Physical Layer Implementation Of Orthogonal Frequency Division Multiplexing For Software Defined Radio On FPGA

Mrs. Kokila. K. S PG Student [#1], **Mr. Anandaraju. M. B** Prof & HOD [#2]

[#] *Department of Electronics and Communication Engineering,*
[#] *B.G.S. Institute of Technology, B.G.Nagar, Mandya-571448, Karnataka, INDIA*

Abstract

Communication is one of the important aspects of life. With the advancement in age and its growing demands, there has been rapid growth in the field of communications. Signals, which were initially sent in the analog domain, are being sent more and more in the digital domain these days. For better transmission, even single – carrier waves are being replaced by multi – carriers. Multi – carrier systems like CDMA and OFDM are now – a – days being implemented commonly. In the OFDM system, orthogonally placed sub – carriers are used to carry the data from the transmitter end to the receiver end. Presence of guard band in this system deals with the problem of ISI and noise is minimized by larger number of sub – carriers.

The aim of the project is to discusses the implementation of a Software Defined Radio (SDR) that uses the Orthogonal Frequency Division Multiplexing (OFDM).The transmitter and receiver were modeled and simulated using the Xilinx(R) blockset in MATLAB Simulink(R) then implemented on hardware using Xilinx(R) System Generator. Performance of the SDR system is evaluated by measuring BER, PAPR and Spectral efficiency.

The experiments presented in this thesis make use of System Generator for DSP, a productivity tool from Xilinx, to design and to simulate system-level models in a MATLAB/Simulink environment, and to obtain BER, PAPR and Spectral efficiency results before implementing the design on actual hardware.

1.Introduction

Software Defined Radio (SDR) is a rapidly evolving radio technology that can support multiple wireless communication standards and systems. It is configured by modifiable software operating on reprogrammable hardware, thus giving the advantages of flexibility and adaptability. This is in contrast to typical radio systems implemented using fixed function hardware. Since SDR is intended for use in wireless communications, bandwidth considerations have to be taken into account. With the emergence of wireless communication technology and increase in number of wireless device users, bandwidth has become even more in demand. Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier technology for reliable high-rate and high-speed data transmission. The available transmission bandwidth is divided into several overlapping, narrow bands modulated by orthogonal carriers. Because of this, OFDM has the advantage of high spectral efficiency and is already becoming the popular choice for wireless communication systems.

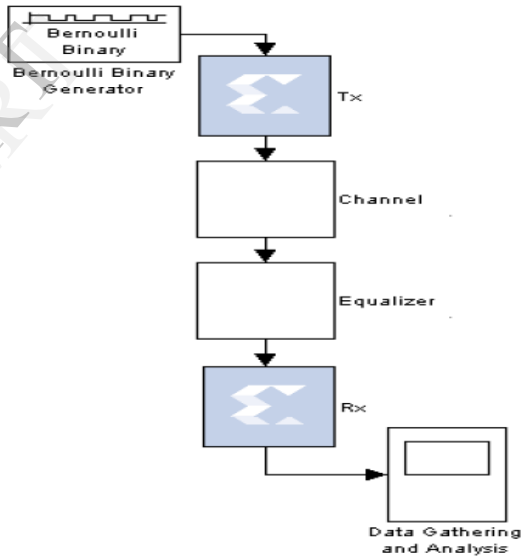


Figure 1. System Block Diagram

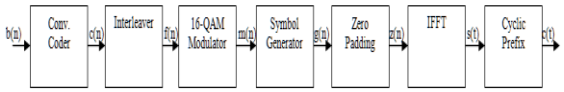
Designing an SDR system that uses OFDM thus offers flexibility, adaptability and bandwidth efficiency for wireless communication systems. The project objectives are to model, simulate and implement an SDR transmitter and receiver system that uses OFDM and to determine the acceptable SDR configurations for speech and audio inputs.

2.SDR System Implementation

2.1 System Level Implementation

Figure 1 shows the system level implementation of the SDR that uses OFDM. The SDR transmitter and receiver were modeled and simulated using the fixed-point Xilinx blockset in MATLAB Simulink. The transmitter and receiver systems are models are converted to two Xilinx blocks using hardware co-simulation design of Xilinx System Generator. These blocks were implemented on a Xilinx Spartan-3 FPGA built in the XtremeDSP (TM) Development platform ML506 Edition using Joint Test Action Group (JTAG) adapter for communication. The channel and equalizer were implemented on a MATLAB workspace. Bernoulli binary data generation, data gathering and analysis were done in MATLAB Simulink.

