A Block based Area-Delay Efficient Architecture for Multi-Level Lifting 2-D DWT

Abhishek Choubey, Basant Kumar Mohanty
Department of Electronics and Communication Engineering
Jaypee University of Engineering and Technology, Guna, Madhya Pradesh, 473226, INDIA

ABSTRACT
In this paper we have proposed a look-up-table (LUT) based structure for high-throughput implementation of multilevel lifting DWT. The proposed structure can process one block of samples to achieve high-throughput rate. Compared with the best of the similar existing structure, it does not involves any multipliers but it requires more adders and 21504 extra ROM words for J=3; its offers less critical path delay as compared to exiting structure. Synthesis results show that proposed structure has less ADP 56% less area and 13% less power compared to existing structure for block size J=2. Similarly proposed structure has 64% ADP and less power 21% as compared to existing structure for J=3. The proposed structure is fully scalable for higher block-sizes and it can offer flexibility to derive area-delay efficient structures for various applications.

Keywords
Look up table (LUT), VLSI, lifting, 2-dimensional (2-D) DWT.

1. INTRODUCTION
TWO-Dimensional (2-D) Discrete wavelet transform (DWT) is widely used in image and video compression [1]. Due to its superior performance over unitary transforms like discrete cosine transform (DCT), DWT has been adopted in image compression standards such as JPEG2000. 2-D DWT is highly computation intensive and implemented in VLSI systems to meet space-time requirement of various real-time applications. Several architectures have been suggested for efficient implementation of 2-D DWT, which could be categorized as either convolution-based or lifting-based structure. The convolution based designs involves more arithmetic and memory resources than lifting based designs [5]. Besides, lifting-based wavelet decomposition has many useful properties like symmetric forward and inverse transforms, in-place computation and integer-to-integer wavelet transform [4].

Multilevel 2-D DWT can be implemented in a straight forward manner using pyramid algorithm (PA). But hardware structures based on PA are not efficient. To overcome this difficulty Vishwanath has proposed the recursive pyramid algorithm (RPA) [2] which requires only one filtering unit and calculates all the decomposition levels. The RPA based designs, however, involves more on-chip memory and complex control circuits that they are equal.

Though hardware utilization efficiency (HUE) of the RPA based designs is much better than PA-based designs, it is still less than 100%. To overcome this problem, Wu et al [3] have suggested a folded scheme, where multi-level DWT computation is performed level-by-level using one filtering unit and one external buffer. Unlike RPA-based designs, folded design involves simple control circuitry and it has 100 % HUE. Keeping this in view, several architectures based have been proposed for efficient implementation of lifting 2-D DWT [5-12]. Most of the designs differ by their number of arithmetic components, on-chip memory, cycle period and throughput rate. It is observed that the on-chip and off-chip memories of all the folded designs are almost independent of input block-size.

Block-processing method is found to have significant potential to reduce memory requirement and memory-bandwidth of the DWT structure. Recently, Hu *et al* [11] have suggested a block-based design for high-throughput implementation multi-level 2-D DWT. The arithmetic components of the proposed structure increases proportionately with block size which results in increase of the logic resource of the structure accordingly. In this paper, we aim at deriving a block based design of 2-D DWT using 2-D lifting algorithm using LUT based multiplier. Keeping these facts in mind, we have derived efficient structure for multi-level 2-D DWT lifting structure computation.

2. MATHEMATICAL FORMULATION FOR LIFTING 2-D DWT AND LUT
1-D lifting DWT computation is performed row-wise and then column-wise to obtain 2- D DWT output. For the j-th level decomposition, the low-low subband of (j-1)-th level (A^{j-1}) is decomposed into four sub bands namely low-low ($\mathbf{A}^j$), low-high ($\mathbf{B}^j$), highlow ($\mathbf{C}^j$) and high-high ($\mathbf{D}^j$) sub-bands. The input data-matrix (**X**) represents the low-low sub-band of the zero-th level. The row and column-wise computation of 2-D lifting DWT can be expressed in separable form as discussed in the following.

