An efficient non-separable architecture for Haar wavelet transform with Lifting Structure

SERWAN A. BAMERNI, AHMED K. AL-SULAIFANIE

Abstract: In this paper, a memory efficient, fully integer to integer with parallel architecture for 2-D Haar wavelet with lifting scheme has been proposed. The main problem in most 2-D architecture is the intermediate or internal memory (on-chip), which is mostly proportional to the image width, the increase of the internal memory lead to increases of the die area and control complexity. The proposed non-separable architecture is derived by rearranging and combining the lifting steps which are carried in both vertical and horizontal directions and performing it in simple and single step. In addition to the elimination of internal memory, the proposed algorithm is outperforming the existing architecture in term of hardware utilization, latency, number of arithmetic operation, power consumption, and used area. Finally, the proposed algorithm has been implemented on Xilinx Spartan 3A Development kit.

Keywords: Haar wavelet transform (HWT), _eld-programmable gate array (FPGA), lifting structure.

I. INTRODUCTION

Over the past few decades, a considerable number of studies have been conducted on two dimensional (2-D) discrete wavelet transforms (DWT) for image or video signals. Unfortunately, not all these proposed algorithms are suitable to be implemented for real time processing due to their complexity or processing time. Ever since Daubechies and Sweldens [1], introduce the lifting scheme to implement DWT, there has been a renewal of interest in employing this technique in hardware and software implementation in many applications, especially in attaining high throughput and low latency processing for high resolution image and video signals.

The authors are with the Electrical and computer engineering department, University of Duhok, IRAQ

Although lifting scheme has sped up the wavelet transform by reducing the number and complexity of the arithmetic operation, it still suffering from the high memory storage resources, where these memory issues dominate the hardware cost and complexity of the architectures for 2-D DWT. The storage resources include transposition memory, temporal memory and frame memory. Transposition memory is used in the 2-D DWT to transpose the intermediate results produced by the DWT in the row direction, for the input to the subsequent DWT in the column direction. Temporal memory is required for storing the partial results produced by applying the DWT in each direction, these two types together are commonly called internal memory. Frame memory is needed in multi-level DWT, that store the intermediate data produced at each level for the succeeding level, which it is usually advised to be the external memory.

Different algorithms have been proposed to decrease the memories hired in the lifting architecture, ranging from 2N to N^2 for internal memory (where N is the width and height of the NxN image) with line based scan [2, 3, 4, 5, 6].

Recently, Iwahashi et al. [20] presented a non-separable lifting scheme, by reassembling the update and predicted processing unit in row and column direction for both 3/5 and 7/9 lifting scheme, so all the band (LL, LH, HL and HH) generated at the same time. The advantage of this algorithm is the elimination of the transposition memory and decreasing the latency 25%, even though the number of arithmetic operations increased proportionally to the square of filter sizes.

In this paper we proposed a most efficient algorithm for implementing Haar wavelet transform with lifting structure. Haar wavelet has found many applications in both signal and image processing [7 – 18]. The proposed algorithm has superiority over the other

algorithm in term of memory usage, latency, number of arithmetic operation, power consumption, and used area.

The rest of the paper is organized as follows. Section 2 introduces the underlying concepts of the Haar wavelet transform and the derivation of the proposed algorithm equations. Section 3 compares the proposed architecture with other related studies. Section 4 presents the FPGA implementation results of different parameters. Finally, a brief conclusion is given in Section 5

II. LIFTING HAAR WAVELET TRANSFORM

II.1. 1-D Lifting scheme of Haar wavelet transform

To achieve the idea underlying the lifting scheme, we first have to decompose the analysis and synthesis filters into their polyphase components. The un-normalized analysis Haar filters banks have the following coefficients:

$$H_a(z) = \frac{1}{2} + \frac{1}{2}\, z^{-1}\,, norm_{Ha} = \frac{1}{\sqrt{2}}, \quad \ldots\ldots\ldots\ldots (1)$$

$$G_a(z) = -1 + z^{-1}\ , norm_{Ga} = \sqrt{2}\,. \quad \ldots\ldots\ldots\ldots (2)$$

The polyphase matrix which consisting of even and odd parts of the high-pass and low-pass filters be

$$P_a(z) = \begin{bmatrix} H_{ae}(z) & G_{ae}(z) \\ H_{ao}(z) & G_{ao}(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -1 \\ \frac{1}{2} & 1 \end{bmatrix} \quad \ldots\ldots\ldots\ldots (3)$$

