2-D Separable Discrete Hartley Transform Architecture for Efficient FPGA Resource

Vijay Kumar Sharma, Richa Agrawal, U. C. Pati, K. K. Mahapatra[#]
Dept. of Electronics & Comm. Engg.
National Institute of Technology, Rourkela, India-769008
#kmaha2@gmail.com

Abstract— **Discrete cosine transform (DCT) is usually used in JPEG based image transform coding. This paper presents separable 2-D discrete Hartley transform (SDHT) and its Distributed Arithmetic (DA) based hardware architecture as an alternate to DCT in transform based coding of image compression. The proposed DA architecture for 1-D DHT has very less computations as compared to existing 1-D DCT. The proposed DHT architecture implemented in FPGA indicates a significant hardware savings as compared to FPGA resources used in an efficient memory based DA approach. The additional advantage of SDHT is that its inverse transform is same as forward transform with a constant division. This is demonstrated through a Xilinx FPGA XC2VP30 device.**

Keywords*— *Distributed Arithmetic, Discrete Hartley Transform, Discrete Cosine Transform, JPEG, Offset Binary Coding.

I. Introduction

Digital image compression is the most important part in the multimedia applications which aims to reduce the number of bits in an image data for its efficient storage (less storage area). JPEG based still image compression follows three steps i.e. transform, quantization and coding to compress an image [1]. Reverse process comprising decoding, dequantization and inverse transform is used for image de-compression [2,3]. Discrete cosine transform (DCT) is used to transform the image from spatial domain to frequency domain. During quantization less important frequencies are discarded and is termed as lossy image compression. Bracewell has drawn attention to the discrete Hartley transform (DHT) as a substitute for the discrete fourier transform (DFT) [4,5]. Many applications of DHT in signal processing and communications have been presented in the literatures [6-8]. DHT is used in JPEG based image compression with DHT replacing the DCT in [9]. PSNR comparable to DCT based image transform has been obtained. The advantage of using DHT over DCT is that a dequantizer is not required at the decoder side and hence facilitates saving of hardware resources.

Hardware implementation of DHT requires a large number of multipliers. Since multiplier requires much more hardware as compared to adder and additionally consumes more power, it is not recommended in a battery running portable device [10,11]. Distributed Arithmetic (DA) is a digital signal processing technique that implements multiplication without the use of multiplier. Two approaches have been emerged for DA. One is look–up table based [12] and other is without look-up table [13]. Field programmable gate array (FPGA) is an emerging technology in VLSI design because of its many advantages like reprogramability, low cost, fast design cycle, in-circuit programmability etc [14-18].

In this paper separable 2-D DHT (SDHT) is proposed in place of DCT for image transform. The image transformed by SDHT is reconstructed after taking inverse SDHT (same as forward with a constant division). No error in the reconstructed image indicates that SDHT can be used for image transform [19-23]. SDHT is implemented in Xilinx FPGA XC2VP30 device using memory based DA and proposed DA for DHT. Proposed DA for DHT requires very less hardware for its implementation as compared to memory based DA [24-30]. It also requires less computation as compared to DCT in [13].

The organization of this paper is as follows. Section II contains the introduction of separable 2-D DHT. Use of SDHT in place of DCT and image reconstruction by taking inverse SDHT is explored in section III. Look-up table based DA for 1-D DHT and its hardware implementation is explained in section IV. Section V describes the proposed DA for 1-D DHT. Section VI discusses the FPGA implementation results and comparisons in terms of hardware utilization. Finally conclusions are drawn.

II. Separable 2-D DHT

Separable 2-D DHT is given by the equation,

$$Y(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\mathrm{f(x,y)cas}\left(\frac{2\pi ux}{N}\right)\mathrm{cas}\left(\frac{2\pi vy}{M}\right) \tag{1}$$

and the 1-D DHT is given by the equation,

$$Y(k)=\sum_{n=0}^{N-1}\mathrm{X(n)cas}\left(\frac{2\pi}{N}nk\right)\quad k=0,1,...N-1 \tag{2}$$

where $\mathrm{H_{nk}}=\mathrm{cas}\left(\frac{2\pi}{N}\mathrm{nk}\right)$, is the transform's Kernel
$\mathrm{cas}(x)=\cos(x)+\sin(x)$ and
X(n) is input data.

