

# FPGA Implementation of Low Complexity Video Encoder using Optimized 3D-DCT

Rajalekshmi R  
Embedded Systems  
Sree Buddha College of Engineering, Pattoor  
India

Arya Lekshmi M  
Electronics and Communication Engineering  
Sree Buddha College of Engineering,  
Pattoor  
India

**Abstract**— Discrete Cosine Transform (DCT) is an essential tool of most of the image and video compression standards, because of its better energy compaction properties. As the demands for the two-way video transmission and video messaging over mobile communication systems increasing, the encoding complexity needs to be optimized. The three-dimensional discrete cosine transform (3D-DCT) and its inverse (3D-IDCT) can be used as an alternative to motion compensated transform coding, because it extends the spatial compression property of 2D-DCT to spatial-temporal compression of video data. In the proposed architecture, a low complexity video encoder using 3D-DCT has been presented. This method converts video data into three-dimensional video cube of  $8 \times 8 \times 8$  pixels and 3D-DCT is then performed, followed by quantization, zig-zag scanning and entropy encoding. The three-dimensional DCT circuit can be realized using few additions and subtractions, thus increasing the area efficiency with low complexity. The proposed architecture is coded in Verilog HDL, synthesized in Xilinx ISE design suite 14.2 and physically realized as a digital prototype circuit using Xilinx Virtex-5 FPGA.

**Keywords**— Video Compression; Three-Dimensional Discrete Cosine Transform; Coefficient Quantization; Multiplication-Free Transform; FPGA implementation.

## I. INTRODUCTION

Video compression/decompression are important process in many applications including internet TV, video conferencing etc., which use DCT based transform because of its outstanding energy compaction properties. The DCT transform eliminates duplication of data in frequency domain. Most of the video compression standards to date use 2D-DCT based transform for reducing the spatial redundancies by eliminating inter-correlations between each pixels within the video frames and motion estimation/compensation for reducing the temporal redundancies between the video frames. However, these algorithms are very hard to implement in hardware and no symmetry exist between encoder and decoder circuit. Motion estimation includes complex calculations for the computation of motion vector and need more sophisticated algorithms. Hence the implementation becomes more complex

Alternative approach is to use a transform-based approach for the encoding of subsequent frames [2], which is the 3D-DCT. The 3D-DCT removes not only the spatial redundancy but also the temporal redundancy of the video frames. The video compression standards such as H.264/AVC [3], MPEG,

H.261 etc. uses hybrid coding (i.e., 2D-DCT followed by prediction/motion compensation) which can be potentially replaced by the 3D-DCT algorithm.

For the fast computation of DCT, numerous efficient algorithms have been proposed. The floating point DCT packs most of the energy to the low frequency region, which is well suited for the image processing applications [8]. But the floating point operations do not meet the real time constraints and also increases the circuit complexity. For the minimization of the number of floating point operations, approximate transforms are used [4], [6], [7], which in turn reduces the computational as well as circuit complexity. Today's most of the compression standards use approximate DCT transforms.

In this context, a large number of approximate DCT have been proposed. R. J Cintra et al. introduced a method for finding DCT by using 22 addition operations, based on the round off function [6]. In this method, the resulting 8-point approximation matrix contains only 0s and  $\pm 1$ s. R. K. Senapati et al. proposed a low complexity  $8 \times 8$  transform matrix for fast image compression [5]. This transform requires only 14 additions and two shift operations. T. D. Tran et al. introduced binDCT, which is an  $8 \times 8$  bi-orthogonal transform. It requires 31 additions and 14 shift operations [8]. The binDCT shows finer approximations to exact DCT. R. J. Cintra et al. proposed the architecture for 1D-DCT, which requires only 14 additions [1]. This work has been extended to 2D-DCT with the use of a transposition buffer. Several studies were performed for the simplification of 3D-DCT, because of the increasing demands of three dimensional applications. The 3-D DIF VR algorithm [9] introduced by S. Boussakta et al., for the computation of 3D DCT-II. This algorithm reduces the number of multiplications and retains the same number of additions. 3D-DIF VR algorithm requires 5568 additions and 1344 multiplication for processing one  $8 \times 8 \times 8$  video cube.