The details and approximate output coefficients can be obtained by multiplying the polyphase matrix with the input signal.

$$\begin{bmatrix} A(z) \\ D(z) \end{bmatrix} = P_a(z) X(z)$$

$$\begin{bmatrix} A(z) \\ D(z) \end{bmatrix} = P_a(z) \begin{bmatrix} X_{ae}(z) \\ z^{-1} X_{ao}(z) \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} A(z) \\ D(z) \end{bmatrix}}_{\substack{out\ of \\ forward \\ stage}} = \underbrace{\begin{bmatrix} H_{ae}(z) & H_{ao}(z) \\ G_{ae}(z) & G_{ao}(z) \end{bmatrix}}_{\substack{analysis\ ployp\ hase \\ matrix}} \underbrace{\begin{bmatrix} X_{ae}(z) \\ z^{-1} X_{ao}(z) \end{bmatrix}}_{\substack{input\ of \\ forward \\ stage}} \quad \ldots\ldots\ldots\ldots (4)$$

As a result, the two equations for both $A(z)$ and $D(z)$ will be as:

$$D(z) = -X_{ae}(z) + z^{-1} X_{ao}(z)$$

$$A(z) = \frac{1}{2} X_{ae}(z) + \frac{1}{2} z^{-1} X_{ao}(z) = X_{ae}(z) - \frac{1}{2} X_{ae}(z) + \frac{1}{2} z^{-1} X_{ao}(z)$$

Therefore, we can write

$$A(z) = X_{ae}(z) + \tfrac{1}{2}\{-X_{ae}(z) + z^{-1} X_{ao}(z)\} \quad \ldots\ldots\ldots\ldots (5)$$

$$A(z) = X_{ae}(z) + \tfrac{1}{2} D(z) \quad \ldots\ldots\ldots\ldots (6)$$

Now denote the even and odd samples of the input as $X_e(z) = X_{ae}(z)$ and $X_o(z) = X_{ao}(z)$ respectively. This corresponds to the following implementation of the forward transform:

$$D(z) = -X_e(z) + X_o(z)$$

$$A(z) = X_e(z) + \frac{1}{2} D(z)$$

And the normalized details and approximate coefficients can be obtained by

$$D_n(z) = D(z) \times \tfrac{1}{\sqrt{2}} \quad \ldots\ldots\ldots\ldots (7)$$

$$A_n(z) = A(n) \times \sqrt{2} \quad \ldots\ldots\ldots\ldots (8)$$

These two equations can be implemented in following steps like any other lifting scheme as:

1. Splitting: The input signal $x(n)$ at sampling rate F_s is split into even $x_e(k) = x(2n)$ and odd $x_o(k) = x(2n-1)$ sequences with half sampling rates.
2. Prediction: the odd sample is predicted based on the even sample through a predicting operator P (in which it is 1 in Haar transform), and the prediction error is defined as the detail signal:

   $d(k) = -x_e(k) + x_o(k)$

3. Updating: the detail signal $d_1(k)$ is update through an updating operator *U*, to acquire the approximation signal:

   $a(k) = x_e(k) + \frac{1}{2} d_1(k)$

4. Scaling: both the details and approximate confidents will be scaled to obtain the normalized coefficients

   $a_n = a \times K_1$

   $d_n = d \times K_2$

These steps can be implemented as shown in the block diagram of Figure 1. While the reconstruction process can easily obtain by reversing the direction of the data flow and operators in the original formula then applying the merging process instead of splitting as shown in Figure 2.

