Y Venkat Seshaiah, B.Chinna rao, P.M.Francis / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1458-1462

Verilog Implementation Of Da Based Dct With High Accuracy Error-Compensated Adder Tree

Y Venkat Seshaiah
M.tech (PG student)
Gokul Eng College

B.Chinna rao
Prof. &Head, Dept.of ECE
Gokul Eng College

P.M.Francis
Asst.Prof. In Dept. of ECE
Gokul Eng College

Abstract

In brief, by operating the shifting and addition in parallel, an error-compensated adder-tree (ECAT) is proposed to deal with the truncation errors and to achieve low-error and high-throughput discrete cosine transform (DCT) design. Instead of the 12 bits used in previous works, 9-bit distributed arithmetic-precision is chosen for this work so as to meet peak-signal-to-noise-ratio (PSNR) requirements. Thus, an area-efficient DCT core is implemented to achieve 1 Gpels/s throughput rate with gate counts of 22.2 K for the PSNR requirements outlined in the previous works.

Index Terms—Distributed arithmetic (DA)-based, error-compensated adder-tree (ECAT), 2-D discrete cosine transform (DCT).

I. INTRODUCTION

Discrete cosine transform (DCT) is a widely used tool in image and video compression applications [1]. Recently, the high-throughput DCT designs have been adopted to fit the requirements of real-time applications [2]–[11].

The multiplier-based DCTs were presented and implemented in [2] and [3].To reduce area,ROM-based distributed arithmetic (DA)was applied in DCT cores[4]–[6].Uramoto*et* .[4]implemented the DA-based multipliers usingROMsto produce partial products together with adders that accumulated these partial products. In this way, instead of multipliers, the DA-based ROM can be applied in a DCT core design to 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 [5] and [6], respectively. Recently, ROM-free DA architectures

were presented [7]–[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 arithmetic logic unit (ALU) were applied in DCT design to reduce area cost [8], [9]. Hence the NEDA architecture is the smallest architecture for DA-based DCT core designs, but speed limitations exist in the operations of serial shifting and addition after the DA-computation. The high-throughput shift-adder-tree (SAT) and adder-tree (AT), those unroll the number of shifting and addition words in parallel for DA-based computation, were introduced in [10] and [11], respectively. However, a large truncation error occurred. In order to reduce the truncation error effect, several error compensation bias methods have been presented [12]–[14] based on statistical analysis of the relationship between partial products and multiplier-multiplicand. However, the elements of the truncation part outlined in this work are independent so that the previously described compensation methods cannot be applied.

This brief addresses a DA-based DCT core with an error-compensated adder-tree (ECAT). The proposed ECAT 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 ECAT, the DA-precision n this work is chosen to be 9 bits instead of the traditional 12 bits so as to achieve the peak-signal-to-noise-ratio (PSNR) [1] requirements.
Therefore, the hardware cost is reduced, and the speed is improved using the proposed ECAT.

This brief is organized as follows. In Section II, the mathematical derivation of the distributed arithmetic is given. The proposed ECAT architecture is discussed in Section II. The proposed 8 X 8 2-D DCT core is demonstrated in Section III. The comparisons and results are presented in Section IV, and conclusions are drawn in Section V.

II. MATHEMATICAL DERIVATION OF DISTRIBUTED ARITHMETIC

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

$$Y = \mathbf{A}^T \mathbf{X} = \sum_{i=1}^{L} A_i X_i \quad (1)$$

Where A*i*,X*i* and L are ith fixed coefficient, ith input data, and number of inputs, respectively. Assume that coefficient 4i is Q-bit two's complement binary fraction number. Equation (1) can be expressed as follows:

$$Y = \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(Q-1)} \end{bmatrix} \cdot \begin{bmatrix} A_{1,0} & A_{2,0} & \cdots & A_{L,0} \\ A_{1,1} & A_{2,1} & \cdots & A_{L,1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,(Q-1)} & A_{2,(Q-1)} & \cdots & A_{L,(Q-1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix}$$

$$= \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(Q-1)} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{(Q-1)} \end{bmatrix} \quad (2)$$

where $y_j = \sum_{i=1}^{L} A_{i,j} X_i$, $A_{i,j} \in \{0,1\}$ for $1 \le j \le (Q-1)$, and $A_{i,j} \in \{-1,0\}$ for $j = 0$.

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 alues. 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.

