International Journal of Advances in Engineering & Technology May 2012.

OPTIMIZED IMPLEMENTATION OF FFT PROCESSOR FOR OFDM SYSTEMS

ABSTRACT

Due to the advanced VLSI technology, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform(IFFT) has been applied to wide field of digital signal processing (DSP) applications. For modern communication systems, the FFT/IFFT is very important in the OFDM. Because of FFT/IFFT is the key computational block to execute the baseband multicarrier demodulation and modulation in the OFDM system. The Fast Fourier Transform (FFT) and its inverse transform (IFFT) processor are key components in OFDM systems. An optimized implementation of the 8-point FFT processor with radix-2 algorithm in R2MDC architecture is the processing element. This butterfly- Processing Element (PE) used in the 8-FFT processor reduces the multiplicative complexity by using a real constant multiplication in one method and eliminates the multiplicative complexity by using add and shift operations in this proposed method.

KEYWORDS: Cooley-Tukey, R2MDC, FFT, VHDL, OFDM.

I. INTRODUCTION

In the last few years, wireless communication plays a significant role in people’s life. The development of wireless communication science and technology was extremely fast due to VLSI technology. Many complicated design of communication system become feasible. There is a rapidly growing demand for high speed, efficient and reliable digital communication system with the purpose of support multimedia transmission such as high-quality voice, video, etc. Orthogonal Frequency Division Multiplexing (OFDM) technology is an effective modulation scheme to meet the demand. This is an advanced modulation technique that effectively expands channel utilization and reduces inter-symbol interference (ISI) and inter-carrier interference (ICI) caused by multi-path effect. Multiple-input multiple-output (MIMO) signal processing technique has been utilized in combination with OFDM for next generation wireless communication system to enhance link throughput as well as the robustness of transmission over frequency selective fading channel. FFT algorithms are based on the fundamental principle of decomposing the computation of the DFT of a sequence of length N into successively smaller DFT using the symmetry, periodicity, compressibility and expansibility properties of WN .The manner in which this principle is implemented leads to a variety of different algorithms, all with comparable improvements in computational speed. It can reduce the computational complexity from O (N2) to O (Nlog2N), and the regularity of the algorithm makes it suitable for VLSI implementation. Due to its advantage, it can be efficiently employed in the modern Orthogonal Frequency Division Multiplexing (OFDM) system. The OFDM signal is made up of many orthogonal characters, and each individual carrier is digitally modulated with a relatively slow symbol rate. This method has distinct advantages in multipath propagation because, in comparison with the single carrier method at the same transmission rate, more time is needed to transmit a symbol. The 256QAM modulation modes are used, and the modulation is adapted to the specific transmission requirements. Whereas 64-QAM modulation mode is used for Wi-Fi. Transmission rates of up to 75 Mbit/s are possible at 1GB speed. The 802.16e is a standard of Wi max in the frequency range of up to 6 GHz with the objective of allowing mobile applications and even roaming. In addition, the number of carriers can vary over a wide range depending on permutation zone and FFT base (128, 512, 1024, and 2048). Based on our development, not only a dedicated FFT/IFFT module can be easily prototyped for fast system verification, but also the resulting compiler can be used as a basis for more advanced research in the future. This work can be organized as: section (I) explains the concept of OFDM, FFT processor necessity.

Section (II)FFT algorithm with the help of radix point. Section (III) architecture of r2mdc and addshift method section (IV) simulation results section(V) synthesis report section (VI) conclusion.

