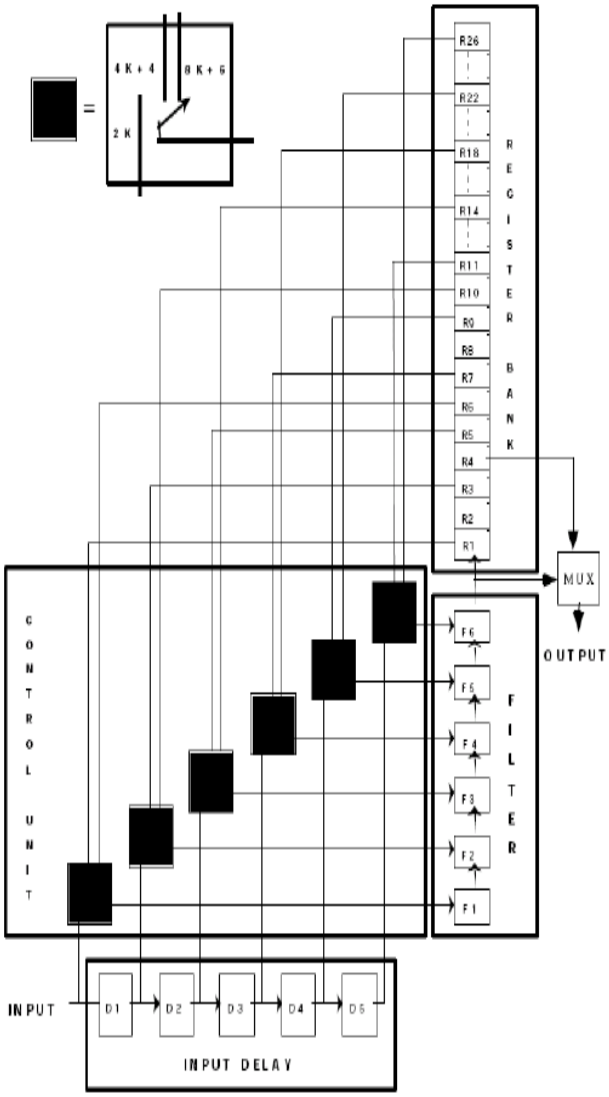

Fig 4. DWT- SA Architecture

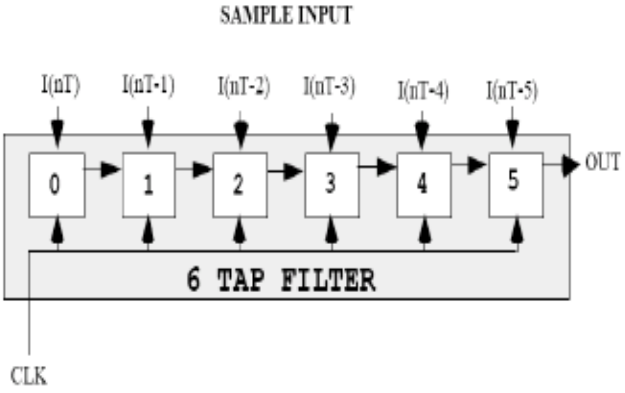

Fig 5.The systolic architecture of a six tap filter.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

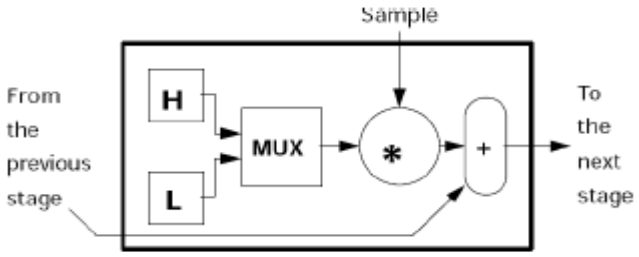

Fig 6. Proposed filter Cell

C. STORAGE UNIT

Two storage units are used in the proposed architecture: Input Delay and Register Bank. The data registers used in these storage units have been constructed from standard D latch. The following presents the structure of each storage unit.

