An ASIC FPGA based low power, high throughput and area efficient video transform core.

[1]v.nandini, [2]k.siva nagi reddy
[1]M.Tech Student of JNTU, Hyderabad, telagana-India
[2]associate professor, ECE Dept, JNTU, Hyderabad, telagana-India

Abstract-— As the technology advances the systems size i.e. the area of the system are reduced. In the same way the digital applications like images and video size are also reduced by different methods. So here we are proposing the video transform core with ASIC FPGA process. The transform core we are using here is discrete cosine transform which is the most prominent because of its energy compaction property.

In the existing paper we are proposing the 2-D forward discrete cosine transform (FDCT) and inverse discrete cosine transform (IDCT). The DCT core used here is the 1-D DCT core. The 1-D DCT core can be used for the linear running of operations of 1ST dimensional and 2nd dimensional (1st FDCT,2nd FDCT,1st IDCT.2nd IDCT) and also the even and odd symmetrical properties of FDCT and IDCT computations can share the hardware resources. The above process is implemented cadence tool and for high throughput we are designing the system in ASIC FPGA for more efficient and increased throughput and low power consumption.

Index Terms-- **area efficiency, forward and inverse discrete cosine transform, high throughput.**

INTRODUCTION

FOR any image and video compression application the discrete cosine transform (DCT) is used extensively because of its unique property (i.e.) energy compaction property. So there copious studies on FDCT /IDCT circuit designs.

In general the direct method (1)-(2) and row column decomposition (3)-(4) can be used for designing the 2-D DCT's which includes transpose memory (TMEM) in order to implement 2D operation. In order to achieve a high throughput rate we preset two 1-D cores which use eight parallel paths and TMEM in (5) and (6) but the same operations have been done.

The low cost multiplexed 2-D DCT cores which uses TMEM and a single 1-D core are introduced in order to eliminate the redundancy (7)-(8).the 1st dimensional and 2nd dimensional cannot be executed synchronously by using 1-D core. Thus the throughput in multiplexed 2-D

DCT's is less when compared with operational frequency due to this extra input buffers should be added because of the 2nd-D data. the multiplier based 1D-DCT core is that simultaneously carried out 1stD and 2nd D operations and moreover an adder based space time scheduling(STS)DCT core which is done in low cost design is presented where simultaneous operations are done(9).

1-D DCT core is more important fundamental component in design. There will be high throughput and low cost for 2D-DCT cores. But due to this will consume more area which will cause area inefficient design.

To eliminate this drawback, in this paper a 2-D DCT core which comprising a single 1-D core is proposed. Due to derivation of the FDCT and IDCT algorithms the DCT core possesses 1st-D and 2nd-D operations for every 8 clock cycle period and FDCT and IDCT computations for every 4 clock cycle period.

The DCT achieves a throughput of 135 mega pixels per seconds and power is 1190 uw when measurement results are done. Moreover to evaluate the performance of the design we were employing ASIC FPGA.when the DCT design was implemented in ASIC FPGA platform the design achieves high throughput at low power and cost.

The remaining part of this paper is as follows. In section II, the 8 x 8 -2D FDCT & IDCT mathematical derivation is introduced .in section III and IV explains the derivation of the proposed 2D-DCT design and construction is evoloved.section V addresses comparisons and section VI

ISSN: 2278 – 7798

gives the brief conclusion drawn.Procedure for Paper Submission

II. MATHEMATICAL DERIVATION OF 8 x 8 2-D FDCT & IDCT.

In this paper the 8x 8 2-D FDCT&IDCT were elected as the MPEG-1/2/4 compression standards. According to the 2-D algorithms of of FDCT&IDCT can be divided into 1-D operations with transpose. The 1-D eight point FDCT & IDCT are considered in the following equations.

$$z_n = \frac{1}{2} k_n \sum_{m=0}^{7} x_m \times \cos\left(\frac{(2m+1)n\pi}{16}\right) \qquad (1)$$

$$x_m = \frac{1}{2} \sum_{n=0}^{7} k_n \times z_n \times \cos\left(\frac{(2m+1)n\pi}{16}\right) \qquad (2)$$

Equation (1) is referred as transpose matrix and equation (2) is referred as inverse transpose matrix. for non-zero n and an and xm represent the input data and the transform output respectively.