$x_e(k) = K_2 a(k) - K_1 \frac{1}{2} d(k)$

$x_o(k) = K_1 d(k) + x_e(k)$

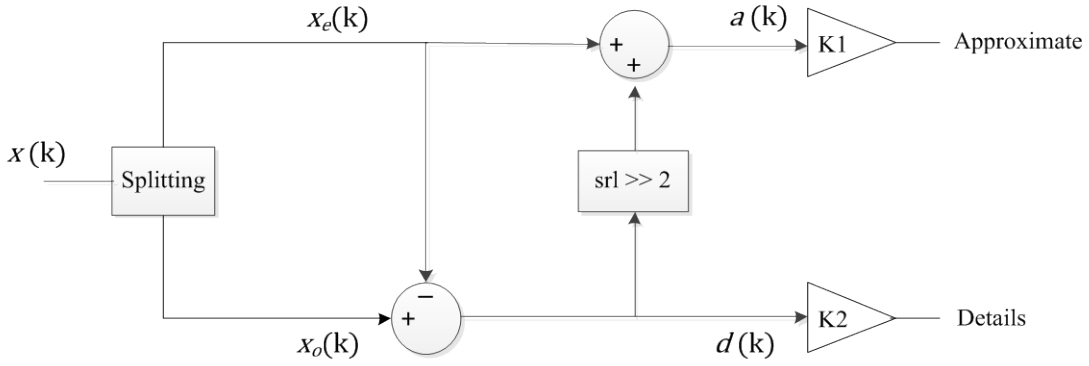

Figure 1. Block diagram of Haar wavelet lifting scheme

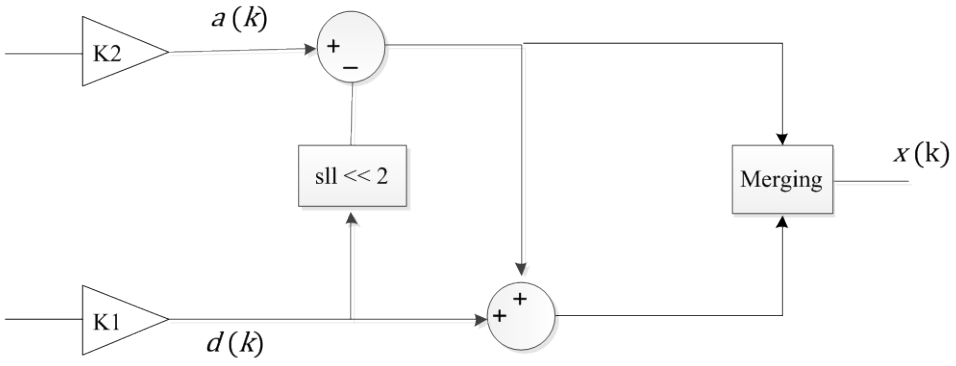

Figure 2 . Block diagram of reverse Haar wavelet lifting scheme

.

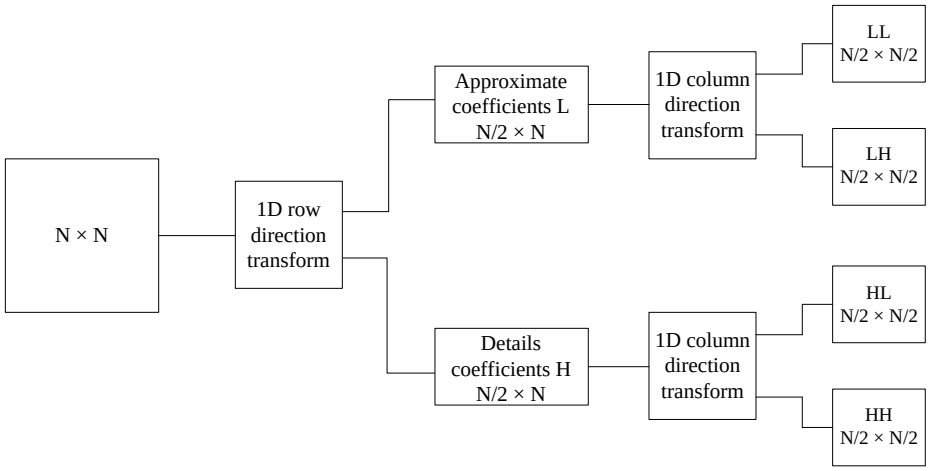

Figure 3. Conventional 2D wavelet transform

II.2. Non-separable 2D Haar wavelet lifting scheme:

In general, most of the 2D transform implemented by applying the a 1D transform in the row direction and then in column direction or vice versa, as shown in Figure 3 for wavelet transform The drawback of this method is the need of a high transposition memory which reachs at least to 2N for a matrix of NxN dimension [5], or modify the structure of the 1-D transform to reduce the transposition memory but that will be at the expense of the increase of frame memory.

In this paper, we propose a non-separable 2-D Haar wavelet lifting scheme which discerning from other architecture of no need for both transposition neither frame memory, no multiplication, zero latency and 100% hardware utility, with fully parallel processing.

