International Journal of Advanced Computer Science, Vol. 2, No.11, Pp. 420-427, Nov., **2012.**

Modern Implementation of a BPSK Modulator on FPGA

Silvana-Oana Popescu, Aurel-Stefan Gontean & Georgeta Budura

Manuscript
Received:
28, Jan., 2012
Revised:
7, Jun., 2012
Accepted:
10, Aug., 2012
Published:
15, Dec., 2012

Keywords
BPSK, modulation, FPGA, simulation, hardware,

***Abstract*— The applications of FPGAs (Field Programmable Gate Array) became an important issue in designing electronic systems. In this paper, original designs of a BPSK modulator on FPGA are presented. Different implementations of the modulator were done in the Matlab/Simulink environment, as well as in System Generator. Three designs of the modulator were implemented on the Spartan 3E Starter Kit board using the VHDL language on Xilinx ISE 12.3. The modulated signal obtained from simulations was compared with the signal obtained after implementation.**

1. Introduction

The BPSK (Binary Phase Shift Keying) is one of the three basic binary modulation techniques. It has as a result only two phases of the carrier, at the same frequency, but separated by 180º. The general form for the BPSK signals are according to Equ. 1, where f_c is the frequency of the carrier.

$$s_i(t) = \begin{cases} s_1(t) = -A\sin(2\pi f_c t), & if\ 0_T \\ s_2(t) = +A\sin(2\pi f_c t), & if\ 1_T \end{cases} \quad \text{(Equ.1)}$$

The waveforms of the BPSK modulation are illustrated in Fig. 1.

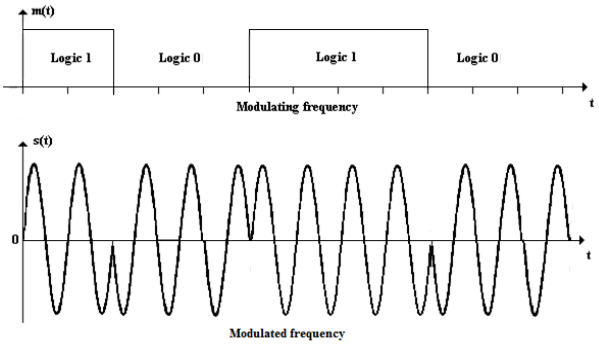


Fig.1 BPSK waveforms [1].

If "1" was transmitted, the modulated signal remained the same as the carrier, with 0º initial phase, but if "0" was transmitted, the modulated signal would change with 180º, like shown in Fig. 1.

The aim of the paper is to generate BPSK modulation which is a popular modulation technique used in communication industry, thus its symbol error performance and bandwidth efficiency. Among available tools for FPGA design, System Generator is a system-level modelling tool that facilitates Xilinx FPGA hardware design in Simulink and Matlab.

The paper is organized into 6 sections. Introduction which represents section 1 describes the basics of the BPSK modulation. In section 2, we offer information about the hardware and software tools used. In section 3, different implementations of the BPSK modulator in Simulink and System Generator are presented. Section 4 is dedicated to the implementation of the modulator on the Spartan 3E Starter Kit board. Section 5 presents the results of the implementations. The final section, section 6, presents conclusions.

2. Hardware and Software Resources

The first part of the Setup Lab measurement used for realizing the BPSK modulator is illustrated in Fig. 2. The resources used are the Spartan 3E Starter Kit board [2], the Xilinx WebPack ISE from Xilinx [3], monitor, a Tektronix oscilloscope [4] and a pulse generator from Agilent [5].

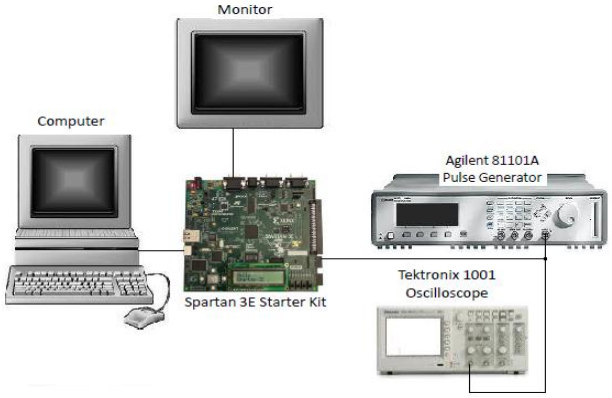


Fig.2 The setup Lab measurement with Spartan 3E Starter Kit board

The second part of the setup measurements consists of only a computer with the Matlab/Simulink environment on it, a Spartan 3E board and a LeCroy oscilloscope.

Silvana Popescu, Aurel Gontean, Georgeta Budura are from the Faculty of Electronics and Telecommunication in University Politehnica Timisoara, Romania (silvana.popescu@etc.upt.ro, aurel.gontean@etc.upt.ro, georgeta.budura@etc.upt.ro).

The Spartan 3E FPGA Starter Kit board is a development platform based on a Xilinx Spartan 3E FPGA. It provides a development platform for embedded processing applications [2]. The Spartan-3E family of FPGAs is designed to be well suited in a wide range of electronics applications [6].

The ISE WebPack from Xilinx is a fully featured front-to-back FPGA design solution and it offers HDL synthesis and simulation, implementation, device fitting and JTAG programming [3].

