

An Orthogonal Approximation of DCT and Reconfigurable Architecture for Image Compression

Reni Tom¹, Sunitha S Pillai²P.G. Student, Dept. of ECE, St. Joseph's College of Engineering & Technology, Palai, Kerala, India¹Assistant Professor, Dept. of ECE, St. Joseph's College of Engineering & Technology, Palai, Kerala, India²

ABSTRACT: The Discrete Cosine Transform (DCT) is popularly used in image and video compression applications. Existing approximation algorithms of DCT are available for only small transform length. In this approximation of DCT provides comparable or better image compression performance than the existing approximation methods. In this paper, a fully scalable reconfigurable parallel architecture for the computation of approximate DCT based on the recursive sparse matrix decomposition and make use of the symmetries of DCT basis vectors. The proposed design is used for the computation of a 32-point DCT or for parallel computation of two 16-point DCTs or four 8-point DCTs. The proposed architecture having many advantages in terms of lower arithmetic complexity and hardware complexity. The DCT used in the image compression can be replaced with the modified approximation DCT. This offer a similar compression like the existing ones with lower complexity. This entire system is designed by Verilog HDL and simulated using Xilinx ISE software simulation tools and synthesised on the Xilinx Spartan 6 FPGA kit.

KEYWORDS: Approximate DCT, Reconfigurable Architecture, Image Compression, Low complexity transforms.

I. INTRODUCTION

From last few decades, demand for communication of multimedia data through the telecommunications network and accessing the multimedia data through Internet is growing explosively. Image compression is very important for efficient transmission and storage of images. There are many image compression techniques available, but still there is need to develop faster and more strong and healthy techniques to compress images. Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. There are three different types of redundancy relevant to images: spatial redundancy, psycho visual redundancy and spectral redundancy. By using data compression techniques, it is possible to remove some amount of redundant information. This will save some amount of file size and allows more images to be stored in a certain amount of disk or memory space. There are various transformation techniques used for data compression.

Discrete Cosine Transform (DCT) are the most commonly used transformation. The use of cosine rather than sine functions is critical for compression, since it requires fewer cosine functions to approximate a typical signal. DCT has high energy compaction property and requires less computational resources. The major advantage of approximating the DCT is to get rid of multipliers which contribute for the computational complexity. Rounding of the fractional value to the nearest value i.e., either 0 or 1 is nothing but approximating. This approximation is done because the computational time is more for fractional calculations. So if the fraction value is rounded off to the nearest value the computational time taken will be less. Most of the existing algorithms for approximation of the DCT target only the DCT of small transform length. Higher length, such as 16-point and 32-point is not possible and some of them are non-orthogonal [2]. If the transform is orthogonal, we can always find its inverse and the kernel matrix of the inverse transform is obtained by just transposing the kernel matrix of the forward transform.

In several applications like video compression or image compression, DCT of different lengths such as $N=8, 16, 32$ are required to be used [3]. Therefore, a given DCT architecture should be potentially reused for the DCT of different lengths instead of using separate structures for different lengths. Here such a reconfigurable DCT structures which could be reused for the computation of DCT of different lengths is proposed. It performs the calculation of a 32-

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016

point DCT or two 16-point DCTs in parallel or four 8-point DCTs in parallel. The reconfigurable architecture proposed is used for the transformation in image compression application. This have low computational complexity than other DCTs. Image is reconstructed using the inverse operations.

Paper is organized as follows. Section II describes a brief study about normal DCT computation for length 8-point and 16-point is carried out. Section III presents the proposed scalable and reconfigurable architecture for DCT computation. Section IV presents the DCT application in Image Compression. Section V presents experimental results showing results of DCT and images tested. Finally, Section VI presents conclusion.

II. RELATED WORK

Different approximation algorithm along with its butterfly diagram and the transformation matrix are shown below.