The equations of the proposed method can be derived as following, with NxN image matrix as shown in figure 4 with the data reading direction as mentioned in the figure, the low pass and high pass subband resulted from applying the 1D Haar wavelet transform is obtained as follow

$$H_{11} = [X_{11} - X_{21}] \times \frac{1}{\sqrt{2}} = \frac{X_{11}-X_{21}}{\sqrt{2}} \quad \ldots\ldots\ldots\ldots (9)$$

$$L_{11} = \left[X_{21} + \frac{1}{2}H_{11}\right] \times \sqrt{2} = \frac{X_{11}+X_{21}}{\sqrt{2}} \quad \ldots\ldots\ldots\ldots (10)$$

In the same way

$$H_{12} = [X_{21} - X_{22}] \times \frac{1}{\sqrt{2}} = \frac{X_{21}-X_{22}}{\sqrt{2}} \quad \ldots\ldots\ldots\ldots (11)$$

$$L_{12} = \left[X_{21} + \frac{1}{2}H_{12}\right] \times \sqrt{2} = \frac{X_{21}+X_{22}}{\sqrt{2}} \quad \ldots\ldots\ldots\ldots (12)$$

Now, to obtained the subbands LL, LH, HL and HH, the 1D transform in row direction is applied on both the L and H respectively.

$$LH_{11} = [L_{11} - L_{12}] \times \frac{1}{\sqrt{2}} = \frac{X_{11}-X_{21}+X_{12}-X_{22}}{2} \quad \ldots\ldots\ldots\ldots (13)$$

$$LL_{11} = \left[L_{12} + \frac{1}{2}LH_{11}\right] \times \sqrt{2} = \frac{X_{11}+X_{21}+X_{12}+X_{22}}{2} \quad \ldots\ldots\ldots\ldots (14)$$

$$HH_{11} = [H_{11} - H_{12}] \times \frac{1}{\sqrt{2}} = \frac{X_{11}-X_{21}-X_{12}+X_{22}}{2} \quad \ldots\ldots\ldots\ldots (15)$$

$$HL_{11} = \left[H_{12} + \frac{1}{2}HH_{11}\right] \times \sqrt{2} = \frac{X_{11}+X_{21}-X_{12}-X_{22}}{2} \quad \ldots\ldots\ldots\ldots (16)$$

Since all subbands coeffiecnts are divided by 2, this division process can be omitted in all application without any effect on the results. The implementation of these equations is shown in Figure 5. Again the same equation can be used to implement the reconstruction process with only reversing the direction of the data flow as shown in Figure 6.

