Review on Design of Floating Point FFT Processor using VHDL

Roshan Pahune
MTech Student
Dept. of Electronics Engg
PCE, Nagpur

A. P. Rathkanthiwar, PhD
Associate Professor
Dept. of Elect & Tele. Engg
PCE, Nagpur

ABSTRACT

The design approach of FFT algorithm for floating point numbers is investigated in this paper. Using Fast Fourier Transform (FFT), the Discrete Fourier Transform (DFT) can be implement very fast. The FFT can be design by radix-2 butterfly algorithm using Decimatiom in Time (DIT) or Decimation in Frequency (DIF) methods . Using IEEE-754 Single precision floating point and Double precision floating-point format the Fast Fourier Transform (FFT) for floating point numbers can be easily computed and simulated using VHDL tools. The floating point number can support wide range of values. It can be represented using three fields sign, exponent and mantissa. The floating point Single precision format is always 32 bit and floating point Double precision format is always 64 bit. In this paper floating point addition, subtraction and multiplication algorithms is used. The IEEE-754 converter is used to convert decimal floating point number into Binary floating point format and it is also useful to verify the result. The floating point FFT processor reduce complexity of computation ,area, delay and power consumption.

Keywords

Floating Point Number, FFT, DIT, Radix-2, VHDL.

1. INTRODUCTION

This paper review and proposes a methodology to design floating point Fast Fourier Transform (FFT) processor using VHDL. This Design follows 32 bit single precision IEEE 754 standards.

Floating point number can be represented using three fields the sign, exponent and mantissa. The memory requirement and power consumption is more for Floating point algorithms.

1.1 Floating Point Number Representation

The IEEE-754 Single precision and Double precision format is used to represents the floating point numbers. The floating point numbers can support the wide range of values. The IEEE-754 floating point forrmat have three basic field: sign, exponent, and mantissa

a) Sign

The sign bit is one bit field. If this bit is 1 then it denotes positive number and if it is 0 then it denotes a negative number.

b) Exponent

The exponent is 8 bit field for IEEE -754 single precision format and 11 bit for IEEE double precision format. For single precision format the bias value of exponent is 127 and double precision format the bias value of exponent is 1023.

c) Mantissa

The mantissa is 23 bit field for IEEE-754 single precision and 52 bit for IEEE double precision format. Mantissa is also known as significant.

Table 1. IEEE-754 Single Precision Format

1	8	23
S	E	M

Table 2. IEEE-754 Double Precision Format

1	11	52
S	E	M

1.2 Fast Fourier Transform

To compute the Discrete Fourier Transform the Fast Fourier Transform is widely used in the field of digital signal processing (DSP) such as filtering, spectral analysis etc. For various applications such as image processing speech, audio, radar and biomedical signal processing.the FFT is widely used. The general formula for FFT is

X(k)=$\sum_{n=0}^{N-1} x(n)\, W_N^{kn}$ for k=0,1,..N-1

In this paper the input is taken as Floating point number. The FFT can be compute using two methods : Decimation in Time and Decimation in Frequency . In DIT-FFT methods we take the data from bit reversal order to normal order and in DIF-FFT we take its converse. The computation of FFT using DIT methods is easy as compare to DIF methods. By using the radix-2 or radix-4 algorithm we can design FFT Processor. In this paper we used the radix-2 algorithm. The radix-2 algorithm is divide the N-point FFT into smaller ones until two point FFT is obtained. This algorithm divide the N- point FFT into two point FFT therefore it is called as a radix-2 algorithm.

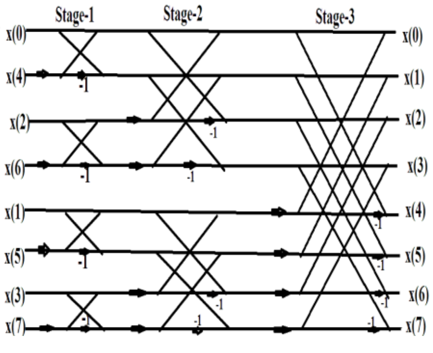

Fig 1: Butterfly Diagram of 8 Point DIT FFT

The FFT processor have very fast execution speed and low power consumption when it is used in wireless communication systems. The floating point FFT is a time consuming and it consumes a larger amount of area and power. When we design floating point 8 point FFT Then it increases complexity.

