

**Military Technical College  
Kobry El-Kobbah,  
Cairo, Egypt**



**6<sup>th</sup> International Conference  
on Electrical Engineering  
ICEENG 2008**

## **FPGA implementation of OFDM physical layer**

*By*

Hesham M. El Maghraby\*    Alaa El Din Rohiem\*    Moataz M. Salah\*    Medhat Mokhtar\*

### **Abstract:**

Future wireless applications, targeting multimedia and high-speed internet access, require techniques to improve the link capacity and robustness. Current and future communication schemes tend to use Orthogonal Frequency Division Multiplexing (OFDM) systems in order to provide high baud rates and less intersymbol interference. The 802.11a, 802.11g, 802.16 and 802.20 standards all rely on the OFDM [1].

This paper presents an implementation of an OFDM physical layer on Field Programmable Gate Array (FPGA). The FPGA implementation allows flexible integration of extensions to improve the quality of service. The difficult task is the implementation of the FFT/IFFT module in the OFDM system. Two approaches of implementation also introduced. Simulation results and Hardware (HW) tests are presented, and a comparison between the two approaches of implementation is discussed.

### **Keywords:**

OFDM, FPGA, FFT.

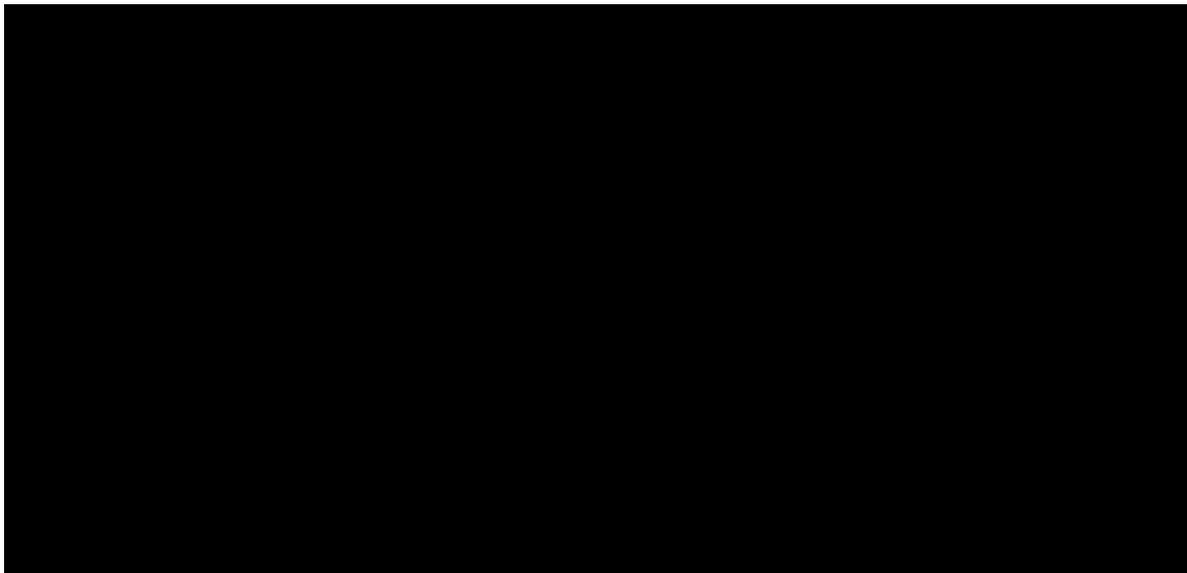
---

\* Egyptian Armed Forces

### 1. Introduction:

Orthogonal Frequency Division Multiplexing (OFDM) could be tracked to 1950's, but it had become very popular at these days, allowing high speeds at wireless communications. OFDM is a multicarrier transmission technique used in applications catering to both Wired and Wireless communications. The OFDM technique divides the frequency spectrum available into many closely spaced carriers, which are individually modulated by low-rate data streams. In this sense, OFDM is similar to FDMA. However, the difference lies in the fact that the carriers chosen in OFDM are much more closely spaced than in FDMA, thereby increasing its spectral usage efficiency. The orthogonality between the carriers is what facilitates the close spacing of carriers. The motivation for using OFDM techniques over TDMA techniques is twofold. First, TDMA limits the total number of users that can be sent efficiently over a channel. In addition, since the symbol rate of each channel is high, problems with multipath delay spread invariably occur.