2.2 Transmitter Model



b(n) – Bernoulli binary data

c(n) – Convolution coder output, multiplexed into a single bit stream

f(n) – Forney interleaver output

m(n) – 16-QAM Modulated Signal

g(n) – Symbol generator output

z(n) – Zero Padding Output

s(t) – IFFT output

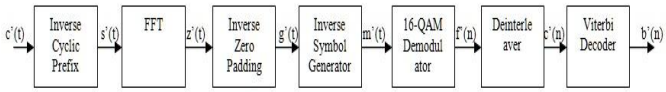
c(t) – Cyclic prefix output

Figure 2. SDR Transmitter Block Diagram

Figure 2 shows the SDR transmitter block diagram. A Simulink Bernoulli binary generator provides input binary data b(n). The Xilinx Convolution Encoder 7.0 block encodes the data with code rate 1/2 using convolutional codes '1111001' and '1011011'. It produces one binary stream for each of the two convolutional codes and multiplexes them into a single stream c(n). As a result, it changes the rate to twice the number of bits per second with respect to its input. The Xilinx Interleaver 6.0 block performs Forney interleaving and produces a binary stream of interleaved data f(n). The 16-Quadrature amplitude modulation is a combination of ASK and PSK so that a maximum contrast between each signal unit (bit, dibit, tribit, and so on) is achieved and produces signal m(n).The symbol generator increases the number of bits by replicating the modulated output and produces the signal g(t). The zero padding block pads the same number of zeros to the LSB and MSB of the symbol generator output and produces the signal z(n). The Xilinx LogiCORETM IP Fast Fourier Transform 7.1 block converts z(n) into N parallel streams, maps them onto orthogonal carriers, converts the N parallel output streams into a serial stream and produces the OFDM signal s(t). The main result of cyclic prefix is that the Intersymbol Interference (ISI) and Intercarrier Interference (ICI) may be spectrally concentrated.

2.3 Receiver Model

At the receiver, the equalized channel output was transformed back to the in-phase and quadrature components of m'(t) by the Xilinx FFT block. The demodulator block demaps them into symbols then converts them into a serial bit stream f'(n). The Xilinx Deinterleaver 6.0 block produces c'(n). It is provided as input to the Xilinx Viterbi Decoder 7.0that recovers the binary data b'(n) that should be the same as b(n).



