Multiplier Less ROM Free DA based DCT

[1]Kasa Srinivasulu, [2]K.Satya Sujith,I.S.T.E ,[3]C.Laxmana Sudheer(M.tech),[4]Y.Chamundeswari(M.tech)
[1]Associate Professor,[2,3,4] Assistant Professor
[1]Brahmaiah College Of Engg.,[2] Audisankara Institute Of Technology

Abstract— **Discrete cosine transform (DCT) is a widely used tool in image and video compression applications. Recently, the high-throughput DCT designs have been adopted to fit the requirements of real-time application. Operating the shifting and addition in parallel, an optimized adder tree (OAT) is proposed to deal with the truncation errors and to achieve low-error and high-speed discrete cosine transform (DCT) design. Instead of the 12 bits used in previous works, 9-bit Distributed Arithmetic was proposed. DA-based DCT design with an optimized adder tree (OAT) is the proposed architecture in which, OAT operates shifting and addition in parallel by unrolling all the words required to be computed. Furthermore, the Error-Compensated Circuit alleviates the truncation error for high accuracy design. Based on low-error OAT, the DA-precision in this work is chosen to be 9 bits instead of the traditional 12 bits. Therefore, the hardware size and cost is reduced, and the speed is improved using the proposed OAT.**

Keywords- **Adders, DCT- Discrete Cosine Transform, DA-Distributed Arithmetic, OAT- optimized adder tree**.

I.INTRODUCTION

Today we are talking about digital networks, digital representation of images, movies, video, TV, voice, digital library-all because digital representation of the signal is more robust than the analog counterpart for processing, manipulation, storage, recovery, and transmission over long distances, even across the globe through communication networks. In recent years, there have been significant advancements in processing of still image, video, graphics, speech, and audio signals through digital computers in order to accomplish different application challenges. As a result, multimedia information comprising image, video, audio, speech, text, and other data types has the potential to become just another data type. Development of efficient image compression techniques continues to be an important challenge to us, both in academia and in industry [1]. In [2] multiplier based DCT's were implemented, later to reduce area ROM-based DA was applied for designing DCT [3]. Then knowing the advantage of ROM-based, DA-based multipliers using ROMs were implemented to produce partial products together with adders that accumulated these partial products. By applying DA-based ROM to DCT core design we can reduce the area required. In addition, the symmetrical properties of the DCT transform and parallel DA architecture can be used in reducing the ROM size in [4], respectively. Recently, ROM-free DA architectures were presented [6]–[11]. Shams *et al.* employed a bit-level sharing scheme to construct the adder-based butterfly matrix called new DA (NEDA) [7]. Being compressed, the butterfly-adder-matrix in [7] utilized 35 adders and 8 shift-addition elements to replace the ROM. Based on NEDA architecture, the recursive form and ALU were applied in DCT design to reduce area cost [8], [9], but speed limitations exist in the operations of serial shifting and addition after the DA-computation. In DA-based computation partial products words are shifted and added in parallel [10] and [11]. However, a large truncation error occurred.

We need to reduce truncation error that error is introduced if the least significant part is directly truncated. In order to reduce truncation error effect several error compensation bias methods have been presented based on statistical analysis of relationship between partial product and multiplier-multiplicand. Hardware complexity will be reduced if truncation error minimized. In general, the truncation part (TP) is usually truncated to reduce hardware costs in parallel shifting and addition operations, known as the direct truncation (Direct-T) method. Thus, a large truncation error occurs due to the neglecting of carry propagation from the TP to Main Part (MP). Distributed arithmetic is a bit level rearrangement of a multiply accumulate to hide the multiplications. It is a powerful technique for reducing the size of a parallel hardware multiply-accumulate that is well suited to FPGA designs. The Discrete cosine transform (DCT) is widely used in digital image processing for image compression, especially in image transform coding. However, though most of them are good software solutions to the realization of DCT, only a few of them are really suitable for VLSI implementation. Cyclic convolution plays an important role in digital signal processing due to its nature of easy implementation. Specifically, there exist a number of well-developed convolution algorithms and it can be easily realized through modular and structural hardware such as distributed arithmetic and systolic array.

