Asian Journal of Information Technology 16 (6): 511-520, 2017
ISSN: 1682-3915

FPGA Implementation of Area Efficient and High Throughput 2D DTCWT Architecture with Pipelined Scheme

Venkateshappa and P. Cyril Prasanna Raj
Department of Electronics and Communication Engineering, MS Engineering College,
562110 Bengaluru, Karnataka, India

Abstract: Dual Tree Complex Discrete Wavelet Transform (DTCWT) decomposes input image into approximation and six detail sub-bands using row and column processing filter banks. Each filter comprising of 10 coefficients requires multipliers and adders that are twice larger than that of discrete wavelet transform computation. In this study, the computation complexity in DTCWT computation is reduced by considering the redundancy in filter coefficients. A multiplexer-demultiplexer based logic is designed to reduce the number of filters by 75%, a pipeline architecture is designed to reduce the number multipliers by 60% as compared with conventional DTCWT architecture. The designed architecture implemented on FPGA and the design operates at frequency of >200 MHz.

Key words: Image compression, DTCWT, SPIHT, DWT, companding, FPGA

INTRODUCTION

Image compression plays an important role in today's multimedia applications by compressing raw image data with compression ratios >100. Image compression standards such as JPEG 2000 (Pastuszk and Abramowski, 2016; Taubman and Marcellin, 2002) and video compression standards such as H.265 (Sullivan *et al*., 2012) recommend use of wavelets for image transformation into frequency sub bands. Encoding techniques (Taubman, 2000) such as arithmetic coding, SPIHT encoding and variable length coding encodes wavelet bands into binary bits. The process of quantization and thresholding of wavelet bands prior to encoding determines the compression ratios (Xiong *et al*., 2003). The sub-bands of Discrete Wavelet Transform (DWT) such as LL, LH, HL and HH (with level 1) consists of intensity, vertical, horizontal and diagonal features (edges), respectively. Given an input image of size N×N after level 1 2D DWT decomposition has four sub bands each of size N/2×N/2. Considering the LL band (dominant information in terms of intensity) and significant features from all three sub bands, SPIHT encoding logic encodes the significant information achieving compression. The significant features from higher sub bands contain edge information in 0, 90 and 45° directions. The input image to be compressed may contain edge information in all other orientations apart from 0°, 90° and 45°. In order to capture the significant orientations such as 0, ±15, ±45 and ±75° Dual Tree Complex Wavelet Transforms (DTCWT) has been proposed by Kingsbury (2001). Discussion on DTCWT demonstrating its advantages in image processing applications with regard to shift insensitive and directionality are presented by Taubman (2000), Xiong *et al*. (2003), Kingsbury (2001) and Fang *et al*. (2010). Fang *et al*. (2010) have proposed a new image compression algorithm that is based on DT-CWT and SPIHT. DTCWT bands have been effectively utilized to perform image processing and image compression. DTCWT computation based on Kingsbury (2001) proposes real and complex filters to compute orientations in six orientations. 2D DTCWT level 1 decomposition requires four filters for row processing and sixteen filters for column processing (Kingsbury, 2000). The computation complexity of DTCWT is twice more than DWT processing. In this study, DTCWT architecture is designed that requires multipliers and adders that are as that of DWT architecture requirements. The redundancies in filter coefficients are eliminated and optimum architecture is designed for DTCWT computation.

MATERIALS AND METHODS

DTCWT algorithm: The 2D DTCWT architecture for level 1 decomposition based on the discussions by Neumann and Steidl (2005) is shown in Fig. 1. The first stage is the row processor with real tree filter pairs $F_h = \{H_{0a}, H_{0b}\}$ and imaginary tree filter pair $F_g = \{H_{1a}, H_{1b}\}$ satisfying the conditions $H_{1x} = H_{0x}(n-1)$. The second

Corresponding Author: Venkateshappa, Department of Electronics and Communication Engineering, MS Engineering College, 562110 Bengaluru, Karnataka, India

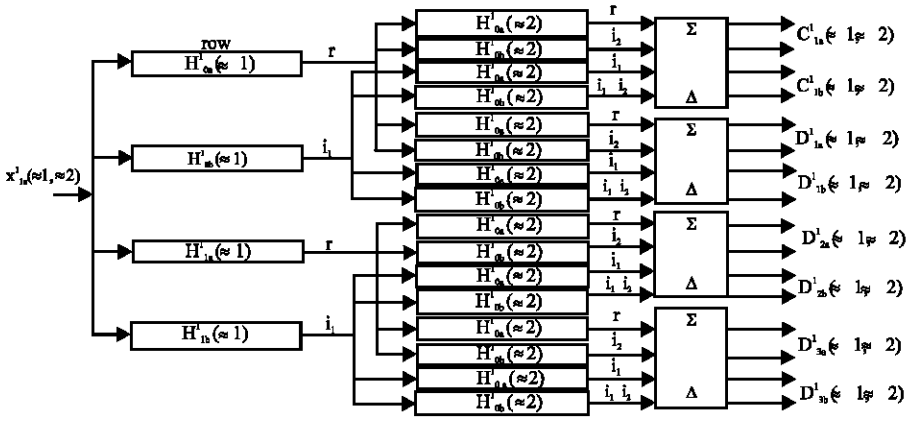


Fig. 1: Single stage dual trees CWT

stage is the column processor consisting of sixteen filters. The input image after two stage filtering gives rise to sixteen sub-bands of which four of them are approximation (low pass) sub bands and remaining are detail (highpass) sub-bands. The four low pass bands comprises of two real low pass and two imaginary low pass bands. Similarly, twelve high pass sub bands comprises of six real high pass bands and six imaginary high pass bands. Denoting the DTCWT output of approximation and detail coefficients as {C, D} for a given input S at level j, the 4 outputs are represented as in Eq. 1-4:

