

VLSI Implementation of Discrete Wavelet Transform Using Systolic Array Architecture for Daubechies3 Wavelet

Rashmi Patil¹, Dr. M. T. Kolte²

¹Research Student, Dept. of Electronics, B. D. C. O. E., Sevagram, Wardha, Maharashtra, India

²Professor and H. O. D. OF ENTDC Dept, M. I. T. C. O. E., Pune, Maharashtra, India

Abstract-The wavelet transform has itself a useful tool in the field of 1-dimensional and 2-dimensional signal compression systems. Due to the growing importance of this technique, there is an increasing need in many working groups for having a development environment which could be flexible enough and where the performance of a specific architecture could be measured, closer to reality rather than in a theoretical way. Our work is new, simple and efficient VLSI architecture for computing the Discrete Wavelet Transform (DWT). The proposed architecture is systolic in nature, modular and extendible to 1-D DWT transform of any size. The systolic array architecture (DWT-SA) has been designed, simulated and implemented in VLSI. Being systolic in nature, the architecture can compute DWT at a rate of $N \times 10^6$ samples/sec corresponding to a clock speed of N MHz's.

Keywords: DWT, FRA, hardware efficiency, six tap Fir Filter, Systolic Array

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.) [1], [2] 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 [3], [4]. Redundancies in video sequence can be removed by using Discrete Cosine Transform (DCT). DCT suffers from the negative effects of blackness and mosquito noise resulting in poor subjective quality of reconstructed images at high compression [5].

Wavelet techniques represents real life non stationary signal which is powerful technique for achieving compression. Wavelet based techniques has efficient 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 [6], [7].

Wavelet based techniques have the following features:

- Basic function matches the human visual profiles resulting in high quality of reconstructed images.
- Flexibility in representing non stationary image signals.
- Low computational complexity [8].
- Efficient parallel VLSI implementation.

2. Discrete Wavelet Transform

DWT is based on sub band coding, is fast computation 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. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales. Wavelets can be realized by iteration of filters with scaling. 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 [2]. A schematic of three stage DWT decomposition is shown in Figure 1.

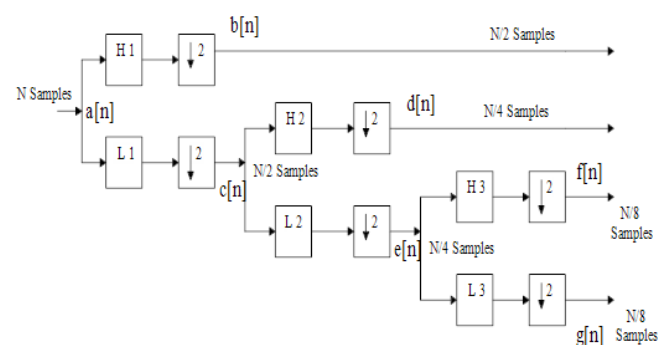


Figure 1: Three Stage DWT Decomposition using Pyramid Algorithm

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 approximation, $c[n]$.

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 $c[n]$. The signal is then passed through 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 filter 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.

3. Data Dependencies within DWT

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

$$\text{High}(z) = g_0 + g_1 z^{-1} + g_2 z^{-2} + g_3 z^{-3} + g_4 z^{-4} + g_5 z^{-5} \quad (1a)$$

$$\text{Low}(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} \quad (1b)$$

For clarity the intermediate and final DWT coefficients in Figure 1 are denoted by a, b, c, d, e, f and g . The DWT computations are complex because of the data dependencies at different octaves. Equations 2a-2n shows the relationship among a, b, c, d, e, f , and g .

1st octave:

$$b(0) = g(0)a(0) + g(1)a(-1) + g(2)a(-2) + \quad (2a)$$

$$g(3)a(-3) + g(4)a(-4) + g(5)a(-5)$$

$$b(2) = g(0)a(2) + g(1)a(1) + g(2)a(0) + \quad (2b)$$

$$g(3)a(-1) + g(4)a(-2) + g(5)a(-3)$$

$$b(4) = g(0)a(4) + g(1)a(3) + g(2)a(2) + \quad (2c)$$

$$g(3)a(1) + g(4)a(0) + g(5)a(-1)$$

$$b(6) = g(0)a(6) + g(1)a(5) + g(2)a(4) + \quad (2d)$$

$$g(3)a(3) + g(4)a(2) + g(5)a(1)$$

$$c(0) = h(0)a(0) + h(1)a(-1) + h(2)a(-2) + \quad (2e)$$

$$h(3)a(-3) + h(4)a(-4) + h(5)a(-5)$$

$$c(2) = h(0)a(2) + h(1)a(1) + h(2)a(0) + \quad (2f)$$

$$h(3)a(-1) + h(4)a(-2) + h(5)a(-3)$$

$$c(4) = h(0)a(4) + h(1)a(3) + h(2)a(2) + \quad (2g)$$

$$h(3)a(1) + h(4)a(0) + h(5)a(-1)$$

$$c(6) = h(0)a(6) + h(1)a(5) + h(2)a(4) + \quad (2h)$$

$$h(3)a(3) + h(4)a(2) + h(5)a(1)$$

2nd octave:

$$d(0) = g(0)c(0) + g(1)c(-2) + g(2)c(-4) + \quad (2i)$$

$$g(3)c(-6) + g(4)c(-8) + g(5)c(-10)$$

$$d(4) = g(0)c(4) + g(1)c(2) + g(2)c(0) + \quad (2j)$$

$$g(3)c(-2) + g(4)c(-4) + g(5)c(-6)$$

$$e(0) = h(0)c(0) + h(1)c(-2) + h(2)c(-4) + \quad (2k)$$

$$h(3)c(-6) + h(4)c(-8) + h(5)c(-10)$$

$$e(4) = h(0)c(4) + h(1)c(2) + h(2)c(0) + \quad (2l)$$

$$h(3)c(-2) + h(4)c(-4) + h(5)c(-6)$$

3rd octave:

$$f(0) = g(0)e(0) + g(1)e(-4) + g(2)e(-8) +$$

$$g(3)e(-12) + g(4)e(-16) + g(5)e(-20) \quad (2m)$$

$$g(0) = h(0)e(0) + h(1)e(-4) + h(2)e(-8) +$$

$$h(3)e(-12) + h(4)e(-16) + h(5)e(-20) \quad (2n)$$

4. Systolic Array (SA) Architecture for DWT

The design of DWT-SA is based on computation schedule derived from Equations 2a-2n, which are the result of applying the pyramid algorithm for $N=8$ to the six stage filter.

As shown in Equations 2a-2n there are eight first octave coefficients computations, four second octave coefficients and two third octave computations. Equations 1a-1b show that the high pass and low pass filter coefficient scan be computed using a six stage multiply and accumulate digital filter, where partial results are computed a each stage separately and then progressively passed to the next higher stage for accumulation. Equations 1a-1b suggests that due to filter coefficient calculations spanning over 5 clock cycles, a temporary storage is required for the incoming data samples.

This architecture is simple, modular, cascadable, and easily extendible for any block size DWT computation, and hence is well suited for VLSI implementation. The main contribution of our work is to implement and analyze DWT algorithm and hardware module used to compute this algorithm. These algorithm and architecture are implemented using Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) and then are simulate using Active HDL.

The DWT-SA architecture comprises of the four basic blocks:

4. 1. The Filter Unit (FU)
4. 2. The Input Delay Unit (ID)
4. 3. The Register Bank (RB) and
4. 4. The Control Unit (CU)

4. 1. The Filter Unit (FU)

The filter unit proposed for this architecture is a six-stage non-recursive FIR digital filter whose transfer functions for the high-pass and low-pass components are shown in Equations 2a-2n.

FIR filters have the following advantages:

- They can have an exact linear phase.
- They are always stable.
- The design method is generally linear.
- They can be realized efficiently in hardware.

The latency of each filter stage is 1. The systolic architecture of a six-stage filter is shown in Figure 2.

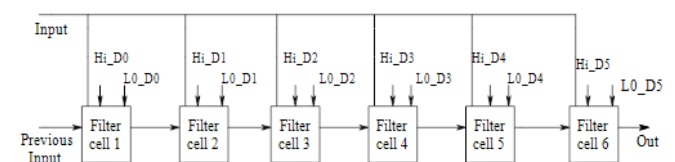


Figure 2: Systolic architecture of a six-stage filter

The filter behaves in a systolic manner by computing partial products of more than one coefficient at a time. The first multiply and accumulate stage computes the first partial product and passes it to the second filter stage where it is added to the second partial product. That repetitive action continues until a complete DWT coefficient is out from the sixth filter stage.

The filter consists of band select signal. If band selects is '1' then it selects low pass coefficients. If band selects '0' then it selects high pass coefficients. The filter is controlled by a clock and responds to its high and low levels for the computation of the high-pass and the low-pass DWT coefficient respectively. The RTL schematic of filter unit is shown in Figure 3.

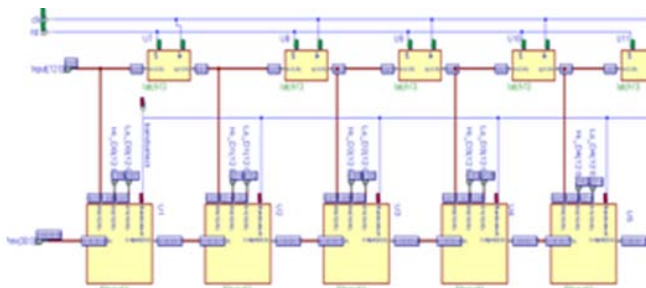


Figure 3: The RTL schematic of filter unit

Equations 1a-1b shows that except for different values of filter coefficients high pass and low pass computations at specific time instants are identical. By introducing additional control circuitry, both computations can perform by the same hardware.

Each multiplication must be executed in one clock cycle. The high pass coefficient calculation is performed during the clock cycle when selects '0', whereas the low pass coefficient calculation is performed during the clock cycle when selects '1'. The partial results are passed synchronously in a systolic manner from one cell to the adjacent cell. The filter cell consist therefore of only one multiplier, one adder and two registers to store high pass and low pass coefficients.

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 4.

