International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Email:editorijoer@gmail.com http://www.ijoer.in

Vol.4., Issue.3., 2016
(May-June)

RESEARCH ARTICLE

ISSN: 2321-7758

VLSI DESIGN OF POWER-DELAY EFFICIENT REDUCED MEMORY FFT PROCESSOR IN MOBILE COMMUNICATION

PRANOTEE V. KACHOLE[1],Prof.D.R.DANDEKAR[2]

[1]M.Tech(Electronics), [2]Professor

Department Of Electronics, B. D. College Of Engineering, Sewagram.

ABSTRACT

In the present work Fast Fourier transform (FFT) processing is one of the key procedures in popular orthogonal frequency division multiplexing (OFDM) communication systems. Structured pipeline architectures, low power consumption, high speed and reduced chip area are the main concerns in this VLSI implementation. This processor can be used in various OFDM-based communication systems, such as Worldwide Interoperability for Microwave access (Wi-Max), digital audio broadcasting (DAB), and digital video broadcasting-terrestrial (DVB-T). We adopt single-path delay feedback architecture. In our proposed work reduction of the size of FFT processor and power minimization is main concern. To reduce the size of FFT processor the repeated operation from radix-2 is minimized with the help of radix-8 .In our present work we are using DIT FFT for obtaining our desired purpose. Thus we consume the low power, highly accurate and efficient processor with reduced chip size. The OFDM based communication systems have high performance

Keywords: Mobile Communication, Reduced Memory, High Throughput, OFDM.

I. INTRODUCTION

The Fourier transform of a function of time itself is a complex-valued function of frequency, whose absolute value represents the amount of that frequency present in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency. The Fourier transform is called the frequency domain representation of the original signal. The term Fourier transform refers to both the frequency domain representation and the mathematical operation that associates the frequency domain representation to a function of time. It has been difficult to generate such a signal, and even harder to receive and demodulate the signal. Mobile communication need to transmits a large number of narrowband carriers, closely spaced in the frequency domain. In order to avoid a large number of modulators and filters at the transmitter and complementary filters and demodulators at the receiver, it is desirable to be able to use modern digital signal processing techniques, such as fast Fourier transform (FFT).

Discrete Fourier Transform (DFT) is a fundamental digital signal processing algorithm. DFT is the decomposition of a sampled signal in terms of sinusoidal (complex exponential) components. The symmetry and periodicity properties of the DFT are exploited to significantly lower its computational requirements [1]. The resulting algorithms are known as Fast Fourier Transforms (FFTs). The basis of the FFT is that a DFT can be divided into smaller

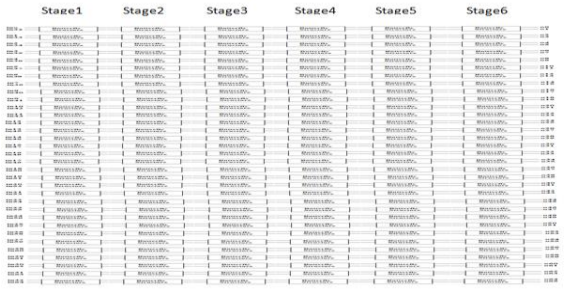
DFTs. In the FFT-64 processor a radix-8 FFT algorithm is used.

1. FFT PROCESSOR

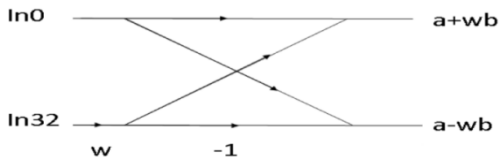
The conventional signal and image processing applications requires high computational power based on Fast Fourier Transform (FFT) in addition to the ability to choose the algorithm and architecture. FFT module is core part of OFDMA system as defined by IEEE Std 802.16-2005. The FFT operates by decomposing an *N* point time domain signal into *N* time domain signals each composed of a single point. The second step is to calculate the *N* frequency spectra corresponding to these *N* time domain signals. Lastly, the *N* spectra are synthesized into a single frequency spectrum. The design about variable point FFT processor is just based on FFT module in OFDMA system application.