The objective of the proposed method is to introduce a new optimized 3D-DCT approximation that possesses extremely low arithmetic complexity, which requires less number of addition operations, and to develop a video encoder. The paper is organized as follows. Section II explains the 3D-DCT in detail. Section III describes the 1D-DCT. Section III presents physical architecture of 3D-DCT and its realization. Section IV explains the application of the proposed design in video encoder. A comparison of the video encoder with and without motion estimation along with the results is given in Section V.

## II. 3D-DCT ALGORITHM

The human eyes are highly sensitive to low frequency components; because of the energy compaction property of the DCT, which concentrates most of the information in a few low frequency components. In addition, it is signal-independent and can be computed efficiently by fast algorithms. For these reasons, the DCT is widely used in image and video compression. Since the common 3D-DCT kernel is separable, the 3D-DCT is usually obtained by applying the 1D-DCT along each of the three dimensions. The  $N \times N \times N$  3D-DCT can be defined [2] as:

$$G(x, y, z) = \sqrt{\frac{8}{RCF}} W_x W_y W_z \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \sum_{f=0}^{F-1} g(r, c, f) C_x^{2r+1} C_y^{2c+1} C_z^{2f+1} \quad (1)$$

Where, R, C, F is the number of rows, columns, frames respectively in each cube,  $G(x,y,z)$  is the DCT domain data,  $g(r,c,f)$  is the time domain data,  $W_n$  is given by:

$$W_n = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } x = 0 \\ 1 & \text{for } x \neq 0 \end{cases} \quad (2)$$

And

$$C_x^{2r+1} = \cos \left[ \frac{(2r+1)x\pi}{2R} \right] \quad (3)$$

Where  $r = 0, 1, 2, \dots, R$ . The same definition is applied on  $C_y^{2c+1}, C_z^{2f+1}$

The reverse 3D-DCT or 3D-IDCT can be written as:

$$g(r, c, f) = \sqrt{\frac{8}{RCF}} \sum_{x=0}^{R-1} \sum_{y=0}^{C-1} \sum_{z=0}^{F-1} G(x, y, z) W_x W_y W_z C_x^{2r+1} C_y^{2c+1} C_z^{2f+1} \quad (4)$$

For the implementation of (1) needs a lot of hardware and takes a lot of time to complete the computation. If  $R=C=F=B$ , then the number of additions and the number of multiplications are  $B^3-1, B^3+B^2+B$  respectively. The complexity per coefficient is  $O(B^3)$ . So the time and hardware needed for mathematical manipulation should be reduced. For that temporal-row-column decomposition algorithm has been chosen for the calculation of 3D-DCT.

The DCT equation exhibits the separability property, so it can be decomposed into three orthogonal and symmetric functions. The equation 1 can be rewrite as

$$G(x, y, z) = \sum_{r=0}^{R-1} \sqrt{\frac{2}{R}} W_x C_x^{2r+1} \left\{ \sum_{c=0}^{C-1} \sqrt{\frac{2}{C}} W_y C_y^{2c+1} \left\{ \sum_{f=0}^{F-1} \sqrt{\frac{2}{F}} W_z C_z^{2f+1} g(r, c, f) \right\} \right\} \quad (5)$$

The equation (5) gives three identical one-dimensional equations. So the 3D-DCT can be calculated by first computing the 1D-DCT of the rows followed by calculating 1D-DCT of columns. The resultant DCT coefficients are again subjected to 1D-DCT, forming fully transformed DCT cube. This method reduces the number of multiplications and number of additions to  $R+C+F, R+C+F-3$  respectively for the computation of one DCT coefficient.

