

# Implementation of Radon-Framelet Based OFDM on FPGA Platform

Suha Q. Hadi <sup>#\*1</sup>, Phaklen Ehkan <sup>#2</sup>, M. S. Anuar <sup>\*\*3</sup>, Sameer A. Dawood <sup>\*\*\*4</sup>, Hisham Ahmed <sup>\*\*\*\*5</sup>

<sup>#</sup> Embedded, Network and Advanced Computing (ENAC),  
School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia  
<sup>\*</sup> Electrical Department, Al-Mustansiriyah University, Baghdad, Iraq

<sup>1</sup> suhaqasim84@yahoo.com

<sup>2</sup> phaklen@unimap.edu.my

<sup>\*\*</sup> Advance Communication Engineering Centre of Excellence (ACE-COE),  
School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia

<sup>3</sup> anuarms@unimap.edu.my

<sup>\*\*\*</sup> Ministry of Science and Technology, Baghdad, Iraq

<sup>4</sup> sameer.ad80@yahoo.com

<sup>\*\*\*\*</sup> School of Electronics, Sudan University of Science and Technology, Khartoum, Sudan

<sup>5</sup> hisham@ieee.org

**Abstract**—This article proposes an efficient design and implementation of Framelet-Based Orthogonal Frequency Division Multiplexing (OFDM) with Finite Radon Transform (FRAT), named as (FRAT-FT-OFDM), on Field Programmable Gate Array (FPGA) platform. Modern FPGA design tool called Xilinx System Generator (XSG) is used to implement the transceiver of FRAT-FT-OFDM system. The FPGA implementation is carried out on a Zynq (XC7Z020-1CLG484) evaluation board with Joint Test Action Group (JTAG) hardware co-simulation. Performance of the implemented architecture has been validated through hardware co-simulation in term of utilized area and consumed power. The results showed that the system was implemented efficiently with few resources and consumed power, and could support real-time operations.

**Keyword** - OFDM, Framelet Transform, Radon Transform, FPGA, Xilinx System Generator

## I. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) is a special case of multicarrier transmission technique for achieving high data rate transmission over multipath fading channel. It has been widely applied in many wireless and mobile networks especially for the communication standards [1]. In conventional OFDM system, Inverse Fast Fourier Transform (IFFT) is used at transmitter side and Fast Fourier Transform (FFT) is used at receiver side to generate orthogonal subcarriers [2]. The derivation of the system model also shows that by introducing a Cyclic Prefix (CP), the orthogonality can be maintained over a dispersive channel and the Inter-Symbol Interference (ISI) can be avoided but this come at the cost of decreasing the transmission efficiency and increasing complexity of the system [3].

OFDM based on wavelet transforms were also proposed by a number of researchers [4-7]. The idea of using wavelet transform instead of Fourier transform, because of its high spectral containment properties compared with Fourier transform. Among these, Framelet transform-based OFDM (FT-OFDM) [6] was shown to have better Bit Error Rate (BER) performance than Discrete Wavelet Transform (DWT)-based OFDM and FFT-based OFDM. However, FT-OFDM is unsuitable for outdoor wireless applications, where data must be transmitted at high rates with Doppler spread robustness. Therefore, Hadi et al. [8] improved the performance of FT-OFDM system by using Finite Radon Transform (FRAT) as a mapping technique and combined it with FT to produce a new model called as FRAT-FT-OFDM system. In this system, Radon transform was used as a mapping technique instead of conventional mapping techniques such as Quadrature Amplitude Mapping (QAM), and the FT was used to realize the multicarrier modulation instead of FFT or DWT.

For the time being, hardware realization of OFDM system using Field Programmable Gate Array (FPGA) is an important area of research due to the speed of implementation, low development costs and less time to market with the ability to reduce the complexity and power consumption without sacrificing in the performance of the system [9]. FPGA is truly parallel in nature so different processing operations do not have to compete for the same resources. As a result, the performance of one part of the application is not affected when additional processing is added [10]. This will be the best choice for OFDM implementation because it grants flexibility to the program design besides the hardware component will be less costly compared to others.

This article aims to present an efficient design and implementation of the FRAT-FT-OFDM model [8] on FPGA platform using a higher abstraction level tool called Xilinx System Generator (XSG) which is a part of System Editions of Vivado® Design Suite. The Vivado Design Suite (v2015.2) will be used in this work. The performance of the proposed architecture has been validated through hardware co-simulation in terms of utilized area and consumed power.

The rest of this article is divided into following sections; Section II gives an overview for the FRAT-FT-OFDM system. Section III presents the design of FRAT-FT-OFDM in FPGA platform. Section IV discusses the hardware co-simulation. Finally, the work of this article is concluded in Section V.

## II. FRAT-FT-OFDM SYSTEM

Fig. 1 depicts the basic blocks of FRAT-FT-OFDM transceiver.

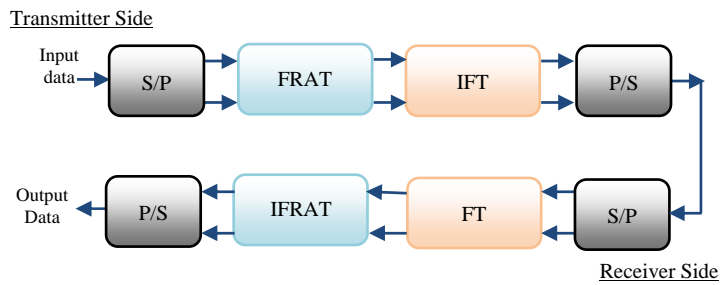


Fig. 1. Basic blocks of FRAT-FT-OFDM transceiver

At the transmitter, a serial binary input data stream is converted to parallel form via serial-to-parallel (S/P) converter to construct symbols and great ( $p \times p$ ) matrix ( $A$ ) which is the first step in the FRAT mapping technique. In the FRAT mapping procedure, a two dimensional FFT (2D-FFT) will be applied on this  $A$  and then the resulted matrix will be reordering by applying the optimal ordering algorithm [8] to produce the optimal matrix ( $F_{opt}$ ). After that one dimensional IFFT (1D-IFFT) will be applied on each columns of the  $F_{opt}$  to obtain FRAT matrix  $\mathfrak{R}$ . In order to increase the bit per Hertz of the mapping, the complex matrix  $\overline{\mathfrak{R}}$  will be constructed from the real matrix  $\mathfrak{R}$ , according to the equation (1) [8].

$$\overline{r}_{i,m} = r_{i,j} + jr_{i,j+1} \quad , \quad 1 \leq i \leq p, \quad \forall j = 1,3,5,\dots,p \quad (1)$$

where  $\overline{r}_{i,m}$  represent to the elements of matrix  $\overline{\mathfrak{R}}$ ,  $r_{i,j}$  represent to the elements of matrix  $\mathfrak{R}$ . After that a set of zero symbols is inserted to the end of the signal to decrease the adjacent carrier's interference. The  $N_f$ -point of inverse FT (IFT) is then applied to the signal to achieve orthogonality between subcarriers. The procedure of IFT is based on the matrix multiplication between the reconstruction matrix ( $W_2$ ) given in equation (2) and the input signal [8]. The  $h_0$ ,  $h_1$ , and  $h_2$  are the low and two high-pass filter coefficients. Finally, the transformed data are converted into serial via parallel-to-serial (P/S) conversion and then sent to the receiver over the wireless channel.

