ijcrr
Vol 04 issue 12
Category: Research
Received on:31/03/12
Revised on:17/04/12
Accepted on:07/05/12

EFFICIENT DISTRIBUTED ARITHMETIC BASED DISCRETE COSINE TRANSFORM CORE WITH ERROR COMPENSATION

G. Dharini[1], Sharon P.S[1], Arun S[1], R. Radhika[2]

[1]Communication Systems, S.A Engineering College, Chennai, Tamil Nadu
[2]Dept of ECE, S.A Engineering College, Chennai, Tamil Nadu

E-mail of Corresponding Author: dharini01@yahoo.co.in

ABSTRACT

In this paper, an error-compensated adder-tree is proposed to deal with the truncation errors by performing shifting and addition operations in parallel thus achieving low-error and high-throughput discrete cosine transform (DCT) design. The Discrete Cosine Transform is a type of Image Transform which expresses a sequence of finitely many data points in terms of a sum of cosine functions at different frequencies. The proposed scheme incorporates 9-bit distributed arithmetic (DA) - precision for this work instead of the 12 bits in the previous works, so as to meet the desired peak-signal-to-noise-ratio (PSNR). Thus, an area efficient DCT core is implemented to achieve 1 Gpels/s throughput rate for the PSNR requirements outlined in the earlier works.

Keywords*:* Distributed arithmetic (DA)-based, 2-D discrete cosine transform (DCT), PSNR – Peak Signal to Noise Ratio.

INTRODUCTION

Digital images have become attractive from the point of storage and transmission. Satellite and Medical images are good examples. They produce an enormous amount of digital data. Image compression is a technique of mapping images from a higher dimensional space to a lower dimensional space. The basic goal of image compression techniques is to represent an image with minimum number of bits of an acceptable image quality. There are several image compression techniques available. These techniques are generally categorized into two namely lossless and lossy techniques. The Discrete Cosine Transform has shown to be near optimal for a large class of images in energy concentration.

The basis of DCT is decomposing the images into several segments or blocks and obtaining the corresponding frequency components of pixels. During the Quantization process the pixels of frequencies with less importance are discarded, hence the term lossy compression. The important frequency components are retained and they are used to reconstruct the image through the decompression process using the Inverse Discrete Cosine Transform. The loss of information during the reconstruction of the images can be controlled in this process during the compression stage.

The proposed architecture 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 Adder Tree, the DA-precision in this work is chosen to be 9 bits instead of the traditional 12 bits so as to achieve the desired peak-signal-to-noise-ratio (PSNR) requirements. Therefore, the hardware cost is very much reduced, and the speed is greatly improved using the proposed architecture.

DISCRETE COSINE TRANSFORM:

A Discrete Cosine Transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies shown Fig 1.

$$DCT(u,v) = \frac{1}{4} c(u)c(v) \sum_{x=0}^{7} \sum_{y=0} f(x,y) [cos \frac{(2x+1)u\pi}{16}][cos \frac{(2y+1)v\pi}{16}] \quad (1)$$

For u = 0,1,2, … N-1 and v = 0,1,2, … N-1 and a(u) and a(v) are defined as follows:

$$a(u) = \sqrt{\frac{1}{N}} \text{ for } u = 0 \quad a(v) = \sqrt{\frac{1}{N}} \text{ for } v = 0$$

$$= \sqrt{\frac{2}{N}} \text{ for } u \neq 0 \quad = \sqrt{\frac{2}{N}} \text{ for } v \neq 0 \quad (2)$$

All fast DCT implementations usually try to avoid multiplication operations by increasing the number of addition operations and decreasing the number of multiplication operations. Addition actually makes the architecture slow as the time complexity for addition is almost the same as that of fast multipliers.

DISTRIBUTED ARITHMETIC:

Distributed Arithmetic (DA) is an efficient method for computing inner products when one of Distributed arithmetic is an efficient method for the input vectors is fixed. Look-up tables and accumulators are used instead of multipliers for computing inner products and has been widely used in many DSP applications such as DFT, DCT and convolution. In particular, there has been great interest in implementing DCT with distributed arithmetic and in reducing the ROM size required in the implementations since the DA-based DCT

A. **Derivation:**

Distributed Arithmetic is proposed to realize inner product of vectors with optimal solutions in terms of hardware requirement. This features implementation without the need of multipliers, and at the same time, without the need of ROM as in DA approach.

$$Y = \sum_{n=1}^{N} A_n X_n \quad \text{-- (3)}$$

$A_k =$

$$\begin{bmatrix} 2^N & 2^{N+1} & \dots & 2^M \end{bmatrix} \begin{bmatrix} A_k^N \\ A_k^{N+1} \\ \cdot \\ \cdot \\ \cdot \\ -A_k \end{bmatrix}^{M}$$

Where

A is a set of predetermined coefficients, and *x* are data values.

