Implementation of Discrete Cosine Transform using VLSI

K. M. Khatri[1], S. S. Agrawal[2]
[1]PG student, [2]Assistant Professor
Dept. of Electronics and Telecommunication,
Government College of Engineering, Aurangabad (M.S.) India

***Abstract*—Discrete Cosine Transform (DCT) is one of the widely used transform for image compression. It de-correlates image data. Independent encoding of the transformed coefficients can be done in compression without losing image quality. In this paper we present implementation of Discrete Cosine Transform (DCT) on VLSI platform. Using the separability property of 2D DCT, the 2D DCT architecture is divided into two 1D DCT blocks with the transpose buffer. The architecture used for calculating 1D DCT is based on Arai's algorithm for DCT calculation. The design is implemented on Spartan 3E board from Xilinx for an 8x8 input block.**

Keywords: Discrete Cosine Transform (DCT), VLSI, image processing, compression

I. INTRODUCTION

Transform coding is integral part of image/video processing applications. Transform coding is based on the concept that pixels in an image have some amount of correlation with the neighbouring pixels. In the similar way, a high correlation exists in the adjacent frames of a video. By using this correlation information the value of the pixel can be predicated from its neighbouring pixels. So transformation maps correlated data of spatial image into uncorrelated transformed coefficients. Transformation is the lossless process and inverse transformation of transformed image gives original image.

Discrete Cosine Transform (DCT) is widely used for transform coding in most image processing applications. The large amount of data in the digital image is a big problem for transmission & storage of images. Different compression schemes have been developed to transmit/store the image & video with fewer amounts of data. Because of its symmetry, energy compaction and simplicity DCT algorithm is more effective for image compression. A DCT based image processing system gives lossy compression of an image because of the use of quantizer. After transformation most of the image information is packed in a few coefficients. So quantizer discards the coefficients with small amplitudes. This is done by dividing the DCT coefficients by the quantization matrix.

The direct computation of 2D DCT using formula requires huge arithmetic operations. This leads to large hardware requirement & time consumption. Thus considering the VLSI implementation, 2D DCT can be found by two successive 1D DCT's according to its separability property. The two dimensional DCT of an image is obtained by using 1D DCT processor & a transpose buffer.

The 2D DCT of N X N matrix is obtained by row-column decomposition technique. An image is decomposed into small blocks of 8x8 pixels for computational efficiency. An image is decomposed into small blocks of 8x8 pixels for computational efficiency. First 1D DCT of each column of decomposed data block is calculated. The resulting matrix is transposed and stored in memory. Then another 1D DCT of each row of the transposed matrix is calculated to get the 2D DCT of decomposed block. This paper describes computation of 2D DCT by two successive 1D DCT computation.

II. DISCRETE COSINE TRANSFORM FORMULAE

The DCT is a transformation that contains a sequence of discrete numbers in which each number is a summation of cosine functions with varying amplitude and frequency.

A. 1D DCT

The N-point DCT and inverse DCT (IDCT) of N-point sequence x(n) is defined as [1]:

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

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

$$\text{where} \quad e(k) = \begin{cases} \frac{1}{\sqrt{2}} & if\ k = 0 \\ 1 & otherwise \end{cases}$$

1D DCT equation for N=8 can be written in matrix form as follows:

$$\begin{bmatrix} X(0)\\X(1)\\X(2)\\X(3)\\X(4)\\X(5)\\X(6)\\X(7) \end{bmatrix} = \begin{bmatrix} c4 & c4 & c4 & c4 & c4 & c4 & c4 & c4 \\ c1 & c3 & c5 & c7 & -c7 & -c5 & -c3 & -c1 \\ c2 & c6 & -c6 & -c2 & -c2 & -c6 & c6 & c2 \\ c3 & -c7 & -c1 & -c5 & c5 & c1 & c7 & -c3 \\ c4 & -c4 & -c4 & c4 & c4 & -c4 & -c4 & c4 \\ c5 & -c1 & c7 & c3 & -c3 & -c7 & c1 & -c5 \\ c6 & -c2 & c2 & -c6 & -c6 & c2 & -c2 & c6 \\ c7 & -c5 & c3 & -c1 & c1 & -c3 & c5 & -c7 \end{bmatrix} \begin{bmatrix} x(0)\\x(1)\\x(2)\\x(3)\\x(4)\\x(5)\\x(6)\\x(7) \end{bmatrix}.$$

Where $ci = cos\frac{i\pi}{16}$

B. 2D DCT

As digital images are two dimensional, 2D DCT is applied to them. Mathematical representation of N x N point 2D DCT can be written as

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

$$\text{where } e(k) = e(l) = \begin{cases}\frac{1}{\sqrt{2}} \; if\ (k,l) = 0\\ 1 \quad otherwise\end{cases}$$

In equation no [], *x(m,n)* is 2D input data and *X(k,l)* is transformed DCT coefficients.

