Comparison of two new methods for implementa BPSK modulator using FPGA

Amar Hebibi[1], Arres Bartil[2], Lahcene Ziet[3]
[1,3]Department of Electronics, Faculty of Technology, Ferhat Abbas University, Setif-1, Algeria
[2]Laboratory of scientific instrumentation, Department of Electronics, Faculty of Technology, Ferhat Abbas University, Setif-1, Algeria

Article Info

Article history:

Received Dec 5, 2019
Revised Feb 8, 2020
Accepted Feb 22, 2020

Keywords:

BPSK
Digital modulator
FPGA
Hardware co-simulation
System generator

ABSTRACT

The design of electronic systems has become mainly dependent on FPGAs applications. This is due to the softness effectiveness progress by reconfigurable computing and reduced time to develop solutions for digital signal processing. In this article, we present the theoretical backgrounds of a BPSK modulation and hardware designs of the BPSK system, a firstly with the help of Matlab/Simulink reliant on the System Generator and a second with Xilinx ISE VERILOG Hardware Description Language. In order to show the differences between them, in terms of efficiency, duration of development and how many resources are used in FPGA. For the projected system, we have a tendency to aimed toward employing a moderately sized, low-value FPGA to implement the system. The Atlys development board by Digilent to configure develops, and run the system, based on a Xilinx Spartan-6 LX45 FPGA.

Corresponding Author:

Amar Hebibi,
Department of Electronics, Faculty of Technology,
168 University Ferhat Abbas, Setif-1, Algeria.
Email: hebibiammar@hotmail.fr

1. INTRODUCTION

The Wireless communication applications required different computational complexes as shown in the Table 1 [1]. However, wireless communication systems are applicable to these challenges; this is in view of a very important role played by the FPGA (Field Programmable Gate Array) has became a fundamental part in implementing DSP (Digital Signal Processing) systems, mostly in areas such as digital communications which are based on digital modulation. A digital modulation technique has happened and it can be visible in all areas of wireless communications systems. The goal of this article is to exam the functionality of two methods of implementing the BPSK system, In terms of the resources used to generate the BPSK modulation, the development time and what method allows us to use the different components integrated into the Atlys board, for example, the National Semiconductor LM4550 AC '97 audio codec. And compare at the same time with other works in this domain

Table 1. Computational complexity of some wireless communication standards

Wireless standard	Approximate complexity (MIPS)
802.11a&b	9000 MIPS
W-CDMA	6000 MIP
IS-95	500 MIPS
GPRS	300 MIPS
IS-136	200 MIPS
GSM	100 MIPS

The first method is to using hardware co-simulation provided by System Generator which is considered as system-level modeling instrument that Simplifies Xilinx FPGA hardware layout in Matlab and Simulink. Second utilize hardware description language (HDL), like VHDL or Verilog HDL, which is a common language for designers. That's what you know by high-level language and simulation, synthesis equipment are accessible. It confers flexible design reconfiguration and various levels of abstraction.

2. RELATED WORKS

The successful implementation of simple communication modulators has opened the doors for the implementation of more complicated systems such as QPSK and QAM. This paper focused on the implementation of BPSK. The work proposed by SilvanaPopescu, AurelGontean and GeorgetaBudura [2], made a suggestion of three applications of a BPSK modulator in System Generator. In the first, the three signals: the carrier, the modulating and the modulated signals were created externally. In the second scheme, the carrier is created external, and the modulating signal is created internally by an LFSR. And in the third scheme, all three signals were created internal a carrier is created internally by two DDS blocks provided in the “Xilinx Blockset” from System Generator, the exception of the modulating signal which can be obtained either inside by the FSR or external by the generator pulse. Implemented the BPSK modulator on the StarterKit Spartan 3E founded on the third proposition of the modulator construct in System Generator with the option to generate the VHDL code.

After one year the same researchers are presenting tow works, an implementation of the BPSK System (Modulator and Demodulator) in the Matlab/Simulink environment [3]. Then, we have a tendency to create a proposal of a BPSK System in System Generator. Both, the modulating signal and the carrier are generated internally, the modulating signal by an LFSR and the carrier by a two DDS Compiler. At the output of a mux block, we obtained the modulated signal. The BPSK System integrated on the Spartan 3E Starter Kit board has an identical principle because of the implementation in System Generator.

