Susrutha Babu Sukhavasi, Suparshya Babu Sukhavasi, Satyanarayana P, Krishna Karthik T,Alekya M, Arafath Ali / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.3001-3008

Implementation Of High Speed DA-Based DCT With High Accuracy Error-Compensated Adder Tree

Susrutha Babu Sukhavasi*, Suparshya Babu Sukhavasi*, Satyanarayana P*, Krishna Karthik T, Alekya M***, Arafath Ali****.**

*Faculty, Department of ECE, K L University, Guntur, AP, India.
**M.Tech -VLSI Student, Department of ECE, K L University, Guntur, AP, India.
***M.Tech –Embedded Systems Student, Department of ECE, GEC, Krishna, AP, India.
****M.Tech -VLSI Student, Department of ECE, AEC, Nalgonda, AP, India.

ABSTRACT

In this paper, 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.

Keywords – Adders, DCT, Distributed Arithmetic

I. INTRODUCTION

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, especially in image transform coding, as it performs much like the optimal Karhunen-Loeve transform (KLT) under a variety of criteria. Many algorithms for the computation of the DCT have been proposed since the introduction of the DCT by Ahmed, Natarajan, and Rao in 1974. 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 exists 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 . The way of data movement forms a significant part in the determination of the efficiency of the realization of a transform using the distributed arithmetic.

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

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.

II. METHODOLOGIES:

It is the process of analyzing the design of Discrete Cosine Transform (DCT) core with Error Compensated Adder Tree based on distributed arithmetic. An efficient technique for calculation of sum of products or vector dot product or inner product or multiply and accumulate (MAC) MAC operation is very common in all Digital Signal Processing hardware designs. An old technique that has been revived by the wide spread use of Field Programmable Gate Arrays (FPGAs) for Digital Signal Processing (DSP). DA efficiently implements the MAC using basic building blocks (Look Up Tables) in FPGAs .The “basic” DA technique is bit-serial in nature. DA is basically a bit-level rearrangement of the multiply and accumulate operation DA hides the explicit multiplications

by without ROM look-ups an efficient technique to implement on Field Programmable Gate Arrays (FPGAs). The speed in the critical path is limited by the width of the carry propagation Speed can be improved upon by using techniques to limit the carry propagation. An increasing number of services and the growing popularity of high definition TV requires higher coding efficiency. As the ongoing demand increases, for better compression performance of the latest video coding standard, the H.264/AVC (Advanced Video Coding) is formulated .The H.264/AVC is also known as MPEG-4 Part 10 (Wiegand,et al., 2003;Sullivan,et al.,2004; Richardson, 2003).An advantage of the H.264/AVC is the simplicity of its transform. Distributed Arithmetic (DA) is an effective method for computing inner products when one of the input vectors is fixed. It uses pre computed look-up tables and accumulators instead of multipliers for calculating inner products and has been widely used in many DSP applications such as DFT, DCT, convolution, and digital filters. In particular, there has been great interest in implementing DCT with parallel distributed arithmetic and in reducing the ROM size required in the implementations since the DA-based DCT architectures are known to have very regular structures suitable for VLSI implementations. Most DA-based DCT implementations use the original DCT algorithm, or the even-odd frequency decomposition of the DCT algorithm along with some memory reduction techniques such as the partial sum technique and/or the offset binary coding technique. On the other hand, the proposed architecture uses the DA-based DCT algorithm and requires less area than the conventional approaches, regardless of the memory reduction techniques employed in the ROM Accumulators (RACs).

