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:

\[ x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{2\pi n k}{N}}, 0 \leq n \leq N-1 \]  

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.

\[ \text{Number} = \text{significand} \times \text{base}^{\text{exponent}} \]  

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.](image)

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,

\[ C_r + jC_j = (A_r + jA_i) + (B_r + jB_i) \]
\[ D_r + jD_j = (W_r + jW_i)(A_r + jA_i) - (B_r + jB_i) \]

A_r, A_i, B_r, B_i 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.](image)

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](image)

![Fig.4. RTL Schematic of 16-point IFFT](image)
IV. RESULTS

The inputs given to the IFFT block is depicted in Fig. 5. The simulation results for the 16-point IFFT is shown in Fig. 6. The verification is done using Modelsim and the resultant values are shown in figures [7] and [8].

The Inverse FFT is carried out for floating point numbers too. The inputs given to the block is depicted in figures [9] and [10].
The technology schematic of the proposed work is as shown below in the Fig. 11.

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

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

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.

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.

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

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.

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.

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.

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