II.MATHEMATICAL DERIVATION OF DISTRIBUTED ARITHMETIC

The inner product is an important tool in digital signal processing
applications. It can be written as follows:

$$Y=A^{T}\cdot X=\sum_{i=1}^{L-1} A_i X_i \quad \text{____ (1)}$$

where A_i, X_i and L are ith fixed coefficient, ith input data, and number of inputs, respectively. Assume that coefficient A_i is Q-bit

two's complement binary fraction number. Equation (1) can be expressed as follows:

$$Y=[2^0 \quad 2^{-1} \quad 2^{-2} \quad \ldots\ldots \quad 2^{-(Q-1)}]\begin{bmatrix} A_{10} & A_{20} & \ldots\ldots & A_{L0} \\ A_{11} & A_{21} & \ldots\ldots & A_{L1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ A_{1(Q-1)} & A_{2(Q-1)} & \ldots\ldots & A_{L(Q-1)} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_L \end{bmatrix}$$

$$Y=[2^0 \quad 2^{-1} \quad 2^{-2} \quad \ldots\ldots \quad 2^{-(Q-1)}]\begin{bmatrix} y_0 \\ y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_{(Q-1)} \end{bmatrix}$$

$A_{i,j}$ stay between [1 , 0] Note that y_0 may be 0 or a negative number due to two's complement representation. In (2), y_0 can be calculated by adding all X*i* values when $A_{i,j}$=1 and then the transform output Y can be obtained by shifting and adding all nonzero yi values. Thus the inner product computation in (1) can be implemented by using shifting and adders instead of multipliers. Therefore, low hardware cost can be achieved by using DA-based architecture.

III. PROPOSED OPTIMIZED ADDER TREE ARCHITECTURE

In general, the shifting and addition computation uses a shift-and-add operator in VLSI implementation in order to reduce hardware cost. However, when the number of the shifting and addition words increases, the computation time will also increase. Therefore, the shift-adder-tree (SAT) presented operates shifting and addition in parallel by unrolling all the words needed to be computed for high-speed applications. However, a large truncation error occurs in SAT, and optimized adder tree architecture is proposed in this brief to compensate for the truncation error in high-speed applications

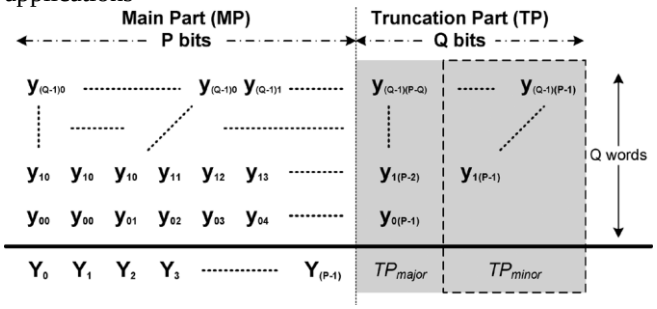

Fig:1: Q,P bit words shifting and addition operations in parallel.

In Fig. 1, the Q P-bit words operate the shifting and addition in parallel by unrolling all computations. Furthermore, the operation in Fig. 1 can be divided into two parts: the main part (MP) that includes _ most significant bits (MSBs) and the truncation part (TP) that has least significant bits (LSBs). a large truncation error occurs due to the neglecting of carry propagation from the TP to MP.

The proposed optimized adder tree architecture is illustrated in Fig. 2 for (P,Q)=(12,6) , where block FA indicates a full-adder cell with three inputs (a, b, and c) and two outputs, a sum (s) and a carry-out (co). Also, block HA indicates half-adder cell with two inputs (a and b) and two outputs, a sum (s) and a carry-out (co).

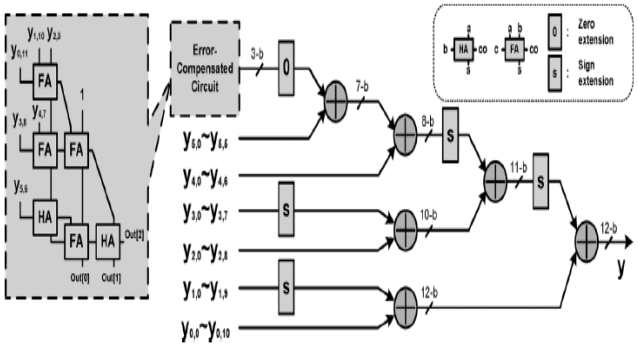

Fig:2 proposed optimized adder tree architecture

IV. PROPOSED 8X8 2-D DCT DESIGN

The 1-D DCT employs the DA-based architecture and the proposed Optimized adder tree to achieve a high-speed, small area, and low-error design. The 1-D 8-point DCT can be expressed as follows:

$$Z_n=\frac{1}{2}K_n\sum_{m=0}^{N-1}x_m\times\cos\left(\frac{(2m+1)n\Pi}{16}\right)$$

Where *xm* denotes the input data;
Zn denotes the transform output.