By not considering the scaling factor ½, the FDCT in equation (1) it can be divided into even and odd parts ze and zo as shown in the following equations.

$$Z_e = \begin{bmatrix} Z_0 \\ Z_2 \\ Z_4 \\ Z_6 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} = C_e.a \qquad (1)$$

$$Z_o = \begin{bmatrix} Z_1 \\ Z_3 \\ Z_5 \\ Z_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} = C_o.b \qquad (2)$$

Where ci=cos (ipi/16)

In the same way, IDCT can be written as following using (2)

$$\frac{1}{2}a = \begin{bmatrix} c_4 & c_2 & c_4 & c_6 \\ c_4 & c_6 & -c_4 & -c_2 \\ c_4 & -c_6 & -c_4 & c_2 \\ c_4 & -c_2 & c_4 & -c_6 \end{bmatrix} = C_e{}^T.Z_e \qquad (3)$$

$$\frac{1}{2}b = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} = C_o{}^T.Z_o \qquad (4)$$

So the same coefficients for FDCT and IDCT computations can be shared from equations 3 and 6 so that it can be further designed into the same hardware resource

III.DERIVATION OF THE PROPOSED FDCT AND IDCT

From the equations 3 and 6 we can say for the even parts of FDCT and IDCT have the same coefficients matrix is used in IDCT and FDCT.Thus the FDCT and IDCT coefficient matrices can be modified so that it can share the same computation resource by using below steps.

1) rewriting the FDCT computation in 3 by reorganizing the inputs by the coefficients

$$= \begin{bmatrix} a0+a3 & 0 & a1+a2 & 0 \\ a0+a3 & 0 & -a1-a2 & 0 \\ 0 & a0-a3 & 0 & a1-a2 \\ 0 & a2-a1 & 0 & a0-a3 \end{bmatrix} \begin{bmatrix} c4 \\ c2 \\ c4 \\ c6 \end{bmatrix} \qquad (5)$$

2) in the same way rearranging the coefficients and inputs of IDCT computations in 7 we get as follows

$$\begin{bmatrix} d2 \\ d3 \\ d0 \\ d1 \end{bmatrix} = \begin{bmatrix} 0 & Z2 & 0 & Z6 \\ 0 & -Z6 & 0 & Z2 \\ Z0 & 0 & Z4 & 0 \\ Z0 & 0 & -Z4 & 0 \end{bmatrix} \begin{bmatrix} c4 \\ c2 \\ c4 \\ c6 \end{bmatrix} \qquad (6)$$

ISSN: 2278 – 7798

3) Eight multipliers are typically applied to calculate the even parts of the FDCT & IDCT computations in (3) & (6) by taking 4 clock cycles. However in the FDCT computation in (10), 2 multipliers are ideal in the each clock cycle. The IDCT computation also has this property in (14) therefore the even part of the FDCT and IDCT computation from (10) & (14) can be merged by removing the ideal cycles and repermitting the FDCT & IDCT computations

$$= c_4 \begin{bmatrix} a_0 + a_3 + a_1 + a_2 \\ a_0 + a_3 - a_1 - a_2 \\ Z_0 + Z_4 \\ Z_0 - Z_4 \end{bmatrix} \qquad (7)$$

$$\begin{bmatrix} d_3 \\ d_2 \\ Z_3 \\ Z_6 \end{bmatrix} = \begin{bmatrix} -Z_6 & Z_2 \\ Z_2 & Z_6 \\ a_0 - a3 & a_1 - a_2 \\ a_2 - a_1 & a_0 - a_3 \end{bmatrix} \begin{bmatrix} c_2 \\ c_6 \end{bmatrix} \qquad (8)$$

In similar manner the odd parts can be rewritten using 4 and 7 to get 17 and 18 by rearranging the coefficients and the inputs as follows

$$\begin{bmatrix} Z_7 \\ Z_5 \\ Z_1 \\ Z_3 \end{bmatrix} = \begin{bmatrix} -b_3 & b_2 & b_0 & -b_1 \\ -b_1 & b_3 & b_2 & b_0 \\ b_0 & b_1 & b_3 & b_2 \\ -b_2 & b_0 & -b_1 & -b_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_3 \\ c_7 \\ c_5 \end{bmatrix} \qquad (9)$$

$$\cdot \frac{1}{2} \begin{bmatrix} b_0 \\ b_1 \\ b_3 \\ b_2 \end{bmatrix} = \begin{bmatrix} Z_1 & Z_3 & Z_7 & Z_5 \\ -Z_5 & Z_1 & -Z_3 & -Z_7 \\ -Z_7 & Z_5 & Z_1 & -Z_3 \\ -Z_3 & Z_7 & Z_5 & Z_1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_3 \\ c_7 \\ c_5 \end{bmatrix} \qquad (10)$$