Main Module's:

- ERROR COMPENSATION CIRCUIT
- ERROR COMPENSATED ADDER TREE

1. ERROR COMPENSATED CIRCUIT

Multipliers are commonly used components in digital signal processing applications (DSP). The multiplier produces 2n–bit output for n bit multiplicand and n bit multiplier input. But for some applications we may only require n bit multiplicand result. We can truncate n least-significant bits and preserve the m most significant bit. Although doing this would cause significant errors the area will be reduced half. To reduce truncation error we propose error compensation methods. In some applications these error could be ignored. The output will obtain MSBs using a rounding operation called post truncation (Post-T), which is used for high-accuracy applications. 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 compensation bias using simulation or statistical analysis. The internal word-length usually uses 12 bits in a DCT design. Consequently, word length P=12 is chosen together with different Q values of 3, 6, 9 and12. The Post-T method provides the most accurate values for fixed-width computation nowadays. In addition, the Direct-T method has the largest inaccuracies of the errors for low-cost hardware design. The proposed ECAT is more accurate than Direct-T and is close to the performance of the Post-T method using a compensated circuit

SUB MODULES'S

1. HALF ADDER
2. FULL ADDER
3. ADDITION ELEMENTS

HALF ADDER :

In this module Half Adder is a digital combinational circuit that is used for the addition of two bits and provides an output in the form of a sum bit and a carry bit. The logical functional equations that relate the outputs S and C of a half adder circuit to the input bits are given below :-

Sum(S) = A ex-OR B

Carry(C) = A.B

Thus a half adder circuit can easily be synthesized by using 1 ex-OR gate and 1 AND gate. Since a half adder circuit can only be used to add two bits, it becomes obsolete in case of multi-bit addition in practical applications.

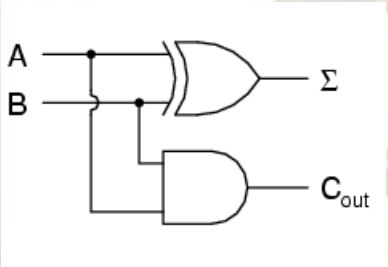

Fig 1: Logic Diagram for Half Adder

FULL ADDER:

In this module describe about full adder operation. The limitation of a half-adder is that it cannot accept a carry-in bit. the carry-in bit represents the carry-out of the previous low-order bit position. Thus a half-adder can be used only for the two least significant digits when adding two multibit binary numbers, since there can be no possibility of a propagated carry to this stage. In multibit addition, a carry bit from a previous stage must be taken into account, which gives rise to the necessity for designing a full adder. A full adder can accept two operands bits, ai and bi, and a carry-in bit ci from previous stage; it produces a sum bit si and a carry-out bit c0. sum bit si is 1 if there is an odd number of 1's at the inputs of the full adder,

whereas the carry-out c0 is 1 if there are two or more 1's at the inputs. The sum and carry out bits will be 0 otherwise. In ripple carry adder, the carry signals must ripple through all the full adders before the outputs stabilize to the correct values; hence such an adder is often called a ripple adder. addition is to be performed the carry-out generated from the least significant stage of the adder propagates through the successive stages and produces a carry-in into the most significant stage of the adder. The time required to perform addition in a ripple adder depends on the time needed for the propagation of carry signals through the individual stages of the adder. Thus ripple carry addition is not instantaneous. The greater the number of stages in a ripple carry adder the longer is the carry propagation time, and consequently the slower the adder

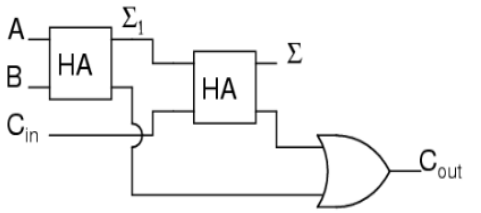

Fig 3– Full Adder Using Half Adder

ERROR COMPENSATED ADDER TREE:

In this module consists of error compensated circuit, adder circuit, sign extension, zero extension. FA indicates a full-adder cell with three inputs (a, b, and c) and two outputs, a sum (s) and a carry-out (co). HA indicates half-adder cell with two inputs (a and b) and two outputs, a sum (s) and a carry-out (co).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 longest. ECAT is suitable for high-speed and low-error applications. authorized doctors can login into the medical. Here also all the details about the doctor are registered by the medical admin. And the medical admin give the authentication details to the particular doctor after getting the authentication details doctor can login to the medical and can start the below processes.

