

# A Reconfigurable Model-Based Design for Rapid Prototyping on FPGA

Syed Jahanzeb Hussain Pirzada, Abid Murtaza, Tongge Xu, and Liu Jianwei

**Abstract**—The digital design methodologies are evolving with the increase of digital systems utilization in daily life. The Model Based Design (MBD) methodology provides a unique methodology for design and implementation of digital systems on Field Programmable Gate Array (FPGA). Recently, a lot of research effort has been put to exploit new methodologies for designing and prototyping of digital systems on FPGA. The FPGA hardware provides prototyping which provides means of verifying your design at an early stage of development cycle. This helps to evaluate design trade-offs by testing the design in real-time on hardware. Making prototypes is a common practice in research-oriented projects. However, it requires excess development time which increases time to market of the product. This paper illustrates the use of reconfigurable MBD for rapid prototyping of digital systems on Microsemi ACTEL FPGAs for improving the design-cycle and time-to-market of a product. The model is simulated to verify the functionality of the design at system-level and a high-level code is generated from the MBD toolset embedded in MATLAB for hardware implementation. Then, a High-Level Synthesis (HLS) is performed on the generated code which converts this high-level code into Verilog-HDL suitable for hardware implementation on FPGA. Hence, this work presents a methodology and its analysis for design of digital system using high-level synthesis on Microsemi ACTEL FPGA.

**Index Terms**—Model-based design, binary phase shift keying, high-level synthesis, field programmable gate array.

## I. INTRODUCTION

The digital design methodologies are evolving with the increase of digital systems utilization in daily life. In rapidly changing digital system design world, all the methods and technological advanced platforms are utilized mixed language platforms for design and implementation. The model-based or building block based design is most popular as it provides rapid prototyping facility. The matrix Laboratory (MATLAB) software provides interface with many hardware programming devices. These interfaces provided by MATLAB made it possible for designers to perform rapid prototyping by using pre-developed codes or by converting the MATLAB code to the hardware specific programming language. For instance the MATLAB software provides interface for Texas Instrument Digital Signal

Processor (DSP) [1], [2] for design of digital systems. Similarly the MATLAB software provides the interfaces for different vendors FPGA [3]-[5] for prototyping. Besides interfacing FPGAs with MATLAB require conversion of MATLAB files into Verilog Hardware Description Language (HDL) [6], [7].

The digital systems are defined through Hardware Description Languages (HDL) or using some high-level language for a specific hardware platform, e.g. a microcontroller or a microprocessor. The Model-based Design (MBD) refers to the use of blocks or diagrams instead of code [8]. These fundamental blocks vary in complexity from a register or a delay element to random bit generator to more advanced blocks like a Direct Digital Synthesizer (DDS) capable of generating Sine / Cosine discrete values for a given frequency and sampling rate. These blocks are provided with configurable specification options to suit designer's needs by the software itself, e.g. Simulink (MATLAB). However, more than just ease of graphical environment to connect basic modules / blocks, the MBD is a paradigm shift for system level development in which the system is built in a bottom-up approach to architect a versatile and complex algorithm, e.g. a control algorithm, signal processing, or image processing algorithms [8].

In case of designing the blocks in FPGAs are implemented using a High-Level Synthesis tool integrated with the MBD software, for example the Symphony High Level Synthesis (SHLS) tool for Microsemi-ACTEL FPGA [9]. Similarly, the Xilinx FPGA provides tools for MBD [10], [11]. Previously, researchers have worked on utilizing the tools in embedded with MATLAB software for Xilinx FPGAs for prototyping on Xilinx FPGAs. S. Kotel *et al.* [12] presented his idea of utilizing the MATLAB embedded tool for video processing. S. Chhabra *et al.* [13] presented the hardware and software co-simulation for image processing application. The research performed were mainly focused on utilization of Xilinx FPGA tools embedded in MATLAB software for prototyping. In this paper, we focus the implementation of a Binary Phase Shift Keying (BPSK) modulator using MATLAB software and SHLS software for Microsemi ACTEL FPGA. The idea is to show the ease of design and implementation of a complex algorithm using the MBD methodology and contrasts it with conventional RTL coding using HDL.

The paper is structured as follows. The previous work is in Section II. Section III provides an introduction to BPSK modulator and the proposed implementation. Section IV provides experimental results. Finally, Section V concludes the paper.

Manuscript received October 10, 2019; revised January 3, 2020. This work was supported by the 2018 Industrial Internet Innovation and Development Project of the Ministry of Industry and Information Technology (no. IIIDP-9.1-2018 of MIIT).

