Simulation of Floating Point FFT Algorithm for Characteristic Frequency Calculation of Protein Family

Harshit Mathur, Vikas Pathak, Yamini Rathore
Department of Electronics and Communication Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan Jaipur-302017, (INDIA)
Email : vikaspathak85@gmail.com
Received 16.08.2018 received in revised form 02.01.2019, accepted 05.01.2019

***Abstract*-The objective of this paper is to implement the 256 point 32-bit floating point FFT algorithm for characteristic frequency calculation of protein family. This characteristics frequency can be further used for finding the location of hotspots in any particular protein of that family. Arithmetic adder/subtractor, and multiplier based on single precision IEEE-754 standards are designed. Simulation result of this FFT algorithm on Xilinx ISE tool is verified with MATLAB software and found similar result for calculation of characteristic frequency of protein family. Once the characteristic frequency for the particular biological function or interaction is determined, the individual "hot spot" amino acids are identified and possible to observe biological behavior of the protein. The VLSI architecture of 256-point FFT algorithm based on floating point data has been designed and synthesized using Xilinx ISE 14.4 tool for ARTIX-7 FPGA and simulated using Xilinx Isim simulator with the help of VHDL programming.**

Keywords **–** VHDL, FFT, DIT, Radix-2, IEEE-754, Genomics, Protein, Hot-spot Region, DFT

1. INTRODUCTION

The Fourier Transform converts time domain information into frequency domain information, with less hardware requirement and fast time utilization. Fast Fourier Transform (FFT) is the efficient algorithm for calculating the Discrete Fourier Transform (DFT). Radix-2 Cooley-Tukey algorithm, Raders algorithm and Radix-2 Butterfly Algorithm are different algorithms for calculating the FFT [1]. Radix-2 Cooley-Tukey algorithm consists of complex multiplier and complex adder. It is very efficient, easiest to implement and is widely used in practice. Design of high performance, low power and low area VLSI circuits are needed for DSP applications. The FFT can be designed by using radix-2 Butterfly algorithm [2]. The main advantage of implementation of Floating-point FFT algorithm includes faster speed, fast processing times, less clock periods and less chip area.

This paper concentrates on the development of the Fast Fourier Transform (FFT), based on Decimation-In-Time (DIT) domain, Radix-2 butterfly algorithm. The 256 Point 32 bit floating point FFT algorithm is designed using VHDL language [3]. The Fast Fourier Transform (FFT) can be computed by using IEEE-754 single precision and double precision floating point format and can be simulated using Xilinx Isim simulator tool [4]. The floating point arithmetic concept is the most used approach of approximating the real numbers for performing numerical calculations on computers. The advantages of using floating point approach are that it can support a much wider range of values. The implementation of the floating point arithmetic [5] using high level languages is more easy and convenient. The floating point operation has found many applications in the field where high precision and dynamic range is required. For implementing floating point arithmetic units, Field Programming Gate Array (FPGAs) have become the best option because of their high integration, high performance and low cost.

Proteins performs various functions in the body like forms the structural components (e.g. skin proteins), store and transport materials (e.g. haemoglobin), regulate the cell processes (e.g. hormones). There are 20 possible types of amino acids in proteins and they are connected with strong bond forming a chain of specific directionality. With the rapid increment in the protein databases, the determination of their relationships with defined functional families or the identification of the biological function of newly sequenced proteins becomes a real problem. Numerical series obtained is analyzed by digital signal analysis methods in order to extract information relevant to the biological function. Only one peak exists for a group of protein sequences sharing the same biological function.

Section-II gives the basic introduction of background of FFT algorithm, Floating point arithmetic and characteristics of protein family. Section-III describes the 256-point FFT algorithm and hardware implementation of floating point adders and multipliers. In section-IV various simulations and synthesis results of hardware and MATLAB simulation of 256 point FFT algorithm and their use in finding the characteristics frequency of protein family is discussed. Finally section-V concludes the paper.

2. BACKGROUND

2.1 Floating point arithmetic

The IEEE has established a standard for Floating point numbers. The IEEE -754 single precision Floating Point standard having 8 bit exponent (with a bias of 127), 23 bit mantissa and the IEEE-754 double precision standard having 11 bit exponent (with a bias of 1023), 52 bit mantissa. The Floating Point Numbers consists of three fields (1) Sign: it indicates sign of numbers (0 for positive and 1 for negative) (2) Exponent: it contains the value of base power (3) Mantissa: it sets the value of the number.