$$W_2 = \begin{bmatrix} h_0(0) & 0 & h_0(2) & h_1(0) & 0 & h_1(2) & h_2(0) & 0 & h_2(2) \\ h_0(1) & 0 & \ddots & h_0(3) & h_1(1) & 0 & \ddots & h_1(3) & h_2(1) & 0 & \ddots & h_2(3) \\ h_0(2) & h_0(0) & h_0(4) & h_1(2) & h_1(0) & h_1(4) & h_2(2) & h_2(0) & h_2(4) \\ h_0(3) & h_0(1) & h_0(5) & h_1(3) & h_1(1) & h_1(5) & h_2(3) & h_2(1) & h_2(5) \\ h_0(4) & h_0(2) & \ddots & h_0(6) & h_1(4) & h_1(2) & \ddots & h_1(6) & h_2(4) & h_2(2) & \ddots & h_2(6) \\ h_0(5) & h_0(3) & 0 & h_1(5) & h_1(3) & 0 & h_2(5) & h_2(3) & 0 \\ h_0(6) & h_0(4) & \vdots & h_1(6) & h_1(4) & \vdots & h_2(6) & h_2(4) & \vdots \\ 0 & h_0(5) & \ddots & 0 & h_1(5) & \ddots & 0 & h_2(5) & \ddots \\ \vdots & h_0(6) & \vdots & \vdots & h_1(6) & \vdots & \vdots & h_2(6) & \vdots \\ \vdots & 0 & \vdots & 0 & \vdots & \vdots & 0 & \vdots & \vdots \\ \vdots & \vdots & \ddots & h_0(0) & \vdots & \vdots & \ddots & h_1(0) & \vdots & \vdots & \ddots & h_2(0) \\ \vdots & \vdots & h_0(1) & \vdots & \vdots & \ddots & h_1(1) & \vdots & \vdots & \ddots & h_2(1) \end{bmatrix}_{N_f \times \frac{3N_f}{2}} \quad (2)$$

At the receiver, the inverse operations for each block in the transmitter side must be performed to produce the correct data stream [8].

### III. XILINX DESIGN OF FRAT-FT-OFDM

The FPGA design of FRAT-FT-OFDM model has been performed using XSG tool. In our design, 4-FRAT mapping, which is equivalent to 16-QAM, is selected since it can easily be modified to implement more complex mapping schemes, such as 6-FRAT and 8-FRAT, which are equivalent to 64-QAM and 256-QAM, respectively. The FRAT matrix dimensions (i.e., matrix  $A$ ) were set to  $(3 \times 3)$ . The number of subcarrier ( $N_f$ ) used in this implementation is 8, whilst the number of useful subcarrier is 6, which means that there are two zeros that will be padded.

Each block in the FRAT-FT-OFDM system are designed and tested by a relatively independent subsystem using XSG with aid of MATLAB SIMULINK. Then, all subsystems will be combined to create an entire system. The following subsections will detail each subsystem blocks.

#### A. Xilinx Design of the Transmitter

Fig. 2 depicts the XSG design for the transmitter side of the FRAT-FT-OFDM. As can be seen, this design consists of four subsystems. The "Gateway In" and "Gateway Out" blocks are used to define the inputs and outputs to the Xilinx design. While the "System Generator" token serves as a control panel for controlling system and simulation parameters, and it is also used to automatically compile designs into low-level representations. The concept design for each subsystem block, with its functionality and specifications illustrated in the following subsections:

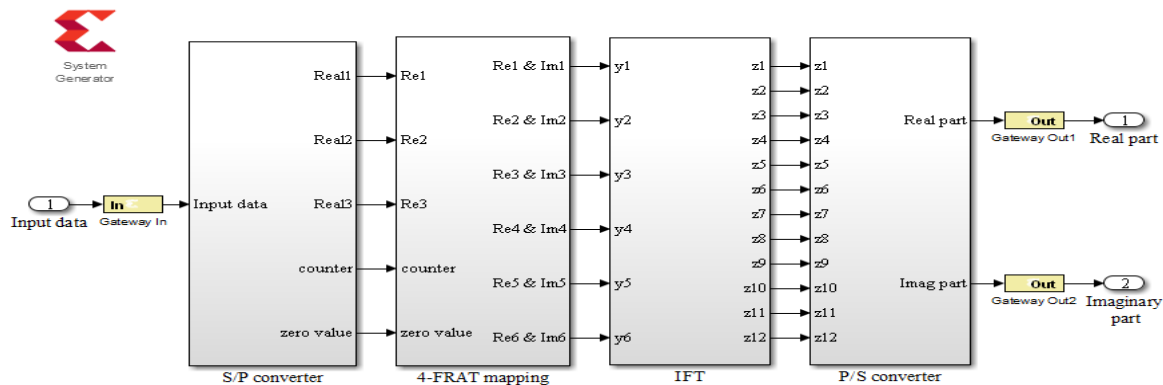


Fig. 2. XSG design of FRAT-FT-OFDM transmitter

#### 1) S/P Subsystem

This subsystem is used to convert the input serial bits to parallel form and construct the symbols, which are then arranged in a  $(3 \times 3)$  matrix (i.e. matrix  $A$ ). The row(s) of the matrix  $A$  is designed for sequential access to the next subsystem. The XSG blocks construction of this subsystem is shown in Fig. 3. The "Serial-to-Parallel" block will convert each two consequential input bits to a single output to be mapped to the corresponding 4-FRAT symbol. Therefore, the clock frequency after this block will be reduced by half. The multiplexer "Mux" blocks will be used to regulate the elements of each row in the matrix, and these blocks are controlled by the "Counter" and the "MCode" blocks. After the entry of the first vector is completed, the "Constant" block with zero value will be used to reset the "Mux" blocks until the next input vector is generated

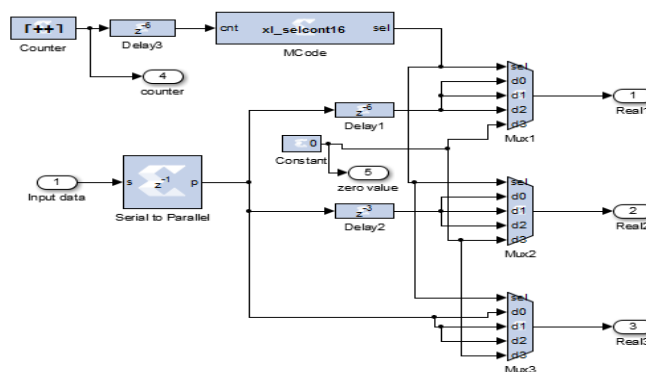


Fig. 3. XSG model of the S/P converter

2) FRAT Mapping Subsystem

The FRAT mapping procedure consist of four steps, so each step of FRAT mapping will be designed in XSG as a separate subsystem, then combined to construct the FRAT mapping subsystem, as shown in Fig. 4.

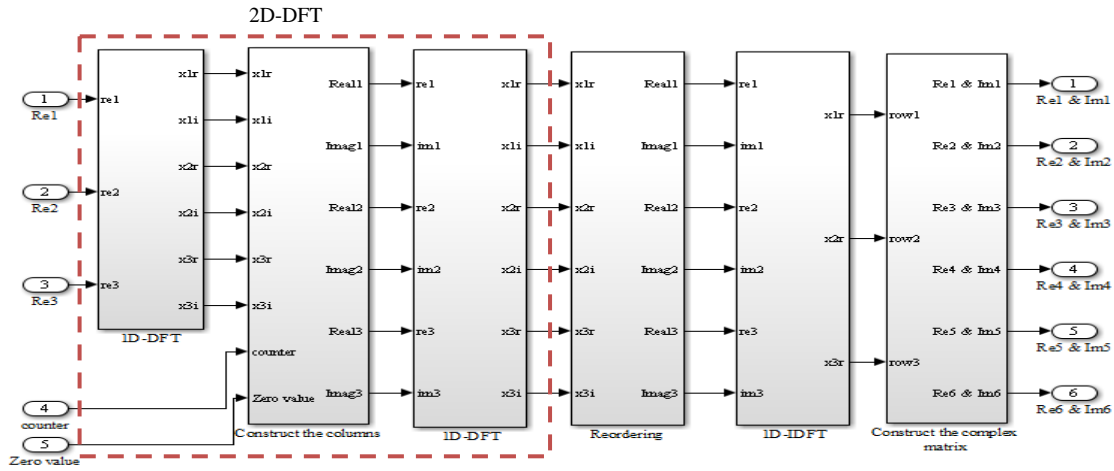


Fig. 4. XSG model of the FRAT mapping subsystem

Step 1: 2D-DFT Subsystem

The first three subsystems in the FRAT mapping design perform the 2D-DFT process to matrix *A*. 2D-DFT is computed using Row-Column (RC) decomposition [11], where the 1D-DFT is computed for each row, and then computed for each column in the resulting matrix. The RC decomposition reported high performance implementation for the 2D-DFT, but it is not favored for large matrix sizes due to the increasing complexity of the design.

