POWER, AREA & TIME ANALYSIS OF HAAR WAVELET USING VLSI ARCHITECTURE

Prof. Sonali R. Tavlare Author[1]
Sipna's college of Engg.. &
Technology, Amravati
sonalitavlare2011@gmail.com

Abstract *– This paper presents a VLSI implementation of Systolic architecture for discrete wavelet transform (DWT) of Haar wavelet. For the implementation of systolic array it required 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 have been simulated and implemented in VLSI. Optimized area, time and power obtained from this architecture for various devices.*

1. 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. The application of compression includes high definition television, video conferencing, and multimedia communication. 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 based on time-scale representations, provide an alternative to time-frequency representation based signal processing

Wavelet techniques represents real life non stationary signal which is powerful technique for achieving compression. Wavelet based techniques has Efficients parallel VLSI implementation, Low computational complexity, Flexibility in representing non stationary image signals. 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.

This paper presents a design and VLSI implementation of efficient systolic array architecture for computing DWT. 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.

2. 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. In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques.

Wavelets can be realized by iteration of filters with rescaling. The resolution of the signal, which is a measure of the amount of detail information in the signal, is determined by the filtering operations, and the scale is determined by up sampling and down sampling (sub sampling) operations.

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].

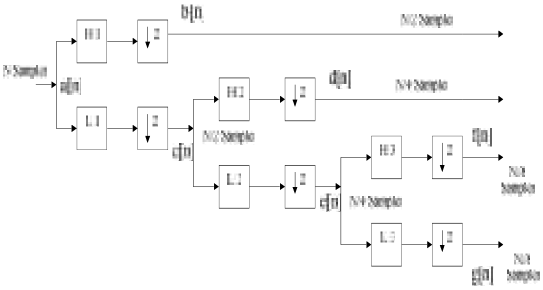

Figure 1. Three stage DWT decomposition using pyramid algorithm.

Here the input signal ‘a[n]’ has N samples. At the first decomposition level, the signal is passed through the high pass and low pass filters, followed by sub sampling by 2. The output of the high pass filter has N/2 samples and b[n]. These N/2 samples constitute the first level of DWT coefficients. The output of the low pass filter also has N/2 samples and c[n]. This signal is then passed through the same low pass and high pass filters for further decomposition. The output of the second low pass filter followed by sub sampling has N/4 samples and e[n]. The output of the second high pass filter followed by sub sampling has N/4 samples and d[n]. The second high pass filtered signal constitutes the second level of DWT coefficients. The low pass filter output is then filtered once again for further decomposition and produces g[n], f[n] with N/8 samples. The filtering and decimation process is continued until the desired level is reached. 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.

2.2 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 as follows:

$$\text{High}(z) = g_0 + g_1 z^{-1} + g_2 z^{-2} + g_3 z^{-3} + g_4 z^{-4} + g_5 z^{-5} \quad (1a)$$
$$\text{Low}(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} \quad (1b)$$

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.

1st octave:

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

2nd octave:

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

3rd octave:

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

3. 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 which performs all necessary computations to generate all high pass and low pass coefficients. In addition, since a set of registers controlled by a global clock is employed. This results in a simple and efficient systolic implementation for 1-D DWT computation.

The design of DWT-SA is based on a computation schedule derived from Eq. 3a - 3n which are the result of applying the pyramid algorithm for eight data points ($N = 8$) to the six tap filter. For the implementation of Haar it required two 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. 2. It comprises of four basic units: Input Delay, Filter, Register Bank, and Control unit.

3.1 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 Computation of any DWT coefficient can be executed by employing a multiply and accumulate method where partial products are computed separately and subsequently added. This feature makes possible systolic implementation of DWT. The latency of each filter stage is 1 time unit (TU). Since partial components of more than one DWT coefficient are being computed at any given time, the latency of the filter once the pipeline has been filled is also 1 (TU). The systolic architecture of a six tap filter is shown in Fig. 3. Here, partial results (one per cell) are computed and subsequently passed in a systolic manner from one cell to the adjacent cell.

3.2 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. By introducing additional control circuitry, computations of both high pass and low pass DWT coefficients can be executed using the same hardware in one clock cycle. The proposed filter cell therefore consists of only one multiplier, one adder, and two registers to store the high-pass and the low-pass coefficients, respectively. In such type of filter cell signed number multiplication problems are occurred. 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. 4.