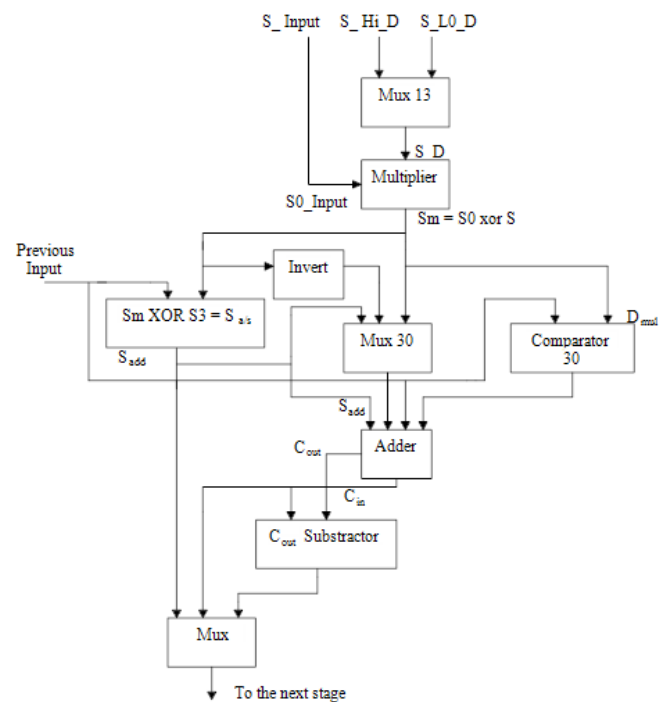


Figure 4: The Proposed Filter Cell

The filter cell consists of following devices:

- Multiplier
- Multiplexer
- Comparator
- Adder
- Not and Xor gate

MATLAB is the software which is used in our project to verify the result with VHDL simulation. The high pass and low pass coefficients are calculated from MATLAB which are in decimal form. We convert these into binary and padding some zeroes in it. Because the filter cell and multiplexer is designed for 31 bit, 13 bit respectively. Out of this 13 bit, 12 bits are data bits and 1 MSB bit is signed bit. The signed bit represents the positive or negative number. The RTL schematic of filter cell is shown in Figure 5.

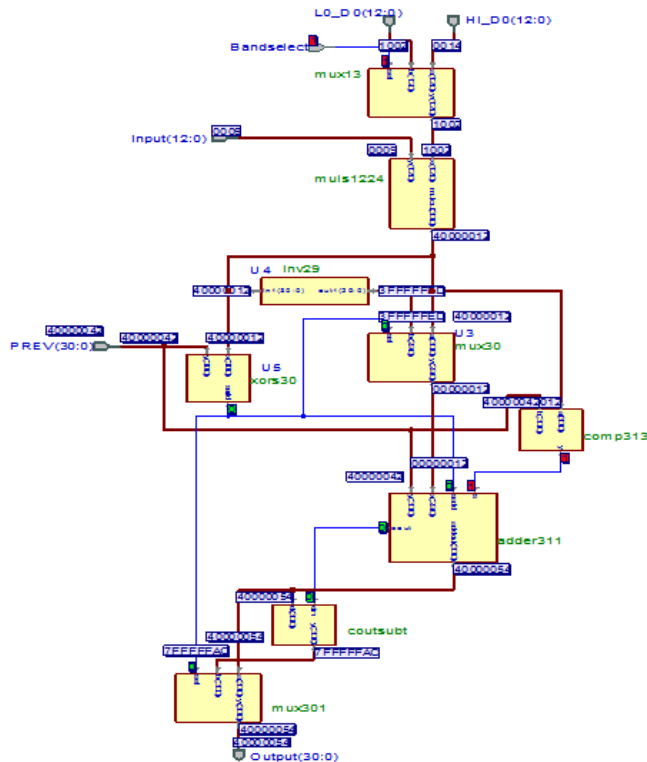


Figure 5: The RTL Schematic of Filter Cell

Two storage units are used in the proposed architecture.

- Input Delay Unit(ID) and
- Register Bank(RB)

The data registers and input delay used in these storage units have been constructed from standard D-latch. The following presents the structure of each storage unit.

4.2 Input Delay Unit (ID)

Equations 2a-2n 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 2 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 retrievable by the filter unit and the control unit. Figure 6 shows the block diagram of the input delay unit.

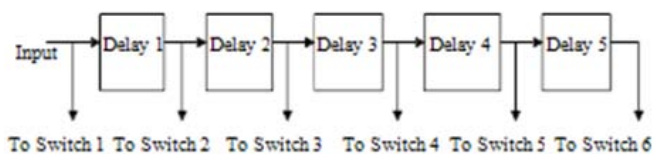


Figure 6: Input Delay Unit (ID)

As shown in figure 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. The RTL schematic of input delay unit is shown in Figure 7.

