Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Ofdm System Using FFT and LFFT

Mr. ABHIJIT D. PALEKAR
M.E. 2nd year
Dept. of Electronics and Telecommunication
GHRCEAM, Amravati, India

Dr. PRASHANT V. INGOLE
Principal and Professor
Dept. of Electronics and Telecommunication
GHRCEAM, Amravati, India

Abstract— A fast Fourier transform (FFT) is an algorithm to compute discrete Fourier transform. A Fourier transform converts time domain signal information into frequency domain. As a result, FFT is a widely used in DSP technique and in many applications such as communication. FFT has been described as most important numerical algorithm. FFT is one of the rudimentary operations in the field of digital signal and image processing. Using FFT is indispensable in most signal processing application. In this paper we propose to use efficient multiplication technique to reduce the partial product which is happened in conventional multiplication technique therefore the FFT and inverse fast Fourier transform (IFFT) with efficient multiplication and with increased speed is used for Orthogonal Frequency Division Multiplexing (OFDM) Modulator and Demodulator blocks. In many applications high speed and efficient multiplication is desired. For this purpose conventional multicarrier technique are usually chosen, but this results in lower spectrum efficiency. So, the principles of OFDM are used. This proposed work will be processing block of an OFDM system, which are applied to FFT and IFFT. Actually, in entire architecture all the mathematical manipulation takes place in transmitter and receiver block i.e. IFFT and FFT block respectively. The speed enhancement is the key contribution of the main processing blocks in OFDM system.

Keywords— FFT, IFFT, OFDM, and VHDL

I. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) has recently become a key modulation technique for both broadband wireless and wire-line applications. It has been adopted for digital audio broadcasting (DAB) and digital terrestrial television broadcasting (DVB). OFDM is a special case of Multicarrier transmission, where a single data stream is transmitted over number of lower rate Subcarrier. The problem of intersymbol interference (ISI) introduced by multipath channel is significantly reduced in OFDM by using the cyclic prefix (CP) as a guard interval between OFDM blocks.

The proposed work would be a brief overview of IFFT & FFT algorithm to be effectively used in OFDM system.OFDM is a special case of multicarrier transmission, where a single data stream is transmitted over a number of lower rate subcarriers. The main reason to use OFDM is to increase the robustness against the selective fading or narrowband interference. In single carrier system if signal get fade or interfered then entire link gets failed whereas in multicarrier system only a small percentage of the subcarriers will be affected. The total signal bandwidth, in a classical parallel data system, can be divided into N non-overlapping frequency sub-channels. Each sub-channel is modulated a separate symbol and then N sub-channels are frequency multiplexed. The general practice of avoiding spectral overlap of sub-channels was applied to eliminate inter-carrier interference (ICI). This is shown in Fig. 1(A). This resulted in insufficient utilization of the existing spectrum.

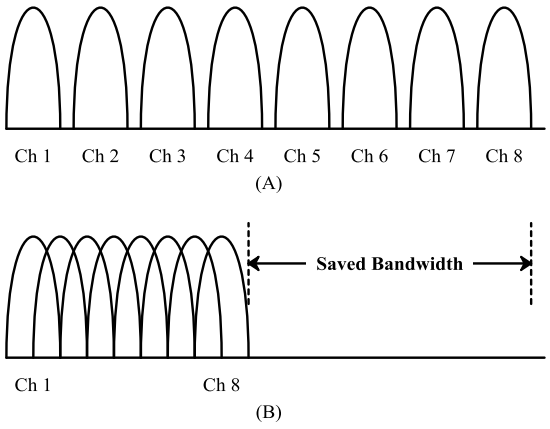

Fig. 1(A) Spectrum of FDM showing Guard Bands (B) Spectrum of OFDM showing Overlapped Subcarriers

An idea was proposed in the mid1960's to deal with this wastefulness through the development of frequency division multiplexing (FDM) with overlapping sub-channels. The sub-channels were arranged so that the sidebands of the individual carriers overlap without causing ICI. This principle is shown in Fig 1(B). To achieve this, the carriers must be mathematically orthogonal. From this constraint the idea of OFDM was born. OFDM is a combination of modulation and multiplexing. Multiplexing generally refers to independent signals, those produced by different sources. In OFDM the signal itself is first split into independent channels, modulated by data and then re-multiplexed to create the OFDM carrier. OFDM is a special case of FDM.

