Design and Implementation of SDR Modulator and Demodulator using Xilinx System Generator

Jaydeep Muley, Niraj Jadhav
Mtech VLSI, School of SENSE
VIT Chennai
India

Augusta Sophy Beulet.P
School of SENSE
VIT Chennai
India

***Abstract*—**SDR is a term which generally refers to the transceiver systems in which the entire physical layer is implemented in software using various DSP algorithms. They have the ability to tune over the frequency ranging from 25MHz to 1.75GHz which makes them more versatile. SDRs are used in wireless and Satellite Communication (SATCOM) domains. They are much better compared to the traditional hardware based devices in terms of cross-functionality, cost and flexibility. They are suitable for today's requirement of smaller and constantly changing bandwidths. In this paper we have used BPSK modulation technique to transmit and receive digital data. The modulator and the demodulator are simulated in MATLAB/Simulink environment using Xilinx system generator.

An SDR consists of various modules which include channel sources of SDR, transmitter block, receiver block, and the most important the channelizers in the receiver.The channelizers are used to select a range of required frequencies from a particular RF band. Later these frequencies are used for modulation and demodulation purpose.

The objective of our project is to assemble an SDR block and simulate it in MATLAB/Simulink and to perform design implementation in ZED board using Xilinx ISE Design Suite/VIVADO.

Keywords*—SDR; MATLAB/Simulink; XILINX-ISE;DDS (Direct Digital Synthesis); system generator; BPSK modulation-demodulation;*

*****

I. INTRODUCTION

SDR can be defined as "Radio in which some or the entire physical layer functions are software defined". They provide a flexible architecture which is suitable for many radio standards. Because of their versatile uses and advantages like adaptability, reconfigurability and multifunctionality, SDRs have totally replaced the traditional methods of implementing communication modules [1].

Usually an SDR consists of four modules i.e. the channel sources, the transmitter, the receiver and the channel sinks. The channel sources are used to provide stimuli to the transmitter. The transmitter modulates and transmits the data signal over a carrier signal. The channelizer in the receiver selects a particular range of frequencies from the received set of frequencies and feeds them to the demodulator. Using the combined channel spectrum, output is obtained at the channel sinks. This output is the data sent by the modulator.The modulator and demodulator blocks are made using Xilinx block set in Simulink. We need to merge these blocks from XILINX –ISE into MATLAB. After simulating various blocks in Simulink a bitstream file is generated using the Xilinx system generator for the required FPGA architecture. This bitstream file is then mounted on the virtex-5 board for hardware analysis of the modules. Generally the modulator and the demodulator are mounted on separate FPGA boards.

II. XILINX DESIGN FLOW

The simulation and testing of the SDR is done using system generator, which is a system level modeling tool from Xilinx. As shown in Fig 1 a synthesizable design can be generated using these Xilinx system generator blocks. After synthesis the simulators output is fed back to Simulink for verification purpose. The result is then displayed in Simulink sink. A synthesizable RTL model is then generated in Verilog and the results are mapped into required FPGA board. Next is the implementation of the design which generates the UCF (User Constraints File). It contains the information regarding pin locations for each I/O pin specified in the HDL design file and the timing info such as clock frequency [2].

III. PROPOSED WORK

This work proposes to design a BPSK modulator and demodulator in Simulink and to generate Verilog codes using hardware co-design block for the BPSK system using Xilinx system generator. It also tests the circuitson virtex-5 board and synthesizes the HDL design to obtain their RTL diagrams.

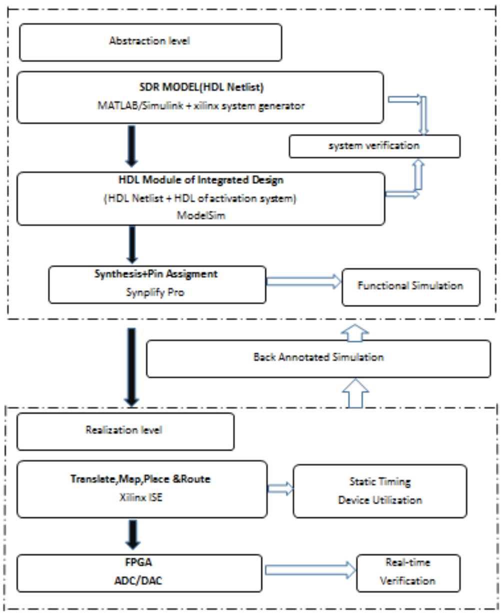


Fig.1. System Generator based design flow

IV. BPSK SYSTEM

Fig 2 represents a BPSK system which includes modulator, noise generator, channel and a demodulator.