OFDM could be considered either a modulation or multiplexing technique, and its hierarchy corresponds to the physical and medium access layer. Basic OFDM system main modules are a QAM or PSK modulator/demodulator, a serial to parallel / parallel to serial converter and these modules converts the serial data into parallel data to be processed by the Inverse Fast Fourier Transform (IFFT), and an IFFT/FFT module and a Guard interval insertion/ removal as shown in Figure 1



*Figure (1): A Transmission and reception scheme based on OFDM*

The narrow-sense OFDM with the Fourier base is simpler to implement than the conventional method. The transmit  $x(n)$  signal is given by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jnk 2\pi / N} \quad n = 0, \dots, N - 1 \quad (1)$$

The perfectly synchronized receiver just performs a Fourier analysis to recover the data symbols  $Y(k)$  from the signal:

$$Y(k) = \sum_{n=0}^{N-1} y(n) e^{-jnk 2\pi / N} \quad k = 0, \dots, N - 1 \quad (2)$$

A Fourier analysis is implemented by means of a fast Fourier transform (FFT). The stream of digitally modulated symbols  $X_{ik}$  is grouped in blocks of length  $N$ . Each block represents the coefficients of the orthogonal components of the symbol to be produced by the IFFT process. The Guard Interval Insertion block inserts a Cyclic prefix (CP) as a guard interval to preserve the orthogonality of the Subcarriers (SCs) and the independence of subsequent OFDM symbols, when the OFDM signal is transmitted over a multipath radio channel [2]. The CP is a copy of the last part of the OFDM symbol. The IFFT module processes  $N_{\text{FFT}}$  coefficients ( $N$  data coefficients + zero padding). To give a concrete example, in the European Digital Audio Broadcasting (DAB) and in the Digital Video Broadcasting-Terrestrial (DVB-T) system, a FFT with  $N_{\text{FFT}} = 2048$  is used (among other FFT modes), and the number of modulated carriers  $N$  is of the order 1500 and 1700, respectively [3].

It worth noting that the truncation of the computed values due to the limited resolution of the computation process of IFFT/FFT may cause a serious approximation problem and a correction process must be considered.

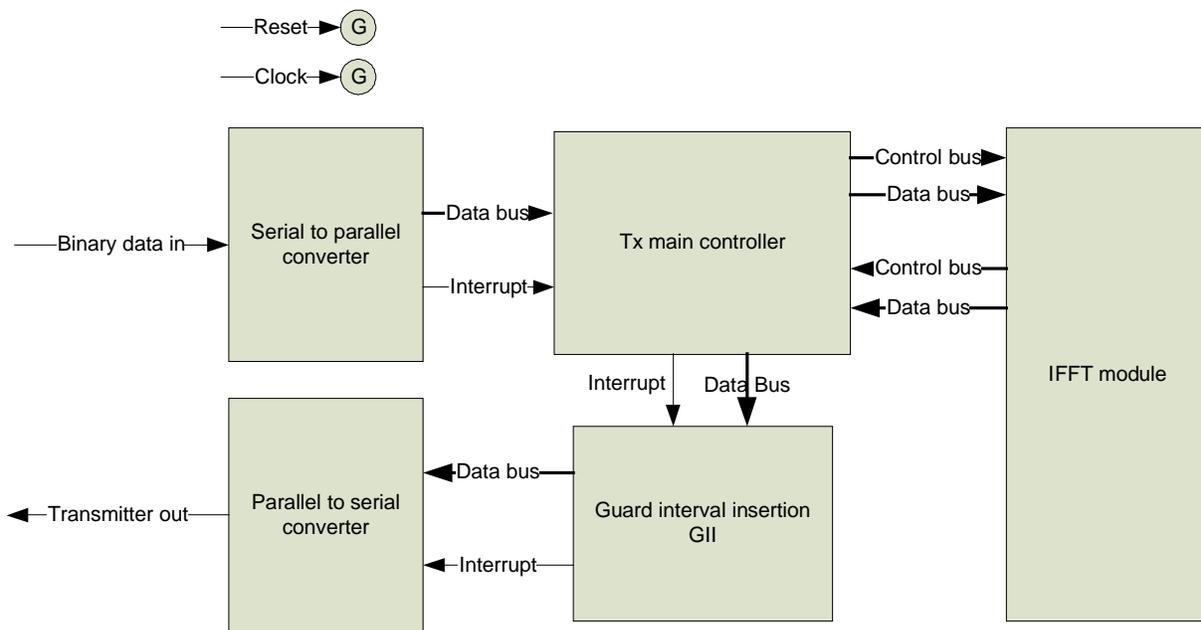
However, the parameters of an OFDM system is governed by the application environment that is specified by the number of multipath, the relative phase of each, and the Doppler shift. The selected parameters in this work are for the demonstration purpose and not dedicated for a specific channel. The implementation using Field Programmable Gate Array (FPGA) gives the privilege of modifying these parameters to suit different systems requirements and also flexibility for reconfiguration and introducing several ways of interfacing with the system.