SUBMODULES:

1. **ERROR COMPENSATED CIRCUIT**

2. **PARTIAL PRODUCT GENERATION**

1. ERROR COMPENSATED CIRCUIT

In this module 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 in operates shifting and addition in parallel by unrolling all the words needed to be computed for high-speed applications. The Q P-bit words operate the shifting and addition in par can be divided into two parts: the main part (MP) that includes P most significant bits (MSBs) and the truncation part (TP) that has least significant bits (LSBs).

2. PARTIAL PRODUCT GENERATION:

In this module A truncated multiplier is an m × n multiplier with m bits output. Partial products can be divided into two subsets. The least significant part (LSP) includes the n less significant columns of the partial product matrix, while the most significant part (MSP) includes the remaining columns. The full-width multiplier output, P is given by P = SMSP + SLSP. Where SMSP and SLSP represent the weighted sum of the elements of MSP and LSP respectively. When a n bits output is needed, the most accurate choice is using the full rounded multiplier: it computes all the matrix of partial products, add a constant to the result on 2n bits and takes only the first n bits of the sum. The error introduced by the full rounded multiplier is calculated. Unfortunately the full rounded multiplier is the solution with the highest area occupation and power dissipation. A second possibility is using a truncated multiplier in which the partial products of the LSP are discarded assuming that their contribution to the n most significant bits of the output is negligible. This solution is very advantageous in terms of hardware performances.

DCT PROCESS

In the DCT process input image is divided into non overlapping blocks of 8 x 8 pixels, and input to the baseline encoder. The pixel values are converted from unsigned integer format to signed integer format, and DCT computation is performed on each block. DCT transforms the pixel data into a block of spatial frequencies that are called the DCT coefficients. Since the pixels in the 8 x 8 neighborhood typically have small variations in gray levels, the output of the DCT will result in most of the block energy being stored in the lower spatial frequencies. On the other hand, the higher frequencies will have values equal to or close to zero and hence, can be ignored during encoding without significantly affecting the image quality.

The selection of frequencies based on which frequencies are most important and which ones are less important can affect the quality of the final image. The selection of quantization values is critical since it affects both the compression efficiency, and the reconstructed image quality. High frequency coefficients have small magnitude for typical video data, which usually does not change dramatically between neighboring pixels. Additionally, the human eye is not as sensitive to high frequencies as to low frequencies. It is difficult for the human eye to discern changes in intensity or colors that occur between successive pixels. The human eye tends to blur these rapid changes into an average hue and intensity. However, gradual changes over the 8 pixels in a block are much more discernible than rapid changes. When the DCT is used for compression purposes, the quantizer unit attempts to force the insignificant high frequency coefficients to zero while retaining the important low frequency coefficients. 8-point 1-D Discrete Cosine Transform implemented by shifting and addition in parallel (multiply and accumulate).

- **DA – BUTTERFLY MATRIX**
- **EVEN ELEMENTS**
- **ODD ELEMENTS**

DA-BUTTERFLY MATRIX

In this module The 1-D DCT employs the DA-based architecture and the proposed ECAT to achieve a high speed, small area and low error design. Recently, ROM-free DA architectures were presented employed a bit-level sharing scheme to construct the adder-based butterfly matrix called new DA (NEDA) .Being compressed, the butterfly-adder-matrix in 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. 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. 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. However, a large truncation error occurred.

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 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 an ECAT architecture is proposed in this brief to compensate for the truncation error in high speed applications.

For the DA-based computation, the coefficient matrix are expressed as 9-bit binary fraction numbers. Using given input data A0 and the transform output A1 needs only one adder to compute (A0 + A1) and two separated ECATs to obtain the results of Z0 and Z1.Similarly, the other transform outputs Ze0 and Ze0 can be implemented in DA-based forms using (10 = 1+9) adders and corresponding ECATs. Consequently, from the existing paper , the proposed 1-D 8-point DCT architecture can be constructed 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 ECATs (one ECAT for each transform output Zn). 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. 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.

