

REAL TIME VLSI IWT DENOSING ARCHITECTURE

ALI NABI SALEH* and AHMED K. AL-SULAIFANIE**

*Directorate of Media Press and Publishing, Duhok, Kurdistan Region-Iraq

**Dept. of Electrical & Computer Engineering, University of Duhok, Kurdistan Region-Iraq

ABSTRACT

In the last decade many efficient VLSI architectures were designed to implement the discrete wavelet transform using the lifting scheme. Most of these architectures do not operate in real-time. This paper presents a scalable real time VLSI architectural to compute an integer wavelet transform (IWT) using the lifting scheme for (5/3) biorthogonal filter. The proposed architecture is projected on (XC3S700A FG484) FPGA chip embedded on a Spartan 3A starter kit board. An efficient formula is driven for controlling the delay introduced between tandem architectures that are adapted to work in real-time. The layout of the integrated VLSI structure is simple and can be connected easily in tandem for computing IWT in real time.

A hard threshold module is designed and attached with proposed architecture to implement real time denosing application. The achievement of the proposed architecture along with supplement hard threshold module is assessed by denosing a four benchmarks signals corrupted by adding white Gaussian noise. The result outcomes show that the proposed IWT architecture has powerful performance in the real time wavelet based signal denosing process. The architecture data word length is selected as 11-bits to avoided arithmetic overflow for two's complement 8-bit integer data input. The maximum operating frequency of the proposed architecture varies from 26 MHz in 1-level to 14MHz in 5-level for decomposition/ reconstruction with hard threshold module in an FPGA implementation. The hardware utilization varies from (50%) in 1-level to (97%) for 5-level.

Index Terms—Integer wavelets transform, VLSI architecture, signal denosing, Wavelet thresholding.

1. INTRODUCTION

The wide range of applications of the Discrete Wavelet Transform (DWT) makes the evaluation of its implementation in VLSI extremely important. The recent developments in technology with falling costs, the realization of the (DWT) on field programmable gate array (FPGA) have been widely used. In particular, the DWT based on the lifting scheme has many advantages over conventional DWT (Daubechies and Sweldens, 1996). The survey of lifting based DWT implementations is given in (Acharya and Tsai 2005). The represents of an efficient VLSI architecture for the implementation of 1D multilevel lifting discrete wavelet using two folded and flexible architectures for analysis and synthesis lifting (5, 3) DWT is given (Chen, 2004). The work in (Oweiss et al, 2007) proposes VLSI architectures to compute a 1D DWT for real-time multichannel streaming data under stringent area and power constraints. The implementations are based on the lifting scheme for wavelet computation and integer fixed point

precision arithmetic, which minimize the computational load and memory requirements. In (Al-Sulaifanie et al, 2010), the design of the integer to integer lifting-based wavelet transform (IWT) architecture is presented. The designed architecture can be connected repetitively in tandem to compute multi-level decomposition and reconstructed IWT. The work in (Bahoura and Ezzaidi, 2010) presents a real-time architecture for forward/inverse wavelet transforms that take into account the group delays of the used filters. The main idea is based on the equalization of the filter path delays. In (Andra et al, 2002), a generalized architecture is suggested. The processor architecture consists of adders, multipliers and shifters that are interconnected in a manner that would support the computational structure of the specific filter.

The work in (Liao et al, 2004) introduced a 1D dual scan architecture for DWT that is achieved 100% hardware utilization by processing two independent data streams together. The architecture consists of a processing element that implements the conventional lifting scheme, one

memory unit and input and output switches are used.

In (Aziz and Pham, 2012), a highly parallel architecture to compute up to five levels of 2-D lossless forward DWT has been presented. Instances of a highly optimised single-level processor are used to create the multi-level processor. The work in (Hsia et al, 2015) addresses the architecture problems of a conventional 1-D lifting-based DWT, by addressing the problems of the arithmetic unit, the single mode filter coefficients and the hardware processing time. Recently for real-time applications, efficient VLSI implementation of DWT is introduced by (Rafi and Din, 2017). The architecture is based on lifting one-dimensional Cohen-Daubechies-Feauveau (CDF) (5,3) wavelet filter, which is easily extended to 2-D implementation.

Since Donoho and Johnstone (Donoho and Johnstone, 1995) originally proposed a method known as the wavelet transform shrinkage (thresholding) to estimate an unknown smoothed signal from data with noise, the wavelet transform has rapidly become very popular for signal denoising. The shrinkage rule controls how the thresholds are applied to the signal, commonly two types of thresholding methods are used, the hard threshold and soft threshold (Mallat, 1999). Wavelet denoising method was used widely in many fields, such as industry control and signal processing. In recent years there has been a fair amount of research on wavelet thresholding and threshold selection for signal and image denoising (Bahoura and Ezzaidi, 2010) because wavelet provides an appropriate basis for separating noisy signal from signal.

This paper is organized as follows: In Section 2, the overview of IWT lifting scheme and wavelet threshold denoising is presented. The proposed architecture with hard threshold module described in detail in section 3. Section 4 describes the typical simulation and results. The Design Statistics is given in section 5. Comparison of performance with other lifting based architectures is elaborated in section 6. The conclusion is given in section 7.