Assume that the coefficient a_n is Q-bit two's complement binary fraction number. The above equation can be expressed as follows:

$$Y = \begin{bmatrix} 2^N & 2^{N+1} & \dots & 2^M \end{bmatrix} \begin{bmatrix} A_1^N & A_2^N & \dots & A_L^N \\ A_1^{N+1} & A_2^{N+1} & \dots & A_L^{N+1} \\ -A_1^M & -A_2^M & \dots & -A_L^M \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_L \end{bmatrix}$$

$$= \begin{bmatrix} 2^N & 2^{N+1} & \dots & 2^M \end{bmatrix} \begin{bmatrix} Y^N \\ Y^{N+1} \\ \dots \\ -Y \end{bmatrix}^{M} \quad \text{-- (4)}$$

Thus, the inner product computation can be implemented by using shifting and adders

instead of multipliers. Therefore, low hardware cost can be achieved by using DA architecture.

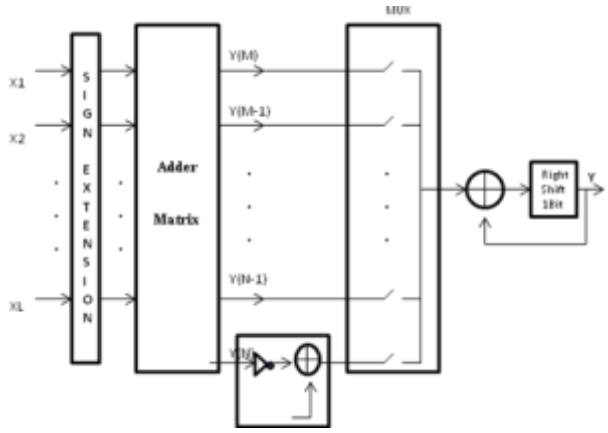

Fig.1. DA Architecture with Addition Operations.

In this figure, input signals are sign extended bits and then fed into the Adder Matrix, which is a butterfly structure with number of output lines determined by DA precision.

B. Error Compensation:

NEDA architecture is the smallest architecture for DA-based DCT designs, but speed limitations exist in the operations of serial shifting and addition after the DA-computation.

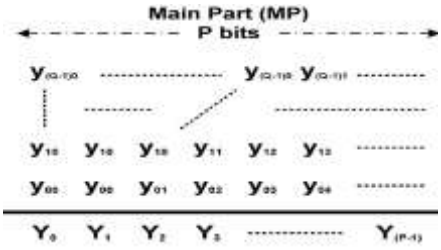

Fig.2. Main part –P bit words

The high-throughput shift-adder-tree and adder-tree, those unroll the number of shifting and addition. However, a large truncation error occurred.

In order to reduce the truncation error, several error compensation methods have been presented based on statistical analysis of the relationship between partial products and multiplier-multiplicand. However, the elements of the truncation part are independent so that the previously described methods cannot be applied.

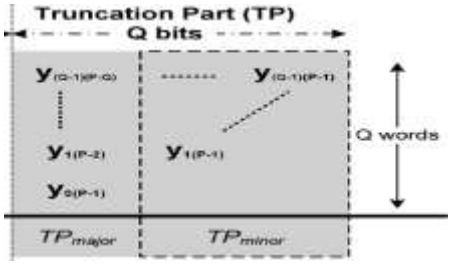

Fig.3. Q P - bit words – Truncation part.

