International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-4 E-ISSN: 2347-2693

Performance Analysis of 4 FDCT Algorithms Using Hardware Synthesis and Simulation

Atri Sanyal[1*], Saloni Kumari[1], Amitabha Sinha[2]

[1*]Department of Computer Application, NSHM College of Management &Technology, Kolkata, India
[2]Department of Computer Application, NSHM College of Management &Technology, Kolkata, India
[3]Department of Computer Science and Engineering, Birbhum Institute of Engineering & Technology, Birbhum, India

[*]Corresponding Author:atri.sanyal@nshm.com, Tel.: +919432183834

Available online at: www.ijcseonline.org

***Abstract*—** In order to find out the best fast DCT algorithms presented among numerous algorithms,four Fast DCT Algorithms which are popular and frequently used are considered in the paper. Referring their dataflow graphs 4 architectures are designed using Matlab Simulink. HDL coder is used to generate automated VHDL code. The block setsused in the Simulink design are manually modified tothe fixed point 16-bit data type. VHDL code is generated using HDL coder. The designs are synthesized using Xilinx ISE 14.5. A test bench program is written to test the 4 algorithms with the same set of data. Using the test bench program, a post route simulation up to the pin level is executed. From the timing report and synthesis report, the results are compared to find out the best FDCT algorithm in terms of hardware utilization and simulated timing performance.Loeffler's Algorithm is performing the best, both in terms of hardware utilization and timing requirement as found from the hardware synthesis report and timing report after post route simulation.

***Keywords*:** FDCTAlgorithm, Dataflow diagram, Matlab Simulink, Xilinx synthesis, Post Route Simulation, Maximum padding delay, Maximum combinational path delay

I. INTRODUCTION

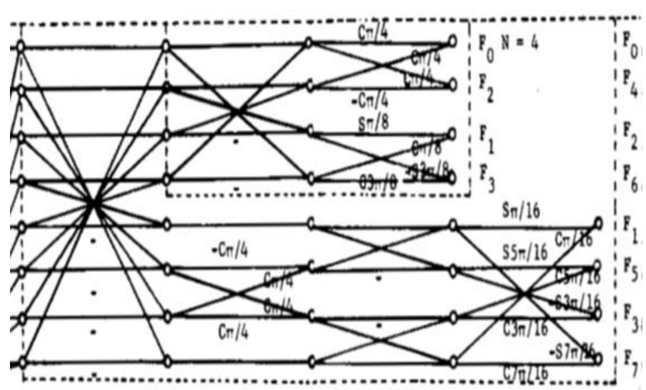
JPEG (Joint Photographic Experts Group) is a dominant format for still image compression. It is the first international standard in the image compression. JPEG is most widely used form of image compression that centers around DCT (Discrete Cosine Transform)[2]. In JPEG method,[3]-[6] total image matrix is broken into the 8*8 sub-blocks of the pixel and then working from left to right and right to the bottom, DCT is applied to each image block.Each block is compressed through quantization.In the beginning, upper-left hand corner of an image is chosen. DCT is designed to work on the pixel values ranging from -128 to 127, therefore the original block is levelled off by subtracting 128 from each entry.

The n rows of an N point DCT matrix T are defined by[1]:
1> For all i=1 to n : $(t1i=\sqrt{1/n})$
2> For all i=1 to n and k=2 to n :
$(t_{ki}=\sqrt{2/n}\cos((\pi(2i-1)(2k-1))/2n)$

The 8 point DCT matrix T (n=8) is defined as

0.3536 0.3536 0.3536 0.3536 0.3536 0.3536 0.3536 0.3536
0.4904 0.4157 0.2778 0.0975-0.0975-0.2778 –0.4157 -0.4904
0.4619 0.1913 -0.1913 -0.4619 -0.4619 -0.1913 0.1913 0.4619
0.4157 -0.0975 -0.4904 -0.2778 0.2778 0.4904 0.0975 -0.4157
0.3536 -0.3536 -0.3536 0.3536 0.3536 -0.3536 0.3536 0.3536
0.2778 -0.4904 0.0975 0.4157 -0.4157 -0.0975 0.4904 -0.2778
0.1913 -0.4619 0.4619 -0.1913 -0.1913 0.4619 -0.4619 0.1913
0.0975 -0.2778 0.4157 -0.4909 0.4904 -4157 0.2778 -0.0975