S. J. H. Pirzada, T. Xu, and L. Jianwei are with the School of Cyber Science and Technology, Beihang University, Beijing, China (e-mail: jahanzebp@hotmail.com, xutg@buaa.edu.cn, liujianwei@buaa.edu.cn).

A. Murtaza is with the School of Electronics and Information Engineering, Beihang University, Beijing, China (e-mail: abid\_murtaza47@hotmail.com).

## II. PREVIOUS WORK

Design and implementation of digital modulators in reconfigurable hardware such as FPGAs has always been a hot topic not only due to the low complexity of the algorithm. There were many researchers who implemented modulators using VHDL [14], [15] which involve great effort to tackle the timing and synchronization of individual modules along with the verification of the overall design.

Previously, F. Ferrandi *et al.* [16] presented Xilinx design flow for IP core generation. The implementation for IP-core generation provides the techniques utilized for generation of IP cores using hardware software co-design. Then, there have been vendor specific implementations with a graphical interface [17], [18] in which Xilinx devices have been targeted. In this paper, we target Microsemi-ACTEL FPGAs suitable for space communications [19] and provide hardware results for BPSK modulator with constant signal to noise ratio which we could not find in existing implementation. However, the basic idea is to show the ease of system design and modification in response to design changes. This paper presents the first implementation of the MBD for prototyping using Microsemi-ACTEL FPGA SHLS tools with MATLAB software.

The MBD implementation of digital down converter using the MATLAB software and simulating it on Modelsim software is presented for its effectiveness on FPGA [20]. The MBD for software define radio is presented by X. Cai *et al.* [21]. The implementation has been focused on utilization of Xilinx FPGA tools for designing of software define radios.

## III. THE BPSK MODULATOR PROPOSED IMPLEMENTATION

The BPSK is a digital modulation technique in which the phase of the carrier (reference signal) is modulated to distinct phases/angles. However, as the name suggests, it uses two phases which are separated by  $180^\circ$ . The BPSK is also called as 2-PSK or Phase Reversal Keying. The Fig. 1 shows the waveform generated from BPSK modulation. The working is explained by the fact that it has one fixed phase, say  $0^\circ$ , when the binary data is at one level, say logic level 0, and when the data is at the other level say logic 1, phase is reversed or set to  $180^\circ$ . Though, the resulting phase corresponding to the logic level is user defined, a binary 0 is  $0^\circ$  and a binary 1 is  $180^\circ$ ; the phase changes when the binary state switches so that the signal is coherent. A BPSK modulator system requires a multiplication of sinusoidal wave with the binary data of  $\pm 1$  as shown in Fig. 2.

However, this is an analog implementation of BPSK modulator in which the multiplier has an analog sinusoidal carrier from an oscillator. Such carrier wave generation is not applicable in digital systems where we can only store discrete values. The simplest of Phase Shift Keying methods to digital modulation, yet BPSK is a complex algorithm for implementation in FPGAs using HDL.

The BPSK modulator can be implemented on FPGA using an IP-core based implementation [22], Custom designed HDL-core based implementation, and the MBD implementation [23], [24].

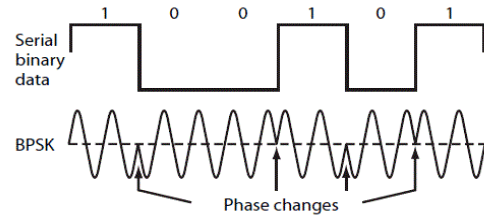


Fig. 1. BPSK waveform.

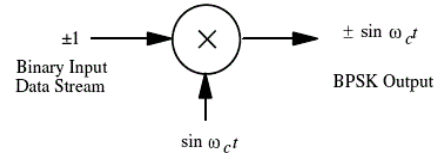


Fig. 2. A BPSK modulator.

### A. Design Using FPGA Vendors IP-Core

Firstly, the IP-cores cannot be utilized for modification of algorithms for research purpose. In communication implementation of BPSK Modulator for implementing modification of algorithm using the IP-core based implementation cannot be performed using IP-cores. The IP-cores cannot be used to conduct further research, and implementation must be done by designing the HDL core by an individual.

### B. Conventional HDL-Core Design for FPGA

While there can be many FPGA implementations using Verilog HDL or VHDL, the basic idea is to generate a carrier wave (Sine or Cosine) of a particular frequency, and multiply it with the incoming data in digital domain. However, only discrete values of carrier can be stored in digital format, which means we need to store samples of sinusoid for a given frequency. Moreover, we also need to know how many samples of the sinusoid are acceptable for a given Analog to Digital Converter (ADC) to produce a smooth waveform. The higher the number of samples in a sinusoid cycle, the better the performance of BPSK modulator at the output of ADC. In digital domain, the generation of a sinusoidal wave is implemented using a CORDIC algorithm or using Look-Up-Table (LUT) based approach.