II. CONVENTIONAL FFT ALGORITHM

Discrete Fourier Transform (DFT) is widely applied to the analysis, design, and the implementation of discrete-time signal processing algorithms and communications. The computational complexity of direct evaluation is rather high so that it is hard to meet high performance and low cost design goal. Therefore, a fast DFT algorithm is required. Since the early paper proposed by Cooley and Tukey in 1965 [3], they provided a lot of ways to reduce the computational complexity from O (N2) to O (N log 2 *N).* After that, the algorithms that can reduce the computational complexity are generally called fast Fourier transform (FFT) algorithms. The fast Fourier transform (FFT) algorithm, together with its many successful applications, represents one of the most important advancements in scientific and engineering computing in this century. The wide usage of computers has been instrumental in driving the study of the FFT, and a very large number of articles have been written about the algorithm over the past thirty years. FFT algorithms are based on the fundamental principle of decomposing the computation of the discrete Fourier transform of a sequence of length N into successively smaller discrete Fourier transforms. The regularity of the algorithms makes them suitable for VLSI implementation. Decomposing is an important role in the FFT algorithms. There are two decomposed types of the FFT algorithm. One is decimation-in-time (DIT), and the other is decimation-in-frequency (DIF) shown in fig 1. The difference between these two types is in the input 6 and output data ordering in signal flow graph (SFG). The DIT algorithm means that the time sequence is decomposed into small subsequence, and the DIF algorithm decomposes the frequency sequence. In addition, there is no difference in computational complexity between these two types of FFT algorithm. Since the low computational complexity of FFT algorithms is desired for high speed consideration in VLSI implementation, here we discuss the computational complexity of different algorithms.

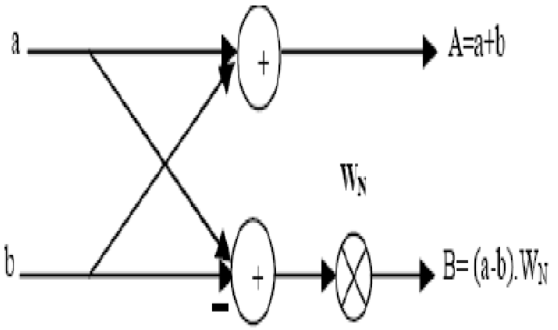


Fig.1.Signal flow graph of decimation in frequency of FFT

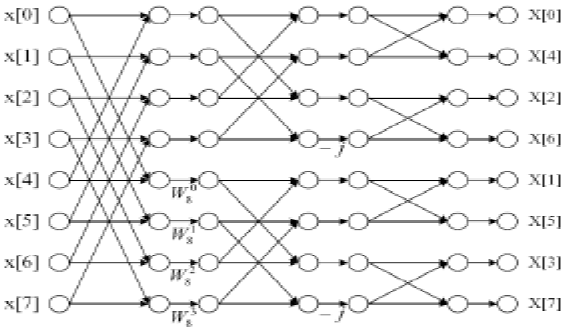


Fig.2.Basic Butterfly Computation

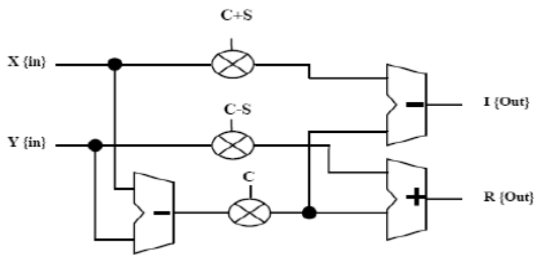


Fig.3.Implementation of Complex Multiplication

Basic Radix-2 butterfly processor shown in Fig, 2. consists of a complex adder and complex subtraction. Besides that, an additional complex multiplier for the twiddle factors WN is implemented. The complex multiplication with the twiddle factor requires four real multiplications and two add/subtract operations. Three twiddle factor values are used i.e. c, c+s, c-s.

III. PROPOSED ARCHITECTURE

A. R2MDC ARCHITECTURE

One of the most straightforward approaches for pipeline implementation of radix-2 FFT algorithm is Radix-2 Multi-path Delay Commutator (R2MDC) architecture. It's the simplest way to rearrange data for the FFT/IFFT algorithm. The input data sequence are broken into two parallel data stream flowing forward, with correct distance between data elements entering the butterfly scheduled by proper delays. 8-point FFT in R2MDC architecture is shown in Fig. 4.At each stage of this architecture half of the data flow is delayed via the memory (Reg) and processed with the second half data stream. The delay for each stage is 4, 2, and 1 respectively. The total number of delay elements is 4 + 2 + 2 + 1 +1 = 10. In this R2MDC architecture, both Butterflies (BF) and multipliers are idle half the time waiting for the new inputs. The 8-point FFT/IFFT processor has one multiplier, 3 of radix- 2 butterflies, 10 registers (R) (delay elements) and 2 switches (S).