1. BAS 2010 Approximation

A multiplication-free transform of order $N > 4$ suitable for image compression by an appropriate extension of the well-known fourth-order integer DCT [4] presented in Fig. 1. Its transform matrix of order 8 is given below.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 & 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -1 & 1 & -2 & 2 & -1 \end{bmatrix}$$

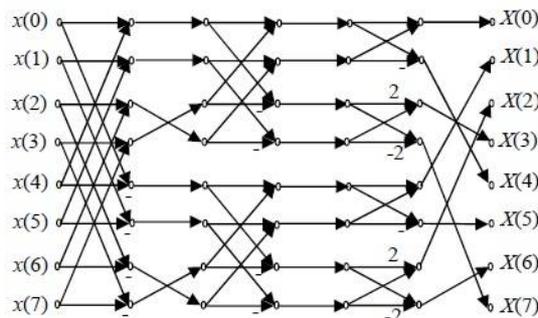


Fig 1 Digital architecture of BAS 2010

2. CB 2011 Approximation

By rounding-off the elements of the exact DCT matrix, CB-2011 Approximation [5] is obtained. The transformation matrix is given as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

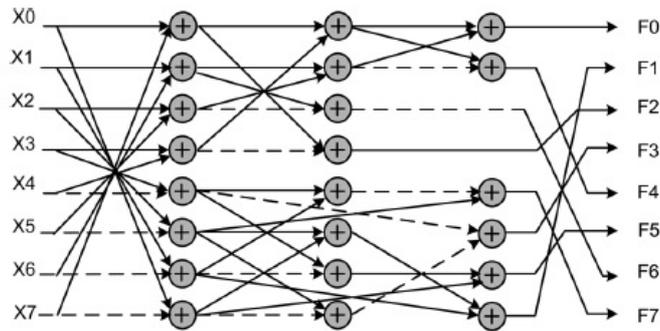


Fig 2 Digital architecture of CB 2011

3. Uma Potluri 2014 Approximation

In [6], approximate DCT with fewer computations in Fig 3 describe the cost of a transformation matrix as the quantity of number-crunching operations required for its calculation. The multiplicative complexity is null because of the matrix elements lies in $\{0, \pm 1, \pm 2\}$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

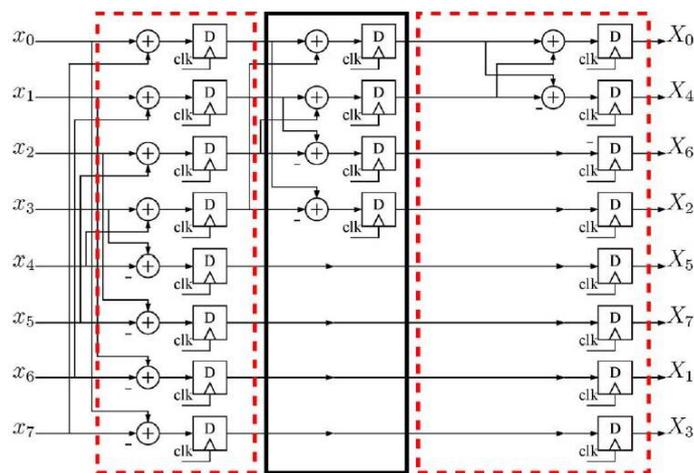


Fig 3 Digital architecture of Uma Potluri 2014

Similarly in 16-point normal DCT input adder unit, 16-point computation section and output permutation unit is present. The block diagram for normal 16-point DCT computation is shown in Fig 4.

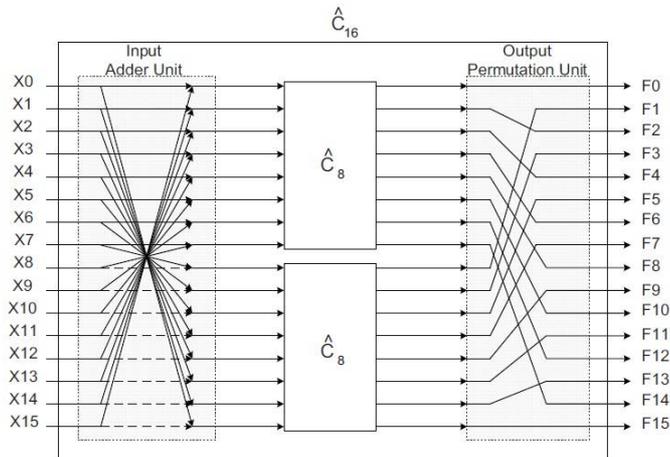


Fig. 4 Block diagram of 16-point normal DCT

III. PROPOSED WORK

So in previous works, if 8-point or 16-point DCT is to be computed then we require a separate architecture for both 8- point and 16-point DCT computation. So to overcome this problem here a generalized architecture is designed in which all the 8-point, 16-point and 32-point DCT computation can be done using the same architecture by the help of select lines. A given DCT architecture should be potentially reused for the DCT of different lengths instead of using separate structures for different lengths.

