
Design and Implementation of Reconfigurable Modulator Using FPGA for Cognitive Radio System

¹Oveek Chatterjee, ²Dr. Rajeshree Raut, ³Ranjit K.Sawant

¹Ex. Student, Dept. of Electronics Engineering, RCOEM, Nagpur, India

²Dept. of Electronics & Design Technology, RCOEM, Nagpur, India

³Dept. of Electronics, Govt. Polytechnic, Hingoli, India

Email: ¹ovkchat5@gmail.com, ²rautrd@rknec.edu, ³ranjitsawant@yahoo.com

Received: 20th September 2018, Accepted: 11th October 2018, Published: 31st October 2018

Abstract

With the increase of smartphones and laptops, number of users using communication systems is increasing day by day exponentially. The frequency spectrum required for these systems is limited. Thus in order to increase the proper utilization of the spectrum, cognitive radio technology is being developed. In this system, spectrum is shared between the primary (licensed) users and secondary (unlicensed) users. Secondary user uses a channel only when it is not being used by the primary user. Also different channels require different modulation techniques based on the noise present in their respective channel. Thus, a secondary user transmitter should be able to cope up with this requirement. In this paper an assumption has been made on the channel selection by frequency reconfiguration, so the paper focuses on the reconfiguration in digital modulation scheme. A way of designing and implementing reconfigurable modulator system using FPGA is discussed in the paper. Different types of noise present in the channel are detected with the help of Fourier transform and decision is taken to dynamically switch between different digital modulators.

Keywords

Cognitive Radio, FFT, BASK, BFSK, BPSK, Microblaze Processor, FPGA

Introduction

We cannot increase the spectrum beyond a certain range because of various environmental and health hazards. Thus, to increase the proper utilization of the available spectrum, dynamic allocation of the spectrum was proposed [1] in contrast to the most widely used static spectrum allocation method. In dynamic spectrum allocation a channel is allocated to a secondary (unlicensed) user when a primary user is not using the channel with the help of a cognitive radio system, proposed by Joseph Mitola [2].

The cognitive radio cycle consists of four main stages [3]:

1. Sensing: It is monitoring the available spectrum and scanning for vacant channels.
2. Analysis: It is analysis of the available vacant channels and reaching a conclusion of allocating one of the vacant channels to the secondary user based of the properties and characteristics of the secondary user and the channel.
3. Reasoning: In this stage of the cognitive cycle, the ways of adapting to the characteristics of the channel are sorted out and the required parameters are set.
4. Adapting: Finally at this stage the secondary user dynamically switches to the new parameters as required by the channel.

In this paper, a novel design of a reconfigurable modulator for a cognitive radio system is proposed which is based on the noise present in the channel, detected in the analysis phase of the cognitive cycle. This paper is organized in 6 sections. Section 1 is general introduction to the work done during this paper. Section 2 deals with design methodology, its implementation and results of the noise detecting signal. In the section 3, the modulation techniques have been simulated in Matlab under different noise inputs to check their performance. The section 3 also deals with the design and implementation of the modulation techniques using Verilog HDL coding and the proposed modulator's architecture is formed by integrating them together to form a reconfigurable modulator system with real time switching capability. In the section 4 of the paper, both the noise detecting system and the reconfigurable modulator are included in a Microblaze processor to develop the desired system, the system setup is also shown in this section. Section 5 shows the expected and the observed output results of the proposed system under given input and noise conditions. The future work is given in section 6 of the paper.

Noise Detection System

This paper proposes a way to characterize the noise present in given channel of available spectrum by computing the DIT FFT [5] calculation of the signals (noise) present in the channel in the analyzing stage of the cognitive cycle after the cognitive radio has reconfigured. A Microblaze system is built using XPS environment

of the EDK tool provided by Xilinx Inc. A multiplier is used as a custom peripheral in the system for computing the FFT algorithm. The system is shown in Fig. 1. This system is then exported to SDK environment. In SDK a C language program for computing FFT is written and the output obtained from Microblaze is printed at the console window of SDK using UART available in the designed Microblaze system.