The following set of recursive relations are derived for row-wise computation of j-th level 2-D lifting DWT.The following set of recursive relations are derived for row-wise computation of j-th level 2-D lifting DWT.

$$s_{11}(m,n) = x(m,2n\text{-}1) + \alpha((x(m,2n) + x(m,2n\text{-}2))$$
$$s_{12}(m,n) = x(m,2n\text{-}2) + \beta((s_{11}(m,n) + s11(m,n\text{-}1))$$
$$u_h(m,n) = s_{11}(m,n\text{-}1) + \gamma((s_{12}(m,n) + s12(m,n\text{-}1))$$
$$u_l(m,n) = s_{12}(m,n\text{-}1) + \delta((u_h(m,n) + u_h(m,n\text{-}1)) \quad (1)$$

where α, β, γ and δ are lifting constants. Since the computation of the low-pass and high-pass components of intermediate outputs ($u_l(m,n)$) and ($u_h(m,n)$), respectively, are of similar form, the computations of those components for column-wise computation are expressed in a general form as:

$$s_{21}(m,n) = u(2m\text{-}1,n) + \alpha((u(2m,n) + u(2m\text{-}2,n))$$
$$s_{22}(m,n) = u(2m\text{-}2,n) + \beta((s_{21}(m,n) + s_{21}(m\text{-}1,n))$$
$$v_h(m,n) = s_{21}(m\text{-}1,n) + \gamma(s_{22}(m,n) + s_{22}(m\text{-}1,n))$$
$$v_l(m,n) = s_{22}(m\text{-}1,n) + \delta((v_h(m,n) + v_h(m\text{-}1,n)) \quad (2)$$

where $x(m, n)$ represents the low-low sub-band components of the (j-1)-th level. $u(m, n)$ represents the intermediate output corresponding to the input $x(m, n)$, which could be low-pass and high-pass component $u_l(m, n)$ and $u_h(m, n)$, respectively. Similarly, $v_h(m, n)$ is the high-pass output corresponding to the intermediate output $u_l(m,n)$ and $u_h(m,n)$, which, respectively, represent the pair of sub-band outputs B(m,n) and D'(m, n). $v_l(m,n)$ is the low-pass outputs corresponding to the intermediate output $u_l(m, n)$ and $u_h(m,n)$ which, respectively, represent the other two sub-band outputs A'(m,n) and C(m,n).

The scale normalization of row and column lifting 2-D DWT can be integrated and performed in one step at the end of the column computation using the following equation:

$$A(m,n) = K^2.A'(m,n)$$

$$D(m,n)=1/K^2.D'(m,n) \quad (3)$$

3. PROPOSED STRUCTURE

The basic lifting structure is comprised of adders and multiplier. The lifting structure is used constant multipliers. The functional element (FE) used generic multiplier which is involves more area; hence we have proposed efficient LUT multiplier which is shown in fig.1. In proposed multiplier partial product values are stored in LUT. The structure of proposed multiplier is shown in fig.2.The proposed multiplier is comprised of 3 LUTs with 16 words. The proposed LUT multiplier for multiplication of a number X with a given constant C is possible for both 2's complement representations of X and C. Besides, both X and C could be fixed point format. The partial product addition of proposed multiplier is shown in figure 3.The word width of partial product rows is 16-bit.

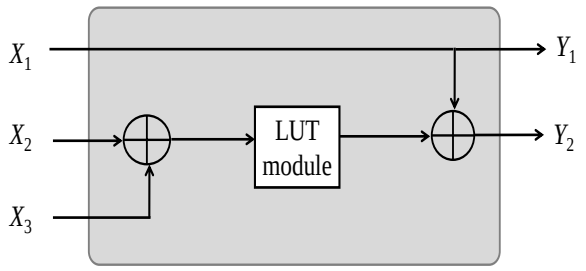

Figure1: Functional element (FE)

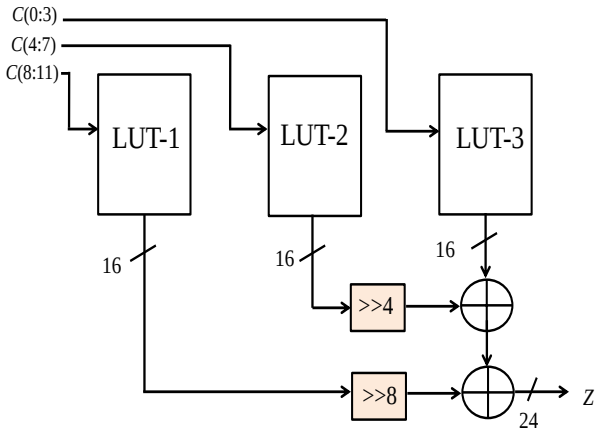

Figure 2: Proposed LUT module (Multiplier)

