International Journal of Computational and Electronics Aspects in Engineering
Vol. 1, Issue. 1, Dec 2014, Revised in March 2020, pp 35-39
https://doi.org/10.26706/ijceae.1.1.20141207

FPGA Implementation of ASK and FSK Modulator Based on Matlab/Xilinx System Generator

Kaliprasanna Swain[1]
kaleep.swain@gmail.com

Manoj Kumar Sahoo[2]
mksahoo@hotmail.com

Gandhi Institute for Technological Advancement (GITA) /ECE, Bhubaneswar, Odisha, India

***Abstract*: This paper presents an efficient approach for the implementation of typical communication structures studied in class. This scheme is beneficial where the objective is to implement the physical working of complex DSP or communication structures or algorithms without requiring detailed knowledge of hardware design and hardware description languages. The approach is based on the Xilinx System Generator for DSP tool, which integrates itself with the MATLAB, based Simulink Graphics environment and relieves the user of the textual HDL programming. The purposed design is the two independent ASK and FSK modulator. The ASK and FSK modulator are then simulated using Matlab/ Simulink environment and System Generator, FPGA design as well as implemented on a Spartan 6 (xc6slx16-3csg324/Nexys3) Kit board. The modulator algorithm has been implemented on FPGA using the VHDL language on Xilinx ISE Design suite 13.2.**

***Index Terms*—** Xilinx, FPGA, Spartan-6, System Generator and ASK, FSK modulator.

I. INTRODUCTION

Digital modern technology is an important content of modern communication. Baseband digital signal contains a wealth of low-frequency part in digital communication systems. If you want long-distance transmission especially in the limited frequency bandwidth transmission channel, you must carry on the carrier modulation to the digital signal, so that the power spectrum of the baseband signal moves to a higher carrier frequency. This is called the digital modulation [1].

In ASK (amplitude shift keying) strength of carrier signal is varied to represent binary 1 or 0 with both frequency & phase remain constant while amplitude changes where, commonly, one of the amplitudes is zero.

ASK is used to transmit digital data over optical fiber.

Mathematically,

$$S(t) = \begin{cases} A_0 \cos(2\pi f_c t), \text{ binary 0} \\ A_1 \cos(2\pi f_c t), \text{ binary 1} \end{cases}$$

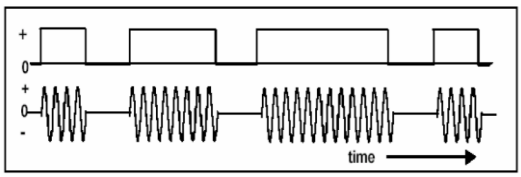

Figure. 1 ASK signal (below) and the message (above)

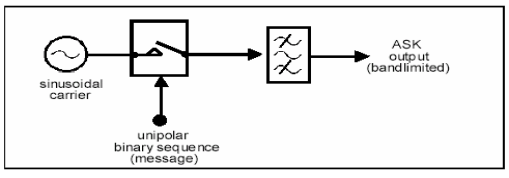

Figure. 2 ASK generation method

In FSK, frequency of carrier signal is varied to represent binary 1 or 0 with peak amplitude & phase remains constant during each bit interval. FSK is less susceptible to errors than ASK, receiver looks for specific frequency changes over a number of intervals, so voltage (noise) spikes can be ignored.

Technical Article
First Online on – 30 Dec 2014, Revised on – 30 March 2020

Cite this article – Kaliprasanna Swain and Manoj Kumar Sahoo, "FPGA Implementation of ASK and FSK Modulator Based on Matlab/Xilinx System Generator", *International Journal of Computational and Electronics Aspects in Engineering*, RAME Publishers, vol. 1, issue 1, pp. 35-39, 2014, Revised in 2020.
https://doi.org/10.26706/ijceae.1.1.20141207

Mathematically,

$$S(t) = \begin{cases} A\cos(2\pi f_1 t), \text{ binary 0} \\ A\cos(2\pi f_2 t), \text{ binary 1} \end{cases}$$

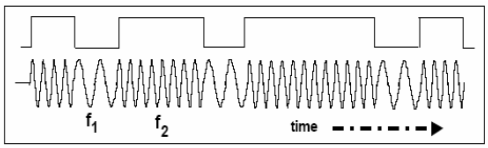

Figure. 3 FSK signal (below) and the message (above)