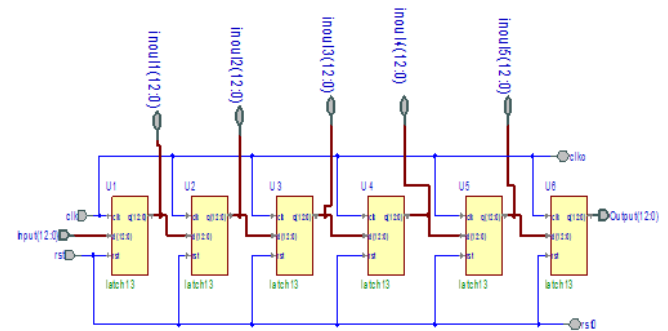


Figure 7: The RTL Schematic of Input Delay Unit

4.3 Register Bank (RB)

Several registers are required for storage of the intermediate partial results. Equations 1a and 1b justify the requirement for the register bank. The number of registers required is 26. 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. The registers are connected serially and are controlled by an overall clock. The output of one register is directly connected to the input of the adjacent register, and thus the register bank is implemented as shown in Figure 8 and its RTL schematic is shown in Figure 9.

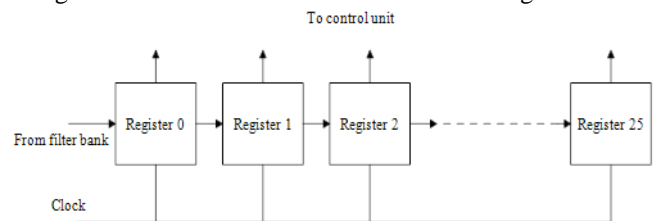


Figure 8: Register Bank (RB)

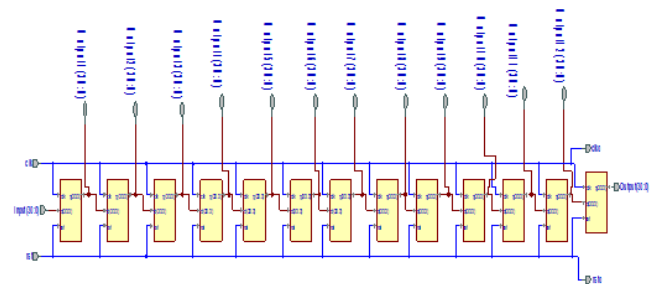


Figure 9: The RTL Schematic of Register Bank (RB)

4.4 Control Unit

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 for every $N/4$ cycles, while the second and third octaves are scheduled every $N/2$ and N clock cycles respectively. In DWT-SA architecture, schedule based on 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. The delay is minimized through the

pipeline facilitating real time operation. The computation schedule in table corresponds to a high hardware utilization of more than 85%.

Table 1: Schedule for one complete set of computations

Cycles	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	--	--
7	b(6)	c(6)
8	d(4)	e(4)
9	b(0)	c(0)
10	f(0)	Low-pass

4.4.1 Register Allocation

In the DWT-SA architecture Control Unit (CU) and the Register Bank (RB) synchronize the availability of operands. There are two schemes which can be employed for this purpose, namely

- Forward Register Allocation (FRA)
- The Forward-Backward Register Allocation (FBRA)

The FRA method uses a set of registers which are allocated to intermediate data on the first come first serve 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; there register is allocated to another operand. The FRA method is simpler, requires less control circuitry and permits easy adaptation of this architecture for coefficient calculation of more than 3 octaves. It results however in less efficient register utilization. In our work we used FRA register allocation.

4.4.2 FRA Register Allocation

To demonstrate the construction of the register allocation using the FRA approach, we will consider the case of computing the coefficients $c(0)$ and $c(2)$. As shown in Table 2 coefficient $c(0)$ is computed in clock cycle one. Whereas coefficient $c(2)$ is scheduled for computation in cycle 3. These two computed coefficients along with other four; $c(4), c(6), c(8), c(10)$ are needed for the computation of coefficient $e(0)$. According to table $e(0)$ computation is scheduled for cycle 4. However the six coefficients $c(0), c(2), c(4), c(6), c(8), c(10)$ will not be available until cycle 12, and therefore the calculation of $e(0)$ has to be scheduled for cycle $4+N$, or 12. Similarly, coefficient $e(4)$ is computed not in cycle 8, but $8+N$, thus cycle 16. The number of registers needed becomes apparent once one complete frame of computations has been scheduled.

Table 2: FRA Register Allocation

Cycle	Com	R1	R2	R3	R4	R5	R6
1	c(0)						
2		c(0)					
3	c(2)		c(0)				
4	e(0)	c(2)		c(0)			
5	c(4)		c(2)		c(0)		
6	g(0)	c(4)		c(2)		c(0)	
7	c(6)		c(4)		c(2)		c(0)
8	e(4)	c(6)		c(4)		c(2)	
9	c(0)		c(6)		c(4)		c(2)
10		c(0)		c(6)		c(4)	
11	c(2)		c(0)		c(6)		c(4)
12	e(0)	c(2)		c(0)		c(6)	
13	c(4)	e(0)	c(2)		c(0)		c(6)
14	g(0)	c(4)	e(0)	c(2)		c(0)	
15	c(6)		c(4)	e(0)	c(2)		c(0)
16	e(4)	c(6)		c(4)	e(0)	c(2)	
17	c(0)	e(4)	c(6)		c(4)	e(0)	c(2)
18		c(0)	e(4)	c(6)		c(4)	e(0)
19	c(2)		c(0)	e(4)	c(6)		c(4)
20	e(0)	c(2)		c(0)	e(4)	c(6)	
21	c(4)	e(0)	c(2)		c(0)	e(4)	c(6)
22	g(0)	c(4)	e(0)	c(2)		c(0)	e(4)
23	c(6)		c(4)	e(0)	c(2)		c(0)
24	e(4)	c(6)		c(4)	e(0)	c(2)	
25	c(0)	e(4)	c(6)		c(4)	e(0)	c(2)
26		c(0)	e(4)	c(6)		c(4)	e(0)
27	c(2)		c(0)	e(4)	c(6)		c(4)
28	e(0)	c(2)		c(0)	e(4)	c(6)	
29	c(4)	e(0)	c(2)		c(0)	e(4)	c(6)
30	g(0)	c(4)	e(0)	c(2)		c(0)	e(4)
31	c(6)			e(0)	c(2)		c(0)
32	e(4)	c(6)			e(0)	c(2)	
33	c(0)	e(4)	c(6)			e(0)	c(2)
34			e(4)	c(6)			e(0)
35	c(2)			e(4)	c(6)		
36	e(0)	c(2)			e(4)	c(6)	
37	c(4)	e(0)	c(2)			e(4)	c(6)
38	g(0)	c(4)	e(0)	c(2)		e(0)	e(4)