System Generator [7] is a DSP tool from Xilinx based on the Matlab/Simulink environment and is used for FPGA design. System Generator is actually a library in Simulink which translates a Simulink model into a hardware realization of the same model. It also maps the system parameters defined in Simulink into entities and architectures, ports and signals in a hardware realization. In addition, System Generator produces command files for FPGA synthesis, HDL simulation and implementation tools in order to obtain the hardware model. Only the subsystems and blocks from the Xilinx Blockset are translated by System Generator into hardware realization. All implementation steps, including synthesis, place and route are automatically performed to generate the FPGA programming file [8].

The System Generator design flow is shown in the following figure.

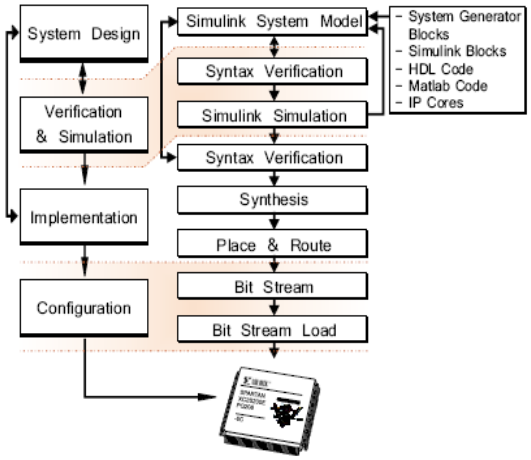


Fig.3 System Generator design flow [9]

In order to program the FPGA, two distinct software packages are used: Matlab and Xilinx ISE. Matlab/Simulink is the software where the system functionality is verified and where the programming takes place and ISE is where the program will be configured to run on the FPGA. The main bridge between the two packages is System Generator, a part of Matlab, which converts the Simulink math code into VHDL code that is recognized by the ISE software. The configuration options of the Xilinx token, in order to generate the VHDL code and the bitstream, is shown in Fig. 4.

The Agilent 81101A is a single-channel pulse generator, capable of generating all standard pulses and bursts up to 50 MHz [5].

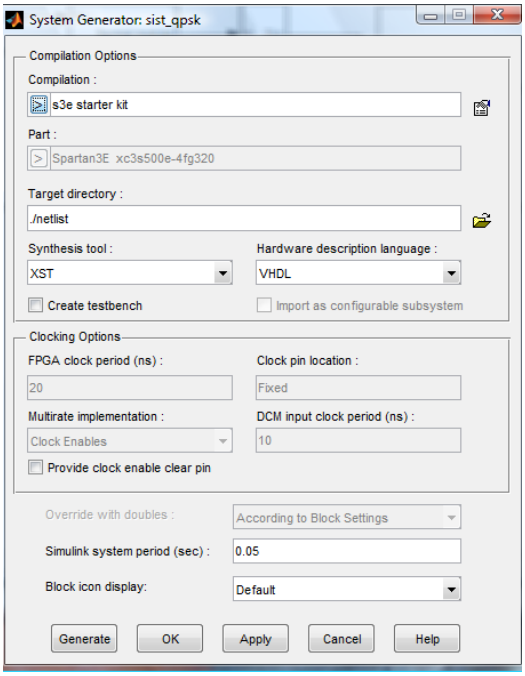


Fig.4 Configuration options of the Xilinx token [7]

3. BPSK Modulator

A. *BPSK Modulator in Simulink*

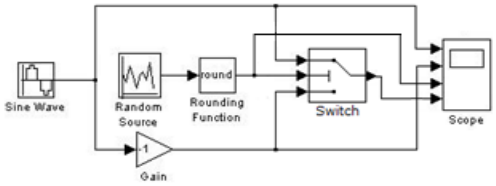


Fig.5 BPSK Modulator in the Simulink environment.

Fig.6 The waveforms on the scope: (a) Sinus (b) –Sinus (c) Modulating signal (d) Modulated signal.

Fig. 5 shows an implementation of a BPSK modulator in the Simulink environment and Fig. 6 the waveforms

generated by the corresponding blocks. The Simulink block set contains: the sine wave block (Fig. 5) which generates a sinus waveform (Fig. 6a) and with the help of the gain block, a sinus with phase difference of 180º (Fig. 6b), the blocks: random source and rounding function (Fig. 5) which generates the binary sequence or the modulating signal (Fig. 6c) and the switch (Fig. 5) which will choose between the first or third output depending on the value of the second input. If the second input is "1", the output value will be sinus, but if the second input is "0", the output will be –sinus (Fig. 6d).

Fig. 7 represents a second implementation of the modulator in the same Simulink environment. The Simulink block set contains approximately the same blocks, the main difference been the embedded Matlab function. This block contains a Matlab language function in a Simulink model. The block accepts multiple inputs and produces multiple outputs [10]. The Matlab code of the embedded Matlab function is shown in table 1 [11].

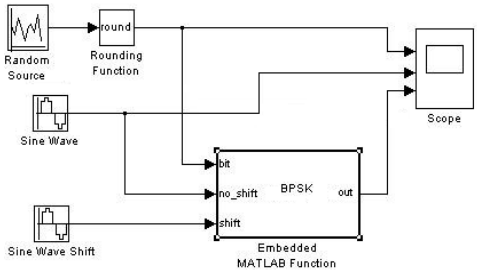


Fig. 7 Second implementation of the BPSK Modulator in Simulink.