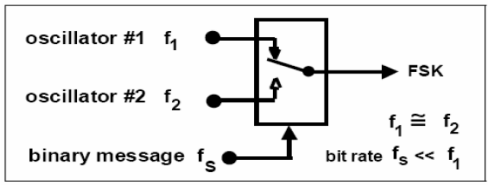

Figure. 4 FSK generation method

The paper is organized into 6 sections. The paper begins with an introduction of ASK and FSK modulation in section 1. Section 2 presents the overview of Xilinx system generator, implementation of the ASK and FSK system in System Generator are presented in section 3. Section 4 is dedicated to the implementation of the system: modulator on the Spartan 6 Nexys 3 Kit. The results are discussed in section 5. The final section 6, presents the conclusions.

II. Overview Of The Xilinx System Generator

The Xilinx System Generator [2-4] for DSP is a system level modeling and design tool that facilitates FPGA design and has the ability to work at a higher level of abstraction. It enables the use of the MathWorks graphical model based Simulink design environment for FPGA design. The System Generator integrates itself with Simulink and FPGA designs are captured by using the Xilinx specific Blockset. Thus, designing a hardware model in Simulink is as simple as designing any other Simulink model with the only difference being the use of Xilinx Blockset instead of those found in Simulink. The System Generator provides many DSP building blocks in the form the Xilinx DSP Blockset for the Simulink environment. The variety in this Blockset ranges from common DSP blocks such as adders, multipliers, registers etc to more complex blocks such as FFTs, filters, memories, forward error correction etc. Thus, previous experience with low level system design and HDLs is not required when using this tool. The System Generator uses the Xilinx ISE software and IP core generators to convert a designed model into the equivalent HDL code. The remaining FPGA implementation steps including synthesis, place and route, etc. are automatically performed to generate a bit file that is downloaded on to the FPGA.

III. Implementation Of The Ask And Fsk System In System Generator

Figure 5 and 7 illustrate the implementation of a ASK and FSK Modulator using System Generator [5] [6] tools in Simulink respectively.

The System Generator Blockset [7] contains RTL schematic is generated and resource utilization is checked as shown in figure. 4 and 5.

- the gateway in blocks: the inputs into the Xilinx portion of the Simulink design;
- the gateway out blocks: the outputs from the Xilinx portion of the Simulink design;
- 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.

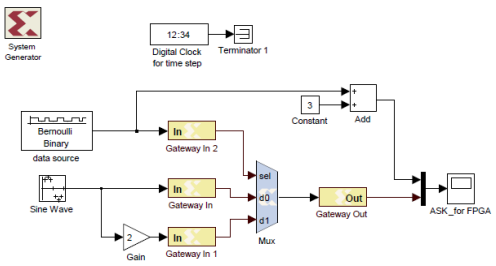

Figure. 5 ASK generation with Simulink / Xilinx System Generator

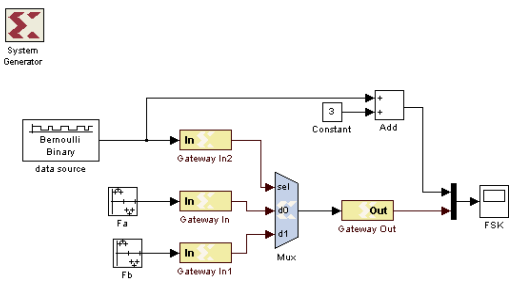

Figure. 6 FSK generation with Xilinx System Generator

Figure 7 and 8 correspond the output signal waveform generated by the scope [7].