2. INTEGER LIFTING SCHEME AND WAVELET THRESHOLD DENOISING

2.1. INTEGER LIFTING SCHEME FOR (5/3) BIORTHOGONAL FILTER

The lifting scheme for (5/3) biorthogonal is composed of three stages: splitting the input signal $x_a[n]$ into even sequence $x_{ae}[n]$ and odd sequence $x_{ao}[n]$. Then the even and odd sequences are processed by the predicting, and updating steps, resulting in the high-pass and the low-pass wavelet coefficients $a(k)$ and $d(k)$ are given as (Daubechies and Sweldens, 1996)

$$d[n] = x_{ao}[n-1] - \frac{1}{2}(x_{ae}[n] + x_{ae}[n-1]) \quad (1)$$

$$a[n] = x_{ae}[n] + \frac{1}{4}(d[n] + d[n+1]) \quad (2)$$

Where $n = 0, 1, 2, \dots, N/2$ and N is the length of input signal.

The lifting scheme backward transform (synthesis process) is simply realized by reversing the order of the forward lifting steps to reconstruct the even sequence $x_{se}[n]$ and odd sequence $x_{so}[n]$ according to following equations:

$$x_{se}[n] = a[n] - \frac{1}{4}(d[n] + d[n+1]) \quad (3)$$

$$x_{so}[n-1] = d[n] + \frac{1}{2}(x_{se}[n] + x_{se}[n-1]) \quad (4)$$

2.2. WAVELET THRESHOLD DENOISING

The thresholding process on wavelet coefficients were used for removal of additive noise from deterministic signals (Donoho et al, 1995). The disparity between signal and noise is achieved by discarding high frequency components below a predetermined threshold. The DWT analysis of a noisy signal is performed through a series of multirate low-pass and high-pass filters. This filter operation decomposes the signal into approximate and detail information so allocate in analyzing it at different frequency bands with different resolutions. Assume the following mathematical model of noisy signal

$$y_i[k] = x_i[k] + w_i[k] \text{ where } k = 1, 2, 3 \dots N \quad (5)$$

Where $w_i[k]$ is an Additive Gaussian White Noise (AWGN) of variance (σ^2).

To recover original signal $x_i[k]$ from noisy signal $y_i[k]$, the wavelet denoising by the

thresholding procedure is processed in three steps (Donoho et al, 1995) (Hamid et al, 1999) :

1. Decomposition of the input noisy signal $y_i[k]$ using DWT into several levels from 1 to J .
2. Thresholding of DWT coefficients levels 1 to J .
3. Reconstruction denoised signal $y_o[k]$ through inverse DWT on the founded DWT coefficients.

3. PROPOSED IWT DENOISING ARCHITECTURE

The design of a VLSI wavelet denoising architecture that has capability of performing a real time denoising based on integer wavelet

transform (IWT) is introduced in this section. In order to intend such structural design, it is required to propose forward (FIWT) and backward (BIWT) architectures with hard threshold module, and controlling the delay introduced between tandem architectures.

3.1 IWT ARCHITECTURE

Real time processing operation can be achieved by introducing traversable delay to make the whole scheme causal in order that lifting scheme work in real time applications. Adding delay to the forward and backward lifting equations (1), (2), (3), and (4), the final time-domain equations for a real time IWT will be:

$$d[n] = x_{ao}[n-1] - \left\lfloor \frac{1}{2}(x_{ae}[n] + x_{ae}[n-1]) \right\rfloor \quad (6)$$

$$a[n-1] = x_{ae}[n-1] + \left\lfloor \frac{1}{4}(d[n-1] + d[n]) \right\rfloor \quad (7)$$

$$x_{se}[n-1] = a[n-1] - \left\lfloor \frac{1}{4}(d[n-1] + d[n]) \right\rfloor \quad (8)$$

$$x_{so}[n-2] = d[n-1] + \left\lfloor \frac{1}{2}(x_{se}[n-1] + x_{se}[n-2]) \right\rfloor \quad (9)$$

Where the operator $\lfloor Y \rfloor$ is floor function that outputs the largest integer not greater than Y .

Figure (1) shows the direct mapping of above equations into real time IWT lifting schematic. In this work, a two's complement signed integers are used in all fixed-point arithmetic's. Forward IWT and backward IWT contain multiplication by $(1/2)$ and multiplication by $(1/4)$. The floor operations $\lfloor \cdot \rfloor$ can be implemented using arithmetic shift right by 1-bit or by 2-bit instead of multiply by $(1/2)$ or by $(1/4)$.

The bit-depth extension is defined as the number of additional bits requested and known as the base 2 logarithm of the Bounded Input

Bounded Output (BIBO) gain. It has been shown in (David et al, 2002) that for $(5/3)$ biorthogonal filter the worst-case bit-depth expansion is 2 bits for up to five levels of decompositions. As well in the situation of adding or subtracting two input numbers in 2's complement representation there is the possibility that overflow resulted. To avoid overflow an extra bit is required to cover all possible outputs. Therefore, the word length of elements used in lifting based IWT is sustained at $(b+3)$ bits. Where b is word length of the signal input data. Therefore, using two's complement representations, the word length of the architecture is 11-bits for integer input of 8-bit per sample.