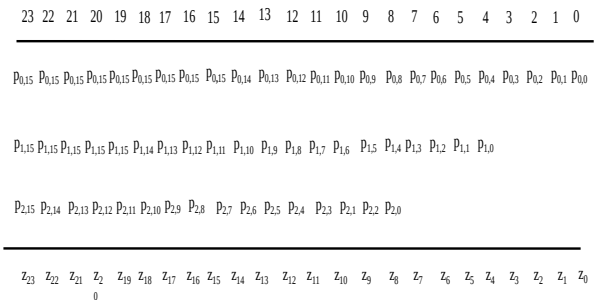

Figure 3: Partial product array of LUT multiplier for 12-bit

The corresponding product values and partial product words of α, β, γ, δ and k and 1/k stored in LUT are shown in Table 1 to 6 respectively. The product word corresponding to C (11:0) is selected. Based on modified PE we have derived 2-D DWT structure for block size 16 is shown in figure 4. It consists of a row processor (RP) and a column processor (CP). RP consists of 8 processing element, where each PE performs lifting computation of 9/7 filter. RP and CP is comprised four functional element. Each PE performs one lifting-step computation. Four processing elements form a pipeline structure. PE- receives a pair of samples in every cycle and compute a pair of 9/7 filter outputs [low-pass{$u_l(n)$} and high-pass {$u_h(n)$}]. The structure of row processor and column is shown in figure 5. The structure of row processor and column processor is similar expect register and shift register.

Table 1 Partial product values of α

Address	Product values	Partial Product values
0000	0	0000000000000000
0001	α	111110.0110101000
0010	2α	111100.1101010000
0011	3α	111011.0011111000
0100	4α	111001.1010100000
0101	5α	110000.0010010000
0110	6α	110110.0111110000
0111	7α	110100.1110011000
1000	8α	110011.0101000000
1001	9α	101100.1111100000
1010	10α	110000.0010010000
1011	11α	111110.1000110111
1100	12α	111100.1111100100
1101	13α	111111.0110001100
1110	14α	101001.1100110000
1111	15α	100000.0100100000

Table 2 Partial product values of β

Address	Product values	Partial Product values
0000	0	0000000000000000
0001	β	111111.1111001010
0010	2β	111111.1110010100
0011	3β	111111.1111011110
0100	4β	111111.1100101000
0101	5β	111111.1011110010
0110	6β	111111.1110111100
0111	7β	111111.1110000110
1000	8β	111111.1001010000
1001	9β	111111.1101111000
1010	10β	111111.0111100100
1011	11β	111111.1010101110
1100	12β	111111.0101111000
1101	13β	111111.0101000010
1110	14β	111111.1100001000
1111	15β	111111.1111001000

Table 3 Partial product values of γ

Address	Product values	Partial Product values
0000	0	0000000000000000
0001	1γ	000000.1110001000
0010	2γ	000001.1100010000
0011	3γ	000000.1110001000
0100	4γ	000011.1000100000
0101	5γ	000100.0110101000
0110	6γ	000101.0100110000
0111	7γ	000110.0010111000
1000	8γ	000011.0001000000
1001	9γ	001010.1001100000
1010	10γ	001000.1101010000
1011	11γ	001000.1011011000
1100	12γ	001010.1001100000
1101	13γ	001011.0111101000
1110	14γ	001100.0101110000
1111	15γ	010001.1010100000

Table 4 Partial product values of δ

Address	Product values	Partial Product values
0000	0	0000000000000000
0001	δ	00000.0111000110
0010	2δ	000000.1110001100
0011	3δ	000001.0101010010
0100	4δ	000001.1100011000
0101	5δ	000010.0011011110
0110	6δ	000010.1010100100
0111	7δ	000011.0001101010
1000	8δ	000011.1000110000
1001	9δ	000101.0101001000
1010	10δ	000100.0110111100
1011	11δ	000100.1110000010
1100	12δ	000101.1110000010
1101	13δ	000000.0111000110
1110	14δ	000110.0011010100
1111	15δ	001000.1101111000