The first subsystem is used to perform 1D-DFT on each row of the matrix *A*, which is computed using equation (3) [12]. Fig. 5 shows the XSG model for this subsystem. In this figure, the “Xilinx Constant Multiplier” blocks have been used to perform the multiplication, while the “AddSub” blocks have been used to combine the results. The input of this subsystem is a real numbers only (the imaginary part = 0), which lead to decreased computational operations.

$$X_k = \sum_{n=0}^{N-1} x_n \exp(-j \frac{2\pi nk}{N}), \quad k = 1, 2, \dots, N \tag{3}$$

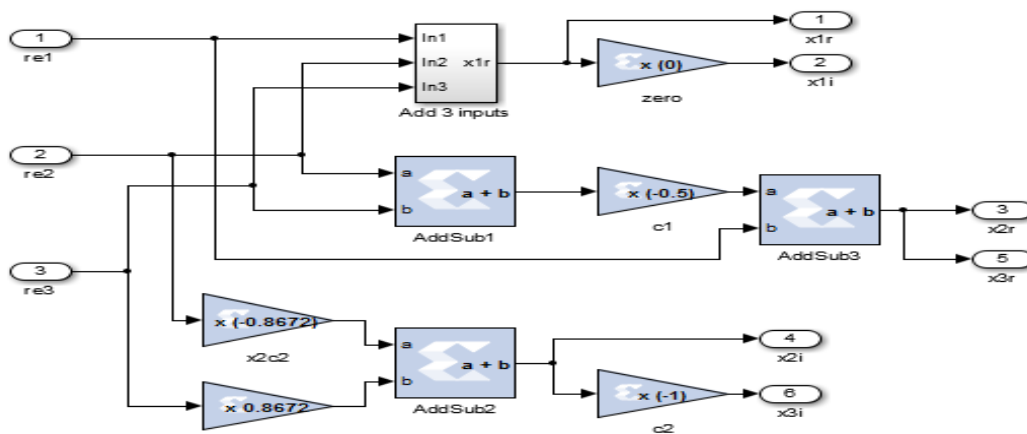


Fig. 5. XSG model of the 1D-DFT (for each row) subsystem

The second subsystem in the design of the 2D-DFT is used to transpose the input matrix. Since, the input to the next subsystem should be in the row form, and we want to perform the 1D-DFT on each column. Therefore, this subsystem has been used to transpose the input matrix and convert each column to the row form before enter to the next subsystem. Fig. 6 illustrates XSG design for this subsystem, the “Mux” blocks will be used to regulate the elements in the form of columns, and these blocks are controlled by the “Counter” and the “MCode” blocks, while the “Delay” blocks are used to control the time of entry and exit of elements.

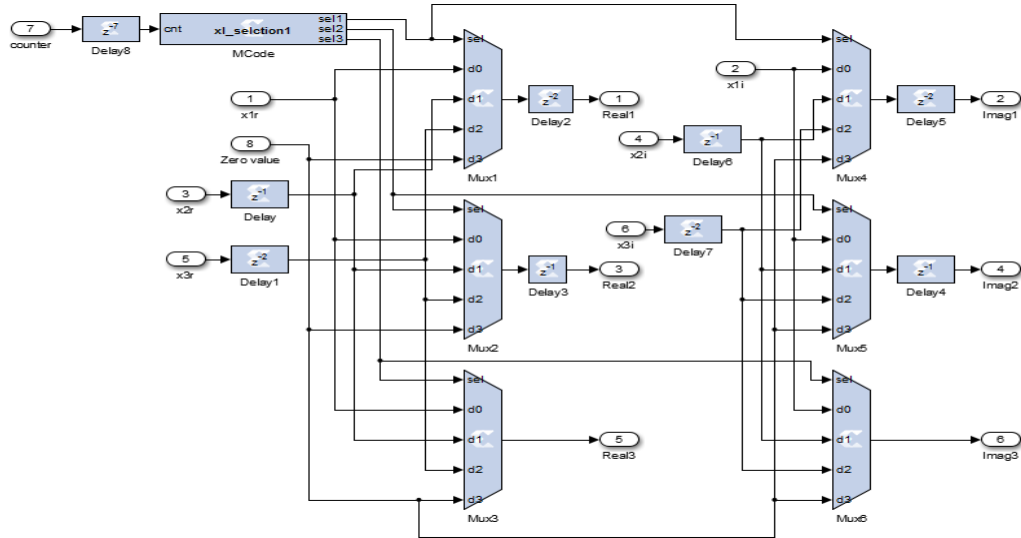


Fig. 6. XSG model of the construct the columns subsystem

The last subsystem in the design of 2D-DFT is used to compute the 1D-DFT on each input columns that comes from the second subsystem. The XSG design for this subsystem is shown in Fig. 7. In this subsystem the design of 1D-DFT is more complex compared with the subsystem given in Fig. 5, because the imaginary part in this subsystem is taken into consideration.

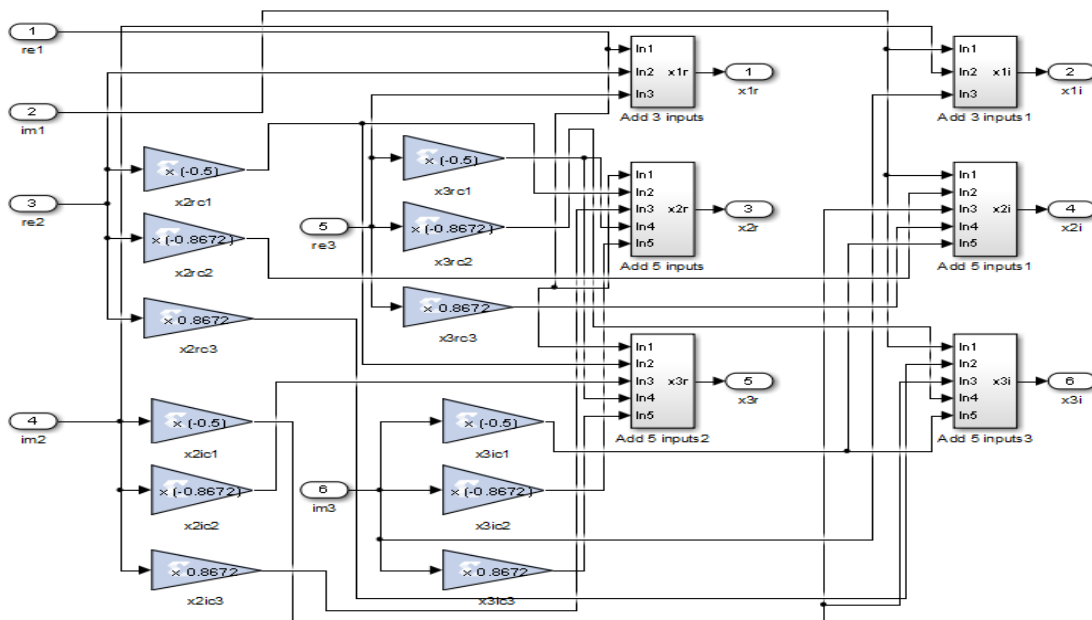


Fig. 7. XSG model of the 1D-DFT (for each column) subsystem

Step 2: Reordering Subsystem

Fig. 8 depicts the XSG design for the reordering subsystem. The action of this design is rearranging the elements of  $(3 \times 3)$  matrix  $A$  and converts it to  $(3 \times 4)$  matrix  $F_{opt}$  (as shown in [13]), which can be achieved by using many of “Time Division Demultiplexer” and “Time Division Multiplexer” blocks. The left part of the figure has been used to reorder the real part of the elements, while the right one has been used to reorder the imaginary part of the elements.