The advantage of using separable 2-D DHT equation (1) is that it can be implemented with 1-D DHT given by equation (2) by row-column decomposition method. Non

978-1-4244-9032-5/10/$26.00©2010 IEEE

separable 2-D DHT equation requires further calculations after taking row-column decomposition [4].

III. IMAGE TRANSFORM AND RECONSTRUCTION FROM SDHT

Fig.1 shows the JPEG baseline encoder which uses DCT based transform for image compression. Fig.2 shows the DHT based encoder. Fig.3a shows the image used for DHT transform by 8x8 block using SDHT. The reconstructed image after taking inverse SDHT as shown in Fig.3 (b) with almost no distortion proves that SDHT can be used in place of DCT for image transform. Energy calculation and thresholding is used in DHT based encoder instead of quantization table in DCT based encoder.

IV. LOOK-UP TABLE BASED DA FOR 1-D DHT

Representing X(n) in 2's complement fractional form

$$X_n = -x_{n,i-1} + \sum_{m=1}^{i-1} x_{n,i-1-m} 2^{-m} \qquad (3)$$

where $x_{n,m}$ is m[th] bit from LSB and m=i-1 is sign bit

Negative of X_n is represented by

$$-X_n = -\bar{x}_{n,i-1} + \sum_{m=1}^{i-1} \bar{x}_{n,i-1-m} 2^{-m} + 2^{-(i-1)} \qquad (4)$$

Representing X_n in Offset binary Code (OBC) we have

$$X_n = \frac{1}{2}[X_n - (-X_n)] =$$

$$\frac{1}{2}\left[-(x_{n,i-1} - \bar{x}_{n,i-1}) + \sum_{m=1}^{i-1} (x_{n,i-1-m} - \bar{x}_{n,i-1-m}) 2^{-m} - 2^{-(i-1)}\right] \qquad (5)$$

$$\text{Define, } d_{n,m} = \begin{cases} x_{n,m} - \bar{x}_{n,m}, & for\ m \neq i-1 \\ -(x_{n,i-1} - \bar{x}_{n,i-1}), & for\ m = i-1 \end{cases} \qquad (6)$$

and $d_{n,m} \in (-1,1)$

Now from Equations (5) and (6),

$$X_n = \frac{1}{2}\left[\sum_{m=0}^{i-1} d_{n,i-1-m} 2^{-m} - 2^{-(i-1)}\right] \qquad (7)$$

From Equations (2) and (7),

$$Y(k) = \sum_{n=0}^{N-1} \frac{1}{2} H_{nk} \left[\sum_{m=0}^{i-1} d_{n,i-1-m} 2^{-m} - 2^{-(i-1)}\right]$$

$$= \sum_{m=0}^{i-1} \left(\sum_{n=0}^{N-1} \frac{1}{2} H_{nk} d_{n,i-1-m}\right) 2^{-m} - \left(\sum_{n=0}^{N-1} \frac{1}{2} H_{nk}\right) 2^{-(i-1)} \qquad (8)$$

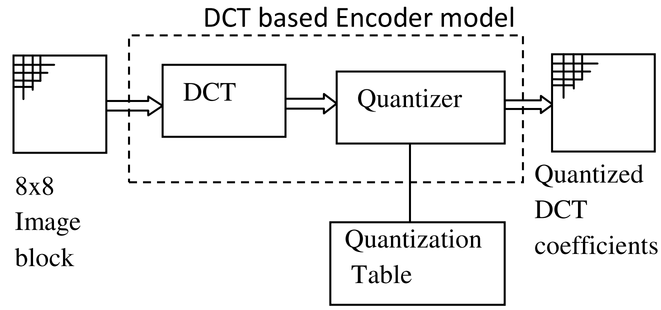

Fig.1 JPEG baseline encoder model

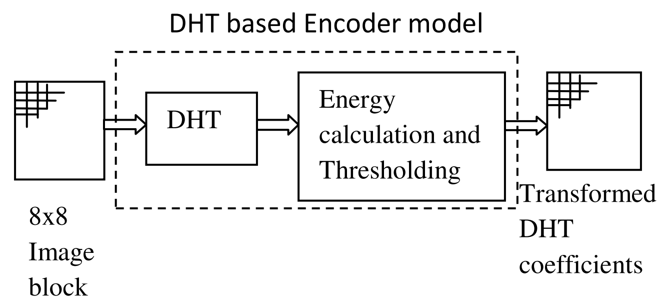

Fig.2 DHT based encoder model with energy quantization.