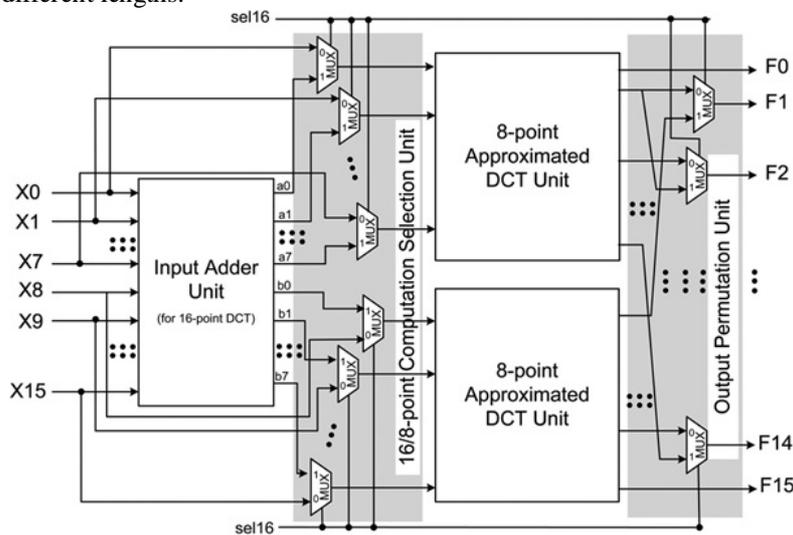


Fig. 5 Reconfigurable architecture for approximate DCT of lengths N=8 and 16.

The reconfigurable architecture for the implementation of approximated 16-point DCT is shown in Fig 5. It consists of computing units like two 8- point approximated DCT units, 16-point input adder unit and permutation unit. The input to the first 8- point DCT approximation unit is fed through 8 MUXes that select either [a(0), a(1), ..., a(7)] or [X(0), X(1), ..., X(7)] , depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. Similarly, the input to the second 8-point DCT unit is fed through 8 MUXes that select either [b(0), b(1), ..., b(7)] or [X(8), X(9), ..., X(15)], depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. On the other hand, the output permutation unit uses 14 MUXes to select and re-order the output depending on the size of the selected DCT. Sel16 is used as control input of the MUXes to select inputs and to perform permutation according to the size of the DCT to be computed. Specifically, sel16=1 enables the computation of 16-point DCT and sel16=0

enables the computation of a pair of 8-point DCTs in parallel. Consequently, the architecture of Fig. 5 allows the calculation of a 16-point DCT or two 8-point DCTs in parallel.

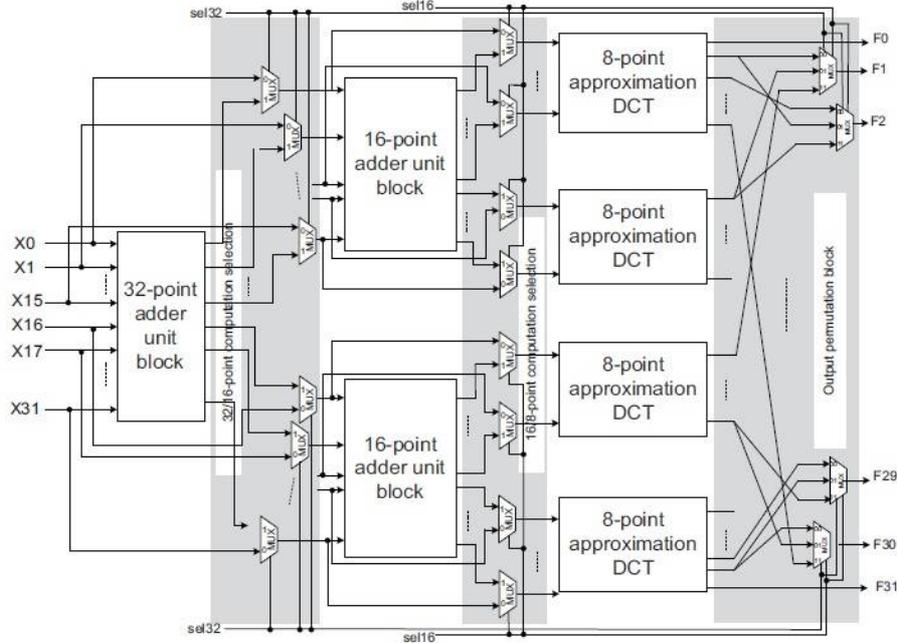


Fig. 6 Reconfigurable architecture for approximate DCT of lengths N=8, 16 and 32.