TABLE 1
MATLAB CODE OF THE EMBEDDED FUNCTION

```
function out = BPSK(bit, no_shift, shift)
if bit==1
            out = no_shift;
    else    out = shift;
end
```

B. BPSK Modulator in System Generator

Fig. 8 and Fig. 10 illustrate an implementation of a BPSK Modulator using System Generator tools in Simulink.

In Fig. 8, the carrier as well as the modulating signal is generated external and the modulated signal is created inside the board and then routed to be seen on the scope as indicated in Fig.9.

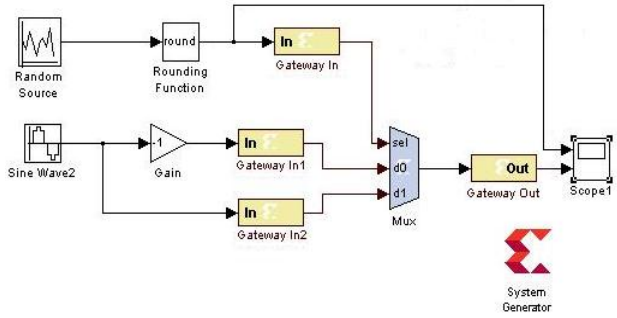


Fig. 8 BPSK Modulator in System Generator.

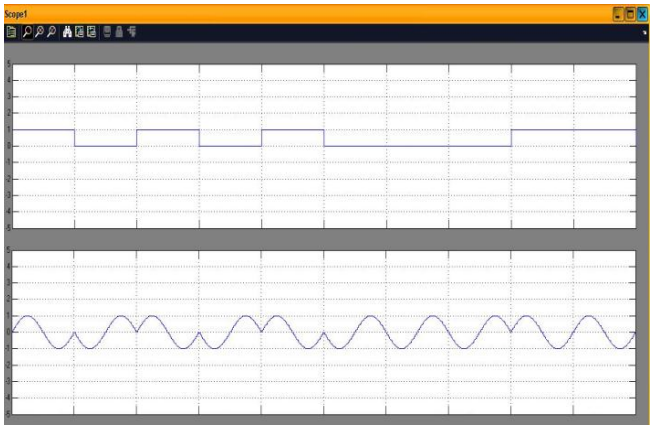


Fig. 9 The modulating and modulated signals.

The Simulink Blockset contains the sine wave and gain blocks, the random source and rounding function and the scope, blocks of which we talked about. The System Generator Blockset contains: the gateway in blocks which are the inputs into the Xilinx portion of the Simulink design, the mux which implements a multiplexer and the gateway out block which is the output from the Xilinx portion of the Simulink design.

In Fig. 10, the carrier is generated external, but the modulating signal is generated internal by a LFSR (Linear Feedback Shift Register). Fig. 11(a, b) illustrates the carrier signal, sinus and –sinus, as well as the modulating signal from the LFSR and the modulated signal obtained on the board. The Simulink Blockset contains the sine wave blocks and the scopes (Fig. 11). The System Generator Blockset contains: the gateway in blocks, the mux, the gateway out blocks and a LFSR.

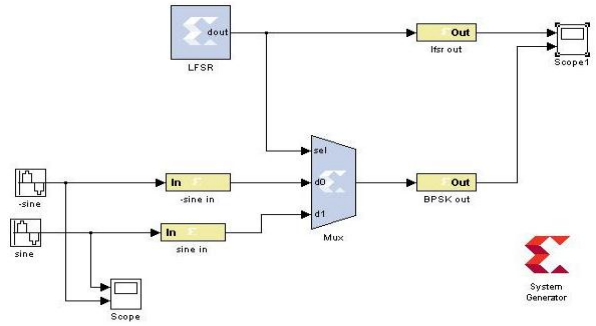
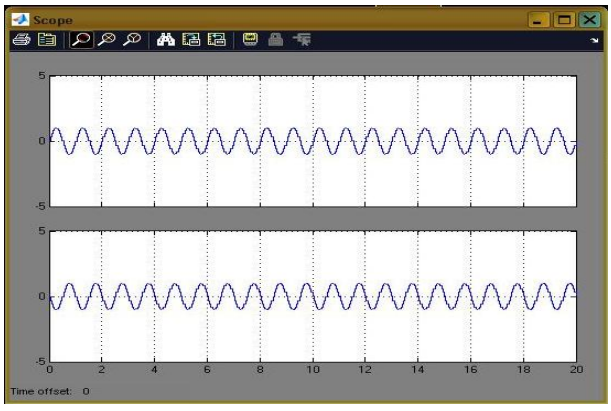


Fig.10 A second implementation of the BPSK Modulator in System Generator.



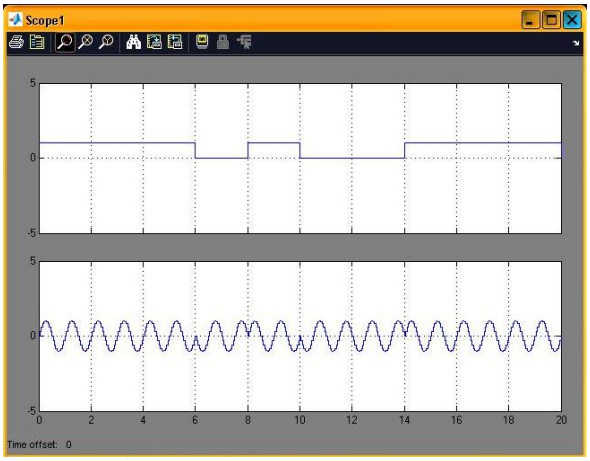


Fig. 11 The waveforms: (a) Sinus (b) –Sinus (c) Modulating signal of the LFSR (d) Modulated signal.