So in the second row input [-b1 b3 b2 b0] of FDCT i.e. in equation can be obtained by shifting one element from the first row input [-b3 b2 b0 –b1] to the right with the particular signs. Similarly the 3rd and 4th rows can be obtained by shifting the 2nd row input element to the right side by the corresponding signs. Therefore the proposed architecture adopts this feature in the odd parts of the FDCT and IDCT computations in order to share hardware resources. The detailed architecture described in the further e section.

IV. PROPOSED 8x8 2-D DCT.

The proposed DCT core consists of single 1-D core in order to achieve area efficient design.

The proposed DCT architecture consists of the following blocks (i)prgenerator module(PRGM),(ii) an even part processing element(PEE),(iii)an odd part processing element(PEO)and (iv)a post generation module(POGM) in fig (1)

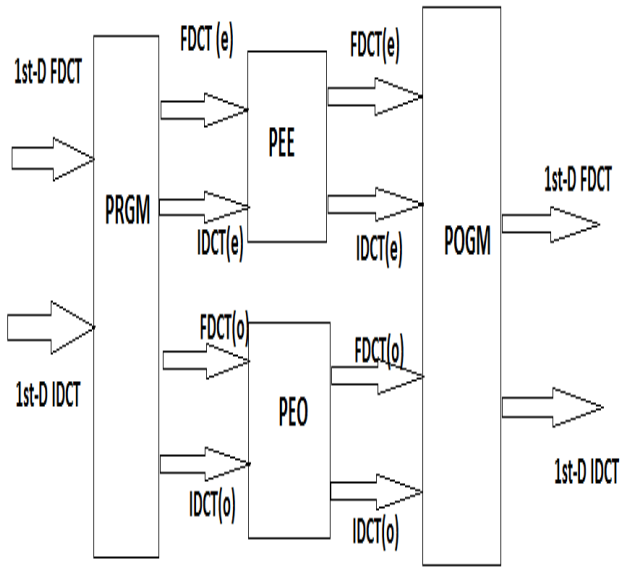


Figure (1).architecture of the proposed 1D-DCT core.

A. PRGM module:

The first module of the proposed DCT core is the pregenerator module (PRGM).it comprises of the two multiplexers which can divide the 1st-D and 2nd-D input data among each other, the LIFO (last in first out)buffers

ISSN: 2278 – 7798

are given to the direct output step. An adder, a sub tractor and two MUX's are given at output stage that gives the run ahead data as (a, b) and (ze, zo) for the FDCT and IDCT data processing appropriately.

The input word length of two input MUX's and LIFO buffers is 21-bits,which observes the FDCT and IDCT data words.so according to the PRGM module FDCT data contains 9MSB's and IDCT data contains 12 LSB's. The MUX we use in this block diagram are the 2-1 MUX which takes the two inputs and gives one output according to the selection line. The LIFO buffers which gives the last in first output vice versa

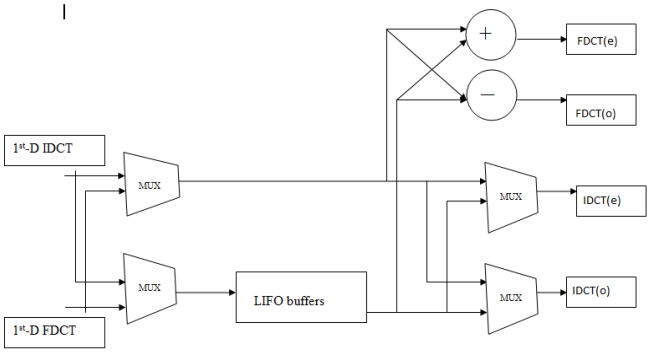


Figure (2).architecture of the PRGM module.