A. *BPSK modulation in simulink*

In BPSK modulation the carrier signal is phase modulated keeping its amplitude and frequency constant. The modulated signal has two phases, one phase representing logic 1 and other representing logic 0. Both differ by 180° phase shift.

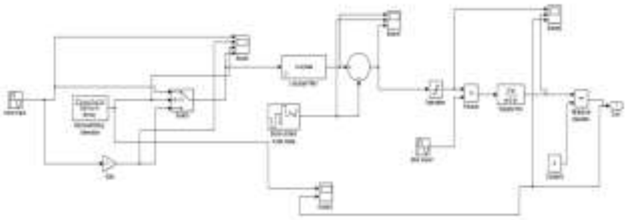


Fig 2 A BPSK system

As shown in Fig3 one of the inputs to the switch is a sine wave and other is a sine wave with 180° phase shift. The data signal is produced by the binary generator and is fed to the select input of the switch. Depending on the data signal the switch selects the sine or phase shifted sine wave and transmits the modulated signal. The output of modulator is shown in Fig 4.

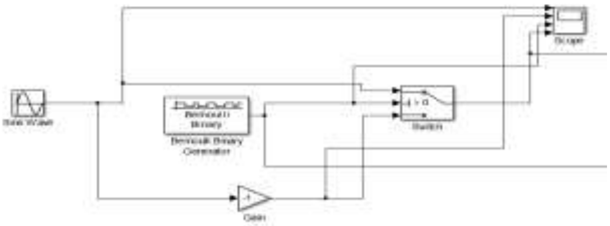


Fig 3 A BPSK modulator

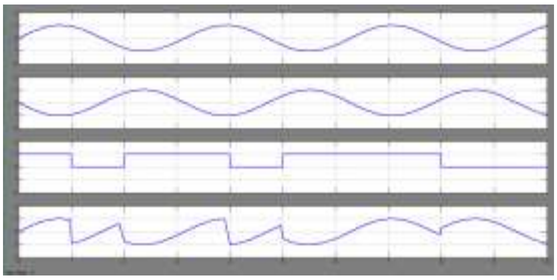


Fig 4 Output of BPSK modulator

The modulated signal is then passed over a channel where it is added with some noise. This job is done using a band limited white noise block with noise power [0.1] and sample time=0.1s. This signal is then passed to the demodulator i.e. the input of saturation block. The saturation block creates an upper and lower bound for the input signal. If the input signal exceeds these limits it is clipped off otherwise it is transmitted as it is. This signal is then multiplied by a sine waveform which is a carrier obtained in theory from the carrier recovery circuit. Then it is given to the transfer function block which provides a transfer function between input and output. This signal is then applied to the relational operator block which is basically a pulse shaper block which produces the demodulated signal given in Fig 5.

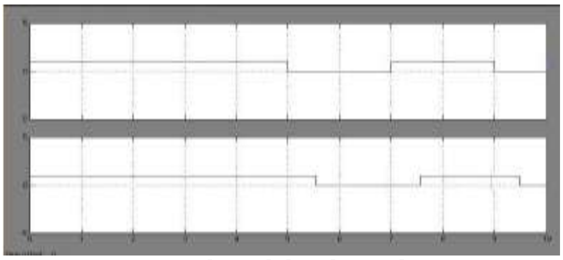


Fig 5 A demodulated signal ‘1’

B. *BPSK system using xilinx system generator*

In this method a DDS(Direct Digital Synthesis) is used. As shown in Fig 6 the modulating signal is generated by the LFSR and the carrier signal is generated by the DDS compiler block which is a direct digital synthesizer. It generates sinusoidal waves using a particular look-up table scheme. A digital integrator (accumulator) generates a phase that is mapped by the lookup table into the output sinusoidal waveform. Other DDS compiler is used to produce sine wave with 180° phase shift. A MUX is used to select between the outputs of both the compilers depending on the select input which is the modulating signal provided by the LFSR. The modulated signal is then transmitted over the channel where noise is added to it after which it arrives at the input of demodulator [3].

