Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 5, Issue. 1, January 2016, pg.198 – 201

A Review Paper on Design and Simulation Analysis of 32 and 64 Point FFT Using Multiple Radix Algorithm

Miss. Priyanka R. Bhonde, Prof. R. D. Ghongade

[1] Electronic and Telecommunication from SGB Amravati University, India

[2] Electronic and Telecommunication from SGB Amravati University, India

[1] priyankabhonde69272@gmail.com; [2] rahulghongade@rediffmail.com

Abstract— Fast Fourier Transform is an algorithm used to calculate Discrete Fourier Transform (DFT) of a predetermined series. This Paper Proposes the performance and simulation of 32 and 64 point FFT using multiple RADIX Algorithms and it focus on Decimation-In-Time Domain (DIT) of the Fast Fourier Transform (FFT). At this point we use Xilinx Design Suite 13.2 Version by using VHDL.

Human needs with technical devices are increasing rapidly. In order to meet their requirements the system should be accurate and The Fast Fourier Transform (FFT) is one of the Basic operations in field of digital signal, image processing DVB-T2 etc. The Synthesis Results Shows the Comparison of 32and 64 Point FFT in terms of Speed and Computational Complexity. Here in the present system oriented approach DFT implementation takes place in a well manner followed by the well effective analysis of the system where area related to the time based strategy of the decimation plays a essential role in its implementation in well effective fashion respectively. There is a huge amount of the efficient analysis of the system associated to the strategy of the transformation of the fast Fourier environment plays a crucial role and the responsibility in favor of the effective implementation of the DFT in well respective fashion.

It can also be applicable for the processing of the images and there is a crucial in its analysis in terms of the pixel wise process takes place in the system in well effective manner respectively. There is a vast number of the applications oriented strategy takes place in the system in w ell effective manner in the system based implementation followed by the well efficient analysis point of view in well stipulated fashion of the transformation related to the fast Fourier strategy plays a crucial role and some of them includes analysis of the signal, Filtering of the sound and also the compression of the data equations of the partial differential strategy plays a major role and the responsibility in its implementation scenario in a well oriented fashion respectively.

Keywords—" Fast Fourier Transform (FFT)"," Decimation-In-Time (DIT-FFT)"," Discrete Fourier Transform (DFT)", "Radix-2, Radix-4", "Radix-"8, "VHDL"," Twiddle factor".

© 2016, IJCSMC All Rights Reserved

1. INTRODUCTION

The speed of the system is all about its internal peripherals. The peripherals are depending on designer preference. At the present it has been paying attention on radix-2 algorithm it have more delay. To cover this problem there is need to modify in the algorithm. This paper deal with the change in algorithm called as multiple radix algorithms and focuses on the design and simulation of 32 and 64 point DIT Fast Fourier Transform (DIT-FFT).

In field of digital signal, image processing the Fast Fourier Transform (FFT) is the Rudimentary operations. The Fast Fourier Transform (FFT) are important in the field of digital signal processing (DSP), widely used in communication systems , especially in orthogonal frequency division multiplexing (OFDM) systems and Field programmable gate array (FPGA). For the computation of Discrete Fourier Transform (DFT) Fast Fourier Transform (FFT) is used. This FFT and DFT are two different things. FFT is a specific family of algorithm which used to compute the DFT; and this paper concentrates on the design and simulation of 32 and 64 point FFT using multiple radix algorithms. In this paper the development of 32 point and 64 point FFT, based on Decimation-In- Time (DIT) or Decimation-In frequency domain using Radix algorithm is mathematical transform. A typical FFT processor is composed of butterfly calculation units, an address generator and memory units. FFT is all about decomposition and breaking the sequence into small sequence and combing them to get complete/total sequence. FFT is used to compute same result more quickly. To combine the results from the previous stage to form inputs to the next stage the Twiddle Factor coefficients are used. There are two way of FFT to compute the DFT.

1. Decimation in-time (DIT) FFT.
2. Decimation-in frequency (DIF) FFT.

For the memory-based processor design, minimizing the necessary memory size is effective for area reduction since the memory costs important part of the processor. On the Other hand, the FFT processor usually adopts on-chip static Random access memory (SRAM) instead of external memory. The main objective of this study is to calculate the area and time delay by comparing different algorithm. VHDL is used for coding simulation and synthesis is performed by using Model SIM ISE and Xilinx ISE Design Suite respectively.

2. LITERATURE SURVEY

In the year 1965 J. W. Cooley and J. W. Tukey [6] developed FFT algorithm to reduce the computations of the Discrete Fourier Transform (DFT) from N2 to multiplications and N(N-1) to N additions.