Table 2: FRA Register Allocation (continued)

Cycle	R7	R8	R9	R10	R11	R12	R13
1							
2							
3							
4							
5							
6							
7							
8	c(0)						
9		c(0)					
10	c(2)		c(0)				
11		c(2)		c(0)			
12	c(4)		c(2)		c(0)		
13		c(4)		c(2)		c(0)	
14	c(6)		c(4)		c(2)		
15		c(6)		c(4)		c(2)	
16	c(0)		c(6)		c(4)		
17		c(0)		c(6)		c(4)	
18	c(2)		c(0)		c(6)		
19	e(0)	c(2)		c(0)		c(6)	
20	c(4)	e(0)	c(2)		c(0)		
21		c(4)	e(0)	c(2)		c(0)	
22	c(6)		c(4)	e(0)	c(2)		
23	e(4)	c(6)		c(4)	e(0)	c(2)	
24	c(0)	e(4)	c(6)		c(4)	e(0)	

25		c(0)	e(4)	c(6)		c(4)	e(0)
26	c(2)		c(0)	e(4)	c(6)		
27	e(0)	c(2)		c(0)	e(4)	c(6)	
28	c(4)	e(0)	c(2)		c(0)	e(4)	
29		c(4)	e(0)	c(2)		c(0)	
30	c(6)		c(4)	e(0)	c(2)		
31	e(4)	c(6)		c(4)	e(0)	c(2)	
32	c(0)	e(4)	c(6)		c(4)	e(0)	
33		c(0)	e(4)	c(6)			e(0)
34	c(2)		c(0)	e(4)	c(6)		
35	e(0)	c(2)		c(0)	e(4)		
36		e(0)	c(2)		c(0)	e(4)	
37			e(0)	c(2)			e(4)
38	c(6)			e(0)	c(2)		

4.4.3 Activity Periods

All the intermediate and the associated periods of activity are listed in Table 3.

Table-3: Activity periods for intermediate results

Sample	Available Cycles	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 38
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 timetable of computations.

4.4.4 Complete Design of Control Unit

The complete design of the control unit for DWT-SA is shown in Figure 10.

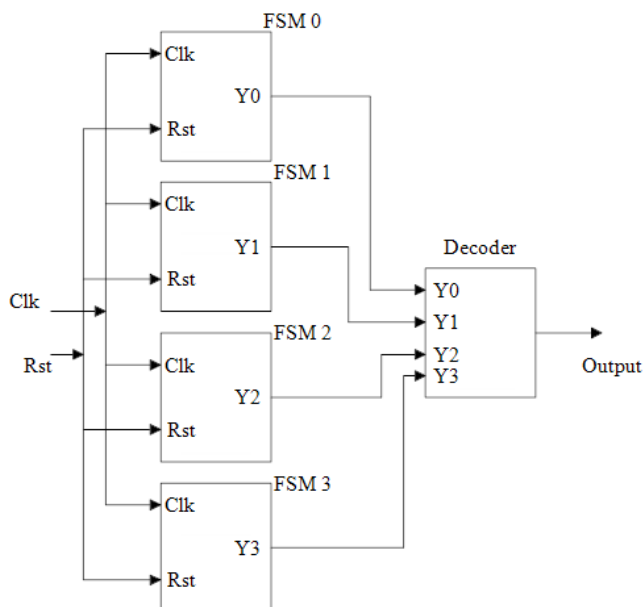


Figure 10: The Control Unit

The control unit uses switch, decoder, and four 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 insured by the CLK signal. The RTL schematic of control unit is shown in Figure 11.

Control logic consists of four FSM. The switch is operated on this state diagram. According to that it accepts data from input delay and register bank. Figure 12 shows the state diagram for FSM0.

