Design and Verification of Scalable, Re-usable 16-Point IFFT Core for DSP Engine

Hajeera Bee
VLSI Design and Embedded Systems (M.Tech)
Bnm Institute of Technology,
Bangalore, India

Dr. Veena S Chakravarthi
CTO, Sensesemi Technologies, Research Head,
Dept of Electronics and Communication
Bnm Institute of Technology
Bangalore, India

Abstract— **Due to the advancements in the Digital Signal Processing applications, the magnitude of the intensity of both Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) have also increased. FFT and IFFT are the primary blocks in building almost all the Digital Signal Processing Systems. They are said to be the counterparts, but tend to function in reverse mode. In order to reduce the mathematical operations, computation complexity, these Fourier Transforms are used, thereby providing the feasibility for the hardware implementation of the same. This paper proposes the design and verification of 16-Point Decimation-In-Time (DIT) Inverse FFT for both complex and floating point numbers. Experimental results show that the proposed 16-point IFFT architecture incorporating approximate Radix-2 Butterfly module achieves good efficiency and overall high precision results.**

Keywords—Fast Fourier Transform, Inverse Fast Fourier Transform, IEEE 754 Single Precision format.

I. INTRODUCTION

Inverse FFT is widely used Digital Signal Processing function with high computation complexity. Inverse Fast Fourier transforms (IFFT) performs the inverse of FFT, it converts a frequency domain function into time domain function. It primarily finds its applications in audio, video processing, filtering. Traditionally IFFT processors can be implemented on DSP or an ASIC. With the invention of FPGAs makes it possible to combine both the features of DSP and an ASIC for the implementation of IFFT processors. The significance of the IFFT and its functions are discussed [I].

In this paper, the implementation of IFFT block utilizes the Decimation-In-Time (DIT) Radix-2 Butterfly unit, which has complex adders, subtractor and complex multipliers, all these are further integrated. Radix-2 Butterfly unit plays an evident role, which is also considered to be the heart of any FFT or IFFT processing algorithms [II]. Henceforth the working of Butterfly unit is predominant in the IFFT processor. The proposed work includes the implementation of 16-point IFFT for complex numbers as well as the floating point numbers, with their verification.

A. General IFFT representation

The basic principle behind the IFFT algorithm is to break down input sequence of length N into smaller sequences. Let x(k) be an N-point sequence , where N is raised to the power of 2. IDFT x(n) of an N-point sequence can be mathematically given as follows:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,e^{j2\pi nk/N}, 0 \leq n \leq N-1 \quad \text{-------}(1)$$

where,
X(k) =Frequency domain samples
X(n)=Time domain samples
N=FFT size
K=0, 1, 2, ………, N-1
The exponential term given in Equation (1) represents the twiddle factor required for IFFT computations.

B. Proposed Solution

In order to cater to the need for precision in a higher order IFFT block, the initial step to be taken is the bit reversal of the inputs (convert to IEEE 754 format) given to the IFFT block, followed by complex addition, complex multiplication and complex subtraction. Floating point numbers are already said to be in IEEE 754 format and doesn’t need any conversion or change.
The floating point number can be represented by taking significand which is scaled using exponent, base for scaling used in this paper is 2.

$$\text{Number} = \text{significand x base}^{\text{exponent}} \quad \text{-------}(2)$$

C. Organization of the Paper

Section II of the paper deals with the system Integration. It gives a brief introduction about the approach adopted in the design. Section III of the paper emphasizes on the implementation of proposed work. Section IV of the paper deals with the presentation of the obtained results. Section V of the paper illustrates the conclusion and the future scope of the proposed design.

II. SYSTEM INTEGRATION

The major operating block of IFFT block is a butterfly block. To design a 16-Point IFFT block we also need to design a 8-Point, 4-Point, 2-Point butterfly wherein 2-point is the basic block. All these sub blocks are integrated together. The basic Radix-2 butterfly unit is as shown in the Fig.1.

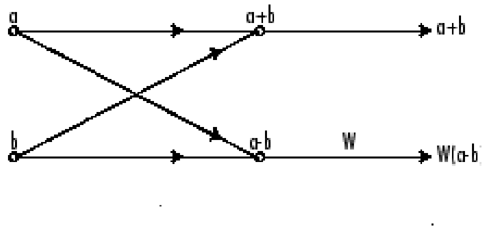


Fig.1. Basic Butterfly of DIT IFFT.

