

VLSI Implementation of Area-Efficient and Low Power OFDM Transmitter and Receiver

D. Kalaivani* and S. Karthikeyan

VLSI Design, Sathyabama University, Chennai - 600119, Tamil Nadu, India;
kalaivanidurai07@gmail.com, karthijoy1@gmail.com

Abstract

Background: This paper consists of an analysis of Fast Fourier Transform (FFT) architectures which are the backbone of any OFDM based wireless networks. **Methods:** By using the FFT concepts we are indeed in developing an efficient architectures for wireless networks which are common in everywhere now-a-days, Our concepts are purely based upon in development we have to test it by using Field Programmable Gate Array (FPGA), we are using Xilinx based Spartan-3e FPGA. **Results:** The concepts are simulated by using Modelsim6.3c and to synthesize by using Xilinx ISE 10.1.

Keywords: FFT, FPGA, OFDM, SDF-SDC

1. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is one of the FDM techniques in transmitting the data with high data rate between the devices; In particular, many wireless standards (Wi-Max, IEEE802.11a, LTE, and DVB) have adopted the OFDM technology as a mean to increase dramatically future wireless communications OFDM is a particular form of Multi-carrier transmission and is suited for frequency selective channels and high data rates¹. This technique transforms a frequency-selective wide-band channel into a group of non-selective narrowband channels, which makes it robust against large delay spreads by preserving orthogonality in the frequency domain²⁻⁷.

OFDM is a multicarrier modulation technique as shown in the Figure 1, in which several carriers are transmitted over the allocated bandwidth to carry the information from source to destination. For this, each carrier may use one of the several available digital modulation techniques like BPSK, QPSK, and QAM.

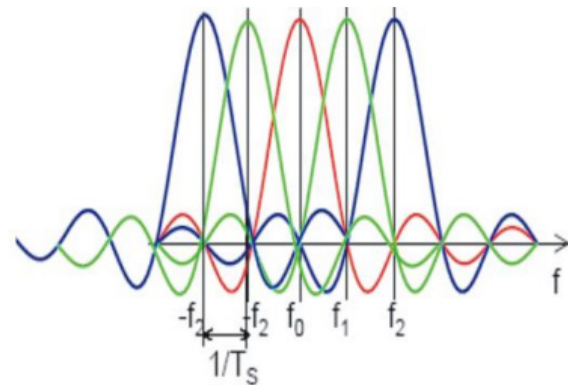


Figure 1. Orthogonal Frequency Division Multiplexing Modulation.

2. Existing System

The input data stream is split into N parallel low bandwidth modulated data streams. Due to orthogonality of subcarriers they do not interfere with one another. Each subcarrier has a low symbol rate. But the combination of subcarriers carrying the information in parallel allows

*Author for correspondence

for high data rates. Low symbol rate is used to reduce the problem of Inter Symbol Interference (ISI)⁸⁻¹⁰. Before modulation the transmitter stage of an OFDM transceiver takes data, Converts the data and encodes it into a serial stream. The generation of OFDM signal is takes place by using Inverse Fast Fourier Transform (IFFT). Reverse process takes place in receiver stage.

3. Transmitter

The Transmitter block diagram will consists of BPSK Modulation Scheme, Serial to parallel block, IFFT, parallel to serial block and Cyclic Prefix- Figure 2.

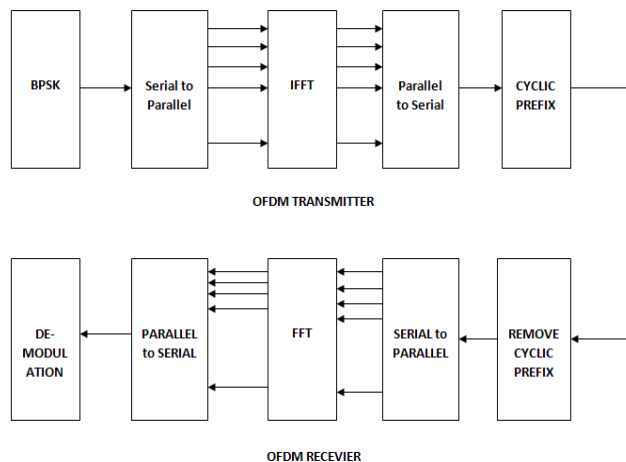


Figure 2. Block diagram of Orthogonal Transmitter and receiver.