(a) (b)

Fig.3 (a) Original image (b) Reconstructed image after inverse SDHT

From equation (8), Y(k) can be computed for different values of k, let for example k=1 the summation $\sum_{n=0}^{N-1} \frac{1}{2} H_{n1} d_{n,i-1-m}$ can be stored in look-up table (ROM) after pre-additions as there can be a maximum of 2^n possible combinations as shown in Table 1 for n=3. Inputs are transmitted in serial, 1 bit at a time. One bit of each input forms the address of memory location and they are accumulated after shifting left the previous result. Second accumulation term is a constant value and it is similar to the first row in table and hence there is no need of additional memory for it. Final output is obtained after i+1 clock cycle where i is number of bits in input (1 clock cycle for adding constant value). From table 1 it can be seen that upper half is same as lower half except sign is opposite. Hence look-up table can be halved as only upper half is stored and bits of X_1 is X-ORed with all other inputs so that when it is 0 upper half is addressed and accumulated with no sign change but when it is 1 again upper half is addressed and accumulated with sign reversed.

TABLE 1

Pre-addition ROM content, for number of inputs n=3 and k=1

$X_{1,m}$	$X_{2,m}$	$X_{3,m}$	values to be stored in ROM
0	0	0	$-(H_{01}+H_{11}+H_{21})/2$
0	0	1	$-(H_{01}+H_{11}-H_{21})/2$
0	1	0	$-(H_{01}-H_{11}+H_{21})/2$
0	1	1	$-(H_{01}-H_{11}-H_{21})/2$
1	0	0	$(H_{01}-H_{11}-H_{21})/2$
1	0	1	$(H_{01}-H_{11}+H_{21})/2$
1	1	0	$(H_{01}+H_{11}-H_{21})/2$
1	1	1	$(H_{01}+H_{11}+H_{21})/2$

V. PROPOSED DA FOR 1-D DHT

Considering the periodicity and symmetry of trigonometric functions equation (1) can be written as,

$$y(0)=[x(0)+x(4)+x(1)+x(5)+x(2)+x(6)+x(3)+x(7)]A$$
$$y(1)=[x(1)-x(5)]C+[x(2)-x(6)]B+[x(0)-x(4)]A$$
$$y(2)=[x(2)+x(6)](-A)+[x(1)+x(5)]B+[x(3)+x(7)](-B)+[x(0)+x(4)]A$$
$$y(3)=[x(2)-x(6)](-B)+[x(3)-x(7)]C+[x(0)-x(4)]A$$
$$y(4)=[x(1)+x(5)](-A)+[x(2)+x(6)]A+[x(3)+x(7)](-A)+[x(0)+x(4)]A$$
$$y(5)=[x(1)-x(5)](-C)+[x(2)-x(6)]B+[x(0)-x(4)]A$$
$$y(6)=[x(0)+x(4)]A+[x(1)+x(5)](-B)+[x(2)+x(6)](-A)+[x(3)+x(7)]B$$
$$y(7)=[x(2)-x(6)](-B)+[x(3)-x(7)](-C)+[x(0)-x(4)]A$$

where,

$$A=\mathrm{cas}(0),\quad B=\mathrm{cas}(\tfrac{\pi}{2}),\quad and\ C=\mathrm{cas}(\tfrac{\pi}{4})$$