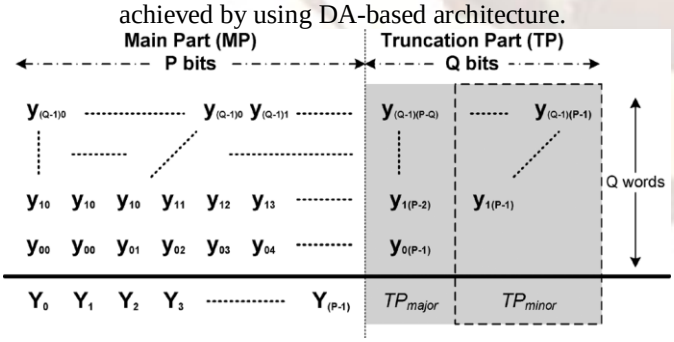


Fig.1.Q P-bit words shifting and addition operations in parallel.

III. ECAT ARCHITECTURE

From (2), the shifting and addition computation can be written as follows:

$$Y = \sum_{j=0}^{Q-1} y_j \cdot 2^{-j}. \quad (3)$$

In general, the shifting and addition computation uses a shift-and-add operator [7] 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 in [10] 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 an ECAT architecture is proposed in this brief to compensate for the truncation error in high-speed applications.

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). Then, the shifting and addition output can be expressed as follows:

$$Y = \mathrm{MP} + \mathrm{TP} \cdot 2^{-(P-2)}. \quad (4)$$

The output Y will obtain the P-bit MSBs using a rounding operation called post truncation (Post-T), which is used for high-accuracy applications. However, hardware cost increases in the VLSI design. In general, the 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 MP. In order to alleviate the truncation error effect, several error compensation bias methods have been presented [12]–[14]. All previous works were only applied in the design of a fixed-width multiplier. Because the products in a multiplier have a relationship between the input multiplier and multiplicand, the compensation methods usually use the correlation of inputs to calculate a fixed [12] or an adaptive [13], [14] compensation bias using simulation or statistical analysis. Note that the addition elements y_{qp} in the TP in Fig. 1 (where 1<=Q<=(Q-1) and P-q-1<=p<=(P-1)) are independent from each other. Therefore, the previous compensation method cannot be applied in this work, and the proposed ECAT is explained as follows.

A. Proposed Error-Compensated Scheme

From Fig. 1, (4) can be approximated as

$$Y \approx \mathrm{MP} + \sigma \cdot 2^{-(P-2)} \quad (5)$$

where σ is the compensated bias from the TP to the MP as listed in (6)–(8)

$$\sigma = \mathrm{Round}(\mathrm{TP}_{\mathrm{major}} + \mathrm{TP}_{\mathrm{minor}}) \quad (6)$$

$$\mathrm{TP}_{\mathrm{major}} = \frac{1}{2}\sum_{j=0}^{Q-1} y_{j(P-1-j)} \quad (7)$$

$$\mathrm{TP}_{\mathrm{minor}} = \frac{1}{4}\left(y_{1(P-1)} + \cdots + y_{(Q-1)(P-Q+1)}\right) + \frac{1}{8}\left(y_{2(P-1)} + \cdots + y_{(Q-1)(P-Q+2)}\right) + \cdots + \left(\frac{1}{2}\right)^{Q} y_{(Q-1)(P-1)} \quad (8)$$

***B.* Performance Simulation for an Error-Compensated Circuit**

In this subsection, comparisons of the absolute average error ξ , the maximum error ξmax, and the mean square error ξm.se for the proposed error-compensated circuit with Direct-T and Post-T are listed. The ξ,, ξmax and ξm.se are defined as follows:

$$\xi = \mathrm{Avg}\{|\mathrm{TP} - \sigma|\}$$
$$\xi_{\max} = \max\{|\mathrm{TP} - \sigma|\}$$
$$\xi_{m.se} = \mathrm{Avg}\{(\mathrm{TP} - \sigma)^2\}$$

C. Proposed ECAT Architecture

The proposed ECAT architecture is illustrated in Fig. 2 for (P,Q)=(12,6) (*case 3*), 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). The comparisons of area, delay, area-delay product, and accuracy for the proposed ECAT with other architectures are listed in Table II. The area and delay are synthesized using a Synopsys Design Compiler with the Artisan TSMC 0.18- μm Standard cell library.