In 2000 Sudha Kiran G. Brundavani P gives idea about 256 bit DIT FFT using Radix-4. This project presents the new high speed FFT architecture based on radix-4 algorithm. The pipelined 256-bit, radix-4 DIT-FFT can be implemented easily by using both FPGA and standard cell technologies, such portability is offered by this algorithm. From the above paper results of radix-4 256 bit 16 point it is understandable that radix-4 having less delay in processing the input when compared with radix-2. Comparing with radix-2 algorithm, 75% of time is saved in radix-4 algorithm. As the delay time is reduced the fastness of the system is increased.

In 2007 Siva Kumar Palaniappan and Tun Zainal Azni Zulkifli represent 16 point FFT using Radix-4.
In the year 2008 Radix-4 floting point FFT processer is develop by H. Nimehr and B Phillips and C.C.Lin.
Until year 2010 FFT processer by by Chen Fong Hsaio Radix were applicable to mixed Radix algorithm. Again in same year Kaushik bhatt implement Radix-4 cordic processer by FPGA. In all cases required mare time so, To overcome this problem, K.Sreekanth Yadav, V.Charishma, Neelima koppala give a general access structure [6] in 2013. In which author develop 64 point FFT using Radix 4 Algorithm; Again K. Sowjanya and B. Leele Kumari 32 and 64 point FFT using radix-2 algorithm The Performance analysis can also be done between single Radix and Split-Radix FFT algorithms by using the parameter, Minimum delay.

3. Proposed Methodology

In this project, we are going to implement the multiple Radix architecture for 32 and 64 point FFT. Also to increase the throughput of system i.e. High speed design of multiple Radix FFT architecture for 32 and 64 point, we are going to implement the pipelined FFT concept.

3.1 Multiple Radix Algorithms

Multiple radix algorithm many radix algorithm such as radix-2, radix-4, radix-8 etc. Any radix algorithm is defined by their base i.e. if base is equals to 2 then it is known as radix-2, if base is equals to 4 then it is known as radix-4 and if base is equals to 8 then it is known as radix-8 algorithm. It represents by 2M where M represents the index/stage and its value is a positive integer. The base is decided the number of input and output to the system. The radix-2, radix-4, radix-8 have 2,4,8 input and output respectively.
The computation of radix made up of butterflies called Radix butterflies. E.g. The computation of radix-N made up of butterflies called Radix-N butterflies. The proposed system is based on multiple radix algorithms.

© 2016, IJCSMC All Rights Reserved

3.2 Radix-2

- The Radix-2 algorithm decompose a N-point DFT into four (N/2)-point DFT.
- Radix-2 algorithm requires Double stages as radix-4 requires.
- No. of stages requires in radix 2 are log2N.
- N/4 butterflies are used in each of (log2N)/2 stages, which is one quarter the number of butterflies in a radix-2 FFT.
- The radix-4 butterfly is consequently larger and more complicated than a radix-2 butterfly.

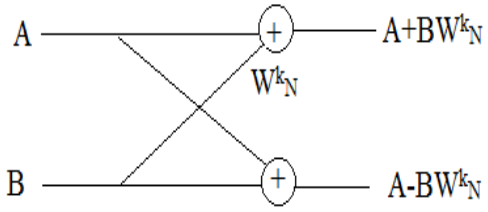

Fig1: Basic **Butterfly Diagram**

3.3 Radix-4

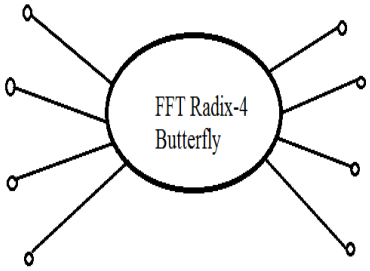

Fig. 2: Basic structure of R4 FFT

- (3N/8)· Log2N complex multiplication and (3N/2) · log2N complex addition. The Radix-4 algorithm decompose a N-point DFT into four (N/4)-point DFT .
- Radix-4 algorithm requires only half as many stages as radix-2 requires.
- No. of stages in radix 4 are log4N.
- N/4 butterflies are used in each of (log2N)/2 stages, which is one quarter the number of butterflies in a radix-2 FFT.
- The radix-4 butterfly is consequently larger and more complicated than a radix-2 butterfly.
- Radix-4 FFT is significantly faster than radix-2 FFT.
- Addressing of data and twiddle factors is more complex.
- Radix-4 FFT requires fewer calculations than a radix-2 FFT.
- A radix-4 FFT combines two stages of a radix-2 FFT into one, so that half as many stages are required.
- The overall number of operations is lower.