Representing in DA form as in [13] for DCT we get Adder/Subtractor matrix for DHT for all data as in Fig.4. Table 2 shows the explanation of Fig.3. Divide and multiply operations are done by shifting. Y^n implies shifting n bits. Negative sign in n implies left shift where as positive sign implies right shift. Positive sign in table 2 implies ALUs perform addition and negative sign implies subtraction operation. As an example, let's take the fourth column for calculating Y(2). Y(2) is the sum of $Y^{-1}(2)*2$, $Y^0(2)$, $Y^1(2)/2$, $Y^2(2)/2^2$, $Y^3(2)/2^3$, $Y^4(2)/2^4$, $Y^5(2)/2^5$, $Y^6(2)/2^6$, and $Y^7(2)/2^7$. The values of $Y^{-1}(2)$, $Y^0(2)$, $Y^1(2)$, $Y^2(2)$, $Y^3(2)$, $Y^4(2)$, $Y^5(2)$, $Y^6(2)$, $Y^7(2)$ can be got from R3, R5, 0, 0, 0, 0, 0, 0, 0. So Y(2) can be calculated as,

$$Y(2)=R3*2+R5$$

Compared to DCT adder/subtractor of [13], DHT adder/subtractor requires less no. of ALU + adders (only 4 ALUs+7 adders in proposed DHT but 9 ALUs +6 adders in DCT).

VI. FPGA IMPLEMENTATION RESULTS AND COMPARISONS

Both memory based and proposed architecture for 1-D DHT is implemented in Xilinx XC2VP30 FPGA device and table 4 and 5 shows the hardware utilization summary which indicate considerable hardware savings in proposed DA for DHT as compared to memory based DA. 2-D DHT is implemented using row column decomposition method

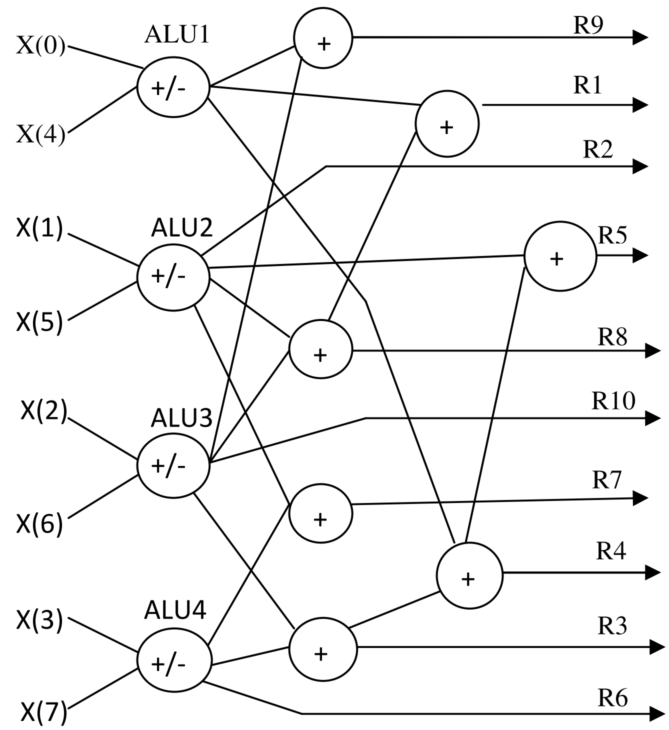

Fig.4 Proposed adder/subtractor matrix of all data for DHT

TABLE 2

Description of additions and subtractions of Fig.4

	Y(0)	Y(1)	Y(2)	Y(3)	Y(4)	Y(5)	Y(6)	Y(7)
ALU1	+	-	+	-	+	-	+	-
ALU2	+	-	+	NO	+	-	+	NO
ALU3	+	-	+	-	+	-	+	-
ALU4	+	NO	+	-	+	NO	+	-
Y^{-1}	0	0	R3	R10	R7	R2	R8	R3
Y^0	R5	R1	R5	R4	R5	R9	R5	R9
Y^1	0	0	0	0	0	R2	0	R6
Y^2	0	R2	0	R6	0	0	0	0
Y^3	0	R2	0	R6	0	0	0	0
Y^4	0	0	0	0	0	R2	0	R6
Y^5	0	R2	0	R6	0	0	0	0
Y^6	0	0	0	0	0	R2	0	R6
Y^7	0	R2	0	R6	0	R2	0	R6

"+" means addition and "-" means subtraction and "NO" means no operation.