ADDER TREE ARCHITECTURE:

The shifting and addition computation can be written as follows:

$$Y = \sum_{j=0}^{Q-1} y_j 2^{-j} \qquad \text{-- (5)}$$

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 increases, the computation time will also increase. Therefore, the shift-adder-tree presented in 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 and an ECAT architecture is proposed to compensate for the truncation error in high-speed applications. The shifting and addition output can be expressed as follows:

$$Y = MP + TP.2^{-(p-2)}. \qquad -(6)$$

A. Proposed ECAT Architecture:

The equation (6) can be approximated as

$$Y = MP + \varphi .2^{-(p-2)}. \qquad -(7)$$

where φ is the compensated bias from the Truncated Part to the Main Part.

$$\Phi = (TP\ major - TP\ minor)\ \text{round} \quad \text{-- (8)}$$

Where

$$TP\ major = 1/2 \sum_{j=0}^{Q-1} y_{j\,(p-1-j)} \quad \text{-- (9)}$$

$$TP\ minor = 1/2 \{y_{1(p-1)} + \ldots + y_{(Q-1)(P-Q+2)}\} + 1/8 \{y_{2(P-1)} + \ldots + y_{(Q-1)(P-Q+2)}\} + (1/2)^{Q} y_{(Q-1)(P-1)} \quad \text{-- (10)}$$

Difference between the major and the minor terms of the truncation part is obtained and it is rounded to the nearest integer. TP major has more weightage than TP minor and thus only the TP major is calculated and the TP minor is simply estimated. Thus, the compensation bias φ can be written as,

Case i) Q = 0,1,2,3

$$= (TP\ major)\ \text{round} \quad \text{-- (11)}$$

Case ii) Q = 4k, 4k + 1

$$= (k-1) + (TP\ major + 0.5)\ \text{round} \quad \text{-- (12)}$$

Case iii) Q = 4k+2, 4k+3

$$= k + (TP\ major)\ \text{round} \quad \text{-- (13)}$$

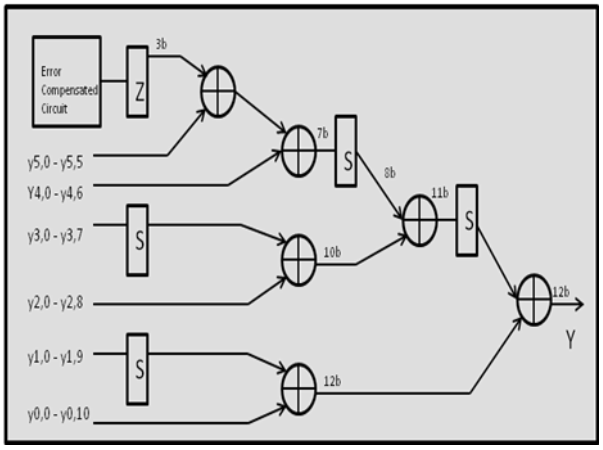

Fig.4. Proposed ECAT Architecture

PROPOSED 2D DCT ARCHITECTURE:

For 2-D DCT computation of a 8x8 2-D data, first row-wise 8x1 1-D DCT is taken for all rows followed by column-wise 8x1 1-D DCT to all columns. Intermediate results of 1-D DCT are stored in transpose memory.

$$F(u) = ½\ C(u) \sum_{i=0}^{7} X(i) \cos\left(\frac{(2i+1)u\pi}{16}\right) \quad \text{-- (14)}$$

The above 1D DCT equation can be simplified as follows:

$$F(0) = [X(0) + X(1) + X(2) + X(3) + X(4) + X(5) + X(6) + X(7)]P$$
$$F(1) = [X(0) - X(7)]A + [X(1) - X(6)]B + [X(2) - X(5)]C + [X(3) - X(4)]D$$
$$F(2) = [X(0) - X(3) - X(4) + X(7)]M + [X(1) - X(2) - X(5) + X(6)]N$$
$$F(3) = [X(0) - X(7)]B + [X(1) - X(6)](-D) + [X(2) - X(5)](-A) + [X(3) - X(4)](-C)$$
$$F(4) = [X(0) - X(1) - X(2) + X(3) + X(4) - X(5) - X(6) + X(7)]P$$
$$F(5) = [X(0) - X(7)]C + [X(1) - X(6)](-A) + [X(2) - X(5)]D + [X(3) - X(4)]B$$
$$F(6) = [X(0) - X(3) - X(4) + X(7)]N + [X(1) - X(2) - X(5) + X(6)](-M)$$
$$F(7) = [X(0) - X(7)]D + [X(1) - X(6)](-C) + [X(2) - X(5)]B + [X(3) - X(4)](-A)$$

Where M = ½ cos π/8, N = ½ cos 3π/8,
P = ½ cos π/4
A = ½ cos π/16, B = ½ cos 3π/16,
C = ½ cos 5π/16, D = ½ cos 7π/16

For eight coefficients computations, 3 modules namely MODULE1, MODULE2 and MODULE3 are constructed as shown in figures 4, 5, 6.