Although System Generator has the ability to create a VHDL code, the code was generated from the beginning because of this design because it is difficult to read the code that was created. The carrier which was indeed generated internal, in a ROM memory, but made of 16 different values, is the only noticeable difference. Was obtained a carrier with 180º phase shift by reading the ROM memory following with 8 samples.

The second work presented by Silvana Popescu,Georgeta Budura, Aurel Gonteanin 2012 [4], BPSK modulator (hwcosim block) possesses two inputs and two outputs depending to the number of the GatewayIn and GatewayOut ports. The block count all the practicality needed for the look to be achieved on the FPGA and is coupled to a bitstream that can be implemented in the FPGA After computing the VHDL code generated by System Generator. B.K.V. Prasad and SaiPriyahave presented the simulation of modulation and demodulation techniques utilize MATLAB Simulink and system generator for simulation and implementation on spartan-3e FPGA board [5], which impact flexibility for designing and testing so the development will be very easy. The FPGA was reconfigured and programmed with the assistance of ARM processor to compile the bit files to pick the desired modulation supported our demand and application that has best channel support.

The proposed approach in work of SilvanaPopescu, GeorgetaBudura, AurelGontean [4] is only appropriate for this design he Said, logical slice used in the board is smaller if we use the VHDL code generated from the System Generator, without writing the code from the starting. This is true because a carrier is generated external so he generates only LFSR and mux and does not work when using the DDS, This is recognized by SilvanaPopescu, AurelGontean, GeorgetaBudura [3] when he said; the code was made from the beginning because the generated code was hard to Read. This is also evident in [2, 5], the third scheme, all three signals were generated internal so The logical use of the board was higher as shown in device utilization summary, This is obvious due to the usage of two DDS and what they contain a lookup table scheme to generate sinusoids.

Also, the time requiredto design a DSP system by writing VHDL code is greater than the time used for the System Generator environment. The essential utility for a design fabricated in System Generator is that it can be confirmed over simulations before realized in hardware. This is what was agreed in [2-5]. The time consumed to simulate the designs with Simulink is smaller than the time consumed to simulate with components from the Xilinx Blocks. Even if utilize System Generator internal Simulink for bit and period time simulations is an order of magnitude faster than working the same simulation through an HDL simulator [6].

The way we implemented our systems is novel and different from what others presented like it, all of these suggestions focus on using shared simulations of devices provided by the system generator. This is the first method a carrier is generated internally by only one DDS compiler. The second method, I did not find an article about, I hope it is an exceptional work.

3. BACKGROUNDS

3.1. Digital modulation

It is the process of varying the characteristics of a periodic waveform known as carrier signal with a modulating signal that typically contains information to be transmitted. It is the process of superimposing the information contents of a modulating signal on a carrier signal by altering its characteristics according to the given modulating signal. Digital modulation is the procedure by which digital symbols are transmitted into waveforms that are appropriate with the characteristics [7]. The modulation technique used in this essay is BPSK (Binary Phase Shift Keying) and it is mostly applied in digital transmission [8]. In all PSK modulation techniques, BPSK modulation is the simplest and most robust form. It is not suitable for broadband applications and can modulate only 1 bps. The BPSK modulator is sort of simple and is illustrated in Figure 1. The modulated signal BPSK s (t) we get as a result of the multiplication of the binary sequence m (t) or the modulation signal by a sinusoidal carrier and. the modulated signal BPSK s (t) is obtained as a result of the multiplication of the binary sequence m (t) or the modulation signal by a sinusoidal carrier. The waveforms of the BPSK signal generated by the modulator are shown in Figure 2.

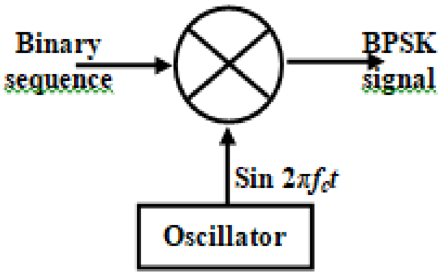


Figure 1. BPSK modulator [9]

