2018 3rd International Conference for Convergence in Technology (I2CT)
The Gateway Hotel, XION Complex, Wakad Road, Pune, India. Apr 06-08, 2018

FPGA Implementation of Pipelined 8x8 2-D DCT and IDCT Structure for H.264 Protocol

Dr. Pankaj Kumar Srivastava[1], Prof. Anil Kumar Jakkani[2]
[1]Professor, ISB&M School of Technology, Pune,
[2]Assistant Professor, ISB&M School of Technology, Pune.

Abstract— **Discrete Cosine Transform (DCT) is the most widely used technique for image/video compression standards. In this paper, an FPGA based 8x8 forward Discrete Cosine Transform (DCT) and inverse Discrete Cosine Transform (IDCT) architectures are implemented for H.264 encoder and decoder respectively. It has been implemented using the pipelined structure for computing the Integer DCT and IDCT. The design is implemented without using any multiplication operation to reduce the complexity. H.264 is one of the recent video compression standards of the ITU-T Moving Picture Experts Group and Video Coding Experts Group. A pipelined architecture is used in this design to increase the speed of operation. The proposed resource effective architectures have been implemented and synthesized on xc2vp30 device that belongs to Vertex-2 Pro family. For Integer DCT architecture implementation, the resource utilization is 14% slices, 1% flip-flops, 11% LUTs, 51% IOBs of xc2vp30 device and reaches an operating frequency of 520.996 MHz. For Integer IDCT architecture implementation, the resource utilization is 28% slices, 2% flip-flops, 24% LUTs, 37% IOBs of xc2vp30 device and reaches an operating frequency of 525.22 MHz. In these designs, it takes only 49 clock cycles to generate the transformed outputs thus increasing the speed of operation.**

Keywords*— *Integer DCT, Inverse Integer DCT, H.264, Pipelined architecture, 2-D DCT and 2-D IDCT.

I. INTRODUCTION

DCT is an effective technique used in the image and video compression standards, such as MPEG [11], H.263 [14], H.264 [13] and JPEG [12]. Now a days the number of applications are increased the use of these video coding standard in many devices like mobile phones, digital camera and CCTV. So it is necessary to develop a high speed and low complexity DCT/IDCT module as one the prime components. To support low power applications, it is necessary to minimize the computational complexity as much as possible. In this design for achieving high speed of operation, pipelined architecture is used. The pipelined architecture also reduces the resource utilization thus reducing the overall FPGA [2] chip cost.

H.264 [6] standard provides the greatly improved quality at any given bit rate. Unlike the current video coding standards, a new transformation technique is used in H.264 that can calculate the results in integer form which is known as 2-Dimensional Integer DCT [4]. Its corresponding inverse transform is known as 2-Dimensional Integer IDCT. In a standard DCT/IDCT implementation, multipliers are used which increases the cost in terms of hardware and increase of complexity. Here 2D Integer DCT and IDCT are implemented using the adders instead of multipliers for the reduction of complexity.

II. 1-D INTEGER DCT AND IDCT ALGORITHM

2D-DCT can be implemented by using two 1-Dimensional DCT transforms sequentially. It can be implemented by employing the row and column decompositions of the given image matrix, one is applied along the column vector of image matrix followed by the row vector transform as shown in Fig. 1.

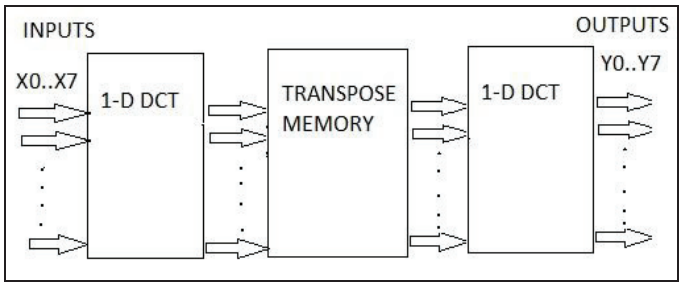

Fig. 1: 2-D DCT module.

Similarly 2-Dimensional IDCT can also be divided into two 1D-IDCT operations performed sequentially as shown in Fig 2.