The basic radix-2 butterfly algorithm for Decimation-In-Time DIT-IFFT is shown in Fig. 1. The equations for the same can be noted as below,

$$C_r + jC_j = (A_r + jA_j)+ (B_r + jB_j)\text{-------}(3)$$

$$D_r + jD_j = (W_r+W_i)(A_r + jA_j)- (B_r + jB_j)\text{-------}(4)$$

A_r, A_j, B_r, B_j are the real and imaginary parts of the inputs a, b respectively. Also C and D are outputs with their real and imaginary parts as shown in Eqs 3 and 4.

In Fig. 1 a and b indicate the complex input from preceding stage while C and D indicate the complex output of the present stage (or complex input to the subsequent stage). The twiddle factors W_N are defined as the co-efficients which are used to compute results from the preceding stage and to get inputs to the next stages of IFFT algorithm.

The only difference between the butterfly of FFT and IFFT is the position of the twiddle factor which can be seen from the Fig.1.

III. IMPLEMENTATION

This section of the paper explains the algorithms used in the implementation of 16-Point IFFT taking account of all the stages required to design or build up an IFFT system. The 16-point IFFT architecture is shown in Fig.2.

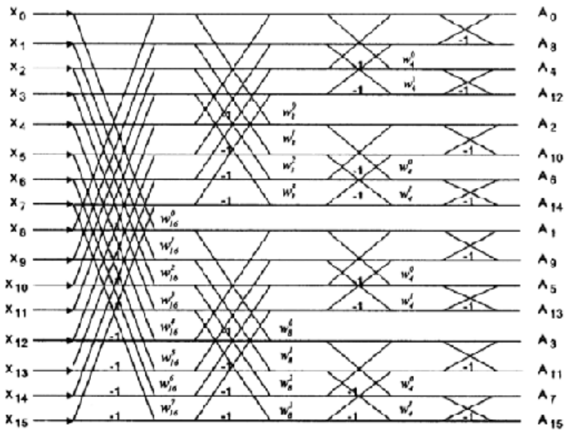


Fig.2. 16-point IFFT architecture.

The following aspects are considered in implementing the proposed design.

- Bit reversal is performed at the input sequence so as to improve the speed of the computations.
- It has the four stages, in each stage, complex addition, subtraction and multiplication are performed for complex values.
- This design also has IEEE 754 single format floating point numbers, so the floating point add, subtract and multiplication are also required.

The RTL schematic for 16-point IFFT for complex numbers is as shown below in Fig.3 and Fig.4.