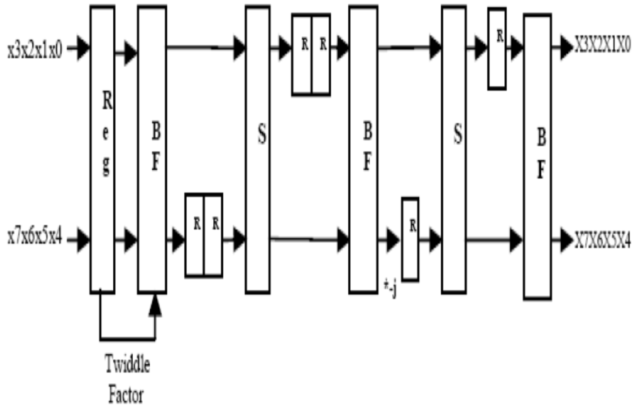


Fig.4.R2MDC ARCHITECTURE

B. ADD AND SHIFT METHOD

Another method proposed eliminates the non-trivial complex multiplication with the twiddle factors (W8 1, W8 3) and implements the processor without complex multiplication. The proposed butterfly processor performs the multiplication with the trivial factor W8 2=-j by switching from real to imaginary part and imaginary to real part, with the factor W8 0 by a simple cable. With the non-trivial factors W1,W3 the processor realize the multiplication by the factor 1/_2 using hard wired shift-and add operation as shown in Fig. 5.

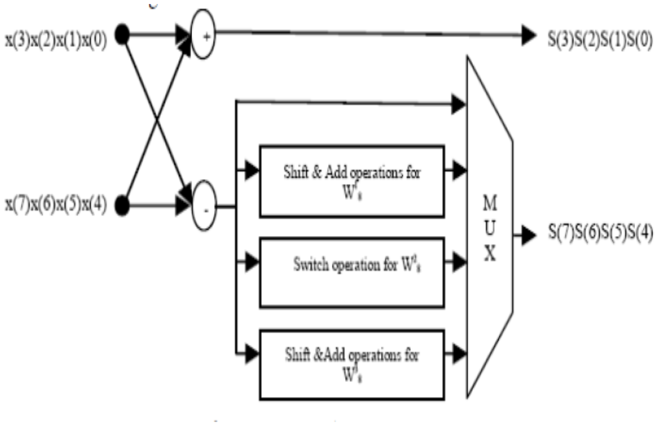


Fig.5. Butterfly processor with no complex multiplication

IV. ANALYSIS AND DESIGN OF FFT USING VHDL

A. Simulation methodology

VHDL is language for describing digital electronic systems. It arose out of the United States government’s very high speed integrated circuits (VHSIC) program, in the course of this program; it became clear that there was a need for a standard language for describing the structure and function of integrated circuits(IC’s). Hence the VHSIC hardware description language (VHDL) was developed. Modeling for simulation and synthesis is a vital part of a range of levels of abstraction, from gate levels up to algorithmic and architectural levels. It will continue to play an important role in the design of silicon-based systems for some time to come. Very high speed integrated circuit hardware description language (VHDL) can be used to model digital systems and introduce some of the basic concepts underlying the language. The ModelSim-Altera simulator compiles the testbench and the netlist (multiplier.vho), annotates the SDF data (in multiplier_vhd.sdo), and runs the simulation for the specified time. A waveform window within the ModelSim-Altera simulator is invoked that shows the expected and actual results of the multiplier. The expected and actual results are also checked in the test bench, and messages that show whether or not the results match are displayed in the simulator’s console window. The data output of the multiplier module changes with a delay after the clock edge because the SDF data is annotated in the gate-level timing simulation.

B. Simulation result

The simulation results are given below.