EVEN ELEMENTS:

In this module we will obtain three output from two input (A,B) according to given structure of proposed 1-D 8 point DCT. Here we are using two Distributed Arithmetic even element.

ODD ELEMENTS:

In this module we will obtain 13 output from four input according to given structure of proposed 1-D 8 point DCT. Here we are using one Distributed Arithmetic odd element.

MODULE DIAGRAM

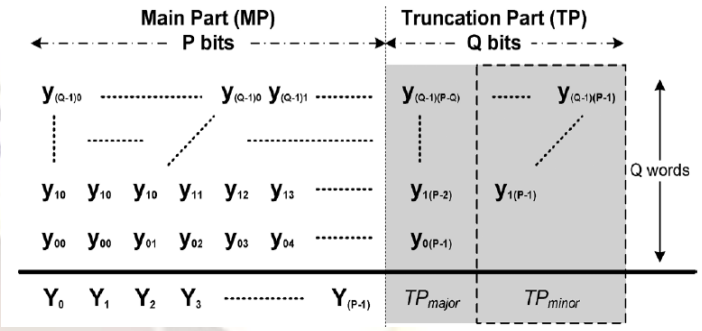

FIG-5: Q, P-bit words shifting and addition operations in parallel

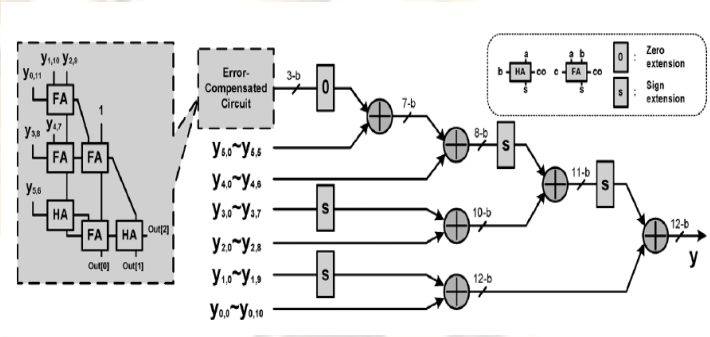

Fig-7 Proposed ECAT Architecture Of Shifting And Addition Operators For (P,Q =12,6)

DISTRIBUTED ARITHMETIC COMPUTATION

It ‘s an efficient technique for performing multiply and add in which the multiplication is re-organized such that multiplication and addition performed on data and single bits of the co-efficient , at the same time . the principle of this technique is based on assumption that we will store the computed values rather than carry out of computation . Assume that we are computing sum of products computation. Where the values Xj represent a data stream and the values (a0,a1,a2,….aN-1) represent a series of co-efficient values. Rather than compute the partial products using gates. We can use memory to generate these and then use fast adders to compute the final multiplication. The inner product is an important tool in digital signal processing applications. 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-based architecture. Distributed arithmetic (DA) is an efficient procedure for computing the inner partial product between a fixed and a variable data vector. DA replaces combinational multipliers used for computing the matrix vector products in DCT by storing the pre