II. OFDM

A. *OFDM System with Multiplier*

Mathematically modulating a waveform and adding it is equivalent to taking an IFFT. This is because the time domain representation of OFDM is made up of different orthogonal sinusoidal signals which are nothing but inverse Fourier transform. The block diagram of digital OFDM system is shown in Fig 2.

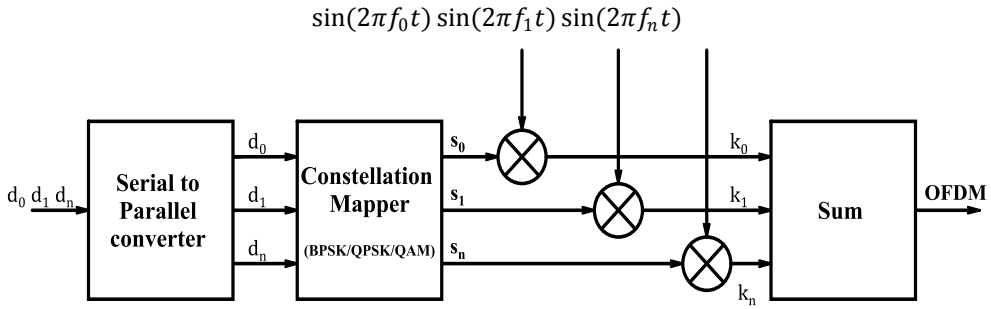

Fig. 2 Implementation of Analog OFDM system.

B. *Fast Fourier Transform*

The fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) are derived from the main function, which is called discrete Fourier transform (DFT). In DFT, the computation for N-points of the DFT will be calculated one by one for each point. While for FFT/IFFT, the computation is done simultaneously and this method saves quite a lot of time. The equations for FFT/IFFT function can be derived from the general DFT equation

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi k}{N}}$$

*X(k)*represents the DFT frequency output at the *k*-the spectral point where *k* ranges from *0* to *(N-1).* The quantity *N* represents the number of sample points in the DFT data frame. The quantity *x(n)* represents the *n*th time sample, where *n* also ranges from *0* to *N-1*. In general equation, *x(n)* can be real or complex. The input can be grouped into odd and even number.

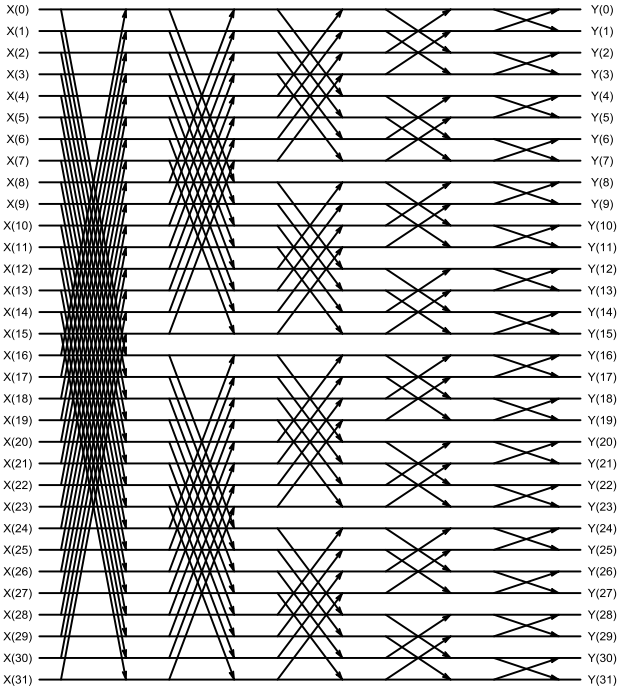

Fig. 3 32-point DIF-FFT

From fig. 3, the FFT computation is accomplished in five stages. The *X(0)* until *X(31)* variables are denoted as the input values for FFT computation and *Y(0)* until *Y(31)* are denoted as the outputs. There are two operations to complete the computation in each stage. The upward arrow will execute addition operation while downward arrow will execute subtraction operation. The subtracted value is multiplied with twiddle factor value before being processed into the next stage. This operation is done concurrently and is known as butterfly process.

C. OFDM using FFT and IFFT

The aim is to design an OFDM transmitter and receiver using FPGA. At the transmitter end, the OFDM signal is generated by implementing the IFFT. At the receiver end, the FFT is implemented.