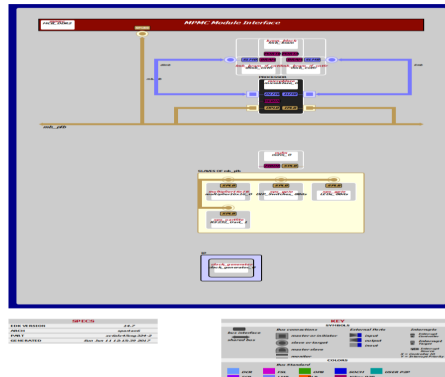


Fig. 1: Microblaze System with Signed Multiplier

FFT obtained by the Designed Microblaze System

To test the designed Microblaze system a composite signal containing known frequency is composed in Matlab. Frequency of $f = 1\text{MHz}$ is used to model this signal. This signal is stored in a variable “x” and 1024 is the length of this signal. The $\text{fi}(x,1,16,15)$ is the function used for converting the “x” to 16 bit fixed point notation. Here 16 bits is the word length of which first bit is the sign bit and remaining 15 bits are the fractional length. Integer representation of this variable is obtained using the function “int”. The same notation is used for imaginary signal (which will be zero for real value signal) and twiddle factors. The signal generated using the above process is used as input in the Microblaze system. The theoretical frequency spectrum is obtained in Matlab using the inbuilt Matlab “FFT” function, shown in the Fig. 2 (a). The twiddle factors needed for the fixed point FFT calculation in Matlab and the Microblaze system are generated using Matlab with the help of a function which produces the required values of the twiddle factors. The fixed point FFT of the given 1MHz signal is shown in the Fig. 2 (b)

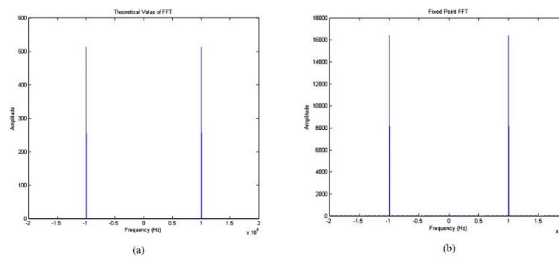


Fig: 2 (a) Frequency Spectrum of 1MHz Signal Using Theoretical FFT Calculation. (b) Frequency Spectrum of 1MHz Signal Using Fixed Point Calculations.

The amplitude scale for fixed point notation is high because of the 16 bit fixed point format used for the calculation of the FFT.

The output obtained from Microblaze is shown in Fig. 2. The Decimation in Time radix-2 FFT algorithm is used for computing the FFT.

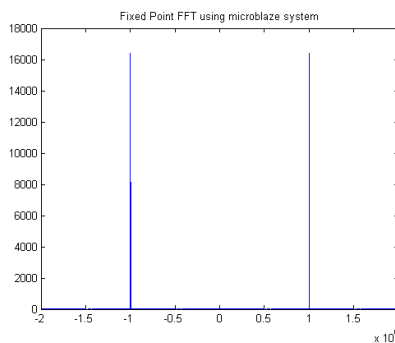


Fig. 3: Frequency Spectrum of 1MHz Signal Using Fixed Point Calculations by Microblaze System

It can be seen that the shape of the spectrum matches with the theoretical shape shown in Fig. 2 (a). Values of the amplitudes shown in Fig. 2 (a) and Fig. 3 do not match since different number representations are used in these cases, one is double notation and the other is fixed-point notation. Although the amplitude levels of Fig. 2(b) and Fig. 3 do match as for both the calculations, fixed point notations are used. The system has been tested for signal with multiple frequencies too. The theoretical FFT of a multiple frequency signal shown in Fig. 4(a) and the output obtained from Microblaze is shown in Fig. 4(b)

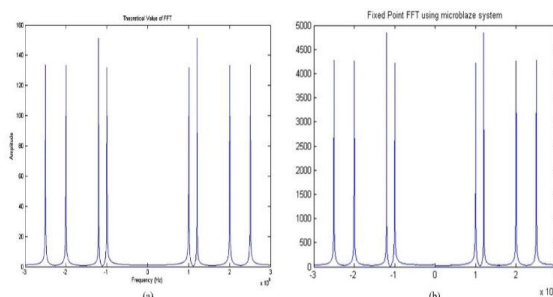


Fig. 4 (a) Frequency Spectrum of Signal with Multiple Frequency Using Theoretical FFT Calculation (b) Frequency Spectrum of Signal with Multiple Frequency using Fixed Point Calculations by Microblaze System

The number of complex multiplications required for execution of N point FFT is $\frac{N}{2} \log_2 N$. The number of real multiplications needed to execute real and imaginary part separately is $4 \times \frac{N}{2} \log_2 N$. Thus for 1024 point FFT 20,480 and for 8192 point FFT 2,12,992 multiplications are required. This algorithm can be applied to check the noise present in the channel before the start of the communication by the secondary user so as to characterize the type of noise present in the system and select appropriate modulation technique for the given channel by using peak search method or zero crossing method which gives the result of frequencies at which the noise is present in the channel.

