Efficient FPGA Design Flow Based QAM Modulator Using System Generator and MATLAB

Ph.D.(Lecturer) Majid S. Naghmash, Ph.D.(Lecturer) Khaleel J. Hammadi,

Ph.D.(Lecturer) Ali M. Hammadi

Abstract

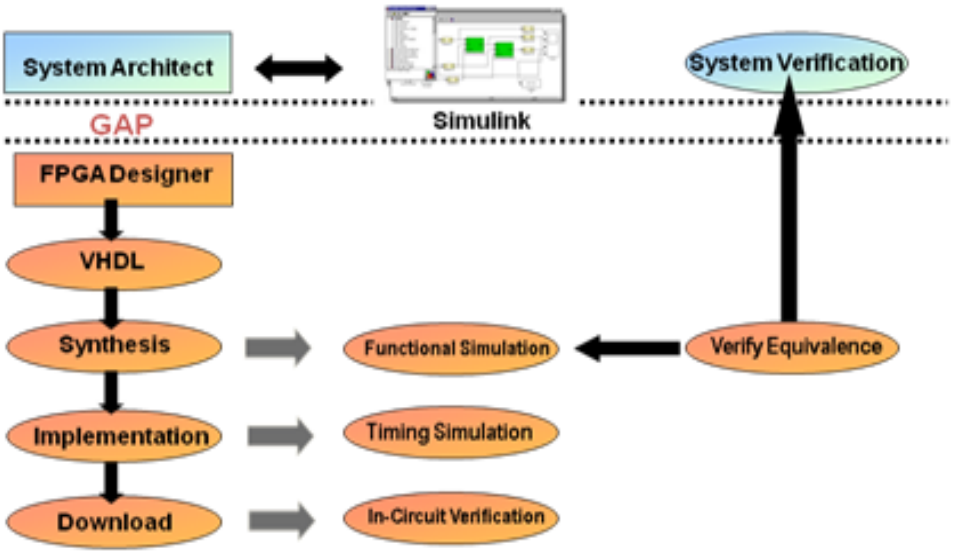
This study presents, efficient FPGA Virtex-5 design flow based QAM modulator using System Generator from Xilinx and MATLAB from mathworks. To allow all digital communication systems for easy adopted with more complicated coding and modulation techniques, the Software Radio (SR) mode has to be used which is extremely in meeting the increasing demands of the wireless communication and mobile industry. However, in this study, the shortest and efficient paths to design an FPGA using MATLAB, Xilinx System Generator, ModelSim, synplify Pro and ISE (Integrated Software Environments) software tools is introduced. The floating point design in MATLAB has been moved to fixed point values using the most attractive and friendly Xilinx DSP system generator software a model based approach associated with assistance software from Mathworks and Synplicity. Result obtained shows an important utilization in Look Up Table (LUT) and Slices in FPGA Virtex-5 design. This demonstrates the ability of less power consumption.

Keywords: FPGA, Virtex-5, SR, MATLAB, Sys. Gen.

* Foundation of Technical Education - college of Electrical and Electronic techniques

1. Introduction

The software radio (SR) offers a flexible wireless communication resolution for an extensive range of claims, as well as cellular phones, GSM and military communication radios [1]. Since all hardware is physically automatic using software turn out to be fairly easy and make the Software Defined Radio (SDR) extremely cost effective [2]. In the surrounded applications of communication systems, the Field Programmable Gate Array is extensively used due to its reconfigurability. The very important preference of SDR claim is FPGA because it's not required hardware change out and increase the device life by repair data stream file. The level of investment an entire system is FPGA developed on a single chip, while permit in platform testing and correcting of the system. Additionally, it presents an opening of utilizing hardware-software co-design to extend a high performance system for unusual applications by excluding processors for definite software functions [3],[4],[5],[6]. The digital design of FPGA can be realized by using hardware description languages (HDL) which involve more information and skills in the tools such as VHDL and Verilog. The environment for model-based design in MATLAB/simulink is almost cover all industrial and scientific areas [7]. In this paper, the Xilinx digital signal processing (DSP) tool in system generator is investigated with graphical interface environments based MATLAB/ simulink and block set of DSP cores is used to model the DSP system. The model based SDR system is designed and implemented using FPGA. The conventional design flow go behind this work is clarified during the block diagram as shown in Figure 1 [8]. firstly, MATLAB/simulink block set has been used to build up a software model for the system followed by hardware physically in very high speed description language (VHDL) and validate its practical simulation using ModelSim. Subsequently, this VHDL model is ported into the MATLAB – Simulink model using System Generator block set talent. The functional of hardware is validate with the data generated from the model. The Xilinx integrated software environment (ISE) tools be used to implement the design on Xilinx FPGA vertix-5.



Figure(1) Separation between DSP and FPGA Design Flows [8].

