



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Efficient FPGA Architecture for Dual Mode Integer Haar Lifting Wavelet Transform Core

^{1,2}Haider Ismael Shahadi, ²Razali Jidin, ²Wong Hung Way and ²Yasir Amer Abbas

¹Department of Electrical Engineering, University of Babylon, Hila, Babil, Iraq

²Electronic and Communication Engineering, Tenaga National University (UNITEN), Putrajaya, Malaysia

Abstract: Discrete Wavelet Transform (DWT) becomes a main part of numerous signal and image processing applications. The lifting scheme has been developed recently as a flexible tool for constructing lossless bi-orthogonal wavelets. This study proposes efficient and high performance architecture for dual mode (forward and inverse) Integer Haar Lifting Wavelet Transform (IHLWT) core. The presented architecture minimizes the hardware area by proposing multipliers-free design. It performs IHLWT with only a single adder and subtractor which have reconfigurable input buses to perform forward or inverse transformations based on mode control signal. The proposed architecture has been implemented on Field Programmable Logic Array (FPGA). The design has been developed by using Hardware Description Language (HDL); this makes the design easily portable across FPGA devices from any vendor. The synthesis of the design showed that it has high operating-frequency and requires low hardware area and low power consumption for various Xilinx FPGA devices. The designed processor has been successfully implemented and tested on Xilinx Spartan6-SP601 Evaluation Board. The implemented hardware has been tested in real time by using many recording audio signals. All the implemented hardware results were identical 100% with IHLWT software results. This design is appropriate for applications require real time and low power consumption IHLWT processor such as wireless multimedia sensor networks.

Key words: Lifting wavelet transform (LWT), field programmable gate array (FPGA), integer to integer haar filter, dual mode architecture, hardware of DWT

INTRODUCTION

The discrete Wavelet Transforms (DWT) are used in many applications such as multimedia processing, numerical analysis, object recognition, data compression and video monitoring and data security. It becomes the most important signal processing techniques which is used to efficiently decompose and reconstruct non-stationary signals. The major advantage of DWT over other transformations is that it performs multi resolution analysis for signals with localization in both frequency and time (Mallat, 1989).

Since, DWT is a computation intensive procedure, various hardware architectures have been recently proposed in order to match real time applications requirements. Generally, the DWT architectures can be classified into three types: Convolution-based DWT such as Altermamm *et al.* (2011); Farahani and Eshghi (2007), Huang *et al.* (2005a), Maamoun and Namane (2009) and Pang and Chauhan (2008), B-spline-based DWT such as

Huang *et al.* (2005b) and lifting-based DWT such as Andra *et al.* (2002), Angelopoulou *et al.* (2008), Aziz and Pham (2012), Chen (2004), Huang *et al.* (2004), Kuzume *et al.* (2004), Liao *et al.* (2004), Shi *et al.* (2009) and Zhang *et al.* (2012). The disadvantages of the convolution based type are it has high redundancy computation operations, requires auxiliary memories to save intermediate results and needs long critical path delays. All These disadvantages result inefficient hardware implementations. As for the B-spline, it requires more hardware resources than the conventional type, but it has lower critical path delay and latency. In comparison to the two categories, the lifting-based type supports parallel input data processing, as well as lower transform computational. Moreover, it is “in-place” operations calculation which does not require any extra memory to save the intermediate results (Sweldens, 1995, 1996). The advantages of the lifting-based consequently will determine its hardware implementation that will yield less area, lower power and higher speed.

Most above literatures use multipliers to design filters of Lifting Wavelet Transform (LWT). In Aziz and Pham (2012), vthe authors propose a multiplier-free processor to implement 5/3 LWT filter. The authors have used shift registers and adders to design their processor. In Andra *et al.* (2002), the authors have proposed separate forward and inverse LWT processors. However, the proposed architectures are complex that require large hardware area.

This study proposes a high speed, low hardware area and low power consumption architecture for dual mode integer to integer Haar type lifting wavelet transformation. Both modes (forward and inverse) utilize common resources without any extra hardware. The proposed architecture is very simple and does not require any multiplier or auxiliary memories. It requires only a single adder, subtractor and two shifters for both modes. The proposed shifter is represented by a simple wiring circuit to perform two mathematical operations, multiplication and floor rounding function in a single simple step.

The proposed architecture performs in parallel the prediction and update in the forward mode and inverse of update and predict in inverse mode. The architecture processes pair of input samples at each clock cycle. Moreover, it is designed in a generic way to be utilized either in the serial input applications which needs splitter to split data to odd and even, or in the parallel input applications which fetch pairs of data from RAM or registers.