From DCT matrix it is clear that Symmetries exist in DCT function and this can be used to reduce the computation load in DCT. The basic n point DCT requires n^2 Multiplication and n(n-1) additions to find the value of y=(T*original) where the original is the image pixel matrix. For 8*8 matrix, it will amount to 8*8=64 multiplication and 8(8-1)=56 addition. There is the number of Fast DCT algorithms[7]-[16], which try to improve the computational load by using the symmetries present in the DCT matrix. The four of these algorithms is considered in this paper. They are Chen's, Arai's, Jeong's and Loeffler's[7]-[10]. The dataflow diagram of these 4 FDCT algorithms are shown in the following picture. These dataflow diagrams are used to implement MATLAB Simulink model in the later stage.



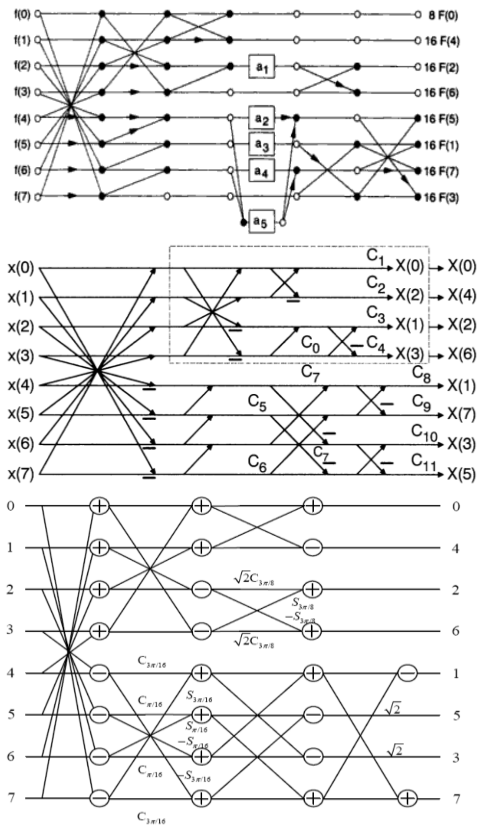


Figure 1. The dataflow diagrams taken from Chen's, Arai's, Jeong's and Loeffler's papers [7]-[10]

The paper has attempted to construct the aforesaid FDCT algorithms using Simulink building blocks, codify and synthesize them, and by downloading them in reconfigurable FPGA board executed post route simulation to find which one of them is better in terms of hardware utilization and timing requirement.

The rest of the paper is organized in the following order. In section II, the related works in this scope of the problem is discussed. In section III, Chen's FDCT algorithm is considered and the implementation of that algorithm in Matlab Simulink is discussed along with the necessary figure and a count of library blocks used in the implementation. The same is done for Arai's, Jeong's and Loeffler's FDCT algorithm in Section IV, V, and VI respectively. In section VII, the automated VHDL code generation by HDL coder,synthesized by Xilinx ISE and post route simulation using a test bench program is discussed. The conclusion from the synthesis and timing report and the future scope is discussed in section VIII.

II. RELATED WORKS

Image compression in JPEG format is a combination of two functions,especially discrete cosine transformation and matrix reductionmethods [6]. A number of the fast cosine transformations are available [7]-[16]. Each one of them has claimed to offer better results than the others in terms of various parameters like number of multiplications, Time parameter, simplicity, scalability etc. A very few attempts have been made in order to code them using a hardware description language like VHDL. Thentest them using a single test bench program, synthesize them in an FPGA ISE, download and see the timing simulation in order to find out the most efficient one in terms of hardware slice consumed and timing delay. A veryshort and incomplete attempt was done in [17] as that was not the prime focus of that study. We have elaborated the process and found the best FDCT in terms of hardware and timing requirement.