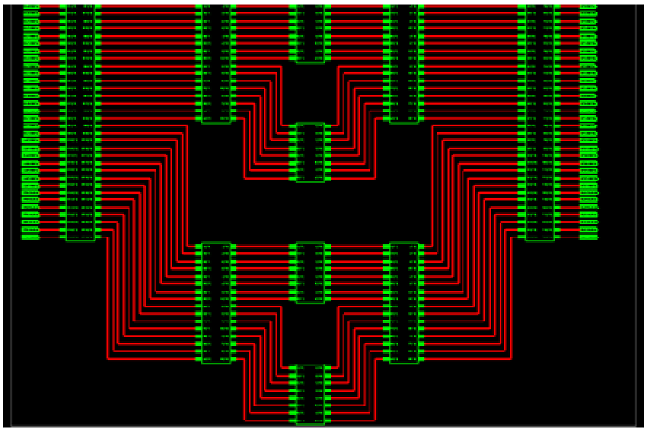


Fig.3. RTL Schematic of 16-point IFFT

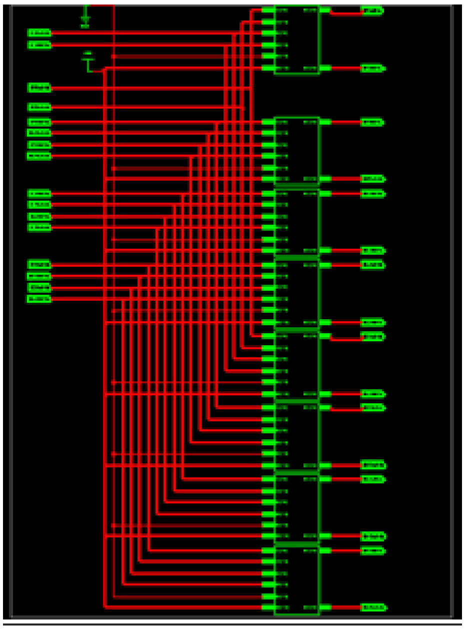
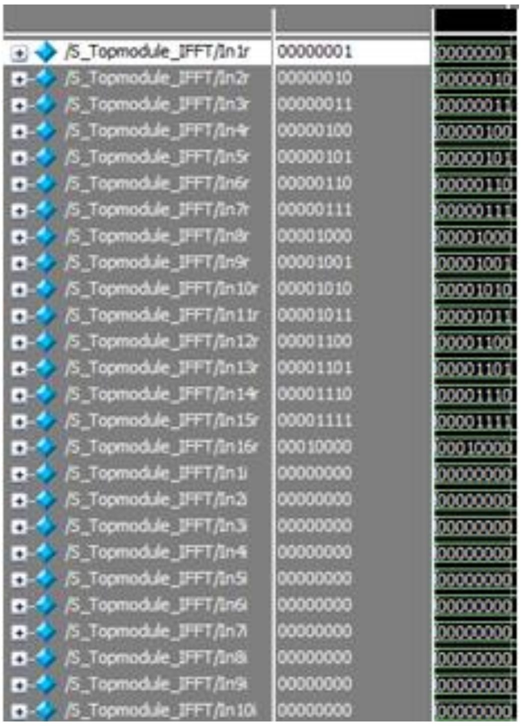


Fig.4. RTL Schematic of 16-point IFFT

The design approach is such that, a 8-point IFFT block is initially constructed using the above mentioned sub modules. The 16 point IFFT block is structurally developed using the 8-point IFFT modules. The design is realized using Verilog and simulated using the XILINX ISim.

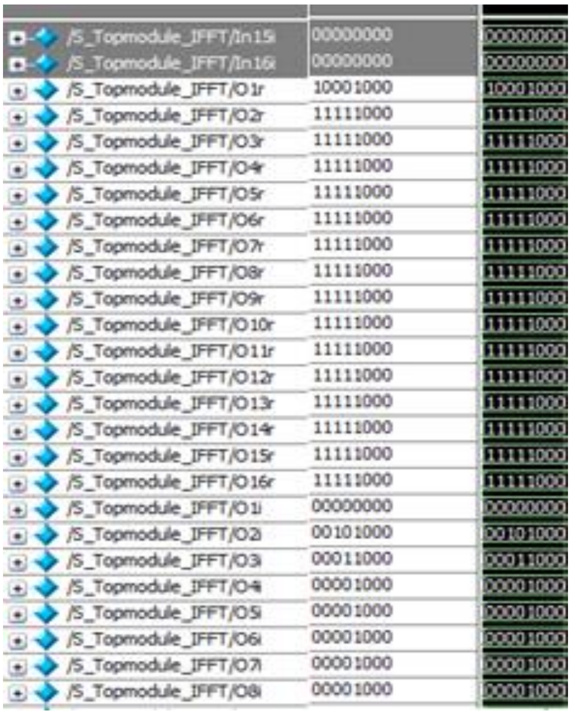
IV. RESULTS

The inputs given to the IFFT block is depicted in Fig.5. The simulation results for the16-point IFFT is shown in Fig.6 The verification is done using Modelsim and the resultant values are shown in figures [7] and [8].



```
/S_Topmodule_IFFT/In1r    00000001   00000001
/S_Topmodule_IFFT/In2r    00000010   00000010
/S_Topmodule_IFFT/In3r    00000011   00000011
/S_Topmodule_IFFT/In4r    00000100   00000100
/S_Topmodule_IFFT/In5r    00000101   00000101
/S_Topmodule_IFFT/In6r    00000110   00000110
/S_Topmodule_IFFT/In7r    00000111   00000111
/S_Topmodule_IFFT/In8r    00001000   00001000
/S_Topmodule_IFFT/In9r    00001001   00001001
/S_Topmodule_IFFT/In10r   00001010   00001010
/S_Topmodule_IFFT/In11r   00001011   00001011
/S_Topmodule_IFFT/In12r   00001100   00001100
/S_Topmodule_IFFT/In13r   00001101   00001101
/S_Topmodule_IFFT/In14r   00001110   00001110
/S_Topmodule_IFFT/In15r   00001111   00001111
/S_Topmodule_IFFT/In16r   00010000   00010000
/S_Topmodule_IFFT/In1i    00000000   00000000
/S_Topmodule_IFFT/In2i    00000000   00000000
/S_Topmodule_IFFT/In3i    00000000   00000000
/S_Topmodule_IFFT/In4i    00000000   00000000
/S_Topmodule_IFFT/In5i    00000000   00000000
/S_Topmodule_IFFT/In6i    00000000   00000000
/S_Topmodule_IFFT/In7i    00000000   00000000
/S_Topmodule_IFFT/In8i    00000000   00000000
/S_Topmodule_IFFT/In9i    00000000   00000000
/S_Topmodule_IFFT/In10i   00000000   00000000
```