Sign	Exponent	Mantissa

In this paper Floating Point addition, subtraction and multiplication algorithms of IEEE-754 (single precision) format are used and which is designed using hardware programming language VHDL. Floating point adder performs both addition and subtraction which uses same hardware for floating point operation [6].

2.2 The FFT Algorithm

The Fast Fourier Transform (FFT) reduces the complexity of computing Discrete Fourier Transform (DFT) [7]. Image Filtering, signal analysis, sound filtering, multiplication of large integer are some of the application of FFT. FFT is based on the complex DFT.

The DFT of a sequence {x(n)} of length N is given by a complex valued sequence{X(k)}

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \quad 0 \le k \le N-1 \qquad (1)$$

Let W_N be the complex valued phase factor, which is an Nth root of unity expressed by

$$W_N = e^{-j2\pi/N} \qquad (2)$$

Hence X(k) becomes

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad 0 \le k \le N-1 \qquad (3)$$

The FFT can be designed by using radix-2 Butterfly algorithm as shown in fig. 1.

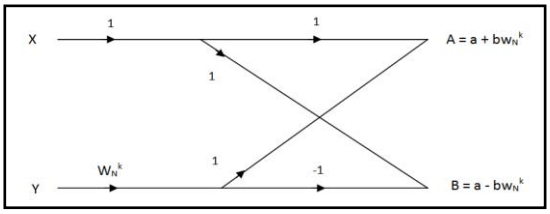

Figure 1: Butterfly Structure

By using Radix-2 Butterfly Algorithm 256 point DIT-FFT is designed [8].

2.3 Characteristic frequency of protein family

Protein is a bio-molecule which is the fundamental building blocks of all living organisms. The proteins are made up of 20 amino acids that are found in all living organism. Protein performs their biological functions through selective interaction with other molecule known as target. With the rapid expansion of the protein databases, the determination of their relationships with defined functional families becomes a real problem. Entire protein sequence can be represented by corresponding character sequence. In order to apply digital signal processing (DSP) techniques, the character sequences need to be converted into numerical sequences [9]. Electron ion interaction potential (EIIP) method is used to convert character sequence into numerical sequence according to table 1.

Table 1: EIIP Values for 20 Amino Acids

Amino Acid	EIIP	Amino Acid	EIIP
Leucine (Leu)	0.0000	Tyrosine (Tyr)	0.0516
Isoleucine (IIe)	0.0000	Tryptophan (Trp)	0.0548
Asparagine(Asn)	0.0036	Glutamine (Gln)	0.0761
Glycine (Gly)	0.0050	Methionine(Met)	0.0823
Valine(Val)	0.0057	Serine(Ser)	0.0829
Glutamine Acid(Glu)	0.0058	Cysteine(Cys)	0.0829
Proline (Pro)	0.0198	Threonine(Thr)	0.0941
Histidine(His)	0.0242	Phenylalanine (Phe)	0.0946
Lysine (Lys)	0.0371	Arginine(Arg)	0.0959
Alanine (Ala)	0.0373	Aspartic acid (Asp)	0.1263

In Protein-Target interaction, both the protein and the target share the same characteristic frequency having opposite phase which resembles resonance and characteristic frequency provides resonant recognition between protein and target. This model is called Resonant Recognition Model (RRM) [6]. Only one peak exits for the group of protein sequences sharing same biological functions.

$$|Y(k)|^2 = |Y_1(k)|^2 * |Y_2(k)|^2 \qquad (4)$$

Characteristic frequency for a particular biological function is being calculated to identify individual "hot-spot" and thus possible to observe biological behaviour of the Protein.