III. CHEN'S FDCT ALGORITHM [7]

First reported in 1977, it is one of the very first FDCT algorithm used extensively and is a fixed complexity algorithm. The FDCT calculations done with an 8x8 matrix require 16 multiplications and 26 additions. Chen's FDCT algorithm can be extended to any value of $N=2^m \geq 2$[7]. The Signal-flow graph presented in [7] and shown in figure 1 has been implemented in Matlab for N=8. The simple dot represents "addition", minus sign represents "subtraction", C represents "cosine function", S represents "sine function", $F_0$to F_8 represents output values. The Matlab implementation has been shown below:

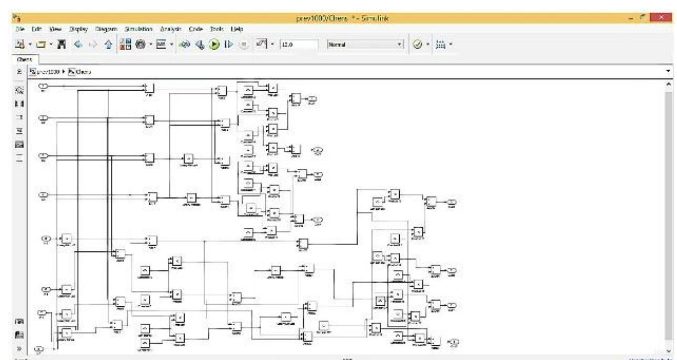


Figure 2. Matlab Implementation of Chen's Algorithm.

In Fig2, 8 input blocks of 16-bit signed integer (source) are taken for taking input, "ADD" blocks are used for "addition", "Unary minus" blocks are used for converting the value to negative, "Product" Blocks are used to multiply the values with constants, "Out" blocks of fixed 16-bit data type is used to display the output. The multipliers, adders and unary minus blocks of every stage is manually converted to fixed 16-bit data type as HDL coder could not automate the codification if the blocks are in floating point data type.

Table 1. Number of library blocks used in implementing Chen's algorithm.

I/O Block	Add	Unary Minus	Product	Constant
8	27	8	18	18

IV. ARAI'S FDCT ALGORITHM [8]

Introduced in 1988, this is one of the fastest algorithms as reported. The Signal-flow graph presented in [8] and shown in figure 1 has been implemented in Matlab for N=8. The FDCT algorithm takes 5 multiplications and 29 additions to compute DCT on an 8x8 pixel matrix f (0) to f(7) represent input values (Pixel values of image block). The black dot represents "addition". The straight line with an arrow represents "minus".
The square block represents "Multiplication". F (0) to F (7) represent input value. C_0 to C_7represent co-efficient used.

Table 2: Values of C in the data flow diagram of Arai's algorithm[8].

C0	C1	C2	C3	C4	C5	C6	C7
8	16	16	16	16	16	16	16

Table 3: values of constants[8]

a1	a2	a3	a4	a5
0.7071	0.5411	0.7071	1.3065	0.3826

Matlab Implementation of this FDCT algorithm is shown below:

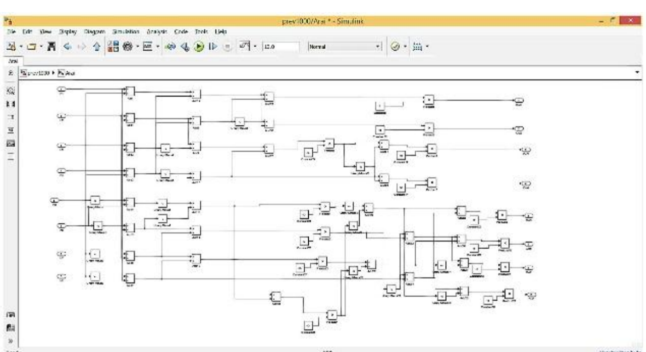


Figure 3. Matlab Implementation of Arai's Algorithm.

In Fig3, while doing the implementation the same procedure is followed in case of implementing the adder, unary minus, product, input and output blocks and the subsequent data type change. The required Simulink blocks for implementing the algorithm is shown in the following table.