The paper is organized as follows. In section 2, a FPGA-based design of an OFDM system is introduced, and the approaches for solving the problems associated with the implementation are discussed. In section 3, the simulation results are introduced to validate the system design. The hardware implementation, test bench, and the testing procedure are included in section 4. Section 5 is the conclusion.

**2. FPGA Design Architecture:**

The design consists of two main units; the transmitter and the receiver. All the designed modules are described in VHDL using the *FPGA advantage 7.2*.

The transmitter Unit, as shown in Figure (2), consists of five main modules which are serial to parallel converter, Tx Main controller, IFFT, Guard interval insertion (GII), and parallel to serial converter.



**Figure (2): OFDM Transmitter unit**

In this design, the symbol is considered of length 16 with 8-bit resolution. Hence the data bus from the Serial to parallel converter of width 128 bits. With proper clock selection of each module, the data streaming between modules is simply controlled by an interrupt signal. The Tx main controller module is the core of the unit that controls the data streaming. It passes the data from the serial to parallel converter module to the IFFT module, and the transformed results to the GII module. The main function of the GII module is to insert the cyclic prefix (CP) in the output data frame and passing them to the parallel to serial converter module. The length of the CP is selected from 10 % to 25 % of the total frame length [3]. It worth notifying that the length of the GI should exceed the maximum excess delay of the multipath propagation channel as shown in Figure (3).

For alleviating the computation error that may arise due to the limit resolution of the

IFFT process, two approaches are tested in this design. In the First approach, the block of data is passed to both the real and imaginary ports of the IFFT module. In the receiver side, the real and imaginary components of the received signal are compared, and the error is estimated and corrected as described later. The second approach is based on splitting the data into real and imaginary parts, and sending them to the corresponding ports of the IFFT in a symmetrical form around the center. The correction technique in the receiver unit is carried by comparing the two halves of both for estimating the correct data.

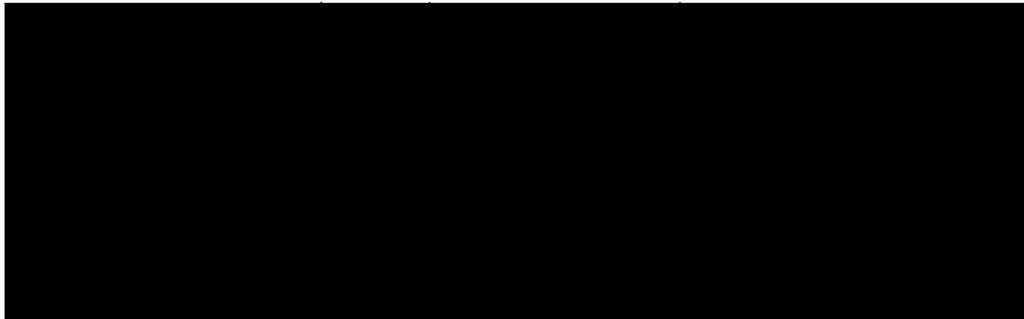


Figure (3): The CP insertion in the Data frame

The IFFT/FFT module is the Xilinx IP core "Fast Fourier transforms v3.2" [4]. The control signals of this module are generated by the Tx main controller module in the transmitter or the Rx main controller module in the receiver.

On the other side, the receiver unit carries the reverse operations in addition to the correction process in the compensator module as shown in Figure (4).