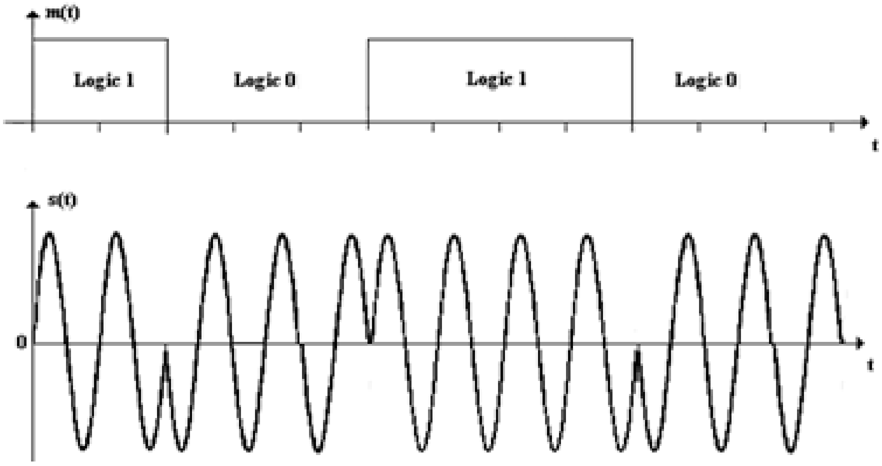


Figure 2. BPSK waveforms [10]

3.2. System generator of xilinx ISE

System Generator Xilinx ISE is a DSP design device from Xilinx that permits the employment of the MathWorks model-based Simulink design environment for FPGA design. Previous expertise with Xilinx FPGAs or RTL design methodologies isn't needed once using System Generator. Designs are captured in the DSP friendly Simulink modeling environment employing a Xilinx specific block. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file [11]. System Generator provides accelerated simulation through hardware co-simulation. System Generator will automatically create a hardware simulation token for a design captured in the Xilinx DSP blocks.

Direct digital synthesizers (DDS), or numerically controlled oscillators (NCO), are necessary elements in several digital communication systems. Synthesizers are used for constructing digital down and up converters, demodulators, and implementing various types of modulation schemes, including BPSK. A common technique for digitally generating a complex or real-valued sinusoid employs a lookup table scheme. The samples of a sinusoid stored in the lookup table. A digital integrator is employed to get a suitable phase argument that's mapped by the lookup table to the required output waveform.

4. DESIGN METHODOLOGIES

The two methods implementation of BPSK modulation in a system can supply a temple that can be utilized conferring to the requirements and the environment of the special application. The most combined approach for the implementation of the modulation techniques is by intermediary of MATLAB/SIMULINK environment or system generator, and then it can be transformedinto Verilog HDL or VHDL for the synthesis and ultimately implemented in FPGA. Another approach is to program the system in Verilog HDL immediately. The modulator for BPSK can be programmed using Verilog and ultimately be implemented in FPGA.

4.1. The first method: design and simulation of BPSK modulator using system generator

The implementation of a BPSK Modulator, illustrated in Figure 3. The functions and operations of each block are discussed in the following descriptions under sub-headings.

a) System Generator Block: The token is used to set Simulink system period: 1, and FPGA clock period: 10 ns (100 MSps sampling rate). It is used to generate VHDL netlist of BPSK modulation [12].
b) LFSR Block: the modulating signal generated internal by the LFSR (Linear Feedback Shift Register) [13].
c) DDS Compiler 4.0 Block: The carrier is created internal by DDS block from System Generator, it use a lookup table scheme to generate sinusoid of 10MHz [14].
d) Inverter Block: invert the signal generated by DDS Block, means negative sinusoid of 10MHz.
e) Mux Block: The mux block contains a multiplexer have one select input and a various number of data inputs which will be aware by the user. The d0 and d1 inputs of mux symbolize the sine waves. The sel input of mux symbolizes the modulating signal and chosses between the d0 and d1 inputs. LFSR is ‘1’, the modulated signal stayed the same as the carrier, but if ‘0’ was transmitted, the abandoned carrier is transmitted [15].
f) Bpskhwcosim Block: BPSK hwcosimacquired after the hardware co-simulation. The new block (BPSK hwcosim) has one output giving to the number of the GatewayOut port. The block comprises all the practicality needed for the design to be acheved on the FPGA and is coupled to a bitstream that will be downloaded inside the FPGA over the co-simulation [16].

Figure 3 represents the signals acquired after implementing the modulator. The results are visualized as shown in Figure 4. We can compare the results obtained after the hardware co-simulation with the results from a function generator. After compilation the VHDL code generated by System Generator, the summary of our design is clarify as in Figure 5. The design summary shows the various synthesizer options that were enabled and some device utilization and timing statistics for the synthesized design.

