



Design of flexible FFT core for fast computing of digital signals in real time applications

Deepa S.

deepamanjulla@gmail.com

BNM Institute of Technology, Bengaluru, Karnataka

Dr. Yasha Jyothi M. Shirur

yashajyothimshirur@bnmit.in

BNM Institute of Technology, Bengaluru, Karnataka

ABSTRACT

To meet needs of the customers, real-time signals are processed dynamically in many of the wireless communications, audio/video processing and industrial control applications. The heart of any computation is DSP Processor. Traditionally, dedicated (application-specific) architectures are used for high performance DSP applications. This trend continues today as more digital signal processing and image/video processing algorithms are implemented on single chips. The three optimization factors which every designer will be concerned about are area, speed, and power. Out of which speed is an important factor to be considered in real-time applications. To facilitate the requirement of fast computing, a scalable and reusable FFT Processor is designed and verified for its functionality. In paper a radix-2 butterfly is designed and implemented for fast computation of digital signals which is scaled to perform 8-point DIT FFT and 16-point DIT FFT. The design also supports reconfigurability feature to meet the design specification.

Keywords: FFT, radix-2, booth multiplier, and fast computation, etc

1. INTRODUCTION

In DSP Processor, Fast Fourier Transform is an efficient algorithm that efficiently computes Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT). The computation speed, performance, accuracy has become vital in latest digital smart devices [1]. Direct calculation of DFT leads to N^2 complex multiplications and $N(N-1)$ complex additions, if there are large number values of N it would take large amount of computation, thus FFT takes less computations which is in the order of $O(N \log N)$ based on divide and conquer algorithm. By employing FFT the computations are reduced. New FFT Architectures with Single Path Delay Feedback (SDF) that attenuates the memory, Multipath Delay Feedback (MDF) and other hardware architectures are discussed in [2].

The feed forward architectures are used for high throughput, but these architectures have high latency [3] are mostly used for FFT implementations. FFT Processors are developed for

FFT/IFFT computations which are more effective than general purpose DSP processors. FFT Processors consist of an important factor, which is responsible for the fast computation, i.e. radix-2 butterfly.

Internal structure of radix 2 butterflies consists of complex multiplier and an addition/subtraction unit. Different architectures of radix 2 butterfly structures are proposed by changing multiplier and adder units. FFT computation associate multiplier and addition operations. -The performance of multiplier in butterfly is very low which affects the hardware units. Replacing the multiplier by booth multiplier can increase the Performance of the unit.

In this paper a 32-bit radix 2 butterflies with booth multiplier and carry look adder is designed using Xilinx ISE tool. Using this radix 2 butterflies a 8-point and 16-point FFT was designed and simulated. Results will prove that the radix 2 butterfly is flexible to design any FFT for fast computation of digital signals.

2. RADIX-2 DIT BUTTERFLY

Radix-2 Butterfly is an important part of FFT Computation. The radix 2 decimation in time works by decomposing N point time domain signal into N time domain signals [5]. It is realized by reversing bits of time domain data. Each reconstruction stage in radix-2 performs 2-point butterflies. To derive the algorithm, let us split the summation which involves $N/2$ data points in the first summation and second sum involves other $N/2$ data points [6].

$$X(K) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \text{ --- (1)}$$

$$X(K) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + W_N^{NK/2} \sum_{n=\frac{N}{2}}^{\frac{N}{2}-1} (n + \frac{N}{2}) W_N^{nk} \text{ --- (2)}$$

Decimating $X(K)$ equation (1) into even and odd numbered samples

$$X(2K) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x(n + \frac{N}{2})] \text{ --- (3)}$$

Where $K = 0, 1, 2, \dots \dots \dots \frac{N}{2} - 1$

$$X(2K + 1) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) - x(n = \frac{N}{2})] \dots \dots \dots (4)$$

Where $K = 0, 1, 2, \dots \dots \dots \frac{N}{2} - 1$

The computation steps are followed replicated over decimation of the $N/2$ point DFTs $X(2K)$ and $(2K+1)$ as shown in equation 3 and 4.