The third implementation of the BPSK Modulator, illustrated in Fig. 12 consists of signals generated internal. The carrier is generated internal by DDS blocks from System Generator and the modulating signal can be generated internal by the LFSR or external by the Agilent 81101A Pulse Generator. Fig.13 illustrates the signals obtained after implementing the modulator.

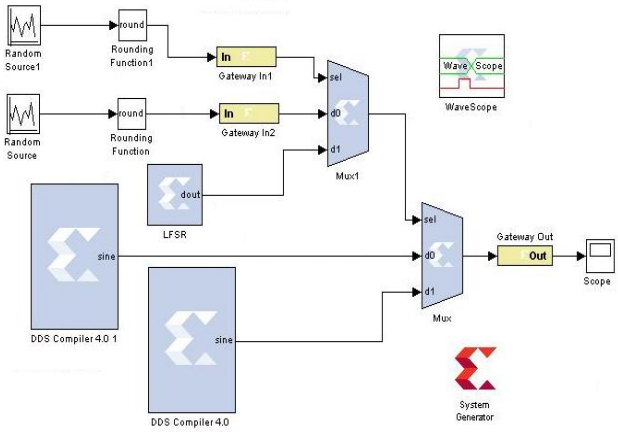


Fig.12 The third model of the modulator

The Simulink Blockset contains the random source and rounding function and the scope, blocks of which we talked about. The System Generator Blockset contains: the gateway in blocks, the mux blocks, the gateway out block, the LFSR block and two DDS compiler blocks.

The DDS Compiler Block is a direct digital synthesizer and it uses a lookup table scheme to generate sinusoids. A digital integrator generates a phase that is mapped by the lookup table into the output waveform [7]. The sinusoids can be seen in Fig. 13(c) and 13(d).

The mux block implements a multiplexer. It has one select input and a configurable number of data inputs that can be defined by the user [7].

With the System Generator WaveScope, the user can view the waveforms generated in a design. It is well suited for analyzing and debugging System Generator designs. The wavescope allows the user to observe the time-changing values of any wires in the design after the conclusion of the simulation [7].

The d0 and d1 inputs of mux1 represent the modulating signal. The sel input of mux1 selects between the d0 and d1 inputs. Because in the System Generator environment, a switch cannot be represented, we replaced it with a random sequence of bits. The same thing happens in mux2, where depending on the output of mux1, either sinus or –sinus is chosen.

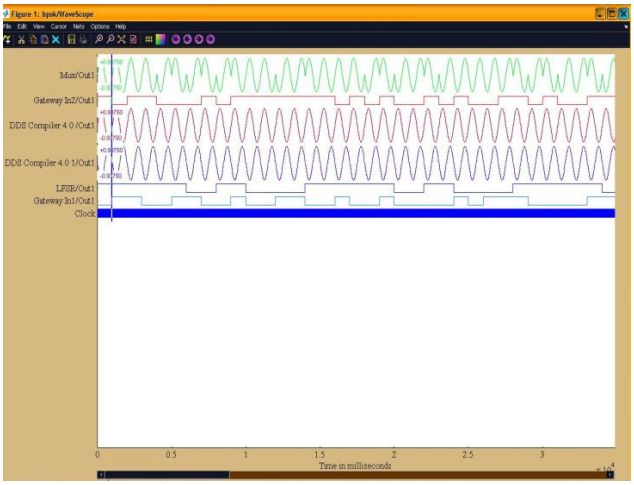


Fig. 13 The waveforms: (a) The modulated signal (b) The modulating signal (c) Sinus (d) –Sinus (e) The output of the LFSR (f) The signal obtained external, from a function generator.

A fourth implementation of the BPSK modulator is illustrated in Fig.14. The modulating signal (Fig. 15c) is generated internal by a LFSR (Linear Feedback Shift Register) (Fig. 14). The carrier is also generated internal by DDS blocks from System Generator (Fig. 14). The DDS Compiler Block is a direct digital synthesizer and it uses a lookup table scheme to generate sinusoids. A digital integrator generates a phase that is mapped by the lookup table into the output waveform [7]. The sine waveforms can be seen in Fig.15 (a) and (b). The mux block implements a multiplexer. It has one select input and a configurable number of data inputs that can be defined by the user. The d0 and d1 inputs of mux represent the sine waves. The sel input of mux represents the modulating signal and selects between the d0 and d1 inputs. If LFSR is '1', the modulated signal remained same as the carrier, but if '0' was transmitted, the yielded carrier is transmitted.

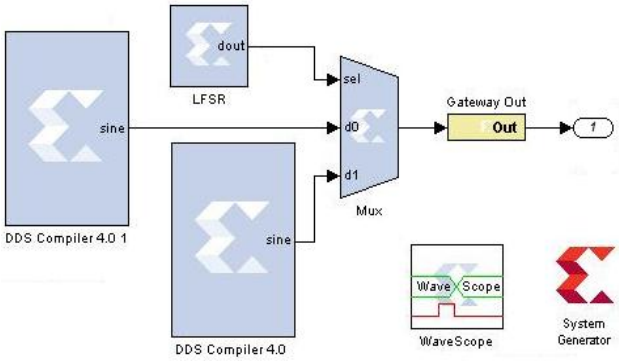


Fig. 14 The fourth implementation of the BPSK Modulator in System Generator.