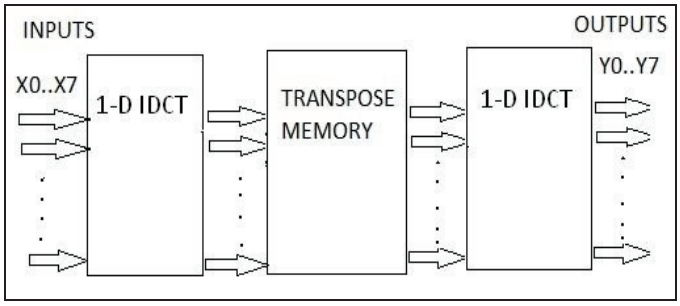

Fig. 2: 2-D IDCT module.

The N-point 1-Dimensional DCT and IDCT can be defined as follows [5].

Forward N-point DCT

$$X(k)= \frac{2}{N} c_k \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right], k=0,1,2,\ldots,N-1 \quad (1)$$

Inverse N-point DCT:

$$x(n) = \frac{2}{N}\sum_{k=0}^{N-1} c_k X(k)\cos\left[\frac{(2n+1)k\pi}{2N}\right], n = 0,1,\ldots\ldots,N-1 \quad (2)$$

Where $c_k = \begin{cases} 1/\sqrt{2}, k = 0 \\ 1, k \neq 0 \end{cases}$

The 2-D Forward DCT and the 2-D Inverse DCT [5] can be represented mathematically as follows.

Forward 2-D DCT

$$Y(u,v) = C(u)C(v)\left[\sum_{m=0}^{(N-1)}\sum_{n=0}^{(N-1)} X(m,n)\cos\frac{(2m+1)u\pi}{2N}\cos\frac{(2n+1)v\pi}{2N}\right] \quad (3)$$

Inverse 2-D DCT

$$X(m,n) = \left[\sum_{u=0}^{(N-1)}\sum_{v=0}^{(N-1)} C(u)C(v)Y(u,v)\cos\frac{(2m+1)u\pi}{2N}\cos\frac{(2n+1)v\pi}{2N}\right] \quad (4)$$

where: $C(u) = \frac{1}{\sqrt{N}}$, $C(v) = \frac{1}{\sqrt{N}}$ for u, v = 0

$C(u) = \sqrt{\frac{2}{N}}$, $C(v) = \sqrt{\frac{2}{N}}$ for u, v = 1 through N-1;

N = 4, 8, or 16…

Equation (3) shows that the 2-D DCT can be computed by applying the 1-D DCT to each of the columns of the image matrix separately and then applying the 1-D DCT to each of the rows in the obtained results. This is the separability property of the 2-D DCT [12].

$$Y(u,v) = C(u)C(v)\left[\sum_{m=0}^{(N-1)}\cos\frac{(2m+1)u\pi}{2N}\sum_{n=0}^{(N-1)} X(m,n)\cos\frac{(2n+1)v\pi}{2N}\right] \quad (5)$$

where: $C(u) = \frac{1}{\sqrt{N}}$, $C(v) = \frac{1}{\sqrt{N}}$ for u, v = 0

$C(u) = \sqrt{\frac{2}{N}}$, $C(v) = \sqrt{\frac{2}{N}}$ for u, v = 1 through N-1;

N = 4, 8, or 16…

The equations (3) and (4) can be expressed in the matrix form as

$$Y = AXA^T \quad (6)$$

$$X = A^T YA \quad (7)$$

Here, the coefficient matrix A of size $N \times N$ is given by

$$A(i,j) = C(i)\cos\left(\frac{(2j+1)i\pi}{2N}\right) \quad (8)$$

A. The 8 point 1-Dimensional Integer DCT Algorithm

Using the above equation (8), 2-D DCT matrix A of a 8×8 size can be computed as follows.