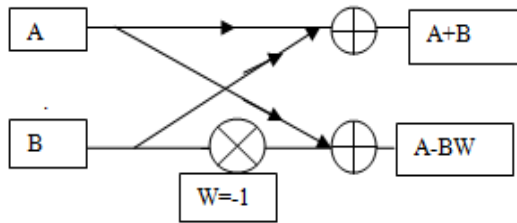


Fig. 1: Signal flow graph of DIT FFT

Radix 2 butterfly consists of multiplier and adder /sub tractor sub modules. Booth Multiplier is used instead of normal multiplier [4] is shown in figure-1. Booth multiplier performs signed binary numbers in 2's complement form. Multiplier consumes large area and more power.

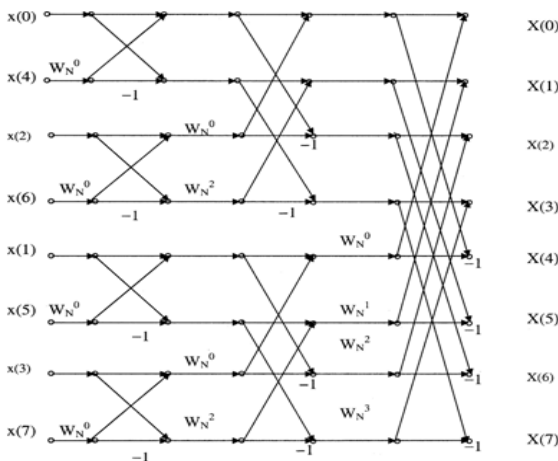


Fig. 2: Signal flow graph of 8-point DIT FFT

Realization of 8-point FFT using decimation in time (DIT) butterfly algorithm [8]. The constants are multiplied for the first two columns of the 8-point FFT signal flow are either 1 or j . In the third column, the multiplication of constants is addition/subtraction operation followed by a multiplication of which can be easily realized using shift and add operation is shown in figure 2. The data flow graph of 16-point DIT FFT is shown in figure 3. Each butterfly block receives 2 inputs, four stages of butterfly is required to build 16-point FFT. Twiddle factor is a constant [7]. The twiddle factor is multiplied by the butterfly inputs and, the result of the multiplication only deals the real part. Radix 2 butterfly designed is scalable for 4-point, 8-point, 16-point DIT FFT. Radix 2 butterfly designed using booth multiplier and carry look ahead adder is more efficient and flexible than other hardware structure of FFT.

Today's VLSI systems requires less area ,low power and less propagation delays .Hardware multipliers such as Wallace tree multiplier, booth multiplier, and booth encoded Wallace tree

multiplier[4].The carry look ahead adder is used in this radix 2 butterfly unit .It reduces the propagation delay and gives faster addition logic.