### C. LUT Based Implementation

The Fig. 3 summarizes LUT based implementation; it involves using phase generation circuit, implemented in the form of accumulator. It uses sinusoid amplitude generation, using LUTs, storing amplitude sample values. It uses digital Sine/Cosine wave generation by cascading phase generation circuit and amplitude generation. It also uses phase reversal, implemented in the form of inverter and multiplexer.

### D. Drawbacks of HDL Based Implementation

The implementation in FPGA using HDL requires synchronization among the modules as well as critical timing of signals due to multiple clock domains, e.g. core clock frequency, sampling frequency and carrier frequency. This results in significant loss of time for verification of the whole circuit, and as the circuit complexity grows, there is exponential rise in verification time. Furthermore, the design is not scalable, e.g. in case we want to make a slight change in

the carrier frequency or increase multiplier precision, we need to change whole look-up table, bit widths, and need to adjust number of samples. Thus a re-design of whole system is inevitable, and new verification cycle waits again. Consequently, we need a better scalable development strategy to encounter such complex problems.

IV. THE EXPERIMENTAL RESULTS

In the MBD, a scalable and easily modifiable modeling and implementation is conveniently achievable by using blocks which have a graphical interface for setting the block or module's parameters. These blocks are pre-verified and can be simulated in one click. The phase generation and waveform convertor in Fig. 3 is collectively called a Direct Digital Synthesizer (DDS), which is the most critical part of BPSK modulator and is readily available as a block in MATLAB. Hence, the carrier generation in digital domain for any carrier frequency, central clock, and any sampling rate is at the fingertips of the designer. Moreover, a designer can also control the number of bits the modulator processes out which is very handy in case the designer intends to increase the modulator's precision and interface with ADC.

A possible equivalent BPSK modulator is designed using the MBD strategy in the Fig. 4. The carrier generation is carried out by a pre-designed available model of DDS. A Bernoulli Binary bit generator is used as an input to the model, which is further, registered (INPUT\_BIT). This defines the input boundary of the system model. The output of multiplexer is a 1 or -1 as dictated by the Bernoulli's bit generation principle. The output of the multiplexer, which is our  $b(t)$  is multiplied with the discrete samples of carrier wave which is our output of modulator. The output is also registered, which marks the output boundary of our model. The SHLS Tool block shows the system is to be synthesized with Symphony HLS, a high-level synthesizer from Synopsis Inc. A Scope serves as an output probe to the system model through which the output of system model can be observed. The simulation results for a BPSK modulator with specifications of data rate of 10 Kbps, carrier frequency of 40 KHz, samples per wave cycle of 40 samples, sampling frequency of 2 MHz, output word length of DDS core 16 bits, and output bits width of multiplier of 12 bits output.

Constraints. The HLS constraints are useful for speed versus area optimization. We used asynchronous reset for the system model to reset the system after an interval of 8 bits time to compensate for imperfect clocks. A tolerance of 0.1% in clock frequency can hugely disturb SNR of BPSK output due to striking of the bit at non-zero crossing of the carrier. The hardware resources utilized for Microsemi-ACTEL ProAsic3E (A3PE1500) are shown in Table I.

TABLE I: HARDWARE RESOURCES

Cell	Count	Area	Count*Area
AND2	9	1	9
AX1	1	1	1
AX1C	1	1	1
DDS	1	10325	10325
DFN1C1	10	1	10
DFN1P1	2	1	2
GND	1	0	0
INV	1	1	1
NOR2B	12	1	12
VCC	1	0	0
XOR2	24	1	24
Total	63		10385

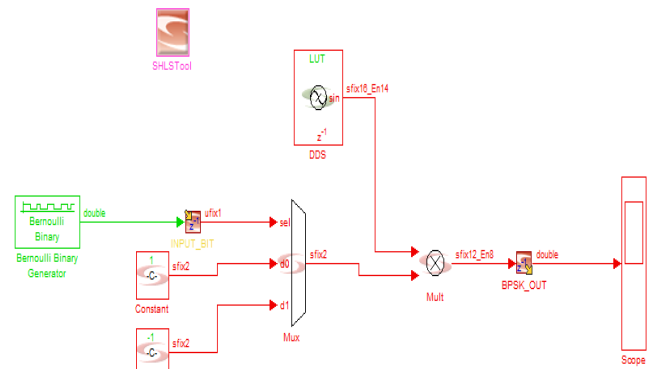


Fig. 4. BPSK Modulator design using Simulink.