MATERIALS AND METHODS

Discrete wavelet transform (DWT): In the convolution based DWT approach (Mallat, 1989), an input signal (S) is filtered independently by two filters: a low-pass filter (h) and a high-pass filter (g). Then, the outputs from the filters are sub-sampled by two to produce the low-pass (approximation (A)) and high-pass (details (D)) sub-band outputs coefficients as shown in Fig. 1a. The two filters (h and g) are named decomposition filters bank. The original signal (S) can be reconstructed by up-sampling A and D streams by insertion zeros between each two samples and then filtering the outputs streams by a reconstruction low-pass and high-pass filters (h and g) respectively. Finally, the filters output are added together to obtain the synthesized signal (S') as shown in Fig. 1b. The two reconstruction filters bank (h and g) are the inverse of decomposition filters (h and g) respectively.

For multi-resolution DWT decomposition, the low-pass sub-band (A) is further analysed in a similar way in order to find the second-level of decomposition and the

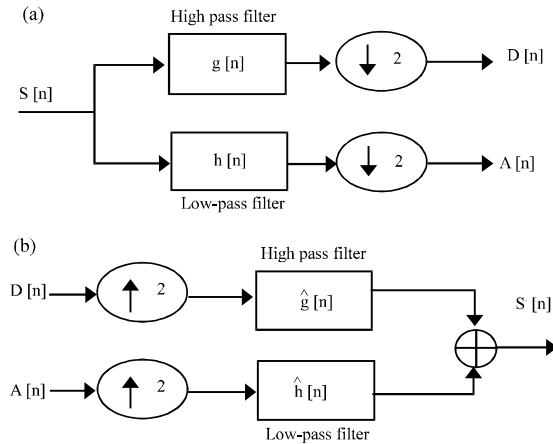


Fig. 1(a-b): Signal decomposition and reconstruction in a one level DWT, (a) 1-level DWT decomposition and (b) 1-level DWT reconstruction

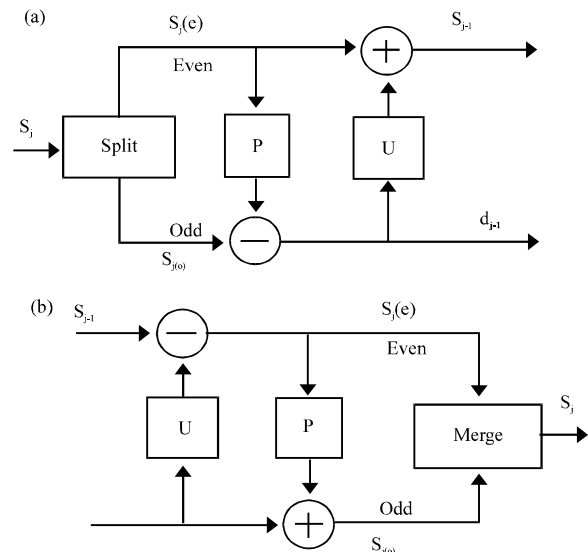


Fig. 2(a-b): One level decomposition and reconstruction of lifting wavelet transform (LWT), (a) Forward lifting scheme and (b) Inverse lifting scheme

procedure repeats. The inverse of the process follows similar multi-level synthesis filtering in order to reconstruct the output signal.

Integer lifting scheme based DWT: Sweldens (1996) proposed the lifting scheme where all the DWT operations can be executed in parallel. Hence it is faster than the convolution based DWT. The integer to integer lifting scheme is computed through three steps: Split, Predict (P) and Update (U). Figure 2 shows the lifting scheme processes.

The first step is to split an input signal (S) into even (S (e)) and odd (S (o)) indexed samples as described in Eq. 1.

Splitting:

$$\text{Spllit } (S_j): S_j(e), S_j(o) \quad (1)$$

The second step is to predict the odd samples based on the evens to find the details coefficients according to Eq. 2:

Predicate:

$$d_{j,1} = S_j(o) - \text{floor}(P \{S_j(e)\}) \quad (2)$$

The third step is to update the even samples based on the odds. The output represents the smooth coefficients of the integer DWT as given in Eq. 3.

Update:

$$S_{j,1} = S_j(e) + \text{floor}(U\{S_j(o)\}) \quad (3)$$

where, floor (z) is a function that finds the largest integer less than or equal to z. This rounding function is used in the integer to integer type of lifting scheme and it is not required for the floating type of LWT.

The inverse of the lifting wavelet transform that is shown in Fig. 2b can be easily computed by three steps: Undo-update, undo-prediction and merging odds and evens together. Equation 4-6 demonstrate these three steps.

Inverse-update:

$$S_j(e) = S_{j,1} - \text{floor}(U\{S_j(o)\}) \quad (4)$$

Inverse-predict:

$$S_j(o) = d_{j,1} + \text{floor}(P\{S_j(e)\}) \quad (5)$$

Margining:

$$S_j = \text{Merge}(S_j(e), S_j(o)) \quad (6)$$

The lifting scheme based DWT has several advantages over the convolution based DWT such as Sweldens (1995, 1996):

- It performs in-place operation calculation. Hence it does not require any extra memory to save its outputs
- All its operations are performed in parallel and that increases the processing speed

- The integer type is lossless transformation and doesn't require extended number of bits per sample for the output coefficients

Integer haar lifting wavelet transform (IHLWT): Haar filter is a one of the famous bank filters that are used with the DWT. In Haar DWT, approximation coefficients are calculated by finding the average of each adjacent samples of the input signal. While the details coefficients are calculated by finding the difference between the adjacent samples of the input signal. Usually, the approximation (smooth) coefficients are close to the original input samples because there are high correlations between the neighbouring samples of the input signal (such as audio signal). Whereas the details signal coefficients has a low power with respect to the original one, because the same reason mentioned above.