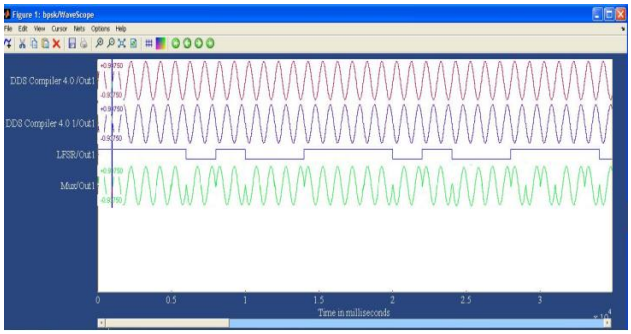


Fig. 15 The waveforms: (a)Sine (b) –Sine (c) The modulating signal (d) The modulated signal.

4. BPSK Modulator on Spartan 3E Starter Kit Board

The first implementation of the BPSK modulator has, as a model, the third implementation in System Generator with the signals routed to a VGA monitor. The carrier is generated internal, but in a ROM and that is the reason of which the sinus signal is represented discontinuous, by instantaneous samples of 16 different values [1], [12], [13]. The only thing that is different is that we used a switch that replaced the mux1 block. The switch behaves as a random sequence of bits which introduces either "1" or "0" depending on its position.

Fig. 16 represents the test bench lab used in implementing the BPSK Modulator on the Spartan 3E Starter Kit Board.

Fig. 16 Test bench lab.

As explained in Fig. 2, the experimental setup consists of a computer, a monitor, an oscilloscope, a pulse generator and the Spartan 3E board. The ISE Web Pack runs on the computer and it programs the Spartan 3E board. The pulse generator generates the signal from Fig. 17 and it is measured with a Tektronik oscilloscope. The pulses are than fed to an entry of a connector on the board. Depending on the position of a slide switch (Fig. 18), the modulating signal is acquired either external, from the pulse generator or internal, from the LFSR. Opposite to System Generator, the switch can be represented or can be configured in VHDL language. The modulated signal obtained is routed to the VGA port of the board, in order to be seen on the monitor.

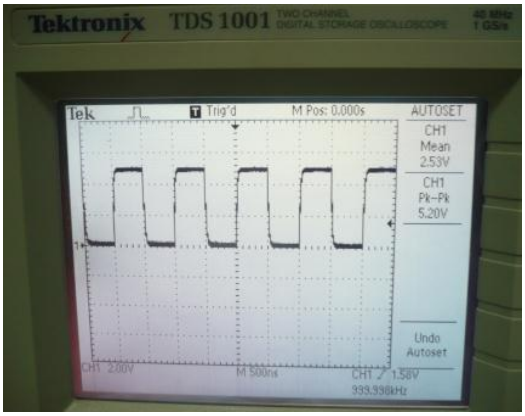


Fig. 17 Signal produced by the Agilent 81101A Pulse Generator.

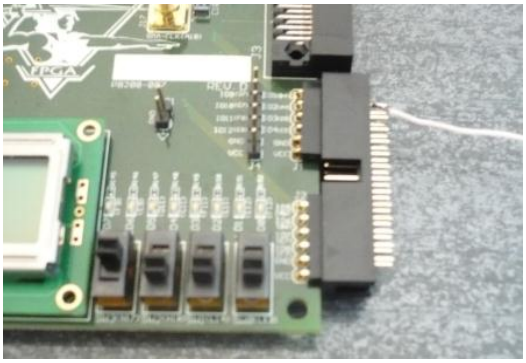


Fig. 18 The slide switches and the connector which makes the junction between the pulse generator and the board.

The second implementation of the BPSK modulator has, as a model, the fourth implementation in System Generator and is illustrated in Fig.14. The experimental setup (Fig. 19) consists of a computer, a Spartan 3E board and an oscilloscope. The Matlab/ Simulink, System Generator and Xilinx ISE packages run on the computer, and the Xilinx ISE programs the Spartan 3E board.

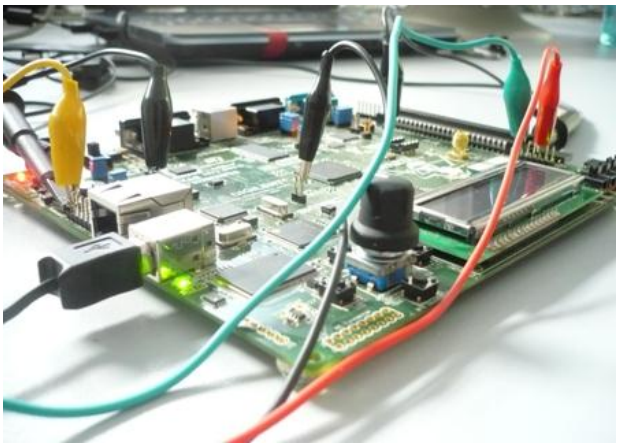


Fig. 19 BPSK Modulator – experimental setup.

The modulating signal is generated internal, in the modulator, by a LFSR. The carrier is also generated internal, and is made of 16 different values kept in a ROM memory [13]. The yielded carrier with 180º phase shift is obtained by reading the ROM memory later with 8 samples. If LFSR

was '1', the modulated signal remained same as the carrier, but if '0' was transmitted, the modulated signal became the yielded carrier. The modulated signal is then sent to the DAC (Digital-to-Analog Converter) on the board in order to be sent through a channel. The principle of the BPSK modulator implemented on the FPGA is illustrated in fig.20.