computed co-efficient in ROMs. It uses lookup tables and additions in place of multiplication. where ak = {a0, a1... aN-1} is the fixed co-efficient vector and xk={x0, x1… xN-1} are the input vector values. The partial products computed from the above polynomial products can be stored in a ROM and addressed by the input and accumulated for each input bit of the multiplicand using bit-serial adders and accumulators. In a ROM-based DA, the size of the ROMs and their access time becomes an overhead. Compared to lumped arithmetic-based architectures, distributed arithmetic architectures are completive in both speed and hardware requirements. In addition, they are extremely regular which makes them most suitable for programmable logic realization. A simple derivation of the distributed arithmetic method is as follows: Let the variable Z hold the result of an inner product operation between a data vector x and a coefficient vector c. As with most hardware applications, we can obtain more performance by using more hardware. In this case, more than one ROM can be computed at a time. A parallel implementation of the inherently serial distribution (SD) DCT is as follows: The sample into M samples and processing these sub samples in parallel. Such a parallel implementation requires M-times as many memory look-up tables and so comes at a cost of increased logic requirements. Since the Rom Accumulator (RAC) size in a distributed arithmetic implementation increases exponentially with the number of coefficients, the RAC access time can be a bottleneck for the speed of the whole system when the RAC size becomes large. Hence, we decomposed the single RAC into two RACs, and added their outputs using a two-input accumulator. The total size of storage is now reduced since the accumulator is less costly than the larger RAC. As with most hardware applications, we can obtain more performance by using more hardware. In this case, more than one ROM can be computed at a time. A parallel implementation of the inherently serial distribution (SD) DCT is as follows: The sample into M samples and processing these sub samples in parallel. Such a parallel implementation requires M-times as many memory look-up tables and so comes at a cost of increased logic requirements. Since the RAC size in a distributed arithmetic implementation increases exponentially with the number of coefficients, the RAC access time can be a bottleneck for the speed of the whole system when the RAC size becomes large. Hence, we decomposed the single RAC into two RACs, and added their outputs using a two-input accumulator. The partitioned-RAC architecture is shown. The total size of storage is now reduced since the accumulator is less costly than the larger RAC. 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 . DCTs are important to numerous applications in lossy compression of audio and images (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (as explained below, fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions. All fast DCT implementations usually try to avoid multiplication operations by increasing the number of addition operations and decreasing the number of multiplication operations. Favoring addition over multiplication might actually make the architecture slower since time complexity for addition is almost the same as the time complexity for fast multipliers. A new algorithm Distributed arithmetic (DA) with low hardware requirement suitable for implementation of Discrete Cosine Transform. The main objectives are to minimize the complexity of operations as much as possible while maintaining low delays and high throughput speed. Distributed arithmetic (DA) is an efficient is a powerful technique that has been used for fast and efficient implementation of DCT on FPGA.