3.1 BPSK Modulation

The BPSK Modulation scheme is one of the Phase shift keying Technique that can be used for the Modulation, with 180o phase shift keying is applied.

3.2 Serial to Parallel

The Serial to parallel block contains Technique that converts serial form input data to parallel form outputs, it converts the data serial inputs to parallel outputs that can be works after getting all inputs in the parallel signal.

3.3 IFFT Block

The IFFT Block is converts the Input data from Frequency domain to Time domain outputs.

3.4 Cyclic Prefix

The Cyclic Prefix is block that assigns empty memory space with your input data in the base of $1/N$ value of the input is to be assigned for cyclic prefix memory.

3.5 Receiver Block

The Receiver block contains Remove cyclic prefix block, serial to parallel block, FFT Block, Parallel to serial block, Demodulation.

3.6 Remove Cyclic Prefix

The Remove cyclic prefix is removes the empty space that created in the transmitter block. The cyclic prefix

Memory is contains noise and errors that can be removed in this block.

3.7 Fast Fourier Transform Block

The Fast Fourier Transform block is performs the input from time to frequency operations. The FFT blocks basically contains different type of radix based FFT architectures as follows,

- Radix-2
- Radix-4
- Mixed- Radix
- Split Radix

In this paper we propose the pipelined implementation of Radix-2 based single delay feedback (R2SDF), Radix-2 single delay commutator (R2SDC), combined architecture is implemented in the receiver architecture of OFDM.

4. Proposed System

The proposed FFT architecture consists of one pre-stage, $\log_2 N - 1$ SDC stages, one post-stage, one SDF stage, and one bit reverser, The pre-stage shuffles the Complex input data to a new sequence that consists of real part followed by the corresponding imaginary part, shown in Table 1.

The corresponding post-stage shuffles back the new sequence to the complex format. The SDC stage t ($t = 1, 2, \dots, \log_2 N - 1$) contains a SDC PE, which can achieve 100% arithmetic resource utilization of both complex adders and complex multipliers. Final, the even data are retrieved in normal order. Thus, the bit reverser requires only $N/2$ data buffer. The last stage of Single Delay Feedback (SDF)

is identical to the radix-2 Single Delay Feedback (SDF), containing a complex adder and a complex subtractor. By using the modified addressing method¹², the data with an even index are written into memory in normal order, and they are then retrieved from memory in bit-reversed order while the ones with an odd index are written in bit-reversed order- Figure 3.

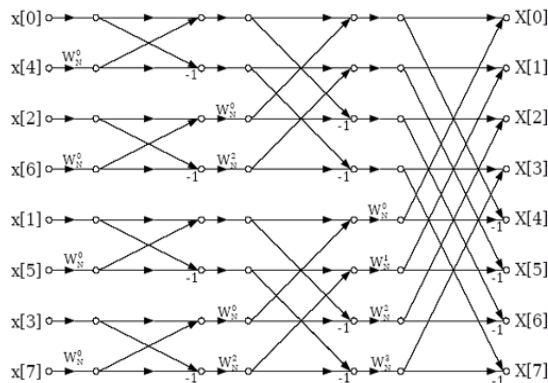


Figure 3. Dataflow graph showing DIT based 8-point FFT.

Normally the Fast Fourier Transform architectures are working based upon the parallel architectures, so FFT's are consume large area and latency, in order to perform low area and latency we are in deed in developing the pruning the FFT's with help of Pipelining Data path, feed forward, feed backward. And some various architectures for reducing the area and delay of the FFT processors, there are various radix-2 based architectures such as R2MDC, R2SDF, R2SDC; these architectures are very useful in making the pipelined operation of FFT's- Figure 4.