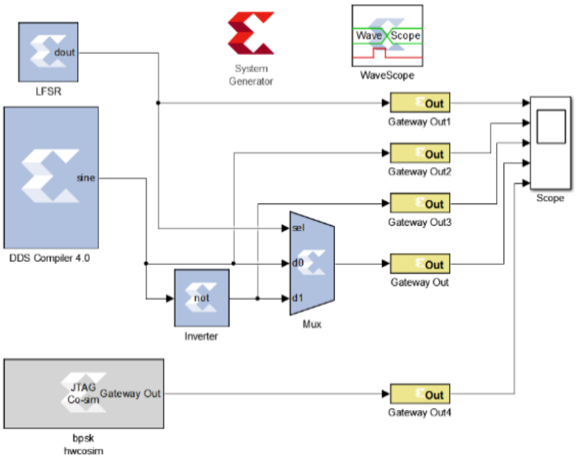


Figure 3. BPSK modulator in system generator with hardware co-simulation [17]

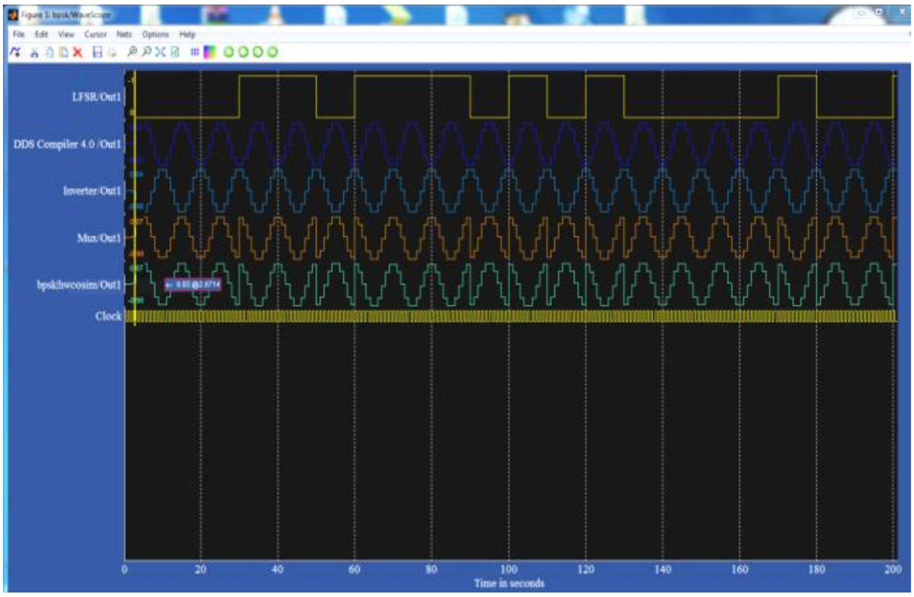
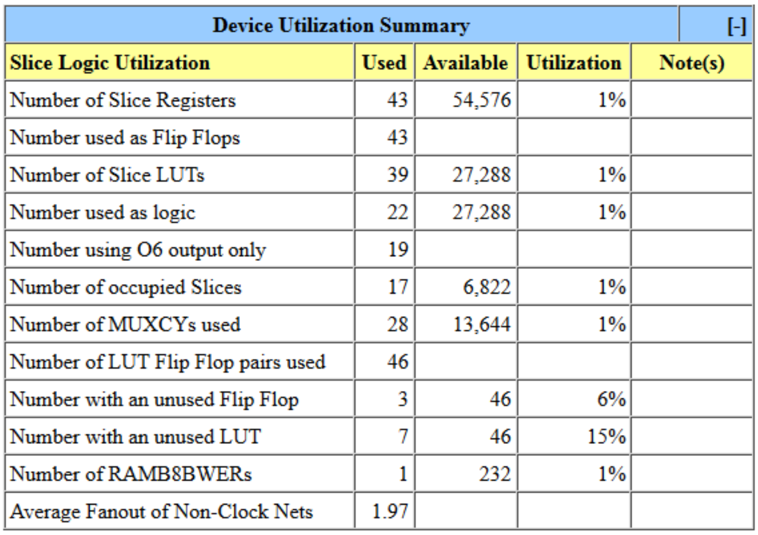