In integer to integer lifting scheme based Harr DWT, an input samples S_j has integer values, therefore $P \{S_j(e)\} = S_j(e)$.

Since, $S_j(e)$ is an integer, hence $\text{floor} \{S_j(e)\} = S_j(e)$. As a result, the prediction of odd can be calculated easily to find the details coefficients as given in Eq. 7.

IHLWT predict:

$$d_{j,1} = S_j(o) - S_j(e) \quad (7)$$

The output of the update stage can be calculated based on the value of $d_{j,1}$. Since smooth coefficients of the Harr filter are calculated by finding the average of the input adjacent samples, therefore $S_{j,1} = \text{floor} \{ (S_j(e) + S_j(o))/2 \} = \text{floor} \{ (S_j(e) + S_j(e) + d_j) / 2 \}$. The final update equation can be written as in the Eq. 8.

Update:

$$S_{j,1} = S_j(e) + \text{floor}(d_j/2) \quad (8)$$

To reconstruct the original signal from the details and smooth coefficients, the inverse of predict and update processes are achieved by undoing the update and prediction as shown in Eq. 9 and 10, then merge the results of the two operations together.

Inverse-update:

$$S_j(e) = S_{j,1} - \text{floor}(d_j/2) \quad (9)$$

Inverse-prediction:

$$S_j(o) = d_{j,1} + S_j(e) \quad (10)$$

Proposed architecture of the dual mode IHLWT core: The general block diagram of the proposed IHLWT architecture is shown in Fig. 3a. The design involves of

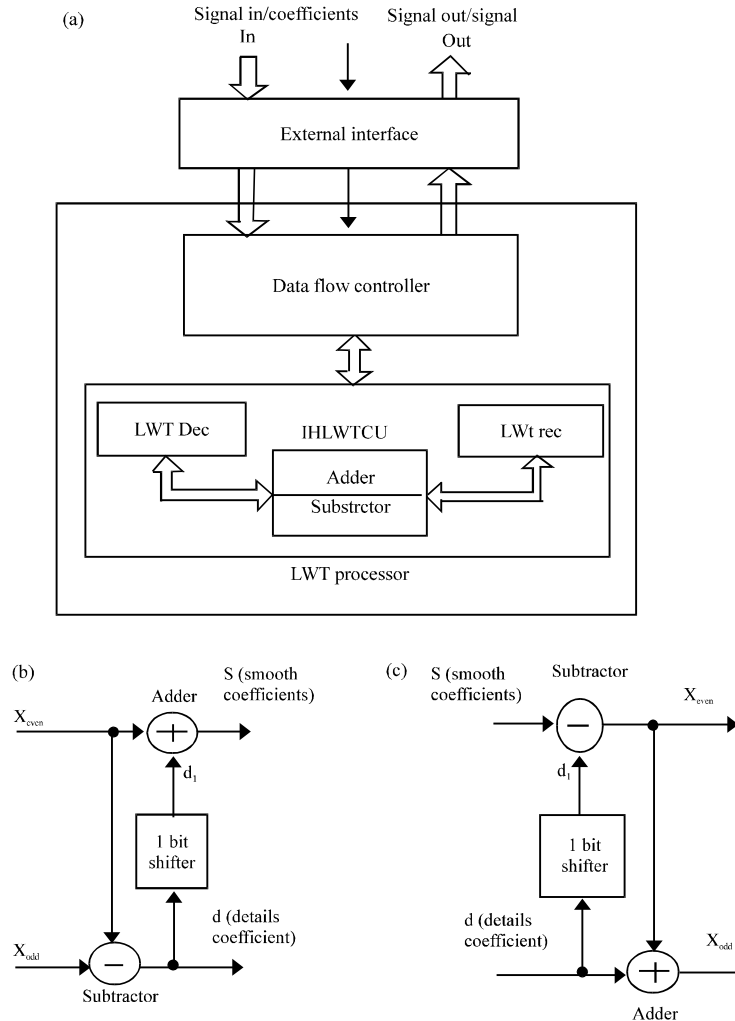


Fig. 3(a-c): General architecture of the proposed design. (a) General block diagram of the proposed integer haar lifting wavelet transform (IHLWT) processor, (b) Hardware architecture of the IHLWT reconstruction mode and (c) Hardware architecture of the IHLWT decomposition mode

two major parts. The external interface to communicate with the host computer is utilized to read the input signal and to write the IHLWT core outputs back. In this study, FPGA- In-Loop (FIL) tool has been used for interfacing. This tool provides an interface between device on chip and Matlab simulator over an Ethernet cable in real time manner. The second main part is the reconfigurable Integer Haar based LWT Computing Unit (IHLWTCU). The IHLWTCU part is the core of the IHLWT processor, this part provide two utilities, each one performs two functions in parallel. The utility selection of the IHLWTCU based on the input mode control signal. If the input mode signal is low (forward mode), then the IHLWTCU performs predict and update operations in parallel. Otherwise, if mode signal is high (inverse mode),

the IHLWTCU performs the inverse of predicate and update operations in parallel.