Table 4. Number of Simulink Library blocks used in Arai's Algorithm.

I/O Block	Add	Unary Minus	Product	Constant
8	28	16	13	13

V. JEONG'S FDCT ALGORITHM [9]

Introduced in 1998, this is the novel FDCT with reduced number of multiplication and addition. Most of the multiplication was shifted to later stage so that propagation errors due to fixed point computation can be reduced.The FDCT required 12 multiplications and 28 additions to compute the DCT on an 8x8 pixel matrix. The data flow diagram of this FDCT presented in [9] is implemented in Matlab. x (0) to x(7) represent input values, which is the pixel value of the image, the straight line with an arrow represents "Addition", Minus sign represents "subtraction",C1 to C11 represent co-efficient, Constant values C_i) on the line represents "product", X (0) to X (7) represent output values.

Table5. Values of C[9]

C0	1/Cos(pi/4)
C1	1.414/4
C2	Cos(pi/4)/2
C3	Cos(pi/4)/Cos(pi/8)
C4	Cos(pi/4)/(4*C(pi/8))
C5	Cos(pi/8)/Cos(pi/8)
C6	1/Cos(pi/8)
C7	Cos((3*pi)/8)/Cos(pi/8)
C8	Cos(pi/8)/4*Cos(pi/16)
C9	Cos(pi/8)/4*Cos((7*pi)/16)
C10	Cos(pi/8)/4*Cos((3*pi)/16)
C11	Cos(pi/8)/4*Cos((5*pi)/16)

Matlab implementation of Jeong's algorithm is shown below:

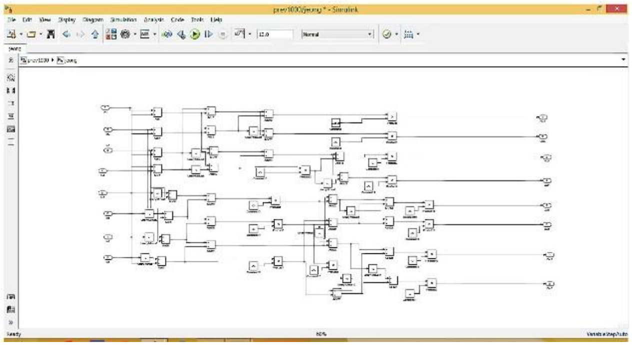


Figure 4. Implementation of Jeong's algorithm

In Fig4, while doing the implementation the same procedure is followed in case of implementing the adder, unary minus, product, input and output blocks and the subsequent data type change. The required Simulink blocks for implementing the algorithm is shown in the following table.

Table 6: Number of Simulink library blocks used in Jeong's Algorithm.

Input/Output Block	Add	Unary Minus	Product	Constants
8	28	12	13	13

VI. LOEFFLER'S FDCT ALGORITHM [10]

The algorithm was proposed on 1989. It is reported in many cases as one of the fastest way to compute DCT and IDCT computations. The algorithm requires 18 products and 27 additions to compute the DCT on an 8x8 pixel matrix. In data-flow diagram presented in [10], 0-7 represents input values, '+' sign represents "addition", '-' sign represents "minus", C represents "cosine function", S represents "sine function". Constants written on the line also represent product. Matlab implementation of this algorithm is shown below:

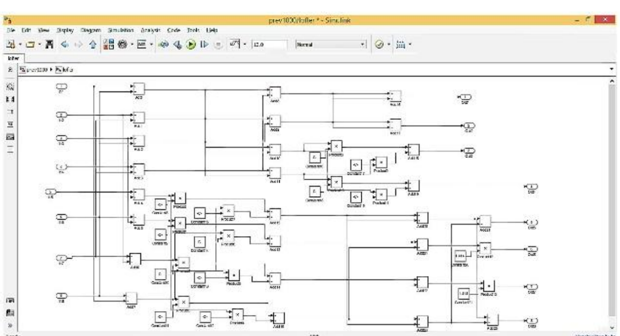


Figure 5. Implementation of Loeffler's algorithm.