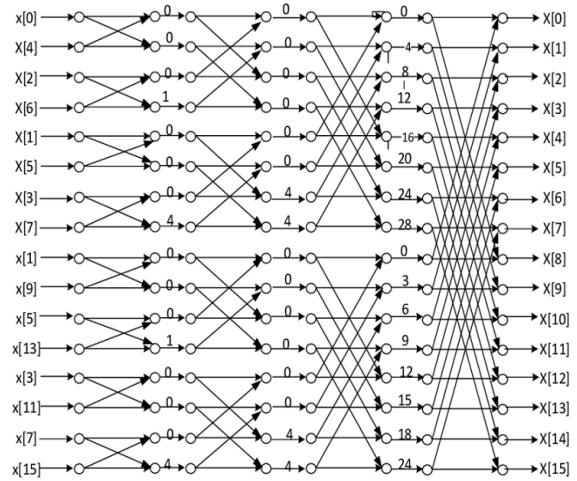


Fig. 3: Signal flow graph of 16-point DIT FFT

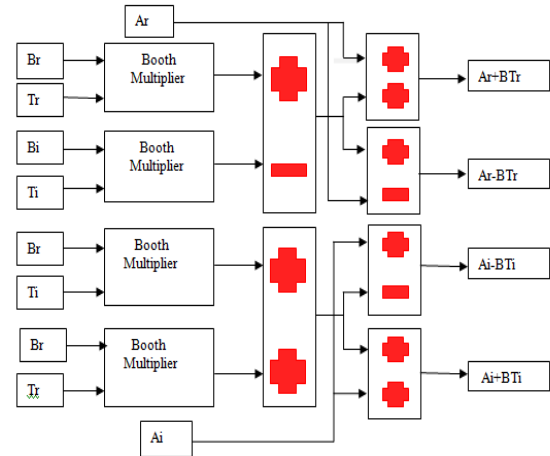


Fig. 4: Hardware architecture of Radix 2 Butterfly

The above block consists of booth multiplier and carry look ahead adder. The inputs are Ar, Br, Tr which are real number and Ai, Bi, Ti are imaginary inputs .It performs multiplication and addition/subtraction of inputs and produces outputs of real and imaginary outputs . $Ar+BTr, Ar-BTr$ are real outputs and $Ai+BTi, Ai-BTi$ are imaginary outputs.

3. ARCHITECTURAL DESIGN

After the design of 32bit radix 2 butterfly. It is inserted in 16-point32-bitFFT. The FFT computation consists of Radix 2 DIT butterfly, memory unit, LUT, address generation unit and control unit. The block diagram of FFT is depicted in figure 5a.

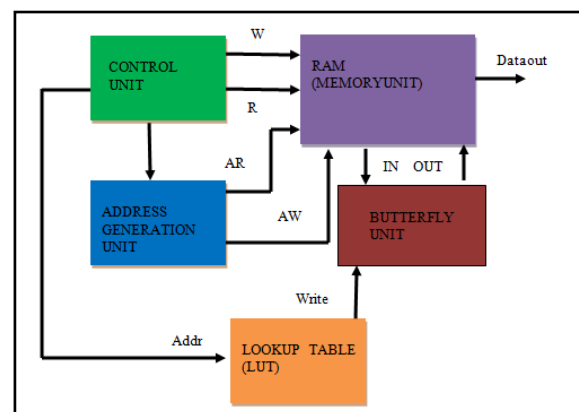


Fig. 5a: Block Diagram of FFT

Look up table (LUT) consists of twiddle factors where address generation unit obtains twiddle factors from LUT and stores the value to memory unit. The above block depicts following parameters W and R are inputs to the memory unit [5]. W represents write and R represents read .control unit reads and writes the address to the RAM [7] .The address generation unit produces valid addresses to the RAM as instructed by the control unit.AR and AW represents address read and address write respectively. Butterfly unit performs necessary computations based on the inputs given by AGU and RAM. The result is drawn to RAM to required address.

4. METHODOLOGY

The signal flow chart for designing this FFT is shown figure-5b. These are the steps followed for designing this project .Literature survey of radix 2 butterfly, FFT was done .according to design specification the radix 2 DIT was designed using booth multiplier and carry look ahead adder which are sub modules .HDL Description of these modules was written in Verilog and simulated in Xilinx ISE tool and these sub modules were joined together to build an top module called FFT. This methodology is shown in figure