TABLE 3

Comparison of adders of DHT and DCT in [13]

scheme	Adder matrix	Adder bit-width
DCT	9 ALU +6	850
Proposed DHT	4 ALU +7	452

and table 6 shows the hardware utilization summary.

8x8 image data of table 7 is implemented and table 8 shows the matlab simulation result. Fig.5 shows the hardware implementation result using memory based DA method and fig.6 shows the hardware implementation result of proposed DA for DHT method. It is evident that proposed DA method implementation has less error as compared to memory based DA. Floating point representation is required to store pre-computed values in ROM for accuracy and hence floating point circuitry has to be used which will further increase the hardware resource in memory based DA. Also memory based DA is slow as it requires i+1 clock cycles (i is number of bits in input image data of one pixel) for 1-D DHT and subsequently 8*((i+1)+(i+1+2)) clock cycles (generally input image pixel is represented by 8-bit, i.e i=8, and hence total clock latency will be 160 clock cycle) for 2-D DHT implementation (after 1-D transform, transformed data representation become 2-bit more than the input representation) where as proposed DA for DHT requires 1 clock cycle for 1-D DHT and 2-D DHT is implemented in total of 8 clock cycles with 50MHz clock. Although memory based DA work with faster clock as it has lesser calculations per clock cycle to perform than proposed DA for DHT.

TABLE 4

Hardware utilization for memory based DA for 1-D DHT

Logic Utilization	Used	Available	Utilization
Number of Slices	561	13696	4%
Number of 4 input LUTs	998	27392	3%

TABLE 5

Hardware utilization for proposed DA for 1-D DHT

Logic Utilization	Used	Available	Utilization
Number of Slices	309	13696	2%
Number of 4 input LUTs	562	27392	2%

TABLE 6

Hardware utilization for proposed DA for 2-D DHT

Logic Utilization	Used	Available	Utilization
Number of Slices	1295	13696	9%
Number of Slice Flip Flops	899	27392	3%
Number of 4 input LUTs	2319	27392	8%

TABLE 7

8x8 block image data to be implemented

30	29	39	42	32	36	46	39
33	34	37	36	36	42	43	33
37	40	34	32	41	45	38	32
40	43	35	36	45	40	35	47
40	42	38	42	46	35	43	75
40	40	41	43	41	40	65	102
40	40	41	36	35	58	95	117
41	41	40	27	30	75	117	121

TABLE 8

Matlab simulation result for 8x8 image data in table 7

2934	-269	-360	-212	-146	-94	0	556
-397	48	260	208	128	83	-143	-370
-214	139	24	29	26	8	120	-253
-147	83	41	36	22	11	55	-170
-104	50	34	25	16	5	34	-117
-64	35	17	17	9	6	21	-66
0	0	6	5	0	0	-2	0
336	-238	-335	-33	-48	-13	-141	426