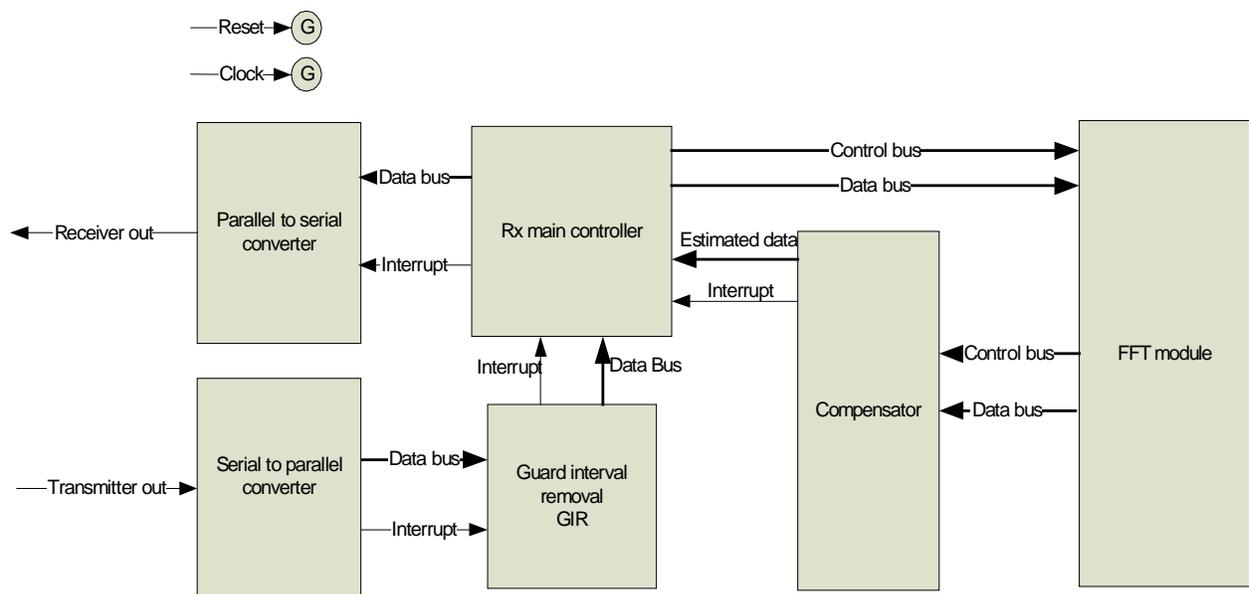


Figure (4): The Receiver module design on FPGA

The compensator module performs the process of the scheme illustrated below. It is valid for the two approaches of transmitted data duplication disused above. It detects the error by comparing the data that have to be similar, and estimated the correct one.

Pseudo code is introduced to illustrate the correction scheme, assuming the two values calculated from the FFT are in the form “Xxx” and “Aaa”. The two values X and A have to be similar, and identical to the transmitted one. The least significant numbers of “Xxx” and “Aaa” (xx and aa) are produced from the IFFT and FFT, and must be truncated in the correction process. This process is governed by the following correction scheme.

```

If (X = A) then
{
  If ((xx + aa) > threshold) then
    Output = X + 1 OR Output = A + 1
  Else
    Output = X OR Output = A
}
Else
{
  If (X > A) then
    Output = X
  Else (X < A) then
    Output = A
}
End of the code

```

### **3. Simulation results:**

The simulation tool, used to verify the validity of the design, is the *ModelSim v6.1c* which is a downstream tool in the *FPGA advantage 7.2* package. The simulation result of the Transmitter unit is shown in Figure (5). The data with word length equals 8 is received in a serial form on the port "sin", and then converted into parallel form on the port "data\_transmitted\_reg". The figure shows the main 2 control signals produced by the Tx main controller module; "start", and "fwd\_inv". These signals synchronize the transformation process and determine the forward or reverse operation as demonstrated in [4]. The real and imaginary coefficients of the data flow to the IFFT module through the ports "xn\_re" and "xn\_im" respectively, and associated with its index. In 53 clock period from the last input entrance to finish its calculation, the data is received at the output ports "xk\_re\_core" and "xk\_im\_core". The Tx main controller stores the output

data in two registers “data\_re\_reg” and “data\_im\_reg”, and passes them synchronized with an interrupt signal to the GII module to add the Cyclic prefix. The output of the GII module is send serially to the receiver unit on "sout\_tx" port.