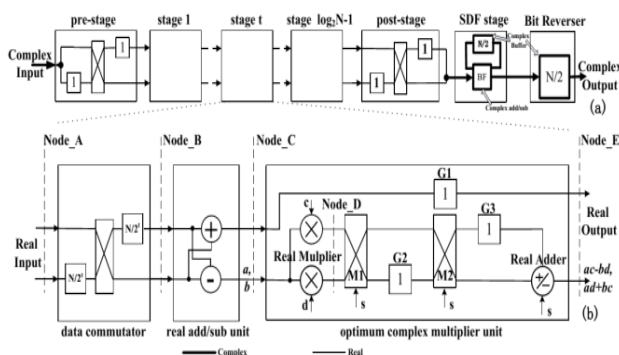


Figure 4. Block diagram showing D pipelined based radix-2 FFT for 8-point.

The proposed system consists of the architecture that can be have the stage-1, stage-2, stage-3, stage-4, stage-5 all are designed for 64-point FFT architectures, as shown in the Figure 5.

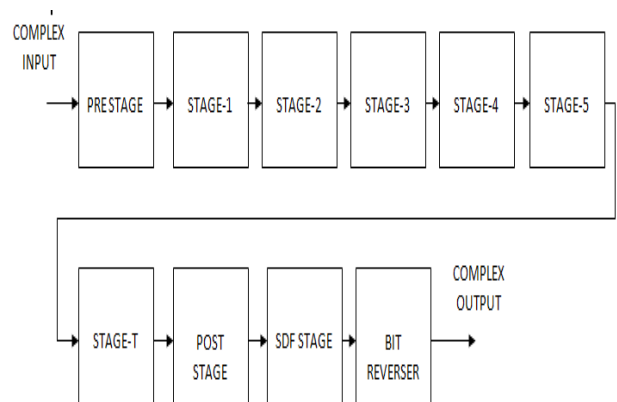


Figure 5. Proposed system block diagram.

The proposed system consists of five stages and the stage-t which is also present in 64-point FFT architecture, All FFT based architectures are Operating based upon the parallel architectures, so we are proposing a Pipelined operation of FFT radix-2 architectures with combination of SDF-SDC architectures in order to achieve higher data rate.

5. Results and Discussion

The Outputs are shown in Figure 6-16.

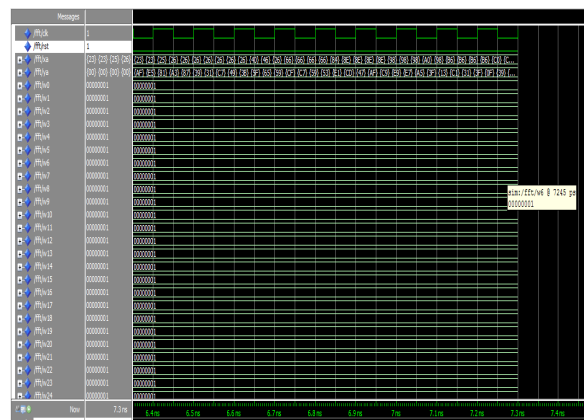


Figure 6. Output screenshot showing simulation waveform for Radix-2 based FFT.

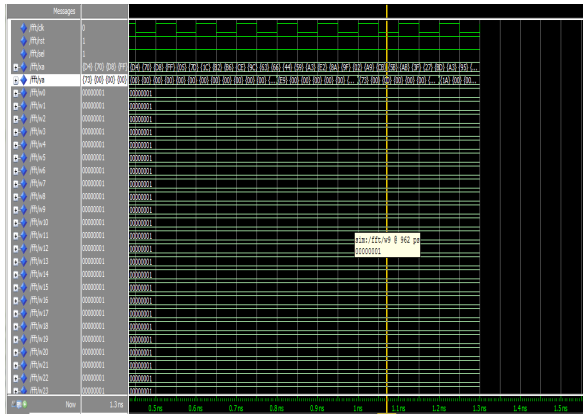


Figure 7. Output screenshot showing simulation waveform for optimum multiplier based FFT.