A reconfigurable design for the computation of 32, 16, and 8-point DCTs is presented in Fig. 6. It performs the calculation of a 32-point DCT or two 16-point DCTs in parallel or four 8-point DCTs in parallel. The architecture is composed of 32-point input adder unit, two 16-point input adder units, four 8-point DCT units and permutation unit. The reconfigurability is achieved by three control blocks composed of 64 2:1 MUXes along with 30 3:1 MUXes. The first control block decides whether the DCT size is of 32 or lower. If $sel_{32}=1$, the selection of input data is done for the 32-point DCT, otherwise, for the DCTs of lower lengths. The second control block decides whether the DCT size is higher than 8. If $sel_{16}=1$ the length of the DCT to be computed is higher than 8 (DCT length of 16 or 32), otherwise, the length is 8. The third control block is used for the output permutation unit which reorders the output depending on the size of the selected DCT. sel_{32} and sel_{16} are used as control signals to the 3:1 MUXes. Specifically, for $\{sel_{32}, sel_{16}\}$ equal to $\{00\}$, $\{01\}$ or $\{11\}$ the 32 outputs of DCT correspond to four 8-point parallel DCTs, two parallel 16-point DCTs, or 32-point DCT respectively.

IV. APPLICATION TO IMAGE COMPRESSION

The block diagram for image compression is as shown in fig 7. The colour image is converted to gray scale image with pixel values ranging from 0 to 255. The value 0 and 255 represents black and white of an image. The image is partitioned into 8×8 blocks[7]. The 2D DCT is applied to the image pixels. The 2D DCT is obtained by the approximate 1D DCT and transposition buffer as shown in fig 8. The transform is applied along the row then column of the image. The transposition buffer is shown in fig 9. The compression is achieved in quantisation in which the transform matrix is divided by the quantisation matrix. The zigzag ordering is done in which higher frequency data are arranged first and then lower frequency data. The run length encoding is done to compress the consecutive data appear in the string. Finally Huffman coding is done in the compression to obtain the code corresponding to the probability of the symbols [8].

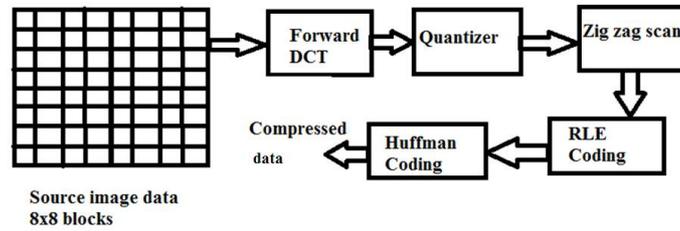


Fig 7 block diagram for image compression.