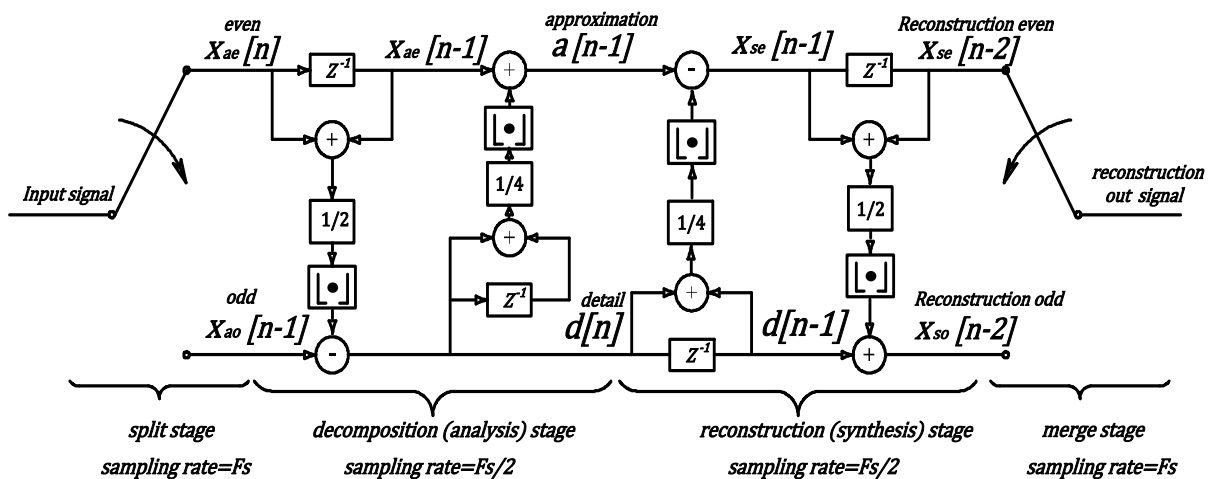


Fig. (1): Real-time IWT lifting schematic of one level ($j = 1$) forward architecture (FIWT) and backward architecture (BIWT), each architecture has one unit delay

3.2 HARD THRESHOLD MODULE

The 2’s complement hard threshold module used for the denoising process on detail coefficients is shown in figure (2). An unsigned binary comparator is used to compare the detail sample value with the threshold. For unsigned comparator the output LT (Less Than) will be “HIGH” if the A-input is less than the B-input. The inequality between unsigned comparator signed 2’s complements comparator can be

balanced as presumed in table (1). It is obvious from table (1) that the signed 2’s complements “less than decision” will be XOR operation output between the MSB of $d[n]$ input sample and the LT output for unsigned binary comparator. The XOR output will control the input selection of final stage MUX. The output of the MUX will be:

$$dt_j[n] = \begin{cases} d_j[n] & \text{if } |d_j[n]| > TH \\ 0 & \text{if } |d_j[n]| \leq TH \end{cases} \quad (10)$$

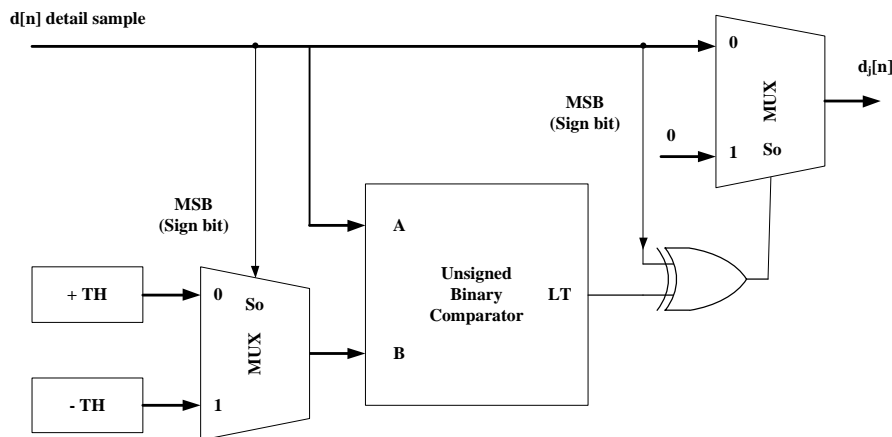


Fig. (2): The 2’s complement hard threshold module.

Table (1): Hard threshold table to convert binary comparator to a 2’s complement comparator

| MSB Sign bit of $d[n]$ | Compare $d[n]$ to TH | LT | XOR output | $dt_j[n]$ output |
|------------------------|----------------------|----|------------|------------------|
| 0 | $d_j[n] > TH$ | 0 | 0 | $d_j[n]$ |
| 1 | $-d_j[n] \geq -TH$ | 0 | 1 | 0 |
| 0 | $d_j[n] \leq TH$ | 1 | 1 | 0 |
| 1 | $-d_j[n] < -TH$ | 1 | 0 | $d_j[n]$ |

3.3 DELAY ALIGNMENT

The real-time IWT schematic of figure (1) can be considered as one level ($j = 1$) forward architecture and backward architecture connected in tandem. Considering that the sampling period of the input signal $x_a[n]$ is T_s . Then due to multirate sampling rate in dyadic manner [14], each of the forward or backward architecture will introduced one delay unit U_1 at sampling frequency $F_s/2$, hence delay unit $U_1 = 2T_s$. Consequently the total time delay between the input signal and reconstructed signal for one level forward / backward is $2U_1 = 4T_s$. In general the delay time introduced by any forward architecture or backward architecture at level (j) is:

$$U_j = 2^j T_s \quad (11)$$

For J levels of analysis and synthesis, at any synthesis stage j , the reconstructed approximation ar_j and detail d_j data streams are merged to produce output ar_{j-1} as shown in figure (3). The delay will be introduced between the reconstructed output ar_j and corresponding detail d_j data streams. Accordingly, these delays should be aligned. The time delay for J levels of decomposition introduced between the outputs

detail d_j at level j and a_j of the last level J can be defined as $TD(j, J)$. From figure (3) The time delay $TD(j, J)$ is the sum of delays introduced by upper levels $j + 1$ to J and can be written as:

$$TD(j, J) = \sum_{k=j+1}^J U_k = \sum_{k=j+1}^J 2^k T_s = T_s (2^{j+1} + 2^{j+2} + \dots + 2^J) \quad (12)$$