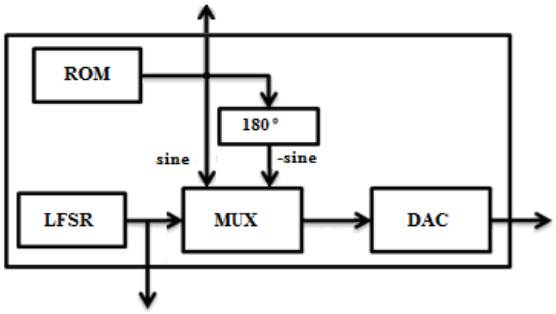


Fig. 20 The principle of the BPSK modulator on the FPGA

Fig. 21 illustrates the setup lab measurement used for realizing the third implementation of the BPSK modulator on FPGA. The experimental setup consists of a computer, a Spartan 3E board and an oscilloscope. The Matlab/ Simulink, System Generator and Xilinx ISE packages run on the computer, and the Xilinx ISE programs the Spartan 3E board.

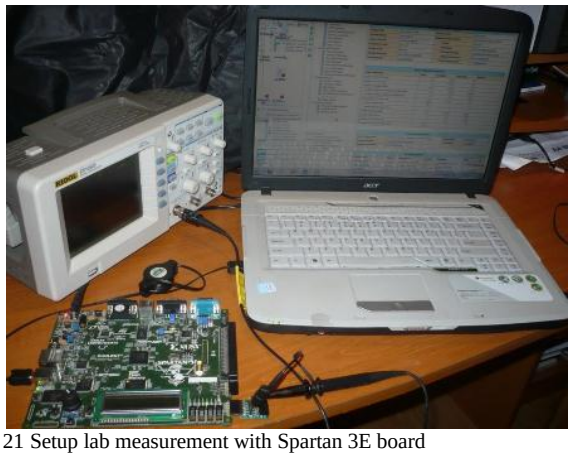


Fig. 21 Setup lab measurement with Spartan 3E board

The purposed design is the same as the second implementation of the BPSK modulator from Fig. 10. The design (Fig. 22) is completed with the new block, the BPSK modulator hwcosim obtained after the hardware co-simulation.

System Generator provides hardware co-simulation which makes possible to incorporate a design running in a FPGA directly into a Simulink simulation. The Simulink environment is used in order to verify the system functionality. When the design is made in Simulink, the results for the compiled portion are generated in hardware. This allows the compilation portion to be tested in actual hardware and can speed up the simulation dramatically. The method is called Hardware/Software Co-Simulation. This method enables building a hardware version of the model and, using Simulink environment, several tests can be performed in order to verify the functionality of the system in hardware. When the compilation is complete, a new library, formed by a single block, encapsulates the hardware implementation of the DSP system. The block has inputs and outputs according with the number of the GatewayIn and GatewayOut ports. This new block includes all the functionality required for the system to be implemented on FPGA and is linked to a bitstream that will be downloaded into the FPGA during co-simulation.

The new BPSK modulator (hwcosim block) has two inputs and two outputs according to the number of the GatewayIn and GatewayOut ports (Fig. 22). The block includes all the functionality required for the design to be implemented on the FPGA and is linked to a bitstream that will be downloaded into the FPGA during the co-simulation.

After the simulation is completed, the results are displayed as shown in Fig. 23.

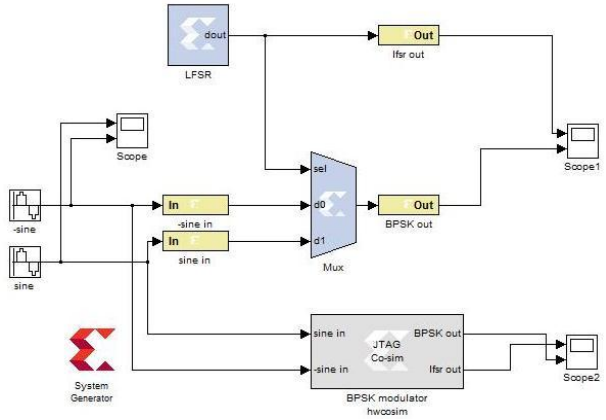


Fig. 22 BPSK modulator

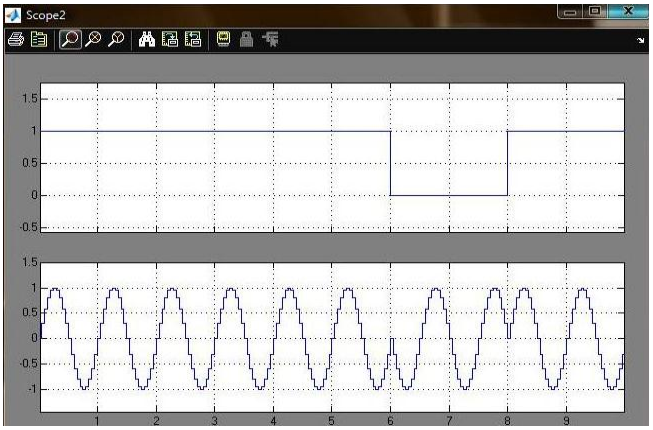


Fig. 23 The modulating and the modulated signals

5. Results

In the first implementation of the BPSK Modulator on the Spartan 3E Starter Kit board, the signals were routed to a VGA monitor. The BPSK modulation can be seen in Fig. 24 and Fig. 25. If the input data is "1", the transmitted signal to the monitor is unchanged and has a green border on the right of the figure, but if the input data is "0", the transmitted signal is yielded with 180º phase shift and has a

red border on the right of the figure.