$$C^{j}_{(a/b)}{}^{j}(Z_1, Z_2) = (2_J)((A^{j}_{a}(Z_1) \pm j(A^{j}_{b}(Z_1)) \\ (A^{j}_{\alpha}(Z_2)+j(A^{j}_{b}(Z_2))\, S(Z_1, Z_2)) \quad (1)$$

$$D^{j}_{(1a/b)}(Z_1, Z_2) = (2_J)((A^{j}_{a}(Z_1) \pm j(A^{j}_{b}(Z_1)) \\ (B^{j}_{a}(Z_2)+j(B^{j}_{b}(Z_2))\, S(Z_1, Z_2)) \quad (2)$$

$$D^{j}_{(2a/b)}(Z_1, Z_2) = (2_J)((A^{j}_{a}(Z_1) \pm j(B^{j}_{b}(Z_1)) \\ (A^{j}_{a}(Z_2)+j(A^{j}_{b}(Z_2))\, S(Z_1, Z_2)) \quad (3)$$

$$D^{j}_{(3a/b)}(Z_1, Z_2) = (2_J)((B^{j}_{a}(Z_1) \pm j(B^{j}_{b}(Z_1)) \\ (B^{j}_{a}(Z_2)+j(B^{j}_{b}(Z_2))\, S(Z_1, Z_2)) \quad (4)$$

The subscript terms a/b is associated with the sign (±) in the first term, C^{j}_{a} and C^{j}_{b} represents the two complex low-pass sub bands, $D^{j}_{1a}(-15^0)$, $D^{j}_{1b}(+15^0)$, $D^{j}_{2a}(-75^0)$, $D^{j}_{2b}(+75^0)$, $D^{j}_{3a}(-45^0)$ and $D^{j}_{3b}(+45^0)$ represents the six complex high pass sub bands.

The term $A_1(Z) = H^{1}_{0}(Z)$ and $B_1(Z) = H^{1}_{1}(Z)$ represents the low pass and high pass filter coefficients and are related as in Eq. 5:

$$H^{1}_{0a}(Z) = H_0(Z),\ H^{1}_{0a}(Z) = H_1(Z) = (Z^{-1})H_0(-Z^{-1}) \\ H^{1}_{0b}(Z) = Z^{-1}\ H_0(Z),\quad H^{1}_{0b}(Z) = Z^{-1}\ H_1(Z) = Z^{-2}H_0(-Z^{-1}) \quad (5)$$

The approximation coefficients obtained after level 1 decomposition is further decomposed into level 2 sub-bands with the row and column processors. Multilevel decomposition is achieved by decomposition of low pass bands at each level. The cascaded filter bank for N levels (j = 1, 2, 3, ..., N) can be represented as in Eq. 6:

$$A^{j}(Z) = H^{1}_{0}(Z), ..., H^{j-1}_{0}(Z^{2j-2}) = H^{j}_{0}(Z^{2j-1}) \\ B^{j}(Z) = H^{1}_{0}(Z), ...,\ H^{j-1}_{0}(Z^{2j-2})\, H^{j}_{1}(Z^{2j-1}) \quad (6)$$

Table 1 shows the filter coefficients for the first stage decomposition and Table 2 shows the filter coefficients for second and third stage decomposition.

Computation complexity in DTCWT: Considering the input image of size N×N, each of the N rows consisting of N pixels is simultaneously processed by four filters consisting 10 filter coefficients. The number of multiplications and additions per output sample is 10 and 9, respectively. For N pixels the numbers of multiplication and addition operations are 10 and 9 N, for N rows of N pixels the number of multiplications and additions are 10 and 9 N^2. First stage consists of four filters thus the number of multiplications and additions are 40 and 36 N^2. The second stage consists of 16 filters thus the number of multiplication and additionoperations for second stage are 160 and 144 N^2. For the first stage the total number of multiplication and addition operations is 200 and 180 N^2. For level 2 decomposition, the number of multiplication and addition operations is 200 and 180 $(N/2)^2$. For j stages of decomposition the total number of multiplication and

Table 1: DTCWT filters for level 1

Tree a		Tree b	
H_{0a}	H_{1a}	H_{0b}	H_{1b}
0	0	0.01122679	0
-0.08838834	-0.01122679	0.01122679	0
0.08838834	0.01122679	-0.08838834	-0.08838834
0.69587998	0.08838834	0.08838834	-0.08838834
0.69587998	0.08838834	0.69587998	0.69587998
0.08838834	-0.69587998	0.69587998	-0.69587998
-0.08838834	0.69587998	0.08838834	0.08838834
0.01122679	-0.08838834	-0.08838834	0.08838834
0.01122679	-0.08838834	0	0.01122679
0	0	0	-0.01122679

Table 2: DTCWT filters for higher levels

Tree a		Tree b	
H_{00a}	H_{01a}	H_{00b}	H_{01b}
0	0	0	0.03516384
0.03516384	0	0	0
-0.08832942			
0.08832942			
0.023389032	0	0	0.023389032
0.76027237	0.58751830	0.58751830	-0.76027237
0.58751830	-0.76027237	0.76027237	0.58751830
0	0.023389032	0.023389032	0
-0.11430184	0.08832942	-0.08832942	-0.11430184
0	0	0	0
0	-0.03516384	0.03516384	0

addition operations is 200 and $180N^2$. For j stages of decomposition the total number of multiplication and addition operations is:

$$200N^2\left(\sum_{i=0}^{j}\frac{1}{2^i}\right) \text{and} 180N^2\left(\sum_{i=0}^{j}\frac{1}{2^i}\right)$$

The input image N×N is stored in a memory of size N^2. The first rows of N pixels are read into the input memory of row processor. The first stage filter processes the N pixels and the results are stored in four output memory of size N. Thus, the first stage row processer requires intermediate memory of size 4 N. The column processor consists of four groups of filter bank with each group comprising of four filters. Thus, the column processor requires 16 N input memory and 16 N output memory. The total number of intermediate memory required for level 1 decomposition is 37N (= N+4N+16N+16N). Second stage decomposition requires 37N/2 memory elements, the total number of intermediate memory for j stages are:

$$37\left(\sum_{i=0}^{j}\frac{1}{2^i}\right)$$

In order to reduce the computation complexity of DTCWT processor, a modified architecture is proposed in the next section.