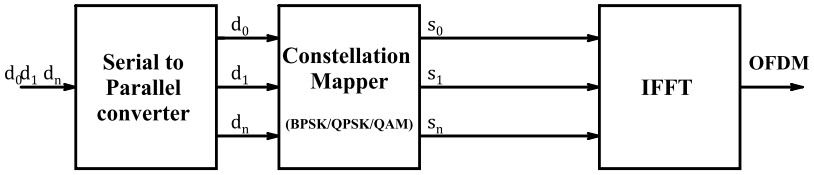

Fig. 4 Implementation of Digital OFDM

The objective is to use High-Speed-Integrated-Circuit to produce VHDL codes that carry out FFT and IFFT function.

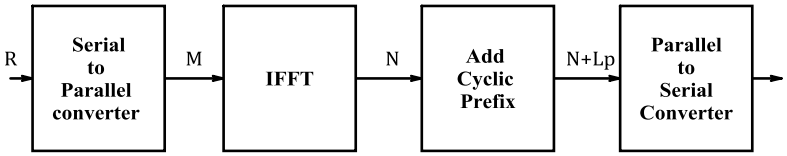

Fig. 5 OFDM Transmitter

Fig. 5 shows the OFDM transmitter communication system. The main focus is the FFT and IFFT part of the OFDM system.

The input symbols are fed to the transmitter in series at R symbols/second. These symbols pass through a serial to parallel converter and output data on M lines in parallel. The M symbols are sent to an IFFT block that performs N-point IFFT operation. The IFFT transform a spectrum (amplitude and phase of each component) into a time domain signal. An IFFT converts a number of complex data points, of length that is power of 2, into the same number of points in time domain. The output is N time-domain samples. In order to preserve the sub-carrier orthogonality a cyclic guard interval is introduced. In this case, assumed a cyclic prefix of length Lp samples is pre-pended to the N samples.

For example, assume N=4 and Lp=2; if the outputs of a 4 point inverse Fourier transform is [1 2 3 4], the cyclic prefix will be [3 4]. The cyclically extended symbol would be [3 4 1 2 3 4]. Therefore, the length of the transmitted OFDM symbol is N+Lp. Pre-pending the cyclic prefix aids in removing the effects of the channel at the receiver. ISI can occur when multi path channel cause delayed version of previous OFDM symbol to corrupt the current received symbol. If the value of Lp is greater than or equal to the size of the transmission channel, the ISI will only affect the cyclic prefix.

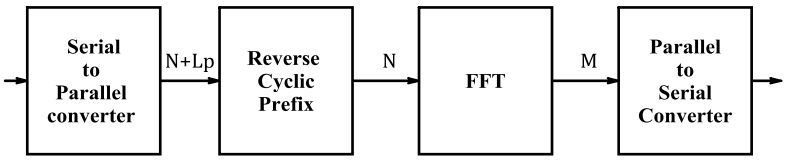

Fig. 6 OFDM Receiver

The received symbol is in time domain and it is distorted due to the effect of the channel. The received signal goes through a serial to parallel converter and cyclic prefix removal. After the cyclic prefix removal, the signals are passed through an N-point fast Fourier transform to convert the signal to frequency domain. The output of the FFT is formed from the first M samples of the output. The work of the project is focused on the design and implementation of FFT for a FPGA kit. The direct mathematical derivation method is used for this design. In this project the coding is done in VHDL & the FPGA synthesis and logic simulation is done using Xilinx ISE Design Suite. In 2011, K. Harikrishna and T. Rama Rao proposed Pipeline architecture for WiMAX technology using Radix-4Decimation in frequency FFT algorithm [1]. They proposed a memory based recursive FFT design which has much less gate counts, lower power consumption and higher speed. The proposed architecture has three main advantages: fewer butterfly iteration to reduce power consumption, pipeline of radix–2^2butterfly to speed up clock frequency, and even distribution of memory access to make utilization efficiency in SRAM ports. They coded this design in Verilog hardware description language with increase in speed & performance of OFDM.

In 2010, Mounir Arioua *et. al.* [2] proposed an optimized implementation of 8-point FFT processor with radix-2 algorithm in R2MDC architecture. The main issue in FFT operation is the complex multiplication so, they tried to reduce the complexity one of two proposed methods replaces the expensive complex multiplication. For this they have applied

the methods to 8-point FFT and compared it to conventional FFT and R2MDC processor. They designed 8-point FFT with radix-2 in R2MDC architectures and was first coded in VHDL. Then they have proposed a novel 8-point FFT processor based on pipeline architecture with no complex multiplication and compared to Cooley-Tukey and R2MDC processor. The proposed architecture gives an advantage in terms of area, complex multiplication reduction approach for Large Points FFT.