III. SIMULATION RESULTS

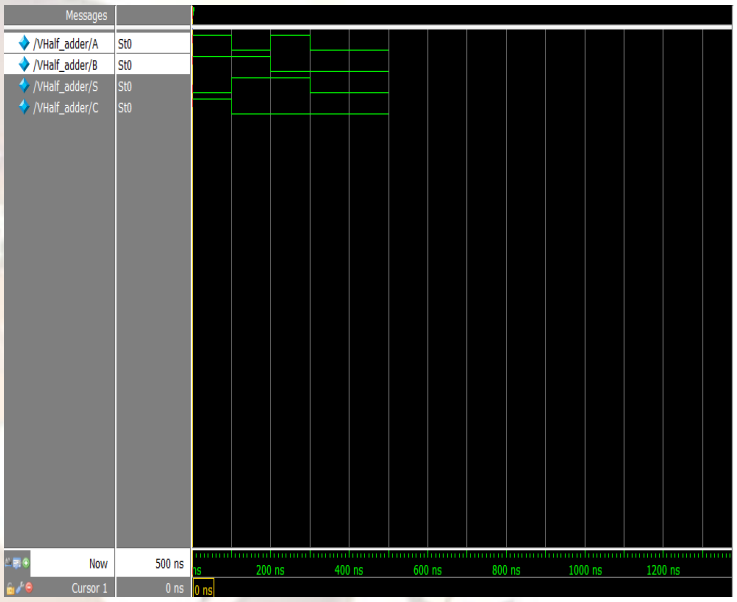

Fig 8 : Half Adder

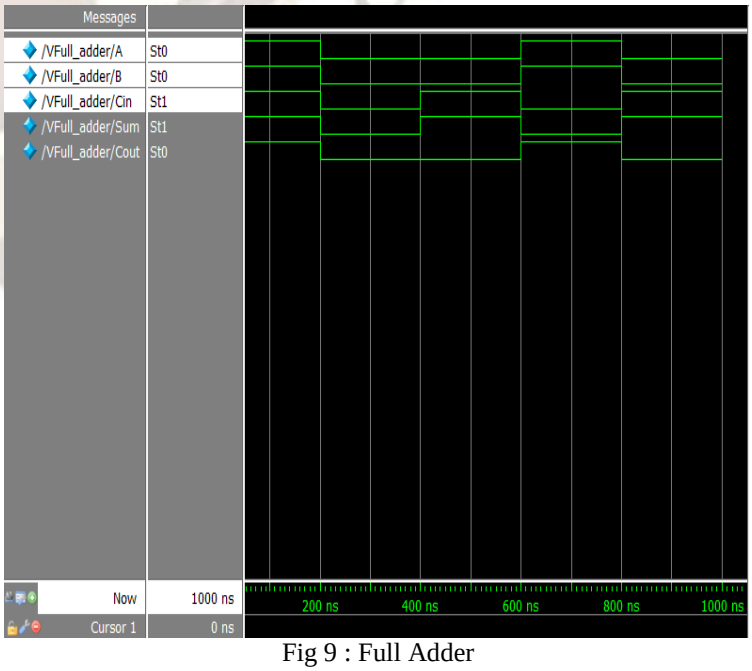

Fig 9 : Full Adder

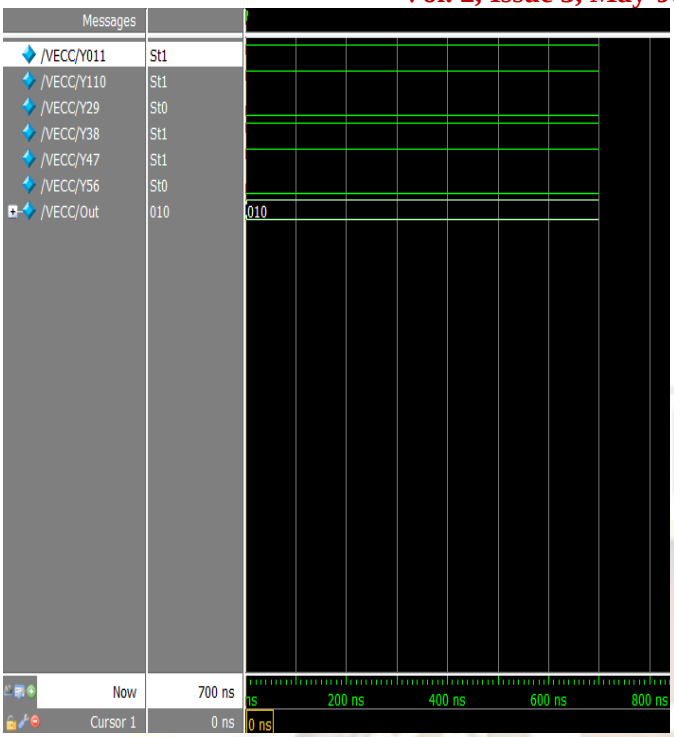

Fig 9 : ECC

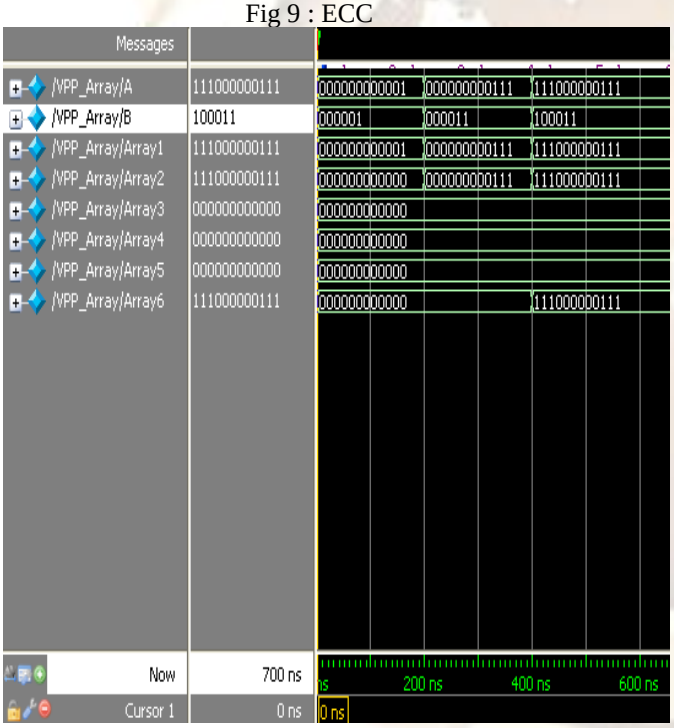

Fig 10 : PARTIAL PRODUCT