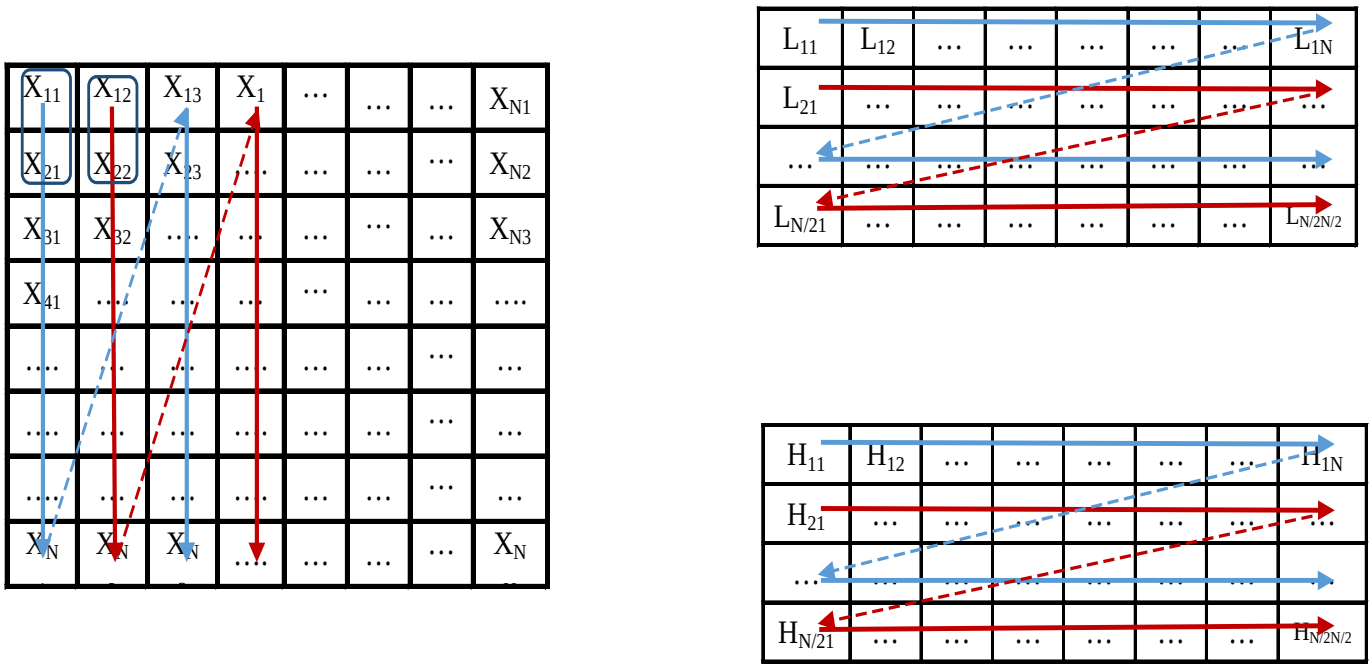

Fig. 4 the data flow direction

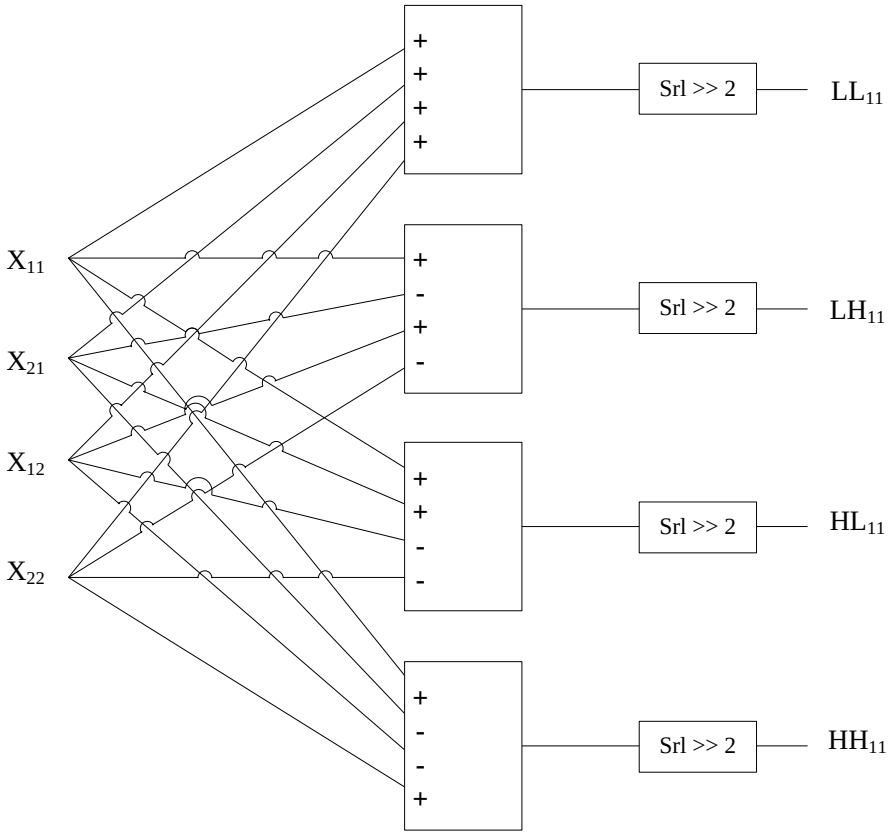

Figure 5. the forward Haar wavelet lifting scheme

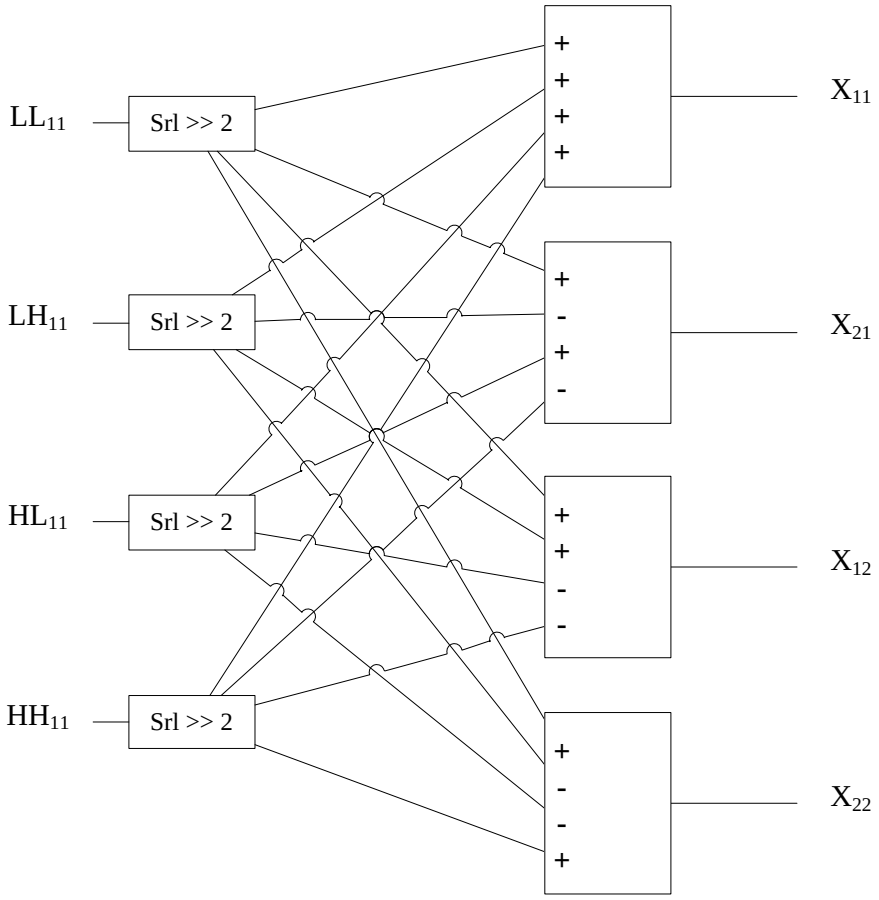

Figure 6. the forward Haar wavelet lifting scheme

III. COMPARISON

The proposed algorithm is compared with former architecture in the literature. Our comparison is restricted to the 2D 3/5 and 7/9 lifting architectures, since there is no pervious lifting architecture for Haar wavelet to the best knowledge of the authors. Table 1 summarizes the performance comparison for 2-D architectures includes evaluation in terms of computing time, system latency, internal memory usage, number of multipliers, and number of adders.

The results show that our proposed method is outperforms the other architecture in most of the criteria especially in the absence of the internal memory and the computing time due to the full parallelism of the algorithm, and the low latency of the system which represents the waiting clock cycles from original image input to first subband coefficient out, and the delay is only equal Tm + 2Ta, which indicate the multiplier and adder delay time.