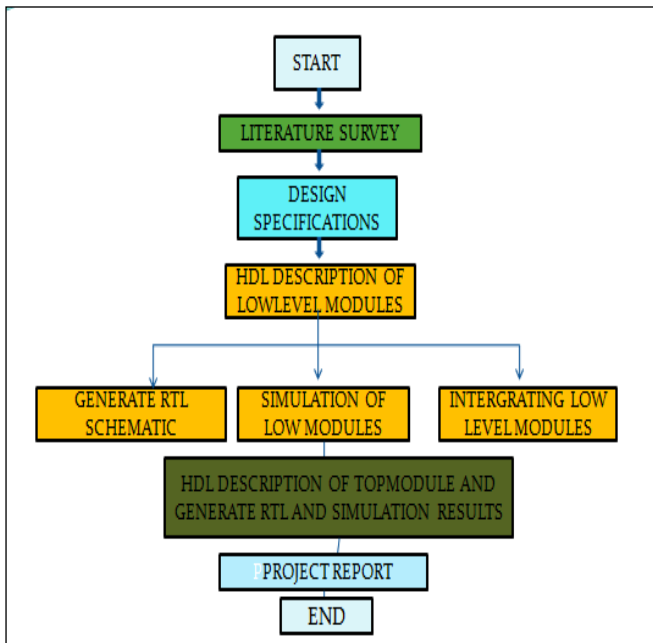


Fig. 5b: Flow Chart of Proposed FFT

5. RESULTS AND DISCUSSIONS

The 32bit radix 2 butterflies and 16-pointFFT was designed and simulated using Xilinx ISE Simulation tool. The result is truncated to 32bit MSB for each iteration. This radix 2 butterfly is introduced to FFT for computing of digital signals. The RTL schematic of 32-bit radix 2 butterfly unit is shown in figure-6a. It describes Ai ,Ar, Bi, Br are real and imaginary input signals and out1r,out1i,out2r,out2i are real and imaginary outputs and z1r_c, z1i_c, z2r_c, z2i_c are carryouts signals.

The output of radix 2 butterfly is shown in figure 6c for the inputs Ar=7, Ai=6, Br=3 ,Bi=1 ,Tr=3,Ti=1, when clk is high, it performs the required operation and z1r_c, z1i_c, z2r_c, z2i_c are the carryouts generated during the computation of inputs. The outputs calculated are as follows where out1r= 7+(3*3) =16, out1i= 6+(3*1)=9, out2r=(3*3)-7=2 ,out2i=(3*1)-6=3.These are the output manual calculation of radix 2 butterfly which is shown in figure 6c.

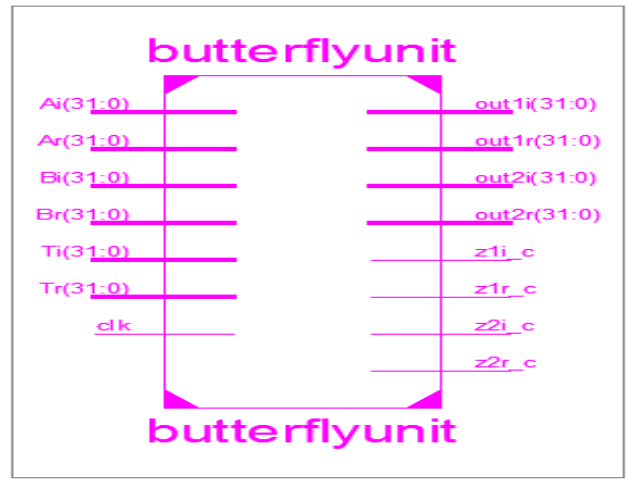


Fig. 6a: RTL Schematic of radix-2 butterfly

Internal RTL Schematic of this radix 2 butterfly unit is shown in figure-6b where booth multiplier and carry look ahead adder is shown below.