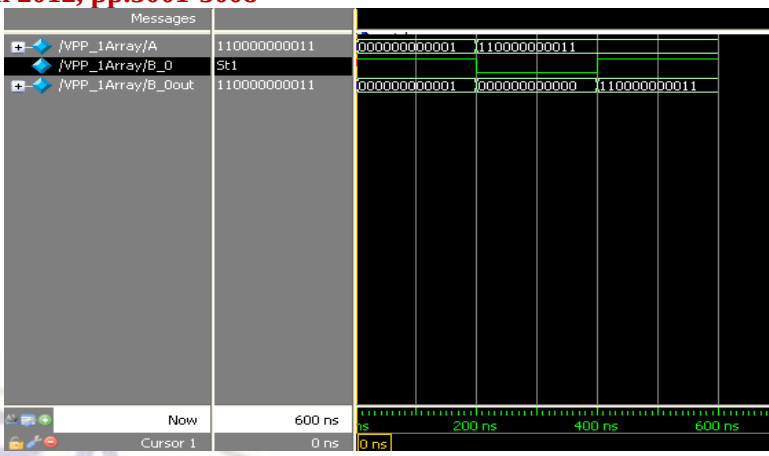

Fig 11 : PARTIAL PRODUCT 1

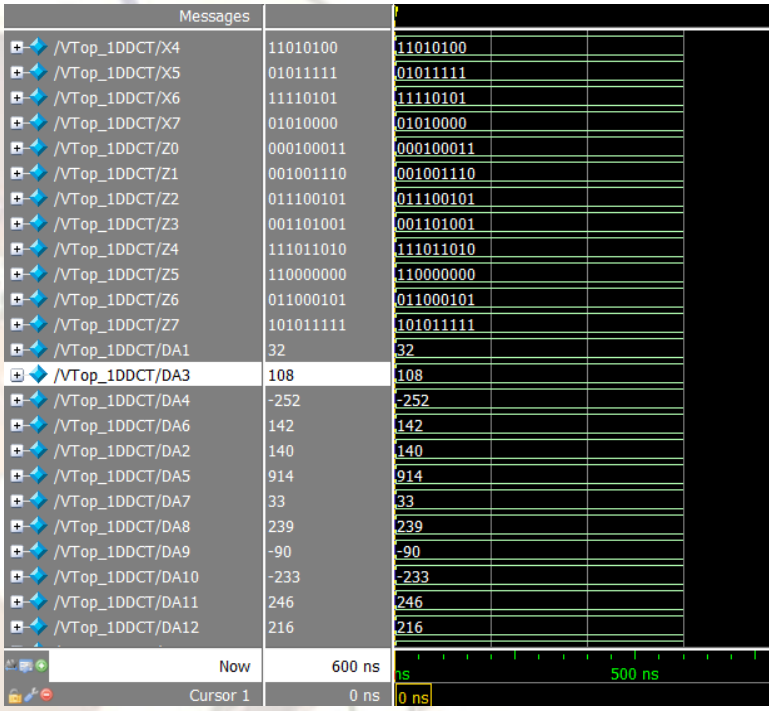

Fig 12 : BUTTERFLY MATRIX

IV. 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 add method, which lead to hardware simplification and speed up over architecture.