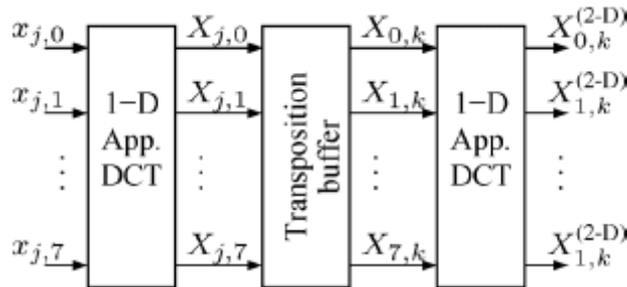


Fig 8 2D DCT

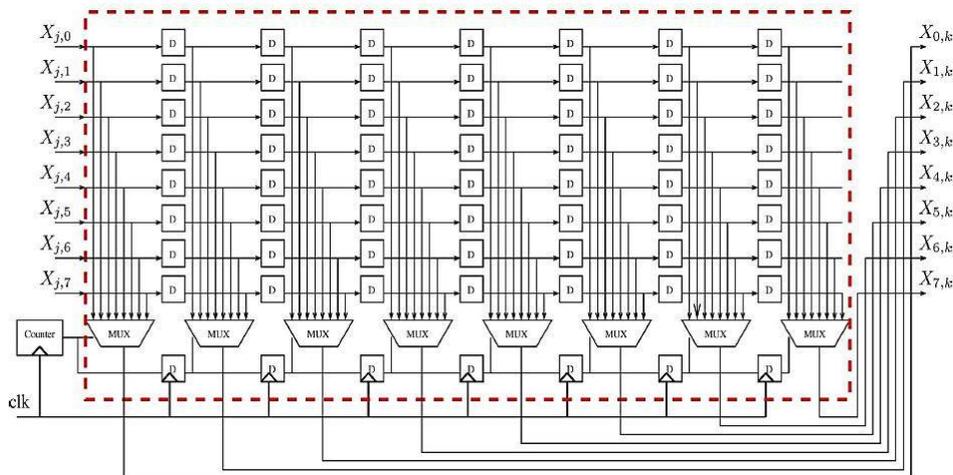


Fig 9 Architecture of transposition buffer.

The block diagram for image reconstruction is as shown in fig 10. This process is just reverse of the compression techniques done. Quantisation matrix used in the compression is used to recover the transform matrix. But the error obtained is not retrieved. The inverse transform for the reconstruction is shown in fig 11.