Fig. 26 represents the design summary which represents the utilization of flip-flops, LUTs, slices used from the capabilities of the FPGA from the Spartan 3E board.

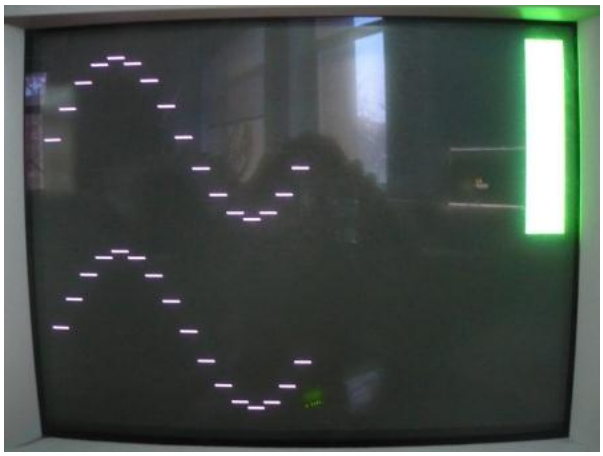


Fig. 24 The transmitted signal if the input is 1

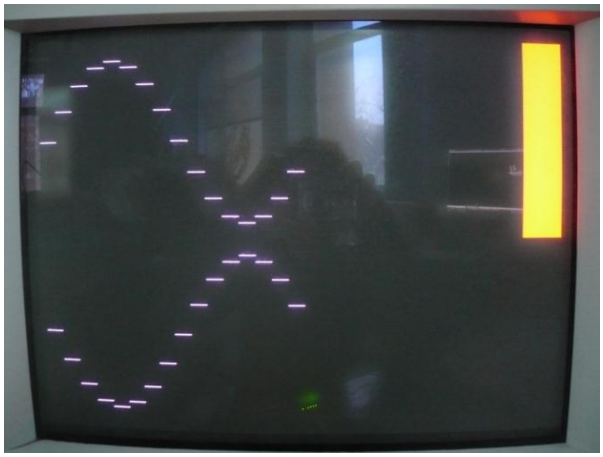


Fig. 25 The transmitted signal if the input is 0

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	679	9,312	7%	
Number of 4 input LUTs	952	9,312	10%	
Number of occupied Slices	765	4,656	16%	
Number of Slices containing only related logic	765	765	100%	
Number of Slices containing unrelated logic	0	765	0%	
Total Number of 4 input LUTs	1,101	9,312	11%	
Number used as logic	952			
Number used as a route-thru	149			
Number of bonded IOBs	39	232	16%	
Number of BUFGMUXs	3	24	12%	
Average Fanout of Non-Clock Nets	3.18			

Fig. 26 Design Summary of the first implementation of the BPSK Modulator on FPGA

In the second implementation of the BPSK Modulator, the signals were routed to the LeCroy WaveSurfer Xs Series Oscilloscope, a high performance digital oscilloscope.

Fig. 27 illustrates the signal in the modulator. The first one represents the modulating signal generated by the LFSR, the second one, the carrier and the third, the modulated signal. Fig. 28 illustrates the design summary of the modulator board. The design summary shows the various synthesizer options that were enabled and some device utilization and timing statistics for the synthesized design.

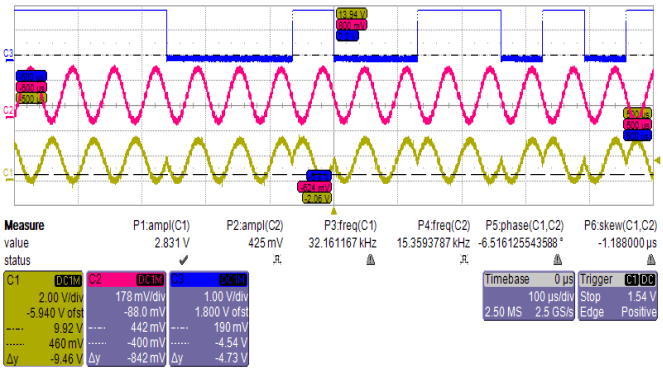


Fig. 27 The waveforms: (a) The modulating signal (b) Sine (c) The modulated signal.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	97	9,312	1%	
Number of 4 input LUTs	111	9,312	1%	
Number of occupied Slices	94	4,656	2%	
Number of Slices containing only related logic	94	94	100%	
Number of Slices containing unrelated logic	0	94	0%	
Total Number of 4 input LUTs	114	9,312	1%	
Number used as logic	111			
Number used as a route-thru	3			
Number of bonded IOBs	20	232	8%	
Number of BUFGMUXs	2	24	8%	
Average Fanout of Non-Clock Nets	3.34			

Fig. 28 The Design Summary of the second implementation of the BPSK Modulator.