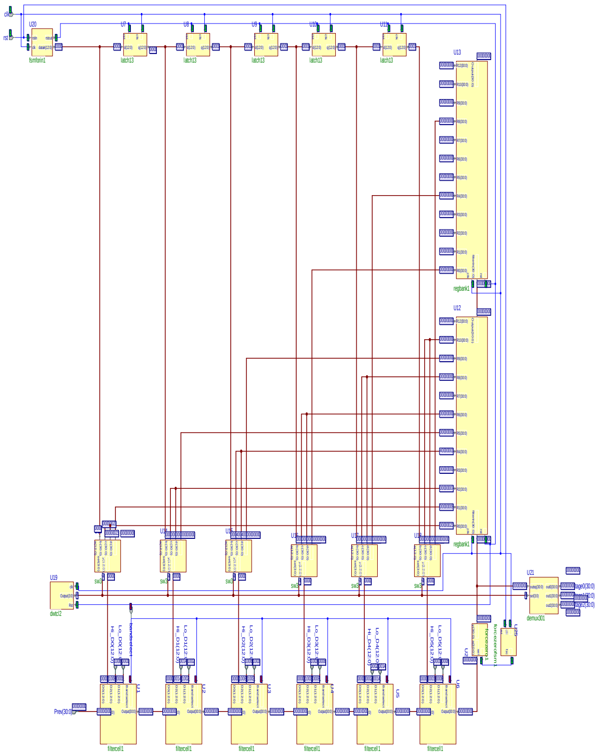

Figure 2. DWT-SA architecture.

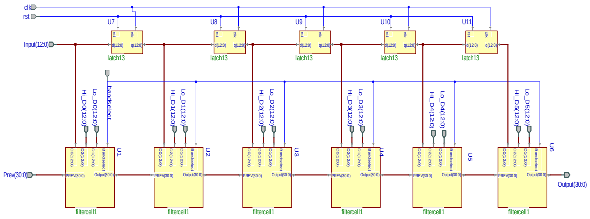

Figure 3. Systolic operation of the six tap filter and delay unit

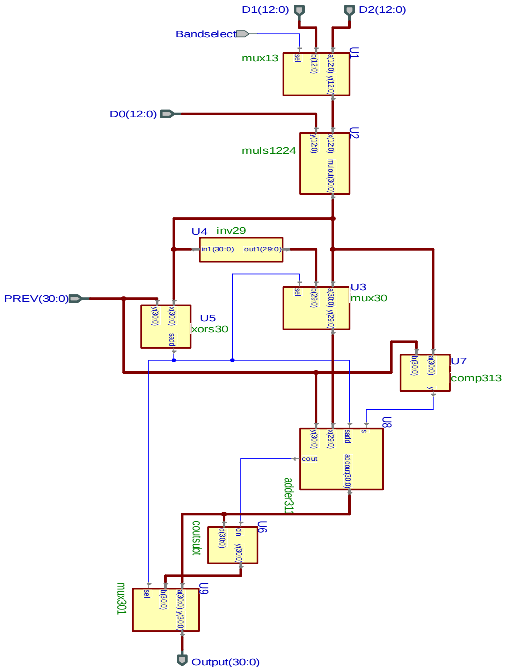

Figure 4. Filter cell.

3.3 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.

3.3.1 Input Delay Unit (ID)

Equations 3a and 3b show that the value of computed filter coefficient depends on the present as well as the five previous data samples. The negative time indexes in Equations 3 correspond to the reference starting time unit 0. It is therefore required that the present and the past five input data values be held in registers and be retrievable by the Filter Unit and the Control Unit. Figure 5 shows the block diagram of the Input Delay (ID) unit.

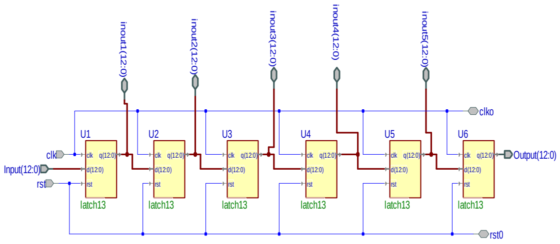

Figure 5: Input Delay Units (ID)

As shown in figure 5, 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.

3.3.2 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 as shown in figure 6.