C.1 Input Delay Unit (ID):

As shown in figure 4, five delays are connected serially. At any clock cycle each delay passes its contents to its right neighbor which results in only five past values being retained. The input of delay is applied to the switch.

C.2 Register Bank Unit (RB)

Several registers are required for storage of the intermediate partial results. 26 data registers connected serially are required to implement RB

D. CONTROL UNIT (CU):

The proposed DWT-SA architecture computes N coefficients in N clock cycles and achieves real time operation by executing computations of higher octave coefficients in between the first octave coefficient computations. The first octave computations are scheduled every N/4 clock cycles, while the second and third octaves are scheduled every N/2 and every N clock cycles, respectively.

D.1 REGISTER ALLOCATION

The next step in designing the DWT-SA architecture is the design of the Control Unit (CU) and the Register Bank (RB). The two components synchronize the availability of operands. The Forward Register Allocation (FRA) method uses a set of registers which are allocated to intermediate data on the first come first served basis. It does not reassign any registers to other operands once its contents have been accessed.

The FBRA scheme is similar, except that once the operand stored has been used; the register is reallocated to another operand. The FRA method is simpler, requires less control circuitry and it results however, in less efficient register utilization. In either scheme, the coefficient computations are periodic and hence, each register containing a specific variable will be reserved for the same variable in the next period.

D.3 COMPLETE DESIGN OF CU

The complete design of the Control Unit for DWT-SA architecture is shown in Fig. 7. The control unit uses switch, decoder and 4 FSM. The switching action is done by using FSM. State diagram is used to represent FSM. CU directs data from the Input Delay (ID), or the Register Bank (RB) to the Filter Unit (FU). By selecting particular select line, switching action of switch is done. And according to that switch, the data is applied to the filter cell.

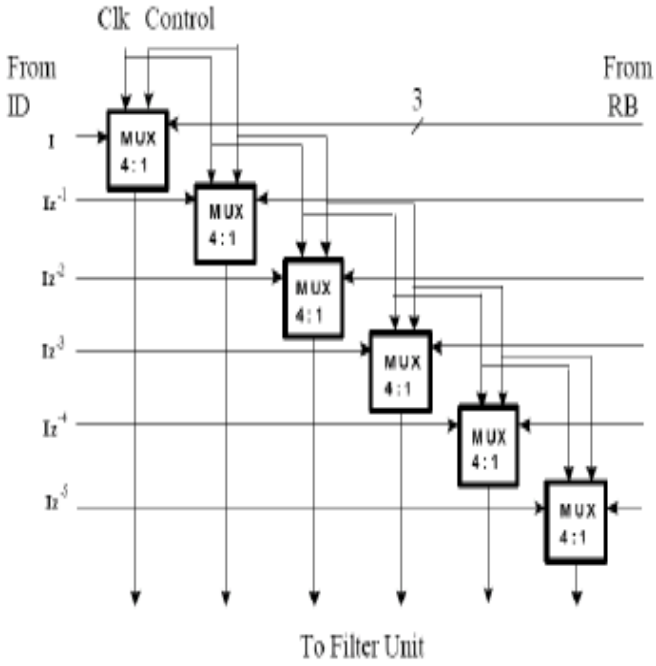

Fig 7. The Control Unit (CU)

The step by step process of entire project is shown in flow chart fig 8 and flow chart for design DWT is shown in fig 9. The first step starts by giving input image to the MATLAB which checks the image size and whether it is 2D/3D and converts the image into 2D, resize to 256 x 256 and then to data in Hexadecimal form.