2. RELATED WORK

“The Implementation of Double Precision Floating Point radix-2 FFT using VHDL”. In this paper the 2 point FFT is design by using radix-2 butterfly algorithm. Using floating point numbers the computation time ,data storge, area, delay and power requirement is less. In this paper the floating-point addition, subtraction and multiplication algorithm is used. Complexity is decrease by using floating point numbers . The 2 point floating point FFT Processor are simulated by VHDL tools using Xilinx ISE software. . This paper explains the design of basic two point butterfly FFT processor. Due to use of floating point numbers the area, delay and power is reduced. In this paper DIT method is used .[1]

”FPGA Implementation of FFT Algorithm Using Floating Point Numbers”. This paper explain the design of 4 point Floating point FFT processor using floating point numbers. In this paper the floating point multiplier, adder and subtractor algorithm is used. The FFT is design using radix-2 algorithm and DIT methodsis used. In this paper 4 point DIT is used. The synthesis and simulation result is verified using Xilinx ISE 9.2 software. The FFT output result is dumped on FPGA kit to verifiy the simulation result. Requirement of Power is very low by using floating point numbers.[2]

“Simple Computation of DIT FFT.” In this paper Decimation in Time-Fast Fourier Transform (DIT-FFT) processor is design This DITFFT processor is 16 bit in length. This processor is designed by radix-2 algorithm for 8 point FFT. In this paper the input of FFT is taken in the form of real numbers. In this paper the Decimation in Time methods is used to compute FFT . There are 3 stage to compute 8 point DITFFT: stage-1 , stage-2 and stage-3. First stage-1 is calculated by giving simple input and output of this stage-1 is given to the input of stage-2 and finally output of this stage-2 is given to the input stage-3 and 8 point DITFFT is compute. In this paper the DITFFT processor is design by radix-2 algorithm and synthesis and simulation is done using VERILOG tools.[3]

“Designing and simulation of 32 Point FFT using Radix-2 algorithm for FPGA”. This paper shows the 32 bit DITFFT processor using radix-2 algorithm, In this paper Decimation in Time method is used to compute the FFT. The synthesis and simulation is done by Xilinx ISE 13.1 software using VERILOG tools. After verifiy the result radix-2 algorithm uses less no of computation and least no of slices. In this paper FPGA kit is used to verifiy the simulation result . The computation is very low by using this 32 bit DITFFT Processor [4]

“Fused Floating Point Add-Subtract an Multiply-Add Unit for FFT Implementation” This paper explain the design of fused floating-point arithmetic operations such as addition and subtraction. This algorithm is used to design Fast Fourier Transform (FFT). In this paper the input taken in the form of fused floating point numbers. The FFT is design by suitable VHDL tools and implemented using FPGA kit .The area ,delay and power is calculated .After implementation of simulation result on FPGA kit the fused primitives are very fast[5].

“Implementation of FFT algorithm using floating point numbers in WI-MAX communication system.” This paper shows the design of FFT processor using WIMAX communication. In this paper the Orthogonal Frequency-Division Multiple Access system is used.. To comupute FFT using wimax technology the OFDM modulator and demodulatoris is used .In this paper Twiddle factor is easy to compute . To implement the Twiddle factor , floating point single precision multiplier is used. The floting point mulplier increase computation speed and reduce time complexity.[6]

3. PROPOSED WORK

As per literature study, it is observed that the work done on Floating Point FFT is very less. In some papers the input of FFT takes integer value. Our methodology uses floating point value as an input to the FFT. In this paper work, we will design & simulate 32 bit 8 Point FFT architecture. Simulation and Synthesis of FFT architecture will be done using suitable VHDL tools.

The design process for this floating point 32 bit 8 Point DIT FFT Architecture is as follows:

1) Design and simulation of Floating point multiplier.
2) Design and simulation of Floating point adder & subtractor.
3) Design and Simulation of 8 Point Floating Point FFT Architecture Model.

i) Upper Node of Basic Floating Point 32 bit DITFFT

ii) Lower Node of Basic Floating Point 32 bit DITFFT

iii) Stage-1 of Floating Point 32 bit 8 Point DITFFT

iv) Stage-2 of Floating Point 32 bit 8 Point DITFFT

v) Stage-3 of Floating Point 32 bit 8 Point DITFFT

The Flow Chart of algorithms used for floating point multiplication and Floating Point addition are shown in figure 3 and figure 4 respectively.