B. PEE module.

According to the proposed DCT core design the 2nd module is the PEE (even part processing element) which consists preprocessing element (PRPE), two constant multipliers and a post processing element

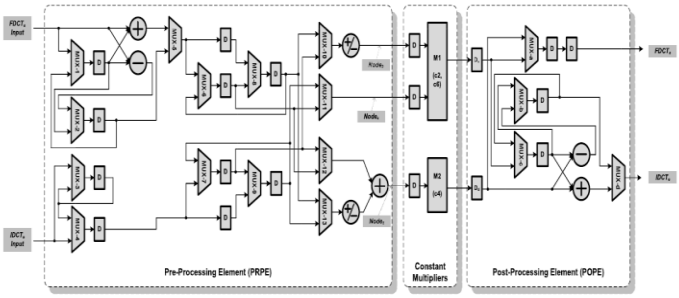


Figure (3) architecture of the PEE module.

a) **Preprocessing element**: as the name of the block suggest that we are calculating the even part of the of both FDCT and IDCT values in this block. The inputs of the preprocessing elements are two. which comprises of 1 input as FDCT even input which has the bit length of 9 bits and the other input comprises of IDCT even input which comprises of 12 bits.PRPE calculates the from the basis of 15 and 16 equations which are processes accordingly.

b) **Constant multipliers:** the next stage of the PEE consists of two constant multipliers. There are K1 and K2.the k1 module calculates the multiplications with the coefficients c2 and c6 and the module K2 accomplish the multiplication with the coefficient c4.

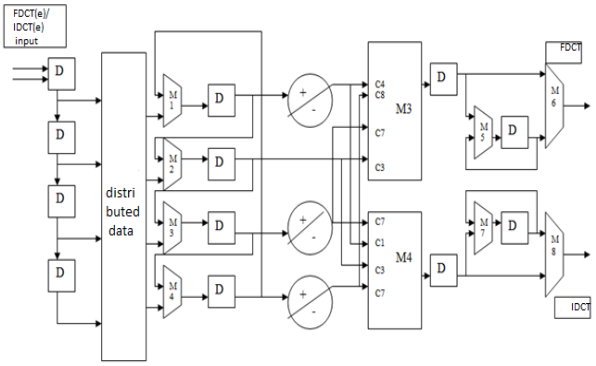
c) **Post processing element:** the final step for the PEE module is POPE module. This gives the final values as the even part of the (FDCTe) and the (IDCTe) data. Here the input data which is given (i.e.)[d3 d2 z2 z6]and [z0 z1 d0 d1] order and is computed and rearranged in following[z4 z2 z6 z0] and [a0 a1 a2 a3] for FDCTe and IDCTe.

C. PEO module:

Corresponding to the PEE module the 1st-D and 2nd D transform have been executed in an eight clock cycle period. In converse the identical characteristics of the even part the PEO uses the some computation paths for the FDCT and IDCT operations.

As mentioned above in contrast to the PEE we are going to use the same operations for PEO based on the equations (17) and (18) the data is distributed module allows the FDCT and IDCT input data as [b3 b2 b0 b1] and [z1 z3 z7 z5].the four shift registers D1 to D4 rotate the {FDCT,IDCT} This is shown in the following figure (4).

ISSN: 2278 – 7798



figure(4) architecture of PEO module.

D. Post generator module:

The transformed data output from PEE and PEO is given to the POGM.the following figure (5) shows.the module of POGM.the upper part of the POGM circuit rearranges the computations results of FDCT and lower part will rearrange the IDCT results

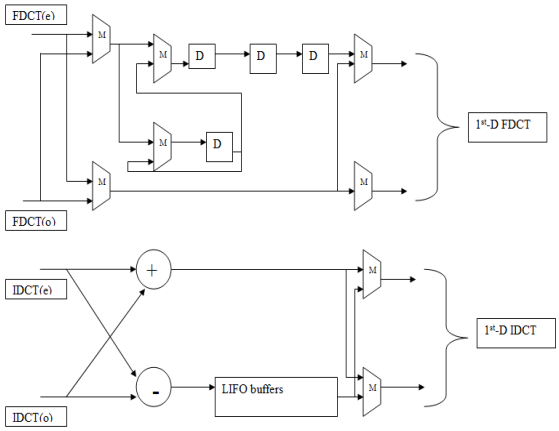


Figure (5) architecture of POGM module.