In the late 1990s, Field Programmable Gate Array (FPGA) began to approach gate counts and speeds of Application Specific Integrated Circuit (ASIC), although ASIC remains superior for both parameters. However, when FPGA reached a semi-equivalent status, it became more attractive than ASIC for design and development work, due to its flexibility and reprogrammability [9]. With efficient massive parallel MAC (Multiply-Accumulate) structures, FPGA quickly became the method of choice for most of the Digital Signal Processing (DSP) algorithm implementations leading to high level of integration of various functionalities. However, the design flow for DSP algorithms and digital logic design via FPGA.s did not develop in unison. The DSP design flow is much shorter. One of the most world-renowned software tools for DSP algorithm design is Matlab, produced by Mathworks Inc [10]. Many companies researching and developing DSP, controls, and communication algorithms currently realize all of their high level modeling in Matlab, including all of the necessary test harnesses. In some cases, especially for real time applications, these designers would utilize a subprogram of Matlab, called Simulink. Simulink is a block diagram approach (connect boxes together) to system modeling that allows for modeling of streaming data, multi rate systems, and other numerically application intensive areas [11]. The issue that arose was that there was a gap between the Matlab/Simulink design flow, and FPGA

design flow, as is illustrated by Figure 1 [12]. Designers needed the parallel processing advantages offered by FPGA implementation. However, there was no software suite of tools that would allow a DSP algorithm designer to interface directly with an FPGA system architect. Furthermore, there was no streamlined process to verify equivalency of a hardware implemented DSP algorithm against the original test harnesses used in Matlab and Simulink. To solve this separation in design flow issue, a new electrical engineering work force skill began to be requested by companies around the world who specialized in both DSP algorithm development and FPGA design. This skill was the ability to execute brute force translation of Matlab and C++ code into Verilog or HDL.

2. Design Flow of FPGA Board

Based on available resource [13-18], the design method and investigational setup of the FPGA design flow is approved as shown in Figure 2. Software and hardware for SR model was designed at register transfer level (RTL) manually in VHDL. This is qualified by summarized the VHDL code in the Xilinx block set from Xilinx system generator library. The hardware block is coupled to the simulink data during Xilinx gateway blocks for input and outputs. The functionality of hardware is confirmed by using HDL code. This method permits to simulating hardware blocks with MATLAB simulink blocks design. The outcome of HDL simulation are changed from double precision values (floating point) to fixed point arithmetic values using output gateway blocks and passed through the block connected to the SDR model. The latency due to hardware is built-in in the model design by using delay block. The verification process between floating point and fixed point were done before connecting together to become integrated design.

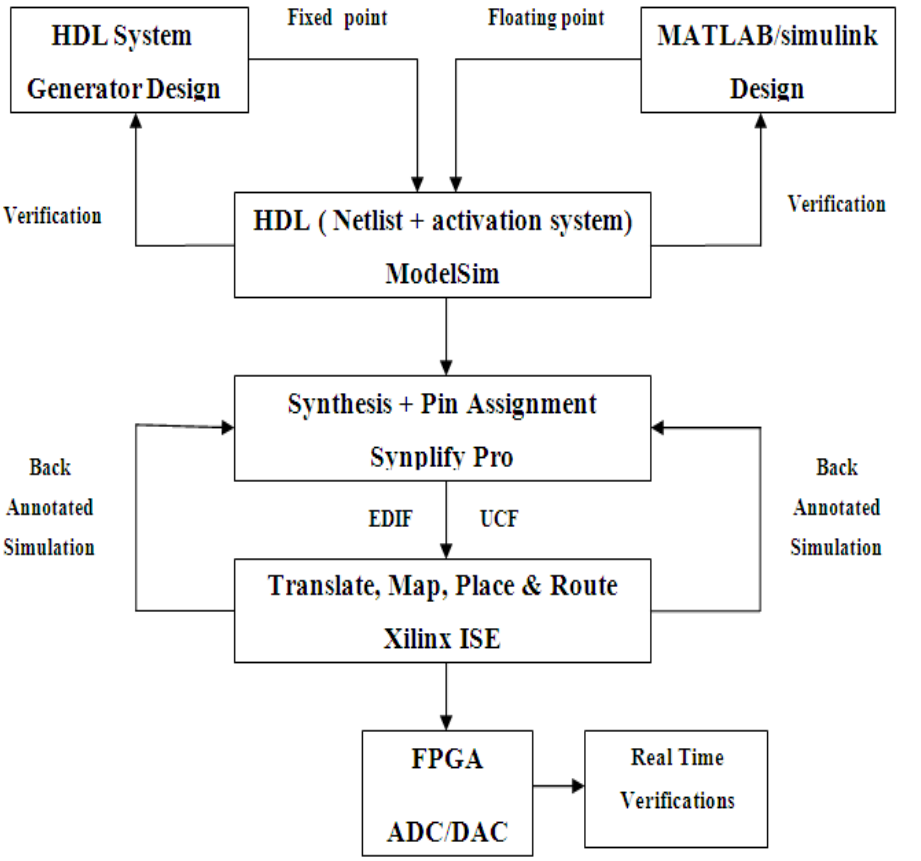


Figure (2) FPGA design flow

2.1 Floating / Fixed Point Conversion

The system generator and MATLAB programs from Xilinx and Mathworks respectively are used to convert the floating point values to fixed point values. The majority realistic FPGA designs are incomplete to limited precision signal processing using fixed-point arithmetic because of the cost and complexity of floating point hardware. The mapping DSP applications against FPGAs, a DSP algorithm designer, who frequently build up his applications in MATLAB, must find out the dynamic range and preferred precision of input, intermediate and output signals in a design