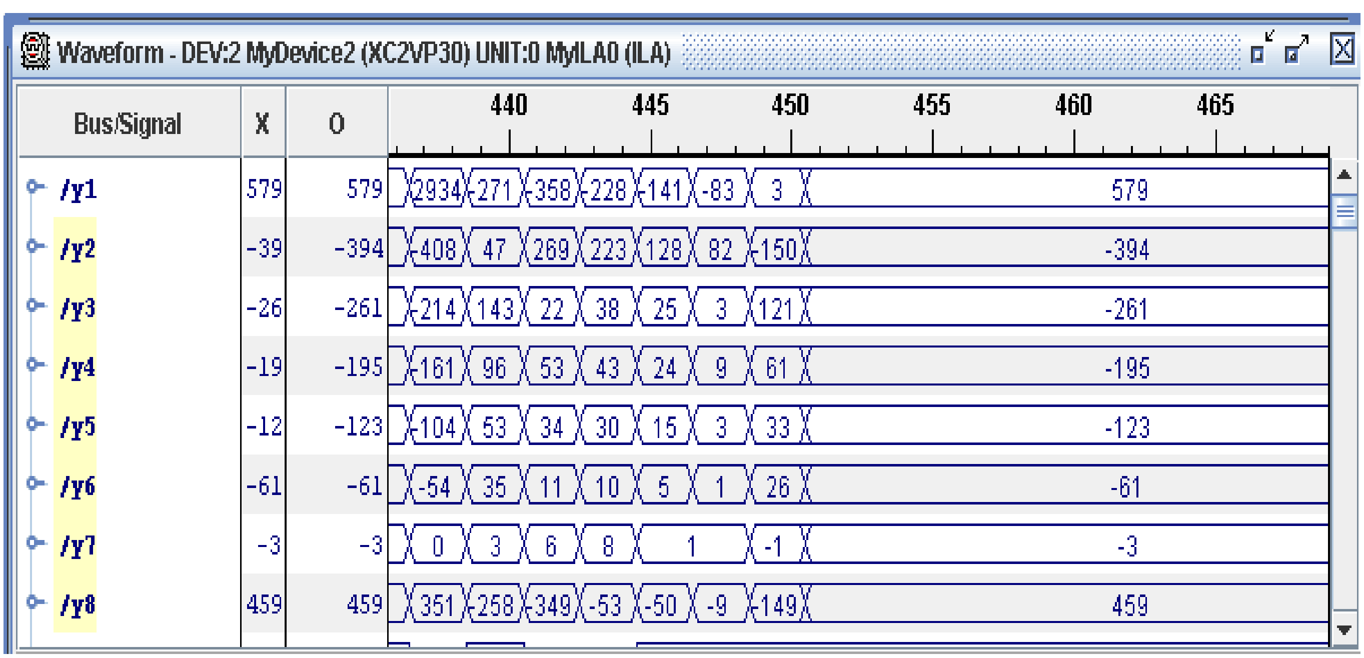

Fig.5 Hardware implementation result of 8x8 image data in table 7 by ROM DA method

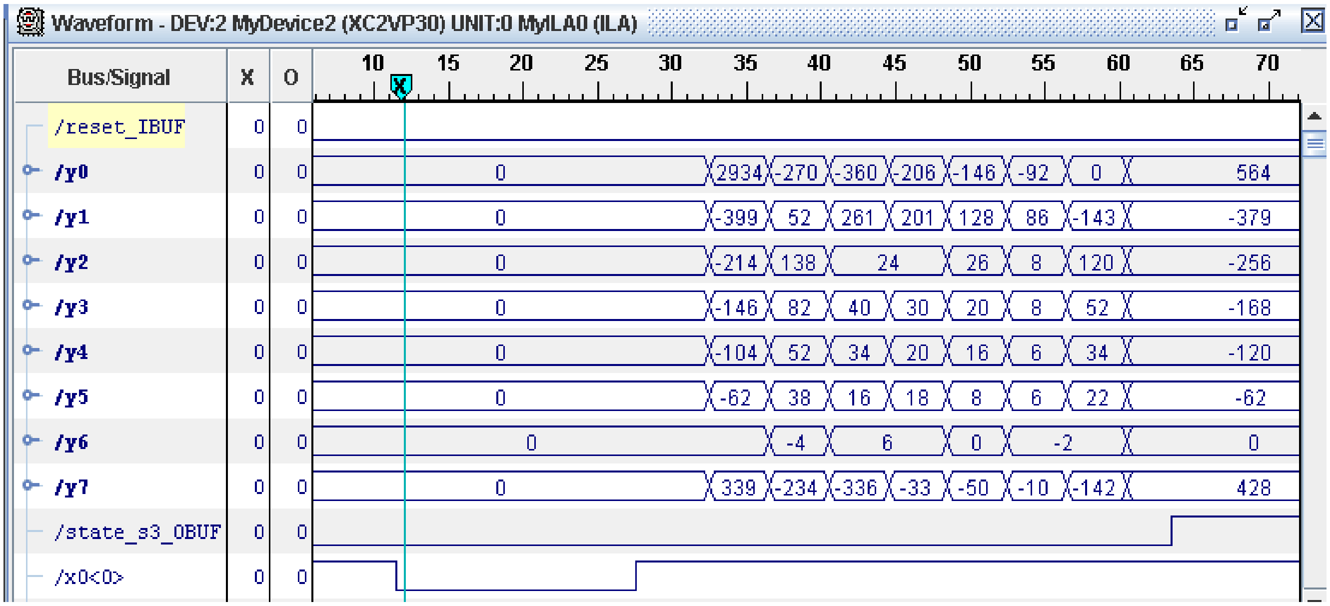

Fig.6 Hardware implementation result of 8x8 image data in table 7 by proposed DA for DHT method