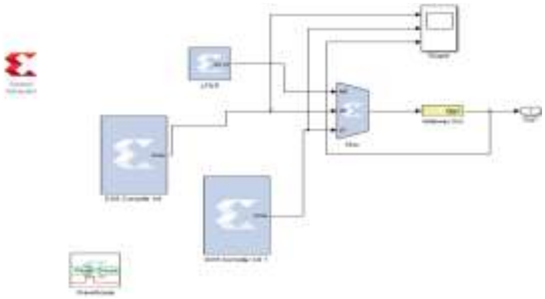


Fig 6 A BPSK modulator

As shown in Fig 7, the carrier wave is obtained using a DDS compiler. The carrier signal and the modulated signal are multiplied and given to the accumulator. The accumulator adds the obtained signal with all the samples multiplied from a period. The input to the reset of accumulator is converted to Boolean form before applying. The obtained result is then compared with a threshold value. If the signal value is above the threshold ‘1’ is transmitted otherwise ‘0’ is transmitted. Hence we get a demodulated signal [3].

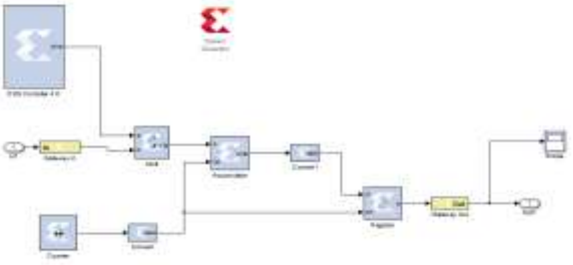
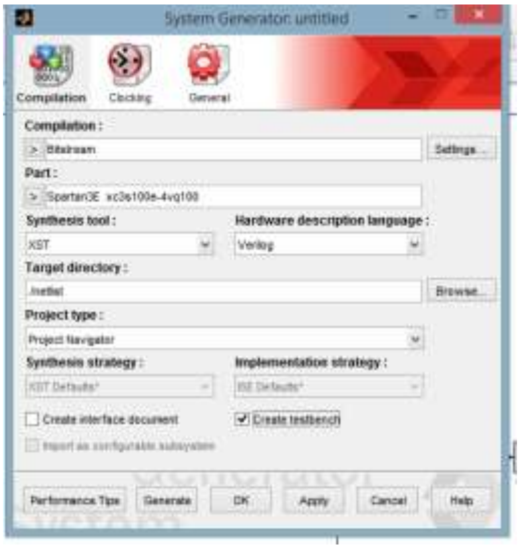


Fig 7 A BPSK demodulator

Both modulator and demodulator are simulated using system generator tool. A bit stream file was generated for SPARTAN 3E kit using Verilog. Fig 8 represents the parameters used in the Xilinx system generator tool.



Compilation : Bitstream
Part : Spartan3E xc3s100e-4vq100
Synthesis tool : XST
Hardware description language : Verilog
Target directory : ./netlist
Project type : Project Navigator
Synthesis strategy : XST Defaults*
Implementation strategy : ISE Defaults*
☐ Create interface document
☑ Create testbench
☐ Import as configurable subsystem

Fig 8 system generator specifications

V. BPSK SYSTEM ON FPGA

The modulator and demodulator are connected using a white Gaussian noise generator block in Simulink. We have implemented the BPSK modulator and demodulator on virtex-5 kit. To perform the simulation open the system generator tool, select Hardware Co-Simulation; ML506; JTAG in the compilation drop-down menu as shown in Fig 9. This will automatically select the virtex-5 kit. NowVerilog is chosen as hardware description language and tick the 'create testbench' option.

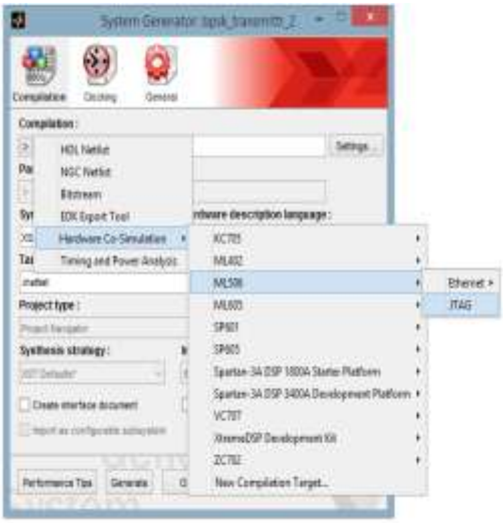
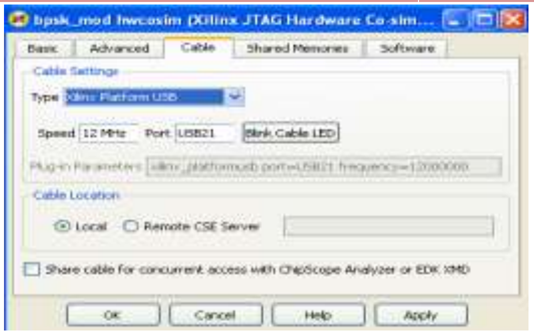