V. DISCUSSIONS.

a. Chip implementation*:*

To verify the performance of the proposed 8x8 1-D DCT core are synthesizes the RTL code and a cadence SOC encounter for placement and routing(p and R).measurement results shows that the DCT core has a latency of clock cycles, operates aMHZ and ugate counts. The throughput can support 1080p high definition television (HDTV) or other high resolution specifications.

The power consumption of the proposed core in FPGA is 1190 uW

b. ASIC FPGA implementation:

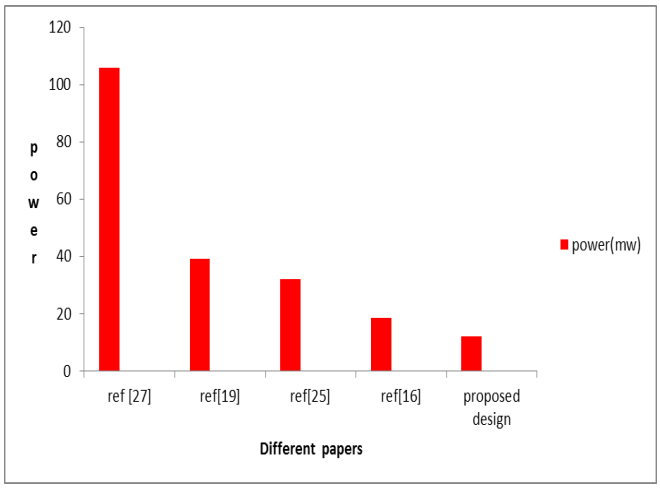
The DCT core was implemented in ASIC FPGA by using cadence –incisive tool to synthesize the power, area and throughput of the design. And the simulation results are shown in following table.

c. Simulated power and throughput analysis.

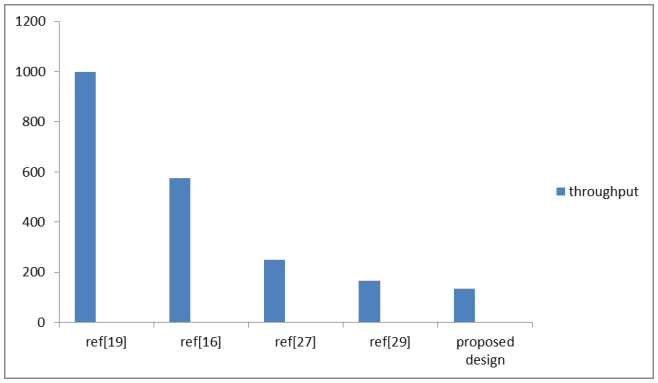
Different designs.	[19]	[25]	[16]	[14]	[28]	[3]	Proposed design
Technology.	180	180	90	90	180	350	180
Throughput(mp/s)	1000	167	575	200	202.8	125	135
Power(mW)	39	32	18.6	0.56	N/A	N/A	1911.08(Uw)

The following graphs represent the data comparing the references papers with the proposed design.

ISSN: 2278 – 7798



I.graph represents the comparision of power with different reference papers and the proposed paper.



II.graph represents the comparision of throughput with different reference papers and the proposed paper.

VI. Conclusion

In this paper, a 1-D 8 × 8 DCT core was proposed. The proposed DCT design uses a single 1-D core to achieve a low-power design. By using the proposed design the 1-D core can linearly execute 1st-D FDCT, 2nd-D FDCT, 1st-D IDCT, and 2nd-D IDCT operations. The results showed that the proposed DCT core possesses a high level of hardware efficiency when supporting multiple FDCT/IDCT functions. Furthermore, the proposed method can easily be applied to other transforms, which are adopted as high-efficiency video coding (HEVC) [10], [11], H.264 [12] and VC-1 [13], by modifying the constant multipliers (M1, M2, M3, M4) in the PEE and PEO. Consequently, the proposed DCT core achieves the goal of creating a high-performance simultaneous transform core and this method can be applied to future image/video transform

VII. REFERENCES

[1] C. T. Lin, Y. C. Yu, and L. D. Van, "Cost-effective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 16, no. 8, pp. 1058–1071, Aug. 2008.

[2] S. F. Hsiao, Y. H. Hu, T. B. Juang, and C. H. Lee, "Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 8, pp. 1568–1579, Aug. 2005.

