Available online at: http://www.ijmtst.com/vol5issue05.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 05, Issue No: 05, May 2019

Design of a Scalable Approximate DCT Architecture for Efficient HEVC Compliant Video Coding Applications

G Nageswari[1] | G V Ravi Kumar[2]

[1]PG Scholar,ECE Department, BVSR College of Engineering , Andhra Pradesh, India.
[2]Associate Professor, ECE Department, BVSR College of Engineering, Andhra Pradesh, India.

To Cite this Article
G Nageswari and G V Ravi Kumar, "Design of a Scalable Approximate DCT Architecture for Efficient HEVC Compliant Video Coding Applications", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 05, May 2019, pp.-16-23.

Article Info
Received on 09-April-2019, Revised on 28-April-2019, Accepted on 06-May-2019.

ABSTRACT

An approximate kernel for the discrete cosine transform (DCT) of length four comes from the 4 point DCT outlined by the High efficiency Video coding (HEVC) standard, and used that for the computation of DCT and inverse DCT (IDCT) of power of 2 lengths. There are 2 reasons to think about the DCT of length four because the basic module. Firstly, it permits to compute DCTs of length 4, 8, 16, and thirty two prescribed by HEVC. Moreover, the DCTs generated by 4 point DCT not solely involve lower complexity however additionally supply higher compression performance. Comparing to existing method, DCT offer better Compression performance. The proposed method can perform HEVC compliant video coding. Unified forward and inverse rework architecture is additionally planned wherever the hardware complexness is reduced by sharing of hardware between DCT and IDCT computation. The planned approximation has nearly an equivalent arithmetic complexness and hardware demand as those of recently planned connected ways, but involves considerably less error energy. In this paper, an energy and area efficient VLSI architecture of an HEVC-compliant inverse transform and dequantization engine is presented. We implement a pipelining scheme to process all transform sizes at a minimum throughput of pixel/cycle with zero-column skipping for improved throughput.

Index Terms: *DCT, HEVC, DFT, IAU, MPEG etc.*

I. INTRODUCTION

The High Efficiency Video Coding (HEVC) standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCTVC). The first edition of the HEVC standard was finalized in January 2013, resulting in an aligned text that will be published by both ITU-T and ISO/IEC. Additional work is planned to extend the standard to support several additional application scenarios, including extended-range uses with enhanced precision and color format support, scalable video coding, and 3-D/stereo/multi-view video coding. In ISO/IEC, the HEVC standard will become MPEG-H Part 2 (ISO/IEC 23008-2) and in ITU-T it is likely to become ITU-T Recommendation H.265. Video coding standards have evolved primarily through the development of the well-known ITUT and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and

MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) standards. The two standards that were jointly produced have had a particularly strong impact and have found their way into a wide variety of products that are increasingly prevalent in our daily lives. Throughout this evolution, continued efforts have been made to maximize compression capability and improve other characteristics such as data loss robustness, while considering the computational resources that were practical for use in products at the time of anticipated deployment of each standard.

The Discrete cosine transform (DCT) plays a vital role in video compression due to its near optimal de correlation efficiency. Several variations of integer DCT have been suggested in the last two decades to reduce the computational complexity. The new H.265/High Efficiency Video Coding (HEVC) standard has been recently finalized and poised to replace H.264/AVC. Some hardware architectures for the integer DCT for HEVC have also been proposed for its real time implementation decomposed the DCT matrices into sparse sub matrices where the multiplications are avoided by using the lifting scheme used the multiplier less multiple constant multiplication (MCM) approach for four-point and eight-point DCT, and have used the normal multipliers with sharing techniques for 16 and 32-point DCTs have used Chen's factorization of DCT where the butterfly operation has been implemented by the processing element with only shifters, adders, and multiplexors proposed a unified structure to be used for forward as well as inverse transform after the matrix decomposition. One key feature of HEVC is that it supports DCT of different sizes such as 4, 8, 16, and 32. Therefore, the hardware architecture should be flexible enough for the computation of DCT of any of these lengths. The existing designs for conventional DCT based on constant matrix multiplication (CMM) and MCM can provide optimal solutions for the computation of any of these lengths, but they are not reusable for any length to support the same throughput processing of DCT of different transform lengths. Considering this issue, we have analyzed the possible implementations of integer DCT for HEVC in the context of resource requirement and reusability, and based on that, we have derived the proposed algorithm for hardware implementation. We have designed scalable and reusable architectures for 1-D and 2-D integer DCTs for HEVC that could be reused for any of the prescribed lengths with the same throughput of processing irrespective of transform size. In this project, algorithms are used for hardware implementation of the HEVC integer DCTs of different lengths 4, 8, 16, and 32. H illustrate the design of the proposed architecture for the implementation of four-point and eight-point integer DCT along with a generalized design of integer DCT of length N, which could be used for the DCT of length N = 16 and 32. Moreover, it demonstrates the reusability of the proposed system. Here, power-efficient designs of transposition buffers for full-parallel and folded implementations of 2-D Integer DCT are used. Here a bit-pruning scheme for the implementation of integer DCT and present the impact of pruning on forward and inverse transforms are used finally comparing the synthesis result of the proposed architecture with those of existing architectures for HEVC.