DTCWT architecture: The DTCWT filter coefficients shown in Table 1 and 2 is represented in Table 3. The similarities in filter coefficients are indicated as h_x and g_x

Table 3: Reduced filter coefficients with common terms

Stage 1			Stage 2					
No.	LP	HP	LP	HP	LP	HP	LP	HP
0	h_0	h_0	h_3	h_0	g_0	g_1	g_1	$-g_0$
1	$-h_1$	$-h_3$	h_3	h_0	g_1	g_1	g_1	g_1
2	h_1	h_3	$-h_1$	$-h_1$	$-g_2$	$-g_6$	$-g_6$	g_2
3	h_2	h_1	h_1	$-h_1$	g_3	g_1	g_1	g_3
4	h_2	h_1	h_2	h_2	g_4	g_5	g_5	$-g_4$
5	h_1	$-h_2$	h_2	$-h_2$	g_5	$-g_4$	g_4	g_5
6	$-h_1$	h_2	h_1	h_1	g_1	g_3	g_3	g_1
7	h_3	$-h_1$	$-h_1$	h_1	$-g_6$	g_2	$-g_2$	$-g_6$
8	h_3	$-h_1$	h_0	h_3	g_1	g_1	g_1	g_1
9	h_0	h_0	h_0	$-h_3$	g_1	$-g_0$	g_0	g_1

terms. Considering the common terms the number of coefficients for level 1 and 2 filters are $\{h_0, h_1, h_2, h_3\}$ and $\{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$.

The responses of four filters denoted as $\{Y_{LR}, Y_{HR}, Y_{LI}, Y_{HI}\}$ for row processing are represented as Eq. 7-20, the responses are obtained by considering the similarities in filter coefficients, x_i represents the input samples:

$$a_{i+n} = X_{i+1+n} + X_{i+2+n} \quad n = 0, 1, 2, 3 \tag{7}$$

$$S_i^1 = h_1 a_{i+n} + h_2 a_{i+n+1} \tag{8}$$

$$S_i^2 = h_1 a_{i+n} + h_3 a_{i+n+1} \tag{9}$$

$$S_i^3 = h_3 a_{i+n} + h_1 a_{i+n+1} \tag{10}$$

$$S_i^4 = h_2 a_{i+n} + h_1 a_{i+n+1} \tag{11}$$

$$Y_{LR} = S_i^1 + S_i^2 \tag{12}$$

$$Y_{HR} = S_i^3 + S_i^4 \tag{13}$$

$$b_{i+n} = X_{i+1+n} + X_{i+2+n} \quad n = 0, 1, 2, 3 \tag{14}$$

$$d_i^1 = h_1 b_{i+n} + h_2 b_{i+n+1}, \; n = 0 \tag{15}$$

$$d_i^2 = h_1 b_{i+n} + h_3 b_{i+n+1}, \; n = 2 \tag{16}$$

$$d_i^3 = h_3 b_{i+n} + h_1 b_{i+n+1}, \; n = 0 \tag{17}$$

$$d_i^4 = h_2 b_{i+n} + h_1 b_{i+n+1}, \; n = 2 \tag{18}$$

$$Y_{LI} = d_i^3 + d_1^4 \tag{19}$$

$$Y_{HI} = d_i^1 + d_i^2 \tag{20}$$

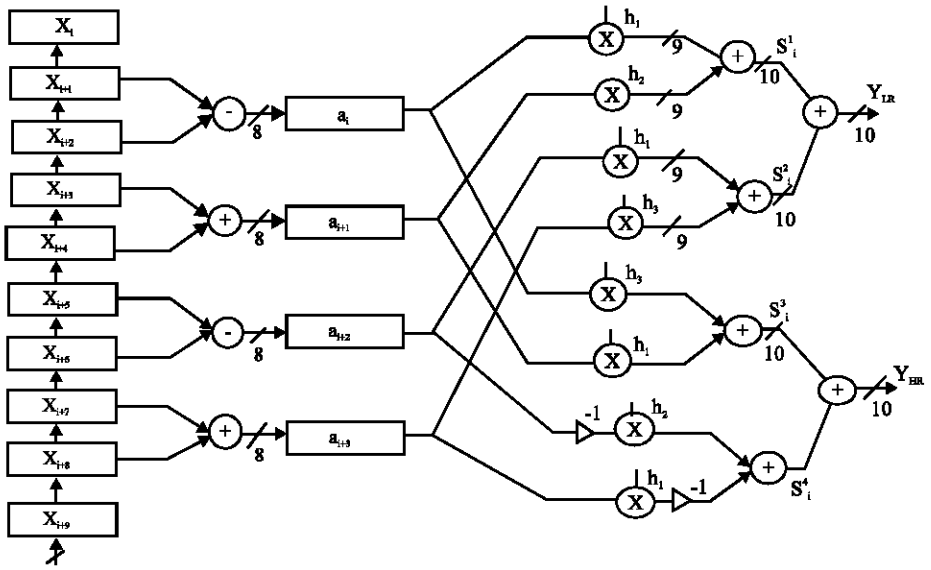


Fig. 2: Reordered pipeline filter architecture (real)