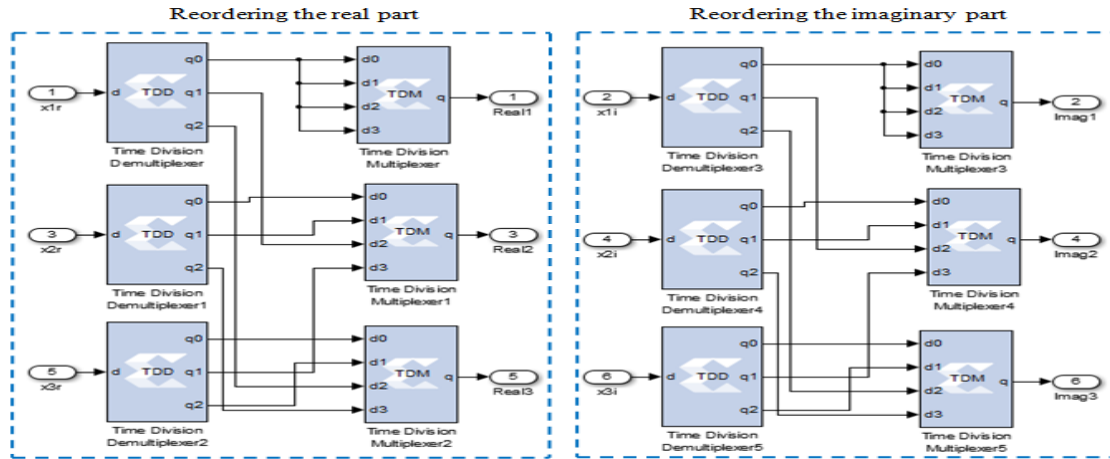


Fig. 8. XSG model of the reordering subsystem

Step 3: IDFT Subsystem

Fig. 9 exhibits the XSG design for Inverse DFT (IDFT) subsystem. The 1D-IDFT has been computed using the following equation [12].

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp(j \frac{2\pi nk}{N}), \quad n = 1, 2, \dots, N \quad (4)$$

As pointed out previously, the multiplication and addition operations are performed using the “Xilinx Constant Multiplier” and “AddSub” blocks, respectively. The imaginary part output from this step is always equal to zero, so the “Terminator” blocks will be used to ignore this value so that it will not be used in the next subsystem. Therefore, only real numbers can access the next subsystem.

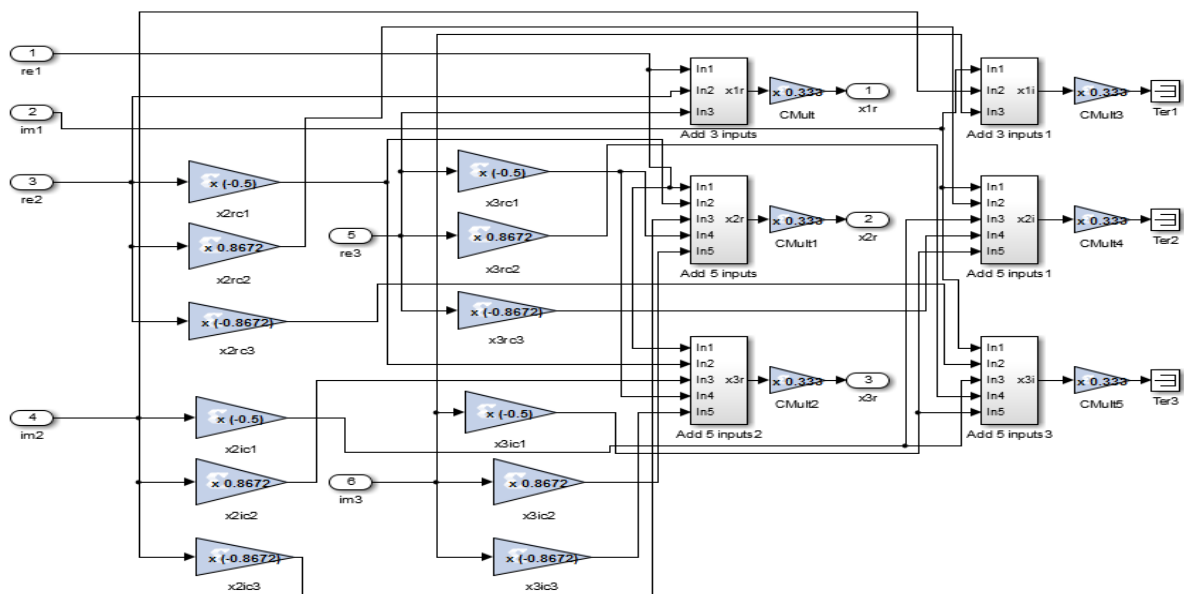


Fig. 9. XSG model of the 1D-IDFT subsystem

Step 4: Construct the Complex Matrix Subsystem

Fig. 10 illustrates the XSG design for this subsystem; actually, this design intends to do two things. The first is to construct a complex matrix which is the last step in the design of FRAT mapping technique. This matrix is designed based on equation (1), so the resulting matrix will be converted from  $(3 \times 4)$  dimensions (i.e., matrix  $\mathfrak{R}$ ) to  $(3 \times 2)$  dimensions (i.e., matrix  $\bar{\mathfrak{R}}$ ). While, the second is to control the input flow to the next subsystem. The purpose of this control is to convert  $(3 \times 2)$  matrix to one vector with a length equal to 6, and then force the real and imaginary components (in this vector) to pass through the same next subsystem (i.e., IFT subsystem) so that each output consists of real values followed by imaginary values. In this case, the design becomes more efficient, and requires fewer resources.

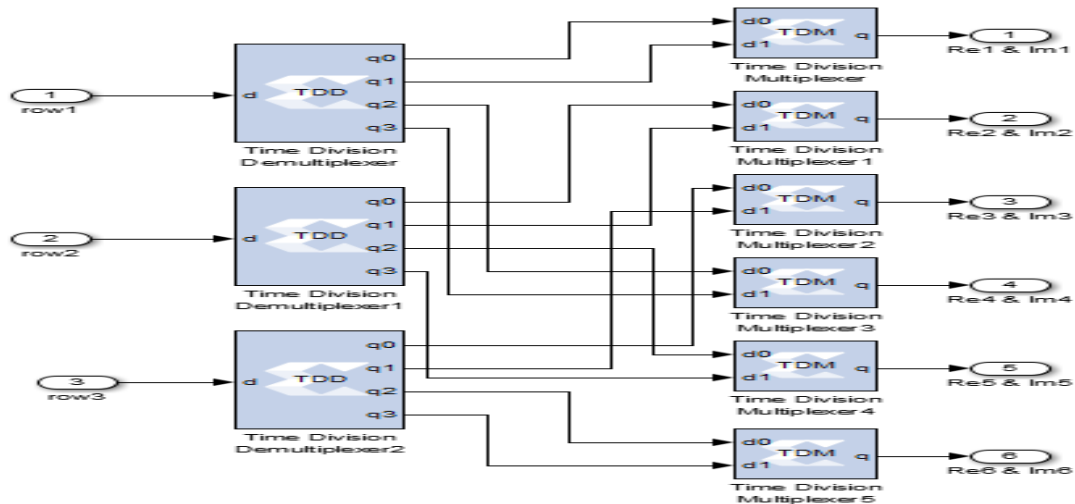


Fig. 10. XSG model of the construct the complex matrix subsystem

3) IFT Subsystem

Fig. 11 depicts the Xilinx design for 8-piont IFT subsystem; this subsystem performs the IFT on the input vector consisting of eight elements, where two of them are zeros padded to the end of this vector. The IFT is performed by applying the matrix multiplication between the  $W_2$  (given in equation 2) and the input vector.