Table 5 Partial product values of *k*

Address	Product values	Partial Product values
0000	0	0000000000000000
0001	k	000001.0010011001
0010	$2k$	000010.0100110010
0011	$3k$	000001.0010011001
0100	$4k$	000100.1001100100
0101	$5k$	000001.001001100
0110	$6k$	000110.1110010110
0111	$7k$	000001.0010011001
1000	$8k$	001001.0011001000
1001	$9k$	001101.1100101100
1010	$10k$	001011.0111111010
1011	$11k$	000001.0010011001
1100	$12k$	001101.1100101100
1101	$13k$	001110.1111000101
1110	$14k$	010000.0001011110
1111	$15k$	010110.1111110100

Table 6 Partial product values of 1/*k*

Address	Product values	Partial Product values
0000	0	000000.0000000000
0001	$1/k$	000000.1101111010
0010	$2/k$	000001.1011110101
0011	$3/k$	000010.1001101111
0100	$4/k$	000011.0111101010
0101	$5/k$	000100.0101100100
0110	$6/k$	000101.0011011111
0111	$7/k$	000110.0001011001
1000	$8/k$	000110.1111010100
1001	$9/k$	001010.0110111110
1010	$10/k$	001000.1011001001
1011	$11/k$	001001.1001000011
1100	$12/k$	001001.1110111101
1101	$13/k$	001010.1100110111
1110	$14/k$	001100.0010110010
1111	$15/k$	010001.0110010010

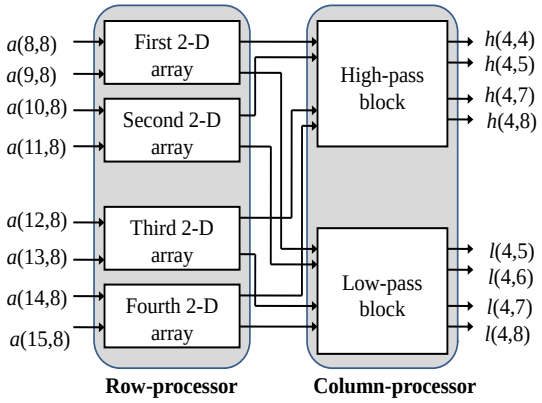

Figure 4. 2-D DWT structure of block size 16

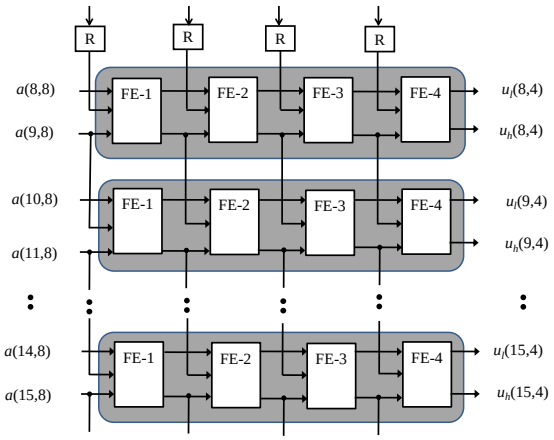

Figure 5. Structure of row processor and column processor for block size 16.

4. HARDWARE-TIME COMPLEXITIES AND PERFORMANCE COMPARISON

4.1 Theoretical comparison

The theoretical estimates of hardware and time complexities of proposed structures and existing structure of [11] are listed in Table 7 for n = 12. As shown in Table 7, the proposed structure does not involve multiplier unlike the existing structure for any block size 2-D DWT implementation. The multiplier requires more area as compared to adder hence proposed structure has less area as compared to existing structure. However proposed structure requires 5210 more words and 21504 words for block size 16 and 64 respectively. The critical path of proposed structure involves $3T_A$ delay for block size 16 and 64. Hence proposed structure is involves less critical path as compared to existing structure.

4.2 Synthesis Results

We have coded the proposed structure and the existing structure of [11] for J=2 and J=3 in VHDL to estimate area, delay and power consumption of the designs. We have synthesized the proposed structure and existing structure Synopsys Design Compiler using TSMC 65-nm CMOS standard cell library. As shown in Table 8, the proposed structure involve 64% and 55% less ADP and consumes 13% and 21% less power than those of the structure of [11] for J=2, block size 16 and J=3, block size 64, respectively.