In 2005, Zheng Wang *Et. Al.* [3] Simplified The Complex Multiplication Operation In The Design Of The FFT Structure For More General Case In Which The FFT Point Is Only Power Of Two Rather Than Four. They Discussed Theoretical Analysis Of The Delay In R2MDC Structure.They Designed And Implemented One Type Structure Of FFT Processor For Multicarrier Communication System. For That The Consideration Of Design Has Several Aspects As Follows, Adopt The DIF Algorithm Because Of The Finite Length Effect In The Hardware, Simplify The Complex Multiplication Operation, Replace The Shift Registers With Dual-Ports RAM, Optimize The Reversed Order Structure. The Theoretical Delay Time Of The Proposed FFT Structure Is Derived, And It Has Been Implemented For More General Usage Of Multicarrier System.

In 2013, Nilesh Chide, *Et. Al.* [4] Proposed The Design Of OFDM System Using IFFT And FFT Blocks And Simulation Was Done On XILINX ISE. They Implemented The OFDM Block By Block And Finally Incorrect All Of Them Together To Form Complete OFDM Circuit. Finally Their Aim Is To Implement The Core Signal Processing Blocks Of OFDM System Using VHDL Language. The Different Blocks Of OFDM System Such As QAM Modulator, 8-IFFT, 8-FFT And Demodulator Is Designed On Xilinx Project Navigator. These Blocks Are Simulated On XILINX ISE Design Suite, Tested For Different Patterns And Results Are Compared.

In 2012, Pradeepa M. and Gowtham P. [5] proposed an optimized implementation of the 8-point FFT processor with radix-2 algorithm in R2MDCarchitecture.The 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 the proposed method. So they proposed conventional FFT algorithm by using butterfly technique and then proposed algorithm has been implemented by using R2MDC architecture. Then the analysis and design of FFT is done using VHDL. They proposed variable-length FFT processor that is suitable for various MIMO OFDM based communication systems. To reduce computational complexity and increase hardware utility, they adopt different radix FFT algorithms. The multiple-path delay commutator FFT architectures require fewer delay elements and different radix FFT algorithms require fewer complex multiplications.

In 2012, Manjunath Lakkannavar and Ashwini Desai [6] implemented the core processing blocks of an OFDM system, namely FFT and IFFT. The FFT and IFFT have been chosen to implement the design instead of the DFT and IDFT because they offer better speed with less computational time. The work also includes in designing a mapping module, serial to parallel and parallel to serial converter module. The 8 points IFFT / FFT decimation-in-frequency (DIF) with radix-2 algorithm is analyzed in detail to produce a solution that is suitable for FPGA implementation. The FPGA implementation of the project is performed using VHDL. The performance of the coding is analyzed from the result of timing simulation using Xilinx.

In 2011, Pawan Verma and Harpreet Kaur [7] developed FFT &IFFT algorithms to be used in OFDM systems. The speed enhancement is the key contribution of the main processing blocks in OFDM system so they have successfully implemented the 8-point IFFT & FFT algorithm using VHDL to be used in the architecture of OFDM transmitter & receiver. The performance of the main processing block of OFDM Trans receiver is upgraded by reducing the clock cycles. The real value inputs are given to FFT blocks while all the imaginary input values are zero. Also, the accuracy in obtained results has been increased with the help of efficient coding in VHDL. The accuracy in results depends upon the equation obtained from the butterfly diagram.

In 2007, N. Madhavi, and R. Teymourzadeh [8] proposed an efficient algorithm using parallel and pipelining methods to implement high speed and high resolution FFT algorithm. Latency reduction is an important issue to implement the high speed FFT on FPGA. The Proposed FFT algorithm shows the latency of 5131 clock pulse when N refers to 1024 points. For new architecture of the Radix-2 FFT algorithm, Verilog code was written and simulated by MATLAB software. That design code is downloaded to the Virtax-2 FPGA board. The new architecture of the high speed and high resolution of the parallel, pipeline and floating point Radix-2 FFT algorithm was designed and investigated. High Speed FFT architecture was obtained by two methods. The pipeline structure and parallel design lead them to have high speed FFT algorithm. Additionally, using a internal RAM makes the design compatible with different type of FPGA board.