The IHLWTCU processes a pair of input samples in each clock cycle to carry out IHLWTCU forward or inverse coefficients. For the forward mode, IHLWTCU reconfigures its architecture as shown in Fig. 3b to compute detail and smooth coefficients according to the Eq. 7 and 8, respectively. While for the inverse mode, IHLWTCU reconfigures its architecture as shown in Fig. 3c to compute odd and even samples according to the Eq. 9 and 10, respectively. The mode signal is used as a control signal to specify the IHLWTCU operation mode. Two multiplexers have been used in IHLWTCU to control on the mode configuration (forward or inverse) as shown in Fig. 4a. The complete IHLWTCU architecture shown in

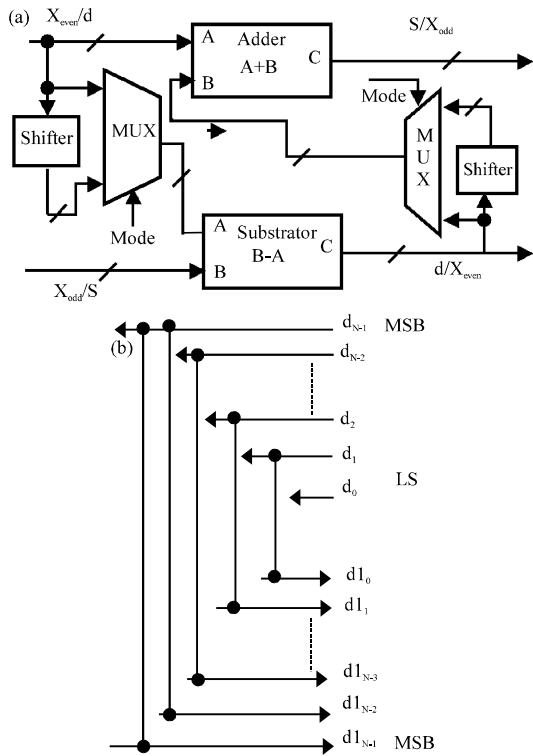


Fig. 4(a-b): Hardware architecture of the IHLWTCU, (a) Hardware block diagram of the integer haar lifting wavelet transform computing unit (IHLWTCU) and (b) Proposed shifter

Fig. 4a has a one adder, one subtractor, two multiplexer and two shifters.

The proposed design of the IHLWTCU is very simple and efficient with low hardware requirements. Both IHLWTCU modes share the same adder and subtractor. Additional important feature of the proposed IHLWTCU is it uses a simple shifter instead of a multiplication and floor operations. This is possible because Eq. 8 and 9 require multiplication by 1/2 and then cancel the floating reminder. The shifter is implemented by shifting the data by a 1 bit toward the Least Significant Bit (LSB). In VHDL the proposed shifter can be represented by only simple signals assignment which look like the wiring circuit that shown in Fig. 4b. In this shifter, all bits of the input integers are shifted toward LSB and keep the sign of an output integer (the Most Significant Bit (MSB)) same as of the input integer. The following two numerical examples for an 8 bits/sample data describe the shifter operation for positive and negative inputs.

Example 1: Let $d = 13$, Then $d1 = \text{floor}(13/2) = 6$.

Now by using the proposed method, since $d = (00001101)_2$ then

	MSB							LSB
Bit No.	7	6	5	4	3	2	1	0
d	0	0	0	0	1	1	0	1
d1	0	0	0	0	0	1	1	0

$d1 = (00000110)_2 = (6)_{10}$

Example 2: Let $d = -13$, Then $d1 = \text{floor}(-13/2) = -7$

Now by using the proposed method, since $d = (11110011)_2$ (2's complement of 13) then

	MSB							LSB
Bit No.	7	6	5	4	3	2	1	0
d	1	1	1	1	0	0	1	1
d1	1	1	1	1	1	0	0	1

$d1 = (11111001)_2 = (-7)_{10}$

As shown in the above two examples, the two mathematical operations (multiplication and floor) have been implemented in a single simple operation. This design does not require any multiplier or memory to store the intermediate results. It has high speed (very low path delay) and it uses only one adder and one subtractor which have reconfigurable inputs based on the mode of the input signal.