$$A = \frac{1}{2}\begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \cos\frac{\pi}{16} & \cos\frac{3\pi}{16} & \cos\frac{5\pi}{16} & \cos\frac{7\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{11\pi}{16} & \cos\frac{13\pi}{16} & \cos\frac{15\pi}{16} \\ \cos\frac{2\pi}{16} & \cos\frac{6\pi}{16} & \cos\frac{10\pi}{16} & \cos\frac{14\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{22\pi}{16} & \cos\frac{26\pi}{16} & \cos\frac{30\pi}{16} \\ \cos\frac{3\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{15\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{27\pi}{16} & \cos\frac{33\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{45\pi}{16} \\ \cos\frac{4\pi}{16} & \cos\frac{12\pi}{16} & \cos\frac{20\pi}{16} & \cos\frac{28\pi}{16} & \cos\frac{36\pi}{16} & \cos\frac{44\pi}{16} & \cos\frac{52\pi}{16} & \cos\frac{60\pi}{16} \\ \cos\frac{5\pi}{16} & \cos\frac{15\pi}{16} & \cos\frac{25\pi}{16} & \cos\frac{35\pi}{16} & \cos\frac{45\pi}{16} & \cos\frac{55\pi}{16} & \cos\frac{65\pi}{16} & \cos\frac{75\pi}{16} \\ \cos\frac{6\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{30\pi}{16} & \cos\frac{42\pi}{16} & \cos\frac{54\pi}{16} & \cos\frac{66\pi}{16} & \cos\frac{78\pi}{16} & \cos\frac{90\pi}{16} \\ \cos\frac{7\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{35\pi}{16} & \cos\frac{49\pi}{16} & \cos\frac{63\pi}{16} & \cos\frac{77\pi}{16} & \cos\frac{91\pi}{16} & \cos\frac{105\pi}{16} \end{bmatrix}$$

Let

$$a = \frac{1}{2\sqrt{2}}, b = \frac{1}{2}\cos\frac{\pi}{16}, c = \frac{1}{2}\cos\frac{3\pi}{16}, d = \frac{1}{2}\cos\frac{5\pi}{16}$$

$$e = \frac{1}{2}\cos\frac{7\pi}{16}, f = \frac{1}{2}\cos\frac{2\pi}{16}, g = \frac{1}{2}\cos\frac{6\pi}{16}$$

A fair approximation of coefficients can be represented in following form

$$Y = (C_f X C_f^T) \otimes E \quad (9)$$

$$C_f = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}^{1/8}$$

$$E = \begin{bmatrix} a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \\ a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \\ af & af & f^2 & af & af & af & f^2 & af \\ a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \\ a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \\ a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \\ af & af & f^2 & af & af & af & f^2 & af \\ a^2 & a^2 & af & a^2 & a^2 & a^2 & af & a^2 \end{bmatrix}$$

Here C_f is known as the 1-D Integer DCT coefficient matrix.

According to property of orthogonal transform, inverse of an orthogonal matrix is equals to its transpose [7].

i.e. $C_f^{-1} = C_f^T$ (10)

From equation (7), (9) and (10), 2-D IDCT can be expressed as

$$X = (C_f^T Y C_f) \otimes E \quad (11)$$

B. The 8 point 1-D Integer DCT and Integer IDCT Architecture

The proposed 8x8 transform can be implemented with adders and subtracters only. The hardware architecture can be implemented by using the 1-D DCT algorithm. Three stages are used in this architecture. Here, 1-D integer DCT block is given with an eight bit integer number. The output of first stage consists of nine bits in which one bit is carry which is generated after addition operation. The outputs of the second and third stage of adders consist of 11, 12-bit integers respectively. The architectures of the 8 point forward 1-D integer DCT and inverse integer DCT transforms are shown in Fig. 3 and Fig. 4 respectively.

Here matrix C_f is used for 1-D integer transforms computation using fast butterfly operations (Here: X and Y represents inputs and the transformed outputs respectively) [4].

