ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Design and Simulation of Floating Point FFT Processor Based on Radix-4 Algorithm Using VHDL

Akashadip A. Jiwane[1], Prof. Prashant R. Indurkar[2], Prof. Ravindra D. Kadam[3]

M. Tech Student (VLSI), Department of EXTC Engineering, BDCOE, Wardha, India[1]

Associate Professor, Department of EXTC Engineering, BDCOE, Wardha, India[2]

Assistant Professor, Department of EXTC Engineering, BDCOE, Wardha, India[3]

ABSTRACT: The Fast Fourier Transform (FFT) is one of the rudimentary operations in field of digital signal and image processing. Some of the applications of the fast Fourier transform include Signal analysis, Sound filtering, Data compression, Partial differential equations, Multiplication of large integers, Image filtering etc. Fast Fourier transform (FFT) is an efficient implementation of the discrete Fourier transform (DFT). The FFT can be designed by radix-4 butterfly algorithm which requires needless computations and data storage. We are here with presenting the design and simulation of a 32 bit floating point FFT processor. We have considered radix-4 DITFFT algorithm. The floating point number can support wide range of values. It is represented using three fields: sign, exponent and mantissa. In this paper floating point addition, subtraction and multiplication algorithms for IEEE-754 (single precision) is used. The IEEE-754 converter is used to convert decimal floating point number into IEEE Binary floating point format and it is also used to verify the results. For performance measurement of this design, various parameters like number of flip flops, number of LUTs, delay and complexity are obtained. Our aim is to design floating point radix-4 FFT processor because it is proposed that radix-4 algorithm requires less area and less time or delay as compared to radix-2 algorithm. The results are compared with existing design and are presented in this paper.

KEYWORDS: FFT, DIT, Floating point, Radix-2, Radix-4.

I. INTRODUCTION

Fast Fourier Transform (FFT) is widely used in the field of digital signal processing (DSP) such as filtering, spectral analysis etc., to compute the discrete Fourier transform (DFT). FFT is used in applications such as speech, audio, image, radar and biomedical signal processing. The FFT also plays a critical role in digital communications.

This Design follows 32 bit single precision IEEE 754 standards. Floating point representations supporting the four basic arithmetic operations; addition, subtraction, multiplication, division and logical operation are described in this section. The IEEE standard for floating point arithmetic (IEEE-754) is a technical standard for floating point computation established in 1985 by the institute of electrical and electronics engineers (IEEE). Many hardware floating point units use the IEEE-754 standard. IEEE floating point numbers have three basic components: the sign, the exponent, and the mantissa. The mantissa is composed of the *fraction* and an implicit leading digit. The exponent base (2) is implicit and need not be stored.

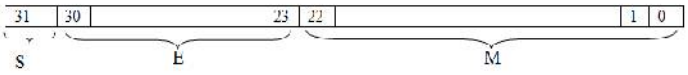

Figure 1: IEEE single precision floating point format

The radix-4 DIT-FFT recursively partitions a DFT into four quarter-length DFTs of groups of every fourth time sample. The outputs of these shorter FFTs are reused to compute many outputs, thus greatly reducing the total

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

computational cost. The radix-4 FFTs require only 75% as many complex multiplications as the radix-2 FFTs. The radix-4 decimation-in-time and decimation-in-frequency Fast Fourier transforms (FFTs) gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs. The radix-4 decimation-in-time algorithm rearranges the DFT equation into four parts: sums over all groups of every fourth discrete-time index n = [0, 4, 8 ... N - 4], n = [1, 5, 9 ... N - 3], n = [2, 6, 10 ... N - 2] and n = [3, 7, 11 ... N - 1], (This works out only when the FFT length is a multiple of four).