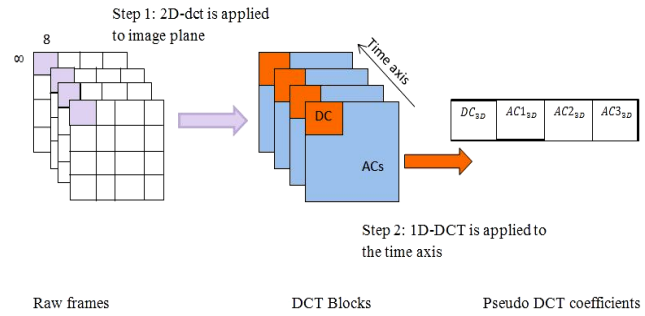


Fig. 1: Computation flow of 3D-DCT

The 3D-DCT can be computed as follows: the DCT and its inverse can be calculated by means of pseudo 3D-DCT, with the intention of reducing the computational complexity. The computational flow of 3D-DCT is illustrated in Fig. 1. First, perform 2D-DCT of the  $8 \times 8$  image. Pick the DC or AC coefficients of the block located in the same position of consecutive channels of the 3D block and then, transform these values to the DCT domain. After performing the step 2, new DC value and AC values are obtained. The 3D-DCT is computed by means of three 1D-DCT structures. The 1D-DCT transform and its architecture is as follows:

## III. 1D-DCT TRANSFORM

Most of the 8-point DCT approximation methods such as Bouguezel-Ahmad-Swamy Approximate DCT [7], CB-2011 Approximation[6], Modified CB-2011 approximation [4] etc. gives the transformation matrix format as,  $[D] * [LC]$

Where D, diagonal matrix contains irrational numbers in the form  $1/\sqrt{k}$ , where k is a small positive integer. The entries of the LC, the low-complexity matrix, comprise only powers of two in  $\{0, \pm 1/2, \pm 1, \pm 2\}$ . So that null multiplicative complexity is attained.

For deriving low-complexity approximate DCT, a search over the  $8 \times 8$  matrix space is done to find the candidate matrix with low computation cost. The good candidate matrix is the one which does not require multiplication operations. Thus encounters the following optimization problem:

$$M_p = \arg \min_{\text{cost}}(M) \quad (6)$$

Where  $M_p$  is the sought matrix and  $\min_{\text{cost}}(M)$  returns minimum arithmetic complexity of matrix, M.

Exact 8-point DCT matrix is given by,

$$U = \frac{1}{2} \cdot \begin{bmatrix} \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 \\ \alpha_0 & \alpha_2 & \alpha_4 & \alpha_6 & -\alpha_6 & -\alpha_4 & -\alpha_2 & -\alpha_0 \\ \alpha_1 & \alpha_5 & -\alpha_5 & -\alpha_1 & -\alpha_1 & -\alpha_5 & \alpha_5 & \alpha_1 \\ \alpha_2 & -\alpha_6 & -\alpha_0 & -\alpha_4 & \alpha_4 & \alpha_0 & \alpha_6 & -\alpha_2 \\ \alpha_3 & -\alpha_3 & -\alpha_3 & \alpha_3 & \alpha_3 & -\alpha_3 & -\alpha_3 & \alpha_3 \\ \alpha_4 & -\alpha_0 & \alpha_6 & \alpha_2 & -\alpha_2 & -\alpha_6 & \alpha_0 & -\alpha_4 \\ \alpha_5 & -\alpha_1 & \alpha_1 & -\alpha_5 & -\alpha_5 & \alpha_1 & -\alpha_1 & \alpha_5 \\ \alpha_6 & -\alpha_4 & \alpha_2 & -\alpha_0 & \alpha_0 & -\alpha_2 & \alpha_4 & -\alpha_6 \end{bmatrix}$$

Where,  $\alpha_p = \cos(2\pi(p+1)/32), p = 0, 1 \dots 6$

The number of retained coefficient is considered as the important parameter in the image compression. In several applications, a small number of coefficients are retained. In this transform it is 10. The 1D-DCT approximation is defined as:

$$U_p = D_p \cdot M_p = D_p \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Where,

$$D_p = \text{diagonal} ([1/\sqrt{8}, 1/\sqrt{2}, 1/\sqrt{2}/1/\sqrt{2}, 1/\sqrt{8}, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}])$$

Matrix  $M_p$  has entries in  $\{0, \pm 1\}$  and the transform matrix can be written as product of three matrices.

$$M_p = P_4 \cdot W_{12} \cdot W_{11} \cdot W_1, \text{ where}$$

$$W_{11} = \text{diag} \left( \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, I_4 \right)$$

$$W_{12} = \text{diag} \left( \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, -1, I_5 \right)$$

$$W_1 = \begin{bmatrix} I_4 & \bar{I}_4 \\ \bar{I}_4 & -I_4 \end{bmatrix}$$

And  $P_4$  is the permutation (1) (2 5 6 8 4 3 7).