2. RADIX-2 ALGORITHM

The Radix-2 algorithm consist of six stages.The structure for 64 point Radix-2 is shown below



The butterfly structure for 2 point DIT FFT is explained below



For (a+wb):-

$=(a_r+ja_i)+(w_r+jw_i)(b_r+jb_i)$

$= (a_r+w_rb_r-w_ib_i)+j(a_i+w_rb_i+w_ib_r)$ (1)

For (a-wb):-

$=(a_r+ja_i)-[(w_r+jw_i)(b_r+jb_i)]$

$=(a_r-w_rb_r+w_ib_i)+j(a_i-w_rb_i-w_ib_r)$ (2)

From equation 1 & 2 it is observed that four multiplier and four adder is required. since there are two nodes present in the butterfly structure therefore total eight multipler and eight adder will be needed.

The radix-2 FFT algorithm is obtained by using the divide-and-conquer approach split the output sequence X(k) into two summations, one of which involves the sum over the first 2/ *N* data points and the second sum involves the last 2/ *N* data points. Thus we obtain,

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk}$$

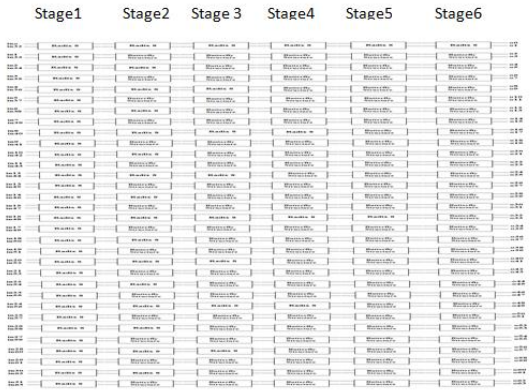
$$= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=0}^{(N/2)-1} x(n+\frac{N}{2}) W_N^{(n+N/2)k}$$

$$= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + W_N^{kN/2} \sum_{n=0}^{(N/2)-1} x(n+\frac{N}{2}) W_N^{nk}$$

3. RADIX-8 ALGORITHM

The Radix-8 algorithm also consist of six stages. In our proposed work it is found that some of the operation in Radix-2 is repeated therefore that repeated operation is optimized by using Radix-8.In our proposed work we have made Radix-8 by using Radix-2 algorithm.

The structure of 64 point Radix-8 is shown below



The butterfly structure for Radix-8 will be

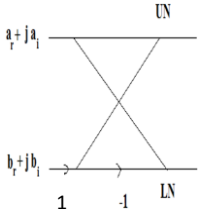


Fig.1

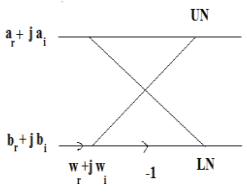


Fig.2

For One Node

$o/p=(a_r+ja_i)+(b_r+jb_i)(w_r+jw_i)$

$=(a_r + b_r w_r - b_i w_i) + j(a_i + b_r w_i + b_i w_r)$.....(1)

For stage 1 the value $w_r=1$ & $w_i=0$

$=(a_r+b_r)+j(a_i+b_i)$.....(2)

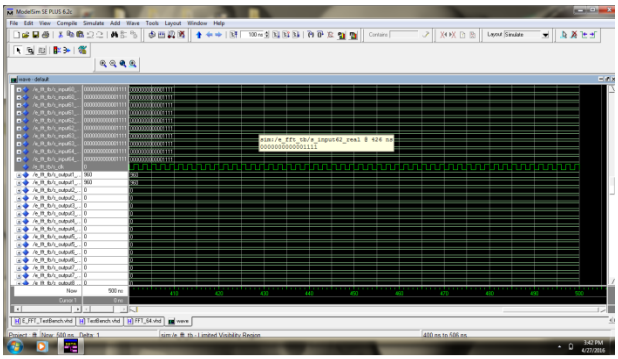
From equation 1 for both node i.e upper node and for the lower node eight multiplier and eight adder is required. So for all the six stages 1536 multiplier and 1536 adder is required. From equation 2 for both node only 4 adder is required. After optimization for all the six stages 1032 mutiplier and 1284 adder is required. Thus Radix-8 reduces the number of operations which result into area minimization.