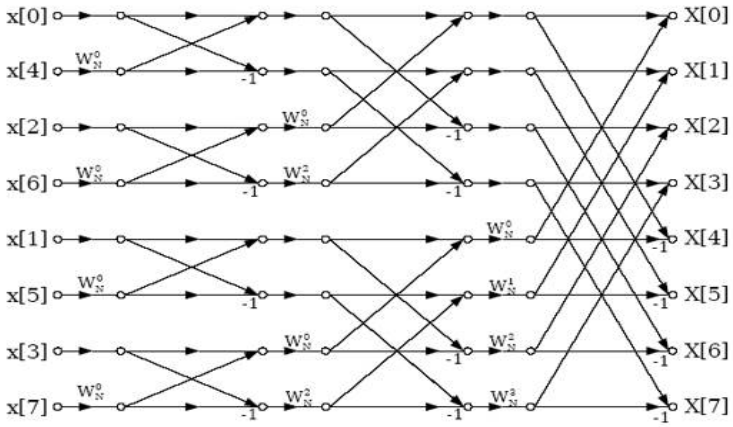

Figure 2: 8-point Fourier Transform

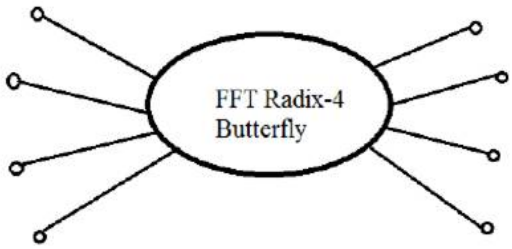

Figure 3: Basic structure of Radix-4

II. RELATED WORK

N. Amarnath Reddy, D. Srinivasa Rao and J. Venkata Suman [1] design and simulation of FFT processor using Radix-4 algorithm using FPGA. In this paper it is shown that a baseband ASIC can be fast and at the same time flexible with a low power consumption. The term fast do not refer to extreme clock frequencies but to the fact that no part of the designs needs more than one clock cycle to process a sample once the pipe is filled. Hence, the design does not need to be clocked any faster than the requested bandwidth and compared to modern CMOS technology this is a low number, in the order of 5-100 MHz .Sudha Kiran G, Brundavani P [2] proposed FPGA Implementation of 256-Bit, 64-Point DIT-FFT Using Radix-4 Algorithm. Human needs with technical devices are increasing rapidly. In order to meet their requirements the system should be accurate and fast. The fastness and accuracy of a system depends on its intra and inter peripherals/algorithms. In the view of this, the proposed paper came into existence. It focuses on the development of the Fast Fourier Transform (FFT) algorithm, based on Decimation-In- Time (DIT) domain, called Radix-4 DIT-FFT algorithm. VHDL is used as a design entity and for simulation Xilinx ISE. The synthesis results show that the computation for calculating the 256-bit 64-point FFT is efficient in terms of speed and is implemented on FPGA Spartan-3E kit. Afreen Fatima [3] worked on designing and simulation of 32 Point FFT using Radix-2 algorithm for FPGA. This paper concentrates on the development of the Fast Fourier Transform (FFT), based on Decimation-In-Time (DIT) domain, Radix-2 algorithm; this paper uses VERILOG as a design entity. The input of Fast Fourier transform has been given by a keyboard using a test bench and output has been displayed using the waveforms on the Xilinx Design Suite 13.1 and synthesis results in Xilinx show that the computation for calculating the 32- point Fast

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Fourier transform is efficient in terms of speed. B. Tharanidevi and R. Jayaprakash [4] proposed implementation of double precision floating point radix-2 FFT using VHDL. The Discrete Fourier Transform (DFT) can be implemented very fast using Fast Fourier Transform (FFT). It has various numbers of applications in the field of signal processing. The FFT can be designed by radix-2 butterfly algorithm which requires needless computations and data storage. It consumes more power. Using IEEE-754 single precision and double precision floating-point format the Fast Fourier Transform (FFT) for real numbers can be computed which is implemented in hardware FPGA. This paper describes two new architectures for floating-point addition, subtraction and product which are repeatedly used in radix-2 butterfly algorithm. This new architecture reduces computation complexity, data storage, area, and power consumption. The algorithms are simulated using VHDL language. Michael J [5] worked on the implementation and comparison of Radix-2 and Radix-4 FFT algorithms. In this project overall analysis is that, the Radix-4 implementation executes faster on the platforms tested than the authors Radix-2 algorithm.

III. RESULT AND SIMULATION

3.1 calculation of 8 point floating point DITFFT

To design the 8 Point FFT processor we have use Decimation in Time method (DIT). In DIT method we have to take different input in the form of odd & even numbers. The final FFT output is taken from serially Y (1) to Y (8**).**

Input: Each input of FFT is represented as real & imaginary values.