Where,  $I_4$  is the identity matrix of order 4.

#### IV. DIGITAL ARCHITECTURES

The detailed architecture for 1D and 2D approximate 8-point DCT is shown in the Fig. 5 and Fig. 4 respectively and the architecture for 3D approximate DCT is being proposed, shown in the Fig. 2. The introduced architectures were then given to the Xilinx FPGA and the obtained results are included in the section VI.

##### A. Proposed Architecture for 3D-DCT

The proposed architecture 3D-DCT uses three parallel realizations of 1D-DCT approximation blocks (Fig. 3), requires only 2688 additions. The input of 3D-DCT is in a form of  $8 \times 8 \times 8$  video cube. The architecture is described as: first 1D approximate DCT block instantiation does the row-wise transform computation of the input 2D image while the second instantiation provides the column-wise transform computation of the intermediate result. Hence 2D-DCT is obtained. Likewise 2D-DCT is computed for 8 blocks of size  $8 \times 8$ , by instantiating 2D approximate blocks. After computing the 2D-DCT coefficients, again 1D-approximate DCT block is needed for the calculation of 3D-DCT. The  $r, c, f$  represents row, column, and frame respectively.

The 1D approximate DCT architecture is detailed in Fig. 4. Detailed architecture of 1D-DCT requires only 14 addition/subtraction operations, shown in Fig. 5. Without the use of multipliers, the delay and area can be reduced.

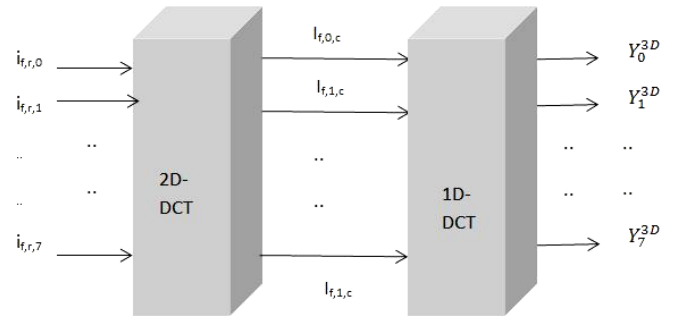


Fig. 2: Proposed 3D-DCT architecture

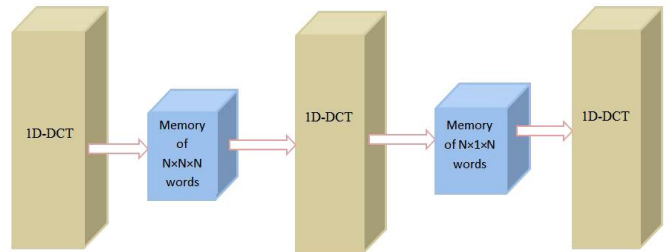


Fig. 3: Serial architecture of the 3D-DCT

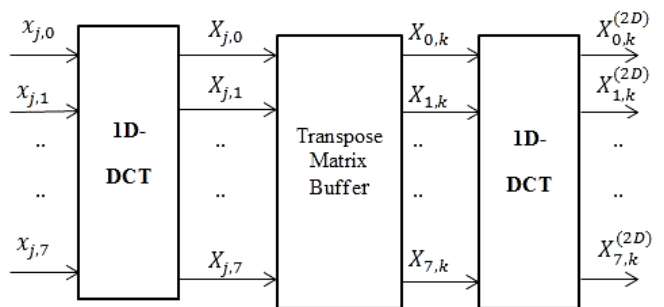


Fig. 4: Architecture of the 2D-DCT

A real-time row parallel transpose matrix buffer circuit is necessary between the 1D approximate DCT blocks, which ensures data ordering for converting the row-transformed data from the first DCT approximation unit to a transposed format. Then these results are given to the second 1D approximation DCT unit. The architecture for the transpose matrix buffer is shown in the Fig. 6.