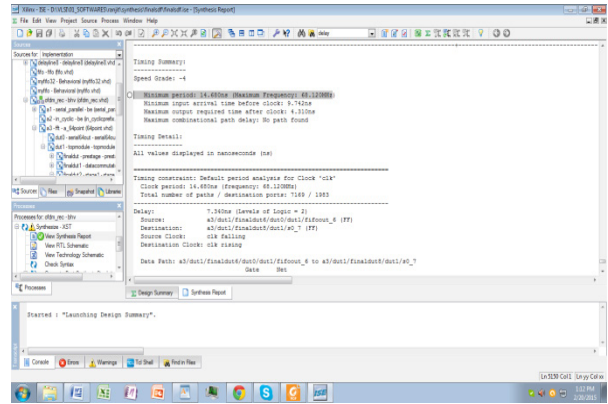


Figure 10. Output screenshot showing the FREQUENCY for Radix-2 based FFT.

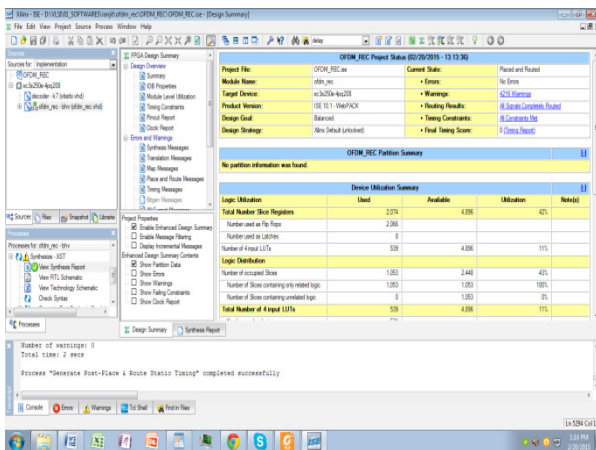


Figure 8. Output screenshot showing the AREA (slices) and LUT for Radix-2 based FFT.

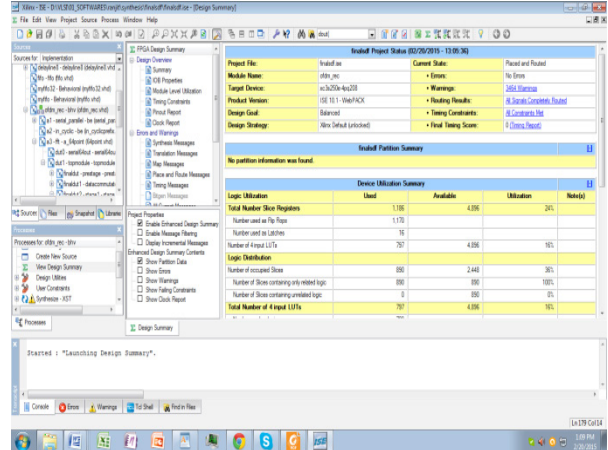


Figure 11. Output screenshot showing the AREA (slices) and LUT for optimum multiplier based FFT.

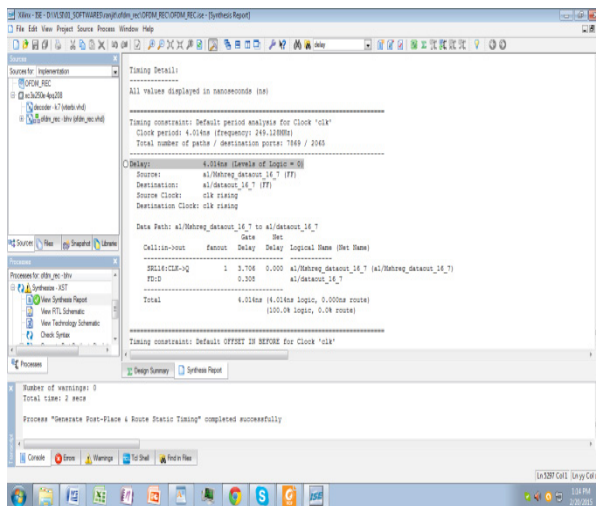


Figure 9. Output screenshot showing the DELAY for Radix-2 based FFT.