3.4 Radix-8

By using the FFT algorithm the computational complexity reduces to , where r represents the Radix-r FFT. The Radix-r FFT can easily derived from DFT by decomposing the N point DFT into a set of recursively related r-point transform and x(n) is powers of r. In Radix-8 algorithm the r is 8.The DIT Radix-8 FFT recursively partitions a DFT into eight quarter-length DFTs of groups of every eighth sample. The outputs of these shorter FFTs are reused to compute many outputs, which greatly decrease the total computational charge. The another type of Radix algorithm is Radix-8 which have advantages over Radix-2 and Radix-4.

© 2016, IJCSMC All Rights Reserved

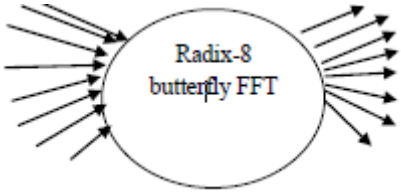

Fig. 2: Basic structure of R4 FFT

4. CONCLUSIONS

In this Paper, the design of 32 and 64 point FFT using Radix-2, Radix-4 and Radix-8, algorithms are performed, and the simulation with all the three algorithms are done using Minimum Delay (ns) as parameter and their synthesis and simulation results are shown by Xilinx synthesis tool on vertex .The test bench wave forms are displayed by using Xilinx ISE Design Suite 13.2. Further, the performance analysis can also be done by taking various parameters into reflection for different or same number of points.

ACKNOWLEDGEMENT

We thankful to incalculably our management for outspreading their support in providing us substructure and allowing us to use them in the successful completion of our research paper.

REFERENCES

[1] Asmita Haveliya, "Design and simulation of 32-point FFT using Radix-2 Algorithm for FPGA Implmentation",2012 second International conference on Advanced Computing and Communication Technologies.
[2] Alam V.Oppenhem, Ronald W.Schaler with John R.back , Discrete Time Signal Processing, second Edition.
[3] B.Parhami, *Computer Arithmetic, Algorithms and Hardware Designs*, 1999.
[4] James W.Cooley and John W.Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series.*
[5] Saad Bouguezel, M.Omair Ahmad, *"Improved Radix-4 and Radix-8 Algorithms"*, IEEE Department of Electrical and Computer Engineering Concordia University 1455 de Maisenneuve Blvd west Montreal, P.q.,Canada
[6] sneha N.Kherde, Meghana Hasamnis*, " Efficient Design and Implementation of FFT"*, International Journal of Engineering science and Technology(IJEST), ISSN:0975-5462 NCICT Special Issue Feb 2011.
[7] Ahmed Saeed,M.Elabably, G.abdelfadeel and M.I.Eladawy, *"Efficient FPGA implementation of FFT/IFFT Processor"*, international journal of circuits and signal processing,Issue3,Volume3,2009.
[8] Ali saidi, *"Decimation in Frequency FFT Algorithm",* Motorola Applied Research, Paging and Wireless Data Group Boynton Beach.
[9] Wei-Hsin chang and Truong Q.Nguyen fellow IEEE, *"On the Fixed point Accuracy Analysis of FFT Algorithms"*,IEEE Transactions ON Signal Processing, Vol. 56, No. 10, Oct 2008.
[10] Jesus Gracia, Juan A. Michell, Gustavo Ruiz, Angel M. Burón, Dept. de Electrónica y Computadores, Facultad de Ciencias, Univ. de Cantabria, Avda, *"FPGA realization of a Split Radix FFT processor"*, Los Castros s/n, 39005 Santander, SPAIN.
[11]HardwareDescriptionLanguage.URL:http://en.wikipedia.org/wiki/Hardware_description_language.
[12] Peter J. Ashenden, *The Designer's Guide to VHDL*, Second Edition.
[13] N. Weste, M. Bickerstaff, T. Arivoli, P.J. Ryan, J. W. Dalton, D.J.Skellern", and T.M. Percivalt "*A 50Mhz 16Point-FFT processor for WLAN applications*".
[14] Rizalafande Che Ismail and Razaidi Hussin *"High Performance Complex Number Multiplier Using Booth-Wallace Algorithm*" School of Microelectronic Engineering Kolej University Kejuruteraan Utara Malaysia.
[15] Bergland, G. D. "*A Guided Tour of the Fast Fourier Transform." IEEE Spectrum* **6**, 41-52, July 1969.
[16] *"Design of a radix-8/4/2 FFT processor for OFDM Systems",* Jungmin Park, Computer Engineering ,Iowa State University.

© 2016, IJCSMC All Rights Reserved