

FPGA Implementation of High Throughput DA - Based DCT with High Accuracy ECAT

Lakshmi Kantham A¹ and G. Kishore Kumar²

^{1,2}*Shri Vishnu Engineering College for Women, Bhimavaram, Andhra Pradesh,
JNTU (Kakinada), INDIA.*
^{1,2}*ECE Department.*

Abstract

The Discrete Cosine Transform (DCT) is the most popular and effective one in practical applications because it gives an almost optimal performance and an important component of image and video compression can be implemented at acceptable cost. This project presents an efficient algorithm for implementing the Discrete Cosine Transform (DCT) with Distributed Arithmetic (DA). The design is free of multipliers, DA-Based ROM and ROM accumulators, thereby the High-throughput DCT designs have been adopted to fit the requirements of real-time applications. The Proposed architecture in brief addresses a DA-based DCT core with an Error Compensated Adder Tree (ECAT) which operates shifting an addition by unrolling all the words required to be computed. Furthermore, the ECAT circuit alleviates the truncation error for high accuracy design. The DA precision in this work is chosen to be 9 bits instead of the traditional 12 bits. By using the proposed architecture the speed is improved, area occupied is reduced and having low error rate using the proposed ECAT.

Keywords: Discrete Cosine Transform (DCT); Distributed Arithmetic (DA); Error Compensated Adder Tree (ECAT).

1. Introduction

The system should always provide correct and accurate results in earlier digital VLSI design. The analog computation produces “good enough” results rather than accurate

results. The data processed by many digital systems may already contain errors and during the signal processing, errors may occur.

To deal with error tolerant problems, some truncated adders/multipliers have been reported, but are not able to perform well in its speed, power, area, or accuracy. Of course, not all digital systems can engage the error-tolerant concept. In digital systems such as control systems, the correctness of the output signal is extremely important and the high-throughput DCT designs have been adopted to fit the real time applications, this denies the use of the error compensated adder tree circuit. 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 accumulates that is well suited to FPGA design. The Discrete cosine transform (DCT) is widely used in digital image processing, especially in image transform coding.

Distributed arithmetic was introduced about two decades ago and has been exploited widely in VLSI implementations of DSP architectures. Many commercial DSP chips employ Distributed Arithmetic approach. Most of these applications are computation intensive with multiplier and/or adder being the predominant operator. The main advantage of distributed arithmetic approach is that it speeds up the multiplication process by pre-computing all the possible medium values and storing these values in a ROM. The input data can then be used to address the memory and get the result directly.

1.1 Existing System

1. The multiplier based discrete cosine transform were presented and implemented. It led into more complexity in hardware structure on application specific integrated circuit (ASIC) products in terms of fabrication and also increases in terms of truncation error.
2. The ROM based distributed arithmetic (DA) were applied in DCT core. Still large truncation error occurred.
3. Poor performance in terms of speed of multiplication process due to applied multiplier based DCT core on FPGA implementation.

1.2 Disadvantages of Existing:

1. Poor performance in terms of high accuracy design for real time applications in DCT core on FPGA implementation.
2. Does not achieve in terms of implementation on CMOS technology DCT core.

2. Methodologies

2.1 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 i th fixed coefficient, i th 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:

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

$$= [2^0 \ 2^{-1} \ \dots \ 2^{-(Q-1)}] \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{(Q-1)} \end{bmatrix}$$

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

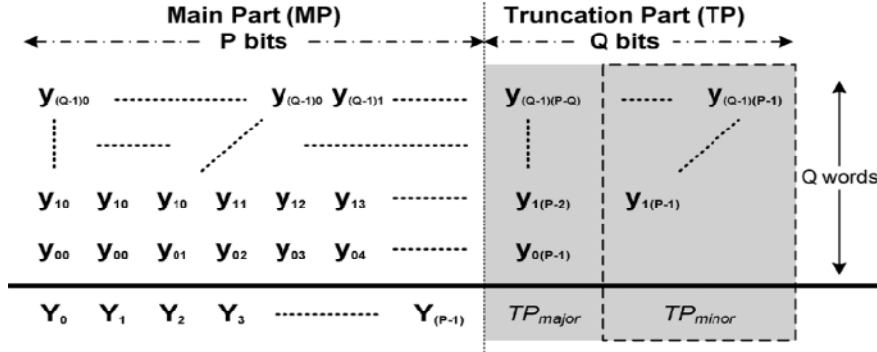


Figure 1: Q P-bit words shifting and addition operations in parallel. This image appears courtesy of [IEEE Transaction On (VLSI) Systems, VOL. 19, NO. 4, APRIL 2011].