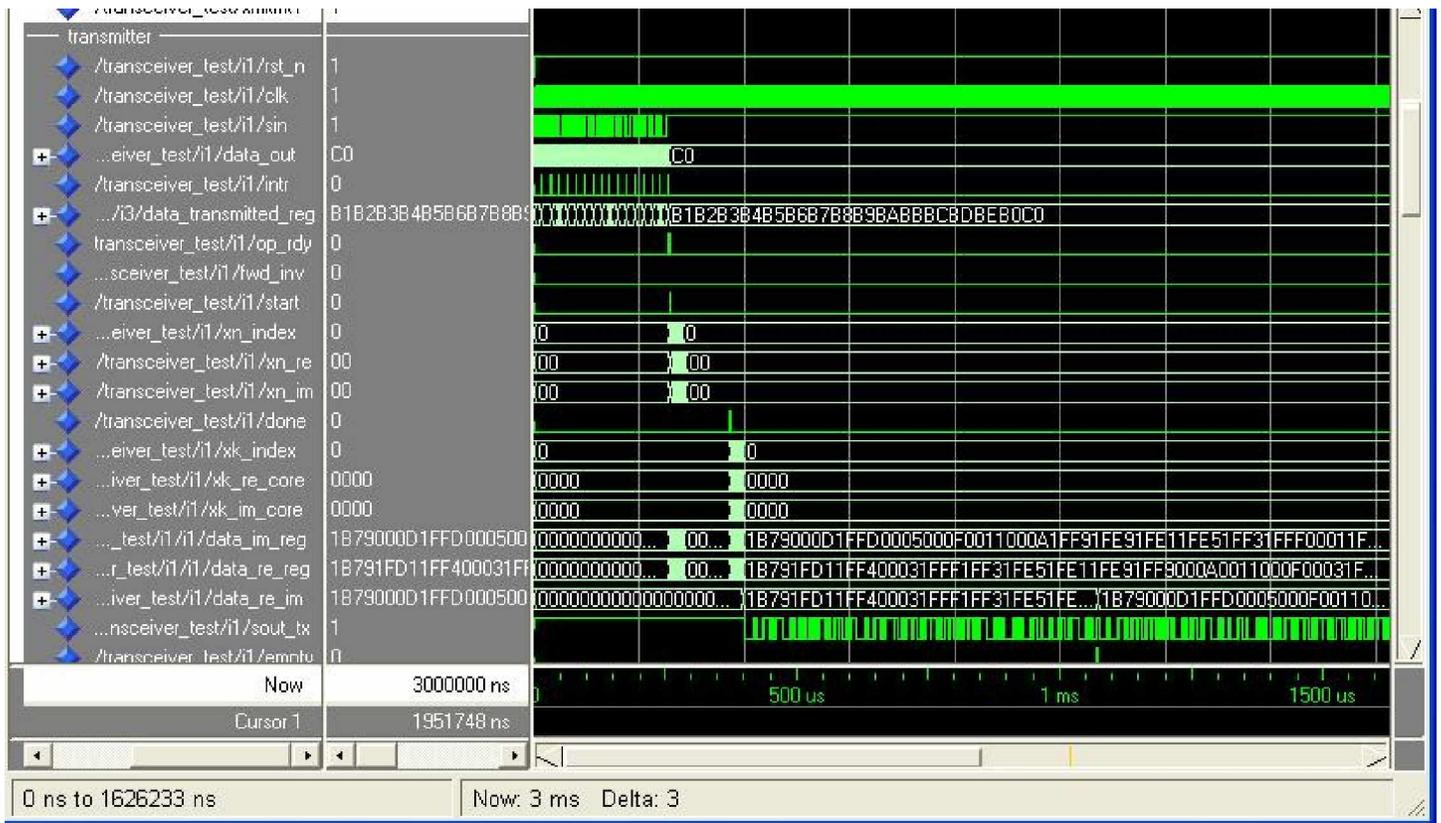


Figure (5): Timing diagram for the Transmitter unit signals

In the Receiver unit, the data is received serially on port "sin" as shown in Figure (6), then the data is stored in a register and passed to the GIR module to remove the Cyclic prefix. The Rx main controller module control the FFT module [4], after finishing calculations (which takes also 53 clock period) the compensator module receives the output data from the FFT module and begin to compare and correct the errors if found according to the correction scheme described in section 2, and as we mentioned the aim of the two approaches is to repeat the data to allow the correction method to take place in the receiver.