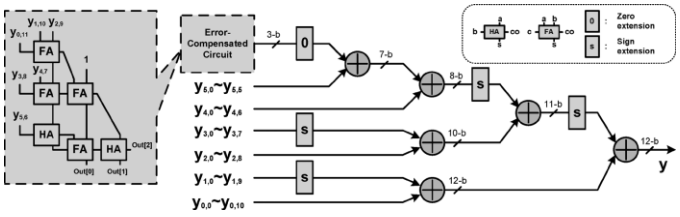


Fig. 2. Proposed ECAT architecture of shifting and addition operators for the (P,Q)=(12,6) example

The proposed ECAT has the highest accuracy with a moderate areadelay product. The shift-and-add [7] method has the smallest area, but the overall computation time is equal to 10.8ns that is the longest. Similarly, the SAT [10], which truncates the TP and computes in parallel, takes 3.72 ns to complete the computation and uses 406 gates, which is the best area-delay product performance. However, for system accuracy, the SAT is the worst option shown in Table I. Therefore, the ECAT is suitable for high-speed and low-error applications.

	Shift-and-add	SAT	Proposed ECAT
Area (gates)	236	406	463
Delay (*ns*)	10.8	3.72	3.89
Area×delay	100 %	59.3 %	70.7 %
ε_{mse}	0.326	6.761	0.218

TABLE 1: COMPARISONS OF THE PROPOSED ECAT WITH OTHER ARCHITECTURES FOR A SIX 8-BIT WORDS EXAMPLE

IV. DISCUSSION AND COMPARISONS

The test image "Lena" used to check system accuracy is comprised of 512 X 512 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 PSNRs are close to 44 and 47 dB for test image and for random 8-bit 256 gray level data inputs, receptively. Table II compares the proposed 8 X 8 2-D DCT core with previous 2-D DCT cores. In [3], a multiplier-based DCT core based on pipeline radix-4square single delay feedback path architecture to achieve high-speed design. The ROM-based DCT core is presented in [4] to reduce adders to reduce the chip area of DCT core. Nevertheless, a speed limitation for shift-and-add is in NEDA design. In [10] and [11], the SAT and AT architectures for DA-based DCTs improve the throughput rate of the NEDA method. However, DA-precision must be chosen as 13 bits to meet the system accuracy with more area overhead. The proposed DCT core uses low-error ECAT to achieve a high-speed design, and the DA-precision can be chosen as 9 bits to meet the PSNR requirements for reducing hardware costs. The proposed DCT core has the highest hardware efficiency, defined as follows (based on the accuracy required by the presented standards).

	Lin et al. [3]	Uramoto et al. [4]	Shams et al. [7]	Chungan et al. [10]	Huang et al. [11]	Proposed
Architecture	Multiplier-based	ROM-based	NEDA	DA-based	DA-based	DA-based
Technology	0.13 μm	0.8 μm	0.18 μm	0.18 μm	0.18 μm	0.18 μm
Multipliers/ROMs	1/0	0/256	0/0	0/0	0/0	0/0
Adders	26	16	92	12+18ALU*+16SAT	50+16AT	46+16 ECAT*
DA-precision	-	-	12 bits	13 bits	13 bits	9 bits
Throughput Rate (pels/sec)	100 M	100 M	77 M†	250 M	400 M	1 G
Gate Counts(NAND2)†	60 K	25.5 K	22.5 K	27.8 K	39.8 K	22.2 K
Hardware Efficiency	1.6	3.92	3.42	9	10.05	45
Accuracy (CCITT◊ Compatible)	Yes	Yes	Yes	N/A	N/A	Yes