Table 4: Comparison of hardware requirements

	1st stage row processing (1st Stage column processing)	
Parameters	Modified architecture with pipeline	Generic DTCWT architecture
Multipliers	16 (64)	40 (160)
Adders	24 (96)	36 (144)
Throughput	1T (1T)	4T (4T)
Latency	14T (24T+2N)	24T (48T+2N)
Pipeline stage	4	Nil
Intermediate memory	36 (144)	20 (80)

From the output terms $\{Y_{LR}, Y_{HR}\}$, there are common terms such as $\{(X_{i+2}-X_{i+1}), (X_{i+3}+X_{i+4}), (X_{i+5}-X_{i+6})$ and $(X_{i+7}+X_{i+8})\}$, thus, the number of arithmetic operations can be reduced. The reduced filter structure for computation of $\{Y_{LR}, Y_{HR}\}$ is shown in Fig. 2. The input data is loaded into 10 input registers and requires 10T clocks, the registers contents are accumulated as the coefficients are similar in 1T clock period. The intermediate register store the accumulated sum in four registers $\{a_i, a_{i+1}, a_{i+2}$ and $a_{i+3}\}$. The multiplication array consisting of eight multipliers are used to multiply the accumulated data with the corresponding coefficients which is designed to perform the operation in 1T clock period. The succeeding arrays of adders accumulate the partial products with two stage addition to compute Y_{LR} and Y_{HR}, simultaneously with 2T clock periods. Thus, after 14T clock periods the first output samples are computed. The architecture shown in Fig. 2 is designed with pipeline approach with four stages of pipeline. The pipeline stage has a latency of 14T and throughput of 1T. The pipeline architecture requires 10 input registers, 14 intermediate registers and 2 output registers as there are 4 adders, 8 multipliers and 4 adders in the intermediate stages. Similarly for computation of $\{Y_{LI}, Y_{HI}\}$ output terms in the stage 1 filter, the common terms in filter coefficients are reorganized to minimize the number of arithmetic operations as in Eq. 7-20. The reduced architecture is shown in Fig. 2.

The reduced architecture consists of four stage pipeline scheme. The latency is 14 T and throughput is 1T. For level 1 row filtering the number of arithmetic operations and timing parameters are summarized in Table 4.

The level 2 processing is performed on the four low pass sub-bands (approximation coefficients), each of size N/2×N/2. The level 2 comprises of stage 1 row processing and stage 2 column processing which are realized using the pipelined structure as shown in Fig. 3. The common terms in filter coefficients are simplified into coefficients represented as $\{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$. The reduced output expressions are given in as in Eq. 6 and the corresponding reduced architecture is shown in Fig. 4.

The reduced architecture for row processing comprises of 10 input registers, into which the input data is first loaded. The data path control unit reads corresponding elements into another register array represented by c and d. The contents of registers c and d are correspondingly multiplied by the filter coefficients and accumulated in two stage adder array. A last stage accumulator unit designed as butterfly structure generates the real and imaginary components. The total number of multiplier and adder units is designed to be 12 and 12, respectively. Similarly, the column processor

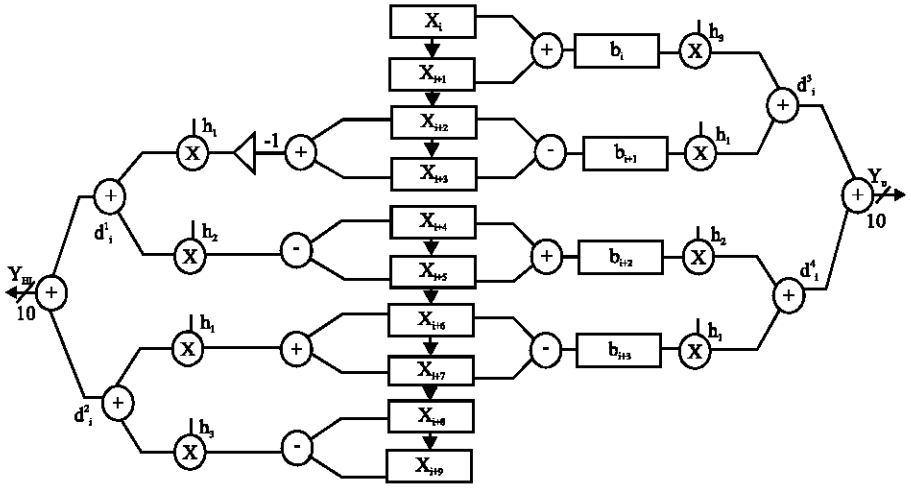


Fig. 3: Reordered pipeline filter architecture (imaginary)

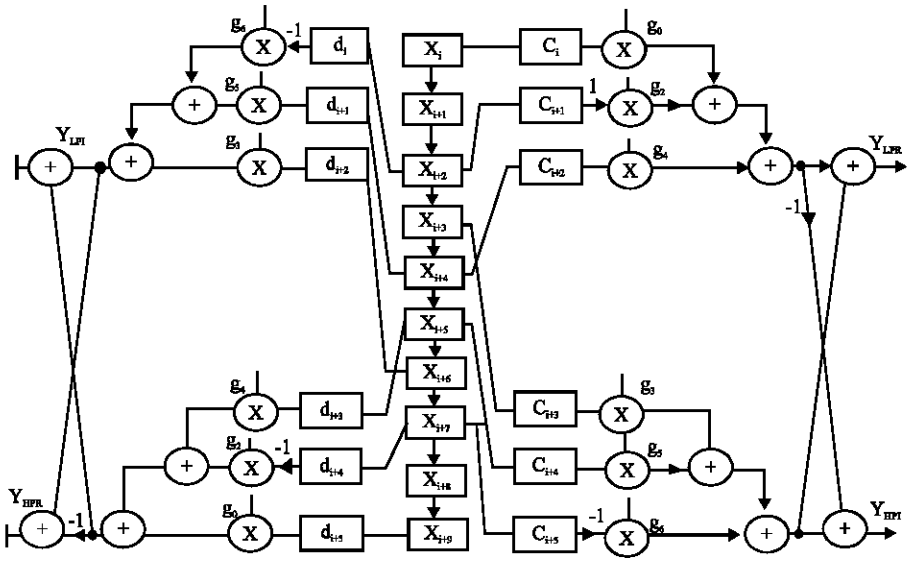


Fig. 4: Optimal pipelined filter architecture

architecture is also designed as per the discussion presented above. In summary, the pipelined architecture for level 1 row processing and level 2 row processing is represented in Eq. 21-24:

$$Y_{LPR} = X_i g_0 - X_{i+2} g_2 + X_{i+4} g_4 + X_{i+3} g_3 + X_{i+5} g_5 - X_{i+7} g_6 \quad (21)$$

$$Y_{HPI} = -(X_i g_6 - X_{i+2} g_2 + X_{i+4} g_4 + X_{i+3} g_3 + X_{i+5} g_5 - X_{i+7} g_6 \quad (22)$$

$$Y_{HPR} = -X_{i+2} g_6 + X_{i+4} g_5 + X_{i+6} g_3 - (X_{i+5} g_4 - X_{i+7} g_2 + X_{i+9} g_0) \quad (23)$$

$$Y_{LPI} = -X_{i+2} g_6 + X_{i+4} g_5 + X_{i+6} g_3 + X_{i+5} g_4 - X_{i+7} g_2 + X_{i+9} g_0 \quad (24)$$

The 1D DTCWT real and imaginary output computation logic presented in Eq. 23 and 24 consists of four stages. The first and the last stage requires adders, second and third stage require multiply and accumulate units.

Area efficient architecture: The top level architecture of 2D-DTCWT for image decomposition is shown in Fig. 5. The architecture consists of row processor, column processor and an intermediate matrix for transformation. The input multiplexer decides the data that can enter into

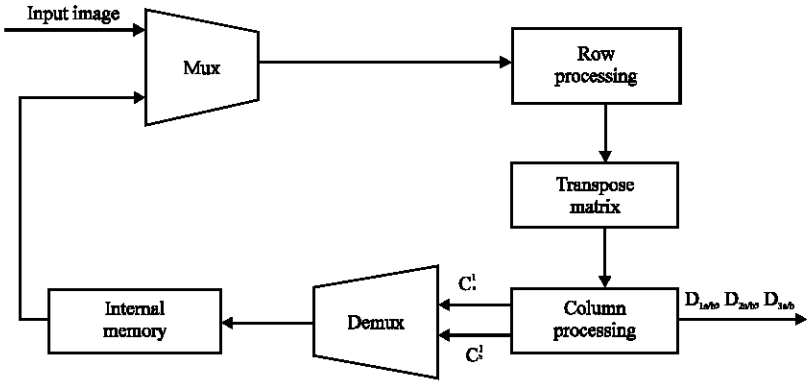


Fig. 5: Top level 2D-DTCWT architecture

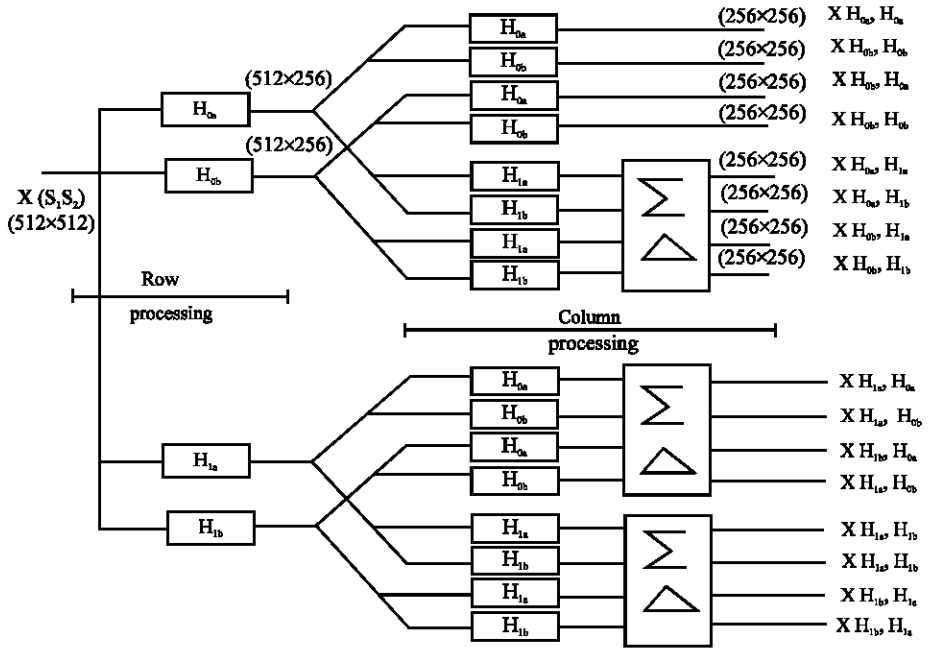


Fig. 6: 2D DTCWT architecture

the module. For first level decomposition the raw input image is selected, the DTCWT processor generates twelve sub bands, the low sub band is selected with the multiplexer at the output and the selected low sub band is stored in the intermediate output memory. The input memory and transpose memory are of size N×N, the output intermediate memory is of size N/2×N/2. The architecture is designed for one-level decomposition. The high pass sub bands are stored in external intermediate memory.

For more than one level decomposition, the proposed architecture need to be modified by including a demultiplexer at the output stage that can store data into multiple output memories. The internal architecture of DTCWT processing is shown in Fig. 6.