II. LITERATURE SURVEY

2.1. Approximation of Algorithms

The discrete cosine transform (DCT) is popularly used in image and video compression. Since the DCT is computationally intensive, several algorithms have been proposed in the literature to compute it efficiently. Recently, significant work has been done to derive approximate of 8-point DCT for reducing the computational complexity. The main objective of the approximation algorithms is to get rid of multiplications which consume most of the power and computation-time, and to obtain meaningful estimation of DCT as well.

a. Signed DCT and Series of Methods

The proposed they signed DCT (SDCT) for 8x8sign, i.e., 1. Bouguezel-Ahmad-Swamy (BAS) have proposed a series of methods. They have provided a good estimation of the DCT by replacing the basis vector elements by 0, 1/2, 1. In the same vein, Bayer and Cintra have proposed two transforms derived from 0 and 1 as elements of transform kernel, and have shown that their methods perform better than the previous method, particularly for low- and high-compression ratio scenarios.

2.3. Integer Transforms

The need of approximation is more important for higher-size DCT since the computational complexity of the DCT grows nonlinearly. On the other hand, modern video coding standards such as high efficiency video coding (HEVC) uses DCT of

larger block sizes (up to 32x32) in order to achieve higher compression ratio. But, the extension of the design strategy used in H264 AVC for larger transform sizes, such as 16-point and32-point is not possible. Besides, several image processing applications such as tracking and simultaneous compression and encryption require higher DCT sizes. In this context, Cintra has introduced a new class of integer transforms applicable to several block-lengths. Cintra have proposed a new 16 x16 matrix also for approximation of 16-pointDCT, and have validated it experimentally. Recently, two new transforms have been proposed for 8-point DCT approximation: Cintraet al. have proposed a low-complexity 8-pointapproximate DCT based on integer functions and Potluri et al. have proposed a novel 8-point DCT approximation that requires only 14 additions.

2.4 UNIFIED FORWARD INVERSE TRANSFORM ARCHITECTURE FOR HEVC

The upcoming HEVC video coding standard supports many transform sizes ranging from 4-point to 32-point in square and rectangular form. Multiple transform sizes improve coding efficiency, but also increase the implementation complexity. Furthermore both forward and inverse transforms need to be supported in various consumer devices. This paper presents unified forward inverse transform architecture for HEVC. The unified architecture makes use of symmetry properties that exist in the HEVC forward and inverse transform matrices to achieve hardware sharing across different transform sizes and also between forward and inverse transforms. It uses 43-45% less area than separate forward and inverse core transform implementations.

III. RELATED WORK

3.1 Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is popularly utilized in image and video compression. Since the DCT is calculatedly accurate, many algorithms are planned within the literature to work out it with accurate. Recently, vital work has been done to derive approximate of 8-point DCT for reducing the computational complexness. There is vital scope for reduction of power consumption in video codec by algorithmic rule and design level performance. In algorithmic rule level, procedure complexness will be considerably reduced by appropriate approximation. Algorithm architecture co design on the opposite hand will give vital reduction in space, computation time, and therefore the general energy consumption in video secret writing and decipherment. it's found that the distinct circular function rework (DCT), inverse DCT (IDCT), and filtering operation will tolerate some errors, and so can enable some approximation. For mobile and time period applications, number DCT is needed to be enforced as hardware accelerator. The HEVC standard permits to use DCT of lengths of N = four, 8, 16, and 32. Therefore, a simple hardware implementation would need a separate hardware section for the DCTs of different lengths. to cut back the silicon space, it's needed to implement the DCTs of various lengths during a reconfigurable hardware which might be designed to calculate the DCT of any of the specified lengths. There are 2 main problems encountered during mapping of DCT algorithms to reconfigurable hardware architectures. But the prevailing DCT algorithms don't give the most effective of all the above 3 necessities. a number of the prevailing strategies are. a unit deficient in terms of quantify ability, generalization for higher sizes, and orthogonal. We intend to maintain orthogonal in the approximate DCT for 2 reasons. Firstly, if the transform is orthogonal, we are able to continuously notice its inverse, and the kernel matrix of the inverse rework is obtained by simply transposing the kernel matrix of the forward rework. This feature of inverse rework may well be wont to cipher the forward and inverse DCT by similar computing structures.