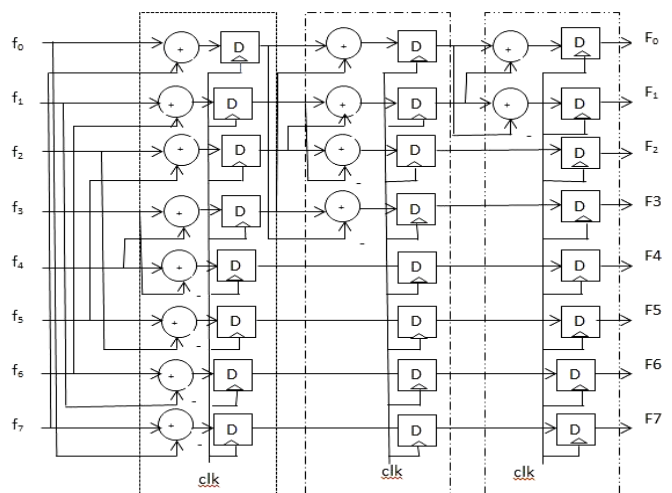


Fig. 5: Digital architecture for approximate 1D-DCT

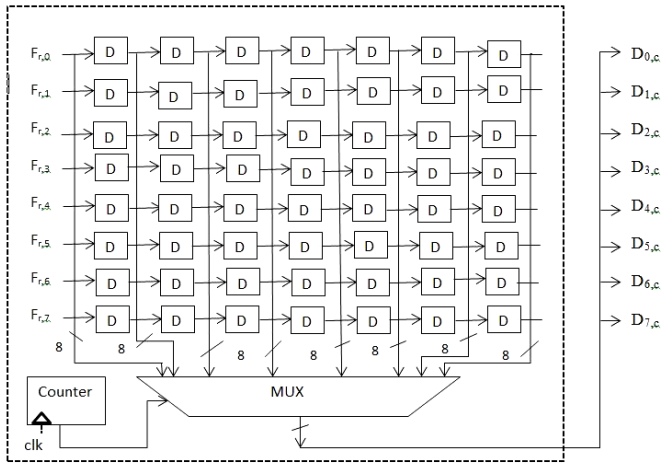


Fig. 6: Architecture of transpose matrix buffer

V. 3D-DCT BASED VIDEO ENCODER

As compared with the other compression standards such as MPEG, H.26x etc., this algorithm exhibit different principle for compressing the temporal information (motion estimation/compensation). The working of the encoder is as follows. The forward DCT converts 2D-image from spatial to frequency domain. The 3D-DCT based video compression algorithm takes full motion video as input. The input is then divided into groups of 8 frames, which can be considered as a three dimensional image with two spatial dimensions (x and y) and a temporal dimension (z). The Fig. 7 illustrates the formation of 8x8x8 video cubes.

Each 8x8x8 cubes are independently encoded using 3D-DCT, quantization, and entropy encoder. The block diagram of video encoder using 3D-DCT is illustrated in Fig. 8. Each 8x8x8 cube of 512 pixels is transformed into frequency domain using the forward 3D-DCT. The architecture for 3D-DCT is explained above. After DCT transformation most of the energy is confined in few low frequency coefficients. Majority of the high frequency coefficients are zero or have negligible value, so only 64 pixel values are needed to be considered for the next section.

In the next block, all the 64 DCT coefficients are quantized using 64 element quantization table. This step is also used for compression, which introduces minimum error while increasing the number of zero-value coefficients and discards visual information to which human eye is not sensitive. Quantization is performed according to the following equation:

$$Q_b = \left\lfloor \frac{F[x,y]}{Q[x,y]} \right\rfloor \tag{7}$$

Where, F(x,y) are the elements before quantization, Q(x,y) are the elements from the quantization table, and Q<sub>b</sub> are the quantized elements.