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:

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

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. 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 or an adaptive methods, compensation bias using simulation or statistical analysis. Note that the addition elements y_{qp} in the TP in Fig. 1 (where $1 \leq q \leq (Q-1)$ and $P-q-1 \leq p \leq (P-1)$) are independent from each other. Therefore, the previous compensation method cannot be applied in this work, and the proposed ECAT is explained as follows.

2.2 Proposed Error-Compensated Scheme:

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

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

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

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

$$\text{TP}_{\text{major}} = \sum_{j=0}^{Q-1} Y_{j(P-1-j)} \quad (7)$$

$$\text{TP}_{\text{minor}} = \frac{1}{4} (Y_{1(p-1)} + \dots + Y_{(Q-1)(P-Q+1)}) + \frac{1}{8} (Y_{1(p-1)} + \dots + Y_{(Q-1)(P-Q+2)}) + \dots + \left(\frac{1}{2}\right)^Q Y_{(Q-1)(P-1)} \quad (8)$$

2.3 Performance Simulation for an Error-Compensated Circuit :

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

$$\xi = \text{Avg}\{ | \text{TP} - \sigma | \} , \xi_{\text{max}} = \max\{ | \text{TP} - \sigma | \} , \xi_{\text{mse}} = \text{Avg}\{ (\text{TP} - \sigma)^2 \}$$

2.4 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 area and delay are synthesized using a Synopsys Design Compiler with the Artisan TSMC 0.18- μm Standard cell library.

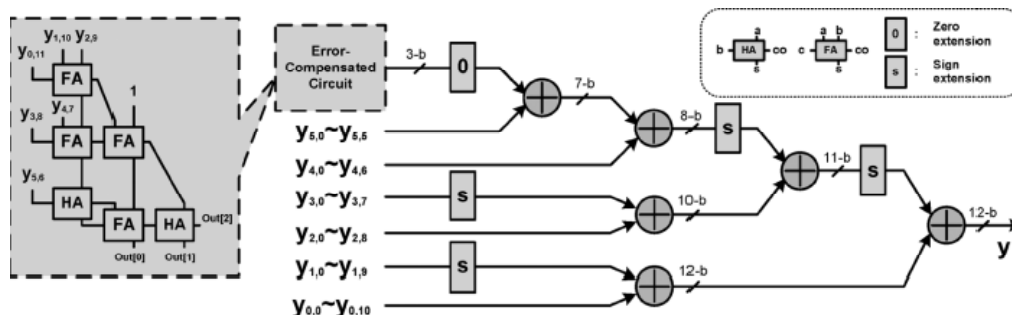


Figure 2: Proposed ECAT architecture of shifting and addition operators for the (P,Q)=(12,6) example. This image appears courtesy of [IEEE Transaction On (VLSI) Systems, VOL. 19, NO. 4, APRIL 2011].

The proposed ECAT has the highest accuracy with a moderate area-delay product. The shift-and-add method has the smallest area, but the overall computation time is equal to 10.8ns that is the longest. Similarly, the SAT, 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.

Table 1: Comparisons of the proposed ECAT with other architectures for a six 8-bit words example. This table appears courtesy of [IEEE Transaction On (VLSI) Systems, VOL. 19, NO. 4, APRIL 2011].

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

2.5 Proposed 8x8 2D DCT Core Design

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

$$Z_n = 1/2 \sum_{m=0}^7 X_m \cos((2m+1)n\pi/16)$$

Where X_m denotes the input data Z_n denotes the transform output $0 \leq n \leq 7$; $k_n = 1/\sqrt{2}$ for $n=0$ and $k_n=1$ for other n values.

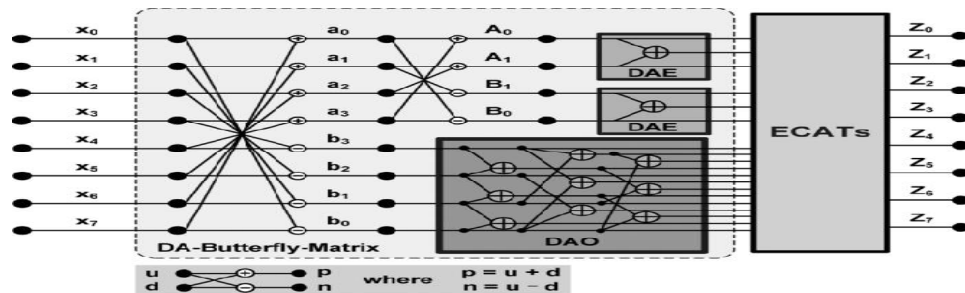


Figure 3: Architecture of the proposed 1-D 8-point DCT. This image appears courtesy of [IEEE Transaction On (VLSI) Systems, VOL. 19, NO. 4, APRIL 2011].