After obtaining the data in terms of Hexadecimal, using Verilog DWT code is developed and the input to this obtained by interfacing MATLAB to Model Sim which implements the proposed DWT architecture and gives the Low/High pass coefficients and then this coefficients values are integrated and interfaced this data to MATLAB the resultant compressed image will be obtained.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

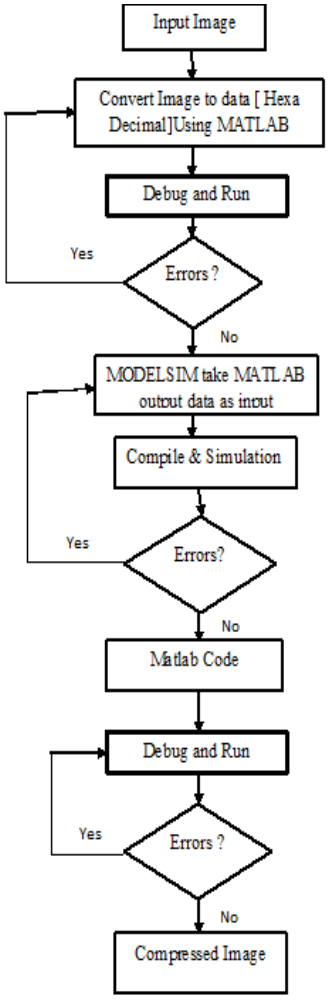

Fig 8 Flow chart of the entire Project.

E. Simulation Results

One of the most popular applications of DWT is video processing. With a frame rate of 30 frames/ sec, the video processor should process a complete frame in less than 33 ms .It has been found that the proposed architecture can execute the DWT computations on a monochrome 256 ×256 frame in 8.42 ns, and 115.902 MHz clock rate.

Fig 10 Input Images to Matlab.

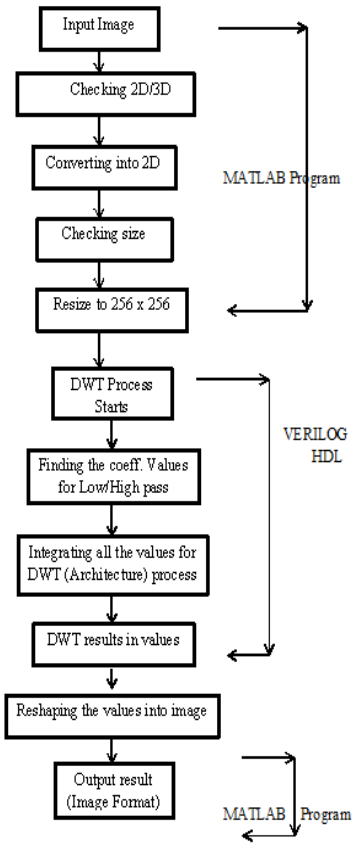

Fig 9 Flow chart of proposed architecture

The input image as shown in fig 10 is fed to the Matlab; it checks and converts the color image into monochrome image of standard size 256 x 256 and then to a Hexadecimal number and stores data in image text file. Using this data we interface the MATLAB to ModelSim for writing the code for DWT in Verilog. After compiling the code we perform the simulation for designed DWT.

The simulation results of DWT of ModelSim of image in fig 10 is shown in fig 11

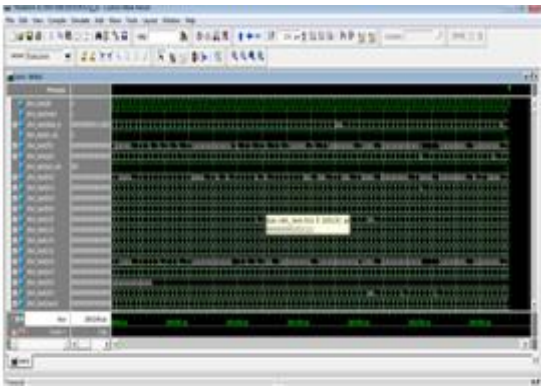

Fig 11 Model Simulator output of implemented DWT

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

After interfacing back the output of MODELSIM for implemented SA-DWT to MATLAB the output image is shown in fig 10

Fig 12 Output from Matlab after performing SA- DWT.