[]can be written as follows:

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

In above equation $\sum_{n=0}^{N-1}x(m,n)cos\left[\frac{(2n+1)k\pi}{2N}\right]$ represent 1D DCT of columns of *x(m,n).* Therefore it is concluded that calculating 1D DCT of columns followed by 1D DCT of rows results in 2D DCT coefficients of 2D input data.

III. ALGORITHM FOR DCT

A.One Dimensional DCT calculation

A high degree of computational complexity exists for calculating 2D DCT. Many authors have developed simple methods for this calculation like [2], [4], [5], [6], [10] & others. Especially for image processing applications many algorithms have been proposed for 2D DCT calculation with less complexity.

This paper uses the architecture proposed in [3] for calculating DCT. It is Arai's scaled DCT algorithm. This algorithm has many simplifications which help to reach higher performance. Because of these simplifications it is possible to calculate eight point 1D DCT with just 29 additions and 5 Multiplications. Thus only 464 additions and 80 multiplications are required for calculating 2D DCT of an 8x8 block. As it is scaled DCT algorithm the scaled outputs can be obtained by 8 scaling multipliers at output for eight point 1D DCT. Fig-1 shows Arai's DCT architecture.

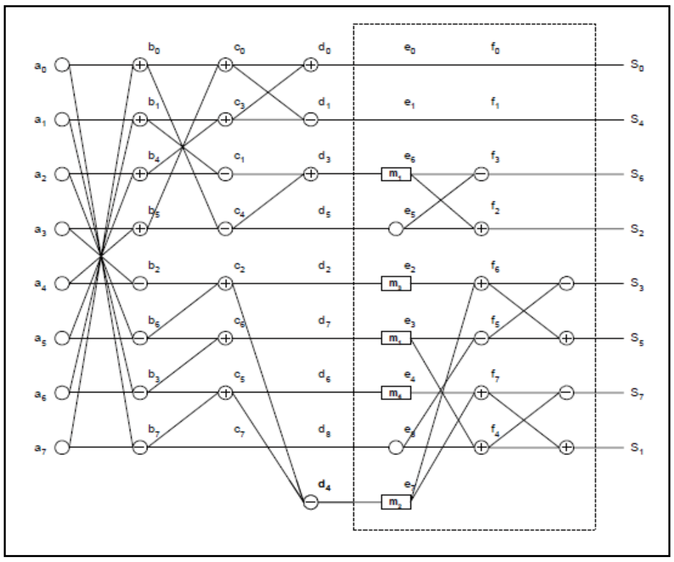


Fig 1: Arai's Eight-point Fast DCT Flow graph

The {ai} are input elements and {si} are scaled DCT coefficients. The multipliers used in structure are of four different values namely m1, m2, m3, m4. These fixed multipliers are given by

$$m_1 = cos\frac{4\pi}{16}$$

$$m_2 = cos\frac{6\pi}{16}$$

$$m_3 = cos\frac{2\pi}{16} - cos\frac{6\pi}{16}$$

$$m_4 = cos\frac{2\pi}{16} + cos\frac{6\pi}{16}$$

B.Two Dimensional DCT Calculation

To calculate 2D DCT there are many proposed architectures as [7], [8], [9] & others. The 2D DCT architecture presented in fig-2 is used in this paper. This architecture is divided into three parts, two 1D DCT architecture and one transpose buffer. The two 1D DCT architectures are similar with different bit widths at internal stages. The 1D DCT architecture discussed in above section is used here in each 1D DCT block.

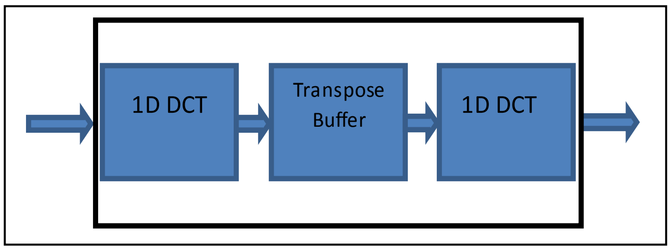


Fig-2: 2D DCT architecture

The input to the 2D DCT architecture in our design is matrixes of 8x8 elements with 8 bit width of each element.