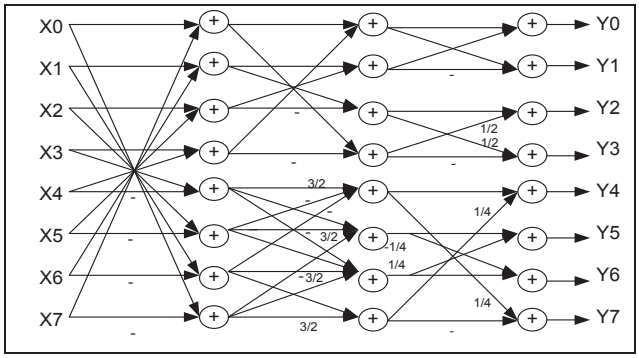

Fig. 3: 1-D Integer DCT architecture [4]

Table 1: Butterfly operations for 8-point 1-D Integer DCT (Here >> stands for right shift operation)

Stage 1	Stage 2	Stage 3
a0 =X0 +X7	b0=a0 + a3	Y0=b0 + b1
a1=X1 + X6	b1=a1 + a2	Y2=b2 + (b3>>1)
a2=X2 + X5	b2=a0 - a3	Y4=b0 - b1
a3=X3 + X4	b3=a1 - a2	Y6=(b2>>1) - b3
a4=X0 - X7	b4=a5 + a6 + ((a4>>1) + a4)	Y1=b4 + (b7>>2)
a5=X1 - X6	b5=a4 - a7 - ((a6>>1) + a6)	Y3=b5 + (b6>>2)
a6=X2 - X5	b6=a4 + a7 - ((a5>>1) + a5)	Y5=b6 - (b5>>2)
a7=X3 - X4	b7=a5 - a6 + ((a7>>1) + a7)	Y7=-b7 + (b4>>2)

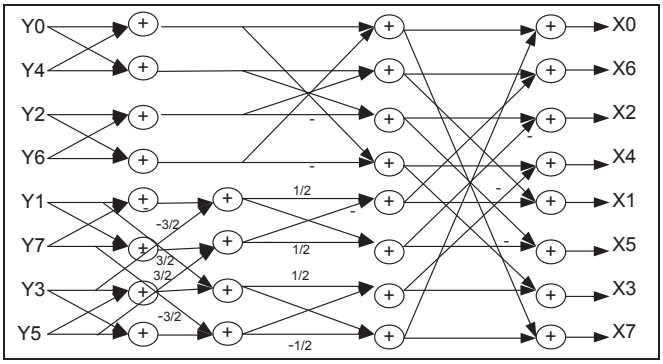

Fig. 4: 1-D Integer Inverse DCT architecture [1]

Table 2: Butterfly operations for 8-point 1-D Integer IDCT

Stage 1	Stage 2	Stage 3
a0=Y0+Y4	b0=a0 + a6	X0=b0 + b7
a4=Y0-Y4	b2=a4 + a2	X1=b2 + b5
a2=(Y2>>1)-Y6	b4=a4 - a2	X2=b4 + b3
a1=Y5-Y3-Y7-(Y7>>1)	b6=a0 - a6	X3=b6 + b1
a3=Y7+Y1 -Y3 –(Y3>>1)	b1=a1 +(a7>>2)	X4=b6 - b1
a5=Y7 –Y1 + Y5 +(Y5>>1)	b7=a7 -(a1>>2)	X5=b4 - b3
a7=Y3 +Y5 + Y1 +(Y1>>1)	b3=a3 + (a5>>2)	X6=b2 -b5
a1= Y5-Y3- Y7 – (Y7>>1)	b5=(a3>>2) - a5	X7=b0 - b7

III. SYSTEM LEVEL REPRESENTATION

H.264/MPEG-4 AVC video encoder [3] consists of the following processing modules:

- 8x8 2-D Integer DCT module
- Quantization and scaling modules

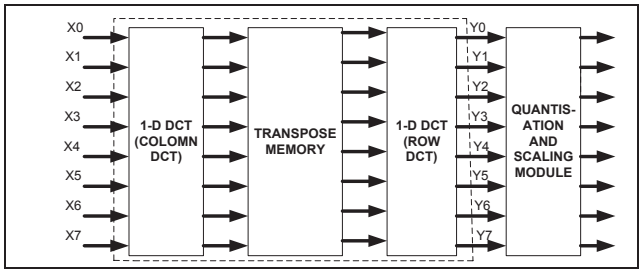

Fig. 5: Block diagram of H.264 encoder and use of 2D Integer DCT

Here, 2D DCT design uses pipelining architecture in which eight image pixel values are given as the inputs at the same time and processed simultaneously as shown in Fig. 5. Here the image in the given video divided into 8x8 blocks. The 2D DCT architecture is given with the each 8x8 image. Here the elements from each column of 8x8 image are applied in one clock period.