The sum for the geometric series in (12) is $TD(j, J) = T_s (2^{J+1} - 2^{j+1}) = 2^{j+1} T_s (2^{J-j} - 1)$. For reconstruction or backward architectures from level J to level j , the same time delay will be introduced since each forward or backward has the same delay. Therefore, the delay alignment between the reconstructed output ar_j and corresponding detail d_j is:

$$\begin{aligned} A_j &= 2TD(j, J) = 2 \times 2^{j+1} T_s (2^{J-j} - 1) \\ &= 4 \times 2^j T_s (2^{J-j} - 1) \\ &= 4(2^{J-j} - 1)U_j \end{aligned} \quad (13)$$

The normalized delay units in each level j inside J tandem IWT architectures can be defined as:

$$D_j = \frac{A_j}{U_j} = 4(2^{J-j} - 1) \quad (14)$$

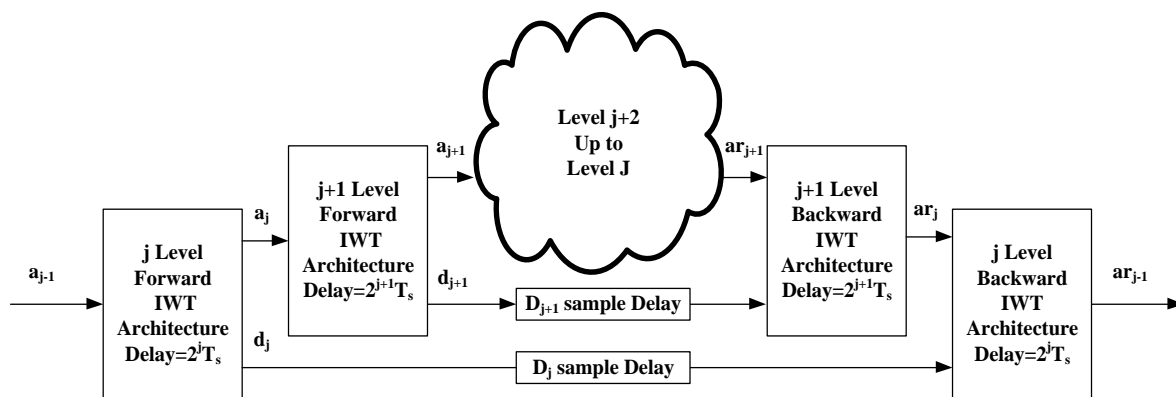


Fig. (3): Delay time alignment in the tandem IWT architectures.

A block diagram for five-level IWT ($J=5$) shown in figure (4). Table 2 shows the normalized including external normalized unit delays is delay units for five levels ($J=5$).

Table (2): shows the normalized delay units for five levels ($J=5$).

| j | 0 | 1 | 2 | 3 | 4 | 5 |
|-------|-----|----|----|----|---|---|
| D_j | 124 | 60 | 28 | 12 | 4 | 0 |

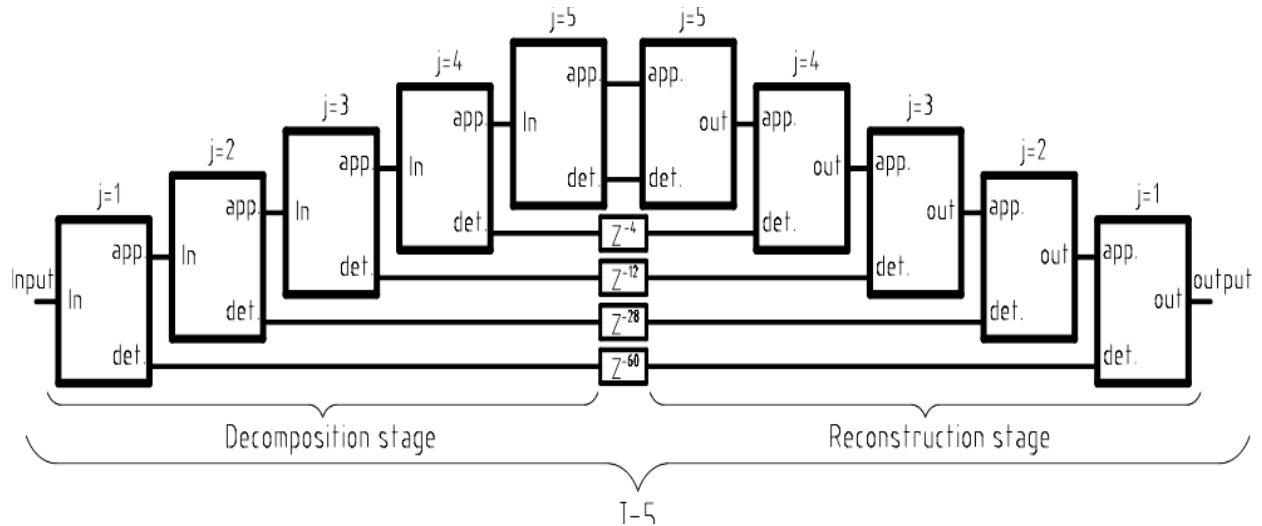


Fig. (4): External unit delays between forward output and reverse input detail, the terms (app) and (det) stands for approximation and detail outputs respectively.

4. EXPERIMENTAL RESULTS

In order to verify the performance of the proposed IWT Denoising Architecture, a simulation model is build using matlabR2011a. A generated Gaussian white noise is added to well-known benchmark signals and hard threshold denoising method is applied to remove the noise.