Another comparison is made with the non-separable algorithm in [6]. Again the comparison is with both 5/3 and 9/7 wavelets, and also the results of the proposed method surpass the other method in many criteria and that is reasonable due to the difference of the filter lengths for each wavelet family, and here the application decides the type of filter to be chosen

.

Table 1. Comparisons of various 2D DWT architectures with the proposed one

	[19] (5/3)	[20] (5/3)	[5] (5/3)	[21] (9/7)	[3] (9/7)	[22] (9/7)	Our Haar
Computing time	$N^2/2 + N + 5$	N^2	$^3/_4 N^2 + ^3/_2 N + 7$	N^2	$22 + N^2 + 3N$	$N^2/4$	$N^2/4$
Internal memory	$3.5N$	$2.5N$	$2N$	$22N$	$5.5N$	$5.5N$	0
Latency	$2N + 5$	3	$^3/_2 N + 3$		-	N	1
Adders	8	6	8	36	8	32	12
Multipliers	4	4	0	36	6	18	4

Table 2. Comparisons of the non-separable architecture proposed in [20] with the proposed one

	[6] (5/3)	[6] (9/7)	Our Haar
Computing time	$N^2/4$ + 11	$N^2/4$ + 11	$N^2/4$
Transposition memory	0	0	0
Line buffers	8	28	0
Latency	5	11	1
Adders	10	24	12
Multipliers	10	24	4

Table 3. Hardware utilization results of the FPGA prototype

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	166	11776	1%
Number of occupied Slices	99	5888	1%
Number of Slices containing only related logic	99	99	100%
Number of Slices containing unrelated logic	0	99	0%
Total number of 4 input LUTs	166	11776	1%
Number of bounded IOBs	97	372	26%
Average Fanout of Non-Clock Nets	1.76		