X(1) = 15.5 - 10.5j
X(2) = 18.5 - 12.5j
X(3) = 13.5 - 11.5j
X(4) = 16.5 - 13.5j
X(5) = 15.5 - 10.5j
X(6) = 18.5 - 12.5j
X(7) = 13.5 - 11.5j
X(8) = 16.5 - 13.5j

Real input values
= {15.5, 18.5, 13.5, 16.5, 15.5, 18.5, 13.5, 16.5}
Imaginary input values
= {-10.5,-12.5,-11.5, -13.5, -10.5, -12.5, -11.5,-13.5}
Using IEEE-754 converter, the decimal floating point value is converted into binary IEEE-754 format.
15.5=01000001011110000000000000000000; -10.5=11000001001010000000000000000000;
18.5=01000001100101000000000000000000; -12.5=11000001010010000000000000000000;
13.5=01000001010110000000000000000000; -11.5=11000001001110000000000000000000;
16.5=01000001100001000000000000000000; -13.5=11000001010110000000000000000000;

Output: We get output values which are in the form of also real & imaginary values.

Y(1) = 128-96j
Y(2) = 0 – 0 j
Y(3) = 6-2j
Y(4) = 0 – 0 j
Y(5) = -12+8j
Y(6) = 0 – 0 j
Y(7) = 2+6j
Y(8) = 0 – 0 j

Real output values
= {128, 0, 6, 0, -12, 0, 2, 0}
Imaginary output values
= { -96, 0, -2, 0, 8, 0, 6, 0 }
The IEEE-754 binary representation of this output values are
128= 01000011000000000000000000000000; -96= 11000010110000000000000000000000;
6= 01000000110000000000000000000000; -2= 11000000000000000000000000000000;
-12= 11000001010000000000000000000000; 8= 01000001000000000000000000000000;

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

2= 01000000000000000000000000000000; 0= 00000000000000000000000000000000;

3.2 RTL View of 8 Point Floating Point FFT

The RTL view of 8 Point Floating Point FFT consists of inputs and output values. Each input and output is 32 bit. There are 16 inputs and 16 outputs. Out of 16 input, 8 inputs are real and 8 inputs are imaginary.