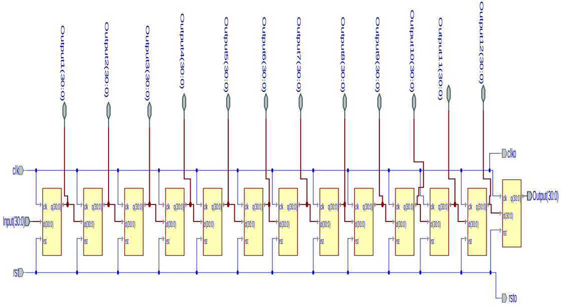

Figure 6: Register Bank (RB).

3.4 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.

In the DWT-SA architecture, a schedule based on filter latency of 1 is proposed to meet the real time requirements in some applications. The computations are scheduled at the earliest possible clock cycle, and computed output samples are available one clock cycle after they have been scheduled as shown in Table 1.

Table 1. Schedule for one complete set of computations.

Init Cycle	High - pass	Low - pass
1	b(0)	c(0)
2	-	-
3	b(2)	c(2)
4	d(0)	e(0)
5	b(4)	c(4)
6	f(0)	g(0)
7	b(6)	c(6)
8	d(4)	e(4)

The schedule presented in Table 2 is periodic with period N, and the hardware is not utilized in cycle KN+2 where k is a non-negative integer. The computation schedule in Table 2 corresponds to a high hardware utilization of more than 85%.

3.4.1 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 FRA method is simpler, requires less control circuitry and permits easy adaptation of the architecture for coefficient calculation of more than 3 octaves. It results however, in less efficient register utilization.

In the FRA register allocation approach where data moves systolically in one direction only, it is possible to increase the number of DWT decomposition octaves by placing additional registers in series after register R26. The new registers hold the intermediate coefficients needed for the computation of the next octave decomposition. Hardware utilization of the higher octave decomposition registers is inversely proportional to the order of computed coefficients.

3.4.2 ACTIVITY PERIODS

All the intermediate results, and the associated periods of activity are listed in Table 2.

Table 2. Activity periods for intermediate results

Sample	Available at cycle	Life period
c(0)	1	1 to 12
c(2)	3	3 to 14
c(4)	5	5 to 16
c(6)	7	7 to 18
e(0)	12	12 to 18
e(4)	16	16 to 38

The number of registers required in this architecture is directly proportional to the number of levels of DWT decomposition, and is calculated during the construction of the timetable of computations. This employs the FRA register allocation method and the number of registers is 26.

3.4.3 COMPLETE DESIGN OF CU

The complete design of the Control Unit for DWT-SA architecture is shown in Fig. 7. It schedules the computation of each DWT coefficient as shown in Table 4.

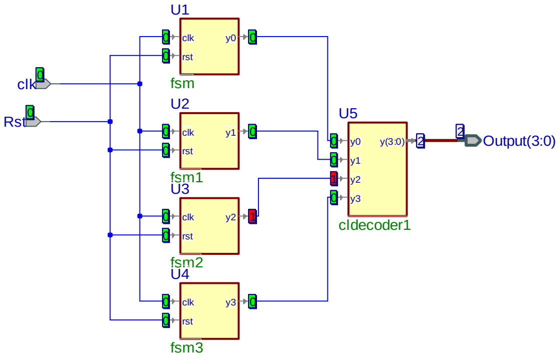

Figure 7. The Control Unit (CU)

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). The CU multiplexes data from the ID every second cycle, and from the RB in cycles 4, 6, and 8. In cycle 2, 6 CU remains idle, i.e. it does not allow any passage of data. Proper timing, synchronization as well as enabling and disabling of the CU are ensured by the global CLK signal.

Control logic consists of 4 FSM. The switch is operated on this state diagram. According to that it accepts data from input delay and register bank. Figure 8 shows the state diagram for FSM0.

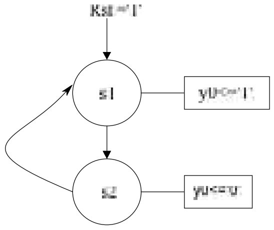

Figure 8: State Diagram for FSM0

Here, there are two states, S1 and S2. When reset is one, it produces output is one. Otherwise it produces zero. Like FSM0 all remaining three FSMs are implement.