$$C_N = P_N \begin{pmatrix} \hat{C}_{\frac{N}{2}} & 0 \\ 0 & S_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{2}} & J_{\frac{N}{2}} \\ J_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{pmatrix}$$

$$C_4 = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}$$

$$\tilde{C}_4(i,j) = 32 \times \mathcal{R}\left(\frac{C_4(i,j) + 32 \times \mathrm{sgn}(C_4(i,j))}{64}\right)$$

$$\tilde{C}_4 = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 64 & 32 & -32 & -64 \\ 64 & -64 & -64 & 64 \\ 32 & -64 & 64 & -32 \end{bmatrix}$$

Moreover, in case of orthogonal transforms, similar quick algorithms area unit applicable to each forward and inverse transforms.

3.2 The High Efficiency Video Coding(HEVC) standard

High Efficiency Video Coding (HEVC) is currently being prepared as the newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The main goal of the HEVC standardization effort is to enable significantly improved compression performance relative to existing standards—in the range of 50% bit-rate reduction for equal perceptual video quality. This paper provides an overview of the technical features and characteristics of the HEVC standard.

The upcoming HEVC video coding standard supports many transform sizes ranging from 4-point to 32-point in square and rectangular form. Multiple transform sizes improve coding efficiency, but also increase the implementation complexity. Furthermore both forward and inverse transforms need to be supported in various consumer devices. This paper presents unified forward inverse transform architecture for HEVC. The unified architecture makes use of symmetry properties that exist in the HEVC forward and inverse transform matrices to achieve hardware sharing across different transform sizes and also between forward and inverse transforms. It uses 43-45% less area than separate forward and inverse core transform implementations.

This paper describes the core transforms specified for the high efficiency video coding (HEVC) standard. Core transform matrices of various sizes from 4 to 32 were designed as finite precision approximations to the discrete cosine transform (DCT). Also, special care was taken to allow implementation friendliness, including limited bit depth, preservation of symmetry properties, embedded structure and basis vectors having almost equal norm. The transform design has the following properties: 16 bit data representation before and after each transform stage (independent of the internal bit depth), 16 bit multipliers for all internal multiplications, no need for correction of different norms of basis vectors during quantization/de-quantization, all transform sizes above 4 can reuse arithmetic operations for smaller transform sizes, and implementations using either pure matrix multiplication or a combination of matrix multiplication and butterfly structures are possible. The transform design is friendly to parallel processing and can be efficiently implemented in software on SIMD processors and in hardware for high throughput processing.

IV. PROPOSED METHOD

4.1 APPROXIMATED FOUR POINT DCT

The proposed architecture for four-point integer DCT is shown in Fig.1. It consists of an input adder unit (IAU), a shift-add unit (SAU), and an output adder unit (OAU). The IAU computes a(0), a(1), b(0), and b(1) according to STAGE- 1 of the algorithm as described in Table I. The computations of ti,36 and ti,83 are performed by two SAUs according to STAGE-2 of the algorithm. The computation of t0, 64 and t1, 64 does not consume any logic since the shift operations could be rewired in hardware.