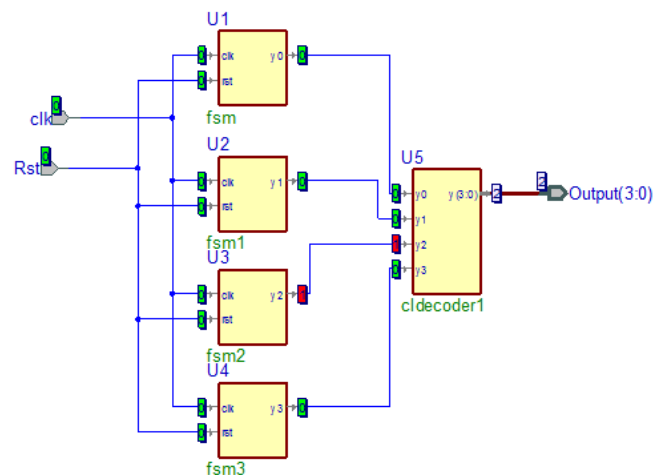


Figure 21: The RTL Schematic of Control Unit

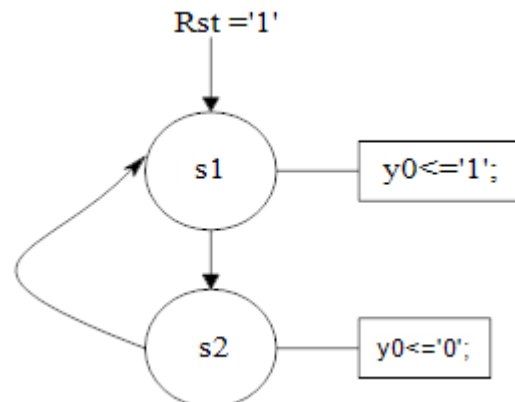


Figure 32: State Diagram for FSM0

Here, there are two states, S1 and S2. When reset is 1, it produces output 1. Otherwise it produces 0. Figure 13 shows the state diagram for FSM1. It consists of four states, S1, S2, S3, S4. When reset is 1, and the output is 0 for first clock. After that it goes to the next state, i. e. S2. At that time clock is incremented by 1. When clock cycle is less than 2, it produces output 0. But, when clock is greater than 2 then it goes to the next state. In S3 state, it produces output 1. This process is repeated for clock 3, 4, 5, 6, and 7. If clock is greater than 7 then it again comes to the S3 state and produces the output.

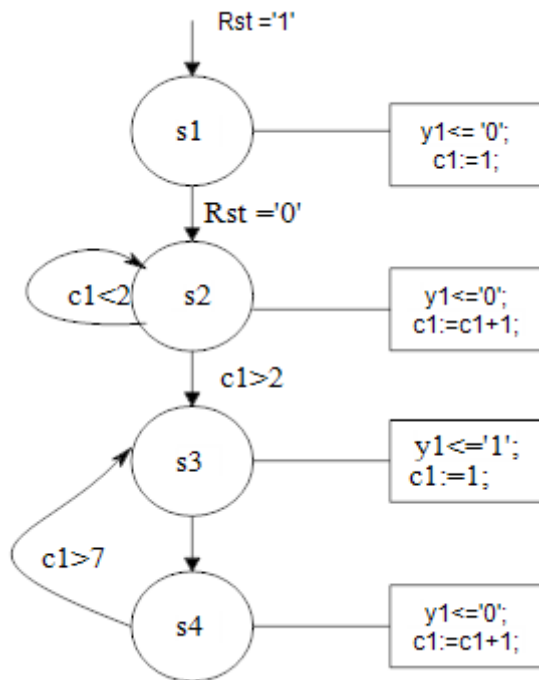


Figure 43: State Diagram for FSM1

Figure 14 shows state diagram for FSM2. When reset is 0, state S2 produces output 0. This occurs when clock is less than 6. For the next clock i.e. Clock is greater than 6, the state S3 produces output 1. But when clock is greater than 7, then state S4 changes its state to the previous state S3.

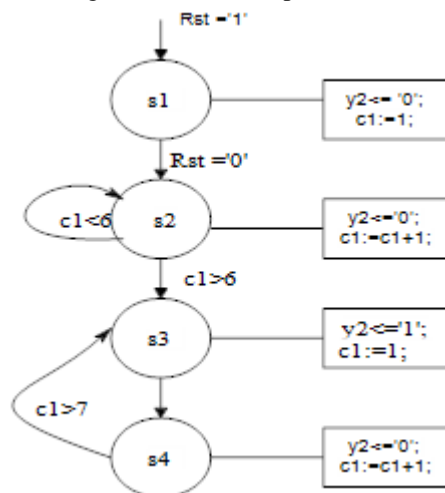


Figure 54: State Diagram for FSM2

The next is FSM3, which consists of S1, S2, S3, and S4. Figure 15 shows the state diagram for FSM3. When reset is 1, output y3 produces 0. In state S2, when clock cycle is less than 8, then output is 0. But in the next clock cycle the output is high. If clock cycle is greater than 7, it changes the state S4 to S3 and produces output high.

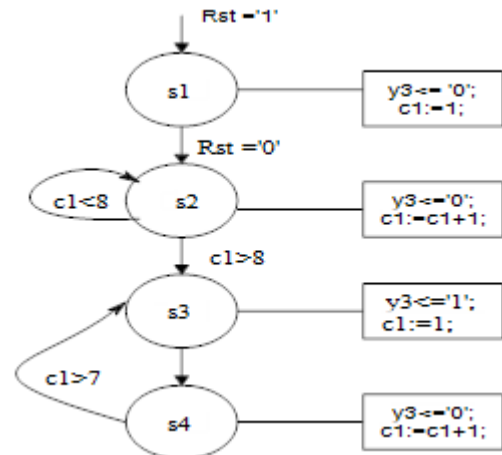


Figure 65: State Diagram for FSM3

In all this FSM, the output is high for particular clock signal. These all outputs are connected to decoder. The function of decoder is to select one of the outputs of FSM according to the select line. The output of decoder is connected to the select line of switch, by selecting particular select line; switching action of switch is done. And according to that switch, the data is applied to the filter cell.