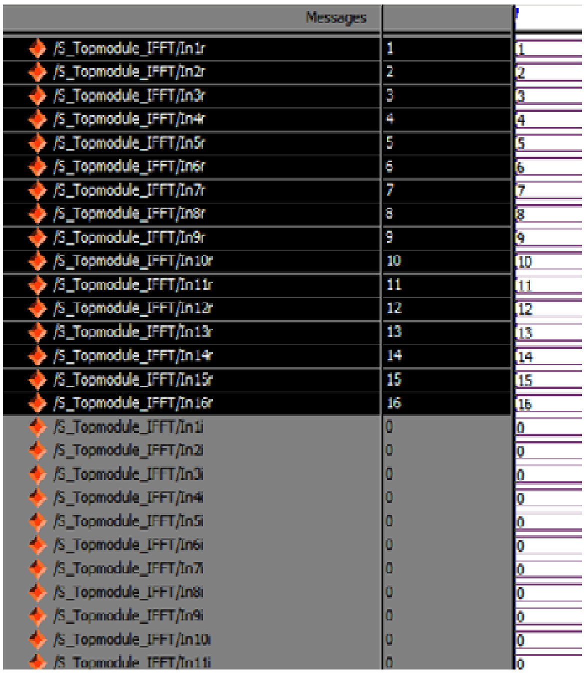
Fig.5. Inputs given to the IFFT block



```
/S_Topmodule_IFFT/In15i   00000000   00000000
/S_Topmodule_IFFT/In16i   00000000   00000000
/S_Topmodule_IFFT/O1r     10001000   10001000
/S_Topmodule_IFFT/O2r     11111000   11111000
/S_Topmodule_IFFT/O3r     11111000   11111000
/S_Topmodule_IFFT/O4r     11111000   11111000
/S_Topmodule_IFFT/O5r     11111000   11111000
/S_Topmodule_IFFT/O6r     11111000   11111000
/S_Topmodule_IFFT/O7r     11111000   11111000
/S_Topmodule_IFFT/O8r     11111000   11111000
/S_Topmodule_IFFT/O9r     11111000   11111000
/S_Topmodule_IFFT/O10r    11111000   11111000
/S_Topmodule_IFFT/O11r    11111000   11111000
/S_Topmodule_IFFT/O12r    11111000   11111000
/S_Topmodule_IFFT/O13r    11111000   11111000
/S_Topmodule_IFFT/O14r    11111000   11111000
/S_Topmodule_IFFT/O15r    11111000   11111000
/S_Topmodule_IFFT/O16r    11111000   11111000
/S_Topmodule_IFFT/O1i     00000000   00000000
/S_Topmodule_IFFT/O2i     00101000   00101000
/S_Topmodule_IFFT/O3i     00011000   00011000
/S_Topmodule_IFFT/O4i     00001000   00001000
/S_Topmodule_IFFT/O5i     00001000   00001000
/S_Topmodule_IFFT/O6i     00001000   00001000
/S_Topmodule_IFFT/O7i     00001000   00001000
/S_Topmodule_IFFT/O8i     00001000   00001000
```