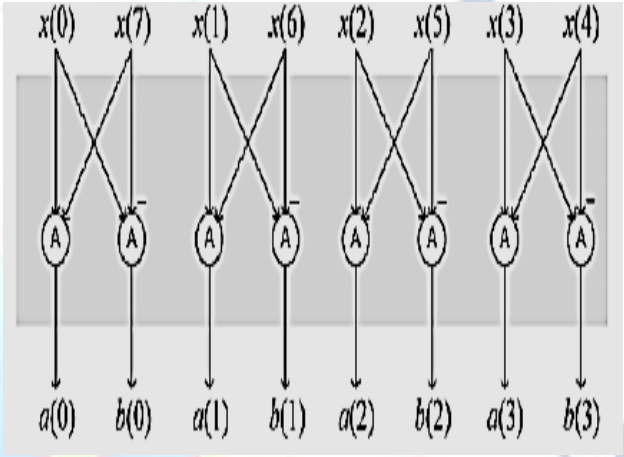


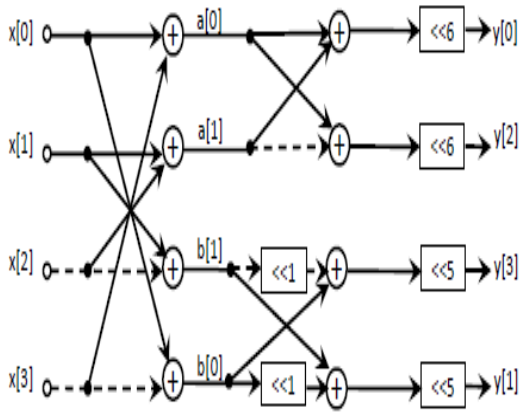
Fig-1.Structure of IAU

The 4 point approximate integer DCT kernel given by can be written as

$$\begin{bmatrix} y(0) \\ y(2) \end{bmatrix} = 64 \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \end{bmatrix}$$

$$\begin{bmatrix} y(1) \\ y(3) \end{bmatrix} = 32 \times \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} b(0) \\ b(1) \end{bmatrix}$$

where a(0) = x(0)+x(3); a(1) = x(1)+x(2); b(0) = x(0)+ x(3); and b(1) = x(1)+x(2). A structure for the computation of 4-point approximate DCT can be derived as shown in it. Dashed arrows represent multiplications by -1.



The structure of SAU is shown in Fig.2. Outputs of the SAU are finally added by the OAU according to STAGE-3 of the algorithm.

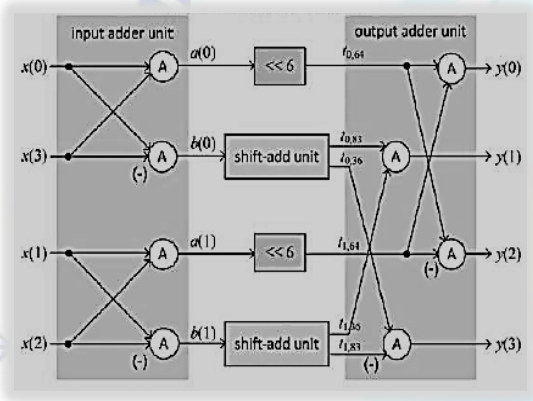


Fig-2.Proposed architecture of four-point integer DCT

Using the proposed scheme, the 8-point approximate DCT can be obtained by

$$\begin{bmatrix} t(0) \\ t(1) \\ t(2) \\ t(3) \end{bmatrix} = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 64 & 32 & -32 & -64 \\ 64 & -64 & -64 & 64 \\ 32 & -64 & 64 & -32 \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ s(2) \\ s(3) \end{bmatrix}$$

where, t(0) = y(0), t(1) = y(2), t(2) = y(4), t(3) = y(6) in upper DCT unit as shown in it.

and t(0) = y(1), t(1) = y(3), t(2) = y(5), t(3) = y(7)) in lower DCT unit as shown in it.

$$s(0) = x(0) + x(7), s(1) = x(1) + x(6$$
$$s(2) = x(2) + x(5), s(3) = x(3) + x(4$$

$$s(0) = x(0) - x(7), s(1) = x(1) - x(6$$
$$s(2) = x(2) - x(5), s(3) = x(3) - x(4$$

The generalized architecture for N-point integer DCT based on the proposed algorithm is shown in Fig.3. It consists of four units, namely the IAU, (N/2)-point integer DCT unit, SAU, and OAU. The IAU computes a(i) and b(i) for i = 0, 1, ...,N/2 − 1 according to STAGE-1 of the algorithm of Section II-B. The SAU provides the result of multiplication of input sample with DCT coefficient by STAGE-2 of the algorithm. Finally, the OAU generates the output of DCT from a binary adder tree of log2 N − 1 stages, respectively, illustrates the structures of IAU, SAU, and OAU in the case of eight-point integer DCT. Four SAUs are required to compute ti,89, ti,75, ti,50, and ti,18 for i = 0, 1, 2, and 3 according to STAGE-2 of the algorithm. The outputs of SAUs are finally added by two-stage adder tree according to STAGE- 3 of the algorithm. Structures for 16- and 32-point integer DCT can also be obtained similarly.