Table 7: Theoretical comparison of proposed structure and the existing structures

DWT level	Designs	Block Size	Multi -plier	Adder	ROM words	On chip Memory	Clock cycle period	CT
J=2	Hu *et al* [11]	16	106	172	0	$2N+256$	$T_M+T_A+T_X$	$N^2/16$
	Proposed (Using LUT)	16	0	320	5120	$6N+160$	$3T_A$	$N^2/16$
J=3	Hu *et al* [11]	64	426	684	0	$3N+908$	$T_M+T_A+2T_X$	$N^2/64$
	Proposed (Using LUT)	64	0	1280	21504	$7N+672$	$3T_A$	$N^2/64$

Table 8: Synthesis results of proposed structure and the existing structures

DWT Level	Designs	Block Size	MCP (ns)	Total area (mm.sqm.)	ADP (µ.sqm. s)	Core Power (mW)
J=2	Hu *et al* [11]	16	1.66	217180.08	.36	12.12
	Proposed structure (using LUT)	16	.86	183352.86	.16	10.47
J=3	Hu *et al* [11]	64	1.81	828532.80	1.45	29.97
	Proposed structure (using LUT)	64	.97	526063.52	.52	23.63

5. CONCLUSION

We have proposed a look-up-table (LUT) based structure for high-throughput implementation of multilevel lifting 2-D DWT structure. The structure is fully scalable for higher block sizes and multi level 2-D DWT implementation. The proposed structure does not involve any multiplier; this is a major advantage of proposed structure when the structure is implemented for higher block sizes. It offers less 64% less ADP and 21% less power compared to best existing structure for J=3.

6. REFERENCES

[1] Y. Meyer, Wavelets: Algorithms and Applications. Philadelphia: Society for Industrial and Applied Mathematics (SIAM), 1993.

[2] M. Vishwanath, The recursive pyramid algorithm for the discrete wavelet transform, IEEE Trans. Signal Processing, vol. 42, no. 3,pp. 673676, Mar. 1994.

[3] P.-C. Wu and L.-G. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 4, pp. 536–545, Apr. 2001.

[4] W. Sweldens, "The lifting scheme: A costom-designe construction of biorthogogal wavelets, " Applied and Computational Harmonic Analysis, vol. 3, no. 2, pp. 186–200, 1996.

[5] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architecture for 1-D and 2-D lifting-based wavelet transform," IEEE Trans. Signal Process., vol. 52, no. 5, pp.1315-1326, May 2004.

[6] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform," IEEE Trans. Signal Processing, vol. 53, no. 4, pp. 1575–1586, Apr. 2005.

[7] C.-C. Cheng, C.-T. Huang, C.-Y. Cheng, C.-Jr. Lian and L.-G. Chen, "On-chip memory optimization scheme for VLSI implementation of line-based two dimensional discrete wavelet transform," IEEE Trans. on circuit and System for Video Technology, vol. 17, no. 7, pp. 814-822, July 2007.

[8] B. K. Mohanty and P. K. Meher, "Memory efficient modular VLSI architecture for high throughput and low-latency implementation of multilevel lifting 2-D DWT," IEEE Trans. Signal Process., vol. 59, no. 5,pp. 2072–2084, 2011.

[9] X. Tian, L. Wu, Y.-H.Tan, and J.-W. Tian, "Efficient multi-input/multi-output VLSI architecture for two-dimensional lifting-based discrete wavelet transform," IEEE Trans. Comput., vol. 60, no. 8, pp.1207–1211, 2011

[10] B. K. Mohanty, A. Mahajan, and P. K. Meher, "Area- and power-efficient architecture for high-throughput implementation of lifting 2-D DWT," IEEE Trans. Circuits Syst. II, Express Briefs, vol. 59, no. 7, pp.434–438, 2012.

[11] Y Hu, and C C Jong "A Memory-Efficient High-Throughput Architecture for Lifting-Based Multi-Level 2-D DWT" IEEE Transactions on signal processing, vol. 61, no. 20, October 15 2013.

[12] B K Mohanty, and A Choubey "Efficient Design for Radix-8 Booth Multiplier and Its Application in Lifting 2-D DWT" Circuits System Signal Processing (CSSP) Vol.36 pp.1129–1149.(2017)