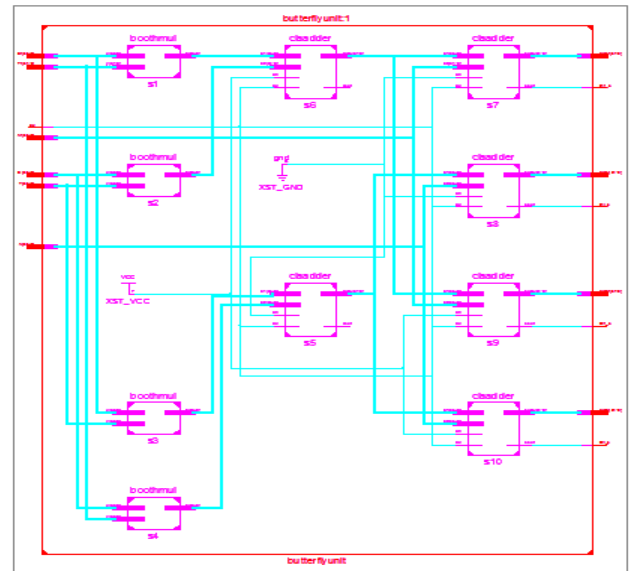


Fig. 6b: Internal Schematic of radix-2 butterfly

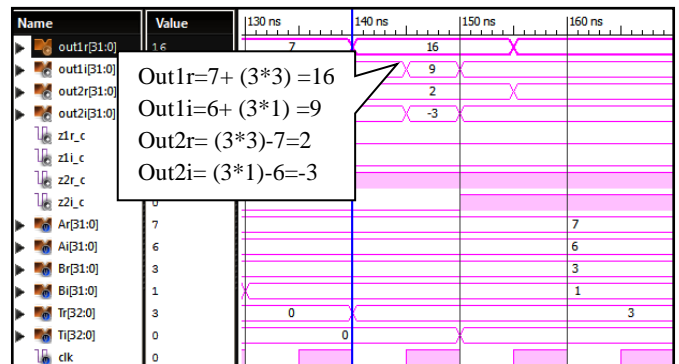


Fig. 6c: Simulation Result of Radix 2 Butterfly

RTL Schematic and simulation result of RAM (memoryunit) is shown in below figure. The inputs are read addr 1, read addr2, write addr1, write addr2, write data1 and write data 2.

The data will be stored in required address. When the address is read the data stored in that address will be displayed in the output.

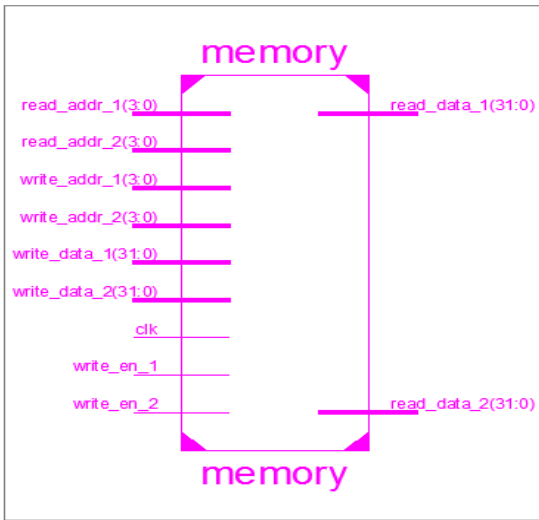


Fig. 7a: RTL Schematic of RAM (memory unit)

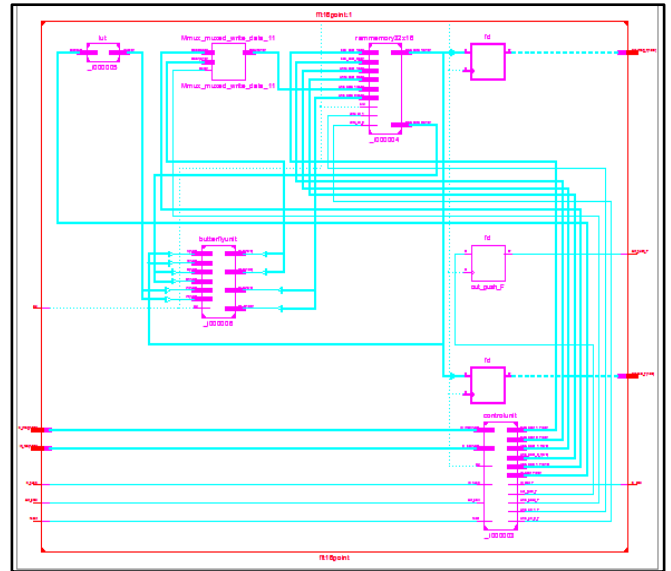


Fig. 8b: Internal RTL Schematic of 16-Point FFT

The input addresses are read addr1=4 and read addr2=6, the data stored in that address is write data1=12, write data2=7. outputs are read data1=12 and read data 2= 7 which is shown in figure 7b.