3.5 TIMING CONSIDARATION

The first octave computations are scheduled every second clock cycle and hence the corresponding switch input is labeled 2k, where k is any non-negative integer. Moreover, its inputs are supplied directly by the ID. Second octave computations are executed in clock cycles 4 and 8 which are reflected by the label 4k+4 and 4(k+1)+4 respectively. The third octave computations are scheduled in clock cycle 10 or 8k+10. Both second and third octave computations use partial results from previous octave computations and therefore use inputs from Register Bank. First results are thus produced 43 (i.e., 5+38) clock cycles after the first input sample has entered the pipeline. Subsequent coefficients are available at the output the pipeline every 8 clock cycles. The DWT coefficients are output from the final filter stage.

3.6 SIMULATION RASULT

The high pass and low pass coefficients of haar are found from MATLAB. Table 3 shows these coefficients in hexadecimal form.

Table 3: High pass and low pass coefficients

Low pass coifficients	Hex form	high pass coifficients	Hex form
LO_D0	0012	HO_D0	1012
LO_D1	0012	HO_D1	0012

All the simulation is done in Active HDL. This paper represents the simulation result of Haar in terms of approximation and details coefficients. Simulation waveform for Haar as shown in figure 9 and 10**.** The result of DWT-SA is in hexadecimal format as shown in table 4 and 5. Depending on function, band select high pass or low pass coefficients. For function select line as ‘1’ low pass coefficients are selects while for ‘0’ high pass coefficients.

Table 4: Approximation coefficients of Haar

Approximation coefficients of db3		
First stage output	Second stage output	Third stage output
ca1	ca11	ca111
24	510	88B0
24	288	

Table 5: Details coefficients of Haar

Details coefficients of db3		
First stage output	Second stage output	Third stage output
cd1	cd11	cd111
0	0	0
	0	

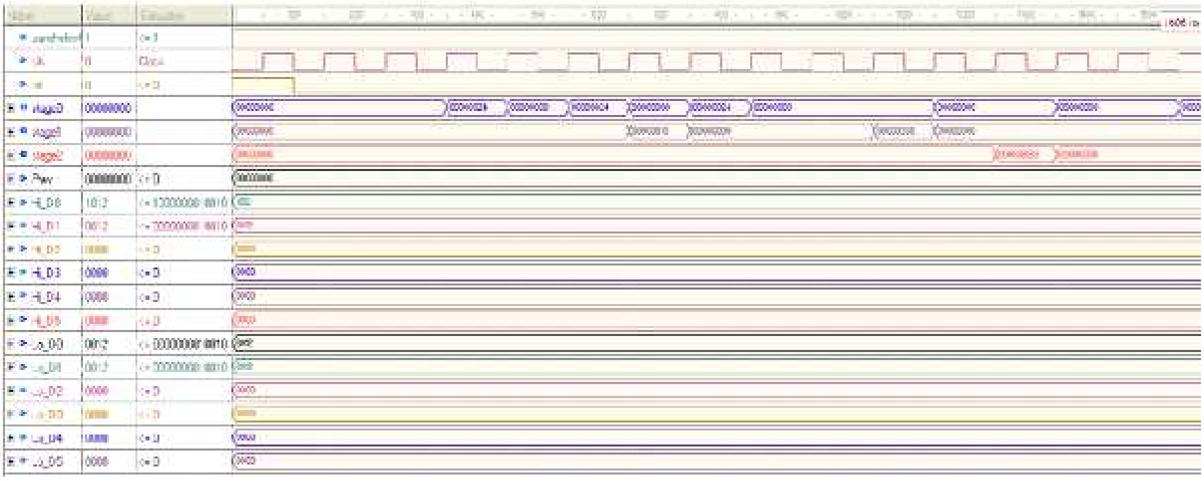

Figure 9: Simulation waveform for Haar when band select ‘1’-

Figure 10: Simulation waveform for Haar when band select ‘0’

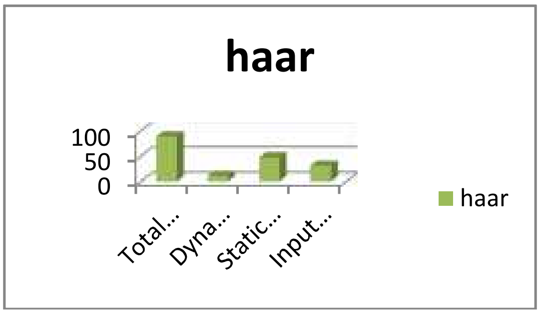

Figure 11**:** Power Analysis for three Wavelets

The Systolic array architecture of DWT is synthesized, placed and routed for cyclone device EP2C20F484C7 by using Quartus II. Figure 11, 12, 13 represents power, area and time analysis of haar and wavelets in graphical form. As given in figure 13, device EP2C20F484C7 provides total thermal power dissipation for Haar is less.