5. DWT-SA Architecture

The proposed architecture is shown in Figure 16 and its RTL schematic is shown in Figure 17. It is obtained by systematic analysis of the FRA register allocation in Table 2 in conjunction with the filter equations 1a and 1b. Delay consists of the DWT-SA architecture consists of the latency period necessary to fill up the filter for the first time, plus the delay through the registers as described in Table 1. The output coefficients are obtained from the final filter stage.

The proposed DWT-SA architecture uses only one filter. The architecture is optimized in hardware by using only a single multiplier and adder set in each filter cell to generate all high pass and low pass coefficient. The coefficients which are present at the output of filter are stored in register.

The control logic controls the switching action of switch. The data which are coming from delay input and register bank is controlled by switch. According to the position of switch, one of the data is selected and performs filtering operations. This process is continued up to the 53 clock cycle.



Figure 76: DWT-SA Architecture

To select the particular output coming from filter unit, one of the small unit is used. When select line is zero, it selects one data. Force zero blocks is used to set the values zero. For controlling of this unit FSM is used. The select line is connected to the FSM. When reset is zero, it produces output. Otherwise it produces 0. This results in a simple and efficient systolic implementation for the computation of 1-DWT and hence is suitable for VLSI implementation.

The RTL view of DWT-SA for Daubechies3 is shown in Figure. This systolic array architecture is designed and implemented using quartus II by device: 'Cyclone; EP2C20F484C7'. Here two elements are used to implement DWT-SA architecture.

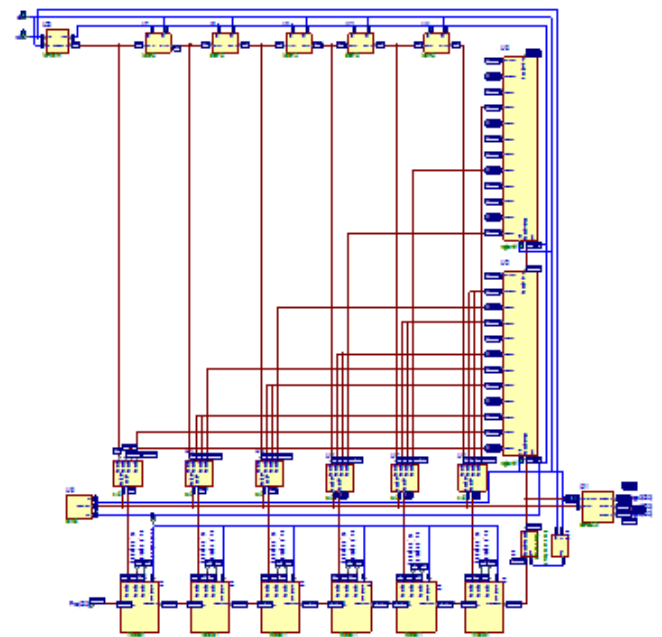


Figure 87: RTL schematic of DWT-SA architecture

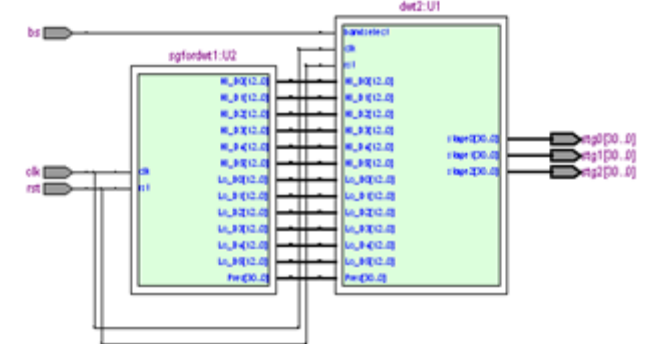


Figure 98: RTL view of DWT-SA for Daubechies3 wavelet

6. Simulation Results of DWT-SA Architecture for Daubechies 3 Wavelet

Simulated waveform of DWT-SA architecture for daubechies3 wavelet is shown in Figure 19 when band select is '0' and Figure 20 when band select is '1'. The result of DWT-SA is in the Hexadecimal format. For function select line as '1' low pass coefficients are selected while for '0' high pass coefficients.