The input image of size 512×512 is processed with four filter banks $\{H_{0a}, H_{0b}, H_{1a}, H_{1b}\}$, the four outputs after processing are of size 512×256. The input image is read row wise and the row processing filters compute the sub-bands, for each row of the input 4 outputs are generated, each of size 512×256. The column processor processes the row outputs simultaneously with four groups of filter bank. The column processor consists of sixteen filters with each filter comprising of 10 coefficients. With each filter requiring 10 multipliers and 9 adders, the total number of multipliers and adders for column processing will be 160 and 54. In order to reduce the number of arithmetic units, a modified architecture is proposed and is presented in Fig. 7. In the modified architecture, the 4 outputs of row processed data is

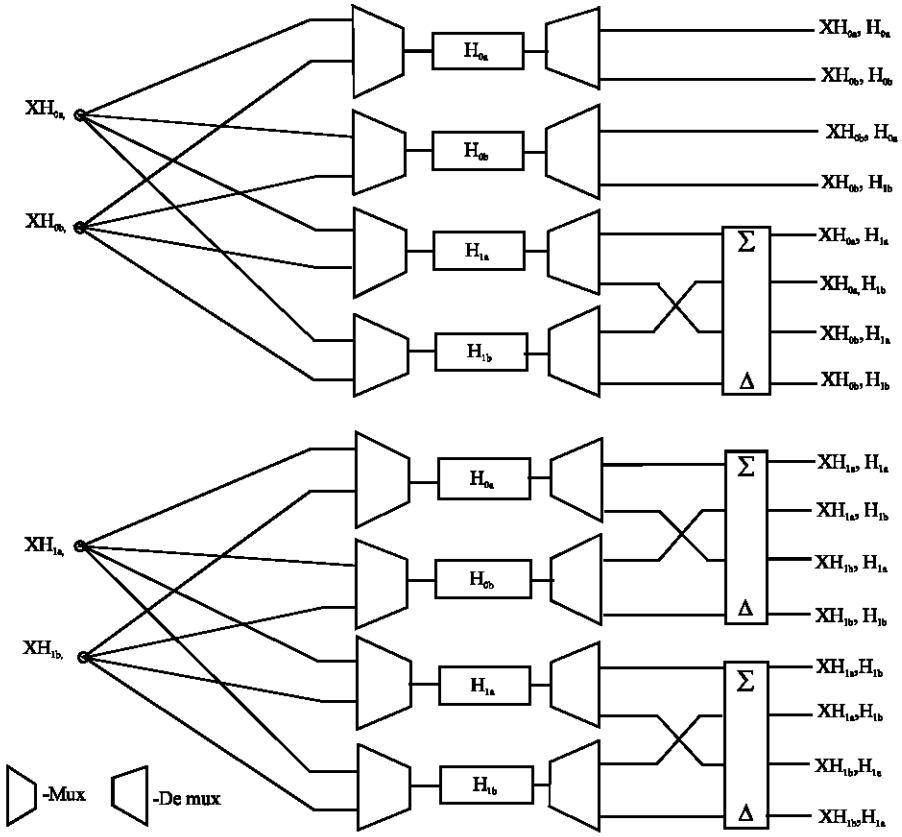


Fig. 7: Reduced column processor method 1

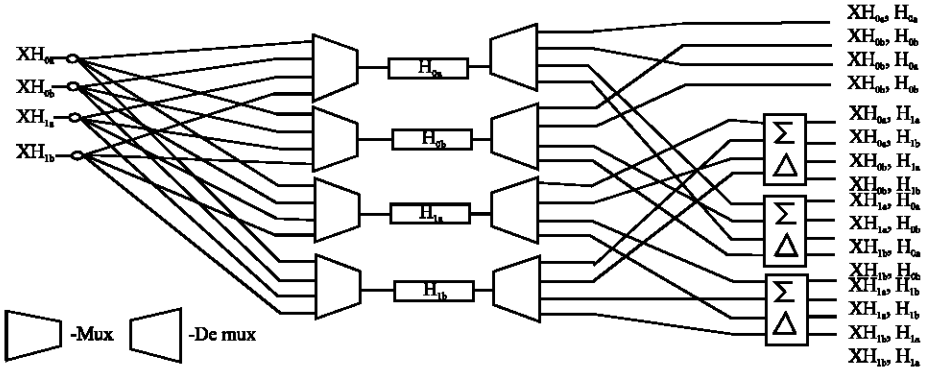


Fig. 8: reduced column processor method 2

processed with eight filters by multiplexing the input samples. At the output of filters, the demultiplexing unit stores the column processed output into corresponding memory elements. The numbers of filters in the modified scheme are reduced by 50% compared with the architecture shown in Fig. 6.

With multiplexing of input data the hardware complexity is reduced, however, there is increase in delay (latency) by two times. In order to reduce the delay, the multiplexers and filters can be designed to operate on both edges of the clock.

Figure 8 shows an further improved version of DWTCWT column processing module. In this architecture, the redundancy in filter banks (Fig. 6) are minimized by processing the row outputs using four filters and with multiplexer-demultiplexer logic. Thus,

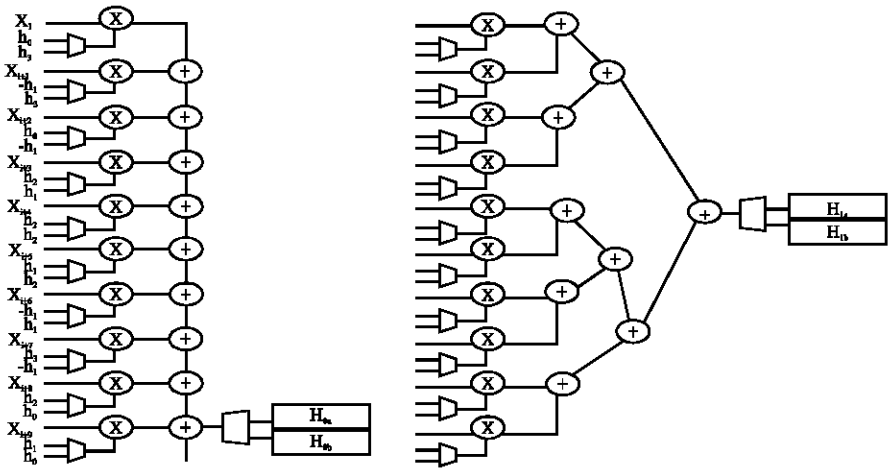


Fig. 9: Optimal multiplexed row processor

there is a reduction of 75% in area and increase in delay. The modified column processors designed are area efficient.