c'(t) – Equalized signal

s'(t) – Inverse cyclic prefix output

z'(t) – FFT output

g'(t) – Inverse zero padding output

m'(t) – Inverse symbol generator output

f'(n) – 16-QAM demodulator output

b'(n) – Recovered binary data

Figure 3. SDR Receiver Block Diagram

3. RESULTS

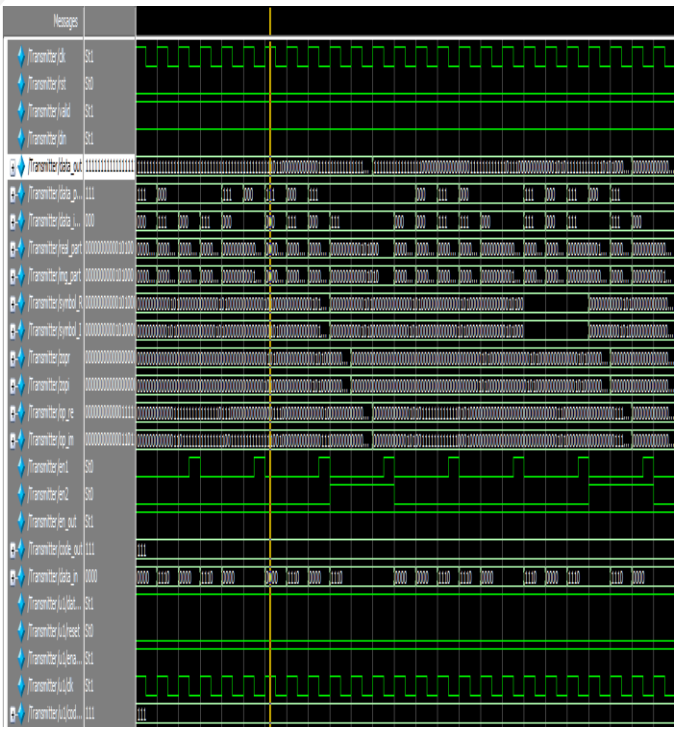


Figure 4. Transmitter Output

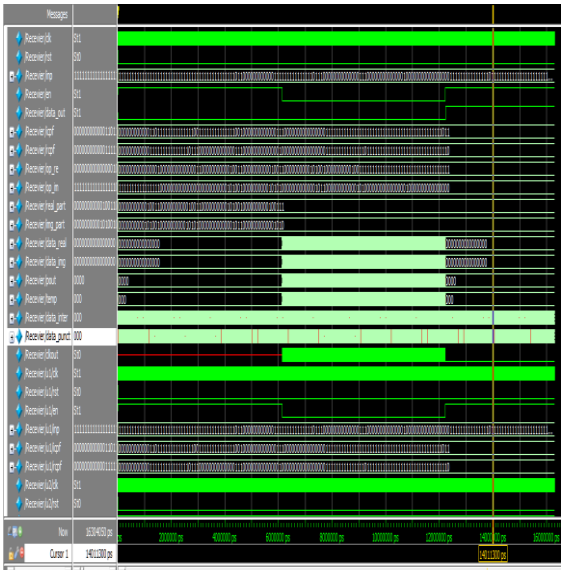


Figure 5. Receiver Output

I. HARDWARE IMPLEMENTATION

4.Hardware Implementation

Xilinx System Generator hardware co-simulation compilation is used to generate two blocks, one each for the transmitter and receiver. They are embedded with their respective Verilog codes, .bit and .ucf configuration files. JTAG hardware co-simulation is used to run the model on Xilinx Spartan-3 FPGA built in the XtremeDSPTM Development Platform ML506 Edition.

Input data to the transmitter comes from Simulink Bernoulli binary generator. The output of the transmitter is sent back to the MATLAB workspace where the effect of a wireless channel is simulated and applied to it. Equalization is also done on the workspace. The equalized output is provided as input to the receiver. The output of the receiver is sent back to the MATLAB workspace for BER, Spectral efficiency and PAPR calculations.

5.Conclusion And Recommendations

An SDR using OFDM was modeled, simulated and implemented using the Xilinx blockset in MATLAB Simulink. The chosen configurations were successfully implemented on a Xilinx Spartan-3 FPGA built in the XtremeDSPTM Development Platform ML 506 Edition using JTAG hardware co-simulation. The results were consistent with simulations on MATLAB Simulink. This shows that FPGA development may be done using the fixed-point Xilinx blockset in MATLAB Simulink. This allows easier, more flexible and cost-effective hardware implementation of systems.

It is recommended to explore other configurations for the existing blocks in the OFDM system such as using other typical code rates like 2/3, 3/4 and 5/6 for the convolutional coder. It is recommended to vary the number of branches in the Forney interleaver or change the type to a block interleaver. Increasing the number of IFFT points to 32, 64 or 128 is recommended to increase the bandwidth and number of subcarriers. The channel model and equalizer, which were implemented offline, should be implemented using blocks from the Xilinx blockset in MATLAB Simulink to allow the transmitter, channel, equalizer and receiver models to be integrated in a single model file for easier porting to the hardware. Further research could include redesigning the system model such that it could configure code rate, interleaving, modulation scheme and FFT automatically based on data input and channel conditions.

6.REFERENCES

[1] S. Venkatachalam, et.al., Implementation of Orthogonal Frequency Division Multiplexing (OFDM) Using Software Definable Radio (SDR) Platform. International Conference on Computational Intelligence and Multimedia Applications, 2007.

[2] H. Harada and R. Prasad, Orthogonal Frequency Division Multiplexing (OFDM) Transmission Technology in Simulation and Software Radio for Mobile Communications. Boston, USA: Artech House, 2002, ch. 4, sec 1, p. 165.

[3] C. S. Peng, WLAN 802.11a OFDM Transceiver. Algorithm, Architecture and Simulation Results. Trans. Wireless Technology Laboratory, National Chiao Tung University, January 2001.

[4] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice-Hall, 1998.

[5] B. S. Atal, et.al., Speech and Audio Coding for Wireless Network Applications. Kluwer Academic Publishers, 1993.

[6] Y. S. Choo, et.al., MIMO-OFDM Wireless Communications with MATLAB. IEEE Press, 2010