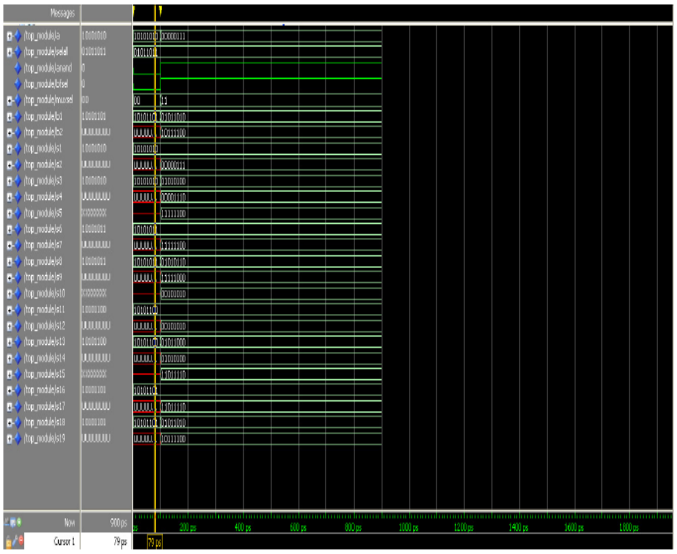


Fig.6. Simulation result of R2MDC architecture

This simulation shows the radix-2 operation. Two inputs are selected with the help of demultiplexer which is present inside the buffer block. Addition and subtraction of two inputs with a twiddle factor multiplication. It produces final fft output. The VHDL Program is compiled in Modelsim v6.4a to generate the butterfly processor operation with the help of R2MDC architecture, Add and shift method to know the design functionality and for power, area analysis Xilinx is used.

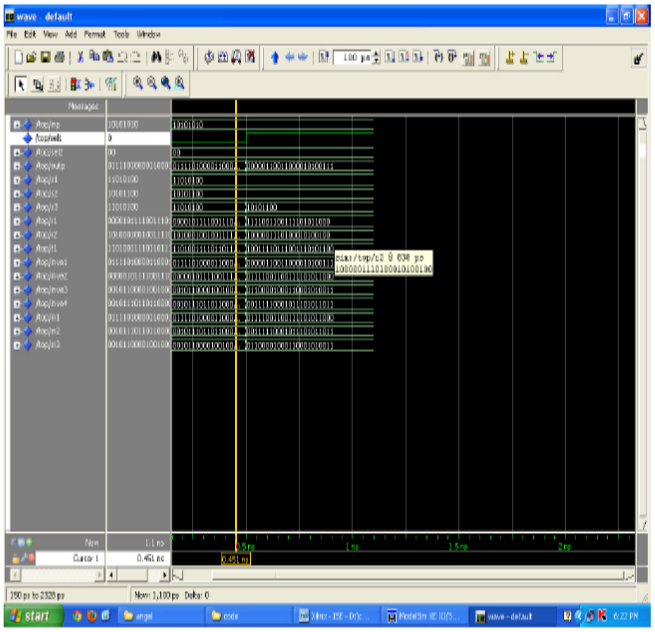
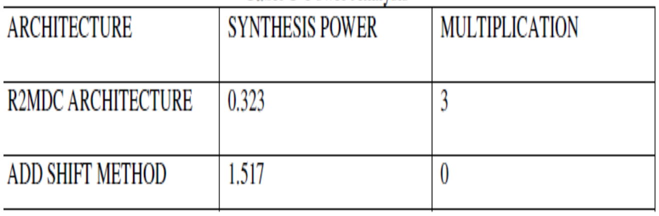


Fig.7. Simulation result of add and shift method

This simulation shows that the input after addition and subtraction of radix-2 process multiplied with a real number of complex variable. It produces the same fft output.

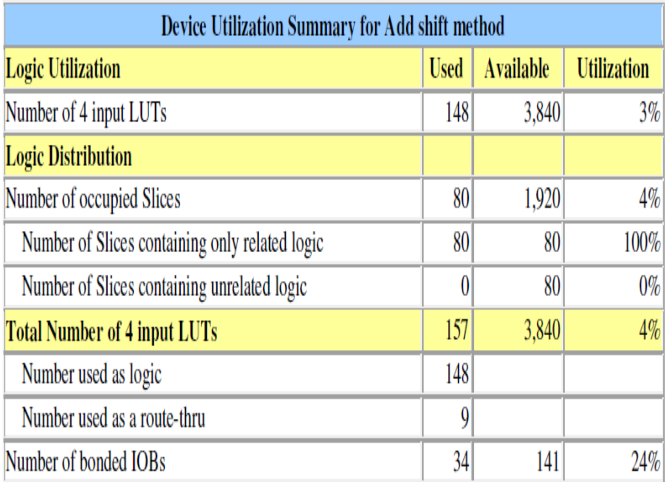
V. Power and Area Analysis of R2mdc and Add Shift Method