- In MODULE1, (1/2)cos(π/4) is expressed in DA form with one input.

- In MODULE2, 1/2cos(π/16), (1/2)cos(3π/16), (1/2)cos(5π /16), and (1/2)cos(7π /16) are expressed in DA with four inputs.

- In MODULE3 (1/2)cos(π /8) and (1/2)cos(3π /8) is expressed in DA with two inputs.

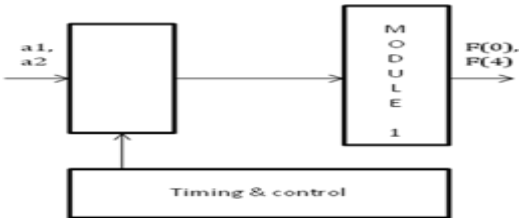

Fig.5. VLSI architecture for Computation of F(0) and F(4)

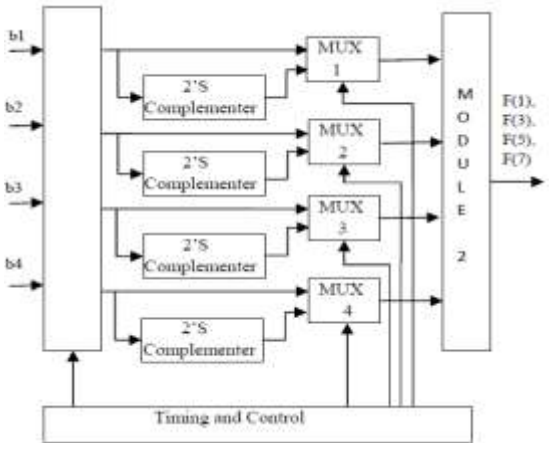

Fig.6. VLSI architecture for Computation of F(1), F(3), F(5) and F(7)

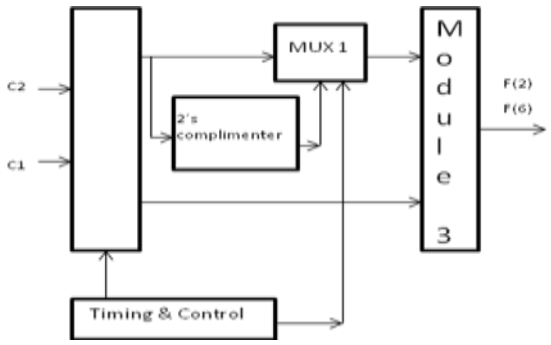

Fig.7. VLSI architecture for Computation of F(2) and F(6)

Let,

a1=X(0)+X(1)+X(2)+X(3)+X(4)+X(5) +X(6) +X(7),

a2=X(0)-X(1)-X(2)+X(3)+X(4)-X(5) -X(6)+X(7),

b1=X(0)-X(7), b2=X(1)-X(6), b3=X(2)-X(5), b4=X(3)-X(4), c1=X(0)-X(3)-X(4)+X(7), c2=X(1)-X(2)-X(5)+X(6).

Considering high-speed implementation, the proposed 2-D DCT is designed using two 1-D DCT cores and one transpose buffer. The DA-precision and transpose buffer word lengths are chosen to be 9 bits and 12 bits, respectively, so that the system can meet the PSNR requirements stated in previous works.

The proposed 8 x 8 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 core can achieve a throughput rate of 1 Gpixels per second.

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.)

	Proposed
Architecture	DA-based
Technology	0.18μm
Multipliers/ROMs	0/0
Adders	46+16 ECAT*
DA-precision	9 bits
Throughput Rate (pels/sec)	1 G
Gate Counts(NAND2)‡	22.2 K
Hardware Efficiency	45
Accuracy (CCITT° Compatible)	Yes

Table.1. Characteristics of the Proposed Architecture

	Shift and add	SAT	Proposed ECAT
Area (gates)	236	406	463
Delay (ns)	10.8	3.72	3.89
Area x Delay	100%	59.3%	70.7%
ε_{mse}	0.326	6.761	0.2218

Table.2. Comparisons of the Proposed ECAT with Other Architectures

Fig. 7 shows a DA-Butterfly-Matrix, that includes two DA even processing elements (DAEs), a DA odd processing element (DAO) and 12 adders/subtractors, and 8 ECATs. 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 will be completed during one clock cycle by the proposed ECATs.