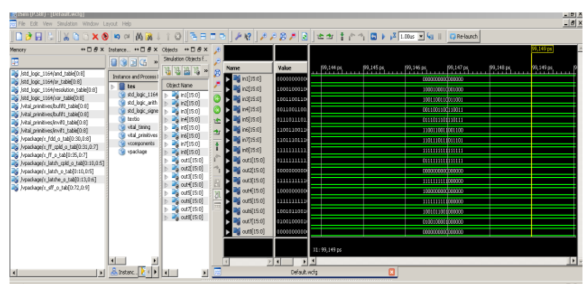
In Fig5, while doing the implementation the same procedure is followed in case of implementing the adder, unary minus, product, input and output blocks and the subsequent data type change. The required Simulink blocks for implementing the algorithm is shown in the following table.

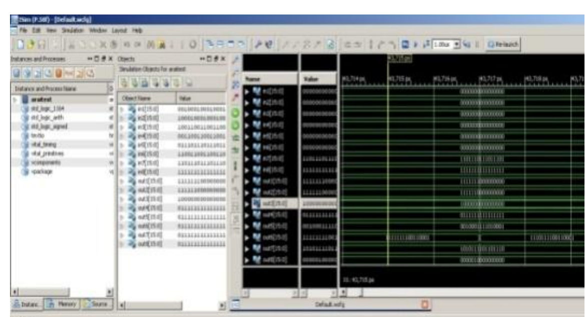
Table 7. Number of Simulink library blocks used in Loeffler's algorithm

Input Block	Add	unary minus	Product	Constants
8	15	11	14	14

VII. VHDL CODE GENERATION, HARDWARE SYNTHESIS AND TIMING SIMULATION

HDL coder automatically generates VHDL code for the four architectures. The codes are manually inspected and modified to minimize the signal loss. Next, the code is synthesized in Xilinx ISE 14.5. It is found that the required number of IOBs needed could be supported by a Virtex 7 series board and so that is chosen to synthesize the model. After synthesis, map and place and route (PAR) the synthesis report is available. Next, a test bench program in VHDL is written to test 4 FDCT architectures with that same set of data. We have chosen post route simulation among 4 available simulations (Behavioural, Translation, Post Map, Post Route) as it is closest to the original hardware timing simulation. The post route simulation result of the 4 FDCT algorithms is shown in the below figure:





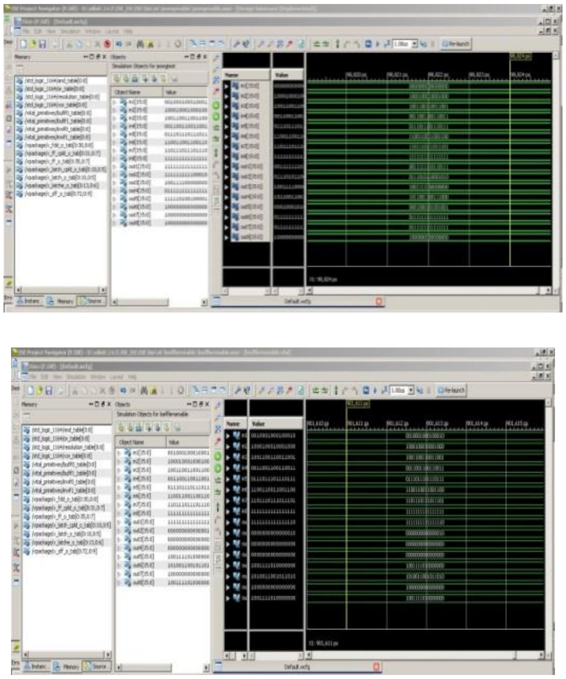


Figure 6. Post Route Timing Simulation Report of Chen, Arai, Jeong and Loeffler's FDCT respectively.

The synthesis and timing report is taken into consideration from which a number of features representing Hardware Requirement and Timing efficiency are selected.