III. PROPOSED WORK

The proposed work will be to design the 32-point FFT & IFFT blocks for OFDM by using VEDIC multiplication and to achieve the efficient multiplication. The design would be coded in VHDL and synthesis will be done in Xilinx ISE software. Then the design will get implemented on FPGA kit. Therefore, we can get the efficient multiplication of FFT and IFFT block to be used in OFDM receiver and transmitter respectively.

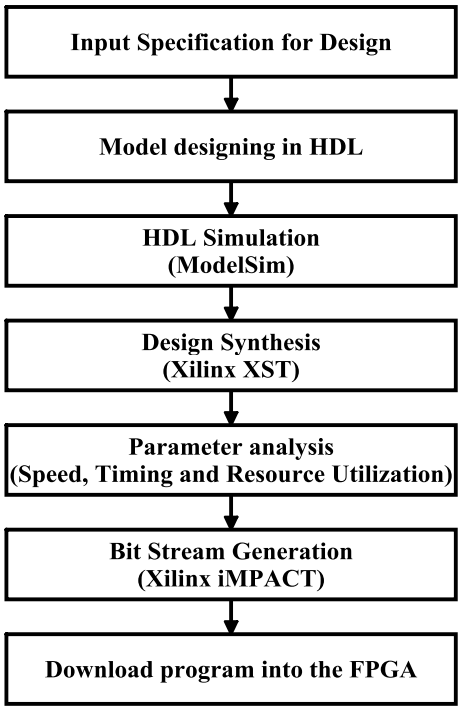

Fig. 7 Proposed Model Flow

IV. CONCLUSION

We could propose a 32-point FFT & IFFT design for communication application like OFDM. The main objective of proposed architecture is to design efficient multiplication of FFT & IFFT using Vedic multiplication. It has numerous advantages such as: increase the speed, efficient timing, and better resource utilization parameter. In summary, speeds performance of our design easily satisfies most application requirements based on OFDM modulated wireless communication system.

REFERENCES

[1] Harikrishna K. and Rao T.R., "FPGA based FFT algorithm implementation in WiMAX communications system," , *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (Wireless VITAE)*, pp.1–6, Feb–March 2011.

[2] Arioua M., Belkouch S., Agdad M., Hassani M. M., "VHDL implementation of an optimized 8-point FFT/IFFT processor in pipeline architecture for OFDM systems," *International Conference on Multimedia Computing and Systems (ICMCS)*, pp.1-5, April 2011.

[3] Zheng Wang, Mingke Dong, Yuping Zhao, "Design and implementation of efficient FFT processor for multicarrier system," *Canadian Conference on Electrical and Computer Engineering*, pp. 1384-1387, May 2005.

[4] Nilesh Chide, Shreyas Deshmukh, Prof. P. B. Borole, "Implementation of OFDM System using IFFT and FFT," *International Journal of Engineering Research and Applications (IJERA)*, Vol. 3, pp. 2009-2014, Jan. –Feb. 2013.

[5] Pradeepa M., Gowtham P., "Optimized Implementation of FFT Processor for OFDM system," *International Journal of Advances in Engineering & Technology (IJAET)*,vol. 3, pp. 429-441,May 2012.

[6] Manjunath Lakkannavar, Ashwini Desai, "Design and Implementation of OFDM (Orthogonal Frequency Division Multiplexing) using VHDL and FPGA," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 1, pp. 211-213, Aug. 2012.

[7] Verma, P., Kaur, H., Singh, M., Singh, B., "VHDL Implementation of FFT/IFFT Blocks for OFDM," *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 186-188, 27-28 Oct. 2009.

[8] N. Mahdavi, R. Teymourzadeh, IEEE Student Member, Masuri Bin Othman, "VLSI Implementation of High Speed and High Resolution FFT Algorithm Based on Radix 2 for DSP Application," *The 5th Student Conference on Research and Development*, pp. 1-4,11-12 Dec. 2007.

[9] Haveliya A., "Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation," *Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, pp. 167-171, 7-8 Jan. 2012.

[10] Correa I. S., Freitas L.C., Klautau, A., Weyl Albuquerque Costa J.C., "VHDL Implementation of a Flexible and Synthesizable FFT Processor," *IEEE Latin America Transactions*, vol. 10, pp. 1180-1183, Jan. 2012.

[11] Modelsim manual, http://support.xilinx.com.

[12] Xilinx, Inc. http://www.xilinx.com/.