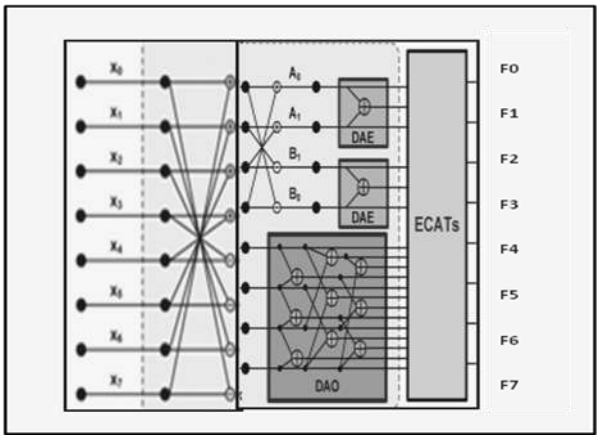

Fig.8. Proposed 2D DCT with ECAT

RESULT AND CONCLUSION

The tables 1 and 2 show the performance characteristics of the proposed design and the comparison of the proposed with the older architectures. Thus a high-speed and low-error 8 x 8 2-D DCT design with ECAT is proposed to improve the throughput rate significantly at high compression rates by operating the shifting and addition in parallel. The proposed error-compensation minimizes the truncation error in ECAT.

Fig.8 shows the VHDL simulation result obtained using Xilinx ISE Simulator for proposed 1D DCT architecture. The DA-precision can be chosen as 9 bits instead of 12 bits so as to meet the PSNR needs. Thus, the proposed DCT core has the highest hardware efficiency than those in previous works. Finally, an area - efficient 2-D DCT architecture is implemented with a maximum throughput rate of 1 Gpixels/s.

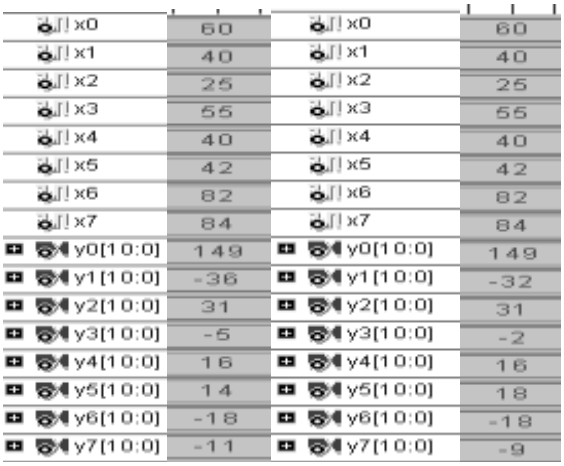

Fig.9. VHDL simulation result using Xilinx for implementation of 1-D DCT

ACKNOWLEDGEMENT

Authors acknowledge the immense help received from the scholars whose articles are cited and included in references of this manuscript. The authors are also grateful to authors / editors / publishers of all those articles, journals and books from where the literature for this article has been reviewed and discussed.

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. White, S.A, .Applications of distributed arithmetic to digital signal processing: a tutorial review., *ASSP Magazine*, IEEE, Volume: 6 Issue: 3 , Jul 1989, Page(s): 4 - 19
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. A. Shams, A. Chidanandan, W. Pan, and M. Bayoumi, "NEDA: A low power high throughput DCT architecture," *IEEE Transactions on Signal Processing*, vol.54(3), Mar. 2006.
7. 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.
8. 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,Feb.1995.
9. Yi Yang; Chunyan Wang; Omair Ahmad, M.; Swamy, M.N.S., .An on-line CORDIC based 2-D IDCT implementation using distributed arithmetic., *Sixth International Symposium on Signal Processing and its Applications*,. 2001 , Vol. 1
10. Wilton, S.J.E.; .Embedded memory in FPGAs: recent research results., *Communications, Computers and Signal Processing, 1999 IEEE Pacific Rim Conference on* , 1999 , Page(s): 292 -29