H.264 video decoder consists of the following processing modules:

- 8x8 2-D Integer IDCT module
- Quantization and scaling modules

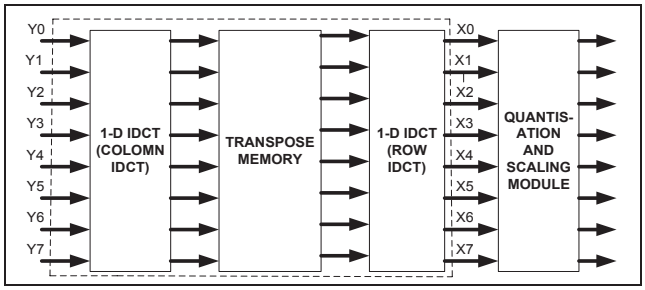

Fig. 6: Block diagram of H.264 decoder and use of 2D Integer IDCT

Figure 6, shows the 2D Integer IDCT design using the pipelining architecture. Here, the eight pixels from one column of 8x8 transformed blocks are applied to the design for every clock period. The entire 8x8 image block can be read in eight clock cycles.

Butterfly diagram is used to represent the DCT operation. Eight inputs each with 8-bit width are given simultaneously to the first 1-D Integer DCT module in each clock cycle, to evaluate column transform. Then processed in parallel and generate the eight outputs (each with twelve bit width) simultaneously after one clock cycle delay. These outputs of first stage are given as the inputs to the transpose memory. Here, Fig. 7 shows block diagram of the transpose memory. The transpose memory performs the transpose operation of column DCT output. The transpose memory size is 96 bytes.

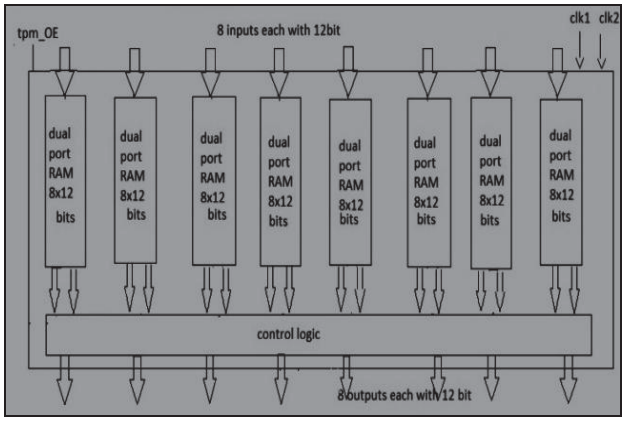

Fig. 7: Transpose memory block diagram

IV. IMPLEMENTATION AND RESULTS

The 2-D Integer DCT and IDCT modules are programmed in Verilog HDL coding using the Xilinx ISE 10.1 software. The top module of 2-D Integer DCT and IDCT are shown in Fig. 8 and Fig. 9 respectively. These modules are synthesized on the FPGA device xc2vp30, which belongs to the family of vertex 2-pro with speed grade -7. The corresponding simulation results are shown in the Fig. 10 and Fig. 11.

Table 3: Clock cycle summary

Block	Task	No. of clock cycles required
Column DCT/IDCT	1-D DCT/IDCT operation on each column.	1
Transpose memory	Transpose operation for all columns	40
Row DCT/IDCT	1-D DCT/IDCT operation .	1

The Verilog and MATLAB coding has been used for the implementation and verification of the work. Corresponding results obtained as shown below.