Figure 109: Simulated waveform of DWT-SA architecture for daubechies3 wavelet when band select is '0'

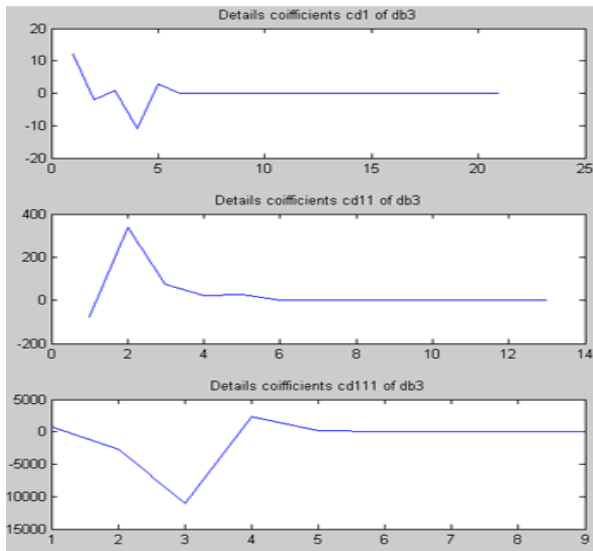


Figure 112: Simulation result of Details coefficients for Daubechies3

7. Conclusion

The main objective of this work is to develop a design resource for designers to implement systolic array architecture of DWT for Daubechies3 wavelet according to the design need such as clock speed, area and power. Systolic array architecture has efficient hardware utilization and it works with data streams of arbitrary size. The design is cascable, for computation of one, two and three decomposition level. The real aim of our work is to decompose the data up to third level; proposed architecture is the best option. The total logic elements required for Daubechies3 is 8% respectively. This 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.

References

- [1] N. Jayant, "High Quality Networking of Audio-Visual Information", IEEE Communications Magazine, pp. 84-95, September 1993.
- [2] E. A. Fox, "Advances in Interactive Digital Multimedia Systems", IEEE Computer Magazine, pp. 9-21, October 1991.
- [3] P. H. Ang, P. A. Ruetz and D. Auld, "Video Compression Makes Big Gains", IEEE Spectrum Magazine, pp. 16-19, October 1991.
- [4] A. K. Jain, "Image Data Compression: A Review", Proceedings of the IEEE, Vol. 69, No. 3, pp. 349- 389, March 1981.
- [5] K. R. Rao and P. Yip, Discrete Cosine Transform - Algorithms, Advantages, Applications. Academic Press: San Diego, 1990.
- [6] H. G. Musmann, P. Pirsch and H. J. Grailerr, "Advances in Picture Coding", Proceedings of the IEEE, Vol. 73, No. 4, pp. 523-548, April 1985.
- [7] A. bl. Netravali and B. G. Haskell, Digital Pictures, Plenum: New York, 1989.
- [8] FBI Systems Technology Unit: "Gray Scale Fingerprint image Compression Status Report, Nov. 4 1991.
- [9] Chao-Tsung Huang, Po-Chih Tseng, and Liang-Gee Chen, "Analysis and VLSI Architecture for 1-D and 2-

D Discrete Wavelet Transform", IEEE Transactions on signal processing, vol. 53, No. 4, April 2005

- [10] Chih-Chi Cheng, Chao-Tsung Huang, Ching-Yeh Chen, Chung-JrLian, and Liang-Gee Chen, "On-Chip Memory Optimization Scheme for VLSI Implementation of Line-Based Two-Dimensional Discrete Wavelet Transform", IEEE Transactions on circuits and systems for video technology, vol. 17, no. 7, July 2007
- [11] XinTian, Lin Wu, Yi-Hua Tan, and Jin-Wen Tian, "Efficient Multi-Input/Multi-Output VLSI Architecture for Two-Dimensional Lifting-Based Discrete Wavelet Transform", IEEE transactions on computers, vol. 60, no. 8, August 2011
- [12] Sze-Wei Lee, Soon-Chieh Lim, "VLSI Design of a Wavelet Processing Core", IEEE transactions on circuits and systems for video technology, vol. 16, no. 11, November 2006
- [13] Chao Cheng, Keshab K. Parhi, "High-Speed VLSI Implementation of 2-D Discrete Wavelet Transform", IEEE transactions on signal processing, vol. 56, no. 1, January 2008
- [14] AmitAcharyya, KoushikMaharatna, Bashir M. Al-Hashimi, Steve R. Gunn, "Memory Reduction Methodology for Distributed-Arithmetic-Based DWT/IDWT Exploiting Data Symmetry", IEEE transactions on circuits and systems—ii: express briefs, vol. 56, no. 4, April 2009
- [15] A. Grzeszczak, VLSI Architecture for Discrete Wavelet Transform, M. A. Sc. thesis, Department of Electrical Engineering, University of Ottawa, Canada, 1995.
- [16] Ms. YaminiS. Bute, 2Prof. R. W. Jasutkar, "Implementation of Discrete Wavelet Transform Processor For Image Compression", International Journal of Computer Science and Network (IJSN) Volume 1, Issue 3, June 2012 www. ijsn. org ISSN 2277-5420

Author Profile

Ms. Rashmi Patil received her B. Eng. Degree in Electronics & Communication from S. S. G. M. C. E. Shegaon, India in 2010 and M. Tech degree in Electronics from R. T. M. N. U. Nagpur, India. Currently she is a research scholar in B. D. C. O. E., Sevagram, India. Her areas of interest are applied electronics, VLSI, VHDL, and Low Power optimization.

Dr. M. T. Kolte has completed his B. Tech, M. E., and Ph. D in Electronics and Telecommunication. He is working as Head of Dept. in M. I. T. C. O. E., Pune, India. He has presented and published many papers in National and International Conference.