Modulator Systems

The aim of this paper is to design a reconfigurable modulator system for cognitive radio system. Thus, the types of modulators used in this paper are very basic and are included just for testing purpose to check the reconfigurability of the system. Advancements can be done in this project by designing practically used modulators for the system.

The modulators used in the system are first designed in Matlab to test their performance by introducing noise signal in the system and a comparison has been made between the three modulator systems and the conditions under which the reconfiguration has to be made is deduced. After analysing the systems they have been realized into hardware using Verilog HDL and FPGA.

3.1 BASK Modulator [6]

The BASK modulator with and without noise values at given carrier frequency is given in Table.1.

BASK system	Bit rate (br) (bits/sec)	Carrier frequency	Noise frequency
Without Noise	106	1 MHz	-
With Noise	106	1 MHz	5 MHz

Table.1. BASK Modulator Input Values

The wave forms obtained by using the values given in Table.1 are given by Fig. 5 (a) & 5 (b).

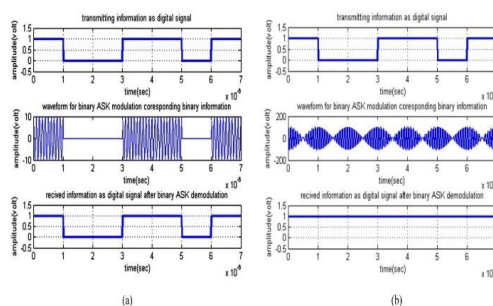


Fig. 5 (a) BASK Modulated and Demodulated Signal Without Noise Signal. (b) BASK Modulated and Demodulated Signal with Noise Signal

As we know that amplitude of a signal is most affected by any noise signal and is also evident from the Fig. 5 (b) that when a noise signal is introduced in a BASK modulator, the performance of the modulator is hampered and the demodulator may read input data '0' as '1' or vice versa leading to erroneous output, which is not desired.

BFSK Modulator [7]

The BFSK modulator system is designed in Matlab for the following three conditions respectively given in Table.2.

1. Without any noise induced in the system.
2. With noise as induced in the BASK modulator system.
3. With noise input affecting the output of the BFSK system

BFSK system		Bit rate (br)	Carrier frequency(f)	Noise
Without Noise	-	1 MHz	1 MHz,8 MHz	-
With Noise	Noise 1	1 MHz	1 MHz,8 MHz	5 MHz
	Noise 2	1 MHz	1 MHz,8 MHz	1 MHz

Table.2: BFSK Modulator Input Values

The wave forms obtained by using the values from Table.2 is shown in Fig. 6, Fig. 7 (a) & 7 (b).

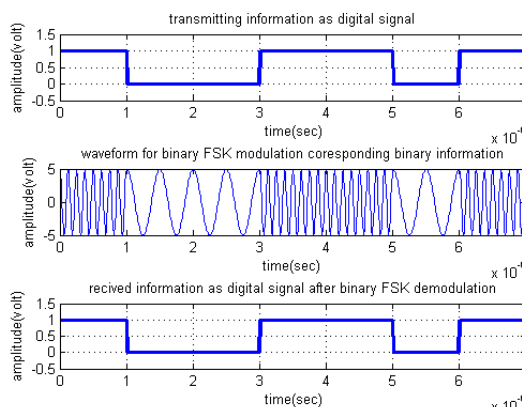


Fig. 6: BFSK Modulated and Demodulated Signal without Noise Signal

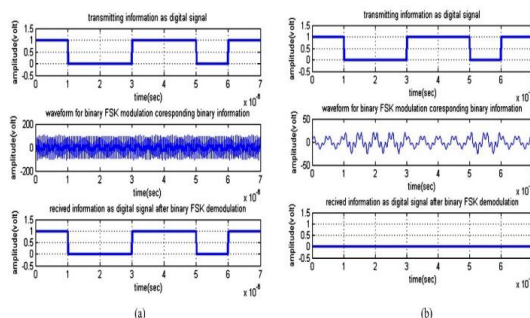


Fig. 7 (a) BFSK Modulated and Demodulated Signal with the Same Noise as Induced in the BASK Modulator. (b) BFSK Modulated and Demodulated Signal with Noise Affecting the Performance of the BFSK Modulator

It can be seen from the Fig. 7 (a) and Fig. 7 (b) that for a noise signal at any frequency other than that of the carrier waves used in the BFSK modulator, the performance of the BFSK modulator is not hampered and the demodulator correctly reads input data '0' and '1'.