4.1. Threshold Estimation

The performance of signal denoising in wavelet domain depends on the evaluation of the threshold value. Several approaches have been considered for estimating threshold value. Among the existing methods the universal threshold is most popular one. It has been proved that the maximum amplitude of the noise has a very high probability of being just below universal threshold TH with orthonormal basis. The universal threshold is given by (Donoho and Johnstone, 1995) (Hamid et al, 1999)

$$TH = \sigma \sqrt{2 \log_e(N)} \quad (15)$$

Where σ the standard deviation of the noise and N is the signal length. The universal threshold may be excessively large as its dependence on the number of samples. Therefore, a level-dependent threshold is assumed, which are more adaptive to the noise and signal features

$$TH_j = \sigma_j \sqrt{2 \log_e(N_j)} \quad (16)$$

Where σ_j is the standard deviation of the wavelet coefficients, estimated on level j. The N_j is the length of data at level j.

4.2. Measurement Criteria

The performance of wavelet denoising scheme is mainly decided by the similarity and smoothness between original signal and the denoised signal. The measurement criteria Signal to Noise Ratio SNR and mean square error (MSE) parameters are used to evaluate the performance of wavelet denoising scheme which defined as (Alfred, 1999)

(1) The input MSE_i can be denoted by formula (17):

$$MSE_i = \frac{1}{N} \sum_{k=0}^{N-1} (y_i[k] - x_i[k])^2 \quad (17)$$

where $x_i[k]$ and $y_o[k]$ signals are given in mathematical model (5). Hence, the variance $\sigma_{w_i}^2$ of the AWGN input $w_i[k]$ is $\sigma_{w_i}^2 = MSE_i$.

(2) The input signal-to-noise ratio (SNR_i):

$$SNR_i = 10 \log_{10} \left(\frac{\sigma_{x_i}^2}{\sigma_{w_i}^2} \right) \quad dB \quad (18)$$

Where $\sigma_{x_i}^2$ is the variance or AC power of the original input signal $x_i[k]$.

(3) The output signal-to-noise ratio (SNR_o):

$$SNR_o = 10 \log_{10} \left(\frac{\sigma_{x_i}^2}{MSE_o} \right) \quad dB \quad (19)$$

Where, $MSE_o = \frac{1}{N} \sum_{k=0}^{N-1} (x_i[k] - y_o[k])^2$. Where N, is the length of input the signal.

4.3. Simulation Setting And Results

A simulation test is done to inspect the performance of the proposed IWT architecture to different signals and to various amounts of noise corruption. For the determination of performance evaluation, the simulation includes four known benchmark signals Block, Bumps, Heavy sine, and Doppler with length N equal to 2048 samples. The SNR_I of the selected noisy signals are 5 dB, 10 dB, 15 dB and 20 dB. The number of wavelet decomposition levels J is set to 5. The hard thresholding denoising method is used.

Table (3) shows the performance compare results between the double precision and integer 11-bit fixed point IWT architectures. For the fixed point the input sample is represented by 8-bit signed two's complement integer numbers. The SNR_I and SNR_O along with mean-square-error

mse accuracy are measured for both cases double precision and fixed point. It is observed that accuracy performance between the double precision and integer 11-bit fixed point IWT architectures are slightly difference as shown in Figure 5.

The fixed point the four benchmark original input signal $x_i[k]$, beside the noisy input signal $y_i[k]$ with SNR_I at 5 dB and 20 dB as well as their denoised output signal $y_o[k]$ are shown in Figures 6 and 7 respectively. It can be observed that a valuable amount of noise has been removed. At 5 dB cases a massive noise is suppressed and the outlook frame of the benchmark signals is recovered. In light 20 dB cases the detail of benchmark signals are entirely recovered with only slightly harmed since hard thresholding method are coarse and rough (Mallat, 1999).

Table (3): Performance evaluation of benchmark signals hard thresholding denoising with double and integer precision (signal length = 2048 samples).

| benchmark signals | Noise input parameters | | double precision | | 11-bit fixed point | |
|-------------------|------------------------|----------|------------------|---------|--------------------|---------|
| | SNR_I dB | MSE_i | SNR_o dB | MSE_o | SNR_o dB | MSE_o |
| Blocks | 5 | 100.2239 | 13.3756 | 15.0812 | 13.7594 | 13.4442 |
| | 10 | 31.6936 | 17.2740 | 6.1460 | 17.8470 | 5.2454 |
| | 15 | 10.0224 | 23.4396 | 1.4861 | 23.7387 | 1.3509 |
| | 20 | 3.1694 | 29.3637 | 0.3799 | 25.8416 | 0.8363 |
| Bumps | 5 | 107.1214 | 13.5798 | 14.5426 | 13.4941 | 14.6313 |
| | 10 | 33.8748 | 18.1829 | 5.0389 | 17.9837 | 5.2038 |
| | 15 | 10.7121 | 23.2565 | 1.5667 | 22.9999 | 1.6395 |
| | 20 | 3.3875 | 26.9030 | 0.6766 | 25.4460 | 0.9335 |
| Heavy sine | 5 | 729.3864 | 18.9779 | 28.4857 | 18.9531 | 28.1131 |
| | 10 | 230.6522 | 23.3179 | 10.4864 | 23.0405 | 10.9690 |
| | 15 | 72.9386 | 27.6311 | 3.8843 | 27.2467 | 4.1644 |
| | 20 | 23.0652 | 32.0214 | 1.4134 | 30.6160 | 1.9169 |
| Doppler | 5 | 454.6249 | 15.1864 | 42.6772 | 15.1732 | 41.8363 |
| | 10 | 143.7650 | 19.1933 | 16.9632 | 19.1356 | 16.8000 |
| | 15 | 45.4625 | 23.5611 | 6.2047 | 23.2806 | 6.4686 |
| | 20 | 14.3765 | 26.9427 | 2.8482 | 26.8339 | 2.8542 |