The difference between the two approaches is in the compensator module in the receiver unit, if we are using the first approach then the compensator can run the correction procedure while receiving the data from the FFT module.

On the other hand in the second approach we will have to wait till the FFT module finish all the calculations and all the data is ready and then begin applying the correction to the output frame which will lead to a little delay in the final estimated data and this

delay is from 99 to 131 clock period.

At the final output of the receiver module we can see in Figure (6) that the estimated data in "data\_estimated\_reg" is equal to the transmitted data in "data\_transmitted\_reg".

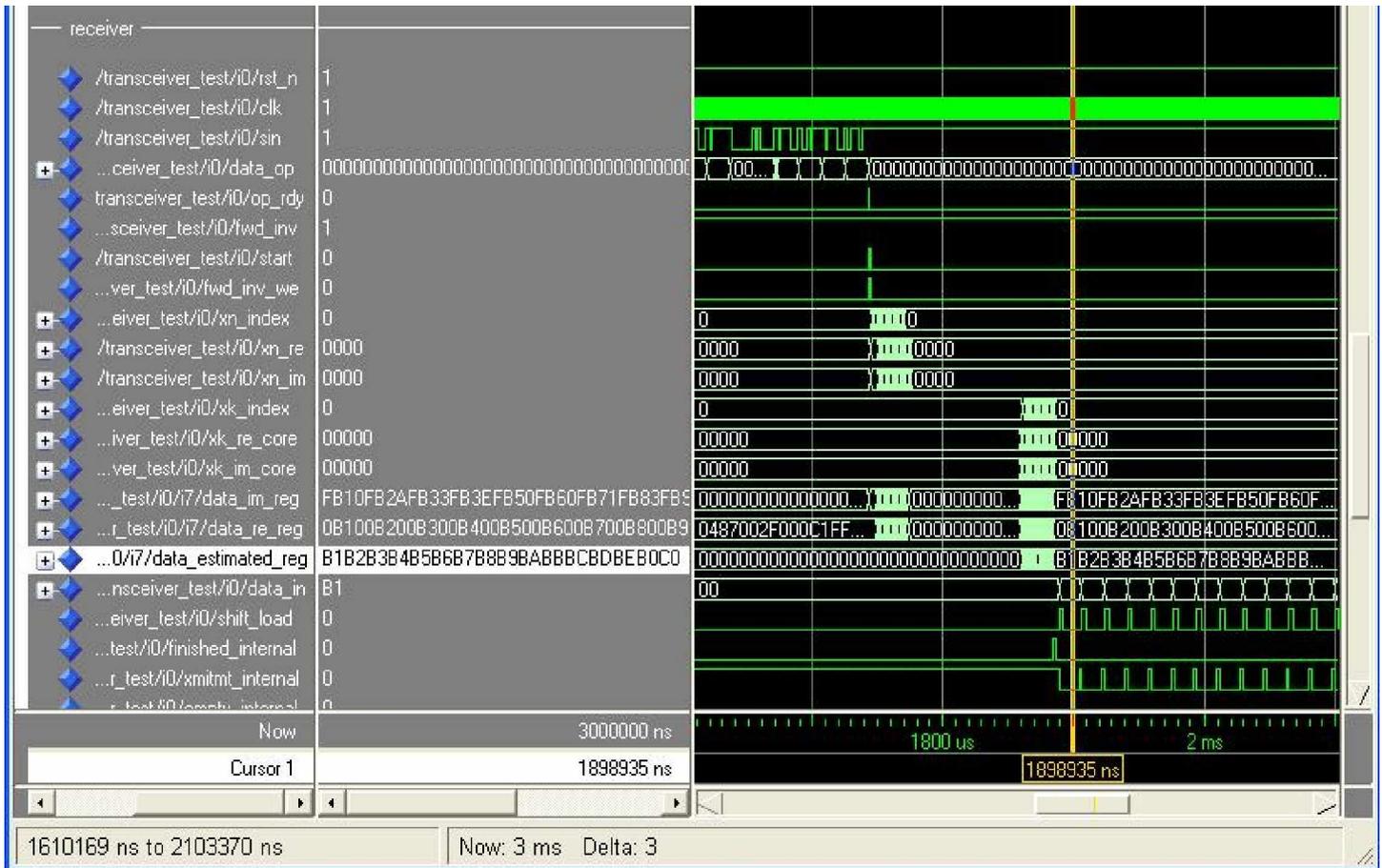


Figure (6): Timing diagram for the Receiver unit signals

#### 4. Design Download and tests:

The download tool is Xilinx software "ISE 9.1i" and the target device is Xilinx Spartan3 xc3s1000-5ft256 chip. The testbench is as shown in figure (7) where the computer (PC1) is the data source, while the computer (PC2) is the data destination. The two computers are connected to the FPGA board through the RS-232 serial port. It was necessary to develop the Universal Asynchronous Receiver Transmitter (UART) protocol to interface with computers in the transmitter and the receiver units. The two boards are connected serially through wire cable. In the following tables, the device utilization for both the Transmitter and the Receiver units are introduced respectively. Through the testing procedure, a block of data is sent from PC1 to the Transmitter unit and received correctly on the PC2 through the Receiver unit. This step is carried many

times for verifying the validity of the applied algorithms and the associated hardware implementation.



*Figure (7): Testing Setup for the design*

*Table (1): Area Report for the Transmitter Unit on xc3s1000-5ft256*

<b>Logic Utilization</b>	<b>Used</b>	<b>Available</b>	<b>Utilization</b>
Number of Slice Flip Flops	2,474	15,360	16%
Number of 4 input Lookup Tables(LUTs)	3,624	15,360	23%
<b>Logic Distribution</b>			
Number of occupied Slices	2,306	7,680	30%