System Generator: bpsk_transmitter_2
Compilation
Clocking
General
Compilation :
Settings...
Hardware description language :
Project type :
Synthesis strategy :
Create interface document
Import as configurable subsystem
Performance Tips
Generate
HDL Netlist
NGC Netlist
Bitstream
EDK Export Tool
Hardware Co-Simulation
Timing and Power Analysis
KC705
ML402
ML506
ML605
SP601
SP605
Spartan-3A DSP 1800A Starter Platform
Spartan-3A DSP 3400A Development Platform
VC707
XtremeDSP Development Kit
ZC702
New Compilation Target...
Ethernet
JTAG

Fig 9 specifications for system generator

After simulating the circuit we obtain a hardware co-design block in a new window. This block is copied to the main circuit and its output is attached to the to-workspace block. Connect the Xilinx virtex-5 ML506 kit. Open the hardware co-design block and select type as Xilinx Platform USB as shown in Fig-10. Simulate the circuit again to get the output waveforms; this time simulated using Xilinx virtex-5 as hardware.



Type: Xilinx Platform USB
Speed: 12 MHz
Port: USB21
Plug-in Parameters: xilinx_platformusb port=USB21 frequency=12000000
● Local ○ Remote CSE Server
☐ Share cable for concurrent access with ChipScope Analyzer or EDK XMD

Fig 10 specifications for co-design block.

VI. RESULTS

After simulating both modules Verilog codes are generated. Codes are synthesized and implemented in Xilinx- ISE. RTL diagrams are generated as shown in fig-12 and fig-14. The device utilization summary of modulator and demodulator is shown in fig-15 and fig-16 respectively.