VII. CONCLUSIONS

SDHT is used for image transform and image is reconstructed by taking inverse of SDHT. Reconstructed image shows that SDHT can be used in place of DCT in transform based coding without errors. Two DA approaches namely (1) memory based (2) proposed without use of memory have been used to implement both 1-D and 2-D DHT in Xilinx XC2VP30 FPGA device. Comparison results indicate the significance hardware savings in the proposed DA based implementation. Proposed technique additionally demonstrates that there is less error and faster speed when compared to memory based DA implementation without use of floating point adder. As compared to existing DCT, less computation is required for DHT.

VIII. ACKNOWLEDGMENT

The authors acknowledge to DIT (Ministry of Information & Communication Technology) for the financial support for carrying out this research work.

IX. REFERENCES

[1] M. Kovac, N. Ranganathan, "JAGUAR: A Fully Pipelined VLSI Architecture for JPEG Image Compression Standard," *Proceedings of the IEEE*, vol.83, no.2, pp. 247-258,Aug. 2002

[2] A. P. Morgan, L. T. Watson, R. A. Young, "A Gaussian Derivative Based Version of JPEG for Image Compression and Decompression," *IEEE Transactions on image Processing,* vol.7, no.9, pp.1311 – 1320, Sep. 1998.

[3] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics,* vol. 38(I), Feb. 1992.

[4] R. N. Bracewell, O. Buneman, H. Hao, J. Villasenor, ''Fast Two-Dimensional Hartley Transform," *Proceedings of the IEEE,* vol.74, pp. 1282 – 1283, Sept. 1986.

[5] A. B. Watson , and A .Poirson, ''Separable two-dimensional discrete Hartley transform," *Journal of the Optical Society of America A,* vol.3, no.12, Dec. 1986.

[6] P. K. Meher, T . Srikanthan, J. C. Patra, ''Scalable and Modular Memory-Based Systolic Architectures for Discrete Hartley Transform," *IEEE Transactions on Circuits and Systems*, vol.53, no.5, pp. 1065 – 1077, May 2006.

[7] C. Moraga, "Generalized Discrete Hartley Transforms," *39th International Symposium on Multiple-Valued Logic*, ISMVL, pp. 185 – 190, May 2009.

[8] Sabri A. Mahmoud, Ashraf S. Mahmoud, "The use of Hartley transform in OCR with application to printed Arabic character recognition," *Pattern Analysis & Applications,* vol.12(4), pp. 353-365, July.2008.

[9] S. K. Pattanaik, K. K. Mahapatra, "DHT Based JPEG Image Compression Using a Novel Energy Quantization Method," *IEEE International Conference on Industrial Technology*, pp.2827-2832, Dec.2006.

[10] Peng Cao, Chao Wang, Jun Yang, Longxing Shi, "Area-Efficient Line-based Two-dimensional Discrete Wavelet Transform Architecture without Data Buffer," *IEEE International Conference on Multimedia and Expo,* ICME, pp. 1094 – 1097, June 28 - July 3, 2009.

[11] XiuHua Ji, CaiMing Zhang, JiaYe Wang, S. H. Boey, "Fast 2-D 8×8 discrete cosine transform algorithm for image coding," *Science in China Series F: Information Sciences*, 52(2), pp. 215-225, 2009.

[12] A. Amira, "An FPGA based System for Discrete Hartley transforms," *International Conference on Visual Information Engineering*, pp. 137 – 140, 7-9 July 2003.

[13] Peng Chungan, Cao Xixin, Yu Dunshan, Zhang Xing, "A 250MHz optimized distributed architecture of 2D 8x8 DCT," *7th International Conference on ASIC,* pp. 189 – 192, Oct. 2007.

[14] Jung Sub Kim , Chi-Li Yu, Lanping Deng , S. Kestur, V. Narayanan, C. Chakrabarti, "FPGA architecture for 2D Discrete Fourier Transform based on 2D decomposition for large-sized data," *IEEE Workshop on Signal Processing Systems*, SiPS, pp. 121 – 126, Oct. 2009.