Figure 4. Simulation result displayed in wave scope



Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	43	54,576	1%	
Number used as Flip Flops	43			
Number of Slice LUTs	39	27,288	1%	
Number used as logic	22	27,288	1%	
Number using O6 output only	19			
Number of occupied Slices	17	6,822	1%	
Number of MUXCYs used	28	13,644	1%	
Number of LUT Flip Flop pairs used	46			
Number with an unused Flip Flop	3	46	6%	
Number with an unused LUT	7	46	15%	
Number of RAMB8BWERs	1	232	1%	
Average Fanout of Non-Clock Nets	1.97			

Figure 5. The design summary of the BPSK modulator [18]

4.2. The second method: design of BPSK modulator using verilog HDL

The BPSK Modulator that we implemented on the Atlys Kit board has, as a model, like showing in Figure 6. The carrier is generated internal; let’s focus on the ATLYS voice output this time, based on the installation and use of the tools we have experienced so far, and the simple Verilog programming. After a short pause, in order to output the sound from the ATLYS audio output terminal (black headphone jack), it is necessary to pilot the AIC control, in fact, an electronic component LM4550B (National Semiconductor ac97 audio codec). Specifically, driving is to do data according to the AIC data format and send it to AIC. First, the initialization with functional data, then the repeated transmission of digital audio data, the sound will be transmitted from the AIC to the headphones. This digital data output function as the sound is "DA conversion of the audio signal (digital-to-analog conversion)". The AIC Block Diagram and the way is chosen for generating sine wave is illustrated in Figure 7.