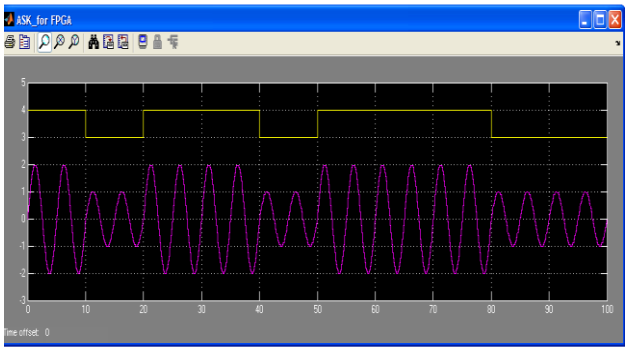

Figure. 7 Simulation result of ASK by Xilinx System Generator Method

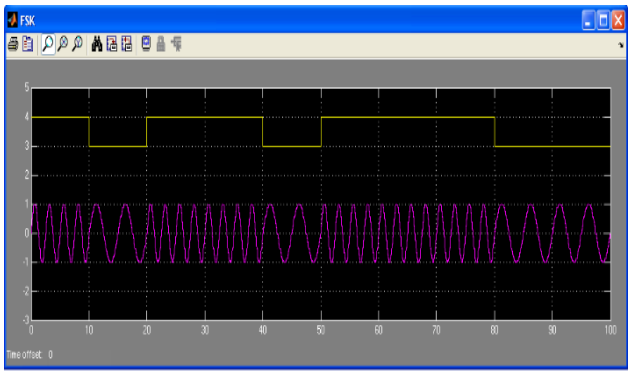

Figure. 8 Simulation result of FSK Xilinx System Generator Method

IV. ASK AND FSK IMPLEMENTATION

The ASK and FSK modulators are implemented on SPARTAN-6 NEXYS3 board[8]. The Nexys3 is a complete, ready-to-use digital circuit development platform based on the Xilinx Spartan-6 LX16 FPGA. The Spartan-6 is optimized for high performance logic, and offers more than 50% higher capacity, higher performance, and more resources Spartan-3 500E FPGA. Nexys3 is compatible with all Xilinx CAD tools, including ChipScope, EDK, and the free WebPack. Spartan-6 LX16 features include: 2,278 slices each containing four 6- input LUTs and eight flip-flops, 576Kbits of fast block RAM, two clock tiles (four CMs & two PLLs), 32 DSP slices, 500MHz+ clock speeds.

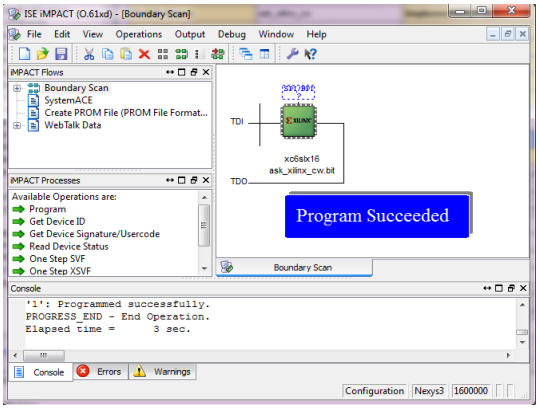

Figure. 9 Configuration mode for FPGA for (ASK)

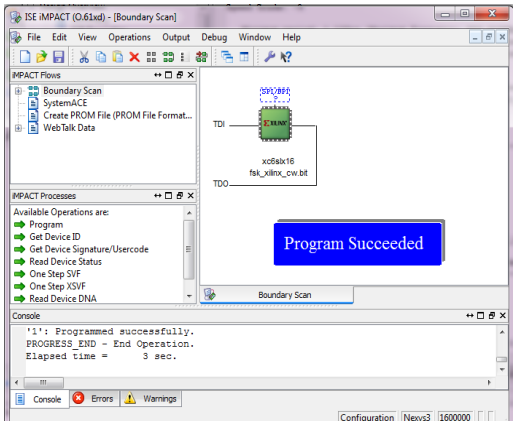

Figure. 10 Configuration mode for FPGA (FSK)

V. RESULT

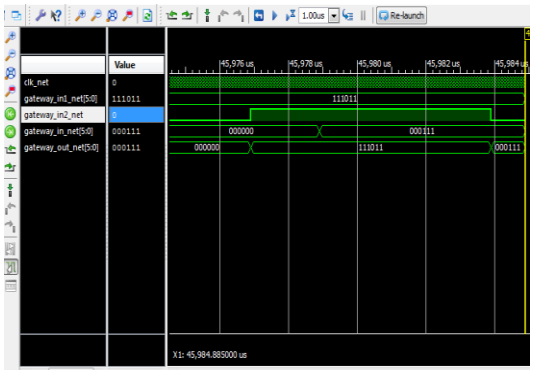

Figure. 11 Behavioral simulation of ASK with 2 step model

ask_xilinx_cw Project Status (06/07/2013 - 14:23:00)			
Project File:	ask_xilinx_cw.xise	**Parser Errors:**	No Errors
Module Name:	ask_xilinx_cw	**Implementation State:**	Programming File Generated
Target Device:	xc6slx16-3csg324	**•Errors:**	No Errors
Product Version:	ISE 13.2	**•Warnings:**	23 Warnings (23 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	0 (Timing Report)