The third implementation has the design summary shown in Fig. 29.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	21	9,312	1%	
Number of 4 input LUTs	25	9,312	1%	
Number of occupied Slices	31	4,656	1%	
Number of Slices containing only related logic	31	31	100%	
Number of Slices containing unrelated logic	0	31	0%	
Total Number of 4 input LUTs	25	9,312	1%	
Number of bonded IOBs	50	232	21%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	1.65			

Fig. 29 The design summary of the third implementation of the BPSK modulator

Comparing the design summary (fig. 29) obtained by generating the VHDL code directly from System Generator with the design summary illustrated in Fig. 26 and Fig. 28, the logic utilization of the board was lower in terms of the slice flip-flops and LUTs used. If in this case, the number of the slice flip-flops was 21 and the number of LUTs was 25

from a total of 9312, in Fig. 28, these numbers are higher: 97 for flip-flops and 111 for LUTs and in Fig. 26, these numbers are even more higher: 679 for flip-flops and 952 for LUTs.

6. Conclusions

We proposed two implementations of the BPSK Modulator in the Matlab/Simulink environment, the first with simple blocks and the second, with a block in which we wrote Matlab code. Then, we made a proposal of four implementations of a BPSK modulator in System Generator. In the first, the three signals: the carrier, the modulating and the modulated signals where generated external. In the second scheme, the carrier is generated external, and the modulating signal is generated internal by a LFSR. In the third scheme, all three signals were generated internal with the exception of the modulating signal which can be obtained either internal by the LFSR, or external by the pulse generator and in the fourth implementation, all signals were generated internal.

In the second part of the paper, we made three implementation of the BPSK modulator on the Spartan 3E Starter Kit. The first one was based on the third proposal of the modulator made in System Generator and the signals were routed to a VGA monitor. If "1" was transmitted, the modulated signal remained same as the carrier, but if "0" was transmitted, the modulated signal was yielded with a 180º phase (Fig. 24 and Fig. 25). The design has been written in the VHDL programming code by Xilinx software. The second implementation had as a model the fourth proposal of the modulator in System Generator and the signals were routed to an oscilloscope. The same principle was applied in the third implementation.

Comparing the design summary obtained with other works in this field [1], the logic utilization of the board was lower in terms of the slice flip-flops and LUTs used. All of these make the design suitable in terms of propagation, implementation and logic utilization of the Spartan 3E boards used in this work.

The time used to simulate the designs with components from the Xilinx Blockset is longer than the time taken to simulate in Simulink. Still, using System Generator inside Simulink for bit and cycle time simulations is an order of magnitude faster than running the same simulation through an HDL simulator [16]. Also, the time for designing a DSP system in the System Generator environment is smaller than the time used for writing VHDL code. A main advantage for a design made in System Generator is that it can be validated through simulations before implementing it in hardware. A unique hardware in the loop co-simulation allows designers to accelerate the simulations while simultaneously verifying the DSP system in hardware [16].

References

[1] S.O.Popescu, A.S. Gontean, F.Alexa, "Improved FPGA – Detector", in *Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI2011)*, Romania, 2011, pp. 455 – 458.

[2] Spartan 3E FPGA Starter Kit Board, User guide, Xilinx, 2011.

[3] ISE 10.1 Quick Start Tutorial, 2008.

[4] Digital Storage Oscilloscopes, TDS 1000B Series, TDS 2000B Series. Data Sheet, Tektronix, 2006.

[5] Agilent 81101A, 50 MHz Pulse Generator, Quick Start Guide, Agilent Technologies, 2000.

[6] Spartan-3E FPGA Family Data Sheet, Xilinx, 2008.

[7] System Generator for DSP. Getting Started Guide. Xilinx. 2008

[8] S.O.Popescu, A.S.Gontean, G.Budura, "Hardware Co-Simulation of the BPSK and QPSK Systems on FPGA", *in press.*

[9] G.E.Martinez-Torres, J.M.Luna-Rivera, R.E.Balderas-Navarro, "FPGA-based Educational Platform for Wireless Transmission using System Generator", *Proceedings of the 2006 IEEE International Conference on Reconfigurable Computing and* FPGAs, Mexico, 2006, pp. 1-9.

[10] S.T.Karris, "Introduction to Simulink with Engineering Applications", *Orchard Publications*, USA, 2006.

[11] J.S.Ruque, D.I.Ruiz, C.E. Carrion, "Simulation and implementation of the BPSK modulation on a FPGA Xilinx 3 xcs200-4ftp256, using Simulink and the System Generator blockset for DSP/FPGA" http://www.ece.unm.edu/xup/docs/collaboration/BPSK_Modulator.pdf

[12] S.O.Popescu, G.Budura, A.S.Gontean, "Review of PSK and QAM – Digital Modulation Techniques on FPGA", *International Joint Conference on Computational Cybernatics and Technical Informatics (ICCC-CONTI)*, Romania, 2010, pp.327-332.

[13] F.Ahamed, A.Scorpino, "An educational digital communications project using FPGAs to implement a BPSK Detector", *IEEE Transactions on Education*, Vol.48, No.1, 2005, pp.191-197.

[14] S.O.Popescu, A.S.Gontean, G.Budura, "Simulation and Implementation of a BPSK Modulator on FPGA", in *Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011)*, Romania, 2011, pp.459-463.

[15] S.O.Popescu, A.S.Gontean, G.Budura, "BPSK System on Sprtan 3E FPGA", accepted at the *10th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI 2012)*.

[16] S.Mittal, S.Gupta, S.Dasgupta, "System Generator: The State-of-art FPGA Design tool for DSP Application", in *Proceedings of the 3rd International Conference on Embedded Systems, Mobile Communication and Computing*, India, 2008. http://home.engineering.iastate.edu/~sparsh/SystemGenerator_FPGA.pdf