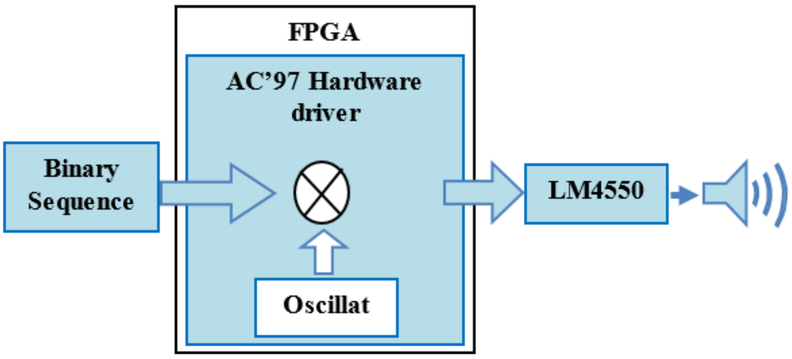


Figure 6. The audio digital analogue BPSK system

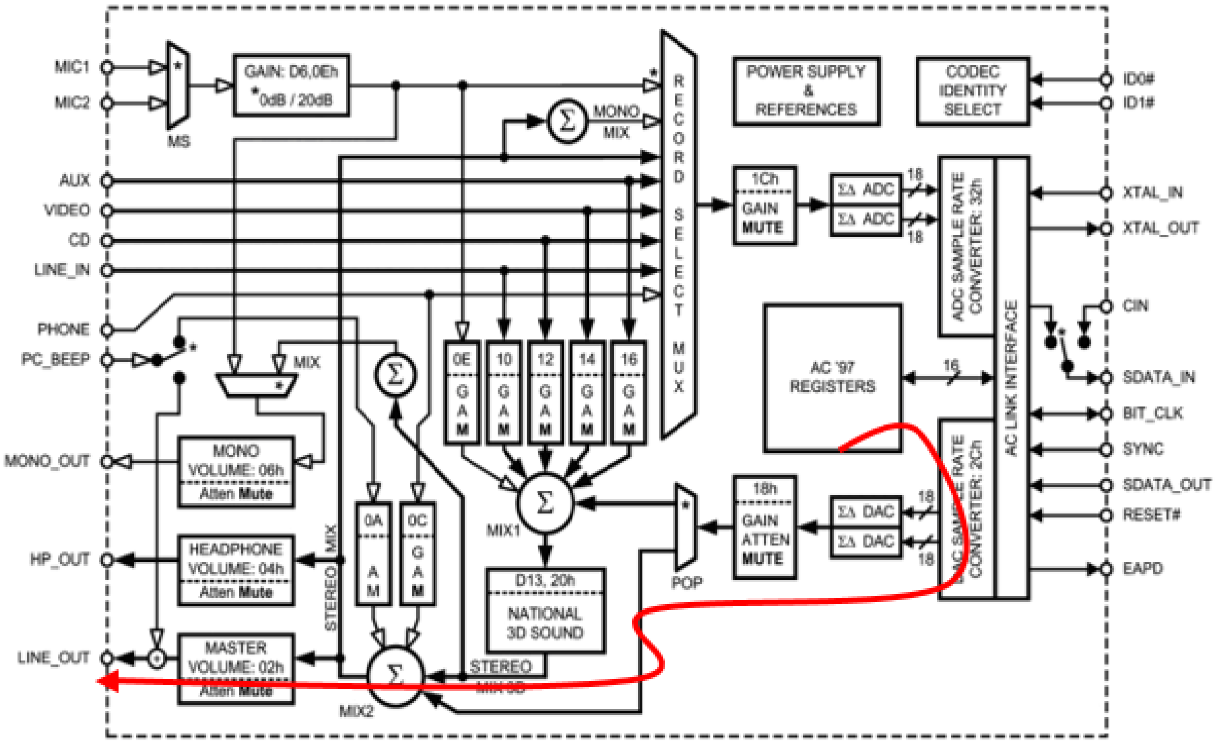


Figure 7. The AIC block diagram [19] and the way chosen for generate sine wave

It is a 440 Hz sine wave that appears frequently in physical and electrical waves and vibrates 440 times per second. Sine waves are the basis of sound, music, physics, and electricity. The modulating signal is generated external by data source DCS 297A and data format DCS 297 Bentered in the FPGA with the help of a Pmod [20], This digital signal is then multiplied with the recovered carrier, generated internal in a audio codec, finally the BPSK signal get out from the black headphone jack, The functionality of this digital modulator was demonstrated through experimental measurements of the real-time modulated signal via an oscilloscope.

Using Xilinx ISE 14.5 [21], BPSK modulated signal can be created. Figure 8 shows the proposed RTL block diagram in FPGA. In this BPSK modulator, when the clock signal (aud_bit_clk) is applied then SINE_WAVE generate the carrier sinusoidal signal (A sin (2πft)) whose frequency is 400 Hz and initial phase is 0º represented by SINE_WAVE. Similarly, INV block inverts the carrier sinusoidal signal (A sin (2πft + π)) whose frequency is 400 Hz and initial phase is 180º. So the output of SINE_WAVE and INV block are same but they are out of phase. The M2_1 block implements a multiplexer. Its one select input (sel) is modulating signal this signal selects an output of SINE_WAVE or INV as an input signal of the multiplexer. The output of the multiplexer (OUT_1) is the BPSK Modulated signal and it is measured with an Oscilloscope.

Figure 9 illustrates the design summary which appears the employment of flip-flops, LUTs, slices used from the ability of the FPGA from the Atlys board [22]. The experimental results were carried out in a laboratory using Atlys Starter Kit Board and a digital oscilloscope, as shown in Figure 10. The yellow signal injected on the Pmod pin (modulating signal) and the blue signal (BPSK signal) gets out from the black headphone jack [23].

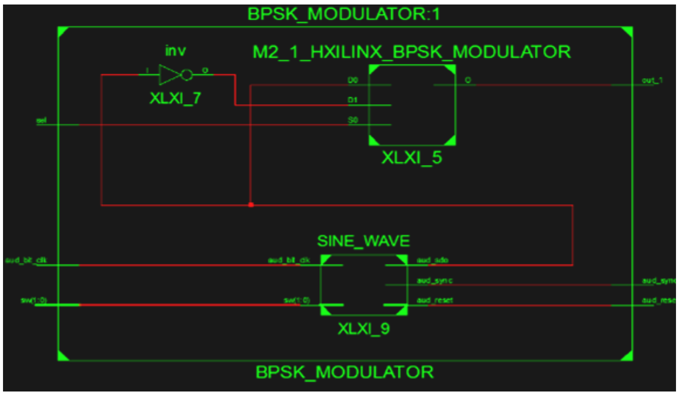
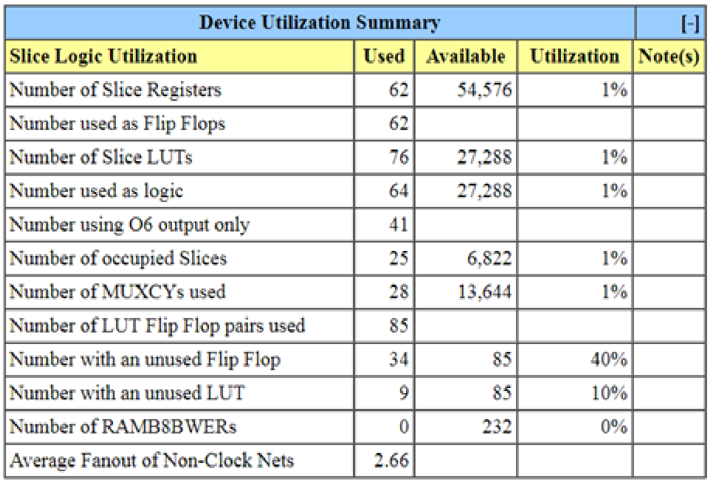