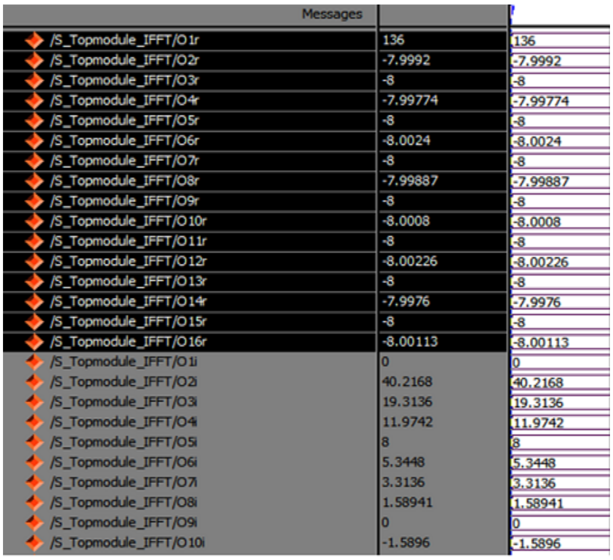
Fig.6. Simulation results of 16-point IFFT

The verification results for 16-point IFFT is as given in the figures below.



```
                               Messages
/S_Topmodule_IFFT/In1r    1    1
/S_Topmodule_IFFT/In2r    2    2
/S_Topmodule_IFFT/In3r    3    3
/S_Topmodule_IFFT/In4r    4    4
/S_Topmodule_IFFT/In5r    5    5
/S_Topmodule_IFFT/In6r    6    6
/S_Topmodule_IFFT/In7r    7    7
/S_Topmodule_IFFT/In8r    8    8
/S_Topmodule_IFFT/In9r    9    9
/S_Topmodule_IFFT/In10r   10   10
/S_Topmodule_IFFT/In11r   11   11
/S_Topmodule_IFFT/In12r   12   12
/S_Topmodule_IFFT/In13r   13   13
/S_Topmodule_IFFT/In14r   14   14
/S_Topmodule_IFFT/In15r   15   15
/S_Topmodule_IFFT/In16r   16   16
/S_Topmodule_IFFT/In1i    0    0
/S_Topmodule_IFFT/In2i    0    0
/S_Topmodule_IFFT/In3i    0    0
/S_Topmodule_IFFT/In4i    0    0
/S_Topmodule_IFFT/In5i    0    0
/S_Topmodule_IFFT/In6i    0    0
/S_Topmodule_IFFT/In7i    0    0
/S_Topmodule_IFFT/In8i    0    0
/S_Topmodule_IFFT/In9i    0    0
/S_Topmodule_IFFT/In10i   0    0
/S_Topmodule_IFFT/In11i   0    0
```

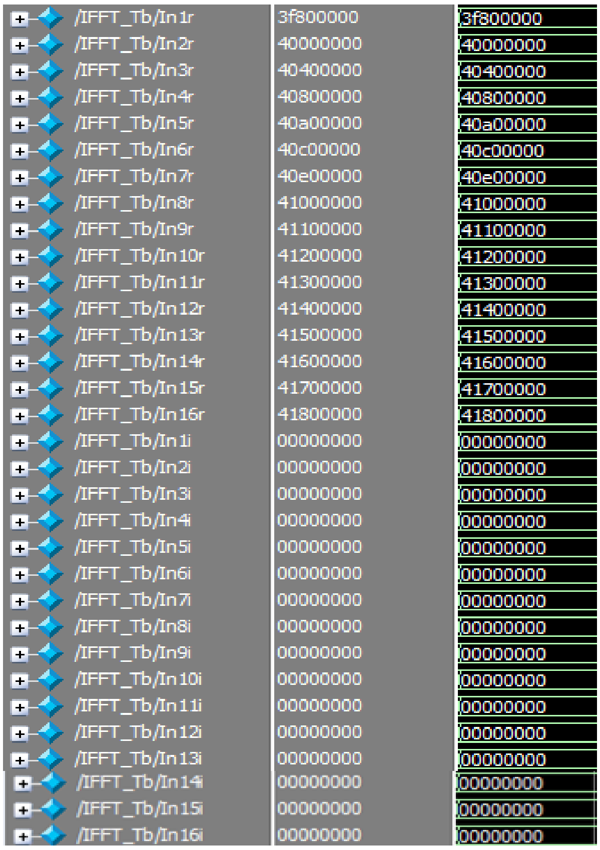
Fig.7. Inputs given to IFFT block



```
                               Messages
/S_Topmodule_IFFT/O1r     136        136
/S_Topmodule_IFFT/O2r     -7.9992    -7.9992
/S_Topmodule_IFFT/O3r     -8         -8
/S_Topmodule_IFFT/O4r     -7.99774   -7.99774
/S_Topmodule_IFFT/O5r     -8         -8
/S_Topmodule_IFFT/O6r     -8.0024    -8.0024
/S_Topmodule_IFFT/O7r     -8         -8
/S_Topmodule_IFFT/O8r     -7.99887   -7.99887
/S_Topmodule_IFFT/O9r     -8         -8
/S_Topmodule_IFFT/O10r    -8.0008    -8.0008
/S_Topmodule_IFFT/O11r    -8         -8
/S_Topmodule_IFFT/O12r    -8.00226   -8.00226
/S_Topmodule_IFFT/O13r    -8         -8
/S_Topmodule_IFFT/O14r    -7.9976    -7.9976
/S_Topmodule_IFFT/O15r    -8         -8
/S_Topmodule_IFFT/O16r    -8.00113   -8.00113
/S_Topmodule_IFFT/O1i     0          0
/S_Topmodule_IFFT/O2i     40.2168    40.2168
/S_Topmodule_IFFT/O3i     19.3136    19.3136
/S_Topmodule_IFFT/O4i     11.9742    11.9742
/S_Topmodule_IFFT/O5i     8          8
/S_Topmodule_IFFT/O6i     5.3448     5.3448
/S_Topmodule_IFFT/O7i     3.3136     3.3136
/S_Topmodule_IFFT/O8i     1.58941    1.58941
/S_Topmodule_IFFT/O9i     0          0
/S_Topmodule_IFFT/O10i    -1.5896    -1.5896
```