V. REFERENCES

1. *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.*

2. *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.*

3 *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.*

4. *C. Peng, X. Cao, D. Yu, and X. Zhang, "A 250 Mhz Optimized Distributed Architecture Of 2 D 8x8 DCT," in Proc. Int. Conf. ASIC, 2007, pp. 189–192.*

5. *C. Y. Huang, L. F. Chen, and Y. K. Lai, "A High-Speed 2-D Transforms Architecture With Unique Kernel For Multi-Standard Video Applications,"In Proc. IEEE Int. Symp. Circuits Syst., 2008, pp. 21–24.*

6. *S. S. Kidambi, F. E. Guibaly, and A. Antonious, "Area-Efficient Multipliersfor Digital Signal Processing Applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 43, no. 2, pp. 90–95, Feb. 1996.*

7. *Finchelstein, D.F.; Sze, V. & Chandrakasan, A.P. (2009). Multicore Processing and Efficient On-Chip Caching for H.264 and Future Video Decoders. IEEE Transactions on Circuits and Systems for Video Technology, Vol.19, No. 11, pp. 1704-1713, doi: 10.1109/TCSVT.2009.2031459, ISSN: 1051-8215*

8. *Lin, Y.K.; Li, D.W.; Lin, C.C.; Kuo, T.Y.; Wu, S.J.; Tai, W.C.; Chang, W.C. and Chang, T.S. (2008). A 242mW 10mm2 1080p H.264/AVC High-Profile Encoder Chip. IEEE International Solid-State Circuits Conference (ISSCC 2008), pp. 314-316, doi: 10.1109/ISSCC.2008.4523183, ISBN: 978-1-4244-2010-0*

9. *Li, Y.; He, Y. & Mei, S. (2008). A highly parallel joint VLSI architecture for transforms in H.264/AVC. Journal of Signal Processing Systems, Vol.50, No.1, (January 2008), pp. 19–32, doi: 10.1007/s11265-007-0111-4, ISSN: 1939-8115*

Authors

S.Susrutha Babu was born in India, A.P. He received the B.Tech degree from JNTU, A.P, and M.Tech degree from SRM University, Chennai, Tamil Nadu, India in 2008 and 2010 respectively. He worked as Assistant Professor in Electronics Engineering in Bapatla Engineering College for academic year 2010-2011 and from 2011 to till date working in K L University. He is a member of Indian Society for Technical Education and International Association of Engineers. His research interests include antennas, FPGA Implementation, Low Power Design and wireless communications and Digital VLSI. He has published articles in various international journals and Conference in IEEE.

S.Suparshya Babu was born in India, A.P. He received the B.Tech degree from JNTU, A.P, and M.Tech degree from SRM University, Chennai, Tamil Nadu, India in 2008 and 2010 respectively. He worked as Assistant Professor in Electronics Engineering in Bapatla Engineering College for academic year 2010-2011 and from 2011 to till date working in K L University. He is a member of Indian Society for Technical Education and International Association of Engineers. His research interests include antennas, FPGA Implementation, Low Power Design and wireless communications and Robotics. He has published articles in various international journals and Conference in IEEE.

Mr. P.Satyanarayana was born on 20th May, 1977 in A.P, India. He received the B.Tech degree from Nagarjuna University in 2000, A.P and M.Tech degree from JNTU Kakinada in 2004. He is working as Associate Professor in ECE dept at K L University. His interested fields include wireless communication, VLSI, DSP and Embedded Systems. His research area is Wireless Communication. He has published articles in various international journals.

Alekhya Madivada was born in A.P,India. She received the **B.Tech** degree in **Electronics & communications Engineering** from Jawaharlal Nehru Technological University in 2008. Presently she is pursuing M.Tech embedded systems Design in , GEC. His research interests include FPGA Implementation, Low Power Design.

T. Krishna Karthik: was born in Gudivada, Krishna(dist),AP, India. He received B.Tech. in Electronics & Communication Engineering from GEC,AP. India Presently he is pursuing M.Tech VLSI Design in KL University, Vijayawada, AP, India. He has undergone 2 international conferences and 2 publishment in IEEE.

Arafath Ali was born in A.P,India. He received the B.Tech degree in Electronics & communications engineering from Jawaharlal Nehru Technological University in 2010. Presently he is pursuing M.Tech VLSI Design in Anurag Engineering College. His research interests include Low Power VLSI Design.