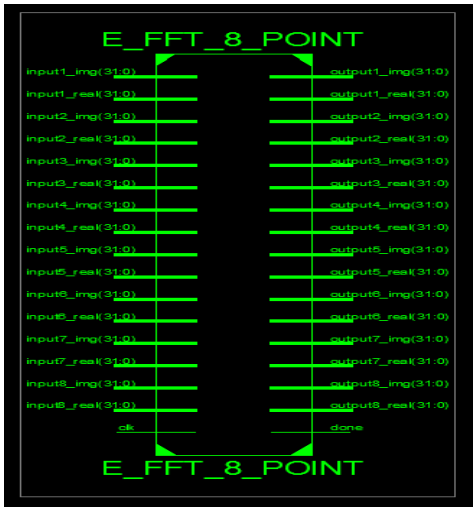

Figure 4: RTL view of 8 Point Floating Point FFT

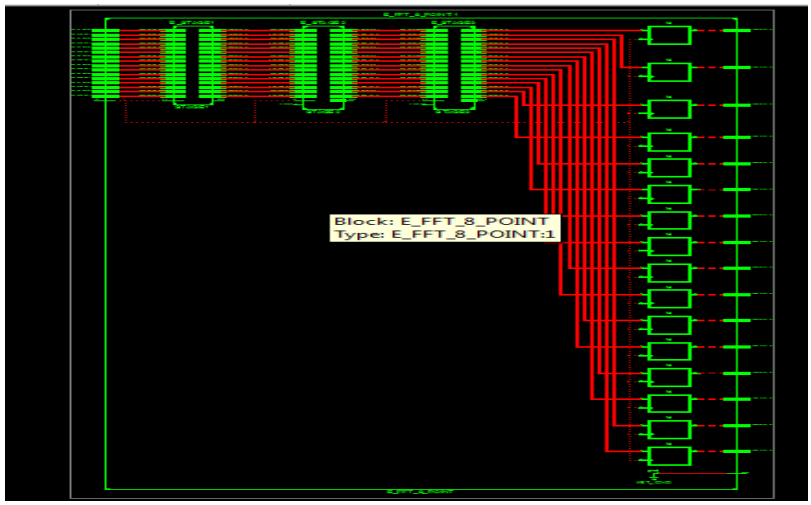

Figure 5: Internal RTL view of 8 Point Floating Point FFT

3.3 Simulation Result of 8 Point Floating Point FFT

The output of FFT in the form of IEEE-754 binary floating point format. There are total 16 output in which 8 output Are real & 8 outputs are imaginary.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

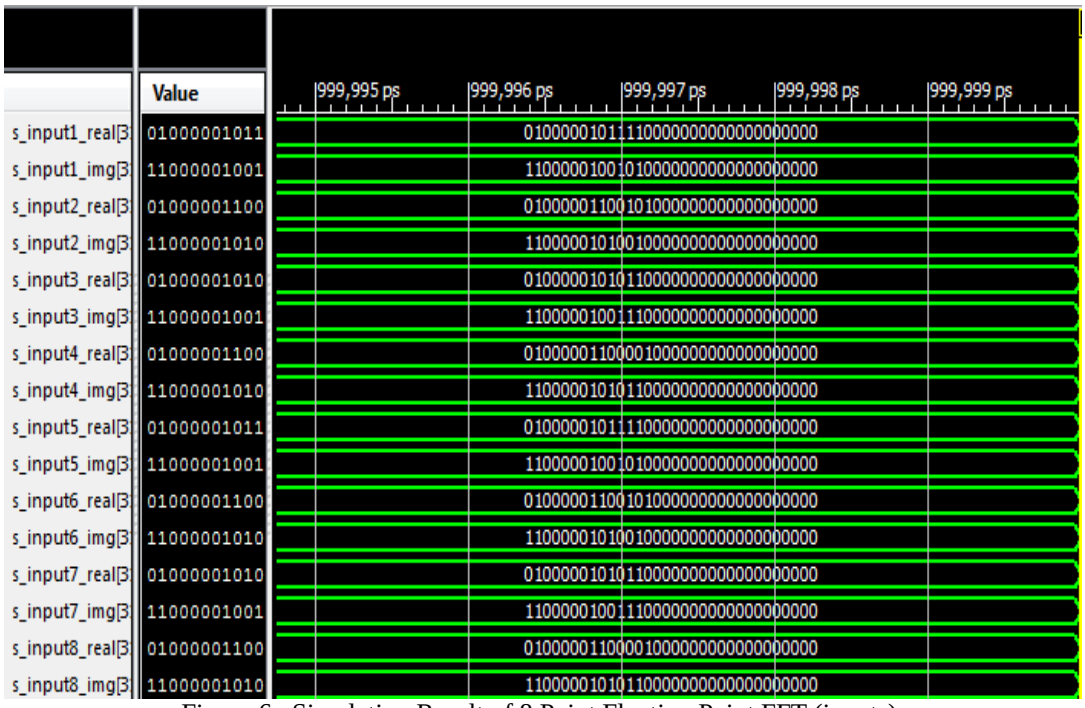

Figure 6: Simulation Result of 8 Point Floating Point FFT (inputs)