Figure 8. RTL Schematic of BPSK Modulator System [24]



Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	62	54,576	1%	
Number used as Flip Flops	62			
Number of Slice LUTs	76	27,288	1%	
Number used as logic	64	27,288	1%	
Number using O6 output only	41			
Number of occupied Slices	25	6,822	1%	
Number of MUXCYs used	28	13,644	1%	
Number of LUT Flip Flop pairs used	85			
Number with an unused Flip Flop	34	85	40%	
Number with an unused LUT	9	85	10%	
Number of RAMB8BWERs	0	232	0%	
Average Fanout of Non-Clock Nets	2.66			

Figure 9. The design summary of the BPSK modulator

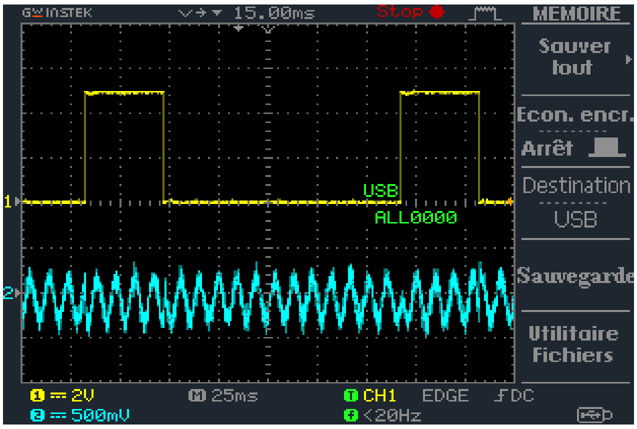


Figure 10. Represents the test bench lab used in implementing the BPSK modulator on the atlys starter kit board

5. COMPARISON RESULTS

The results clearly show that the first method it's implementing BPSK system in FPGA using the Xilinx system generator is comparing the design summary obtained with other works in this field [2-6] the logic utilization of the board was lower in terms of the slice flip-flops and LUTs used. These entire make the design suitable in terms of propagation, implementation and logic utilization of the Atlys boards used in this work.

Although System Generator has an option to generate the VHDL code, but it remains difficult to read this I find it problematic, also the inability to exploits the different components integrated in the Atlys board for example the National semiconductor LM4550 AC '97 audio codec, By Matlab/ Simulink environment alone without return to ISE and write the code his own is the second problematic. The second method of implementation is to use the Verilog HDL language directly to access and control the AC '97 audio codec to provide a sinusoidal signal. What is considered the carrier and also programming a Pmod pin as a digital input is a modulating signal, and programming the AIC output from so give a BPSK signal.

If we comparing the summary of the design of this method by the first one in finding that the resources consume a little larger, also the time to develop is great but the advantage of this method is that the system works autonomously with results real; reliable and fast, their code is easy to read [25].

6. CONCLUSION AND FUTURE WORK

At this work, two new methods of implementation BPSK modulator are developed and compared, in terms of efficiency, duration of development and how many resources are used in FPGA. The first is the implementation of a BPSK modulatorwas done entirely in VHDL with the assistance of Xilinx SystemGenerator tool as in several papers, simply using a one DDS block and a rivers block, which creates a phase shift of 180° between the two sinusoidal. The second method, the implementation wasdone entirely in Verilog HDL without the help of Xilinx SystemGenerator or DSP Builder tools, where I managed to use AC '97 audio codec to generate a sinusoidal signal (carrier), I programmed the Pmod pin as a digital input (the modulating signal) and BPSK signal, I extracted it by programming the AIC as analog output.