In (Input Data) =

$$\begin{bmatrix} 0 & 8 & 16 & 24 & 32 & 40 & 48 & 56 \\ 1 & 9 & 17 & 25 & 33 & 41 & 49 & 57 \\ 2 & 10 & 18 & 26 & 34 & 42 & 50 & 58 \\ 3 & 11 & 19 & 27 & 35 & 43 & 51 & 59 \\ 4 & 12 & 20 & 28 & 36 & 44 & 52 & 60 \\ 5 & 13 & 21 & 29 & 37 & 45 & 53 & 61 \\ 6 & 14 & 22 & 30 & 38 & 46 & 54 & 62 \\ 7 & 15 & 23 & 31 & 39 & 47 & 55 & 63 \end{bmatrix}$$

***MATLAB results*:**

dct2_MATLAB (Simulation Results in MATLAB) =

$$\begin{bmatrix} 2016 & -1240 & 0 & -104 & 0 & -8 & 0 & -72 \\ -155 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Idct2_MATLAB (Simulation Results in MATLAB) =

$$\begin{bmatrix} -259.9 & 318.1 & 896.1 & 1474.1 & 2052.1 & 2630.1 & 3208.1 & 3786.1 \\ -187.6 & 390.4 & 968.4 & 1546.4 & 2124.4 & 2702.4 & 3280.4 & 3858.4 \\ -115.4 & 462.6 & 1040.6 & 1618.6 & 2196.6 & 2774.6 & 3352.6 & 3930.6 \\ -43.1 & 534.9 & 1112.9 & 1690.9 & 2268.9 & 2846.9 & 3424.9 & 4002.9 \\ 29.1 & 607.1 & 1185.1 & 1763.1 & 2341.1 & 2919.1 & 3497.1 & 4075.1 \\ 1014 & 679.4 & 1257.4 & 1835.4 & 2413.4 & 2991.4 & 3569.4 & 4147.4 \\ 173.6 & 751.6 & 1329.6 & 1907.6 & 2485.6 & 3063.6 & 3641.6 & 4219.6 \\ 245.9 & 823.9 & 1401.9 & 1979.9 & 2557.9 & 3135.9 & 3713.9 & 4291.9 \end{bmatrix}$$

Verilog Simulation results:

dct2_VERILOG (Simulation Results in VERILOG)=

$$\begin{bmatrix} 2016 & -1240 & 0 & -104 & 0 & -8 & 0 & -72 \\ -144 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Idct2_VERILOG (Simulation Results in VERILOG)

$$\begin{bmatrix} -259 & 319 & 897 & 1475 & 2053 & 2631 & 3209 & 3787 \\ -188 & 390 & 968 & 1546 & 2124 & 2702 & 3280 & 3858 \\ -116 & 462 & 1040 & 1618 & 2196 & 2774 & 3352 & 3930 \\ -43 & 535 & 1113 & 1691 & 2269 & 2847 & 3425 & 4003 \\ 29 & 607 & 1185 & 1763 & 2341 & 2919 & 3497 & 4075 \\ 1014 & 678 & 1258 & 1836 & 2414 & 2992 & 3570 & 4148 \\ 174 & 752 & 1330 & 1908 & 2486 & 3064 & 3642 & 4220 \\ 245 & 823 & 1401 & 1979 & 2557 & 3135 & 3713 & 4291 \end{bmatrix}$$

Implemented schematics for forward and Inverse Integer DCT are also shown with their simulation in the Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