Table 4. Timing summary of the proposed algorithm

Speed Grade	-4

Minimum period	1.263ns (Maximum Frequency: 791.766MHz)
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	15.651ns

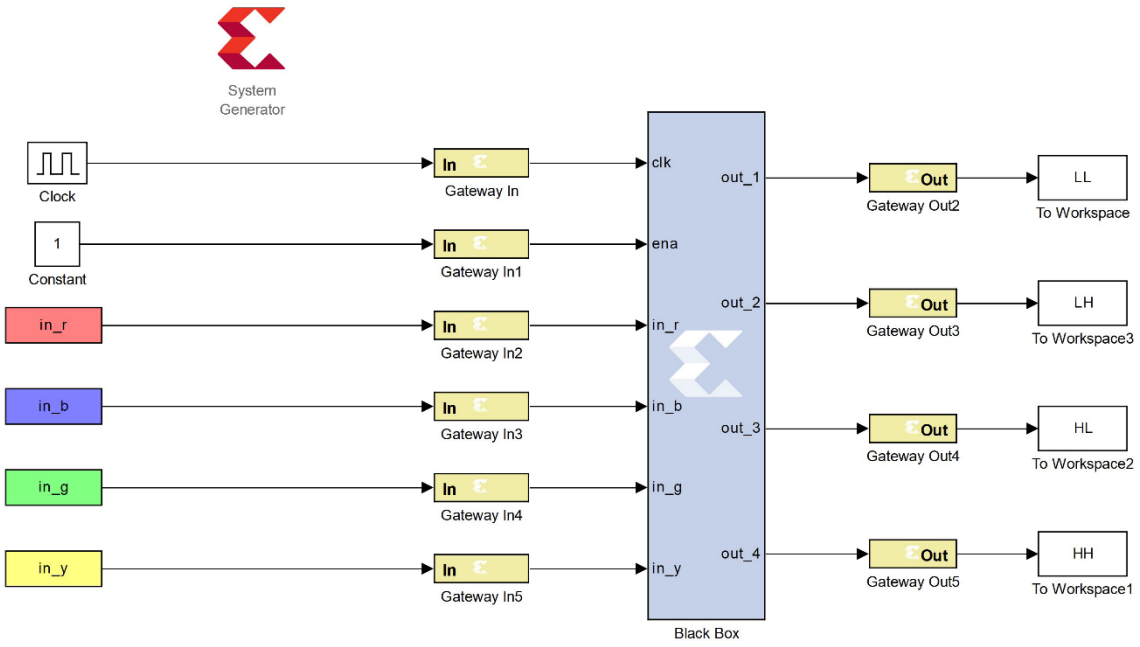

Figure 6. XSG system implementation

IV. FPGA DESIGN IMPLEMENTATION

The proposed architectures were implemented on FPGA using Xilinx System Generator (XSG) and Xilinx Spartan 3A Development kit. XSG is a high-level software tool that enables the use of MATLAB/Simulink environment to create and verify hardware designs for Xilinx FPGAs quickly and easily. It also provides full power consumption rated frequency reports, further includes a code generator that automatically generates HDL code from the created model. Generated HDL code can be synthesized and implemented in the Xilinx FPGAs. The XSG blocks are like standard Simulink blocks except that they can operate only in discrete-time and fixed-point format.

Table 3 summarizes the device utilization of the system implementation which shows the low number of system resources. Also, table 4 shows the maximum estimated frequency for the system is about 791 MHz, finally the consumed by this architecture is only 31mW.

V.CONCLUSION

An efficient architecture for implanting Haar wavelet transform is proposed in the paper. This algorithm has money advantages, the main one is the abandon of both the transportation and frame memory, also the full integer to integer transform due to the absence of multiplication processor in addition to the low hardware utilization, low power consumption and the fully

parallel process. Finally, the algorithm has successfully been verified using Spartan 3A Development kit.

References

[1] Daubechies, Ingrid, and Wim Sweldens. "Factoring wavelet transforms into lifting steps." Journal of Fourier analysis and applications 4.3 (1998): 247-269.

[2] Chen, Pei-Yin. "VLSI implementation of lifting discrete wavelet transform using the 5/3 filter." IEICE TRANSACTIONS on Information and Systems 85.12 (2002): 1893-1897.