3. HARDWARE IMPLEMENTATION OF 32-BIT SINGLE PRECISION FLOATING POINT ARITHMATIC AND 256-POINT FFT ALGORITHM

3.1 Floating point adder/subtractor

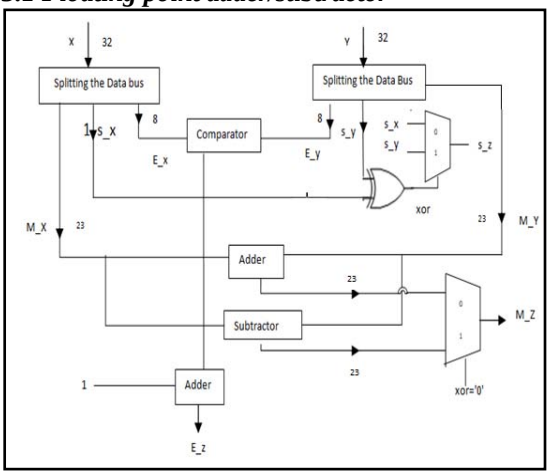

Figure 2: Block diagram of proposed floating point adder

Two inputs X and Y of 32 bit is splitted into sign (S_X, S_Y) of 1 bit, exponent (E_X, E_Y) of 8 bit and mantissa (M_X, M_Y) of 23 bit as shown in Fig. (2). E_X and E_Y is compared in comparator and difference of E_Y and E_X is taken as 'd'. Then, d and '1' is added in 23-bit binary adder. The exclusive oring of S_X and S_Y is done and output obtained is taken as xor. Xor is taken as a select line for multiplexer, input to the multiplexer is S_X and S_Y and output is S_Z. When xor = '0', then S_Z = S_X else S_Z = S_Y. M_X and M_Y of 23 bit is applied to binary subtractor. Similarly, M_X and M_Y are applied to floating point subtractor. Output of floating point adder and subtractor acts as the input of multiplexer, xor as a select line and output obtained is stored in M_Z. Now, concatenate S_Z, E_Z, M_Z which is the result of floating point adder/subtractor.

3.2 *Floating point multiplier*

Two inputs A and B of 32 bit are splitted into sign (S_A, S_B) of 1 bit, exponent (E_A, E_B) of 8 bit and mantissa (M_A, M_B) of 23 bit as described in Fig. (3). Exclusive ORing of S_A and S_B is performed and result obtained is taken as S_Z. E_A and E_B is added in 8 bit parallel adder and then bias (127) is subtracted from it. Now result is stored as E_i which is supplied as an input to demultiplexer and IP (47) as select line. The output of demultiplexer is taken as E_i + 1 when IP (47) = '1' else E_i is E_Z. M_A and M_B is applied to binary multiplier and output obtained is of 47 bits and stored in IP. If IP (47) = '1' then M_Z = IP (24 to 46) else M_Z = IP (23 to 45). Concatenate S_Z, E_Z, M_Z and this concatenated result is the result of floating point multiplier.

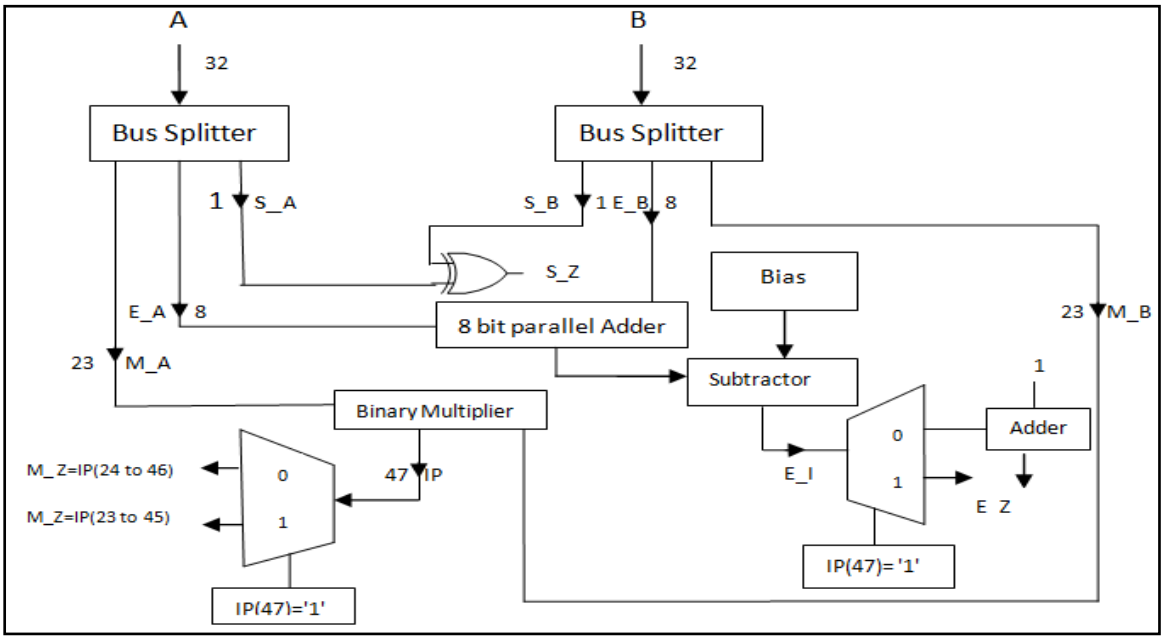

Figure 3: Block diagram of proposed floating point multiplier

3.3 VLSI Architecture of 256-Point FFT

For implementing 256-point DIT-FFT first of all VLSI architecture of 8-point FFT (as shown in fig. (4)) is designed using butterfly structure. Twiddle factor are multiplied by using floating point multiplier. Butterfly structure contains floating point adder and floating point subtractor.