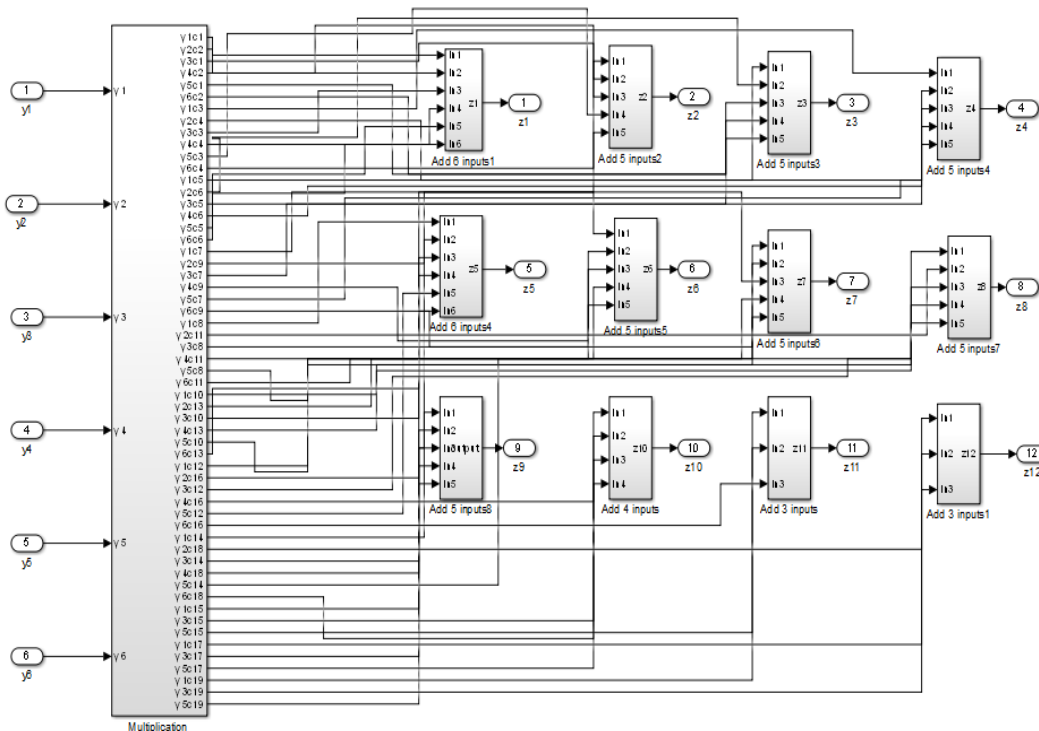


Fig. 11. XSG design of 8-piont IFT subsystem.

4) P/S Subsystem

The XSG design of the P/S converter subsystem is illustrated in Fig. 12. At first, the “Time Division Multiplexer” block has been used to convert the incoming parallel input data to one serial sequence consisting of the real components, followed by imaginary components. Then, two “Mux” blocks are used to separate the real and imaginary parts of the data, in order to transmit them simultaneously.

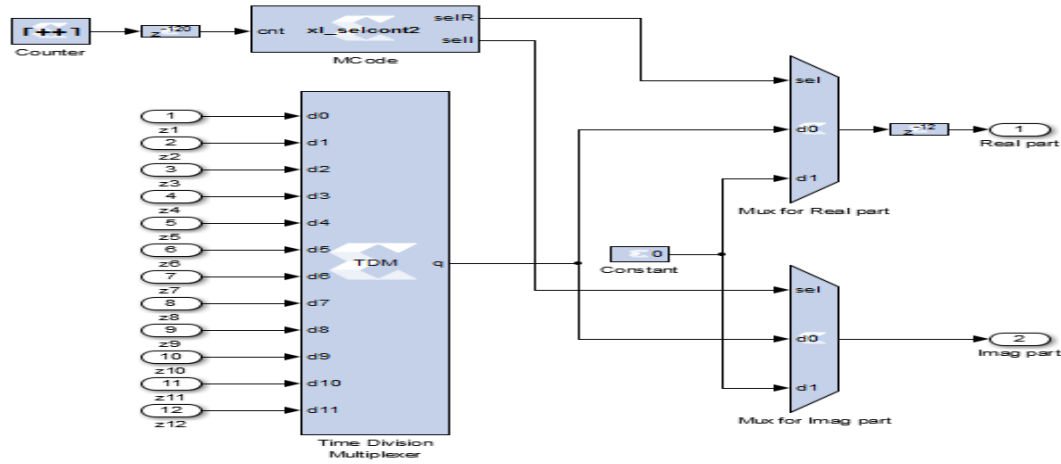


Fig. 12. XSG model of the P/S converter

B. Xilinx Design of the Receiver

Fig. 13 illustrates the XSG design for the receiver side design of the FRAT-FT-OFDM system. The concept design for each subsystem block, with its functionality and specifications, is illustrated in the following subsections:

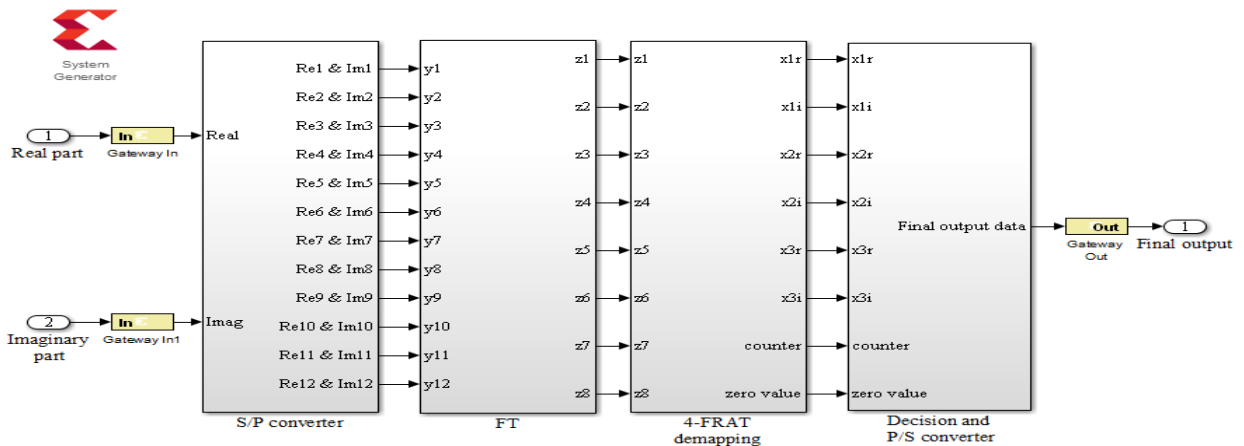


Fig. 13. XSG design of FRAT-FT-OFDM receiver

1) S/P Subsystem

This subsystem converts the serial input data to the parallel form, while simultaneously controlling access of the parallel data to the next subsystem (i.e., FT subsystem) where the real part goes to the next subsystem. After waiting for few sampled periods, the imaginary part goes to the same subsystem to reduce the utilization of resources. The XSG design of this subsystem is illustrated in Fig. 14. “Mux” block has been used to arrange the flow of the input data with the aid of the “Counter” and “MCode” blocks, where d0 and d1 input ports are allocated to pass the real and imaginary components, respectively, while the d2 is allocated to reset the multiplexer between the input vectors. First, the serial real component passes through multiplexer to the “Time Division Demultiplexer” block in order to have a parallel form. After 12 sampling periods, specified by the “Delay” block the imaginary components passes to the same “Time Division Demultiplexer” block in order to have the parallel form as well.