The first 1D DCT receives the row wise data from input matrix and stores results in transpose buffer. Transpose buffer gives column wise inputs to the second 1d DCT architecture. The result of this is nothing but 2D DCT coefficients.

IV. RESULTS

A. Synthesis and Implementation

The RTL schematic of the implementation of DCT on 8x8 image is shown in fig. The schematic consists of 4 blocks which are included in the top module. The first block named mem_in is initialized with the 64 bytes of 8x8 input block. The dct_module block receives 8-8 bytes from mem_in and calculates the 1D DCT. These results are stored in ram naming mem_out and also in the internal array of dct_module. In dct_module the 1D DCT array is transposed and 2D DCT is calculated. The results of the 2D DCT are stored in mem_out2. The VHDL is used for modelling the design. The ISE project Navigator from Xilinx was used for synthesis.

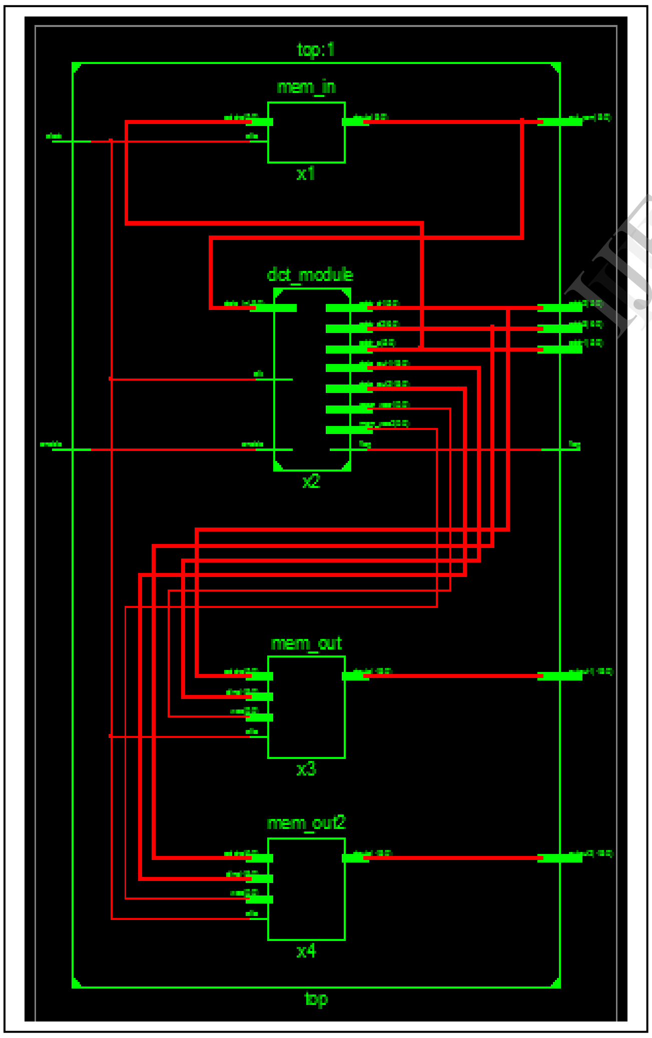
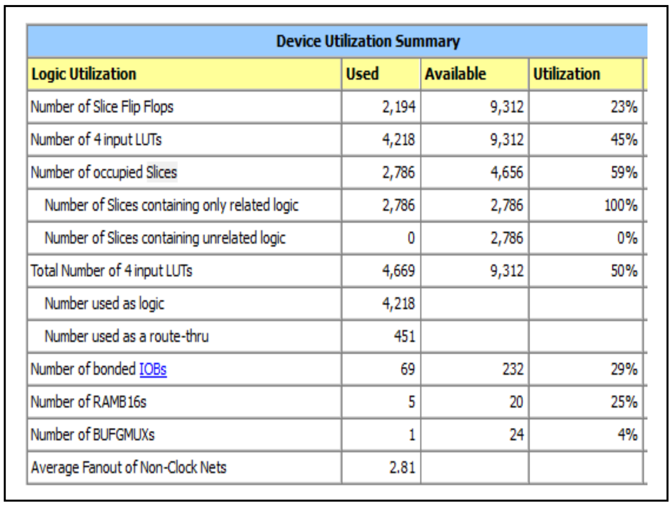


Fig 3: RTL schematic of top module for DCT of an image

The different resources in the FPGA are LUTs, Flip Flops, bonded IOBs, etc. The utilization of these resources for calculating DCT of 8x8 image is shown in Fig 4.



Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	2,194	9,312	23%
Number of 4 input LUTs	4,218	9,312	45%
Number of occupied Slices	2,786	4,656	59%
Number of Slices containing only related logic	2,786	2,786	100%
Number of Slices containing unrelated logic	0	2,786	0%
Total Number of 4 input LUTs	4,669	9,312	50%
Number used as logic	4,218		
Number used as a route-thru	451		
Number of bonded IOBs	69	232	29%
Number of RAMB16s	5	20	25%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	2.81		

Fig 4: Device Utilization Summary For 8x8 image size

B. Simulation Results

The simulation results of the design are shown in Fig 5. The results of DCT are verified by results obtained using MATLAB software.

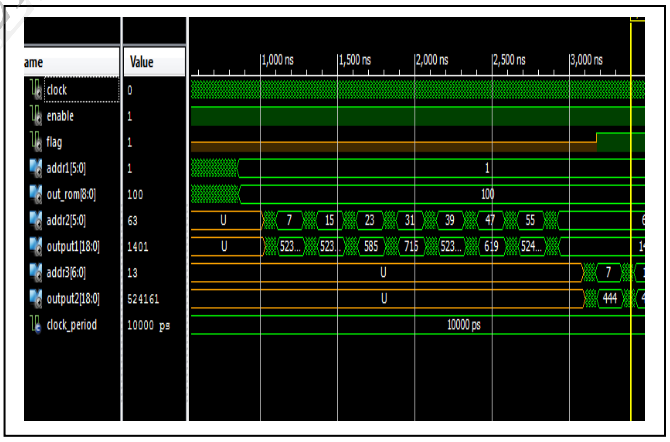


Fig 5: Simulation Results

C. FPGA Implemetation

The successful implementation of the design was done on device xc3s500e-4fg320 from the Spartan 3E family. The results from the board are observed using chip scope pro tool of Xilinx.

V. CONCLUSION

In this paper hardware implementation of DCT of 8x8 size input image is shown. Row column decomposition technique resulted in less hardware complexity. 1D DCT structure used here gives minimum transformation loss and ease of hardware implementation. The read and write access of transformation buffer have been effectively controlled during 2D DCT implementation.

This work can be used as one of the black box for image compression process. By using other functional blocks along with this DCT block a high performance image compression technique can be achieved.

ACKNOWLEDGMENTS

We thank the Principle Government Engineering College Aurangabad for providing the research environment required for work experiment.

REFERENCES

[1] Sutasinee Sanesaowarod, Chugiat Garagate and Dusit Thanapatay, “Novel Design of Fast Pipelined 2D QDCT on FPGA”, IEEE 2012
[2] A. Pradini, “VLSI Design of a High-Throughput Discrete Cosine Transform for Image Compression Systems”, 2011 International Conference on Electrical Engineering and Informatics 17-19 July 2011, Bandung, Indonesia, IEEE 2011
[3] Y. Arai, T. Agui, and M. Nakajima, “A fast DCT-SQ scheme for images,” in *Trans. IEICE*, vol. E-71, no. 11, p. 1095, Nov. 1988.
[4] Luciano Volcan Agostini,” Pipelined Fast 2-D DCT Architecture for JPEG Image Compression”
[5] Subramanian P, A Sagar Chaitanya Reddy,” VLSI Implementation of Fully Pipelined Multiplierless 2D DCT/IDCT Architecture for JPEG, ICSP 2010 Proceedings, IEEE 2010
[6] Ming-Ting Sun, Ting-Chung Chen, “Vlsi Implementation Of A 16 X 16 Discrete Cosine Transform”, IEEE Transactions On Circuits And Systems, Vol. 36, No. 4, April 1989
[7] Jie Liang and Trac D. Tran, “Approximating the DCT with the Lifting Scheme: Systematic Design and Applications”, IEEE 2000
[8] Yuh-Ming Huang, Ja-Ling Wu’r, and Chiou-Ting Hsu, “A Refined Fast 2-D Discrete Cosine Transform Algorithm With Regular Butterfly Structure”, IEEE Transactions on Consumer Electronics, Vol. 44, No. 2, MAY 1998
[9] Vassil Dimitrov And Khan Wahid, “Multiplierless Dct Algorithm For Image Compression Applications”, International Journal "Information Theories & Applications" Vol.11
[10] Trac D. Tran, “The BinDCT: Fast Multiplierless Approximation of the DCT”, Ieee Signal Processing Letters, Vol. 7, No. 6, June 2000