In order to further reduce hardware complexity, the row processor unit is modified by considering common terms in the filter coefficients. The row processor unit shown in Fig. 6, consists of four filters. The H_{oa} and H_{ob} pair and H_{1a} and H_{1b} pair are realized with the architecture shown in Fig. 9.

The architecture is designed with ten multipliers and nine adders. Each multiplier is loaded with H_{oa} coefficient at time t and H_{ob} coefficient at time t+1. The input remains common in both time intervals, the multiplied data is accumulated in the adder array and the demultiplexer stores the output in the corresponding memory according to the processing time. The data flow is controlled using a state machine that sequences the control to the multiplexer and demultiplexer. The row processor is designed to operate with 50% reduction in area. The proposed DTCWT architecture is modeled using Verilog HDL and is synthesized using Xilinx ISE targeting Virtex 5 FPGA. The design is functionally verified for an input image of size 16×16, the results are verified and compared for its logic correctness.

RESULTS AND DISCUSSION

Figure 10 shows the block diagram of the top level architecture for 2D DTCWT processor. The basic 2D DWT architecture for image decomposition as presented by Cheng and Parhi (2008), Mohanty and Meher (2013), Zhang *et al.* (2012), Darji *et al.* (2014) and Hu and Jong (2013) is modified and 2D DTCWT architecture is derived. The 2D DWT architecture consists of input memory, intermediate memory, output memory and DTCWT processor. The memory controller reads the individual frame that are stored in the input memory and the processed data is further stored in the output memory. The memory controller controls the address generation logic to read corresponding data from the input memory and store data in the corresponding location in the output memory.

The design is synthesized in the Xilinx tool, post place, map and route simulation also have been carried out. The synthesized netlist of DTCWT is shown in Fig. 11. The design is optimized for power, area and timing. RTL synthesized block diagram of DTCWT is obtained using Xilinx ISE and the synthesis report is analyzed for estimation the performances of the proposed design in terms of area, speed and power. FPGA implementation results of DTCWT and Inverse DTCWT are shown in Table 5 and 6.

For DTCWT computations out of 19,200 registers available 2% is the utilization. In case of inverse DTCWT utilization is of 10%. Table 6 presents the power report of DTCWT and Inverse DTCWT.

The maximum operating frequency of the proposed design on Virtex-5 pro FPGA is found to be 268 MHz for DTCWT and 156 MHz for inverse DTCWT. The operating frequency can be further improved by utilizing the internal resources such as multipliers and adders on Virtex 5 platform. Parallel modulation operations combined with pipelining of DTCWT and Inverse DTCWT computation can improve the operating speed at the cost of area.

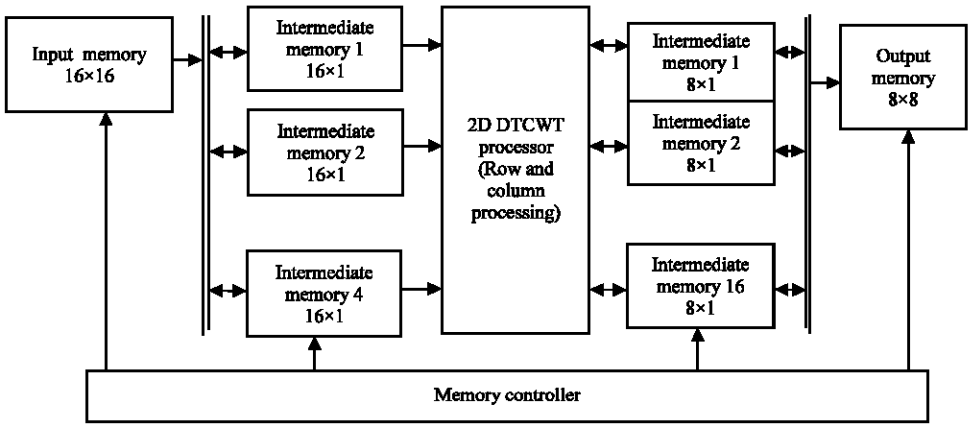
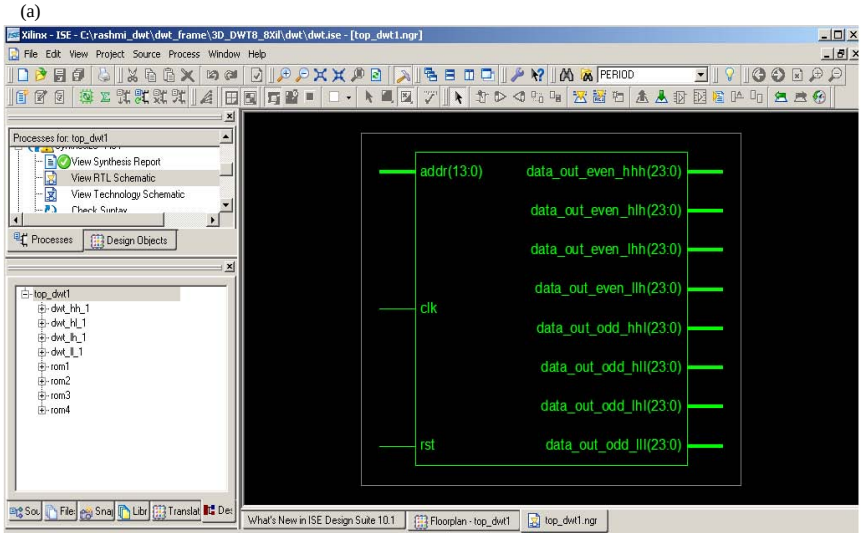