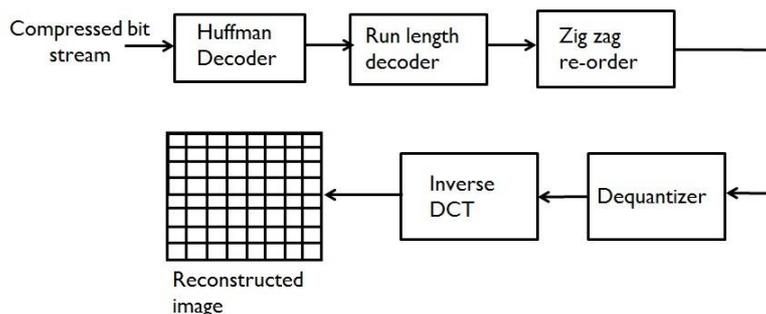


Fig 10 block diagram for image decompression

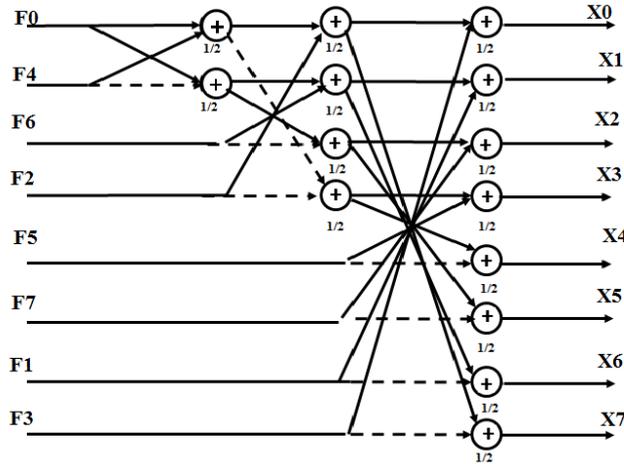


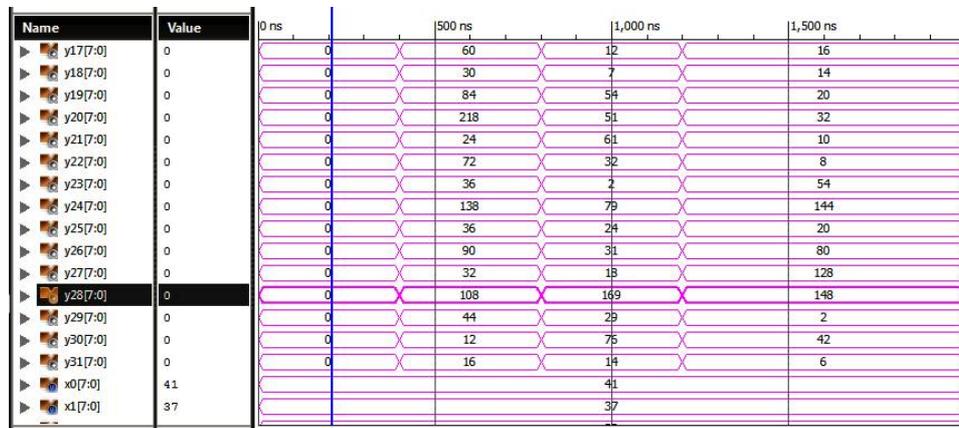
Fig 11 IDCT with the 14 adder

V. EXPERIMENT RESULT

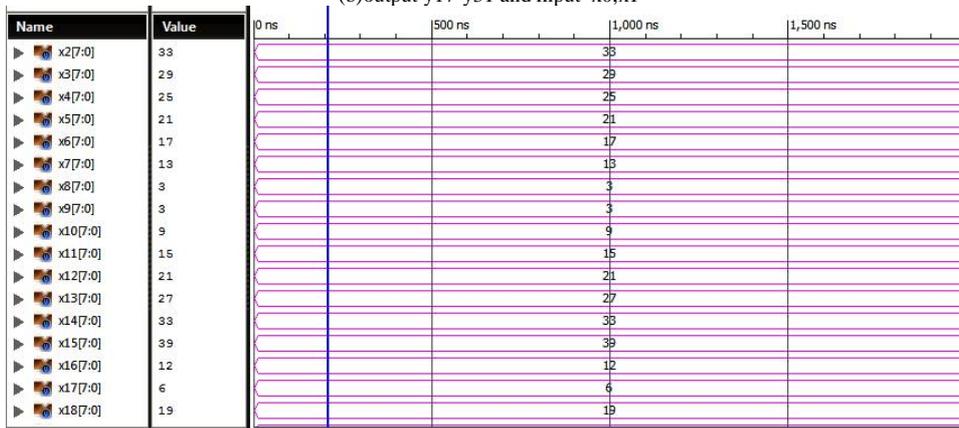
The simulation result of reconfigurable architecture for DCT of length 8, 16 and 32 using the approximation of 14 addition is shown in fig 12. The input are denoted by X0-X31 and output by Y0-Y31 in fig.

Name	Value	0 ns	500 ns	1,000 ns	1,500 ns
y0[7:0]	208	0	110	216	208
y1[7:0]	130	0	66	20	130
y2[7:0]	126	0	50	0	126
y3[7:0]	42	0	10	28	42
y4[7:0]	74	0	6	0	74
y5[7:0]	8	0	6	4	8
y6[7:0]	30	0	64	0	30
y7[7:0]	82	0	8	12	82
y8[7:0]	28	0	6	150	28
y9[7:0]	122	0	6	30	122
y10[7:0]	30	0	10	6	30
y11[7:0]	82	0	2	35	82
y12[7:0]	30	0	0	6	30
y13[7:0]	146	0	0	6	146
y14[7:0]	44	0	30	0	44
y15[7:0]	10	0	6	15	10
y16[7:0]	26	0	98	19	26

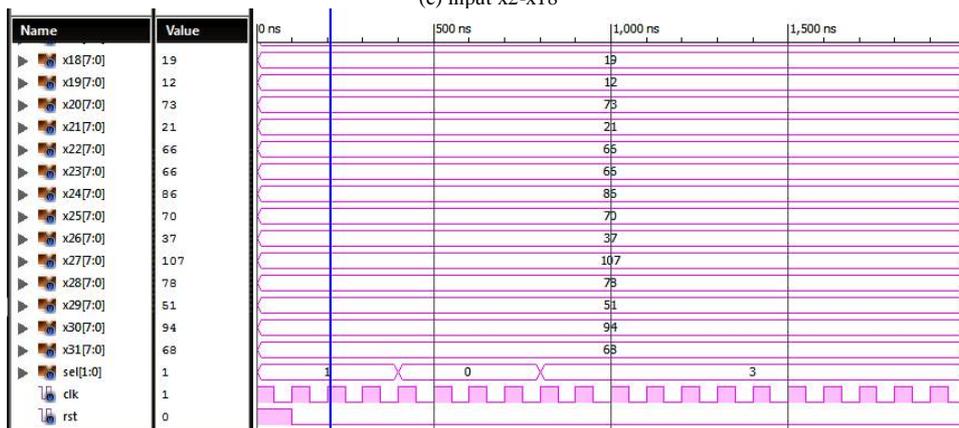
(a) Output y0-y16



(b)output y17-y31 and input x0,x1



(c) input x2-x18



(d) input x18-x31

Fig 12 input and output coefficients of reconfigurable 32 point DCT.

The image compression is done with the reconfigurable architecture thus proposed. Test image for the image compression “lena” is taken from the standard database [9]. For simplicity only some portion of the image is taken for the experiment. ie, eye portion. The lena image, cropped lena image and reconstructed image is shown in fig 13. The compressed data in bit stream is shown in fig 14.