For implementation of N point FFT, two N/2 point FFT are used. In fig. (5) inputs (x(0) to x(N-1)) are divided into even part (x(0), x(2), x(4)……x(N-2)) and odd part (x(1), x(3), x(5)…….x(N-1)). Output of even N/2 point FFT is G(0), G(1)……G(N/2-1) and output of odd N/2 point FFT is H(0), H(1)…….H(N/2-1). The output of odd N/2 point FFT is multiplied with twiddle factor by using floating point multiplier. The output obtained of N point FFT is X(0), X(1), X(2)……X(N-1).

3.4 Methodology

(a) Protein sequence input applied to MATLAB 2014a version is taken from database. Data is available publicly on the web, one of which is the retrieval system and Entrez search of the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH). The NIH nucleotide sequence database is called GenBank and contains all publicly available DNA sequences and protein sequences.

(b) Now this protein sequence which is obtained from database that is in character sequence (character sequence of amino acid) is converted into numerical values (real values). The conversion is done using Electron ion interaction potential (EIIP) method. EIIP is unique number used to represent each amino acid or nucleotide, irrespective of its position in a sequence. Numerical series obtained is then analyzed by digital signal analysis methods in order to extract information relevant to the biological function.

(c) In MATLAB, numerical values obtained from EIIP method is then converted into floating point numbers. This floating point number is given as an input to FFT block through text file which is created by MATLAB. To perform the floating point operations in Xilinx ISE 14.4, the implementation of floating point arithmetic is done. General purpose arithmetic adder, subtractor and multiplier based on 256 point single precision IEEE-754 standards are designed.

(d) FFT block read the text file created by MATLAB using VHDL Test Bench. Test bench is an environment used to verify the correctness of a design. A test bench is non-synthesizable VHDL file which applies a sequence of controlled inputs to a circuit and compares its output against the expected output. If there is any mismatch, an error is displayed in the VHDL simulator's log.

(e) FFT block is implemented by using 256 point Decimation in Time-Fast Fourier Transform (DIT-FFT) algorithm. The FFT can be designed by using radix-2 Butterfly structures. The VHDL code was synthesized using Xilinx ISE design suite 14.4 tools and simulated using Isim simulator.

(f) The text file is again created by Xilinx with the help of test bench so that MATLAB can read the output of Xilinx. The text file write operation takes place. Interfacing between Xilinx and MATLAB is done through test bench.

(g) Now MATLAB converts the floating point data which is obtained from FFT block is converted into real value. The results of implementation of FFT algorithm is verified with MATLAB using chirp signal. Result of Xilinx and MATLAB is verified for calculation of characteristic frequency of protein family.

(h) Plot of characteristic frequency is obtained in MATLAB. . Only one peak exists for a group of protein sequences sharing the same biological function. It was shown that proteins and their interacting targets display the same characteristic frequency in their interactions. Once the characteristic frequency of any protein family for the particular biological function or interaction is determined, the individual "hot spot" amino acids are identified that contributed most to this specific characteristic frequency and thus, possible to observe biological behavior of the protein.

4. RESULT AND DISCUSSION

4.1. Floating point adder

Table II shows the synthesis report of hardware units used in floating point adder.

Table 2: Device Utilization Summary of Adder

Logic Utilization	Used	Available	Utilization
Number of Slice Register	49	126800	0%
Number of slices LUTs	370	63400	0%
Number of fully used LUT-FF Pairs	49	370	13%
Number of bonded IOBs	96	210	45%

The simulation result of floating point adder shown in fig.(7) the addition and subtraction of two floating point number which is of 32 bit and represented by x and y and the result of addition is stored in z which is also of 32 bit. Others are intermediate signal used to simplify the code. Floating point adder performs both addition and

subtraction which uses same hardware for floating point operation.

Table 3 shows the utilization of devices or logic which are embedded in the FPGA for floating point Multiplier.