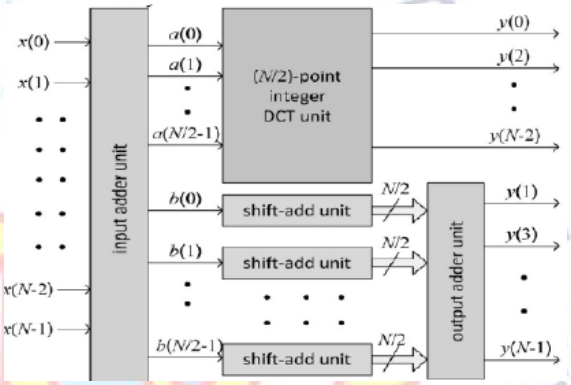


Fig-3.Proposed generalized architecture for integer DCT of lengths N = 8.

4.2 RECONFIGURABLE ARCHITECTURE FOR HEVC-COMPLIANT 2-D APPROXIMATE DCTS

A reconfigurable architecture for the approximate 8 point DCT is shown in it. It consists of two 4point approximate DCT units and an 8 point IAU that generates input operands for the pair of 4point DCT units. 8 MUXes are used to select appropriate input depending on whether it is used for 8point DCT calculation or for 4point DCT calculation. The control bit sel8 = 0 or 1 is used for the computation of 4point or 8 point DCTs, respectively. The OPU uses 8 MUXes to select and re-order the output depending on the selected DCT size. Moreover, the proposed reconfigurable architecture could be used for the computation of 2D DCT. A N-point 2 D DCT could be computed by row-column decomposition technique by using two Npoint 1D DCT stages. In the first stage 1D DCT of each column of the input matrix is computed to generate intermediate outputs, and stored in a transposition buffer to allow the 1D DCT computation of each row of these intermediate outputs during the second stage of computation. Accordingly a control-bit stage = 0 or 1 is used by the stage selection unit of Fig.4 to perform the

computation of the first or the second stage of 2D DCT, respectively. The architecture of 2D DCT can be obtained by using a folded structure or a full-parallel structure. Folded structure consists of one 1D DCT module of Fig.4 and a transposition buffer, while full-parallel structure consists of two 1D DCT modules

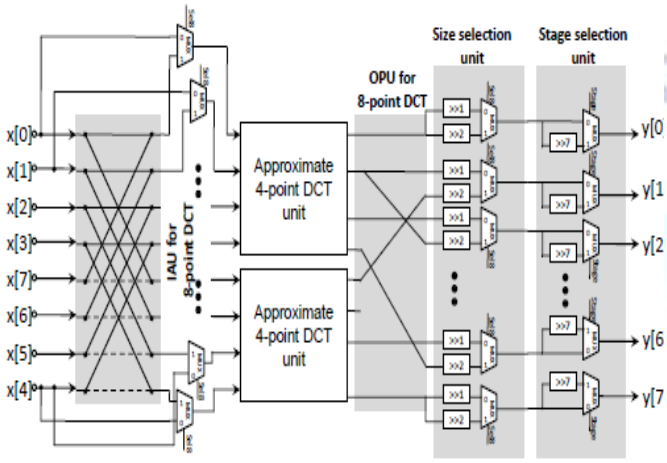


Fig-4.Reconfigurable architecture for HEVC-compliant 2-D approximate DCT

To meet the scaling requirement of HEVC and in order to preserve the norm of residual block, two scaling factors ST1 = 2(B+M-9) and ST2 = 2(M+6) are defined after the first and the second stages of 2-D DCT computation, respectively, as discussed in it. The bit-depth B of the video sequence is taken to be 8. Therefore, after the first forward DCT, 1 right-shift operation is done for 4point DCT (sel8 = 0), and 2 right-shift operations are done for 8point DCT (sel8 = 1). For the second stage of the DCT computation, 8 right-shift operations are needed for 4-point DCT (sel8 = 0), and 9 right-shift operations are done for 8point DCT (sel8 = 1). Since the stage selection unit follows the size selection unit which includes 1 or 2 right shift operations, only 7 right-shift operations are needed for the computation of second stage. Hence, scaling by ST1 and ST2 is realized jointly by the size selection and stage selection units. The reconfiguration scheme can be extended for higher DCT sizes.