Figure. 12 Design Summery of Project for ASK

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	0	18,224	0%	
Number of Slice LUTs	3	9,112	1%	
Number used as logic	3	9,112	1%	
Number using O6 output only	0			
Number using O5 output only	0			
Number using O5 and O6	3			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	3	2,278	1%	
Number of LUT Flip Flop pairs used	3			
Number with an unused Flip Flop	3	3	100%	
Number with an unused LUT	0	3	0%	
Number of fully used LUT-FF pairs	0	3	0%	
Number of slice register sites lost to control set restrictions	0	18,224	0%	
Number of bonded IOBs	19	232	8%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	

Figure. 13: Design summery of Device Utilization for ASK

```
Speed Grade: -3

  Minimum period: 1.128ns (Maximum Frequency: 886.643MHz)
  Minimum input arrival time before clock: No path found
  Maximum output required time after clock: No path found
  Maximum combinational path delay: 5.550ns

================================================================================
Timing constraint: TS_clk_2ec59898 = PERIOD TIMEGRP "clk_2ec59898" 10 nS HIGH 5 nS
  Clock period: 1.128ns (frequency: 886.643MHz)
  Total number of paths / destination ports: 1 / 1
  Number of failed paths / ports: 0 (0.00%) / 0 (0.00%)
--------------------------------------------------------------------------------
Slack:                  8.872ns
  Source:               persistentdff_inst/q (FF)
  Destination:          persistentdff_inst/q (FF)
  Data Path Delay:      1.128ns (Levels of Logic = 1)
  Source Clock:         clk rising at 0.000ns
  Destination Clock:    clk rising at 10.000ns

  Data Path: persistentdff_inst/q (FF) to persistentdff_inst/q (FF)
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q              1   0.447   0.579  q (q)
     end scope: 'persistentdff_inst:q'
     begin scope: 'persistentdff_inst:d'
     FD:D                     0.102          q
    ----------------------------------------
    Total                     1.128ns (0.549ns logic, 0.579ns route)
```

Figure. 14 Timing Report of ASK

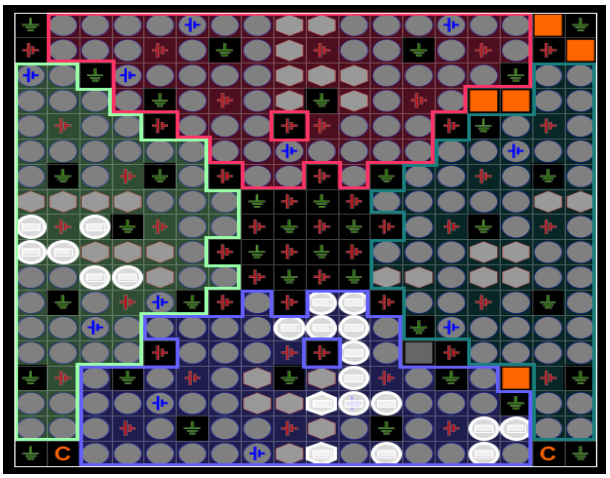

Figure. 15 Xilinx PACE device manager, the light color are used pin for Xilinx Spartan -6 device Result for FSK

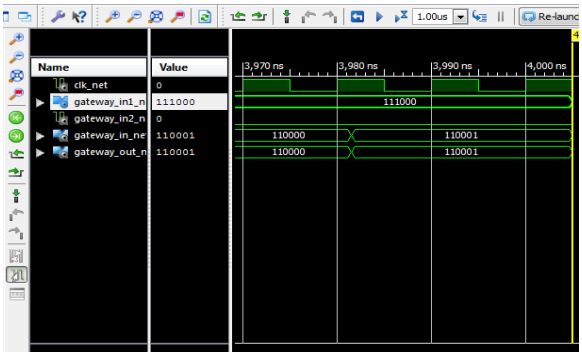

Figure. 15 Behavioral simulation of FSK with 2 step model

fsk_xilinx_cw Project Status (06/07/2013 - 14:02:17)			
Project File:	fsk_xilinx_cw.xise	**Parser Errors:**	No Errors
Module Name:	fsk_xilinx_cw	**Implementation State:**	Programming File Generated
Target Device:	xc6slx16-3csg324	**• Errors:**	No Errors
Product Version:	ISE 13.2	**• Warnings:**	28 Warnings (7 new, 0 filtered)
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Figure. 16 Design Summery of Project (FSK)