4.2 Floating point multiplier

Table 3: Device Utilization Summary of Multiplier

Logic Utilization	Used	Available	Utilization
Number of slices LUTs	64	63400	0%
Number of fully used LUT-FF Pairs	0	64	0%
Number of bonded IOBs	96	210	45%

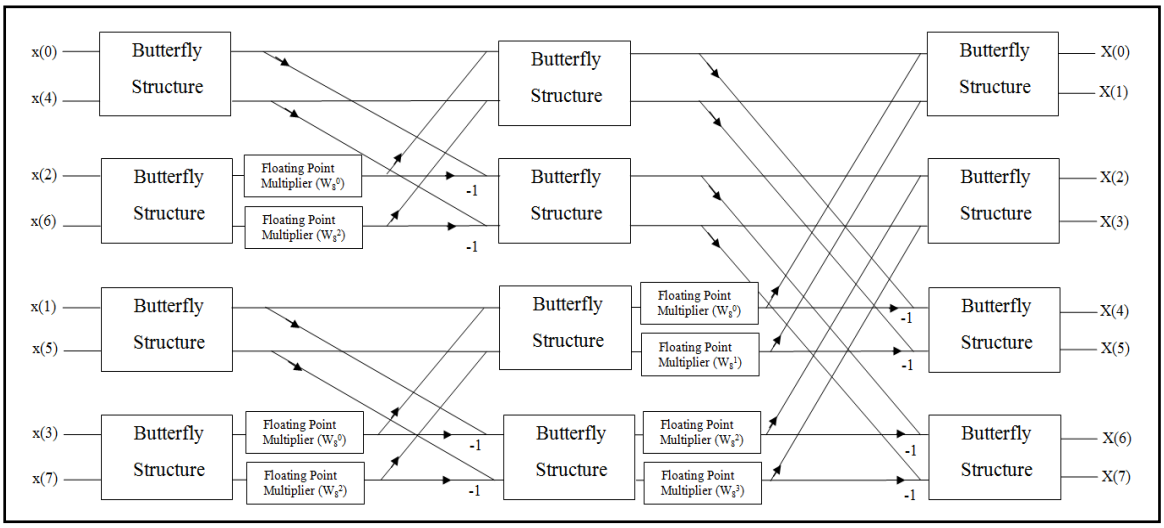

Figure 4: Block diagram of 8 point DIT-FFT

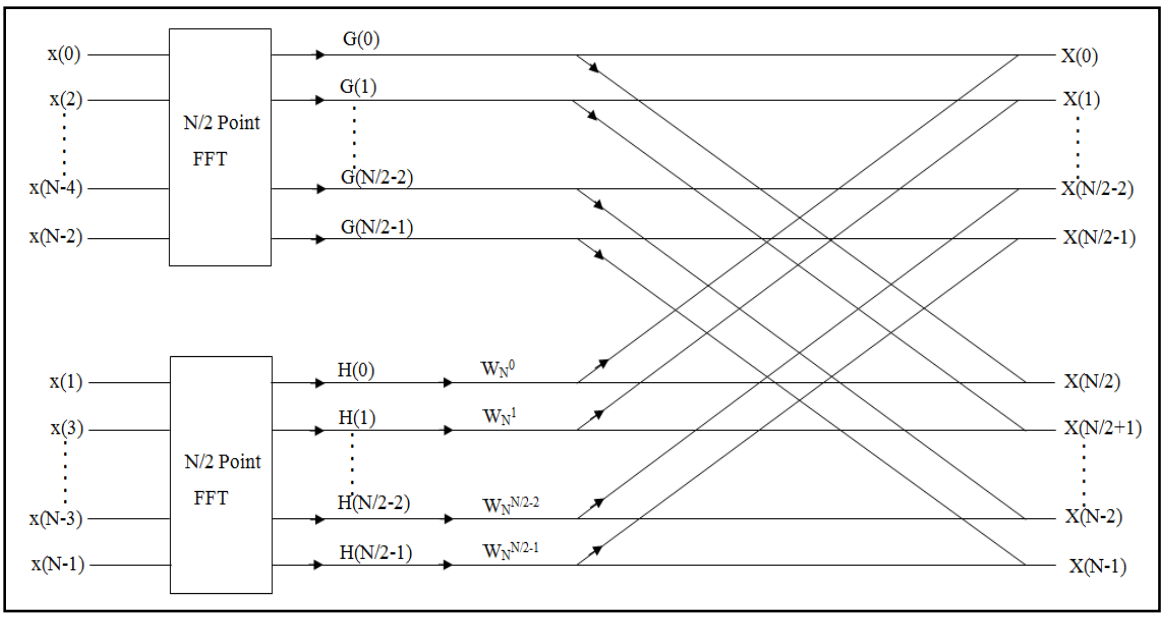

Figure 5: Block diagram for implementing N point FFT from N/2 point FFT

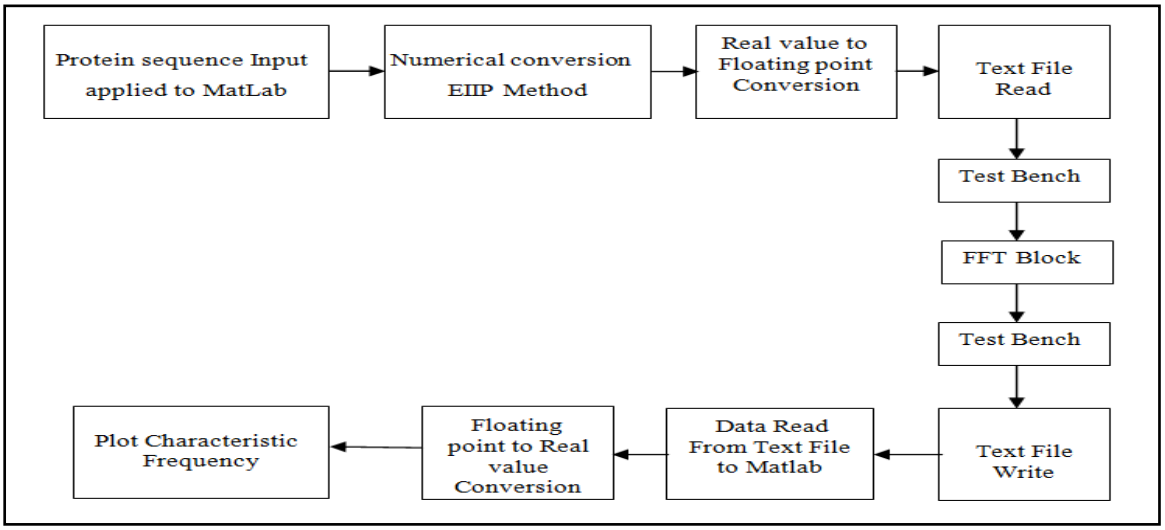

Figure 6: Methodology Flow Chart

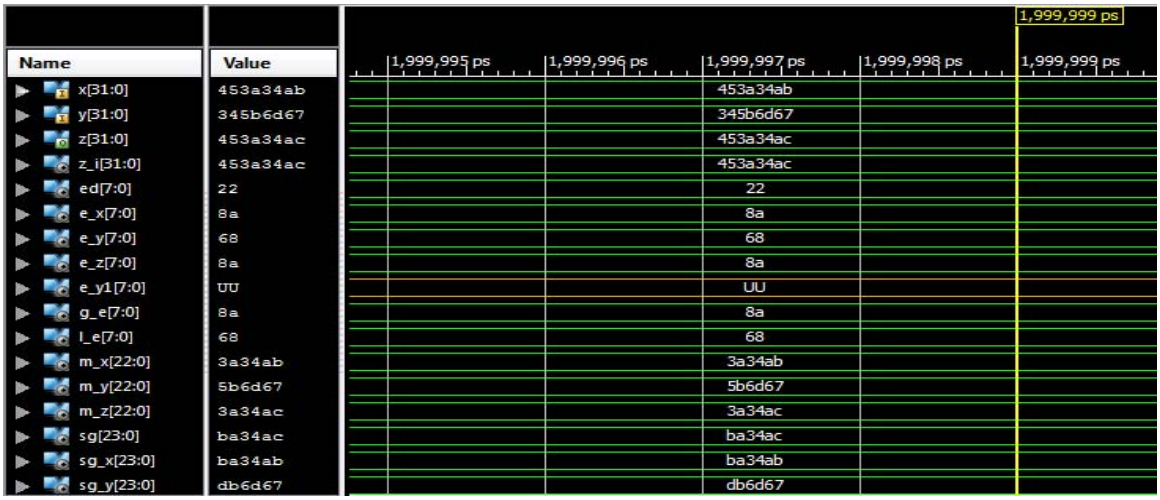

Figure 7: Simulation Result of Floating Point Adder

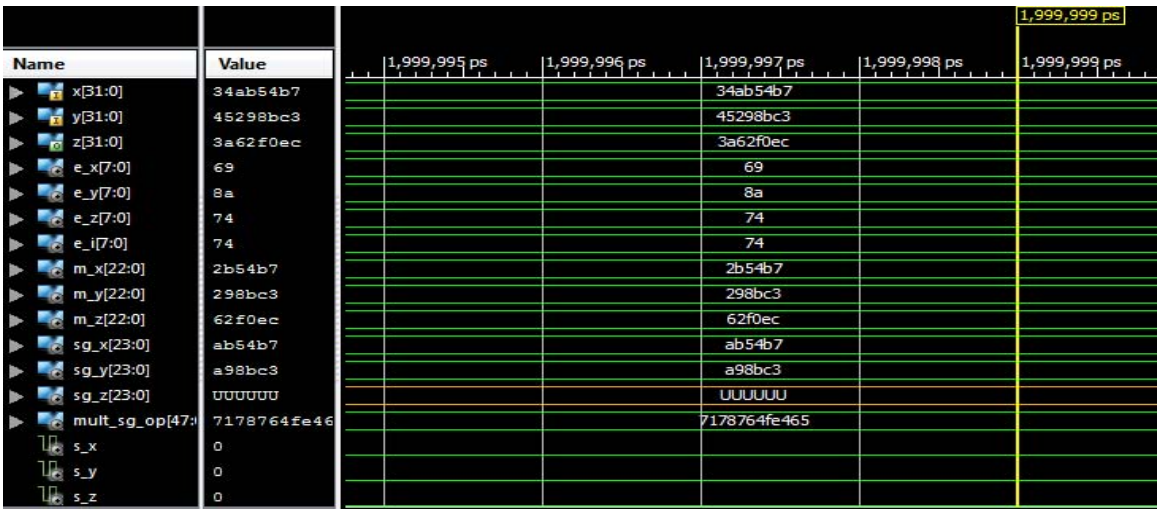

Figure 8 : Simulation Result of Floating Point Multiplier

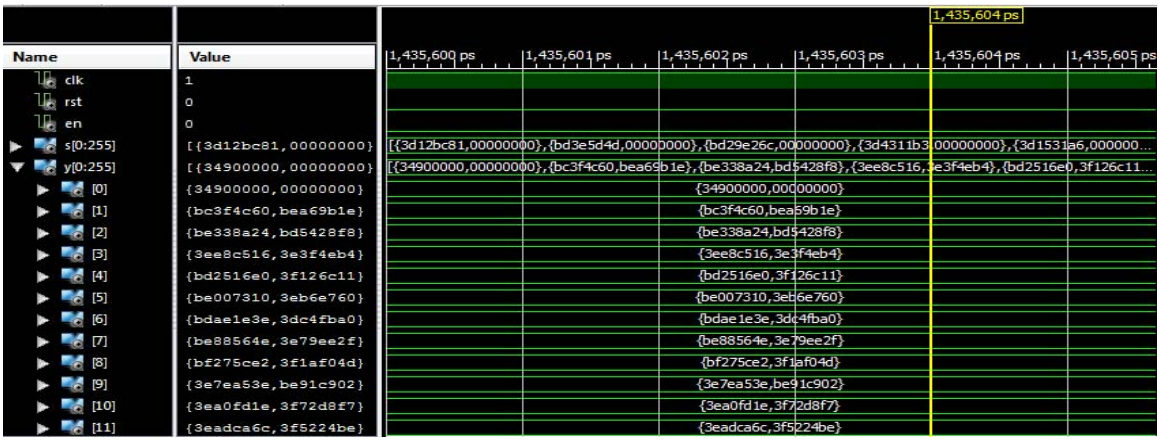

Figure 9: Simulation Result of 256 point FFT

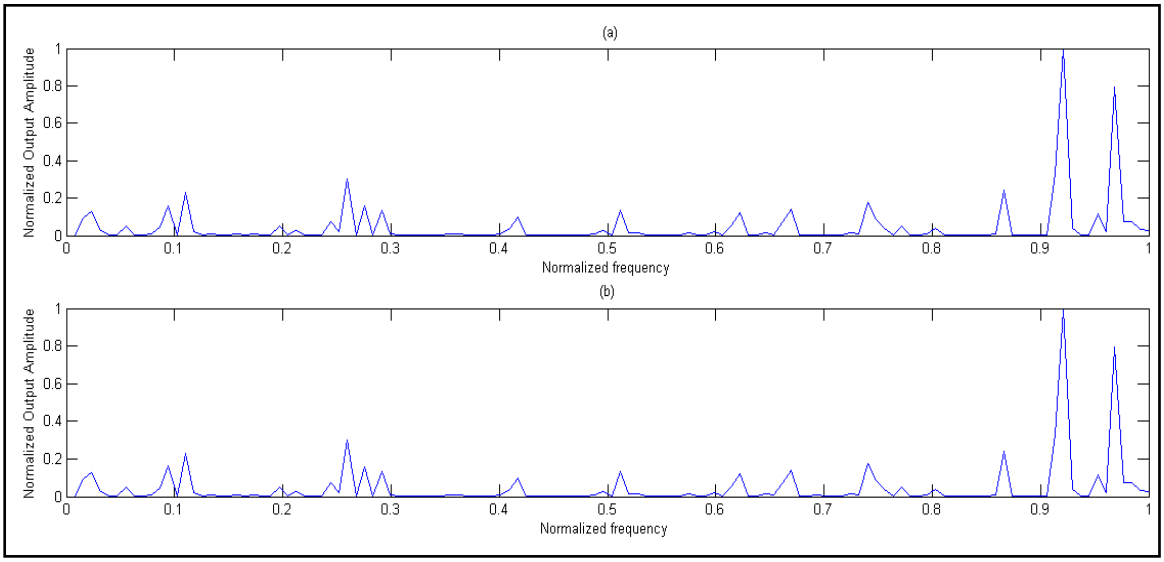

Figure 10: Application of 256-point FFT for Characteristic Frequency calculation of Protein family

he simulation result of floating point Multiplier shown in fig. (8) the multiplication of two floating point number which is of 32 bit and represented by x and y and the result of addition is stored in z which is also of 32 bit. Others are intermediate signals.