Table 1 demonstrates the data flow through the nodes of the IHLWTCU of the forward mode for a 16 samples input signal. The word "Process" in the table refers to IHLWTCU process (predict and update) and the $X_0, X_1, X_2, X_3, \dots$ are the input signal samples values. In the table, there is no IHLWTCU operation throughout the first clock cycles because of enable signal is low. During the second period of the clock cycle the IHLWTCU performs the first predict and update operations with inputs from the nodes X_{odd} and X_{even} . At the beginning of third clock cycle, the value of the outputs replaced by the first detail and smooth coefficients. The total time required to process a 16 samples is 9 clock cycles. Only one clock cycle delay is needed between whole input samples and whole output coefficients. Therefore, to process N samples of input signal, IHLWTCU requires only $N/2+1$ clock cycles.

Table 2 shows the data flow through the nodes of the IHLWTCU of the inverse mode for a 16 input coefficients. The data flow in IHLWTCU nodes are same as in Table 1 except that the input data is the coefficients and the output data is the reconstructed signal as well as the IHLWTCU processes are the inverse predict and inverse update.

Table 1: Data flow of the proposed integer haar lifting wavelet core for a signal of 16 samples in forward mode (mode signal = logic 0)

Process	Nop	P and U	P and U	P and U	P and U	P and U	P and U	P and U	P and U	P and U	P and U	Nop
Enb	0	1	1	1	1	1	1	1	1	1	1	0
Clk	1	2	3	4	5	6	7	8	9	10	11	11
Xeven	0	X0	X2	X4	X6	X8	X10	X12	X14	0	0	0
Xodd	0	X1	X3	X5	X7	X9	X11	X13	X15	0	0	0
d		d0	d1	d2	d3	d4	d5	d6	d7			0
S		S0	S1	S2	S3	S4	S5	S6	S7			0

Nop: No process, P: Predict, U: Update, enb: enable, X_{even}: Input even sample, X_{odd}: Input odd sample, S: Smooth coefficient, d: Detail coefficient

Table 2: Data flow of the proposed integer Haar lifting wavelet core for 16 Coefficients (8 smooth and 8 detail coefficients) in Inverse Mode (Mode Signal = Logic 1)

Process	Nop	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	IP and IU	Nop
Enb	0	1	1	1	1	1	1	1	1	1	1	0
Clk	1	2	3	4	5	6	7	8	9	10	11	11
d	0	d0	d1	d2	d3	d4	d5	d6	d7	0	0	0
S	0	S0	S1	S2	S3	S4	S5	S6	S7	0	0	0
X _{even}			X0	X2	X4	X6	X8	X10	X12	X14	0	0
X _{odd}			X1	X3	X5	X7	X9	X11	X13	X15	0	0

Nop: No process, IP: Inverse predict, IU: Inverse update, enb: Enab signal xodd -input odd sample, X_{even}: Input even sample, S: Smooth coefficients, d: Detail coefficient

RESULTS

The proposed dual mode IHLWTCU has been implemented using IEEE standard VHDL. This allows portable across FPGA boards from numerous vendors. Xilinx ISE has been used to simulate, synthesize and implement the proposed hardware. This section describes these steps.

Simulation: In order to validate the operations of the proposed architecture, the test bench simulation has been used. All the results of the test-bench simulation are compared with the Matlab program for the dual mode IHLWTCU to verify the results. Figure 5 shows a sample of the tests. In the first part of the clock cycles of Fig. 5, the mode control signal is low; therefore, IHLWT core performs forward transformation to carry out detail and smooth coefficients onto their outputs buses Y1 and Y2 respectively. While in the second part of the clock cycles, the mode control signal is high, therefore the IHLWT core performs inverse transformation to carry out reconstructed even and odd samples onto their outputs buses Y1 and Y2, respectively.

Synthesis: The proposed architecture has been successfully synthesized for various Xilinx FPGA devices. Table 3 shows the results of five different families Xilinx FPGA devices. The required hardware area is very little, only 16 or 8 of Configurable Logic Blocks (CLB) slices are required for IHLWTCU based on the used FPGA board. Furthermore, the proposed design has very high maximum operating frequency. For all the FPGA devices in Table 3, maximum operating frequency is higher than the system clocks of the FPGA device. This feature (high maximum frequency) will allow the design to be built in hardware applications which need very high speed DWT processor.

Table 3: Various xilinx board synthesis results for the proposed IHLWTCU architecture

FPGA device	No. of CLB slices	Maximum operating frequency (Mz)
Spartan3E-XC3S500E	16	626.56
Spartan3A-XC3S700a	16	674.76
Spartan6-XC6SLX16	8	375.094
Virtex-XC4VFX12	16	734.214
Virtex5- XC5VLX110T	8	667.55

Power consumption analysis: Power consumption is a vital design factor besides maximum operating frequency and hardware area. In addition to the low CLBs slices number and high operating frequency, the proposed design has very low power consumption. Table 4 shows the summary of the power consumption analysis for the proposed architecture for number of FPGA devices in different operating frequencies. The quiescent power for each FPGA device represents the dominated part of the total consumption power. It is for the whole device including idle FPGA resources. Consequently, the total power consumed by the IHLWTCU core alone is lower than those reported in Table 4. This makes the proposed design very suitable for using in low power applications such as mobile and wireless multimedia sensor networks.