F Synthesis Results

By performing the synthesis to the design using Xilinx ISE it is observed that the minimum delay of architecture is 8.42 ns and it has maximum frequency of 115.902 MHz as shown in figure 12. We observed from the synthesis report that the architecture is required more area. But the speed is more. The fig.14 (a) and (b) shows the RTL schematic and its inner view obtained from Xilinx ISE of the implemented DWT after performing the synthesis.

G Comparison of Lifting-Based DWT with SA-DWT

Finally, we are comparing the results of lifting-based DWT with the Systolic Array DWT in table1 to show that the proposed architecture is having more hardware utilization efficiency and it is more efficient in terms of speed. But the area used or taken by the SA-DWT is more and the design will allows looks more complexity.

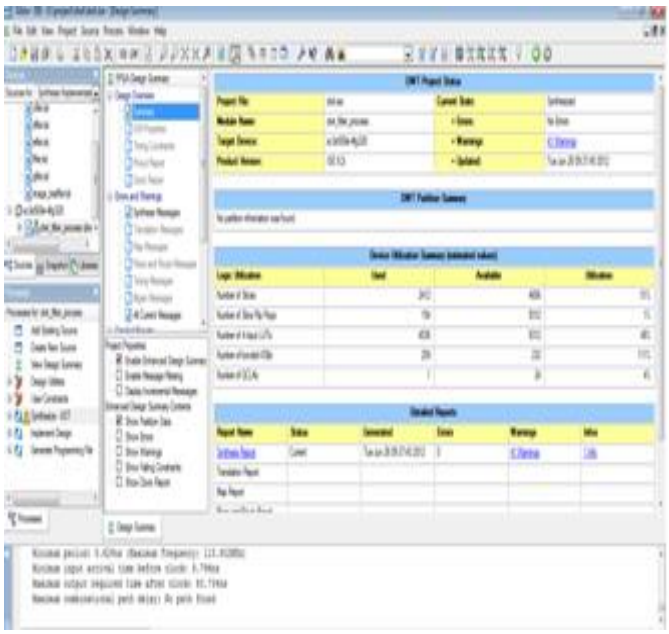

Fig 13 Synthesis report of design

Table1.Comparision of lifting based DWT with SA-DWT

S.No	Parameters	Lifting-DWT	SA-DWT
1	Minimum Delay	19.75 ns	8.42 ns
2	Maximum Frequency	50.62 MHz	115.90 MHz
3	No. of Slices Used	692	2412
4	No. of Slice Flip-flops Used	293	154
5	No. of IOB Bonded Used	105	259