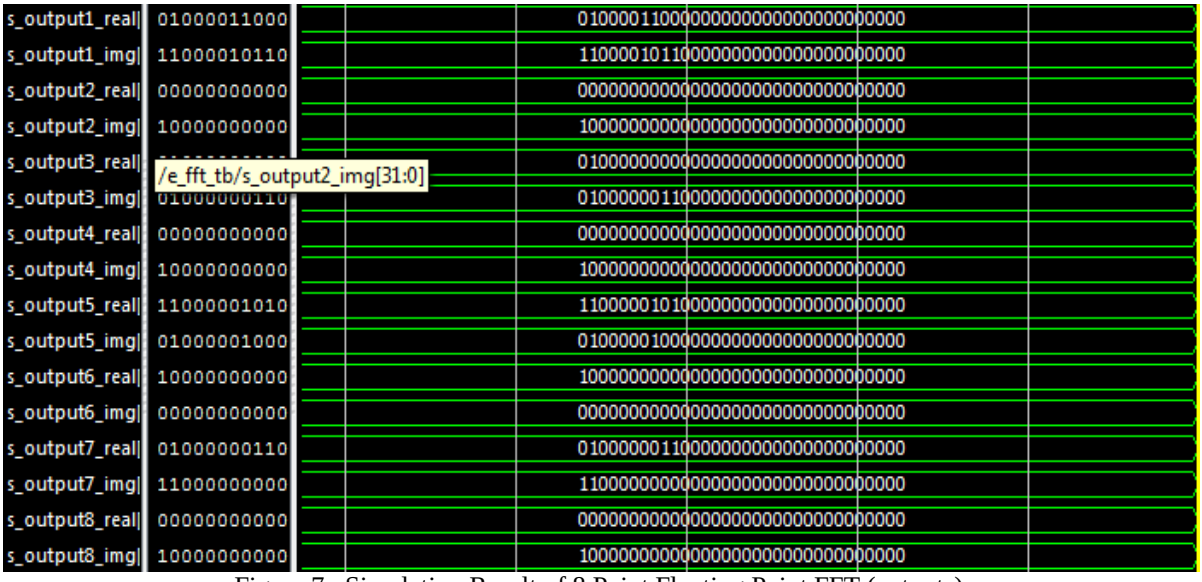

Figure 7: Simulation Result of 8 Point Floating Point FFT (outputs)

3.4 Comparison Table of Radix4 FFT

1) Design-A: It is presented by N**. Amarnath Reddy, D. Srinivasa Rao and J. Venkata Suman.** "Design and simulation of FFT processor using Radix-4 algorithm using FPGA". In International Journal of Advanced Science and Technology Vol.61, (2013), Pp.53-62 http://dx.doi.org/10.14257/ijast.2013.61.06.

2) Design-B: It is presented by **Gatla Srinivasa, p.Masthanaiah, p.Veeranath, Durga gopal.** "Vhdl implementation of a flexible and synthesible FFT processor based on radix 4". international journal of electrical, electronics and computer systems (ijeecs) , issn (online) 2347-2812, volume -1, issue -3, 2013 .

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

3) Design-C: It is a design presented by the Author in this paper.

Table 1: Compare design summary of FFT

	Design A Radix-4	Design B Radix-4	Design C Radix-4
Number of slices (Area)	14957	1174	1536
Number of 4 input LUT	25088	----------	54465
Number of bounded IOB	4096	62	1026
Gate delay	19.103 ns	2.1 us	30.398 ns
Net delay	14.247 ns	----------	11.687 ns
Logic gate used	74.6 %	----------	38.4 %
Product version	Xilinx ISE 10.1	Quartus II	Xilinx ISE 13.1
Selected device	Vertex-7	----------	Vertex-7

IV. CONCLUSION

In this paper, the 8 Point floating point FFT Processor is design and simulated using VHDL tools. The VHDL code has been successfully synthesized using Xilinx ISE 13.1i tools and simulated using ISE Simulator. The output values of FFT are verified using MATLAB tool and IEEE-754 converter. The Minimum period required for computation of FFT is 30.398 ns. The Floating point FFT achieved the maximum frequency of 32.897 MHz with delay 30.398 ns and area of 1536 slices. The total memory requirement is 623812 kilobytes. From the above comparison table it is clear that the design C of radix4 FFT has better delay and area as compare to others.

V. FUTURE SCOPE

The floating point supports the wide range of values. The floating point FFT Processor for double precision (64 bit) can also be designed and simulated. Also this design can be implemented using suitable hardware platform. In this paper, we have presented design & simulation of 8 point floating point FFT processor and the same design can be extended to 16 point & 32 point.

REFERENCES

[1] N. Amarnath Reddy, D. Srinivasa Rao and J. Venkata Suman, "Design and simulation of FFT processor using Radix-4 algorithm using FPGA," in International Journal of Advanced Science and Technology Vol.61, (2013), Pp.53-62 http://dx.doi.org/10.14257/ijast.2013.61.06.