The result of the quantization step is the collection of small valued coefficients, in which a large number of values are zero. These results are then converted into a compact binary sequence using entropy encoder. The quantized DCT coefficients are re-ordered in a zig-zag manner using zig-zag scanning and re-ordered sequence will be in a form DC, AC1,

AC9... AC63. Then this 1D-data vector is subjected to the entropy encoder for compression.

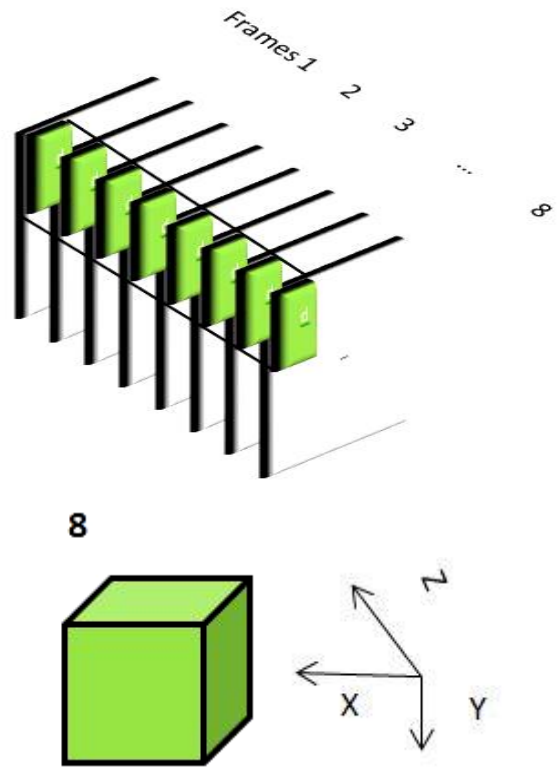


Fig. 7: Forming 8x8x8 video cube

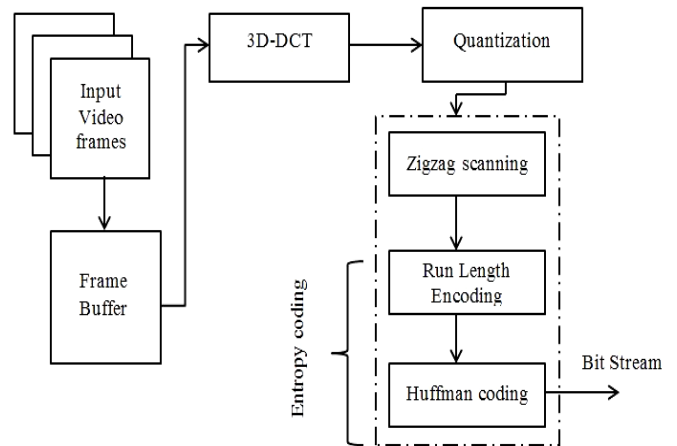


Fig. 8: Video encoder using 3D-DCT

The entropy encoder consists of run-length-encoding, and Huffman encoder. From the data vector, the run lengths of zero is computed and then to the Huffman encoder. The output of the entropy encoder gives a binary sequence which represents the compressed 8x8x8 block. The video decoder is designed in such a way that all the above said steps are inverted and implemented in reverse manner.

VI. FPGA IMPLEMENTATION AND DISCUSSIONS

The proposed architecture is coded in Verilog HDL, simulated using Xilinx ISE Design suite 14.2 and synthesized in Xilinx Virtex-5, XC5VLX110T-2FF1136 device. The obtained results are shown below. Fig. 9 shows simulation result of video encoder with 3D-DCT. The input is assumed as

an 8-bit resolution. The variable out\_sig gives the bit stream of the input video cube, when the variable out\_rdy is set. The serial architecture of 3D-DCT is employed here; so only three 1D-DCTs are used, which requires 64 clock cycles to process one 8×8×8 video cube. This architecture uses 659 registers to store the transformed results and the throughput is  $f_{clk}$ .