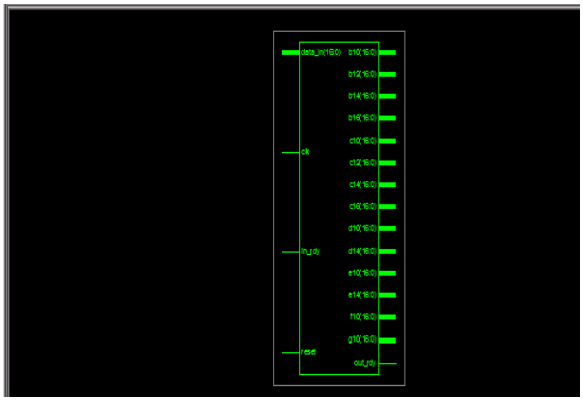

Fig 14 (a) RTL schematic view of design

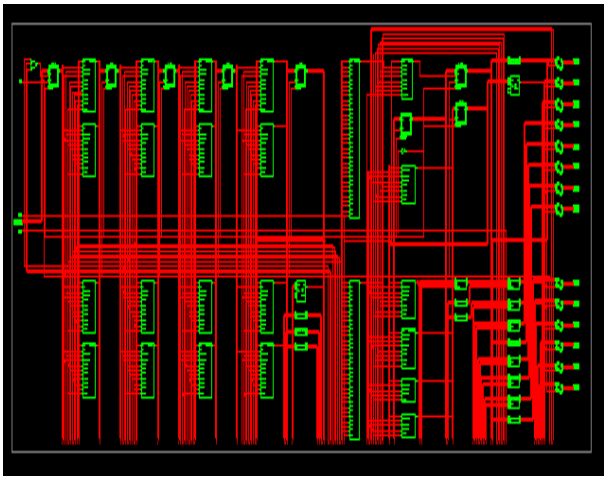

Fig 14 (b) RTL schematic inner view

IV. CONCLUSION

A systolic VLSI architecture for computing one dimensional DWT in real time has been presented. The architecture is simple, modular, cascadable, and has been implemented in VLSI. The implementation employs only one multiplier per filter cell, and hence results in a considerably smaller chip area. The DWT-SA architecture does not use any external or internal memory modules to store the intermediate results and therefore avoids the delays caused by access, read, write and refresh timing.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The architecture has been simulated in VLSI and has high hardware utilization efficiency than the referred. By performing the synthesis to the design using Xilinx ISE it is observed that the minimum delay of architecture is 8.42 ns and it has maximum frequency of 115.902 MHz

REFERENCES

1. I. Daubechies, “Orthonormal bases of compactly supported wavelets,” Comm. Pure Appl. Math, Vol. 41, pp. 906-966, 1988.
2. S. G. Mallat, “A theory of multiresolution signal decomposition: the wavelet representation,” IEEE Trans. on Pattern Recognition and Machine Intelligence, Vol. 11, No. 7, July 1989.
3. M. Vetterli and C. Harley, “Wavelets and filter banks: theory and design,” IEEE Transactions on Signal processing, Vol. 40, No. 9, pp. 2207-2232, 1992.
4. Y. Meyer, Wavelets: Algorithms and Applications, SIAM, Philadelphia, 1993
5. R. A. Devore, B. Jawerth and B. J. Lacier, “Image compression through wavelet coding,” IEEE Trans. on Information Theory, Vol. 38 .
6. O. Rioul and M. Vetterli, “Wavelets and signal processing,” IEEE Signal processing Magazine, pp. 14-38, Oct. 1991.
7. R. A. Gopinath, Wavelets and Filter Banks – New Results and Applications, PhD Dissertation, Rice University, Houston, Texas, 1993.
8. S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models", IEEE Trans. On Acoustics, Speech and Signal Processing Vol. 37, No. 12, pp. 2091-2110, Sept. 1989.
9. K. K. Parhi and T. Nishitani, “VLSI architectures for discrete wavelet transforms”, IEEE Trans. On VLSI Systems, pp. 191-202, June 1993.
10. Aware Wavelet Transform Processor (WTP) Preliminary, Aware Inc., Cambridge, MA.
11. A. D. Booth, "A signed binary multiplication technique", Quarterly Journal of Mechanics and Applied Mathematics, Vol. 4, pp. 236-240, 1951.

AUTHOR PROFILE

Mr. M.Nireesh Kumar, obtained my B.Tech degree from Kuppam Engineering College, affiliated to JNTU Hyderabad, Chittoor District, Andhra Pradesh, India in the year 2008. Now I am pursing M.Tech degree from Jaya Prakash Narayan College of Engineering, Mahbubnagar affiliated to JNTU Hyderabad, Mahbubnagar District, Andhra Pradesh, India. I participated in national seminars. I am interested & Research in the field of VLSI and image processing.

Mr. J.Hemanth, working as an Asst. Professor in Dept. of ECE, in Vemu IT, Chittoor (D), A.P, India, obtained my M.Tech degree from SVCET, affiliated to JNTU Anantapur, Chittoor (D)in the year 2012 , B.Tech degree from KEC, affiliated to JNTU Anantapur, Chittoor (D), A.P, India in the year 2010 and my interested areas are VLSI and Image Processing.

Mr. K.Durga Prasad working as an Assoc. Professor in Dept. of ECE, in JPNCE College, Mahbubnagar, A.P, India obtained M.Tech degree from JPNCE, affiliated to JNTU Hyderabad, Having 11 years of experience in teaching. My interested areas are Radar Systems, VLSI and Image Processing.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)