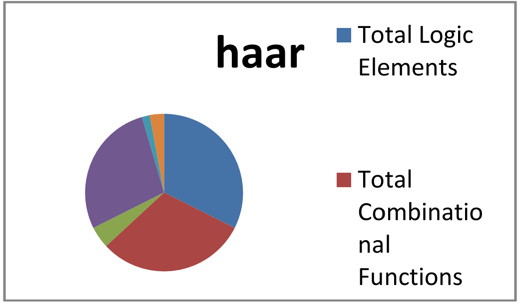

Figure 12**:** Area Analysis for three Wavelets

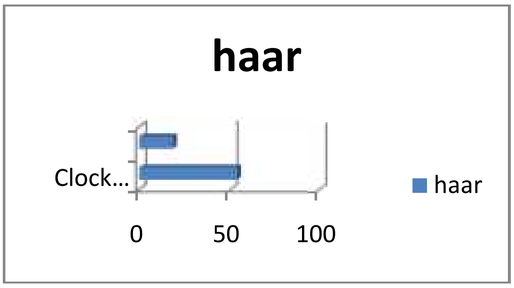

Figure 13: Time Analysis for three Wavelets

simulation waveform for approximation and details coefficients for three stages.

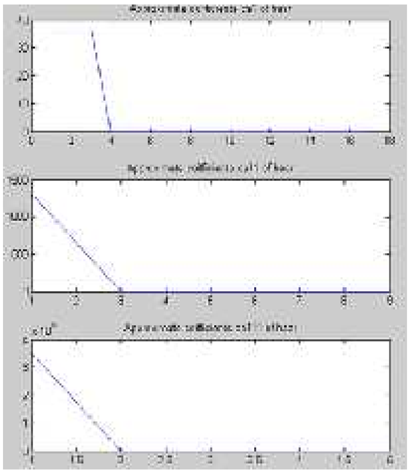

Figure 14: Simulation result of Approximation coefficients for Haar

For the verification of VLSI result, MATLAB is used. The approximation and details coefficients are obtained from MATLAB. Figure shows the

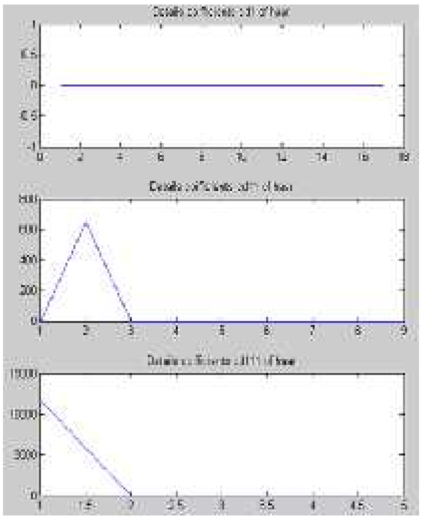

Figure 14: Simulation result of Details coefficients for Haar

4. 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.

Device C5AF256A7 provides total thermal power dissipation for Haar is 48.69 mW which is less. For EP2C70F89618 device, Haar provides dynamic thermal power dissipation is 13.51mW. The total logic elements required for haar is 5%.

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.

5. 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] R. A. Gopinath, Wavelets and Filter Banks – New Results and Applications, PhD Dissertation, Rice University, Houston, Texas, 1993.

[5] Y. Meyer, Wavelets: Algorithms and Applications, SIAM, Philadelphia, 1993.

[6] A. N. Akansu and R. A. Haddad, Multiresolution Signal Decomposition: Transform, Sub bands and Wavelets, Academic Press Inc., 1992.

[7] O. Rioul and M. Vetterli, “Wavelets and signal processing,” IEEE Signal processing Magazine, pp. 14-38, Oct. 1991.

[8] R. A. Devore, B. Jawerth and B. J. Lacier, “Image compression through wavelet coding,” IEEE Trans. on Information Theory, Vol. 38 .

Author Biographies:

Author: Prof.S.R.Tavlare
Email: Sonali1608@gmail.com
Assistant Proffesor in Sipna’s C.O.E.T, Amravati.