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	0	18,224	0%	
Number of Slice LUTs	3	9,112	1%	
Number used as logic	3	9,112	1%	
Number using O6 output only	0			
Number using O5 output only	0			
Number using O5 and O6	3			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	2	2,278	1%	
Number of LUT Flip Flop pairs used	3			
Number with an unused Flip Flop	3	3	100%	
Number with an unused LUT	0	3	0%	
Number of fully used LUT-FF pairs	0	3	0%	
Number of slice register sites lost to control set restrictions	0	18,224	0%	
Number of bonded IOBs	20	232	8%	
Number of LOCed IOBs	20	20	100%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	

Figure. 17 Design summery of Device Utilization(FSK)

```
Speed Grade: -3

  Minimum period: 1.128ns (Maximum Frequency: 886.643MHz)
  Minimum input arrival time before clock: No path found
  Maximum output required time after clock: No path found
  Maximum combinational path delay: 5.550ns

================================================================================
Timing constraint: TS_clk_9f243137 = PERIOD TIMEGRP "clk_9f243137" 10 nS HIGH 5 nS
  Clock period: 1.128ns (frequency: 886.643MHz)
  Total number of paths / destination ports: 1 / 1
  Number of failed paths / ports: 0 (0.00%) / 0 (0.00%)
--------------------------------------------------------------------------------
Slack:                  8.872ns
  Source:               persistentdff_inst/q (FF)
  Destination:          persistentdff_inst/q (FF)
  Data Path Delay:      1.128ns (Levels of Logic = 1)
  Source Clock:         clk rising at 0.000ns
  Destination Clock:    clk rising at 10.000ns

  Data Path: persistentdff_inst/q (FF) to persistentdff_inst/q (FF)
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q              1   0.447   0.579  q (q)
     end scope: 'persistentdff_inst:q'
     begin scope: 'persistentdff_inst:d'
     FD:D                     0.102          q
    ----------------------------------------
    Total                     1.128ns (0.549ns logic, 0.579ns route)
```

Figure. 18 Timing Report of FSK

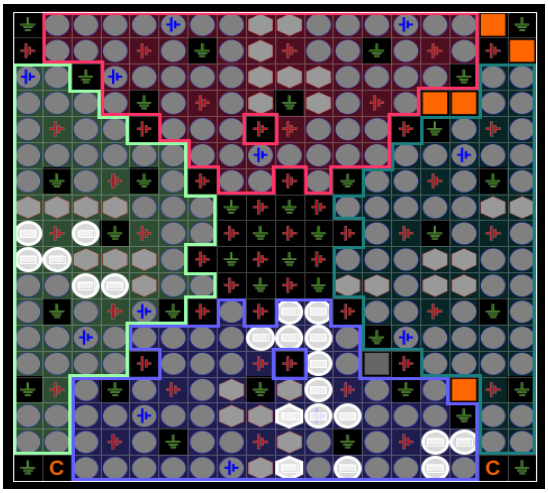

Figure. 19 Xilinx PACE device manager, the light color are used pin for Xilinx Spartan -6 device (FSK)

VI. Conclusion

The use of the Xilinx System Generator tool for DSP education and research is presented. It is shown that this tool is ideal for developing FPGA based hardware without the requirement of learning HDLs and Hardware Design.

References

[1] H.Nguien, E.Shwedyk, "A First Course in Digital Communications",Cambridge University Press, 2009.

[2] S. Athar, M.A.Siddiqi and S. Masud, "Teaching and research in fpga based digital signal processing using xilinx system generator", ICASSP 2012, 978-1-4673-0046-9/12/ ©2012 IEEE.

[3] S.O. Popescu, A.S.Gontean and D.Ianchis, "QPSK Modulator on FPGA", SISY 2011, 2011 IEEE 9th International Symposium on Intelligent Systems and Informatics, September 8-10, 2011, Subotica, Serbia.

[4] 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.

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

[6] ISE 13.2, Quick Start Tutorial, Xilinx

[7] Xilinx System Generator for DSP User Guide, r10.1.1, April 2008

[8] Nexys3™ Board Reference Manual Revision: December 28, 2011, www.digilentinc.com