Number of Slices containing only related logic	2,306	2,306	100%
<b>Total Number of 4 input LUTs</b>	<b>3,973</b>	<b>15,360</b>	<b>25%</b>
Number used as logic	3,624		
Number used as a route-thru	86		
Number used for Dual Port RAMs	104		
Number used as Shift registers	159		
Number of bonded IOs	15	173	8%
IOB Flip Flops	1		
Number of MULT18X18s	4	24	16%
Number of GCLKs	3	8	37%
<b>Total equivalent gate count for design</b>	<b>76,965</b>		
Additional JTAG gate count for IOs	720		

**Table (2): Area Report for the Receiver Unit on xc3s1000-5ft256**

<b>Logic Utilization</b>	<b>Used</b>	<b>Available</b>	<b>Utilization</b>
Number of Slice Flip Flops	2,752	15,360	17%
Number of 4 input LUTs	3,946	15,360	25%
<b>Logic Distribution</b>			
Number of occupied Slices	2,652	7,680	34%
Number of Slices containing only related logic	2,652	2,652	100%
<b>Total Number of 4 input LUTs</b>	<b>4,367</b>	<b>15,360</b>	<b>28%</b>
Number used as logic	3,946		
Number used as a route-thru	76		
Number used for Dual Port RAMs	128		
Number used as Shift registers	217		
Number of bonded IOs	12	173	6%
IOB Flip Flops	2		
Number of MULT18X18s	4	24	16%
Number of GCLKs	3	8	37%
<b>Total equivalent gate count for design</b>	<b>86,164</b>		
Additional JTAG gate count for IOs	576		

### **5. Conclusions:**

The paper presents an implementation of an OFDM physical layer on FPGA. The architecture presents two techniques for solving the approximation problem in the IFFT/FFT operations. The validity of the proposed solution is tested by experiments and simulations. It shows that the two proposed techniques are successfully alleviated the approximation problem with a slight delay in the second method. The design download results and reports are introduced.

### **References:**

- [1] Ron Olexa, "Implementing 802.11, 802.16, and 802.20 Wireless Networks", Elsevier Inc., 2005.
- [2] Ramjee Prasad, "OFDM for Wireless Communications Systems", Artech House Inc., London, 2004.
- [3] Henrik Schulze and Christian L'uders, "Theory and Applications of OFDM and CDMA", Meschede, Germany, 2005.
- [4] Xilinx IP core Data sheet, "Fast Fourier Transform v3.2", 2006.