[3] G. Karakonstantis, N. Banerjee, and K. Roy, "Process-variation resilient and voltage-scalable DCT architecture for robust low-power computing," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol. 18, no. 10, pp. 1461–1470, Oct. 2010.

.[4] Y. H. Chen and T. Y. Chang, "A high performance video transform engine by using space time scheduling strategy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 655–664, Apr. 2012.

[5] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. SignalProcess.*, vol. 54, no. 3, pp. 955–964, Mar. 2006.

[6] Y. H. Chen, T. Y. Chang, and C. Y. Li, "High throughput DA-based DCT with high accuracy error-compensated adder tree," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no.4, pp. 709–714, Apr. 2011.

[7] S. C. Hsia and S. H. Wang, "Shift-register-based data transposition for cost-effective discretecosine transform," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp.725–728, Jun. 2007.

[8] N. J. August and D. S. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 414–422, Jun. 2004.

[9] Y. H. Chen and T. Y. Chang, "A high performance video transform engine by using space- time scheduling strategy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 655–664, Apr. 2012.

[10] M. Martuza and K. Wahid, "A cost effective implementation of 8×8 transform of HEVC from H.264/AVC," in *Proc. IEEE CCECE*, Feb. 2012, pp. 1–4.

[11] P. T. Chiang and T. S. Chang, "A reconfigurable inverse transform architecture design for HEVC decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 1006–1009.

[12] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no. 7, pp. 560–576, Jul. 2003.

[13] H. Kalva and J. B. Lee, "The VC-1 video coding standard," *IEEE Multimedia*, vol. 14, no. 4, pp. 88–91, Oct./Dec. 2007.

[14] J. Lee, N. Vijaykrishnan, and M. Irwin, "Inverse discrete cosine transform architecture exploiting sparseness and symmetry properties," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 655–662, May 2006.

[15] S. F. Hsiao, Y. H. Hu, T. B. Juang, and C. H. Lee, "Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 8, pp.1568–1579, Aug. 2005.

[16] G. Karakonstantis, N. Banerjee, and K. Roy, "Process-variation resilient and voltage-scalable DCT architecture for robust low-power computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 10, pp. 1461–1470, Oct. 2010.

ISSN: 2278 – 7798

[17] G. Ruiz, J. Michell, and A. Buron, "Parallel-pipeline 8 × 8 forward 2-D ICT processor chip for image coding," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 714–723, Feb. 2005.

[18] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 955–964, Mar. 2006.

[19] Y. H. Chen, T. Y. Chang, and C. Y. Li, "High throughput DA-based DCT with high accuracy error-compensated adder tree," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 4, pp. 709–714, Apr. 2011.

[20] S. C. Hsia and S. H. Wang, "Shift-register-based data transposition for cost-effective discrete cosine transform," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 725–728, Jun. 2007.

[21] H. C. Hsu, K. B. Lee, N. Y. Chang, and T. S. Chang, "Architecture design of shape-adaptive discrete cosine transform and its inverse for MPEG-4 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 375–386, Mar. 2008.

[22] J. I. Guo, R. C. Ju, and J. W. Chen, "An efficient 2-D DCT/IDCT core design using cyclic convolution and adder-based realization," *IEEE Trans. Circuits Syst. Vide Technol.*, vol. 14, no. 4, pp. 416–428, Apr. 2004.

[23] N. J. August and D. S. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 414–422, Jun. 2004.

[24] A. Madisetti and A. N. Willson, "A 100 MHz 2-D 8 × 8 DCT/IDCT processor for HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp.158 165, Apr. 1995.

[25] Y. H. Chen and T. Y. Chang, "A high performance video transform engine by using space-time scheduling strategy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 655–664, Apr. 2012.

[26] T. Dias, S. Lopez, N. Roma, and L. Sousa, "A flexible architecture for the computation of direct and inverse transforms in H.264/AVC video codecs," *IEEE Trans. Consume. Electron.* vol. 57, no. 2, pp. 936–944, May 2011.

[27] J. Park, S. Kwon, and K. Roy, "Low power reconfigurable DCT design based on sharing multiplication," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 2002, pp. 3116–3119.

[28] M. Martuza, C. McCrosky, and K. Wahid, "A fast hybrid DCT architecture supporting H.264, VC-1, MPEG-2, AVS and JPEG codecs," in *Proc. Int. Conf.*

ISSN: 2278 – 7798