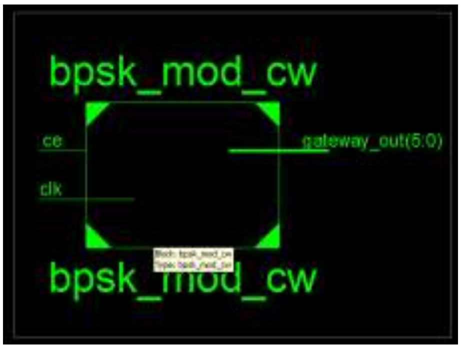


Fig 11RTL of modulator

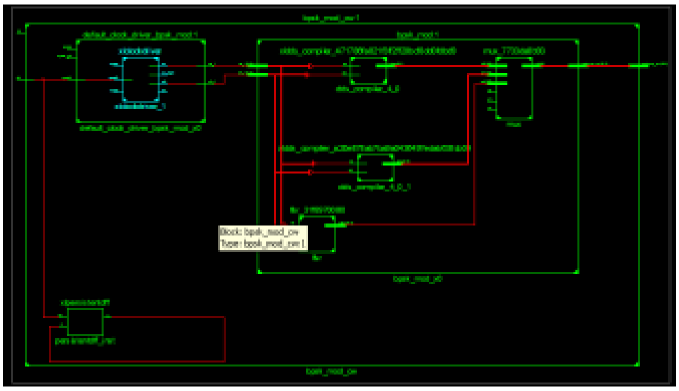


Fig 12 RTL diagram of modulator

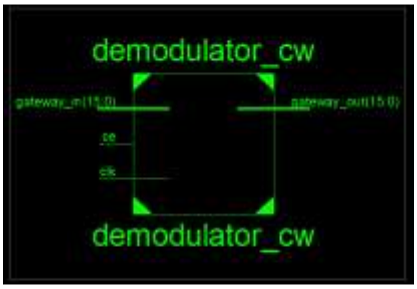


Fig 13 RTL of demodulator

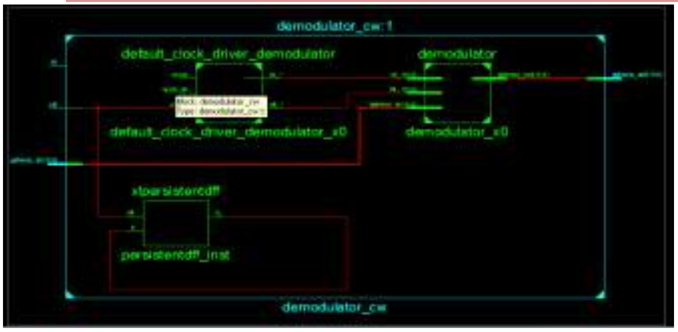
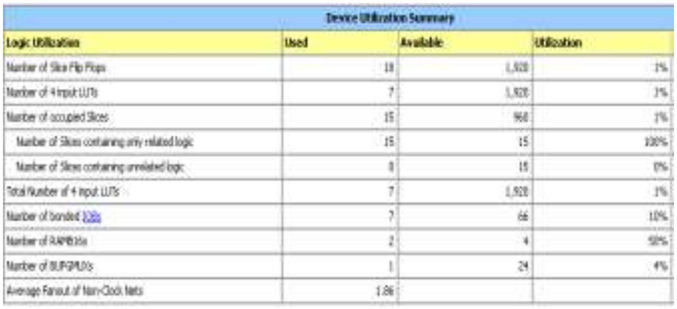
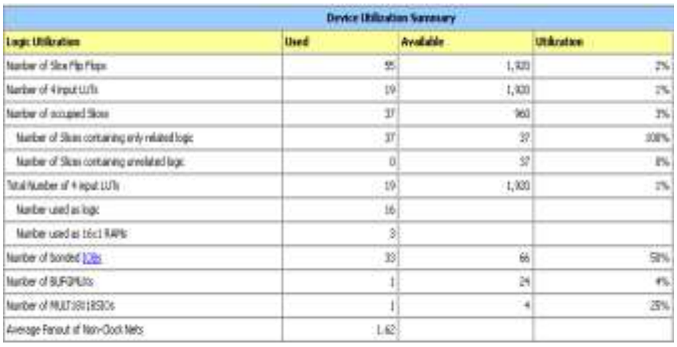


Fig 14 RTL diagram of demodulator



Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	18	1,920	1%
Number of 4 input LUTs	7	1,920	1%
Number of occupied Slices	15	960	1%
Number of Slices containing only related logic	15	15	100%
Number of Slices containing unrelated logic	0	15	0%
Total Number of 4 input LUTs	7	1,920	1%
Number of bonded IOBs	7	66	10%
Number of RAMB16s	2	4	50%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	1.86		

Fig 15 DUT of modulator



Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	55	1,920	2%
Number of 4 input LUTs	19	1,920	1%
Number of occupied Slices	37	960	3%
Number of Slices containing only related logic	37	37	100%
Number of Slices containing unrelated logic	0	37	0%
Total Number of 4 input LUTs	19	1,920	1%
Number used as logic	16		
Number used as 16x1 RAMs	3		
Number of bonded IOBs	33	66	50%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	1	4	25%
Average Fanout of Non-Clock Nets	1.62		

Fig 16 DUT of demodulator

VII. CONCLUSIONS

The modulator and demodulator using BPSK technique were simulated and implemented successfully. The blocks required for simulation were taken from the Xilinx block set repository in Simulink. These blocks were arranged to form modulator and demodulator circuits. Using system generator tool the required hardware co-design block was generated and used to obtain required waveforms. Verilog code for both blocks was generated and RTL diagrams were obtained. These blocks can be used as a mode of data transfer in a SDR. Depending upon the applications the frequency of operation can be varied.

FUTURE WORK

Assembling and compiling of various blocks of SDR in Simulink. Generating bit stream file using Xilinx system generator. Generating and compiling of HDL netlist. Implementation of thebitstream file obtained on Zed board.

REFERENCES

[1] Rini Supriya.L, Mr.Senthil Murugan, and Dr.R.C.Biradar, “Design and Implementation of Software Defined Radio Using Xilinx System Generator,”International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012 1 ISSN 2250-3153

[2] Tahseen Flaih Hasan,”FPGA Design Flow for SDR Transceiver using System Generator”International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3, Issue-4, April 2014

[3] S.O. Popescu, A.S.Gontean and G.Budura,”BPSK System on Spartan 3E FPGA” SAMI 2012 • 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics, January 26-28, 2012, Herl’any, Slovakia.

[4] SMITH, S. W. (2003) Digital Signal Processing - A Practical Guide for Engineers and Scientists.