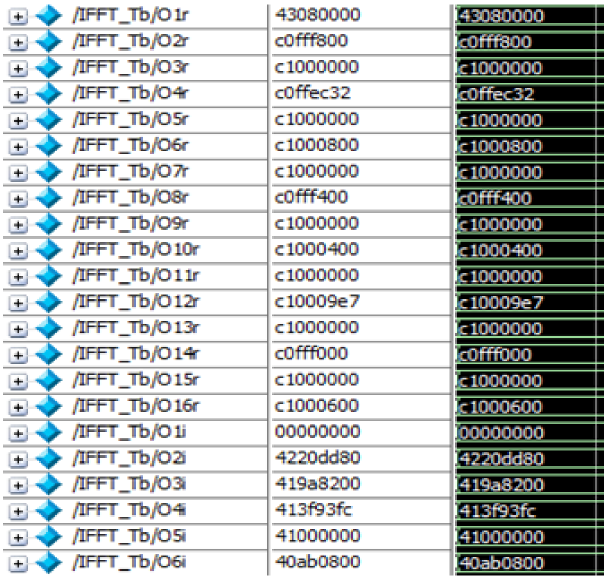
Fig.8. Verification results of 16-point IFFT

The Inverse FFT is carried out for floating point numbers too. The inputs given to the block is depicted in figures [9] and [10].