Although, when the frequency of the noise signal is near to the frequencies of carriers, the BFSK system's performance is hampered and erroneous output is generated as shown in the Fig. 7 (b)

BPSK Modulator System [7]

The BPSK modulator system is designed in Matlab for the following three conditions respectively the input values are shown in Table.3

1. Without any noise induced in the system.
2. With noise as induced in the BASK modulator system.

3. With noise as induced in the BFSK modulator system.

BPSK system		Bit rate (br)	Carrier frequency(f)	Noise
Without Noise	-	1 MHz	1 MHz	-
With Noise	Noise 1	1 MHz	1 MHz	5 MHz
	Noise 2	1 MHz	1 MHz	1 MHz

Table.3. BPSK Modulator Input Values

The wave forms obtained by using the code with input values as given in Table.3 given by Fig. 8, Fig. 9 (a) and Fig. 9 (b) respectively.

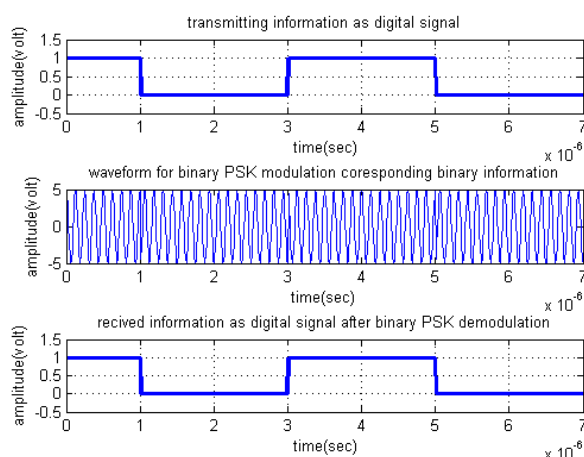


Fig. 8: BPSK Modulated and Demodulated Signal Without Noise Signal

It can be seen from the Fig. 9 (a) and Fig. 9 (b) that for any noise signal that affects the performance of BASK and BFSK modulator the performance of the BPSK modulator and the demodulator doesn't get affected and correctly reads input data '0' and '1'.

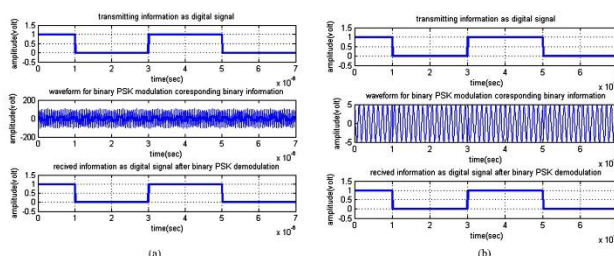


Fig.9 (a) BPSK Modulated and Demodulated Signal with the Same Noise as Induced in the BASK Modulator. (b) BPSK Modulated and Demodulated Signal with the Same Noise as Induced in the BFSK Modulator

Proposed Modulator System

The BASK, BFSK and BPSK modulators are then realized in hardware using Verilog HDL code with the same logic as shown in the Matlab codes and the output of the system is then obtained on the console window of the Xilinx ISE simulator by performing post route simulation of the designed system. The carrier signal for the simulation is generated in the MATLAB and then stored in memory array of the HDL code after converting them into fixed point notation. The carrier waves are stored in the code using an "initial block".

The proposed architecture is obtained by combining the architecture and characteristics of the three modulation techniques given above and switching between these three techniques in run time based on a control signal. As mentioned earlier the control system for our reconfigurable modulator will be based on the characteristics of the noise present in the selected channel.

The following conclusions are drawn from the section 3.1, 3.2 and 3.3 for reconfiguring the modulator.

1. As BASK is the simplest modulator, it is assumed that when there is no noise present in the channel on which the secondary user is going to transmit data, BASK modulator should be used.
2. If any type of noise signal is detected in the channel by the noise detecting system, the modulator should reconfigure itself from the BASK modulator to any of the other two modulators.
3. If the noise present in the channel is around the frequency range of the carrier signals used for the BFSK system then the modulation technique should be reconfigured to BPSK.