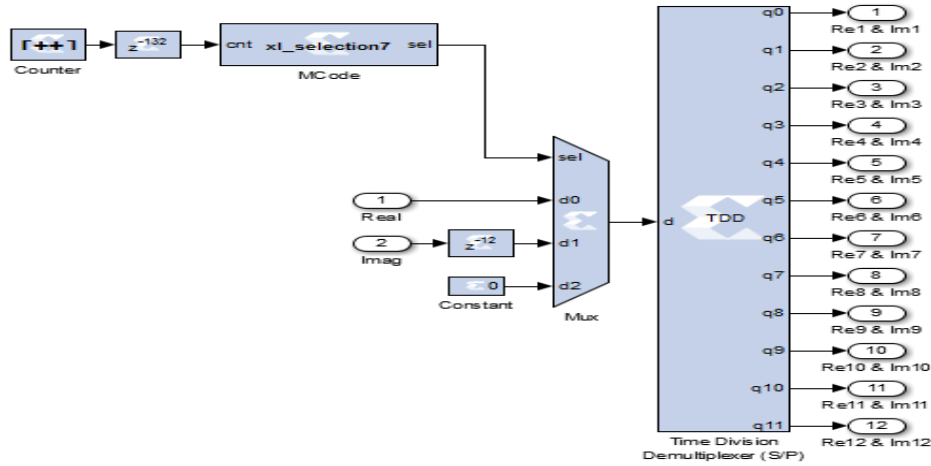


Fig. 14. XSG model of S/P converter

2) FT Subsystem

The purpose of this subsystem is to perform 8-point FT on the input vector, consisting of twelve elements. The FT can be achieved by applying the matrix multiplication between the transformation matrix,  $W_1$ , (which is the transpose of  $W_2$  given in equation 2) and the input vector.

3) FRAT Demapping Subsystem

FRAT demapping consists of four steps as well. Each step will be designed in XSG as a separate subsystem, and then all of these subsystems will be combined to form the FRAT demapping subsystem, as shown in Fig. 15.

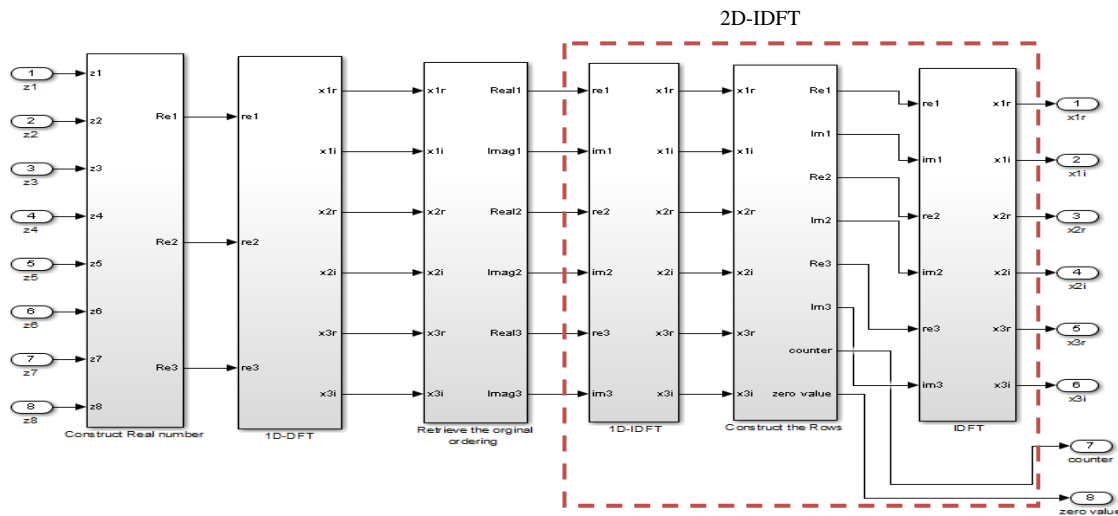


Fig. 15. XSG model of FRAT demapping subsystem

Step 1: Construct Real Matrix Subsystem

Fig. 16 illustrates the XSG design for this subsystem; this design serves two purposes. The first remove zeros padding that were added in the transmitter side previously. The second is to construct a real matrix with (3x4) dimensions from complex numbers. Each incoming data from the previous subsystem consists of real components, followed by the imaginary components, so in order to construct a real matrix; first we should separate the two components and then rearrange them. This has been achieved using “Time Division Demultiplexer” and “Time Division Multiplexer” blocks.

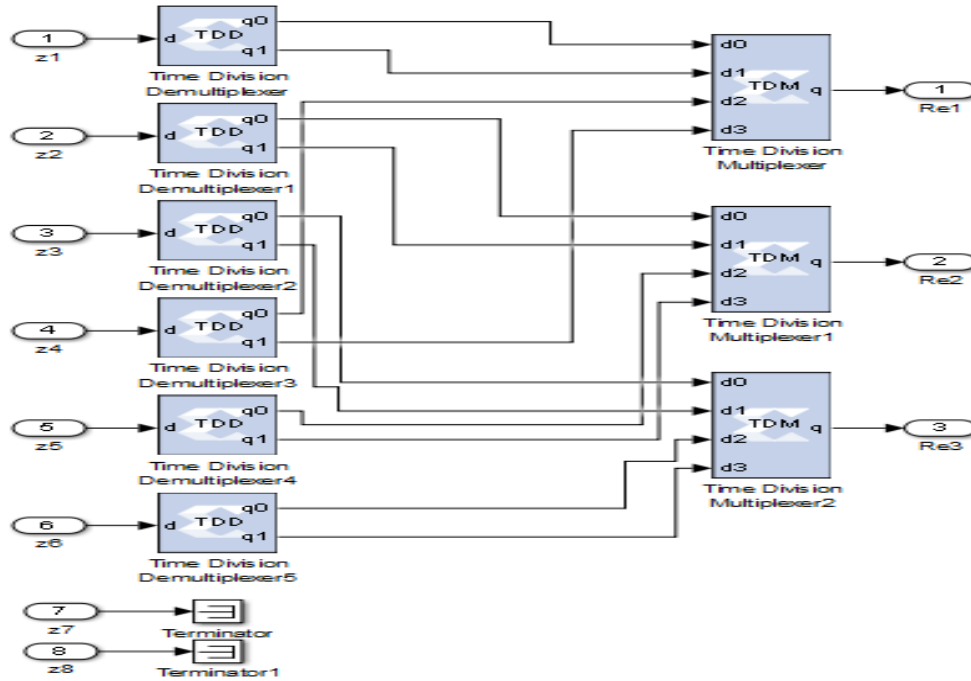


Fig. 16. XSG model of construct real matrix subsystem

Step 2: 1D-DFT Subsystem

In this subsystem, 1D-DFT will be computed on each column of the matrix constructed in step 1. The XSG design of this subsystem was illustrated in Fig. 5, so no need to repeat the design description of this subsystem.

Step 3: Retrieve the Original Ordering Subsystem

Fig. 17 depicts the XSG design for retrieving the original ordering subsystem; the action of this design rearranges the elements of (3×4) matrix, and comes back again to (3×3) matrix. Similar to the design of *Reordering subsystem* in the transmitter, this design has been implemented using “Time Division Demultiplexer” and “Time Division Multiplexer” blocks.