Table 8: Comparison of 4 FDCT algorithms taken from Synthesis and post route timing report after post route timing simulation in Xilinx ISE 14.5

	Chen's	Arai's	Jeong's	Loeffler's
Number of Slice LUTs	658	907	820	588
Number of occupied Slices	275	374	339	226
Number of bonded IOBs	256	256	256	256
Number of DSP48E1s	18	13	13	11
Multiplier(s)	18	13	13	14
Adder/Subtractor(s).	34	44	40	33
Multiplexer(s)	122	128	114	98
Maximum combinational path delay:	17.51	21.16	18.21	14.49
Maximum Padding Delay after PAR	25.046	31.07	29.724	9.999

VIII. CONCLUSION AND FUTURE SCOPE

Among 4 FDCT algorithms, we can see that the number of slice LUTs is lowest in Loeffler's and a close second is Chen's. This indicates the hardware requirement of these two algorithms is lowest. All the four algorithms take, 8 numbers of 16-bit input and output, so number of required IOB is same in all. We can count the number of multipliers per algorithm as the floating point multiplication is the most time-consuming operations in these algorithms, in that respect, the lowest count came from Arai's and Jeong's. Moreover Jeong's algorithm postpones the multiplication at the later stage, which makes it faster as is visible from the maximum combinational path delay timing after synthesis and Maximum Padding Delay timing after Place and Route inbetween Arai's and Jeong's. The floating point multiplier is automatically implemented in the DSP logic block of the FPGA. Further, it was seen that for the first three algorithms the number of multipliers and number of DSP blocks are equal making this clear that all the multipliers were implemented in the DSP core, but in Loeffler's though the no of multipliers are one higher (14) than Arai's or Jeong's (13) but only 11 of them was implemented in DSP core making it actually the algorithm with the lowest numbered complex multiplier. We can see the superiority of the Loeffler's algorithm over the others in terms of Maximum combinational path delay time after synthesis and Maximum Padding Delay time after Place and Route, we can conclude that with respect to hardware utilization and maximum combinational delay Loeffler's is the best FDCT algorithm to use. The future scope of this paper is to further construct an improved version of a combined FDCT architecture as in [18] and to synthesize and simulate to get an accurate estimate of the efficiency of architecture which was absent in [18].This combined FDCT architecture then can be generalized for other transformations required in image processing as in [19] and obtain an accurate estimate of efficiency.

REFERENCES

[1].Ken Carben and Peter Gent, "Image Compression and Discrete Cosine Transform", Math45 college of Redwood

[2] Gregory K. Wallace, "The JPEG Still Picture Compression Standard" IEEE Transactions on Consumer Electronics, December, 1991.

[3] Rafael C. *Gonzalez*. University of Tennessee. Richard E.*Woods*, *"Digital Image*. *Processing* Third Edition."

[4]William B. Pennebaker, Joan L. Mitchell, "JPEG: Still Image Data Compression Standard", Springer Publications

[5] Wei-Yi Wei , "An Introduction to Image Compression", Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC

[6] A. Mardin, T. Anwar, B. Anwer, "Image Compression: Combination of Discrete Transformation and Matrix Reduction", International Journal of Computer Sciences and Engineering, Vol.5, Issue.1, pp.1-6, 2017

[7] W. Chen, C.H.Smith, and S.C.Fralick,"A fast computationalalgorithm for the discrete cosine transform,"IEEE, Trans, *COMM-25*, pp.1004-1009,Sep.1977.

[8].Arai Y, Aqui T, Nakajima M: A fast DCT-SQ Scheme for images, Trans IEICE #71 (1988), 1095-1097

[9].Yeonsik Jeong, Imgeun Lee, Hak Soo Kim, Kyu tae Park, "Fast DCT algorithm with fewer multiplication stage" , Electronics Letters 16th April 1988 vol.34, No. 8

[10]. C. Loeffler, A. Lightenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11multiplications", Proc. IEEE ICASSP, vol. 2, pp. 988–991, Feb. 1989.

[11] B.G. Lee, "FCT - A Fast Cosine Transform," IEEE International Conference on Acoustics, Speech and Signal Processing San Diego 1984, pp. 28A.3.1-28A3.4, March 1984.