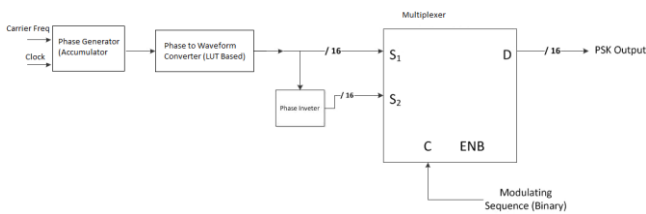


Fig. 3. BPSK waveform.

A. Synthesis Results

Symphony HLS software provides versatile options for system-level optimizations and generating the RTL code for the model, including target device e.g. specific FPGA like Microsemi-ACTEL ProAsic3E (A3PE1500), RTL options for language settings, design options for synchronous / asynchronous reset, clock resets domain, and HLS

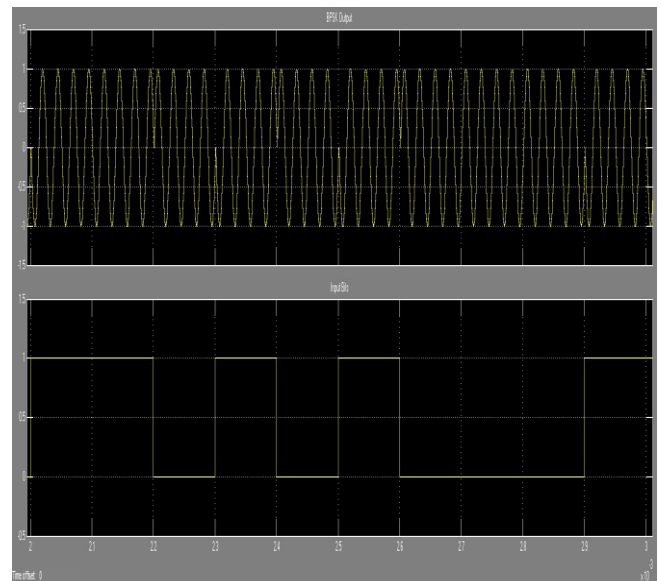


Fig. 5. BPSK output waveform of the System Model.

V. CONCLUSION

The goal of rapid prototyping is to quickly create the system design, verify it and modify if necessary without re-iterating the design cycle and verification step. However, as we see in conventional BPSK modulator design, the design requires cumbersome amount of effort for wave generation in discrete domain, meet the timing requirements of individual modules, and verify the design with non-trivial test benches and covering all test scenarios. A slight modification in the design due to specification changes requires complete re-designing, and verification cycle. Such a redesign step is not needed in the MBD strategy where the modules are not only pre-verified by the software tool but can take any specification, eliminates timing and verification stage, simulates the design without writing test-benches and generating the RTL code for your system, thanks to High-Level Synthesis strategy.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

SP conducted the initial research, concept, and written paper; AM collected the dataset for simulation; TX and LJ has reviewed the article and supervision of whole research; all authors had approved the final version.

ACKNOWLEDGMENT

This work was supported by the 2018 Industrial Internet Innovation and Development Project of the Ministry of Industry and Information Technology (no. IIIDP-9.1-2018 of MIIT).

REFERENCES

[1] Embedded Target for TI TMS320C6000, The MathWorks, Inc, March 2006.

[2] F. A. Melo, M. A. Leonardi, and A. L. Neve, "Digital signal processing with matlab and dsp kits," in *Proc. Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop, 2004*, pp. 15–18.

[3] Xilinx. [Online]. Available: <https://www.xilinx.com>

[4] Microsemi ACTEL. [Online]. Available: <https://www.microsemi.com>

[5] Altera. [Online]. Available: <https://www.intel.com>

[6] "IEEE Standard for Verilog Hardware Description Language," in *IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001)*, pp. 1-590, April 7, 2006.

[7] "IEC/IEEE International Standard - Behavioural languages - Part 1-1: VHDL Language Reference Manual," in *IEC 61691-1-1:2011(E) IEEE Std 1076-2008*, pp.1-648, May 19, 2011.

[8] K. Bender, M. Broy, I. P. éter, A. Pretschner, and T. Stauner, "Model based development of hybrid systems: specification, simulation, test case generation," in *Proc. Modelling, Analysis and Design of Hybrid Systems, LNCIS, 2002*.