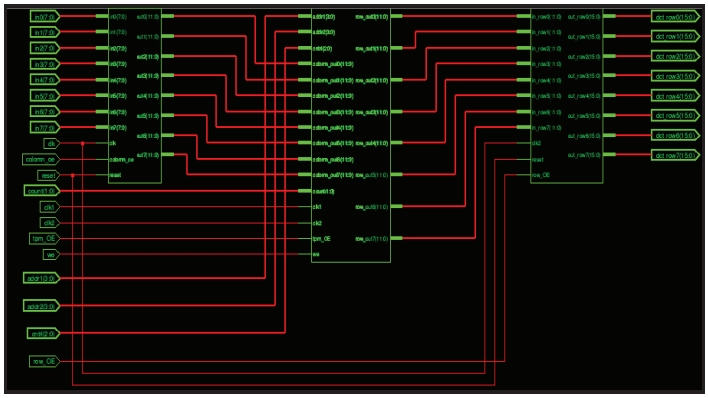

Fig. 8: RTL schematic 2-D Integer DCT

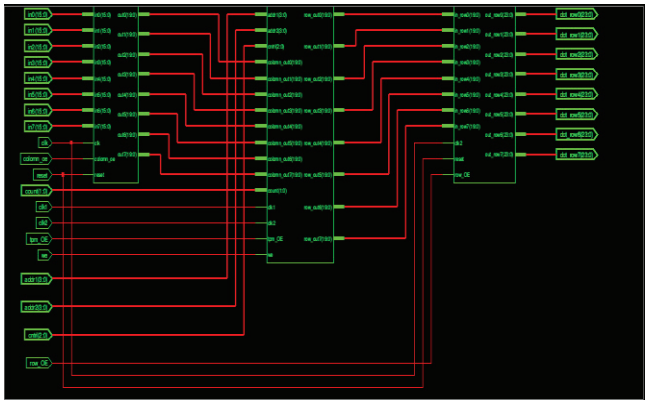

Fig. 9: RTL schematic 2-D Integer IDCT

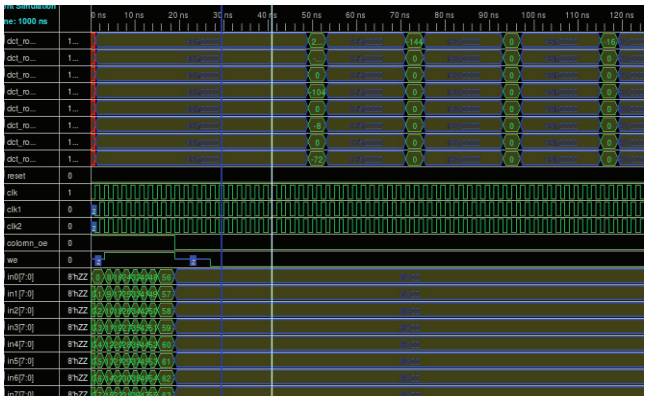

Fig. 10 Simulation results of 2-D Integer DCT chip

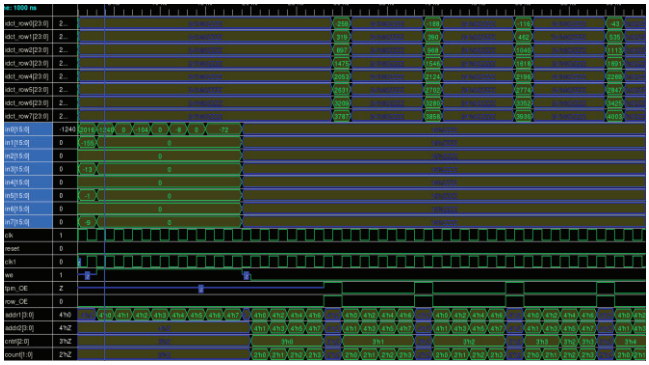

Fig. 11 Simulation results of 2-D Integer IDCT chip

Device utilization summary of 2-D Integer DCT and IDCT module for Vertex 2-Pro (xc2vp30) has been shown in Table 4 and Table 5 respectively.

Table 4: Device utilization summary of 2-D Integer DCT module for Vertex 2-Pro (xc2vp30)

Logic utilization	**Used**	**Available**	**Utilization**
No. of Slices	1988	13696	14%
No. of Slice Flip-Flops	423	27392	1%
No. of 4 input LUTs	3257	27392	11%
No. of bonded IOBs	213	416	51%

Table 5: Device utilization summary of 2-D Integer IDCT module for Vertex 2-Pro (xc2vp30)

Logic utilization	**Used**	**Available**	**Utilization**
No. of Slices	3856	13696	28%
No. of Slice Flip-Flops	769	27392	2%
No. of 4 input LUTs	6701	27392	24%
No. of bonded IOBs	209	416	37%

A comparison has been made with the different implementations of 2D Integer DCT in Table 6. It has been seen that our design takes less number of LUTs and IOBs which results in the reduction of resources utilization. Also the design is implemented with at the high clock frequency which implies the high speed of operation.