The phase component of a signal is least affected by the noise and the 180° phase shift between the two carrier waves used in BPSK modulation provides more immunity to noise than the other two modulator systems.

The architecture of the reconfigurable modulator system as proposed in the Fig. 10 (a) is implemented on DigilentAtlys Spartan 6 (XC6SLX45) FPGA development board using mix modelling style of coding. The Register Transfer Logic (RTL) schematic of the architecture generated by the Xilinx ISE software is shown in the Fig. 10 (b).

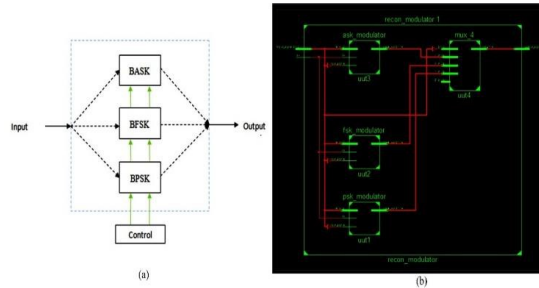


Fig. 10 (a) Block Diagram of Proposed Reconfigurable Modulator. (b) RTL Schematic of Reconfigurable Modulator

Logic Resource Utilization of the Proposed Architecture

The Logic resources utilized by the reconfigurable modulator system proposed in this paper is shown in the Fig. 11

Number of 8-bit Buses	1	16	60%
Number of 4-bit Buses	5	116	10%
Number of 2-bit Buses	40	518	55%
Number of 1-bit Buses	83	405	57%
Number of 3-bit Buses	28	3388	10%
Number of 1-bit Registers	52	242	0%
Number of 4-bit Registers			
Number of 2-bit Registers			
Number of 1-bit Registers			
(continued on next page)			

Fig. 11: Resource Utilization of Reconfigurable Modulator

Timing Constraints of the Proposed Architecture

The results of the timing analysis done by the Xilinx ISE software for the reconfigurable modulator on XC6SLX45 device with speed grade -2 is as shown in the Fig. 12

```

Timing Summary:
-----
Speed Grade: -2

Minimum period: 6.524ns (Maximum Frequency: 153.271MHz)
Minimum input arrival time before clock: 10.401ns
Maximum output required time after clock: 7.474ns
Maximum combinational path delay: 8.061ns
    
```

Fig. 12: Timing Constraints of Reconfigurable Modulator

The timing analysis shows that the critical path for the system is 8.061 ns and thus the proposed architecture can work on almost all of the present FPGA boards.

Reconfigurable Modulator Design Environment

After doing the analysis of the system in Xilinx ISE, a Microblaze system is designed using XPS environment of the EDK tool. The Verilog code developed in the ISE is then used as a custom peripheral along with the multiplier custom IP stated above in section 2. The system is shown in Fig. 13. This system is then exported to SDK environment, in the SDK the C language program for computing FFT written before is used for generating a control signal for the Modulator system. This control signal is then mapped to the control input of the modulator to control the real time reconfiguration of the modulator with the help of handshaking signals. The output obtained from Microblaze is printed at the console window of SDK using UART available in the designed Microblaze system.

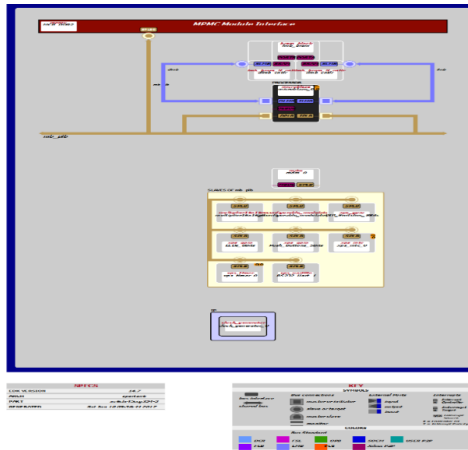


Fig. 13: Microblaze System with Reconfigurable Modulator as Custom IP

1. Simulation Results

The carrier waves used in the simulation have frequencies as per Table 4.