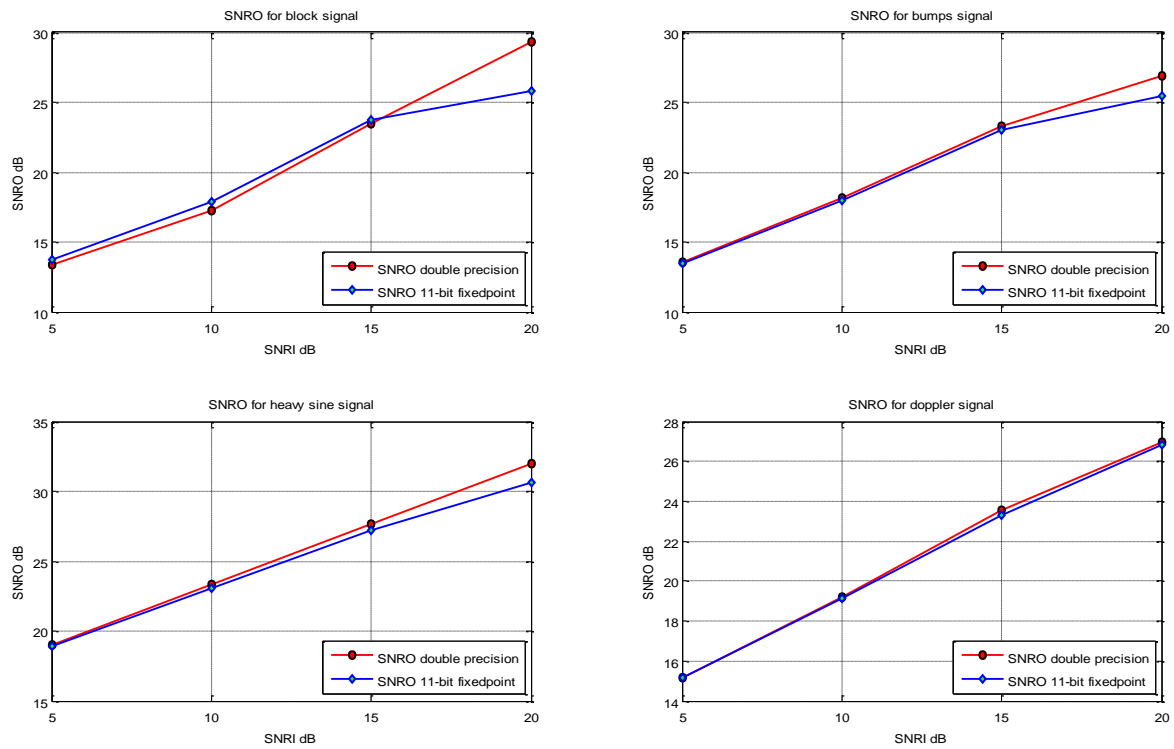


Fig. (5): Performance evaluation of benchmark signals hard thresholding denoising with double precision and 11-bit integer fixed point precision (signal length = 2048 samples).

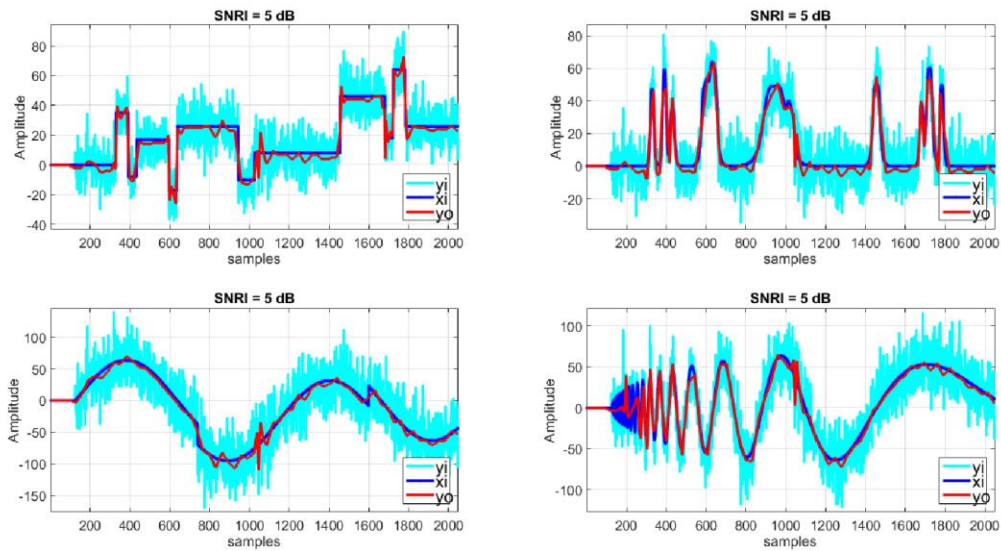


Fig. (6): Benchmark signals denoising by proposed 11-bit fixed point IWT architecture SNR_I at 5 dB. The signals are: Blocks (upper left); Bumps (upper right); Heavy sine (lower left); and Doppler signal (lower right).