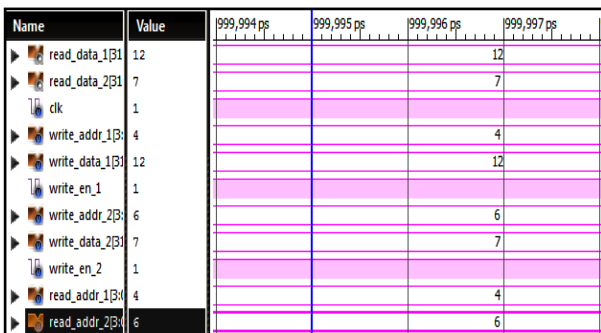


Fig. 7b: Simulation of RAM (memory unit)

RTL Schematic and simulation of 16-point FFT is shown below. In_real, in_imag, clk, inpush, outstallare inputs to the FFT. Inpush signal and outstall signals are given to control unit for pushing the address to memory unit.out_real, out_imag are outputs in frequency domain.

In_real, in_imag, clk, in_push, out_stallare inputs to the FFT. In_push signal and out_stall signals are given to control unit for pushing the address to memory unit.out_real, out_imag are outputs in frequency domain. Internal schematic of 16-point FFT contains LUT, Address generation unit, RAM, Control unit and radix 2 butterfly is shown in figure 8b

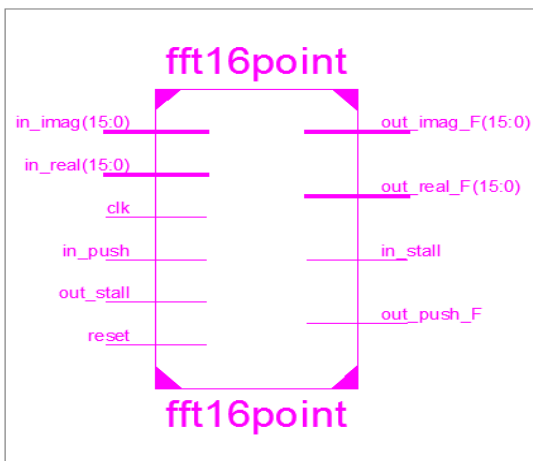


Fig. 8a: RTL Schematic of 16-Point FFT

The above simulation result shows all 16-points which are computed by FFT. The inputs `are real and imaginary points with clk. The inputs and outputs are calculated in MATLAB using functions. These outputs are calculated for reference to analyze the outputs in simulation and verify the outputs using MATLAB is equal with simulation results. Figure below shows the inputs and outputs of 16 point FFT using MATLAB where in represents inputs and out represents outputs.

in(1) = -2561 + i*-3377;	out(1) = -82 + i*-994;
in(2) = -2569 + i*-32188;	out(2) = -8807 + i*3553;
in(3) = -14817 + i*15351;	out(3) = 7156 + i*-8548;
in(4) = -28260 + i*-31024;	out(4) = 473 + i*77;
in(5) = -30765 + i*31905;	out(5) = 3658 + i*3208;
in(6) = -1165 + i*13370;	out(6) = -2149 + i*-4429;
in(7) = -14399 + i*-15020;	out(7) = -1169 + i*2887;
in(8) = 19194 + i*-16779;	out(8) = -3813 + i*3582;
in(9) = 30233 + i*-9814;	out(9) = -794 + i*9424;
in(10) = -3968 + i*6186;	out(10) = -739 + i*8453;
in(11) = 32245 + i*20839;	out(11) = 3176 + i*-5334;
in(12) = 27959 + i*17329;	out(12) = -5172 + i*-3190;
in(13) = 13304 + i*13889;	out(13) = -232 + i*-3490;
in(14) = -16235 + i*-13473;	out(14) = 7997 + i*5046;
in(15) = -20234 + i*13674;	out(15) = 2117 + i*-3754;
in(16) = 10751 + i*-26756;	out(16) = -4194 + i*-9881;