Modulation Technique	Carrier wave Frequency
BASK	f =1MHz
BFSK	f1=1MHz, f2=2MHz
BPSK	f=1MHz

Table 4: Modulation Techniques and The Frequencies Used

The simulation conditions are given below:

- At first the system is simulated for no noise condition. That is the input to the noise detecting system is given a zero value signal.
- Then a noise signal of any random frequency and amplitude is given as an input to the FFT calculation algorithm which detects the noise present in the channel.
- Then a noise signal having frequency close to the frequency used for BFSK signal is given to the noise detection unit.

For every condition the systems output is measured for 4.5 μ s.

For an ideal reconfigurable modulator with the functioning as mentioned in the previous sections, the output for the stated conditions should be as follows:

- For no noise condition the modulator should perform as BASK modulator.
- The modulator should then reconfigure itself to BFSK modulator for the second condition.
- As soon as the noise detecting system detects a noise of frequency close to the frequency used for the BFSK system, the modulator should then work as the BPSK modulator.

The input signal and the expected and actual output signal waveform is shown in Fig. 14

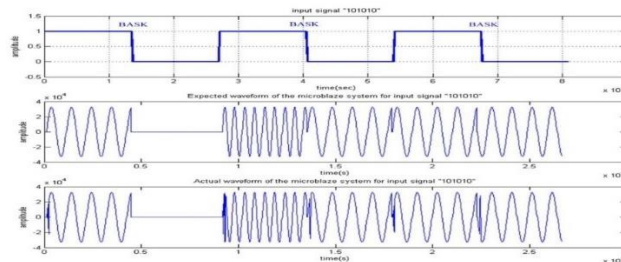


Fig. 14: Expected and Practical Output Waveform of Reconfigurable Modulator for Input Signal “101010”

It can be seen from Fig. 14 that the simulated results are near about similar to the expected results. The actual output signal have glitches in it because whenever a particular modulation technique is configured, the C code requires time to initialize the modulator. During this period the C-code written in SDK and the HDL code written for the modulators take time to get into synchronization. Thus, the glitches are only present in the starting of the output for each modulation technique.

Conclusion & Future Scope

It can be concluded from the above sections that both, the noise detecting system and the designed modulator architecture works as expected in detecting any noise present in the particular channel and performed required reconfiguration in the modulation system.

The future scope of the system is as follows:

- The input signals for both the noise detecting system and the reconfigurable modulator can be made dynamic by interfacing external ADC to the system and thus the input and the control signals required can be obtained by function generators or any other signal source as per the requirement. The noise detecting system can be made more robust by measuring the Euclidean distance in the constellation diagram.
- The basic modulator systems can be replaced with widely used modulator systems such as M-PSK and QAM modulation techniques to fit the system for real world application.
- A timer can be included in the design and used as interrupt to calculate the clock cycles required for the C-code used in the system to further optimize the performance of the system.
- Moreover, in order to optimize the resources required the different modulator sub blocks can be implemented using a single sub block with the help of partial reconfiguration provided by the Xilinx partial reconfiguration tool [8].

References

1. Mohammed Hawa, Khalid A. Darabkh, Loay D. Khalaf, Jamal S. Rahhal, "Dynamic resource allocation using load estimation in distributed cognitive radio systems", AEU - International Journal of Electronics and Communications, Volume 69, Issue 12, December 2015
2. J. Mitola III and G.Q. Maguire, Jr., "Cognitive Radio: Making Software Radios More Personal", IEEE Personal Communication, Vol. 6, No. 4, Aug. 1999, pp.13–18.
3. Sumin D. Joseph, S. Manoj, Chetan Waghmare, K. Nandakumar & Ashwin Kothari, "UWB Sensing Antenna, Reconfigurable Transceiver and Reconfigurable Antenna Based Cognitive Radio Test Bed", Wireless Personal Communications, Vol. 80, No. 3, Feb. 2015, ISSN 0929-6212.
4. A. I. Mecwan, N. P. Gajjar, "Implementation of Software Defined Radio on FPGA", IEEE International Conference on Current Trends in Technology, Ahmadabad, December 2011.
5. J.G. Proakis, D.G. Manolakis, Digital Signal Processing– Principles, Algorithms and Applications, Third Edition, NJ, USA, Prentice-Hall International, 1996.
6. C. Erdoğan, I. Myderrizi, and S. Minaei, FPGA Implementation of BASK-BFSK-BPSK Digital Modulators, IEEE magazine 2012.
7. J. S. Chitode, Digital Communications, Technical publications, first edition: 2007-2008.
8. Xilinx Inc., Xilinx Partial Reconfiguration User Guide, UG702 (v14.1) April 24, 2012.