The eight separated ECATs 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 ECATs. Furthermore, the proposed 2-D DCT core synthesized by using Xilinx ISE 12.2, 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).

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.

3. Simulation Results:

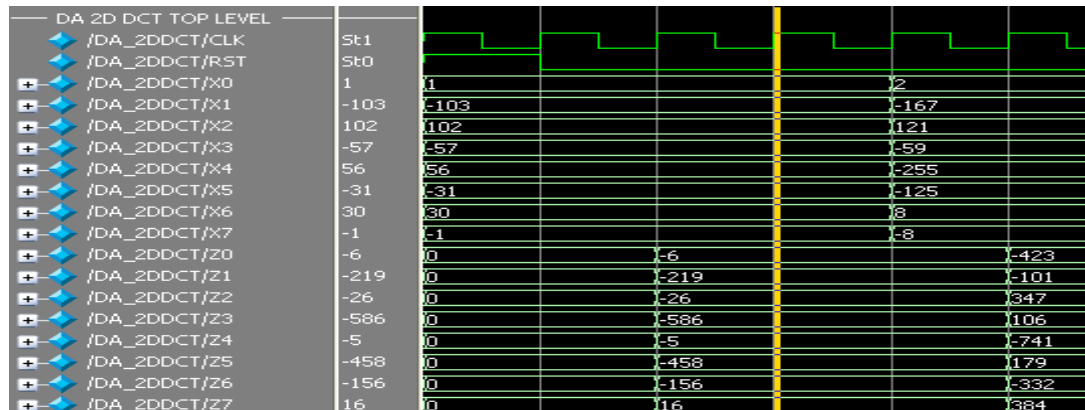


Figure 4: Simulation results of Distributed Arithmetic DCT.

4. Conclusion

This paper presents an FPGA implementation of efficient architecture for computing the 2-D DCT with distributed arithmetic. The proposed architecture requires less hardware than conventional architectures which use the original DCT algorithm or the even-odd frequency decomposition method. The modules of the transpose memory and parallel Distributed Arithmetic 2-D DCT architecture were designed and synthesized. 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.

References

- [1] Yuan-Ho Chen, Tsin-Yuan Chang, and Chung-Yi Li. (2011) "High Throughput DA-Based DCT With High Accuracy Error-Compensated Adder Tree" *IEEE*.
- [2] Y. Chen, X. Cao, Q. Xie, and C. Peng, "An area efficient high performance DCT distributed architecture for video compression,"(2007) in *Proc.Int. Conf. Adv. Comm. Technol.*, pp. 238–241.
- [3] S.Indumathi, Dr.M.Sailaja (2012)," Optimization of ECAT through DA-DCT", in *IOSR*, **3**, Issue 1, pp.39-50.
- [4] Uramoto S, Inoue Y, Takabatake A, Takeda J, Yamashita Y, Yerane H, Yoshimoto M. A (1992), 100-MHz 2-D discrete cosine transform core processor. *IEEE J. Solid-State Circuits*; **27** (4): 492–499.

- [5] M. R. M. Rizk and M. Ammar (2007), "Low power small area high performance 2D-DCT architecture," in *Proc. Int. Design Test Workshop*, pp. 120–125.
- [6] C. Peng, X. Cao, D. Yu, and X. Zhang,(2007),"A 250 MHz optimized distributed architecture of 2D 8x8 DCT," in *Proc. Int. Conf. ASIC*, pp. 189–192.
- [7] S. S. Kidambi, F. E. Guibaly, and A. Antonious, (,Feb. 1996.)"Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. **43**, no. 2, pp. 90–95.
- [8] Y. Chang and C.Wang, (Apr. 1995)"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.
- [9] S. Ghosh, S. Venigalla, and M. Bayoumi,(2005) "Design and implementaion of a 2D-DCT architecture using coefficient distributed arithmetic," in *Proc. IEEE Comput. Soc. Ann. Symp. VLSI*, pp. 162–166.
- [10] P. K. Meher,(Dec. 2006) "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.