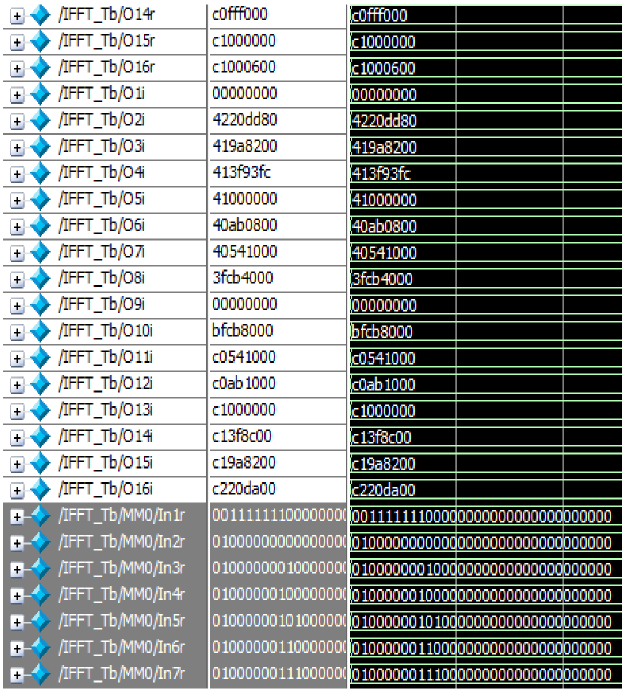


/IFFT_Tb/In1r	3f800000	3f800000
/IFFT_Tb/In2r	40000000	40000000
/IFFT_Tb/In3r	40400000	40400000
/IFFT_Tb/In4r	40800000	40800000
/IFFT_Tb/In5r	40a00000	40a00000
/IFFT_Tb/In6r	40c00000	40c00000
/IFFT_Tb/In7r	40e00000	40e00000
/IFFT_Tb/In8r	41000000	41000000
/IFFT_Tb/In9r	41100000	41100000
/IFFT_Tb/In10r	41200000	41200000
/IFFT_Tb/In11r	41300000	41300000
/IFFT_Tb/In12r	41400000	41400000
/IFFT_Tb/In13r	41500000	41500000
/IFFT_Tb/In14r	41600000	41600000
/IFFT_Tb/In15r	41700000	41700000
/IFFT_Tb/In16r	41800000	41800000
/IFFT_Tb/In1i	00000000	00000000
/IFFT_Tb/In2i	00000000	00000000
/IFFT_Tb/In3i	00000000	00000000
/IFFT_Tb/In4i	00000000	00000000
/IFFT_Tb/In5i	00000000	00000000
/IFFT_Tb/In6i	00000000	00000000
/IFFT_Tb/In7i	00000000	00000000
/IFFT_Tb/In8i	00000000	00000000
/IFFT_Tb/In9i	00000000	00000000
/IFFT_Tb/In10i	00000000	00000000
/IFFT_Tb/In11i	00000000	00000000
/IFFT_Tb/In12i	00000000	00000000
/IFFT_Tb/In13i	00000000	00000000
/IFFT_Tb/In14i	00000000	00000000
/IFFT_Tb/In15i	00000000	00000000
/IFFT_Tb/In16i	00000000	00000000

Fig.9. Floating point input values for 16-point IFFT



/IFFT_Tb/O1r	43080000	43080000
/IFFT_Tb/O2r	c0fff800	c0fff800
/IFFT_Tb/O3r	c1000000	c1000000
/IFFT_Tb/O4r	c0ffec32	c0ffec32
/IFFT_Tb/O5r	c1000000	c1000000
/IFFT_Tb/O6r	c1000800	c1000800
/IFFT_Tb/O7r	c1000000	c1000000
/IFFT_Tb/O8r	c0fff400	c0fff400
/IFFT_Tb/O9r	c1000000	c1000000
/IFFT_Tb/O10r	c1000400	c1000400
/IFFT_Tb/O11r	c1000000	c1000000
/IFFT_Tb/O12r	c10009e7	c10009e7
/IFFT_Tb/O13r	c1000000	c1000000
/IFFT_Tb/O14r	c0fff000	c0fff000
/IFFT_Tb/O15r	c1000000	c1000000
/IFFT_Tb/O16r	c1000600	c1000600
/IFFT_Tb/O1i	00000000	00000000
/IFFT_Tb/O2i	4220dd80	4220dd80
/IFFT_Tb/O3i	419a8200	419a8200
/IFFT_Tb/O4i	413f93fc	413f93fc
/IFFT_Tb/O5i	41000000	41000000
/IFFT_Tb/O6i	40ab0800	40ab0800



/IFFT_Tb/O14r	c0fff000	c0fff000
/IFFT_Tb/O15r	c1000000	c1000000
/IFFT_Tb/O16r	c1000600	c1000600
/IFFT_Tb/O1i	00000000	00000000
/IFFT_Tb/O2i	4220dd80	4220dd80
/IFFT_Tb/O3i	419a8200	419a8200
/IFFT_Tb/O4i	413f93fc	413f93fc
/IFFT_Tb/O5i	41000000	41000000
/IFFT_Tb/O6i	40ab0800	40ab0800
/IFFT_Tb/O7i	40541000	40541000
/IFFT_Tb/O8i	3fcb4000	3fcb4000
/IFFT_Tb/O9i	00000000	00000000
/IFFT_Tb/O10i	bfcb8000	bfcb8000
/IFFT_Tb/O11i	c0541000	c0541000
/IFFT_Tb/O12i	c0ab1000	c0ab1000
/IFFT_Tb/O13i	c1000000	c1000000
/IFFT_Tb/O14i	c13f8c00	c13f8c00
/IFFT_Tb/O15i	c19a8200	c19a8200
/IFFT_Tb/O16i	c220da00	c220da00
/IFFT_Tb/MM0/In1r	001111111000000000	00111111100000000000000000000000
/IFFT_Tb/MM0/In2r	010000000000000000	01000000000000000000000000000000
/IFFT_Tb/MM0/In3r	010000000100000000	01000000010000000000000000000000
/IFFT_Tb/MM0/In4r	010000001000000000	01000000100000000000000000000000
/IFFT_Tb/MM0/In5r	010000001010000000	01000000101000000000000000000000
/IFFT_Tb/MM0/In6r	010000001100000000	01000000110000000000000000000000
/IFFT_Tb/MM0/In7r	010000001110000000	01000000111000000000000000000000