By neglecting the scaling factor 1/2, the 1-D 8-point DCT in above equation can be divided into even and odd parts: Ze and Zo as listed in below equations, respectively

$$Z_e=\begin{bmatrix} Z_0 \\ Z_2 \\ Z_4 \\ Z_6 \end{bmatrix}\begin{bmatrix} C_4 & C_4 & C_4 & C_4 \\ C_2 & C_6 & -C_6 & -C_2 \\ C_4 & -C_4 & -C_4 & C_4 \\ C_6 & -C_2 & C_2 & -C_6 \end{bmatrix}=\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_2 \end{bmatrix}$$

$$Z_o=\begin{bmatrix} Z_1 \\ Z_3 \\ Z_5 \\ Z_7 \end{bmatrix}\begin{bmatrix} C_1 & C_3 & C_5 & C_7 \\ C_3 & -C_7 & -C_1 & -C_5 \\ C_5 & -C_1 & C_7 & C_3 \\ C_7 & -C_5 & C_3 & -C_1 \end{bmatrix}=\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_2 \end{bmatrix}$$

Where C_i=COS $(i\pi/16)$
Below shows bit level formulation for Z0 and Z4
Let see Z_4 evaluation

TABLE: 1: bit level formulation

Z_1		Z_4	
weight	value	weight	value
-2^0	0	-2^0	A_1
2^{-1}	$B_0+B_1+B_2$	2^{-1}	A_0
2^{-2}	B_0+B_1	2^{-2}	A_1
2^{-3}	B_0+B_3	2^{-3}	A_0
2^{-4}	$B_0+B_1+B_3$	2^{-4}	A_0
2^{-5}	B_0+B_2	2^{-5}	A_1

Where $A_0=(X_0+X_7)+(X_4+X_3)=a_{0+}a_3$
$A_1= (X_1+X_6)+(X_2+X_5) =a_{1+}a_2$
$B_0=(X_0+X_7)-(X_4+X_3)=a_{0-}a_3$
$B_1= (X_1+X_6)-(X_2+X_5) =a_{1-}a_2$

Input data $\mathbf{A_0}$ and $\mathbf{A_1}$, the transform output $\mathbf{Z_0}$ needs only one adder to compute ($\mathbf{A_0}$ + $\mathbf{A_1}$) and two separated optimized adder trees to obtain the results of $\mathbf{Z_0}$ and $\mathbf{Z_4}$. Similarly, the other transform outputs $\mathbf{Z_0}$ and $\mathbf{Z_4}$ can be implemented in DA-based forms using 10(=1 + 9) adders and corresponding optimized adder trees. Consequently, the proposed 1-D 8-point DCT architecture can be constructed as illustrated in Fig. 3 using a DA-Butterfly-Matrix, that includes two DA even processing elements (DAEs), a DA odd processing element (**DAO**) and 12 adders/subtractors, and 8 optimized adder trees (one optimized adder tree for each transform output $\mathbf{Z_n}$). The eight separated optimized adder trees work simultaneously, enabling high-speed applications to be achieved. After the data output from the DA-Butterfly-Matrix is completed, the transform output **Z** will be completed during one clock cycle by the proposed optimized adder trees. In contrast, the traditional shift-and-add architecture requires **Q** clock cycles to complete the transform output **Z** if the DA-precision is **Q**-bits here is 9 bits

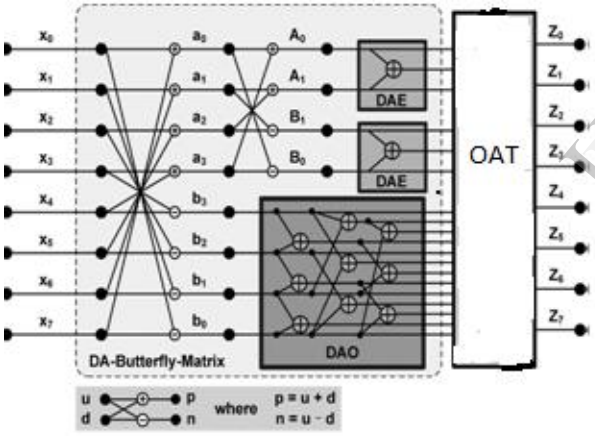

Fig 3. Architecture of the proposed 1-D 8-point DCT.

With high-speed considerations in mind, the proposed 2-D DCT is designed using two 1-D DCT cores and one transpose buffer. For accuracy, the DA-precision and transpose buffer word lengths are chosen to be 9 bits and 12 bits, respectively, meaning that the system can meet the PSNR requirements outlined in previous works. Moreover, the 2-D DCT core accepts 9-bit image input and 12-bit output precision.