4. IMPLEMENTATION

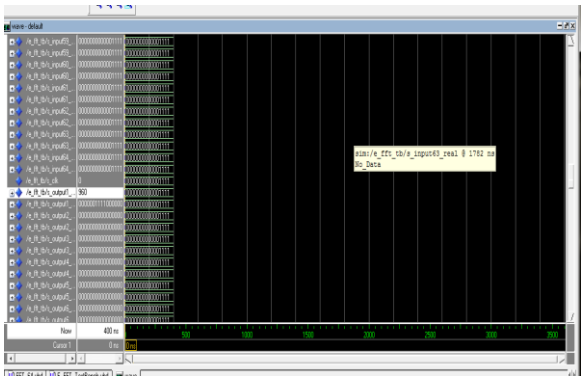
The code is written for 64 point Radix-2 DIT FFT processor and 64 point Radix-8 DIT FFT processor using VHDL. The IEEE Standard 1076 defines VHDL. It is a programming language that has been designed and optimized for describing the behavior of digital circuits and systems. By using HDL one can describe any hardware at any level. VHDL makes design reuse feel natural. The simulations are done using the ModelSim tool and Xilinx. Modelsim is a multilanguage HDL , for simulation of Hardware description language such as VHDL ,Verilog etc. ModelSim SE offers high-performance and advanced debugging capabilities. We have used 13.1 version of Xilinx for simulation result. ModelSim XE stands for Xilinx Edition, and is specially designed for integration with Xilinx ISE. Xilinx ISE[2] (Integrated Synthesis Environment)[3] is a software tool produced by Xilinx for synthesis and analysis of HDL designs.

5. SIMULATION AND RESULT

Let 15+15j is given as a input to the butterfly structure we obtain the output as 960+960j . It can be seen below in the following waveform for 64 point radix-2.



For the same input the simulation result for 64 point Radix-8 is obeserved below.



6ss4 point Radix-2 requires 1536 multiplier and 1536 adder whereas after optimization 64 point Radix-8 requires 1032 mutiplier and 1284 adder. Thus, due to optimization of some adder and multipler, minimization of area is possible. As power is directly proportional area therefore power efficiency is increased. Delay will remain as it is.

6. CONCLUSION

This paper shows the implementation of 64 point Radix-2 and 64 point Radix-8. The performance parameters like area have been calculated for both Radix-2 and Radix-8 and it has been clear from the simulation results that the Radix-8 algorithm based processor yields a higher performance as compared to an Radix-2 based processor. As it is found that power $\propto$ area [1] therefore as area reduces then power will also declines proportionally.

7. ACKNOLEDGEMENT

I would like to express my sincere gratitude to my advisor, Prof. D.R.Dandekar for their guidance and unreserved help during the study time. Without them the completion of this paper would have been impossible.

8. **REFERENCES**

[1]. Chu Yu, Yi-Ting Liao, Mao-Hsu Yen, Pao-Ann Hsiung, and Sao-JieChen, "A Novel Low-Power 64-point Pipelined FFT/IFFT Processor for OFDM Applications," in Proc. IEEE Int'l Conference on Consumer Electronics. Jan. 2011, pp. 452-453

[2]. Chia-Hsiang Yang, Tsung-Han Yu, StudentMember, and Dejan Marković,, "Power and Area Minimization of Reconfigurable FFT Processors: A 3GPP-LTE Example " IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 47, NO. 3, MARCH 2012.

[3]. M. Ayinala, M. Brown, K.K. Parhi, "Pipelined parallel FFT architectures via folding transformation", IEEE Transactions on VLSI Systems, pp. 1068-1081, vol.20, no. 6, June 2012.

[4]. Ren Chen, Neungsoo Park, Viktor K. Prasanna, "High Throughput Energy Efficient Parallel FFT Architecture on FPGAs" IEEE 2014, This work has been funded by DARPA under grant number HR0011-12-2- 0023.

[5]. J. Lee, H. Lee, S. I. Cho, and S. S. Choi, "A High-Speed two parallel radix-24 FFT/IFFT processor for MB-OFDM UWB systems," in Proc IEEE Int. Symp. Circuits Syst., 2006, pp. 4719–4722.

[6]. M. Garrido, K. K. Parhi, and J. Grajal, "A pipelined FFT architecture for real-valued signals," IEEE Trans. Circuits Syst. I, Reg Papers, vol. 56, no. 12, pp. 2634–2643, Dec. 2009.