Fig. 8c: Inputs and outputs of 16-point FFT

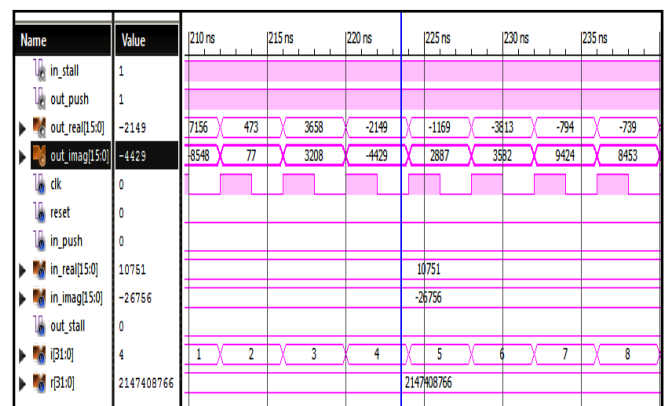


Fig. 8d: Simulation Result of 16-point FFT

6. CONCLUSION

In this paper, a scalable and reusable radix-2 butterfly is designed, implemented, and verified for the functionality. The radix-2 algorithm was proposed where radix 2 butterfly consists of booth multiplier and carry look ahead adder which reduces the complexity and helps in fast computation of digital

signals which is flexible for other FFT Processors for fast computation. Using which 8-Point FFT and 16-point FFT Processor using are performed to check the scalability. The design of radix-2 butterfly, 8-point FFT and 16 point FFT was simulated using Xilinx ISE tool.

7. ACKNOWLEDGEMENT

This project is carried under the flagship of *Project Bhageerathi Chip Development Program* initiated by the Dept. of ECE, BNMIT in association with Mentor Graphics-A Siemens Business Solution. We would like to thank BNMIT management for encouraging us to carryout research in the latest technology. And our sincere thanks to Visvesvaraya Technological University, Belgaum for motivating us to publish the paper.

8. REFERENCES

- [1] Ngoc-Hung Nguyen¹, Sheraz Ali Khan¹, Cheol-Hong Kim², and Jong-Myon Ki¹“An FPGA-Based Implementation of a Pipelined FFT Processor for High-Speed Signal Processing Applications”,*School of Electrical Engineering, University of Ulsan, Ulsan, South Korea 2017*
- [2] Brett W. Dickson, and Albert A. Conti“Parallel Extensions to Single-Path Delay-Feedback FFT Architectures”, *A. Conti is with Cognitive Electronics Inc., 201 South Street, Boston, MA 02111, 2014*
- [3] Tapsi Gupta, Janki ballabh Sharma, “A High-Speed Single-path Delay Feedback Pipeline FFT Processor using Vedic-Multiplier” *IEEE, 2017*
- [4] Arathi Ajay¹, Dr. R. Mary Lourde²” VLSI Implementation of an improved multiplier for FFT Computation in Biomedical Applications”,*Department of Electrical and Electronics Engineering BITS Pilani, Dubai Campus Dubai, UAE.*
- [5] Luis M. Ledesma-Carrillo, Misael Lopez-Ramirez, Ana L. Martinez-Herrera, Eduardo Cabal-Yepez, Arturo Garcia-Perez.” FPGA-Based Reconfigurable Unit for Image Filtering in Frequency Domain”,*Division de Ingenierias, Campus Irapuato-Salamanca, Universidad de Guanajuato / Carr. Salamanca-Valle km 3.5+1.8, Comunidad de PaloBlanco, 36700 Salamanca, Guanajuato, Mexico, 2013*
- [6] Lakshmi Santhosh, Anoop Thomas,” Implementation of Radix 2 and Radix 22 FFT Algorithms on Spartan6 FPGA”*IEEE, 4th ICCNT – 2013*
- [7] C. Gonzalez-Concejero, V. Rodellar, A. Alvarez-Marquina, E. Martinez de Icaya and P.Gomez-Vilda” An FFT/IFFT design versus Altera and Xilinx cores” *International Conference on Reconfigurable Computing and FPGAs, 2008*
- [8] Josue Saenz S., Juan J. Raygoza P., Edwin C. Becerra A.“FPGA Design and Implementation of Radix-2 Fast Fourier Transform Algorithm with 16 and 32 Points.” *ROPEC 2015 – Electronic*
- [9] Koushik Maharatna, Eckhard Grass, and Ulrich Jagdhold “Na 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM” *IEEE MARCH 2004.*