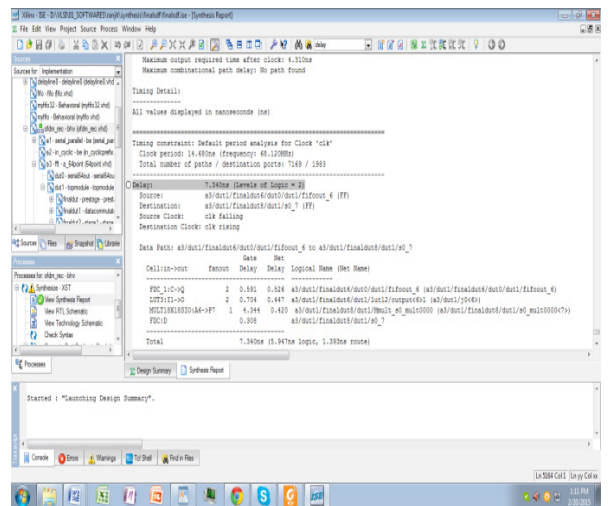


Figure 12. Output screenshot showing the DELAY for optimum multiplier based FFT.

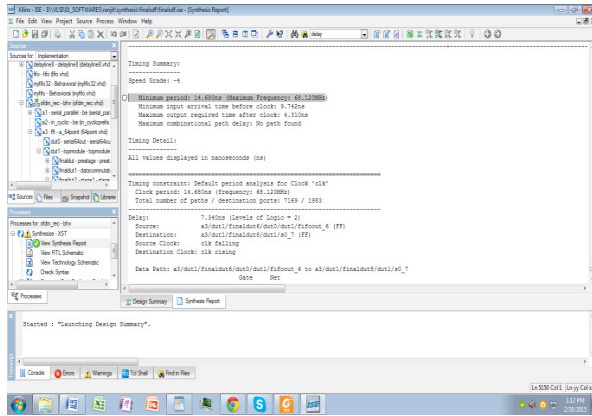


Figure 13. Output screenshot showing the FREQUENCY for optimum multiplier based FFT.

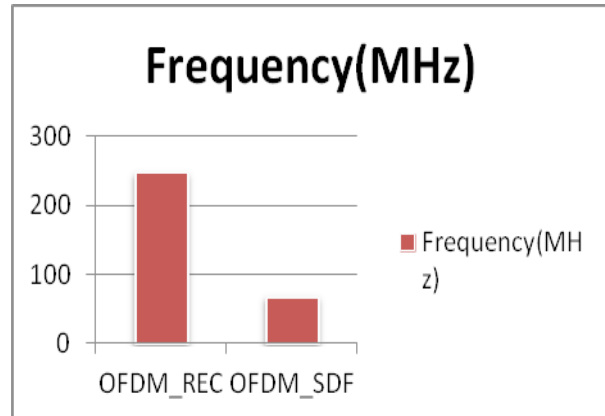


Figure 15. Comparison chart showing OFDM with Radix-2 based FFT with OFDM based optimum multiplier and their Frequency.

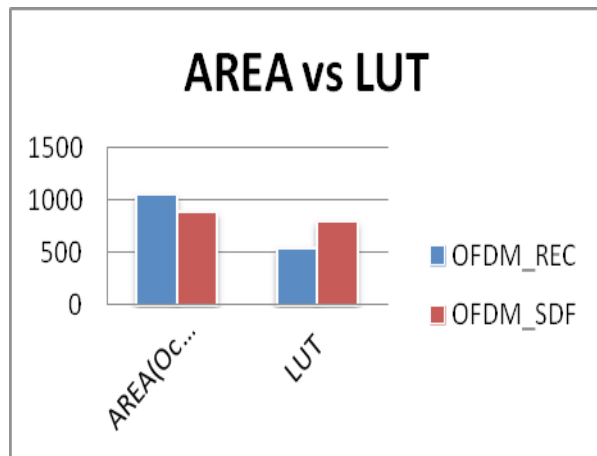


Figure 14. Comparison chart showing OFDM with Radix-2 based FFT with OFDM based optimum multiplier and their AREA (slices) and LUT.