FPGA implementation: The proposed design has been successfully implemented onto Spartan6-SP601 Evaluation Platform Board. The implementation tests are achieved by employing the FIL tool which is used to provide an interface between PC and on chip device through Ethernet cable. This tool is a new utility in MATLAB 2012 and earlier, it uses the FPGA vendor synthesis tools to synthesize the VHDL codes and download the bit stream files onto the FPGA board. The device on chip can be hosted by PC through MATLAB simulation model in order to read and write from and to

Table 4: Power analysis of the proposed architecture for various xilinx FPGA

On chip	Power (mW)				
	Spartan3E-XC3S500E@50 MHz	Spartan3A-XC3S700a@50 MHz	Spartan6-XC6SLX16@100 MHz	Virtex4- XC4VFX12@100 MHz	Virtex5- XC5VLX110T@100 MHz
Clocks	1.84	3.64	2.18	7.07	5.71
Logic	0.3	0.18	0.33	0.56	0.24
Signals	0.5	0.0	1.26	1.21	0.52
I/Os	56.89	0.19	60.8	0.56	60.8
Total dynamic	60	4	65	9	67
Quiescent	82	32	21	165	1188
Total power	141	36	85	174	1255

Table 5: Comparison between the proposed and some of the literatures for one dimension discrete wavelet transform (DWT) architectures

DWT architecture	Filter type	DWT design process	Hardware requirements	No. of cycle to decompose N sample	Control complexity	Critical path delay
Proposed (Pang and Chauhan, 2008)	Integer haar	F and I	1 A, 1S, 2 Mux	1+N/2	Simple	Ta+Ts
(Altermann <i>et al.</i> , 2011)	Daub-4	F	4 Mul, 3 A, 9 R	7 + N	Complex	Tm+2 Ta
(Liao <i>et al.</i> , 2004)	Haar	F	2 Mul, 2 A, 2R, Separate controller	2 + N	Moderate	2 Tm
(Huang <i>et al.</i> , 2004)	Daub-4	F	4 Mul,2 scaling Mul,4 A,7 R,3 D,6 Mux	3 + N	Complex	4Tm+8Ta
(Andra <i>et al.</i> , 2002)	9/7	F	4 Mul,2 scaling Mul,8 A,10 R,6 S	5 + N	Moderate	Tm
(Aziz and Pham, 2012)	5/3	F or I	2 processors(each with 1 Mul,2 A,2 R), 2 scaling Mul	12 + N	Moderate	Tm
	5/3	F	2 A, 3 R, 1 SR, 5 Mux	2+N	Simple	2Ta

Daub: Daubechies, F: Forward, I: Inverse, Mul: Multiplier, A: Adder, S: Subtractor, R: Register, Mux: Multiplexer, Ta: Adder time delay, Ts: Subtractor Time Delay ~Ta, Tm: Multiplier time delay

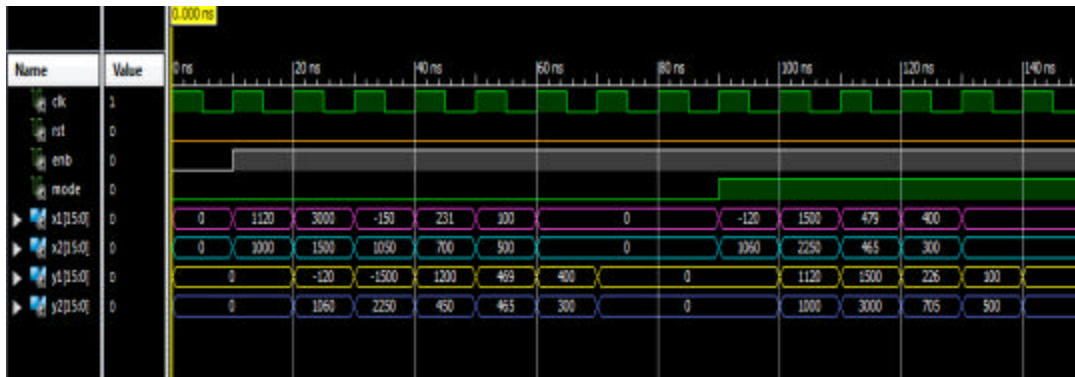


Fig. 5: Test bench simulation for the proposed IHLWT processor

device, respectively. Figure 6 shows the model that was used in the tests. The model consists a one bit counter free running which used as external clock. The input samples (X1 and X2) are either entered from row selector which forward two samples each clock cycle, or from workspace memory of IHLWT coefficients depending the operating mode. The selection of the input buses has been achieved by manual switches as shown in Fig. 6. The core on chip has been tested by using audio signals as input in both real time recording and signal from Matlab work-space. Various sampling frequencies of audio signal are used in tests (8, 16 and 44.1 kHz). Figure 7 shows a sample of the tests (5 sec audio signal

with 8 kHz sampling frequency) in forward mode. All results were identical 100% with the Matlab software program of the IHLWT Core.