TABLE II :COMPARISONS OF DIFFERENT 2-D DCT ARCHITECTURES WITH THE PROPOSED ARCHITECTURE

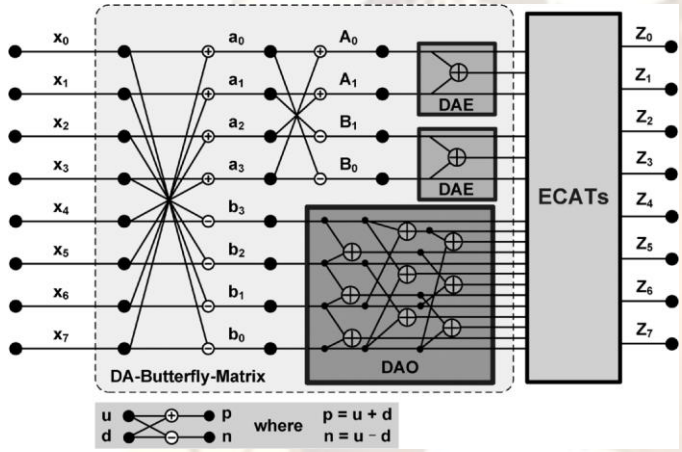
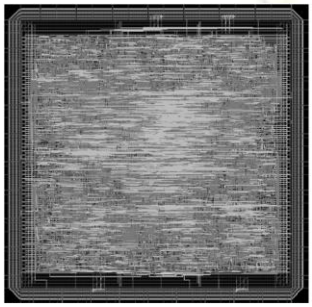


Fig. 3. Architecture of the proposed 1-D 8-point DCT



Characteristics	
Technology	0.18 μm
Supply Power	1.8 V
Die Size	690 μm x 683 μm
Gate Count	22.2 K
Max Freq.	125 MHz
Power	39 mW @125MHz (Max. Freq.) 8.6 mW @ 25MHz (HDTV Spec.)

Fig. 4. Core layout and characteristics

Furthermore, the proposed 2-D DCT core synthesized by using Xilinx ISE 9.1, and the Xilinx XC2VP30 FPGA can achieve 792 mega pixels per second (M-pels/sec) throughput rate (up to about 7 folds of previous work [16]). Table III compares the proposed 2-D DCT core with previous FPGA implementations.

FPGA-Chip	XC2VP30			XC3S200	
Architecture	[15]	[16]	Proposed	[17]	Proposed
# of 4 input LUTs	10310	2618	2990	2271	2847
# of Slices	5729	2823	1872	1221	1585
# of Slice Flip Flops	3736	3431	1837	616	1817
Clock Freq. (MHz)	149	107	99	50	61
Throughput (M-pels/s)	149	107	792	400	488
Power (mW)	N/A	N/A	166.8	281	91

TABLE III: COMPARISONS OF 2-D DCT ARCHITECTURES IN FPGAS

IV.RESULTS AND DISCUSSION

The 8-point 1-D DCT architecture and the implementation were discussed in the previous chapters. Now this chapter deals with the simulation and synthesis results of the implemented 1-D 8-point DCT. Here ModelSim tool is used in order to simulate the design and checks the functionality of the design. Once the functional verification is done, the design will be taken to the Xilinx tool for Synthesis process and the net-list generation.

Simulation Result:

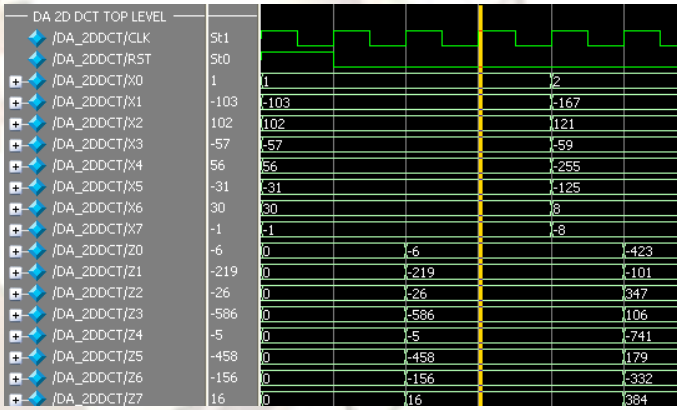


Fig : Simulation results of Distributed Arithmatic DCT

V. CONCLUSION

In this brief, a high-speed and low-error 8 X 8 2-D DCT design with ECAT is proposed to improve the throughput rate significantly up to about 13 folds at high compression rates by operating the shifting and addition in parallel. Furthermore, the proposed error-compensated circuit alleviates the truncation error in ECAT. In this way, the DA-precision can be chosen as 9 bits instead of 12 bits so as to meet the PSNR requirements. Thus, the proposed DCT core has the highest hardware efficiency than those in previous works for the same PSNR requirements. Finally, an area-efficient 2-D DCT core is implemented using a TSMC 0.18- μm process, and the maximum throughput rate is 1 Gpels/s. In summary, 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.

[6] P. K. Meher, "Unified systolic-like architecture for DCT and DST using distributed arithmetic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 3, 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.