[15] M. A. Ibarra-Manzano, D.-L. Almanza-Ojeda, M. Devy, J.-L. Boizard, J.-Y. Fourniols, "Stereo Vision Algorithm Implementation in FPGA Using Census Transform for Effective Resource Optimization," *12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools,* DSD, pp. 799 – 805, Dec. 2009.

[16] J. A. Valero, J. Septien, D. Mozos, H. Mecha, "3D FPGA resource management and fragmentation metric for hardware multitasking," *IEEE International Symposium on Parallel & Distributed Processing, 2009*. IPDPS 2009, pp.1 – 7.

[17] Uwe Meyer-Baese, Digital signal processing with field programmable gate arrays*,* Springer 2001.

[18] Xilinx Inc., www.xilinx.com.

[19] Liang Tao, Hon Keung Kwan, "Fast Parallel Approach for 2-D DHT-Based Real-Valued Discrete Gabor Transform," I*EEE Transactions on Image Processing,* vo.18(12), pp. 2790-2796, Dec.2009.

[20] Philippe Carre, Eric Andres "Discrete analytical Ridgelet transform," HAL - CCSD, Jan. 22, 2009.

[21] Carlos Hernando Ramiro, Manuel Blanco-Velasco, Eduardo Moreno-Martínez, Fernando Cruz-Roldán, José Sáez Landete, "Efficient Source Coding in a Thresholding-based ECG Compressor using the Discrete Wavelet Transform," *BIOSIGNALS*, pp. 259-264, INSTICC Press, 2009.

[22] Sadik Ali M. Al-Taweel, Putra Sumari, Saleh Ali K. Alomari, Anas J.A. Husain, "Digital Video Watermarking in the Discrete Cosine Transform Domain," *Journal of Computer Science*, vol.5 (8), pp. 536-543, 2009.

[23] M.-T. Sun, T.-C. Chen, A.M. Gottlieb, ''VLSI Implementation of a 16x16 Discrete Cosine Transform," *IEEE Transactions on Circuits and Systems*, vol.36, no. 4, pp. 610 – 617, Apr. 1989.

[24] Chih-Hsien Hsia, Jing-Ming Guo, Jen-Shiun Chiang, "Improved Low-Complexity Algorithm for 2-D Integer Lifting-Based Discrete Wavelet Transform Using Symmetric Mask-Based Scheme," *IEEE Transactions on Circuits and Systems for Video Technology,* vol.19(8), pp. 1202 – 1208, Aug. 2009.

[25] Dhaha Dia, Medien Zeghid, Taoufik Saidani, Mohamed Atri, Belgacem Bouallegue, Mohsen Machhout, Rached Tourki, "Multi-level Discrete Wavelet Transform Architecture Design," *Proceedings of the World Congress on Engineering 2009 Vol I, WCE '09*, pp191-195, July 1 - 3, 2009, London, U.K.

[26] Ibrahim Saeed Koko, Herman Agustiawan, "Parallel Form of the Pipelined Intermediate Architecture for Two-dimensional Discrete Wavelet Transform*," IAENG International Journal of Computer Science*, vol.36(2), pp. 94-111, 2009.

[27] K. Kyriakoulakos, D. Pnevmatikatos, "A novel SRAM-based FPGA architecture for efficient TMR fault tolerance support," *International Conference on Field Programmable Logic and Applications, 2009*. FPL 2009, pp.193 – 198. Prague.

[28] C. Chakrabarti, J. Jaja, "Systolic architectures for the computation of the discrete Hartley and the discrete cosine transforms based on prime factor decomposition," *IEEE Transactions on Computers*, vol.39, no.11, pp. 1359 – 1368, Nov. 1990.

[29] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine,* vol.6, no.3, pp.4-19, Jul.1989.

[30] A. Shams, A. Chidanandan, W. Pan, and M. Bayoumi, "NEDA: A low power high throughput DCT architecture," *IEEE Transactions on Signal Processing,* vol.54, no.3, Mar. 2006.