[9] Symphony High Level Synthesis Tool (SHLS). [Online]. Available: <https://www.microsemi.com/product-directory/dev-tools/4899-synphony>

[10] Xilinx Inc. Embedded System Tools Reference Manual. [Online]. Available: <http://www.xilinx.com>

[11] Xilinx System Generator User's Guide. [Online]. Available: <http://www.xilinx.com>

[12] S. Kotel, M. Zeghid, A. Baganne, T. Saidani, Y. I. Daradkeh, and T. Rached, "Fpga-based real-time implementation of aes algorithm for video encryption," in *Proc. Recent Advances in Telecommunications, Informatics and Educational Technologies, 2014*, pp. 27–36.

[13] S. Chhabra and K. Lata, "Hardware software co-simulation of

obfuscated 128-bit AES algorithm for image processing applications," in *Proc. IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 2018, pp. 191-194.

[14] C. Erdoğan, I. Myderrizi, and S. Minaeiz, "FPGA implementation of BASK-BFSK-BPSK digital modulators [testing ourselves]," in *Proc. IEEE Antennas and Propagation Magazine*, 2012.

[15] F. Ahamed and F. A. Scarpino, "An educational digital communications project to implement a BPSK detector," *IEEE Trans. Commun.*, vol. 34, no. 5, pp. 423–429, May 2000.

[16] F. Ferrandi, G. Ferrara, R. Palazzo, V. Rana, and M. D. Santambrogio, "VHDL to FPGA automatic IP-core generation: A case study on Xilinx design flow," in *Proc. 20th IEEE International Parallel & Distributed Processing Symposium*, 2006.

[17] S. O. Popescu, A. S.Gontean, and G. Budura, "BPSK system on Spartan 3E FPGA," in *Proc. 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*, 2012.

[18] C. J. Harsha, S. D. Hanwate, and A. S. Mali, "Hardware implementation of BPSK system on Virtex2-Pro FPGA using Xilinx system generator," *Transactions of International Refereed Journal of Engineering and Science*, vol. 2, issue 1, January 2013.

[19] Pro-ASIC3FPGAs. [Online]. Available: <http://www.microsemi.com/products/fpga-soc/radtolerant-fpgas/rt-pro-asic3>

[20] J. Liu, "Model based design workflow for FPGA compliance with DO-254 standard," in *Proc. 2012 International Symposium on Information Technologies in Medicine and Education*, 2012, pp. 1026-1030.

[21] X. Cai, M. Zhou, and X. Huang, "Model-based design for software defined radio on an FPGA," *IEEE Access*, vol. 5, pp. 8276-8283, 2017.

[22] A. Al-Safi and B. Bazuin, "FPGA based implementation of BPSK and QPSK modulators using address reverse accumulators," in *Proc. 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2016, pp. 1-6.

[23] P. Rapaka, V. Babu, and G. J. Chitra, "FPGA implementation of BPSK modem," *Transactions of Int. Journal of Engineering Research and Applications*, vol. 3, issue 6, pp. 1976-1985, 2013.

[24] W. Song, J. Zhang, and Q. Yao, "Design and implement of BPSK modulator and demodulator based on modern DSP technology," in *Proc. 2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2009, pp. 1135-1137.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).



**S. J. H. Pirzada** was born at Attock, Pakistan. In 2007, he received his BE degree in electronics engineering from NED University of Engineering and Technology, Karachi, Pakistan. In 2012, he received an MS degree in electrical, electronics, control and instrumentation engineering from Hanyang University, Seoul, South Korea. Since 2018 he is enrolled for the Ph.D. degree in the School of Cyber Science and Technology at Beihang University, Beijing, China. His research interest is in the field of cryptography for satellite applications.



**A. Murtaza** was born at Karachi, Pakistan. In 2010, he received the M.Sc. electronics degree from the University of Karachi, Karachi, Pakistan. He is working with Pakistan's National Space Agency SUPARCO since 2010. He is currently working towards the Ph.D. degree in space technology applications at Beihang University, Beijing, China. His research interests include information security, space information network, and applications.



**T. Xu** graduated from Beijing University of Aeronautics and Astronautics in 1993 with a master's degree in engineering. He is now associate professor of School of Cyber Science and Technology at Beihang University, Beijing, China. His research areas are network management and flow / protocol analysis technology, UNIX / Linux system development, large information system design and development technology, public opinion big data analysis and mining.



**L. Jianwei** was born at Shandong, China. He received BS and MS degrees in electronics and information engineering from Shandong University, Shandong, China in 1985 and 1988 respectively. He received his Ph.D. degree in electronics and communication systems from Xidian University, Shaanxi, China in 1998. He is now the dean of the School of Cyber Science and Technology at Beihang University, Beijing, China. His current research interests include

wireless communication networks, cryptography, and information and network security.