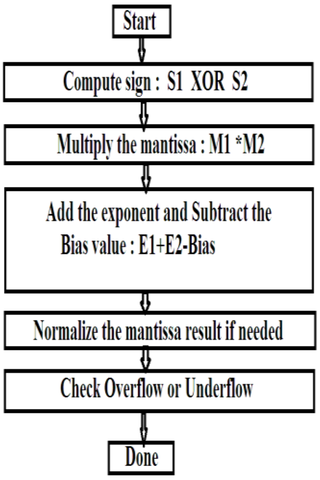
Start
Compute sign : S1 XOR S2
Multiply the mantissa : M1 *M2
Add the exponent and Subtract the Bias value : E1+E2-Bias
Normalize the mantissa result if needed
Check Overflow or Underflow
Done

Fig 2: Flow chart of Floating Point Multiplication Algorithm

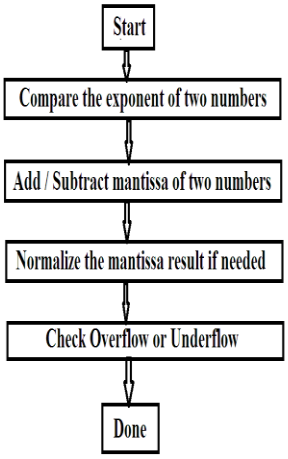
Start
Compare the exponent of two numbers
Add / Subtract mantissa of two numbers
Normalize the mantissa result if needed
Check Overflow or Underflow
Done

Fig 3: Flow chart of Floating Point Addition and Subtraction Algorithm

4. CONCLUSION

After studying various papers it is observed that work done on FFT is very less. In most paper the input of FFT taken is in the form of real number. In this paper we considered Floating Point Input. Each 32 bit input consisting of 16 bit real and 16 bit imaginary values. Various algorithms for floating point operations needed for FFT processor are presented. Methodology for implement the Floating Point FFT processor is also presented.

5. REFERENCES

[1] B. Tharanidevi and R. Jayaprakash "Implementation of double precision floating point radix-2 FFT using VHDL".International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 10, October 2013

[2] Hilal Kaptan1, Ali Tangel1 ,Suhap Sahin2,"FPGA Implementation of FFT Algorithm Using Floating Point Numbers" Kocaeli University, College of Engineering, Department of Electronics and Communication Engineering, Veziroglu Yerleskesi, 41040, Izmit, Turkey

[3] Neha V. Mahajan, Dr. J. S. Chitode " Simple Computation of DIT FFT" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 5, May 2014 ISSN: 2277 128X

[4] Afreen Fatima " Designing and Simulation of 32 Point FFT using Radix-2 algorithm for FPGA,"IOSR Journal of Elecrical and Electronics Engineering (IOSR0JEEE) e-ISSN:2320-3331, Volume9, Issue1 Ver,III (Jan 2014), PP 42-50.

[5] Prasanna Palsodkar, Ajay Gurjar "Fused Floating Point Add-Subtract an Multiply-Add Unit for FFT Implementation" 2014 2nd International Conference on Devices, Circuitand Systems (ICDCS).

[6] 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

[7] Chenlu Wu , 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

[8] 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

[9] Addanki Purna 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.

[10] Veera Kamat Dhakankar, Prof. Sonia Kuwelkar " Design and Implementation of Pipelined Floating Point Fast Fourier Transform Processor" International Journal for Innovative Research in Science & Technology| Volume 1 | Issue 11 | April 2015 ISSN (online): 2349-6010

[11] W. B. Ligon, S. McMillan, G. Mpnn, F. Stivers, and K. D Underwood "A Re-evelation of the Practicality of Floating Point Operations on FPGAs", Proceedings, IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 206- 215, Napa, CA, Apr. 1998. (ICANN'99).

[12] S. Şahin, A. Kavak., "Implementation of Floating Point Arithmetic Using an FPGA", Mathematical Methods in Engineering. Editors K.TAS, J.A.T.Machado and D.Baleanu Springer Book, 2007.

[13] Asmitha Haveliya, Amity University Lucknow, India "Design And Simulation Of 32-Point FFT Using Radix-2 Algorithm For FPGA Implementation" 2012 Second International Conference on Advanced Computing & Communication Technologies.

[14] H. Saleh and E.E. Swartzlander, Jr., "A Floating-Point Fused Add-Subtract Unit," Proc. IEEE Midwest Symp. Circuits and Systems (MWSCAS), pp. 519- 522, 2008.

[15] Sneha N.kherde and Meghana Hasamnis, "Efficient Design and Implementation of FFT," in International Journal of Engineering Science and Technology, 2011.

[16] D. Venkatesh Babu, K. Naresh Kumar "Verilog Implementation of Floating Point FFT With Reduced Addressing Logic" International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 9, September 2014

[17] Brian Hickmann, Andrew Krioukov, and Michael Schulte "A Parallel IEEE P754 Decimal Floating-Point Multiplier"

[18] I. S. Correa, L. C. Freitas, A. Klautau and J. C. W. A. Costa ,"VHDL Implementation of a Flexible and Synthesizable FFT Processor" IEEE Latin America Transcations, VOL. 10, NO. 1, JAN. 2012

[19] J. G. Proakis, Digital signal processing: principles, algorithms, and applications, Prentice-Hall Intemational, 1996.

[20] J.Bhaskar A VHDL PRIMER,Third Edition.