4.3 UNIFIED INVERSE AND FORWARD RECONFIGURABLE DCT ARCHITECTURE

For mobile applications wherever devices ought to support playback video together with video capture and recording, forward and inverse transforms ought to be enforced within the same device. Some works are done recently to appreciate each forward and inverse work within the same device. The objective of unified architectures is to maximally share the hardware by the forward and therefore the inverse transforms.

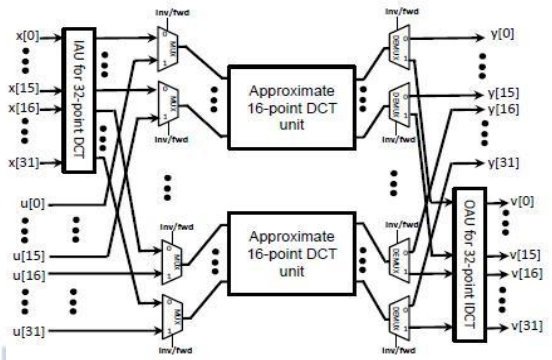


Fig-5:**UNIFIED INVERSE AND FORWARD RECONFIGURABLE DCT ARCHITECTURE**

V. RESULTS ANALYSIS

5.1 Simulation Results

4-POINT DCT:

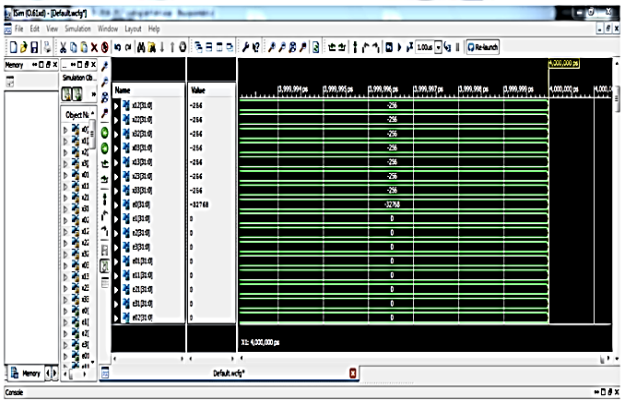


Fig-6: 4-point DCT simulation results

8-POINT DCT:

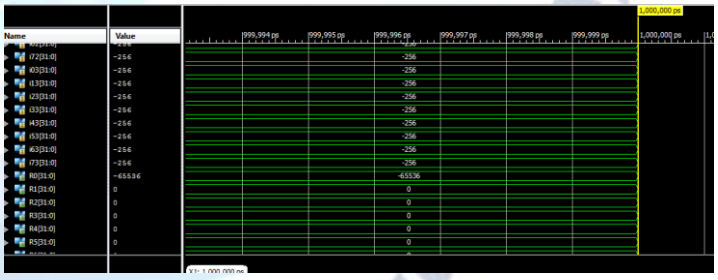


Fig-7: 8-point DCT simulation results

16-POINT DCT:

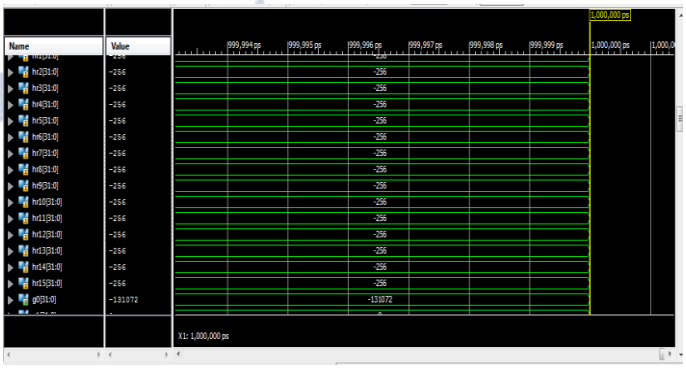


Fig-8: 16-point DCT simulation results

32-POINT DCT:

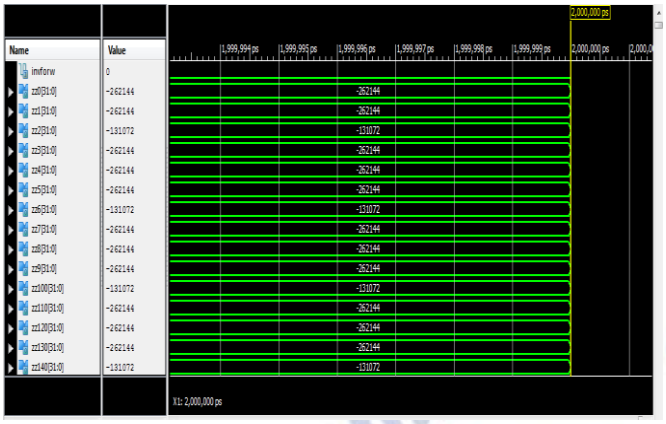


Fig-9: 32-point DCT simulation results

5.2 DEVICE UTILIZATION SUMMARY:

4-POINT DCT:

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	1408	5720	24%
Number of fully used LUT-FF pairs	0	1408	0%
Number of bonded IOBs	880	102	862%

Fig-10: 4-point DCT Device Utilization results

EIGHT PT DCT:

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	3552	5720	62%
Number of fully used LUT-FF pairs	0	3552	0%
Number of bonded IOBs	1760	102	1725%

Fig-11: 8-point DCT Device Utilization results

16-POINT DCT:

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	8576	5720	149%
Number of fully used LUT-FF pairs	0	8576	0%
Number of bonded IOBs	3520	102	3450%

Fig-12: 16-point DCT Device Utilization results

32-POINT DCT:

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	4044	5720	70%
Number of fully used LUT-FF pairs	0	4044	0%
Number of bonded IOBs	4097	102	4016%

Fig-13: 32-point DCT Device Utilization results

5.3 RTL SCHEMATIC:

UNIFIED DCT:

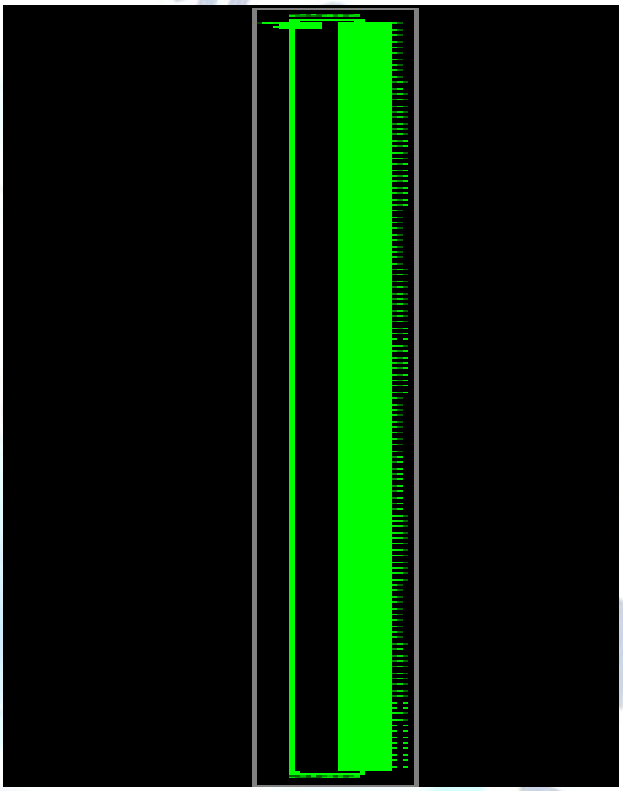


Fig-14: Unified DCT RTL Schematic

FOUR POINT DCT:

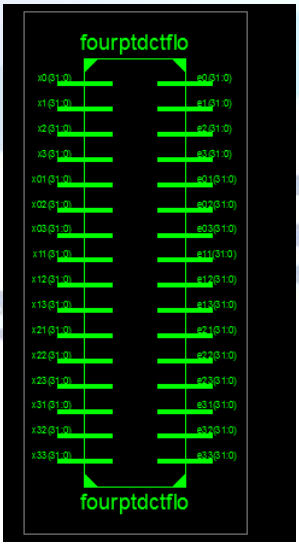


Fig-15: 4-point DCT RTL Schematic

EIGHT POINT DCT:

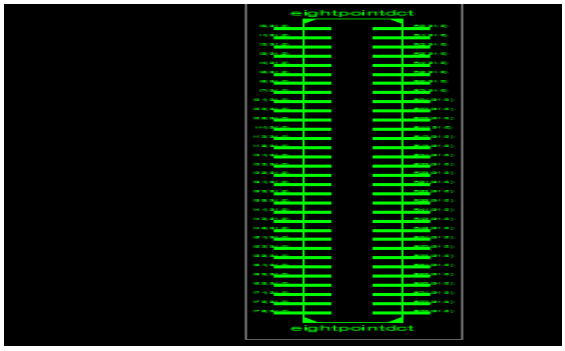


Fig-16: 8-point DCT RTL Schematic

SIXTEEN POINT DCT:

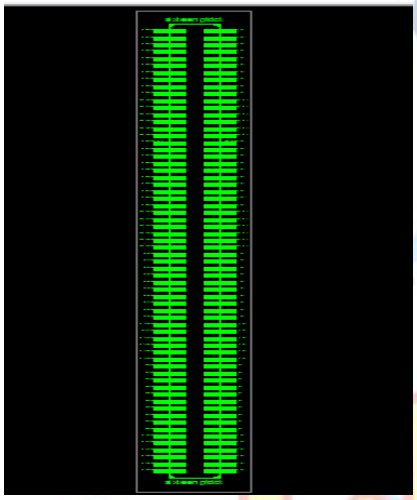


Fig-17: 16-point DCT RTL Schematic

VI. CONCLUSION

In this paper, we've projected a completely scalable HEVC compliant orthogonal number approximation of DCT of any power of two length, N>4, that gives versatile trade-off between space and time complexities, and would possibly merely be realized throughout reconfigurable hardware with terribly low reconfiguration overhead. The projected approximation provides better compressed image quality with similar arithmetic complexity and hardware consumption compared with earlier approximation schemes. A completely scalable and reusable architectures are also projected for the computation of approximate DCT where 8 point DCT structure are accustomed compute a pair of 4 point DCT exploitation. The reuse structure of 32point DCT could be designed for parallel calculation of two 16-point DCTs or four 8 point DCTs or eight 4 point DCTs. The projected reuse Scalable design can help real-time writing for high- definition video sequence.

REFERENCES

[1] Sullivan, G.J. and Ohm, J. and Woo-Jin Han and Wiegand, T., Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Trans. Circuits and Syst. Video Technology, 2012, Vol. 22, N. 12, pp. 1649-1668,

[2] Maher Jridi, Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE, "A Scalable Approximate DCT Architectures for Efficient HEVC Compliant Video Coding", IEEETransactions on Circuits and Systems for Video Technology.

[3] Cisco-Networks, Visual Networking Index: Global Mobile Data TrafficForecast Update 20142019, 2015, 3 Feb,

[4] High efficiency video coding (HEVC), https://hevc.hhi. fraunhofer.de,2013, 20 Nov,

[5] Sullivan, G.J. and Ohm, J. and Woo-Jin Han and Wiegand, T., Overviewof the High Efficiency Video Coding (HEVC) Standard, IEEE Trans. Circuitsand Syst. Video Technol, 2012, Vol. 22, N. 12, pp. 1649-1668,

[6] MPEG Systems Technologies - Part 11: Energy-Efficient Media Consumption(Green Metadata), ISO/IEC JTC1/SC29/WG11, Final Draft InternationalStandard 23001-11 (w14853), Information-Technology, 2014,Oct,

[7] Fernandes, F.C. and Ducloux, X. and Zhan Ma and Faramarzi, E. andGendron, P. and Jiangtao Wen, The Green Metadata Standard for Energy-Efficient Video Consumption, Multimedia, IEEE Transactions on, 2015,Jan, Vol. 22, N. 1, pp.80-87,

[8] Bossen, F. and Bross, B. and Suhring, K. and Flynn, D., HEVC Complexityand Implementation Analysis, IEEE Trans. Circuits and Syst. VideoTechnol, 2012, Vol. 22, N. 12, pp. 1685-1696,

[9] Jridi, M. and Alfalou, A. and Meher, P. K., A Generalized Algorithm and Reconfigurable Architecture for Efficient and Scalable Orthogonal Approximation of DCT, Circuits and Systems I: Regular Papers, IEEE Transactions on, 2015, Feb., Vol. 62, N. 2, pp. 449-457,

[10] Bouguezel, S. and Ahmad, M.O. and Swamy, M.N.S., Binary Discrete Cosine and Hartley Transforms, Circuits and Systems I: Regular Papers,IEEE Transactions on, 2013, Vol. 60, N. 4, pp. 989-1002