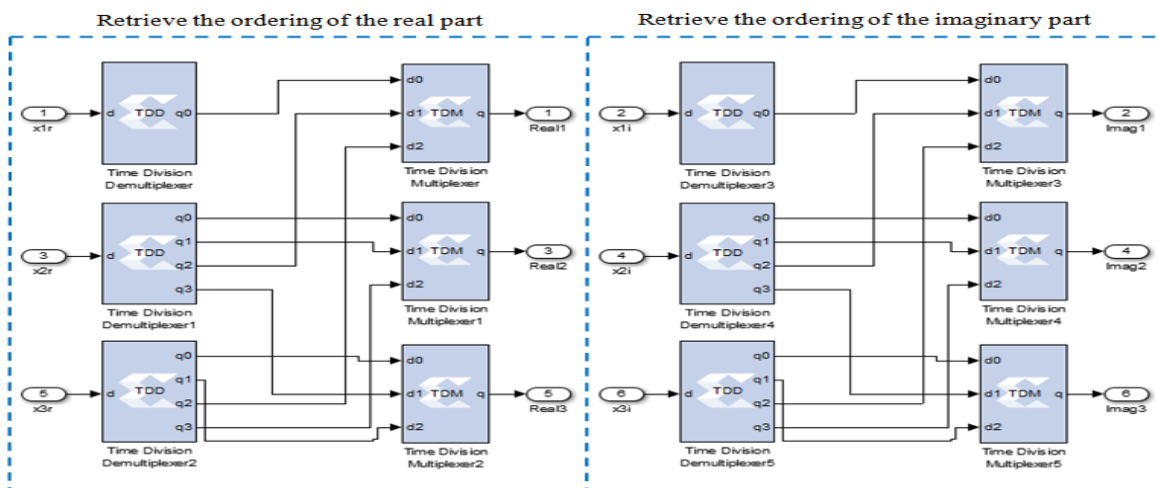


Fig. 17. XSG model of retrieve the original ordering subsystem

Step 4: 2D-IDFT Subsystem

The last step in the FRAT demapping design is performing the 2D-IDFT process for input matrix consists of (3×3) elements. The 2D-IDFT is computed by using inverse procedure of 2D-DFT, where Column-Row (CR) decomposition should be performed. In this case, the 1D-IDFT is computed for each column, and then computed for each row in the resulting matrix. Therefore, the XSG design for this subsystem will be similar procedure of the design 2D-DFT in the transmitter which was described in detail previously.

#### 4) Decision and P/S Subsystem

This subsystem performs a suitable threshold decision for the incoming data in order to complete the FRAT demapping design, then convert the data to the binary in serial form, which is the final step in the design of the receiver side.

Fig. 18 illustrates the XSG design for this subsystem, where two “MCode” blocks is used to perform the threshold decision on the incoming data; one for the real components and the other for the imaginary components. A number of “Delay” and two “Mux” blocks with aid of “Counter” block and another “MCode” block has been used to control the entree of the incoming data to the decision “MCode” blocks. Then, the 4-FRAT symbol is converted to binary data with serial form through “Parallel-to-Serial” block. Because of the imaginary component of the output is always equal to zero, so this component will be neglected using “Terminator” block.

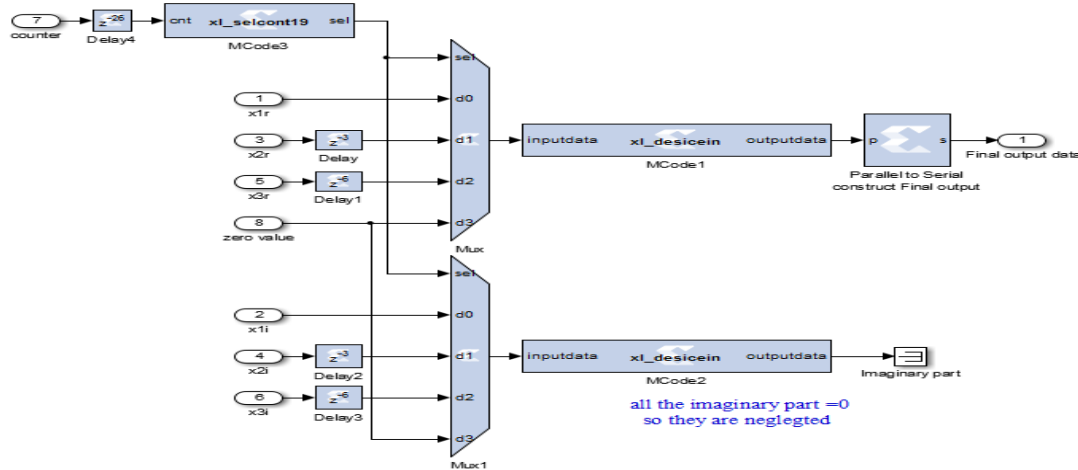


Fig. 18. XSG model of decision and P/S converter subsystem

### IV. HARDWARE CO-SIMULATION

System Generator provides several methods for transforming the models built using SIMULINK into hardware and to simplify the hardware verification and accelerate simulation process, one of these methods is called Hardware Co-simulation. This method provides a hardware-in-the-loop verification, where the inputs and test vectors are generated by SIMULINK, and then the System Generator send them to the design on the FPGA target board using a “Joint Test Action Group (JTAG)” connection in order to carry out the system simulation on the FPGA platform. After that, System Generator read the output back from JTAG and sends it to SIMULINK for displayed.

Fig. 19 shows a picture of a hardware co-simulation model for the FRAT-FT-OFDM system and its programming to Zynq (XC7Z020-1CLG484) FPGA board using a JTAG connection. As seen from this figure, the output from the FPGA board is identical to the output from the Matlab simulation model, and both are identical to the input data without any error in the bit sequences, meaning that the complete model has been successfully loaded onto the FPGA board.

On the other hand, the VHDL (Very high speed Hardware Description Language) codes were automatically generated from the System Generator block sets and then analyzed with Vivado Design Suite for further system verification at Register Transfer Level (RTL), where synthesis/implementation operations are performed. Many hardware and software reports will be produced post-operation, such as the device resources utilization report, timing report, and power consumed report for the FPGA design.

The resource utilization report for the transmitter and the receiver design of the Xilinx FRAT-FT-OFDM model are depicted in Figs. 20 and 21, respectively. It shows that our design utilized very few resources compared with the available resources of the used board, so that this design performs well in the context of resource consumption for both the transmitter and receiver sides.

Figs. 22 and 23 represent the power analysis reports for the transmitter and receiver design, respectively. These reports show that the transmitter design consumed about 14% from the total power of the chip, while the receiver design consumed about 15% from the total power of the chip, and this consumed power are distributed between the clocks, signals, logic elements, and Input/Output pins with specific rates. These reports summarized that there is a little power consumed by Xilinx FPGA design compared with the total power of the used FPGA chip which is a favorite command in any design.