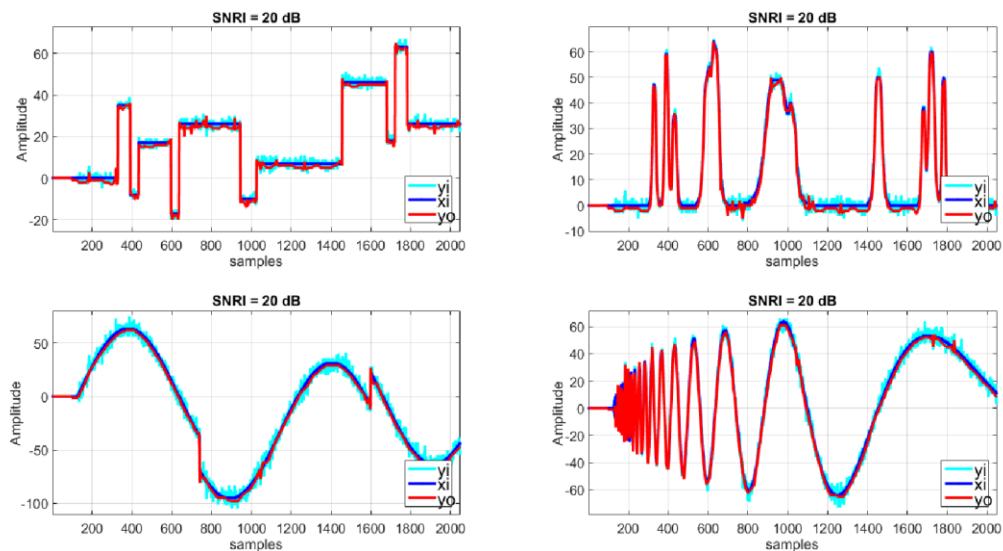


Fig. (7): Benchmark signals denoising by proposed 11-bit fixed point IWT architecture SNR_i at 20 dB. The signals are: Blocks (upper left); Bumps (upper right); Heavy sine (lower left); and Doppler signal (lower right).

5. DESIGN STATISTICS

The number of used elements inside the (XC3S700A FG484) FPGA on the Xilinx Spartan 3A used to implement the IWT architecture is listed in table (4). The percentage used slices inside FPGA are ($\approx 0\%$) in 1-level IWT and ($\approx 9\%$) in 5-level IWT.

Depending on the design summary file in Xilinx ISE, the maximum operating frequency, minimum period (Inverse of maximum operating frequency) after clock (critical path) are listed in table (5). The power consumption by the (XC3S700A FG484) FPGA for up to 5-level FIWT / BIWT with hard threshold module is about 61.347 milli watt.

Table (4): elements used inside (XC3S700A FG484) FPGA for IWT Architecture

| Architecture | AND gate | Invertor | OR gate 2input | XOR 2input | Flip flop | Slices | 4-input LUT | 3-input LUT | 2-input LUT | 2-input MUX |
|---------------------------------|----------|----------|----------------|------------|-----------|--------|-------------|-------------|-------------|-------------|
| Forward IWT (1level) | 64 | 2 | 32 | 96 | 49 | 25 | 2 | 0 | 0 | 0 |
| Backward IWT (1level) | 64 | 0 | 32 | 96 | 16 | 8 | 0 | 0 | 0 | 0 |
| Forward IWT with hard threshold | 128 | 2 | 64 | 193 | 65 | 33 | 17 | 11 | 0 | 0 |
| Forward IWT with hard threshold | 640 | 15 | 320 | 965 | 1157 | 579 | 95 | 40 | 40 | 40 |

Table (5): Timing analysis results

| Architecture | Level j | Minimum period (ns) | Maximum Frequency (MHz) |
|----------------------------------------|---------|---------------------|-------------------------|
| FIWT | One | 25.304 | 39.519 |
| BIWT | One | 27.411 | 36.482 |
| FIWT / BIWT with hard threshold module | One | 54.171 | 18.460 |
| FIWT / BIWT with hard threshold module | Five | 70.542 | 14.176 |

6. COMPARISON OF PERFORMANCE

6.1 Scalability And Computational Complexity

The proposed IWT architecture can be extended to any level of decomposition by connecting the blocks of the structure in cascade. This expansion doesn't affect the maximum working frequency of the system, it just increases the number of element used. On the other hand, the number of processed coefficient decrease at each new level that is because only the approximate coefficient of the previous level will be processed in the new level. So the total number of processed coefficient at J level can be calculated as:

$$n_j = 2N \sum_{j=1}^J \frac{1}{2^j} = 2N(1 - 2^{-J}) \quad (20)$$

where N is the length of the input samples, and J is number of levels.

6.2 Hardware utilization

The hardware utilization of the proposed structure increase as the number of the level increase which can be estimated as (Al-Sulaifanie et al, 2010)

$$UZ_J = \frac{n_j}{2N} = \sum_{j=1}^J \frac{1}{2^j} = 1 - 2^{-J} \quad (21)$$

As stated in this equation the hardware utilization of the structure varies from 50% for one level to about 97% for 5 levels as shown in table 6.

Table (6): Percentage hardware utilization for up to five-level real time IWT

| Level j | 1 | 2 | 3 | 4 | 5 |
|--------------------------|-----|-----|------|------|------------|
| Hardware Utilization (%) | 50% | 75% | ≈88% | ≈94% | ≈97% (Max) |

6.3 Comparison With Other Lifting Scheme

The proposed architecture in this work is compared with other lifting scheme algorithms as listed in Table 7. In term of the number of Multipliers, Adders and Shifters and Computation time, Hardware utilization, Switching complexity and the Memory usage. The comparison shows that our proposal has advantage in term of number

of multiplier in which it doesn't need any multiplier where the multiplication is performed by shift rotate. Also, as stated before the hardware utilization may reach to 100% as the number of the level increase, exceeding the other architecture hardware utilization, in addition to its advantage in simplicity, computation time and with no memory used.