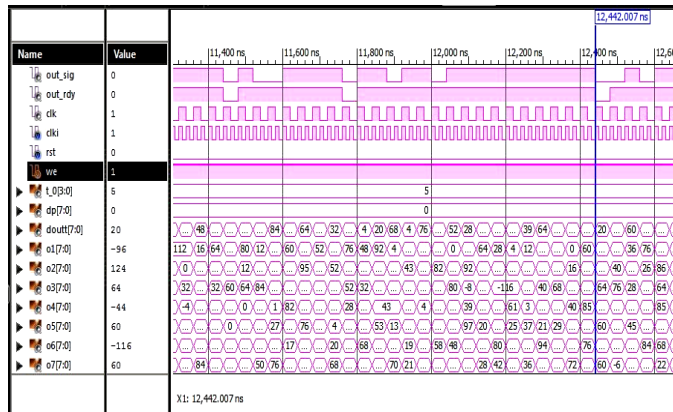


Fig. 9: Simulation result of video encoder with 3D-DCT

After synthesizing the Verilog code, it is then routed to the FPGA board, using JTAG based hardware co-simulation. JTAG is a serial communication standard, which enables boundary scan technology providing access to many logic signals of complex ICs.

TABLE I

HARDWARE RESOURCE CONSUMPTION USING XILINX VIRTEX-5 XC5VLX110T-2FF1136 DEVICE

	Video encoder with motion estimation	Video encoder with 3D-DCT
LUT	1313[1%]	3460[5%]
Bonded IOBs	6	6
FF	780[21%]	241[2%]
Registers	3065[6%]	659[0%]

The FPGA implementation can be evaluated by means of configurable LUTs, FF count, delay, maximum operating frequency, gives the hardware complexity as well as the real-time performance. 15.190ns, 14.798 are the obtained delay of

the video encoder using motion estimation, 3D-DCT-based video encoder respectively. The synthesis report of video encoder with 8×8×8 video cube as input and video encoder using 2D-DCT and motion estimation is tabulated in table I. The table I show that the hardware resource utilization of video encoder using 3D-DCT is lesser as compared to the other. Hence the area required is less. The proposed 3D-DCT architecture uses only adders, the complexity is also less.

## VII. CONCLUSION

The video encoder using optimized 3D-DCT is proposed and implemented in Xilinx Virtex-5 XC5VLX110T-2FF1136 device. The proposed 3D-DCT saves 75% of additions and 100% of multiplications. The result shows a low complexity and area efficient architecture, which can be used in multimedia portable systems where low power and area minimization are prominent. Hence the architecture is well suited for real time low power applications.

## REFERENCES

- [1] Uma Sadhvi Potluri, Arjuna Madanyake, Renato J. Cintra, Fábio M. Bayer, Sunera Kulasekara and Amila Edirisuriya, "Improved 8-point Approximate DCT for image and Video Compression Requiring Only 14 Additions", IEEE Transactions on circuits and systems, vol. 61, no. 6, June 2014
- [2] M. Bakr and A. E. Salama, "Implementation of 3D-DCT based video encoder/decoder system", Int. Sym. on signals, circuits, and systems, vol. 2, pp. 389-392, 2003
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, Jul. 2003
- [4] F. M. Bayer and R. J. Cintra, "DCT-Like transform for image compression requires only 14 additions", Electronics Letters, vol.48, no. 15, pp. 919-921, 2012
- [5] R. K. Senapati, U. C. pati, K. K. Mahapatra, "A low complexity orthogonal 8×8 transform matrix for fast image compression", in proc. Of the Annual IEEE India Conf. (INDICON), pp. 1-4, 2010
- [6] F. M. Bayer and R. J. Cintra, "A DCT approximation for image compression", IEEE Signal Processing Letters, vol. 18, no. 10, pp. 579-583, October 2011
- [7] S. Bouguezel, M. O. Ahmed, and M. N. S. Swamy, "Low-complexity 8×8 transform for image compression", Electronics letters, vol. 44, no. 21, pp. 1249-1250, 2008
- [8] T. D. Tran, "The binDCT: fast multiplierless approximation of the DCT", IEEE Signal Processing Letters, vol. 7, pp. 141-144, June 2000
- [9] S. Boussakta and H. O. Alshibami, "Fast algorithm for the 3-D DCT-II", IEEE Trans. on Signal Processing, vol. 52, pp. 992-1001, April 2004