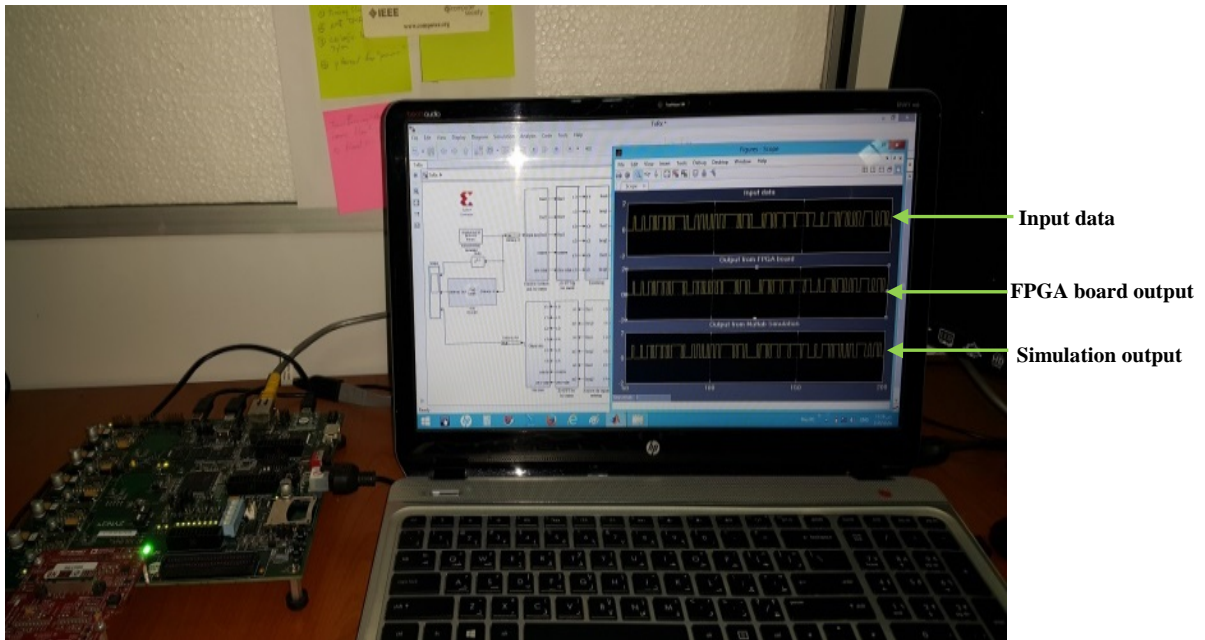


Fig. 19. Hardware Co-simulation model for the FRAT-FT-OFDM

Utilization - Post-Implementation			
Resource	Utilization	Available	Utilization %
FF	970	106400	0.91
LUT	4913	53200	9.23
Memory LUT	36	17400	0.21
I/O	28	200	14.00
BUFG	1	32	3.12

Fig. 20. The utilization area report for the transmitter side of the FRAT-FT-OFDM 1

Utilization - Post-Implementation			
Resource	Utilization	Available	Utilization %
FF	1314	106400	1.23
LUT	5357	53200	10.07
Memory LUT	48	17400	0.28
I/O	28	200	14.00
BUFG	1	32	3.12

Fig. 21. The utilization area report for the receiver side of the FRAT-FT-OFDM

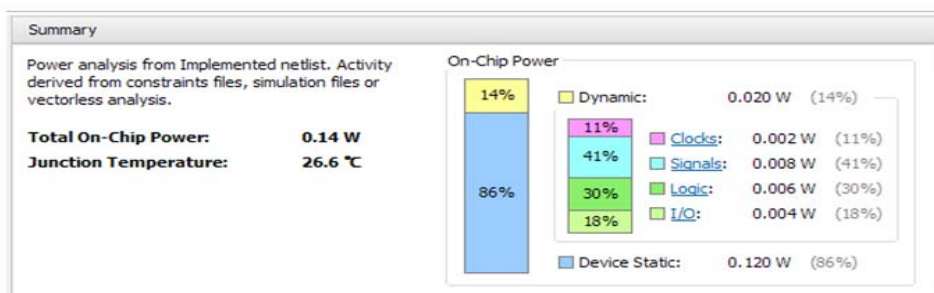


Fig. 22. Power analysis report for the transmitter side of the FRAT-FT-OFDM

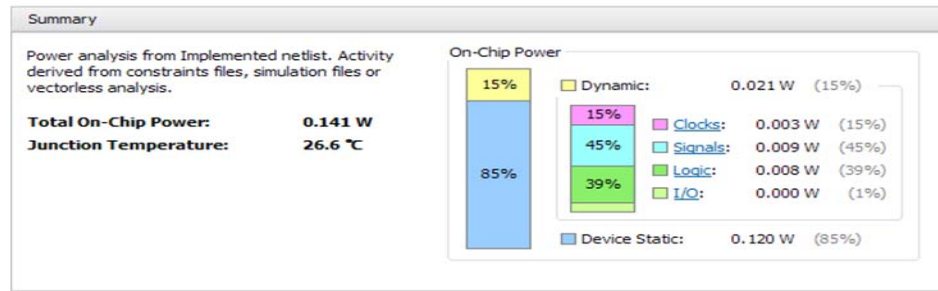


Fig. 23. Power analysis report for the receiver side of the FRAT-FT-OFDM

## V. CONCLUSION

In this article, FRAT-FT-OFDM transceiver was designed and successfully implemented on the FPGA platform using Xilinx System Generator (XSG). The XSG is compatible with MATLAB SIMULINK, making it possible to provide design parameters, bit- and cycle- true simulations, hardware co-simulation, and VHDL code generation. After a successful design for the FRAT-FT-OFDM, the system was verified with hardware co-simulation, and successfully implemented on Zynq (XC7Z020-1CLG484) evaluation board. The hardware simulation and synthesis results presented that the implemented system is properly working, efficient in terms of resource utilization, and supports real-time operations with little consumed power.

## REFERENCES

- [1] A. Almajdoob, "Design and FPGA Implementation of an OFDM System Based on 3GPP LTE Standard over Multipath Fading Channel," M. Eng. thesis, Concordia University Montréal, Québec, Canada, Nov. 2012.
- [2] S. S. Manure and U. L. Naik, "Design and performance analysis of DWT/FFT based OFDM systems," in Proc. ARTCom 2011, 2011, pp. 167-170.
- [3] A. Goldsmith, *Wireless communications*, Cambridge university press, 2005.
- [4] R. Dilmaghani and M. Ghavami, "Comparison between wavelet-based and Fourier-based multicarrier UWB systems," *IET Commun.*, vol. 2, pp. 353-358, 2008.
- [5] M. Veena and M. S. Swamy, "Performance analysis of DWT based OFDM over FFT based OFDM and implementing on FPGA," *International Journal of VLSI Design & Communication Systems*, vol. 2, pp. 119-130, 2011.
- [6] A. A. Sabri, & Z. Q. Abdulkareem, "Performance analysis of framelet based OFDM System under different channel conditions," *Engineering and Technology Journal*, vol. 30, pp. 795-810, 2012.
- [7] M. K. Gupta and S. Tiwari, "Performance evaluation of conventional and wavelet based OFDM system," *International Journal of Electronics and Communications (AEÜ)*, vol. 67, pp. 348-354, 2013.
- [8] S. Q. Hadi, P. Ehkan, M. S. Anuar, and S. A. Dawood, "Discrete framelet transform based OFDM system using N-FRAT mapping over Rayleigh fading channel," *Wireless Networks*, doi: 10.1007/s11276-016-1367-z, 2016.
- [9] K. Rajitha, U. R. Nelakuditi, V. N. Mandhala, and T.-h. Kim, "FPGA Implementation of Watermarking Scheme Using XSG," *International Journal of Security and Its Applications*, vol. 9, pp. 89-96, 2015.
- [10] B. Harikrishna and S. Ravi, "A Study of Architecture of FPGA Routing," *International Journal of Electronics and Communication Engineering (IJECE)*, vol. 2, pp. 69-76, 2013.
- [11] C.-L. Yu, J.-S. Kim, L. Deng, S. Kestur, V. Narayanan, and C. Chakrabarti, "FPGA architecture for 2D Discrete Fourier Transform based on 2D decomposition for large-sized data," *Journal of Signal Processing Systems*, vol. 64, pp. 109-122, 2011.
- [12] K. Fazel and S. Kaiser. *Multi-carrier and spread spectrum systems: from OFDM and MC-CDMA to LTE and WiMAX*. John Wiley & Sons, New York, USA, 2008.
- [13] O. M. Faraj, "Design and simulation of a proposed Radon based OFDM transceiver," in *International Workshop on Satellite and Space Communications*, 2006, pp. 18-22.