Table 6: Comparison with other implementations of Integer DCT**.**

Chip name	**Total no. of LUTs**	**Total no.of IOBs**	**Clock frequency**
Wael, high performance 8x8 2-D DCT chip[11]	29018	1800	68.5Mhz
2-D DCT Pipelined structure for H.264 (our chip)	3,257	213	520.996Mhz
2-D DCT Transform (without RAM) (our chip)	22,586	1540	37.551Mhz

V. CONCLUSION

The pipelined architectures of 2-D Integer DCT/IDCT are designed to minimize the resources requirement and increase the throughput of the design. After the pipelining levels of 1-D DCT/IDCT, the speed of operation is increased more than the twice. It is also observed that the 8x8 2-D Integer DCT gives better compression performance compared to the 4x4 Integer 2-D DCT. These 2-D Integer DCT/IDCT modules can be most effectively used for high profile encoder/decoder of H.264 standard.

REFERENCES

[1] Frank Bossen, "*ABT cleanup and complexity reduction,"* Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 5th Meeting: Geneva, Switzerland, October, 2002.

[2] HDL Chip Design, "A Practical guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL/Verilog", by Douglas Smith.

[3] T. Wiegand, G. J. Sullivan, "*Overview of the H.264/AVC video coding standard*," IEEE Transactions on circuits and systems for video technology, vol.13 No.7 July.2003

[4] S. Gordon, D. Marpe, and T. Wiegand, "*Simplified Use of 8x8 Transform – Proposal*," JVT of ISO/IEC MPEG and ITU-T VCEG.

[5] N. Ahmed et al., "*Discrete cosine transform*," IEEE Trans. Comput., pp. 90-93, Jan. 1974.

[6] G. Sullivan, P. Topiwala, & A. Luthra, "*The H.264 Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions*," SPIE Conference on Application of Digital Image Processing XXVII, USA, 2004.
[7] Kim, Jeong & Lee, "Efficient DCT/IDCT and quantization implementation of multimedia ASIP for H.264".
[8] Henrique S. Malvar, Antti Hallapuro, Marta Karczewicz, and Louis Kerofsky, "*Low-Complexity transform and quantization in H.264/AVC"*, IEEE transactions on circuits and systems for video technology, Vol.13, No.7, July 2003.
[9] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "*Low-Complexity Transform and Quantization in H.264/AVC*," IEEE Transactions on Circuits and Systems For Video Technology, Vol. 13, No. 7, pp. 598-603, July 2003.
[10] Ihab Amer, Wael Badawy, and Graham Jullien "A high-performance hardware implementation of the H.264 simplified 8X8 Transformation, Quantisation". Colorado, USA, August 2005 IEEE.
[11] ITU-T, "*Video coding for low bit rate communication*," ITU-T Rec. H.263; v1: Nov. 1995, v2: Jan. 1998, v3: Nov. 2000.
[12] Agostini, Silva, "*Pipelined fast 2-D DCT architecture for JPEG image compression*," 14th Symposium on Integrated Circuits and Systems Design, 2001, pp: 226-31.
[13] I. E. G. Richardson, "H.264/MPEG-4 Part 10: Transform & Quantization," A white paper. [Online] Available: http://www.vcodex.com, 2002.
[14] Joint Video Team of ITU-T, and ISO/IEC JTC 1: Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 or ISO/IEC 14496-10 AVC). JVT Document, JVT-G050r1 (2003)

Author's Profile

Pankaj Kumar Srivastava received M.Tech Degree with specialization in Microwave Engineering from Govt. College of Engineering, Pune, in June 2006. He did Ph.D in the field of Wireless Communication from SGGS Nanded, Maharashtra in 2014. His Research interests are Wireless Communication, Image Processing, Automation & Robotics, VLSI & Embedded System. He has worked as a Lecturer in Cummins College of Engineering for Women, Pune, Assistance Professor & Head of Electronics & Communication Department in Siddhant College of Engineering, Pune and TSSM'S BSCOER, Pune. He has worked as an Associate Professor and Research Coordinator in PVPIT, Pune. Presently he is associated with ISB&M School of Technology, Pune as an Professor & Principal, Affiliated to University of Pune,(India). Life time member of ISTE.

Anil Kumar Jakkani received M.Tech. Degree with specialization in Communication Systems from National Institute of Technology, Surat in June 2011. He is a coordinator of Innovation & Research Centre, ISB&M SOT. Presently, he is working as an Assistant Professor in ISB&M School of Technology.