Comparison with some of the related works: Table 5 shows a comparison between the proposed design and some of that of the literatures. In the table, comparisons for the most important factors in the hardware design such as the required components for the processor, processing speed, DWT design mode, control complexity and critical path delay. The table shows superior of the proposed design over the others in most features.

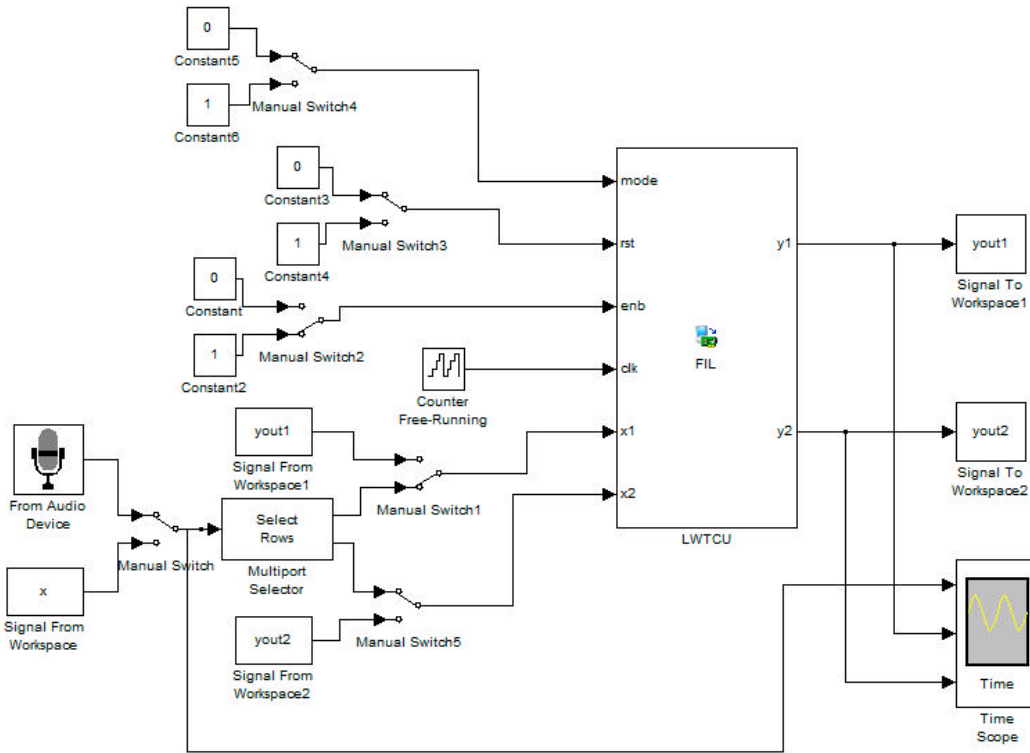


Fig. 6: Implementation model of the proposed IHLWT core on SP601 FPGA board via real time FIL tool

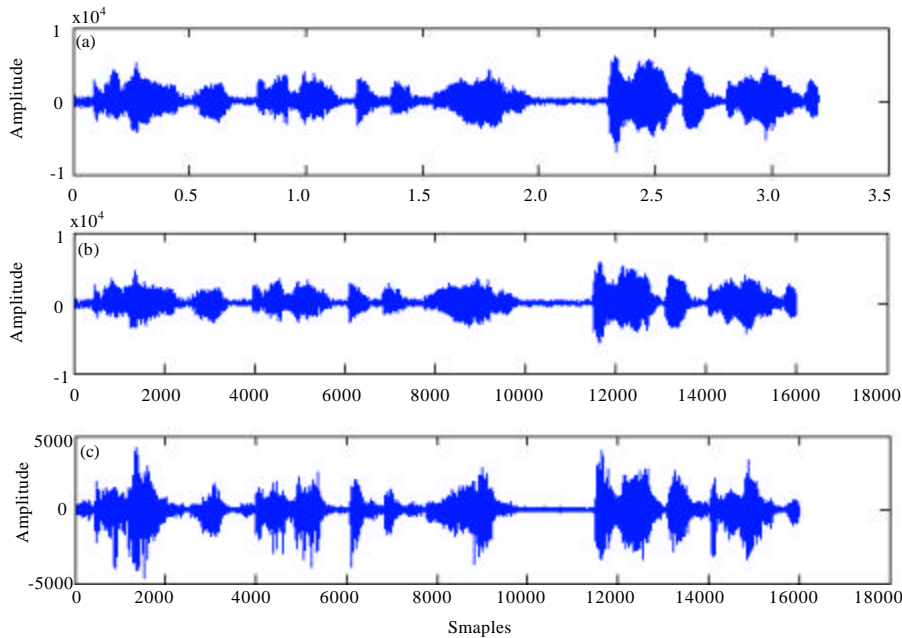


Fig. 7(a-c): Sample test from the experimental results of DWT audio decomposition on FPGA board, (a) Original signal, (b) Smooth coefficients and (c) Details coefficients