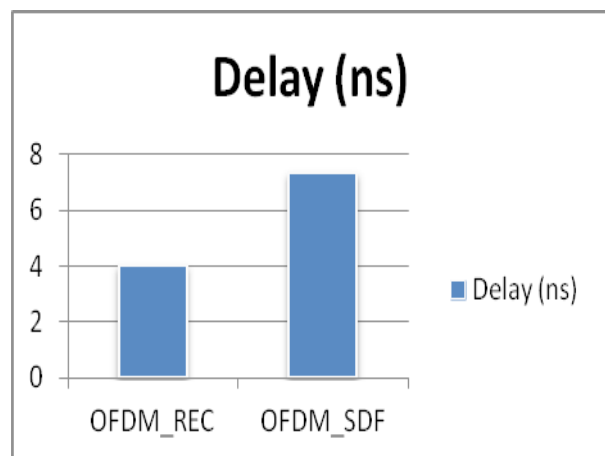


Figure 16. Comparison chart showing OFDM with Radix-2 based FFT with OFDM based optimum multiplier and their Delay.

Table 1. Comparison table showing OFDM with Radix-2 based FFT with OFDM based optimum multiplier

	AREA (Occupied Slices)	LUT	Delay (ns)	Frequency (MHz)	Power (W)
OFDM_REC	1053	539	4.014 ns	248.128	2.772
OFDM_SDF	890	797	7.340 ns	68.120	0.295

6. Performance Analysis

The results shows that the pipelined implementation of this architecture is having low area, high delay and, less frequency is extracted from the results, the delay is high since the architecture is implemented using pipeline architectures.

7. References

1. Cimini LJ. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Trans Commun.* 1985; 665–75.
2. Lin YW, Liu HY, Lee CY. A 1 -GS/s FFT/IFFT processor for UWB applications. *IEEE J Sol-State Circ.* 2005; 40(8):1726–35.
3. Cheng C, Parhi KK. High throughput VLSI architecture for FFT computation. *IEEE Trans. Circuits Syst II, Exp Briefs.* 2007 Oct; 54(10):339–44.
4. Tang SN, Tsai JW, Chang TY. A 2.4-GS/s FFT Processor for OFDM-Based WPAN Applications. *IEEE Trans Circuits Syst II, Exp Briefs.* 2010 Jun; 57(6):451–5.
5. Jung Y, Yoon H, Kim J. New efficient FFT algorithm and pipeline implementation results for OFDM/DMT applications. *IEEE Trans Consum Electron.* 2005; 49:14–20.
6. Karthikeyan S, Shanmugapriya S, Jayashri S. Sub-head Transmission of heterogeneous data by cloned agent to android mobile. *Research Journal of Applied Sciences, Engineering and Technology.* 2014; (1):24–34.
7. ShinM, Lee H. A high-speed, four-parallel radix- 24 FFT processor for UWB applications. *Proceeding of IEEE ISCAS;* 2008. p. 960–3.
8. McClellan JH, Purdy RJ. Applications of digital signal processing to radar. *Applications of Digital Signal Processing;* Englewood Cliffs, NJ: Prentice-Hall; 1978.
9. Rabiner LR, Gold B. Theory and Application of Digital Signal Processing. Prentice-Hall, Inc., USA; 1975. p. 604–9.
10. Wold EH, Despain AM. Pipeline and parallel-pipeline FFT processors for VLSI implementation. *IEEE Trans Comput.* 1984; 33(5):414–26.