[12] H. S. Hou, "A Fast Algorithm For Computing the Discrete Cosine Transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, No. 10, pp.1455-1461, Oct. 1987

[13] C. W. Kok, "Fast Algorithm for Computing Discrete Cosine Transform," IEEE Trans. Signal Process. , vol. 45, NO.3, pp.757- 760, Mar. 1977

[14] P. Lee and F.-Y. Huang, "Restructured Recursive DCT and DST Algorithms," IEEE Trans. Signal Process. vol. 42, NO. 7, pp.1600-1609, Jul. 1994

[15] Z. Cvetkovic and M. V. Popovic, "New Fast Recursive Algorithms for the Computation of Discrete Cosine and Sine Transforms," IEEE Trans. Signal Process., vol. 40, NO. 8, pp.2083-2086, Aug. 1992.

[16] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," Signal Process., vol. 6, pp. 267–278, Aug. 1984.

[17].Chen's-Yu-Pao,"Design and Evaluation of a Data Dependent Low Power 8*8 DCT/IDCT", A Master of applied science (Electrical) Thesis from Concordia University, Monheal, Quebec, Careda pp.9-14

[18] Atri Sanyal, Swapan K Samaddar, "A Combined Architecture for FDCT Algorithms", Proc IEEE 3rd International Conference on ICCCT 2012, Nov 23-25,2012, MNNIT Allahabad, India. IEEE Computer society, PP 33-37, ISBN: 978-0-7695-4872-2/12

[19].Swapan Kumar Samaddar, Atri Sanyal, Amitabha Sinha, "A Generalized Architecture for Linear Transform", Proc. IEEEInternational Conference CNC 2010, Oct 04-05, 2010, Calicut, Kerala,India.

Authors Profile

Atri Sanyal is an Assistant Professor in the Department of Computer Application of NSHM College of Management and Technology, Kolkata (affiliated to MAKAUT, WB). He is currently pursuing his Ph.D. from MAKAUT, WB under Prof. Amitabha Sinha in the field of Reconfigurable computing architecture of image processing applications. He holds a M.Sc. in Computer Science degree from Banaras Hindu University and a ME(CSE) degree from West Bengal University of Technology (currently renamed MAKAUT, WB). He has written 12 conference and journal papers and co-authored two books published by Lap Lombard publishing, Germany. His research interests are Image processing, Reconfigurable computing, Computer Architecture, Data Mining etc. He has guided a number of M.sc (CS) and BCA students in their projects. He has teaching and academic administration experience of more than 12 years.

Saloni Kumari is a final year student of BCA from NSHM College of Management and Technology Kolkata (affiliated to MAKAUT,WB). She is pursuing her final year project under the supervision of Atri Sanyal. Her research interest includes Computer architecture, Image and signal processing.

Dr. Amitabha Sinha is the Director of Birbhum Institute of Engineering and Technology, an AICTE approved Govt Aided College under MaulanaAbulKalam Azad University of Technology, West Bengal (MAKAUT,WB).
With a graduation in Electronics & Tele-Communication Engineering from Bengal Engineering College (Now IIEST), Shibpore and a PostGraduation in Electronics from University of Kent at Canterbury (U.K.), Prof. Sinha holds a Ph.D degree in Computer Sc. &Engg. from Indian institute of Technology (IIT), Delhi which he had obtained in 1984. He is a Fellow of the Institute of Engineers (India). Prof. Amitabha Sinha has been working in industry, premier academic institutes, R&D centers and IT/Telecom organizations in India & abroad for more than thirty two (32) years and his areas of research include Embedded System Design, VLSI design, Digital Signal Processing, Re-configurable Architecture using FPGAs, Software Defined Radio, Processor Architecture and System On-chip Design, etc. He had published more than 85 research papers in International journals and conferences, out of which more than 30 in journals. He has co-authored four books published by Lap Lambard publishing, Germany.Prof. Sinha had chaired a no of conferences including IEEE and delivered invited talks in India, U.S.A., Singapore, China, Germany, Russia, Australia and Hong-Kong. He has guided more than 100 M. Tech. students and a number of Ph.D students.