[2] Gatla Srinivasa, p. Masthanaiah, p.Veeranath. Durga gopal, "vhdl implementation of a flexible and synthesible FFT processor based on radix 4" international journal of electrical, electronics and computer system(ijeecs) Issn (online) 2347-2812, volume -1, issue-3,2013.

[3] K. Jayaram and C. Arun, "Survey report for Radix-2, Radix-4 and Radix-8 FFT Algorithms," in International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 7, July 2015.

[4] Sudha Kiran G, Brundavani P, "FPGA Implementation of 256-Bit, 64-Point DIT-FFT Using Radix-4 Algorithm," in International Journal of Advanced Research in Computer Science and Software Engineering 3(9), September - 2013, pp. 126-133.

[5] Afreen Fatima, "Designing and simulation of 32 Point FFT using Radix-2 algorithm for FPGA," in IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331, Volume 9, Issue 1 Ver. III (Jan. 2014), PP 42-50.

[6] Remya Ramachandran, Vanmathi.k, "Simulation of radix-2 fast Fourier transform using Xilinx," in international Journal of computer science engineering (IJCSE), vol. 3 no.02 mar 2014, ISSN : 2319-7323.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

[7] Manohar Ayinala, and Keshab K. Parhi, "FFT Architectures for Real-Valued Signals Based on Radix-2 and Radix-4 Algorithms," in IEEE transactions on circuits and systems, 2013.
[8] P S Raja Kumar, G. Surya Narayana Reddy "Implementation of FFT algorithm using floating point numbers in wimax communication system".International Journal of VLSI and Embedded Systems-IJVES ISSN: 2249 – 6556 "Vol 04, Article 07133; July 2013
[9] ChenluWu , Wei Cao, Xuegong Zhou, Lingli Wang , Fang Wang , Baodi Yuan "A Reconfigurable Floating-Point FFT Architecture."978-1-4673-6417-1/13/$31.00 ©2013 IEEE.
[10] Sukhvir Kaur , Parminder Singh Jassal " Synthesis of Double Precision Float-ing Point Multiplier using VHDL "Journal of Research in Electrical and Electronics Engineering (ISTP-JREEE) . Vol 3, Issue 2, March 2014.
[11] AddankiPurna Ramesh, Rajesh Pattimi "High Speed Double Precision Floating Point Multiplier "International Journal of Advanced Research in Computer and Communication Engineering"Vol. 1, Issue 9, November 2012.
[12] J. G. Proakis, Digital signal processing: principles, algorithms, and applications, Prentice-Hall Intemational, 1996.
[13] J.Bhaskar, A VHDL PRIMER,Third Edition.

BIOGRAPHY

Mr. Akashadip A. Jiwane received the B.E degree in Electronic and Telecommunication Engineering in the year 2014 from RTMNU, Nagpur, India. He is currently doing MTECH degree in VLSI system design from BDCOE, Wardha, under RTMNU, Nagpur, India. He has published two papers in IORD journal of science and technology. His research focuses on the floating point FFT based on Radix-4 using VHDL. He has successful coordination in national level Tech-Fest organized by AVBIT college of Engineering. He was president in department of EXTC in AVBIT. His areas of interest are digital signal processing, fast Fourier Transform (FFT), Communication and VLSI.

Prof. Prashant R. Indurkar received the MTECH degree in Electronic from RTMNU, Nagpur, India. He is currently working as Associate Professor in the department of EXTC, B.D. College of engineering, Wardha, India and his total teaching experience is 24 year. He has published 40 papers in National / International conferences and journals to his credit. He is the member of the ISTE and IETE. His areas of interest are Digital, Embedded system and FPGA.

Prof. Ravindra D. Kadam received the Electronic in the year 1995 from RTMNU, Nagpur and the MTECH degree in Electronic C-DAC in the year 2011 from RTMNU, Nagpur, India. He is currently working as Assistant Professor in the department of EXTC, B.D. College of engineering, Wardha, India. His teaching experience is 12 year and also 2 year experience in industry. He has published 30 papers in National / International conferences and journals to his credit. He is the member of the ISTE and IE. His research focuses on the FPGA implementation of memory less V.D. with hybrid register exchange method. His areas of interest are signal system, VLSI, control system and FPGA.