After comparing the two methods I concluded that, for performance improvement in terms of area, power and delay, it is good for own digital design. For that require learning VHDL / Verilog / System Verilog is good one. In future work, the author studies the implementation of other digital modulators such as higher-order QAMs. With other method of implementation like the use MicroBlaze Hardware Design.

REFERENCES

[1] Source, J. Crocket, "DSP Architectures for Wireless Communications," *1998 International Symposium on Advanced Radio Technologies. And Krishna Yarlagadda, DSP Architectures for Communications*, 2003.

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

[3] S. O. Popescu, A. S. Gontean and G. Budura, "BPSK System on Spartan 3E FPGA," *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, pp. 301-306, 2012.

[4] Silvana POPESCU, Aurel Gontean, Georgeta Budura, "Hardware Co-Simulation of the BPSK and QPSK Systems on FPGA," *IFAC Proceedings Volumes*, vol. 45, no. 7, pp. 299-304, 2012.

[5] B. K. V. Prasad, R. Sai Priya, "Implementation and Reconfiguration of Basic Digital Modulation Design Models," *Journal of Theoretical and Applied Information Technology*, vol. 90, no. 1, pp. 208-218, 2016.

[6] Mittal, S., Gupta, S., Dasgupta, S., "System Generator: The State-of-art FPGA Design tool for DSP Application," *Proceedings of the 3rd International Conference on Embedded Systems, Mobile Communication and Computing*, India, 2008.

[7] B. Sklar, "Digital Communications-Fundamentals and Applications," *Prentice Hall*, 2001.

[8] S. O. Popescu, G. Budura, A. S. Gontean, "Review of PSK and QAM-Digital Modulation Techniques on FPGA," *2010 International Joint Conference on Computational Cybernetics and Technical Informatics*, Timisoara, pp. 327-332, 2010.

[9] F. Xiong, "Digital Modulation Techniques," *Artech House*, UK, 2000.

[10] 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 (SACI 2011)*, Romania, pp.455-458, 2011.

[11] "System Generator for DSP Getting Started Guide," 2009. [Online]. Available: www.xilinx.com, UG639 (v11.4).

[12] "System Generator for DSP User Guide," 2009. [Online]. Available: www.xilinx.com, p. 14, UG639 (v11.4).

[13] Peter Alfke, "Application Note, Efficient Shift Registers, LFSR, Counters, and Long Pseudo-Random Sequence Generators," Available: www.xilinx.com XAPP 052 July 7, 1996 (Version 1.1).

[14] Xilinx Company, "Xilinx logicore ip dds compiler v4.0 datasheet, 2011. [Online]. Available: www.xilinx.com/support/documentation/ipdocumentation/dds ds558.pdf.

[15] Ken Chapman, "Application Note: Spartan-6 Family, Virtex-6 Family, 7 Series FPGAs, Multiplexer Design Techniques for Datapath Performance with Minimized Routing Resources," [Online]. Available: www.xilinx.com, XAPP522 (v1.2).

[16] "System Generator for DSP Getting Started Guide," 2009. [Online]. Available: www.xilinx.com, page 185, UG639 (v11.4).
[17] "RTL and Technology Schematic Viewers Tutorial," 2011. [Online]. Available: www.xilinx.com, UG685 (v13.1).
[18] "ISE In-Depth Tutorial," 2011. [Online]. Available: www.xilinx.com, Chapter 6: Design Implementation.
[19] "LM4550B datasheet Texas Instruments," 2006. [Online] Available: https://datasheetspdf.com/pdffile/685847/NationalSemiconductor/LM4546B/1
[20] Atlys™ FPGA Board Reference Manual, www.digilentinc.com, Expansion Connectors, p. 17, Revised April.
[21] "ISE Design Suite 14: Release Notes", Installation, and Licensing, UG631, vol. 14, no. 7, 2013.
[22] "Atlys™ FPGA Board Reference Manual", www.digilentinc.com, Revised April 11, 2016.
[23] "Atlys™ FPGA Board Reference Manual", www.digilentinc.com, Audio (AC-97), p. 12, Revised April 11, 2016.
[24] "ISE Quick Start Tutorial," [Online]. Available: www.xilinx.com, page 28, Verification of Synthesis, View RTL Schematic.
[25] "Aaron Marcus, Design, User Experience, and Usability: Theories, Methods, and Tools for Designing he User Experience", Third international conference, DUXU 2014, Part I.