Table 1 Power Analysis



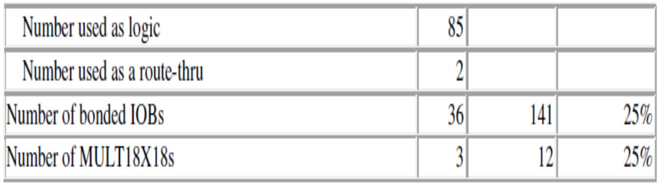
ARCHITECTURE	SYNTHESIS POWER	MULTIPLICATION
R2MDC ARCHITECTURE	0.323	3
ADD SHIFT METHOD	1.517	0

Table 2 Area Analysis



Device Utilization Summary for Add shift method			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	148	3,840	3%
Logic Distribution			
Number of occupied Slices	80	1,920	4%
Number of Slices containing only related logic	80	80	100%
Number of Slices containing unrelated logic	0	80	0%
Total Number of 4 input LUTs	157	3,840	4%
Number used as logic	148		
Number used as a route-thru	9		
Number of bonded IOBs	34	141	24%

With the simulation output a vcd file is created in the transcript of modelsim and it is synthesis in Xilinx to show the power and area analysis for r2mdc and add shift method.



Number used as logic	85		
Number used as a route-thru	2		
Number of bonded IOBs	36	141	25%
Number of MULT18X18s	3	12	25%

VI. CONCLUSION

In this paper, two pipeline-based FFT Architectures are proposed. Our proposed variable-length FFT processor that are suitable for various MIMO OFDM-based communication systems, such as IEEE 802.11n, and IEEE 802.16 WiMAX, can perform 256/128/64/16/8-point with 1-4 data sequences. The proposed 8-point FFT processor is used for IEEE 802.11n, and the proposed 8-point FFT processor is used for IEEE 802.11n and IEEE 802.16. To reduce computational complexity and increase hardware utility, we adopt different radix FFT algorithms and multiple-path delay commutates FFT architecture in our processors. The multiple-path delay commutator FFT architectures require fewer delay elements and different radix FFT algorithms require fewer complex multiplications. The processor Implementation are fabricated using UMC 0.18 *m* μ process and their area are 1.5162 2 *mm* and 2.5122 2 *mm*.

REFERENCES

[1]. .M.Arioua, S.Belkouch, M.M.Hassani, “Complex multiplication reduction in pipeline FFT architecture,” In Proc. of 20th Intern. Conf. on Computer Theory and Applications (ICCTA), Alexandria, Egypt, Oct. 2010

[2]. B.Wang, Q. Zhang, Tao, M.Huang, “Design of Pipelined FFT Processor Based on FPGA,” In Proc. of the 2nd Intern. Conf. on Computer Modeling and Simulation (ICCMS’10), pp. 432-435, ISBN 978-1- 4244-5642-0, Jan. 2010, Sanya, Hainan

[3]. Design of COFDM Transceiver Using VHDL International Journal of Computer Applications (0975 – 8887) Volume 5– No.7, August 2010

[4]. FPGA Implementation of OFDM Transceiver for a 60GHz Wireless Mobile Radio System 2010 International Conference on Reconfigurable Computing

[5]. IEEE 802.15 WPAN Task Group 3c (TG3c) Millimeter Wave Alternative PHY Friday, 5 February 2010,

[6]. Hardware Implementation Low Power High Speed FFT Core The International Arab Journal of Information Technology, Vol. 6, No. 1, January 2009

[7]. P.Verma, H. Kaur, M.Singh, M, B.Singh, “VHDL Implementation of FFT/IFFT Blocks for OFDM,” In Proc. of Intern. Conf on Advances in Recent in Communication and Computing, pp. 186-188, PI. 978-1- 4244-5104-3 Kerala, 2009.

[8]. X. Jinsong, L. Xiaochun, W. Haitao, B. Yujing, Z. Decai, Z. Xiaolong, W. Chaogang, “Implementation of MB-OFDM Transmitter Baseband Based on FPGA”, in *4th IEEE International Conference on sCircuits and Systems for Communications* (ICCSC 2008), 2008, 26-28 May 2008