Table (7): Comparisons of different Lifting-Based architecture.

| architectures | multipliers | adders | Register | shifter | Computation time | Hardware Utilization | complexity | Memory usage |
|------------------|-------------|--------|----------|---------|------------------|----------------------|------------|--------------|
| Andra et al 2002 | 2 | 4 | | 2 | ~2N | ~50% | Simple | Yes |
| Liao et al. 2004 | 2 | 4 | | 2 | ~N | ~97% | high | Yes |
| Chen 2004 | 2 | 4 | | 2 | ~N | ~97% | Moderate | No |
| Aziz et al 2012 | 0 | 2 | 3 | 1 | N+2 | ~100% | simple | No |
| Hsia et al 2015 | 0 | 9 | 14 | 2 | ~N | - | Moderate | No |
| Rafi et al 2017 | 0 | 4 | 4 | 2 | N+7 | - | simple | No |
| This paper | 0 | 4 | 2 | 2 | ~N | ~100% | Simple | No |

7. CONCLUSIONS

In this work, a VLSI integrated architecture for computing a 1-D IWT has been proposed. The architecture is simple and can be connected in cascade for computing multi-level decompositions. The architecture is independent of the size of the input signal and it is not requiring any memory, which is advantageous in VLSI design with respect to both area and time. To avoid overflow, an 11-bits word length architecture is implemented for two's complement 8-bits integer data input. The hardware utilization varies from (50%) in 1-level to (97%) for 5-level. An efficient formula is driven for controlling the delay introduced between tandem architectures that are adapted to work in real-time. The architecture is fitting for use in real time processing systems. The real time processing application in this work is the wavelet based denosing. The proposed architecture with hard threshold module is projected on (XC3S700A FG484) FPGA chip embedded on a Spartan 3A starter kit board. The performance of the proposed architecture in denosing is evaluated by denosing a four benchmarks signals corrupted by adding white Gaussian noise. The results expose a good performance of the proposed architecture in the denosing application. The proposed architecture is capable of operating with frequency up to about 39 MHz in 1-level to about 14MHz in 5-level for FIWT/ BIWT in an FPGA implementation. Although the proposed IWT architecture in this work uses the denosing application, it can be extended for other applications, like compression, which needs approximately the same procedure.

REFERENCES

- Ahmed Khorsheed Al-Sulaifanie, Arash Ahmadi and Mark Zwolinski (2010) "Very Large Scale Integration Architecture for Integer Wavelet Transform". IET Computers & Digital Techniques, Vol. 4, Iss. 6, pp. 471-483.
- Aziz, S. M and D.M. Pham (2012) "Efficient parallel architecture for multi-level forward discrete wavelet transform processors" Computers and Electrical Engineering, An International Journal, 38: 1325-1335, Elsevier
- Alfred Mertins (1999) "Signal Analysis, Wavelets, Filter banks, Time-Frequency Transforms and Applications", John Wiley and sons Ltd.
- Chen P.Y. (2004) "VLSI implementation for one-dimensional multilevel lifting-based wavelet transform", IEEE Transactions on Computers, Vol.53, No. 4, pp. 386-398.
- Chih-Hsien Hsia, Jia-Hao Yang, and Weihua Wang (2015) "An Efficient VLSI Architecture for Discrete Wavelet Transform" Proceedings of APSIPA Annual Summit and Conference 2015.
- D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, (1995a) "Wavelet shrinkage: Asymptopia" J. Roy. Stat.Soc., ser B, vol. 57, pp. 301-369.
- D. L. Donoho, I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage (1995b)" Journal of the American Statistical Association December 1995, Vol. 90, No. 432, 1200-1224.
- David S. Taubman and Michael W. Marcellin (2002) "JPEG2000: Image Compression Fundamentals, Standards and Practice" Kluwer Academic Publishers, Massachusetts.
- H. Liao, M.K. Mandal, and B.F. Cockburn, (2004) "Efficient Architectures for 1-D and 2-D Lifting-Based Wavelet Transform," IEEE Transactions on Signal Processing, vol. 52, no. 5, pp. 1315-1326.
- Hamid.Krim, Dewey. Tucker, Stephane Mallat and David Donoho, (1999) "On Denoising and Best Signal Representation" IEEE Trans. on Information Theory, vol.45, No.7, pp. 2225-2238.
- Ingrid Daubechies and Wim sweldens, (1996). "Factoring Wavelet Transforms into Lifting Steps". Bell Laboratories, Lucent Technologies.
- Kishore Andra, Chaitali Chakrabarti, and Tinku Acharya, "A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform" IEEE Transactions on Signal Processing, Vol. 50, Issue.4, pp. 966-977, April 2002.
- M. Bahoura and H. Ezzaidi, (2010) "FPGA-Implementation of Wavelet-based Denoising Technique to Remove Power-Line Interference from ECG Signal," in Information Technology and Application in Biomedicine (ITAB), 10th IEEE International Conference.
- M. Bahoura and H. Ezzaidi, (2010) "Real-Time Implementation of Discrete Wavelet Transform on FPGA" IEEE 10th International Conference Signal Processing (ICSP2010).
- Mohammad Rafi and Najeed-ud-Din (2017) "A High Speed Architecture for Lifting-based 2-D Cohen-Daubechies-Feauveau (5,3) Discrete Wavelet Transform used in JPEG2000" International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems Vol. 6, No. 1., 2017.
- Oweiss K.G., Mason A., Suhail Y., Kamboh A.M., Thomson K.E. (2007) "A scalable wavelet transform VLSI architecture for real-time signal processing in High-density intra-cortical implants", IEEE Trans. Circuits Syst. I, Regul.Pap., 54, (6), pp. 1266-1278.
- Stephane Mallat (1999) "A Wavelet Tour of Signal Processing" second edition, Academic press, Elsevier (USA).
- Tinku.Acharya and Ping-Sing Tsai (2005) "JPEG2000 Standard for Image Compression Concepts, Algorithm and VLSI Architectures" John Wiley & Sons, Inc., New Jersey.