V.RESULTS AND DISCUSSIONS

The test image "Lena" used to check system accuracy is comprised of 256X256 Pixels with each pixel being represented by 8-bit 256 gray level data. After inputting the original test image pixels to the proposed 2-D DCT core, the transform output data is captured and fed into MATLAB to compute the inverse DCT using 64-bit double-precision operations. The proposed DCT core has
the highest hardware efficiency, defined as follows (based on the accuracy required by the presented standards)

$$hardware\,efficiency = \frac{Throughput\,rate}{gate\,count}$$

Furthermore, the proposed 2-D DCT core synthesized by using Xilinx ISE 10.1, simulated by using modelsim6.4d and the Xilinx XC2VP30 FPGA can achieve 1067 megapixels per second (M-pels/sec) throughput rate which is 7 folds of previous work of [16]

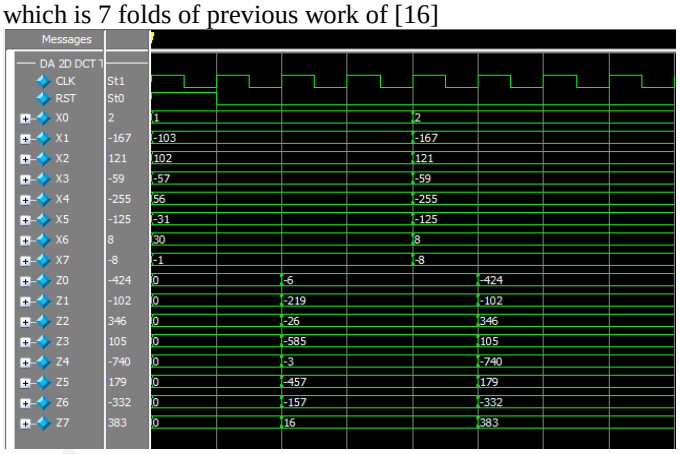

FIG4: Simulation result of DA-DCT

VI.CONCLUSION

The paper contributed with specific simplifications in the multiplier stage, by using shift and adds method, which lead to hardware simplification and speed up over architecture. The proposed 8X8 2-D DCT core has a latency of 10 clock cycles and is operated at 125 MHz As a result of the 8 parallel outputs, the proposed 2-D DCT core can achieve a throughput rate of 1 Gpixels per second (8X125MHz), meeting the 1080 p (1920X1080X60 pixels/s) high definition television (HDTV) specifications for 200 MHz based on low power operations. The maximum throughput rate is 1 Gpels/s.as, the proposed architecture is suitable for high compression rate applications in VLSI designs.

REFERENCES:

[1] Y.Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[2] Y. Chang and C.Wang, "New systolic array implementation of the 2-D discrete cosine transform and its inverse," *IEEE Trans. Circuits Syst Video Technol.*, vol. 5, no. 2, pp. 150–157, Apr. 1995.

[3] 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. Syst., vol. 16, no. 8, pp. 1058–1071, Aug. 2008.

[4] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, Y. Yamashita, H. Yerane, and M. Yoshimoto, "A 100-MHz 2-D discrete cosine transform core processor," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 492–499, Apr. 1992.

[5] S. Yu and E. E. S. , Jr., "DCT implementation with distributed arithmetic," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 985–991, Sep. 2001. 714 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 4, APRIL 2011

[6] P. K. Meher, "Unified systolic-like architecture for DCT and DST using distributed arithmetic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 12, pp. 2656–2663, Dec. 2006.

[7] 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.

[8] M. R. M. Rizk and M. Ammar, "Low power small area high performance 2D-DCT architecture," in *Proc. Int. Design Test Workshop*, 2007, pp. 120–125.

[9] Y. Chen, X. Cao, Q. Xie, and C. Peng, "An area efficient high performance DCT distributed architecture for video compression," in *Proc. Int. Conf. Adv. Comm. Technol.*, 2007, pp. 238–241.

[10] C. Peng, X. Cao, D. Yu, and X. Zhang, "A 250 MHz optimized distributed architecture of 2D 8□8 DCT," in *Proc. Int. Conf. ASIC*, 2007, pp. 189–192.

[11] C. Y. Huang, L. F. Chen, and Y. K. Lai, "A high-speed 2-D transform architecture with unique kernel for multi-standard video applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 21–24.

[12] S. S. Kidambi, F. E. Guibaly, and A. Antonious, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 2, pp. 90–95, Feb. 1996.

[13] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.

[14] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixedwidth multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.

[15] C. C. Sun, P. Donner, and J. Gotze, "Low-complexity multi-purpose IP core for quantized discrete cosine and integer transform," in *Proc.IEEE Int. Symp. Circuits Syst.*, 2009, pp. 3014–3017.

[16] A. Tumeo, M. Monchiero, G. Palermo, F. Ferrandi, and D. Sciuto, "a pipelined fast 2D-DCT accelerator for FPGA-based SoCs," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2007, pp. 331–336

[17] S. Ghosh, S. Venigalla, and M. Bayoumi, "Design and implementation of a 2D-DCT architecture using coefficient distributed arithmetic," in *Proc. IEEE Comput. Soc. Ann. Symp. VLSI*, 2005, pp. 162–166.

[18] Yuan-Ho Chen, Tsin-Yuan Chang, and Chung-Yi Li," High Throughput DA-Based DCT With High Accuracy Error-Compensated Adder Tree" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 4, APRIL 2011