Fig. 10: Top level architecture of 2D DTCWT processor

(a)



(b)

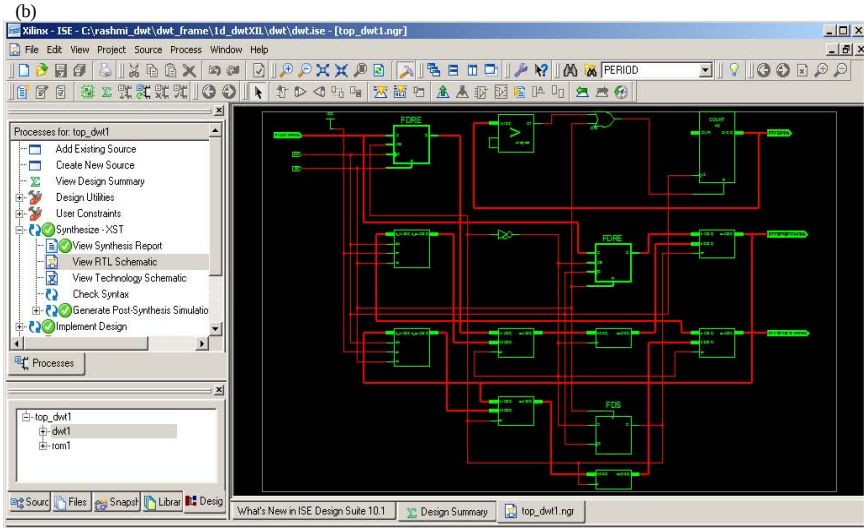


Fig. 11: Snapshot of DTCWT Synthesized netlist; a) Top module; b) Internal architecture

CONCLUSION

In this study, we have proposed hardware efficient and high throughput architecture for DTCWT computation. The proposed architecture is optimized to realize minimum number of multipliers and adders by minimizing the number of filters coefficients by rearrangement. The row filters are designed with pipelined structure and the column filters are designed with reordered structure. The row processing and column processing units are designed to operate in pipelined process improving throughput. The proposed architecture is implemented on Virtex 5 FPGA using Xilinx ISE. The designed architecture is faster, area efficient and achieves higher throughput as similar to DWT processing modules. The intermediate memory stages can be further optimized with memory reuse techniques. The proposed architecture can process 1024×1024 frame image in <40 msec by consuming <3 W of power.

REFERENCES

Cheng, C. and K.K. Parhi, 2008. High-speed VLSI implementation of 2-D discrete wavelet transform. IEEE Trans. Signal Process., 56: 393-403.

Darji, A., S. Agrawal, A. Oza, V. Sinha, A. Verma, S.N. Merchant and A.N. Chandorkar, 2014. Dual-scan parallel flipping architecture for a lifting-based 2-D discrete wavelet transform. IEEE Trans. Circuits Syst. II: Express Briefs, 61: 433-437.

Fang, L.H., M.G. Feng and X.H. Jie, 2010. Images compression using dual tree complex wavelet transform. Proceedings of the 2010 International Conference on Information Science and Management Engineering, August 7-8, 2010, IEEE, Xi'an, China, ISBN:978-1-4244-7670-1, pp: 559-562.

Hu, Y. and C.C. Jong, 2013. A memory-efficient high-throughput architecture for lifting-based multi-level 2-D DWT. IEEE. Trans. Signal Process., 61: 4975-4987.

Kingsbury, N., 2000. A dual-tree complex wavelet transform with improved or thogonality and symmetry properties. Proceedings of the International Conference on Image Processing Vol. 2, September 10-13, 2000, IEEE, Vancouver, British Columbia, ISBN:0-7803-6297-7, pp: 375-378.

Kingsbury, N.G., 2001. Complex wavelets for shift invariant analysis and filtering of signals. J. Applied Comput. Harmonic Anal., 10: 234-253.

Mohanty, B.K. and P.K. Meher, 2013. Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT. IEEE. Trans. Circ. Syst. Video Technol., 23: 353-363.

Neumann, J. and G. Steidl, 2005. Dual-tree complex wavelet transform in the frequency domain and an application to signal classification. Intl. J. Wavelets Multiresolution Inf. Process., 3: 43-65.

Pastuszak, G. and A. Abramowski, 2016. Algorithm and architecture design of the H. 265/HEVC intra encoder. IEEE. Trans. Circuits Syst. Video Technol., 26: 210-222.

Sullivan, G.J., J. Ohm, W.J. Han and T. Wiegand, 2012. Overview of the High Efficiency Video Coding (HEVC) standard. IEEE. Trans. Circuits Syst. Video Technol., 22: 1649-1668.

Taubman, D. and M. Marcellin, 2002. JPEG: 2000 Image Compression Fundamentals, Standards and Practice. Springer, Berlin, Germany, ISBN:978-1-4613-5245-7, Pages: 777.

Taubman, D., 2000. High performance scalable image compression with EBCOT. IEEE Trans. Image Process., 9: 1158-1170.

Xiong, Z., X. Wu, S. Cheng and J. Hua, 2003. Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms. IEEE Trans. Med. Imaging, 22: 459-470.

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.