Fig.10. Simulation results of 16-point floating numbers

The technology schematic of the proposed work is as shown below in the Fig. 11.

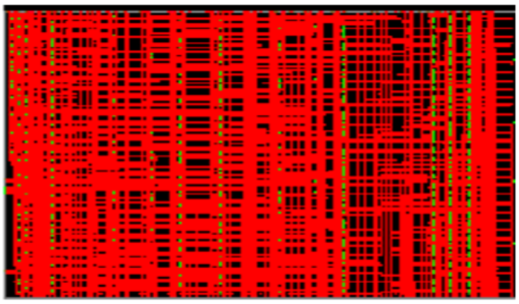


Fig.11. Technology Schematic of 16-point IFFT

V. CONCLUSION AND FUTURE SCOPE

This paper shows that the 16-point IFFT block for complex numbers as well as floating point numbers can be efficiently implemented with very good accuracy by making use of radix-2 butterfly architecture. The results have been extracted. The estimated area was found to be 61% and delay of 42.938ns. Since the design is made scalable and re-usable, the future scope of this work is to design for N-point IFFT and compare the computations for complex and floating point numbers, area and delay associated with it.

REFERENCES

[1] Dr.Uma B V, Harsha R Kamath, Mohith S, Sreekar V, Shravan Bhagirath, “Area and Time Optimized Realization of 16 Point FFT and IFFT Blocks by using IEEE 754 Single Precision Complex Floating Point Adder and Multiplier”, International Conference on Soft Computing Techniques and Implementations- (ICSCTI), Faridabad, India, Oct 8-10, 2015.

[2] Abdelmohsen Ali and Walaa Hamouda, “A multi-mode IFFT/FFT processor for IEEE 802.11ac: design and implementation”, Wireless Communications and Mobile Computing, 2015.

[3] Konguvel Elango, Kannan Muniandi, “Hardware Implementation of FFT/IFFT Algorithms incorporating Efficient Computation Elements”, Journal of Electrical Engineering & Technology, 2019.

[4] Anirban Ganguly, Anirban Chakraborty, Ayan Banerjee, “A Highly Accurate Current Mode Analog Implementation of Radix-2 FFT/IFFT Processor”, Eighth International Symposium on Embedded Computing and System Design (ISED), 2018.

[5] Yu-Wei Lin and Chen-Yi Lee, ”Design of an FFT/IFFT Processor for MIMO OFDM Systems”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 54, NO. 4, April 2007.

[6] Chu Yu, “A 128/512/1024/2048-Point Pipeline FFT/IFFT Architecture for Mobile WiMax”, IEEE 2nd Global Conference on Consumer Electronics (GCCE), 2013.

[7] Sabin Tandukar, Anupama Shree Dhamala, Wilson Lin, Steve C. Chiu, “Embedded Plant/Process Simulator: Design and System Integration for Scalable FFT/IFFT Computations”, IEEE, 2018.

[8] Avinash Kumar Singh, Ashutosh Nandi, “Design of Radix-2 Butterfly Structure using Vedic Multiplier and CLA on Xilinx”, IEEE Conference on Emerging Devices and Smart Systems (ICEDSS), 2017.

[9] Kasina Madhusudhana Rao **,**.Ravi Tejesvi ,G. AnanthaRao, "Verilog Implementation of 32 Point FFT Using Radix-2 Algorithm on FPGA Technology", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),(2014), pp. 40-43.

[10] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008, Aug. 29, 2008, pp. 1–58.