4.3. Simulation of 256 Point DIT-FFT

The simulation results in fig. (9) have 256 floating point input signals which has two parts namely real part and imaginary part. And the output obtained is 256 floating point values which also has real and imaginary parts.

4.4 Application of 256-point FFT for Characteristic

Frequency calculation of Protein family

Fig. 10(b) is the result of protein sequence i.e. Basic fibroblast growth factor (FGF) in Xilinx tool which is verified with the result of MATLAB tool fig. 10(a). These figure also clearly shows the same characteristics frequency which is 0.904 (normalized frequency) is achieved for FGF protein family in Xillinx and MATLAB tool. Once the characteristic frequency of any protein family for the specific biological function is obtained, then individual "hot spot" for amino acids are identified. Thus, it is possible to observe the biological activities of the protein which provides a global perspective in drug design.

5. CONCLUSION

The VLSI architecture of 256-point FFT Algorithm for 32-bit single precision floating point data is proposed in this paper. The implemented

design is synthesized using Xilinx ISE 14.4 for FPGA Artix7 family and simulated using Xilinx Isim simulator. The proposed design is verified using chirp signal of 3 frequencies based on 256Hz sampling frequency and found similar result as compare to MATLAB. The proposed VLSI architecture is also used for the application of characteristic frequency calculation of FGF protein family for hotspot region identification. Here also the same results are obtained for hardware and software implementation.

REFERENCES

[1] U Akshata. Gopika. D. K and Hameem Shanavas. I, "Hardware Implementation of Decimation in Time FFT", MIT International Journal of Electronics and Communication Engineering, (2013), 3, 39–42.

[2] Douglas Lyon, "The Discrete Fourier Transform, Part 2: Radix 2 FFT", Journal of Object Technology, (2009), 8, 21-33.

[3] J. Bhasker, "VHDL Primer" 3rd ed., Pearson Education (Singapore) Ltd., Indian Branch, Delhi 110092

[4] Remya Ramachandran, Vanmathi. K, "Simulation of radix-2 fast fourier transform using Xilinx", International Journal of Computer Science Engineering, (2014), 3, 125-130.

[5] Dhiraj Sangwan, Mahesh K. Yadav, "Design and Implementation of Adder/Subtractor and Multiplication Units for Floating-Point Arithmetic", International Journal of Electronics Engineering, (2010), 2, 197-203.

[6] Preethi Sudha Gollamudi, M. Kamaraju, "Design of High Performance IEEE- 754 Single Precision (32 bit) Floating Point Adder Using VHDL", International Journal of Engineering Research & Technology, (2013), 2(7), 2264-2275.

[7] Suciu. R, Reusens. P, Pollet. T, Cendrillon. R, "Floating point implementation of an FFT block–optimization issues", Katholieke University Leuven, (2003), 7, 1134-1138.

[8] Ranganadh Narayanam, Artyom M. Grigoryan and Bindu Tushara D, "Implementation of FFT by using MATLAB: SIMULINK on Xilinx Virtex-4 FPGAs: Performance of a Paired Transform Based FFT", International Journal of Advanced Computer Research, (2013), 3.2, 72-78.

[9] Yanay Ofranand and Burkhard Rost, "Protein–Protein Interaction Hotspots Carved into Sequences", Spanish National Cancer Research Centre, Spain, (2007), 3,119-123