CONCLUSION

Low hardware area and high speed IHLWT architecture has been presented. Very simple circuit,

requiring only one adder and subtractor with two simple proposed shifters, is used for both forward and inverse transformations. The proposed shifter circuit is only represented by a simple wiring circuit. It is used to

compute floor and multiplication operation in a one simple step. The overall design is very simple and generic. It can be effortlessly configured to process one dimension signal for various signal lengths. The proposed design has been successfully synthesized for various Xilinx FPGA devices. The results of synthesis show that the design has very high operating frequency and low number of CLB slices for various FPGA devices. Furthermore, the power analysis's of the proposed architecture show that the design has very low power consumption for different Xilinx devices. The IHLWT core has been successfully implemented on a Xilinx Spartan6-SP601 Evaluation Platform Board. Many real time audio recording tests have been performed for the on-chip device to process in both forward and inverse modes. All the results obtained have been identical when compared to the Matlab software results.

REFERENCES

- Altermann, J., E. Costa and S. Almeida, 2011. High performance Haar wavelet transform architecture. Proceedings of the 20th European Conference on Circuit Theory and Design, August 29-31, 2011, Linkoping, Sweden, pp: 596-599.
- Andra, K., C. Chakrabarti and T. Acharya, 2002. A VLSI architecture for lifting-based forward and inverse wavelet transform. *IEEE Trans. Signal Process.*, 50: 966-977.
- Angelopoulou, M.E., K. Masselos, P.Y.K. Cheung and Y. Andreopoulos, 2008. Implementation and comparison of the 5/3 lifting 2D discrete wavelet transform computation schedules on FPGAs. *J. Signal Process. Syst.*, 51: 3-21.
- Aziz, S.M. and D.M. Pham, 2012. Efficient parallel architecture for multi-level forward discrete wavelet transform processors. *Comput. Electr. Eng.*, 38: 1325-1335.
- Chen, P.Y., 2004. VLSI implementation for one-dimensional multilevel lifting-based wavelet transform. *IEEE Trans. Comput.*, 53: 386-398.
- Farahani, M.A. and M. Eshghi, 2007. Implementing a new architecture of wavelet packet transform on FPGA transform with tree structure. Proceedings of the 8th WSEAS Conference on Acoustics and Music: Theory and Applications, June 18-20, 2007, Vancouver, Canada, pp: 37-41.
- Huang, C.T., P.C. Tseng and L.G. Chen, 2004. Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Trans. Signal Process.*, 52: 1080-1089.
- Huang, C.T., P.C. Tseng and L.G. Chen, 2005a. Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform. *IEEE Trans. Signal Process.*, 53: 1575-1586.
- Huang, C.T., P.C. Tseng and L.G. Chen, 2005b. VLSI architecture for forward discrete wavelet transform based on B-spline factorization. *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, 40: 343-353.
- Kuzume, K., K. Nijjima and S. Takano, 2004. FPGA-based lifting wavelet processor for real-time signal detection. *Signal Process.*, 84: 1931-1940.
- Liao, H., M.K. Mandal and B.F. Cockburn, 2004. Efficient architectures for 1-D and 2-D lifting-based wavelet transforms. *IEEE Trans. Signal Process.*, 52: 1315-1326.
- Maamoun, M. and A. Namane, 2009. VLSI design for high-speed image computing using fast convolution-based discrete wavelet transform. Proceedings of the World Congress on Engineering World, Volume 1, July 1-3, 2009, London, UK., pp: 817-821.
- Mallat, S.G., 1989. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11: 674-693.
- Pang, J. and S. Chauhan, 2008. FPGA design of speech compression by using discrete wavelet transform. Proceedings of the World Congress on Engineering and Computer Science, October 22-24, 2008, San Francisco, USA., pp: 1-6.
- Shi, G., W. Liu, L. Zhang and F. Li, 2009. An efficient folded architecture for lifting-based discrete wavelet transform. *IEEE Trans. Circuits Syst. II: Express Briefs*, 56: 290-294.
- Sweldens, W., 1995. Lifting scheme: A new philosophy in biorthogonal wavelet constructions. Proceedings of SPIE Wavelet Applications in Signal and Image Processing III, Volume 2569, July 12-14, 1995, San Diego, CA., USA., pp: 68-79.
- Sweldens, W., 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied Comput. Harmonic Anal.*, 3: 186-200.
- Zhang, W., Z. Jiang, Z. Gao and Y. Liu, 2012. An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Trans. Circuits Syst. II: Express Briefs*, 59: 158-162.