[3] Wu, Bing-Fei, and Chung-Fu Lin. "A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec." IEEE Transactions on circuits and systems for video technology 15.12 (2005): 1615-1628.

[4] Mei, K., N. Zheng, and H. van de Wetering. "High-speed and memory-efficient VLSI design of 2D DWT for JPEG2000." Electronics Letters 42.16 (2006): 907-908.

[5] Hsia, Chih-Hsien, Jen-Shiun Chiang, and Jing-Ming Guo. "Memory-efficient hardware architecture of 2-D dual-mode lifting-based discrete wavelet transform." Ieee transactions on circuits and systems for video technology 23.4 (2013): 671-683.

[6] Iwahashi, Masahiro, and Hitoshi Kiya. "A new lifting structure of non separable 2D DWT with compatibility to JPEG 2000." Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE, 2010..

[7] Yang, Yuning, and Andrew J. Mason. "Frequency band separability feature extraction method with weighted haar wavelet implementation for implantable spike sorting." IEEE Transactions on Neural Systems and Rehabilitation Engineering 25.6 (2017): 530-538.

[8] Mousavi, A., and D. Asemani. "Signal detection improvement using Unbalanced Haar Wavelet transform for digital receivers." Electrical Engineering (ICEE), 2017 Iranian Conference on. IEEE, 2017.

[9] Lionnie, Regina, and Mudrik Alaydrus. "An analysis of Haar Wavelet Transformation for androgenic hair pattern recognition." Informatics and Computing (ICIC), International Conference on. IEEE, 2016.

[10] Sikka, Neha, Sanjay Singla, and Gurinder Pal Singh. "Lossless image compression technique using Haar wavelet and vector transform." Research Advances in Integrated Navigation Systems (RAINS), International Conference on. IEEE, 2016.

[11] Narayan, Yogendera, and Pankaj Nanglia. "A comparative analysis for Haar wavelet efficiency to remove Gaussian and Speckle noise from image." Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on. IEEE, 2016.

[12] Selvaperumal, Sathish Kumar, et al. "Haar wavelet transform based text extraction from complex videos." Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016 2nd International Conference on. IEEE, 2016.

[13] Shinde, Swapnil R., and Sudeep D. Thepade. "Gender classification with KNN by extraction of Haar wavelet features from canny shape fingerprints." Information Processing (ICIP), 2015 International Conference on. IEEE, 2015.

[14] Kaur, Sukhpal, and Madan Lal. "An invisible watermarking scheme based on Modified Fast Haar Wavelet Transform and RSGWPT." Recent Advances in Engineering & Computational Sciences (RAECS), 2015 2nd International Conference on. IEEE, 2015.

[15] Adi, Prajanto Wahyu, Farah Zakiyah Rahmanti, and Nur Azman Abu. "High quality image steganography on integer Haar Wavelet Transform using modulus function." Science in Information Technology (ICSITech), 2015 International Conference on. IEEE, 2015.

[16] Mahesh, Mahita, et al. "Image cryptography using Discrete Haar Wavelet transform and Arnold Cat Map." Communications and Signal Processing (ICCSP), 2015 International Conference on. IEEE, 2015.

[17] Tabassum, Fahima, Md Imdadul Islam, and Mohamed Ruhul Amin. "A simplified image compression technique based on Haar Wavelet Transform." Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on. IEEE, 2015.

[18] Prasad, Narasimha, Kishor Kumar Reddy, and Ramya Tulasi Nirjogi. "A Novel Approach for Seismic Signal Magnitude Detection Using Haar Wavelet." Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on. IEEE, 2014.

[19] Andra, Kishore, Chaitali Chakrabarti, and Tinku Acharya. "A VLSI architecture for lifting-based forward and inverse wavelet transform." IEEE Transactions on Signal Processing 50.4 (2002): 966-977.

[20] Chen, Shung-Chih, and Chung-Cheng Wu. "An architecture of 2-D 3-level lifting-based discrete wavelet transform." Proceeding of the VLSI Design/CAD Symposium. 2002.

[21] Jung, Gab-Cheon, and Seong-Mo Park. "VLSI implementation of lifting wavelet transform of JPEG2000 with efficient RPA (Recursive Pyramid Algorithm) realization." IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 88.12 (2005): 3508-3515.

[22] Xiong, Chengyi, Jinwen Tian, and Jian Liu. "Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme." IEEE transactions on image processing 16.3 (2007): 607-614.