implementation to ensure that the algorithm reliability criterion are meet. The initial step in a flow to drawing MATLAB applications into hardware is the exchange of the floating point MATLAB algorithm into a fixed point description using System Generator. The floating point calculation in MATLAB can be transformed to a fixed point of exact accuracy for hardware design using system generator. The methods have been included in the Accel FPGA behavioral synthesis tool that reads in high-level descriptions of DSP applications written in MATLAB, and generates synthesizable RTL models in VHDL or Verilog by system generator and mapped onto the Xilinx Virtex-5 FPGAs. Two phenomenon could happen through the floating point to fixed point transition, overflow and quantization error. To simulate the simulink model via system generator design, the signal has to convert before they reach any Xilinx blocks devoted for hardware. Therefore, they first converted from double precision floating point values to fixed point values represented in binary format with decimal for hardware accepted as shown in Figure 3. In this case, the total number of bits is 12, and the binary point is 9, leaving 2 bits for integer plus the sign bit. Quantization shown in the right side of Figure 3, always occurs in the floating point to fixed point conversion process, since there must be some limit set to the number of decimal places that are kept for hardware representation. In this case, the whole number of bits is 12, and the binary point is 9, leaving 2 bits for the integer, plus the sign bit. Quantization, shown in the right side of Figure 3, always occurs in a floating point to fixed-point conversion, since there must be some limit set to the number of decimal seats that are reserved for hardware demonstration.

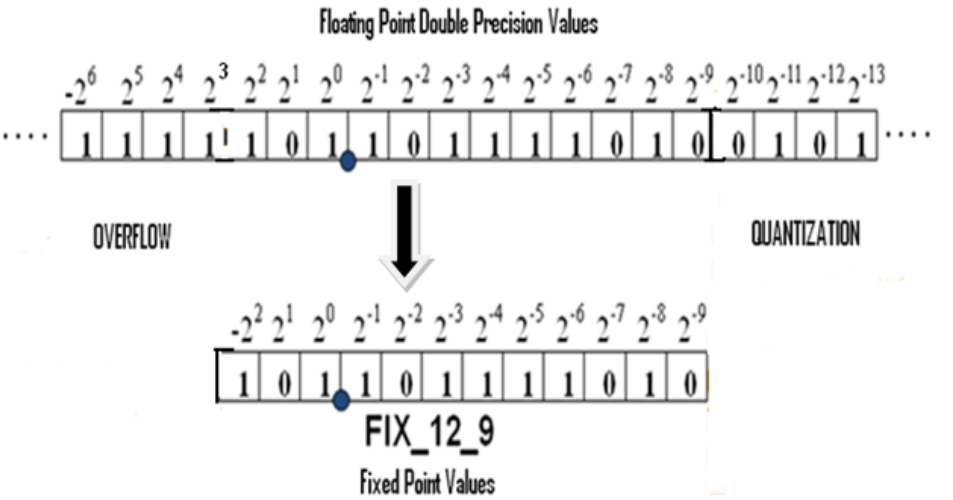


Figure (3) Floating point to Fixed-Point Conversion using System Generator

2.2 Design of HDL Code

The ModelSim software is used to design HDL module of activation system (plug-in Avnet Electronic Marketing P240 Analog Module) and Virtex-4 FPGA on-board ICS8442 Programmable Low Voltage Differential Signaling (LVDS) Clock Synthesizer with exact configurations. The HDL module of activation system should include all the specifications, characteristics, and features to ensure that ADC and DAC in P240 can function correctly. The HDL module of activation system and HDL netlist of SDR system in Verilog language are verified first before they are combined to become the HDL module of the integrated design. The simulated result of HDL netlist of the SDR system is similar to the simulation result of the System Generator design. It should be noted that these simulation results are similar to the System Generator design. Based on the real-time implementation of the integrated design using FPGA and P240 Analog Module, the ADC and DAC in P240 are activated and configured first before the FPGA design is run to avoid instability of ADC and DAC, which can produce undesired output to or from FPGA during the process of configuring ADC and DAC. Thus, the main clock

enable (*CE*) of FPGA design is disabled during the transfer of serial programming interface (SPI) codes to ADC and DAC in P240. Controlling the main *CE* rather than the clock would be easier. This enable clear (*CE_CLR*), which requires additional logics to adjust the sampling phase of all the multi-sample data when de-asserted. The simulation results of the HDL module of integrated design is given in Figures 4.

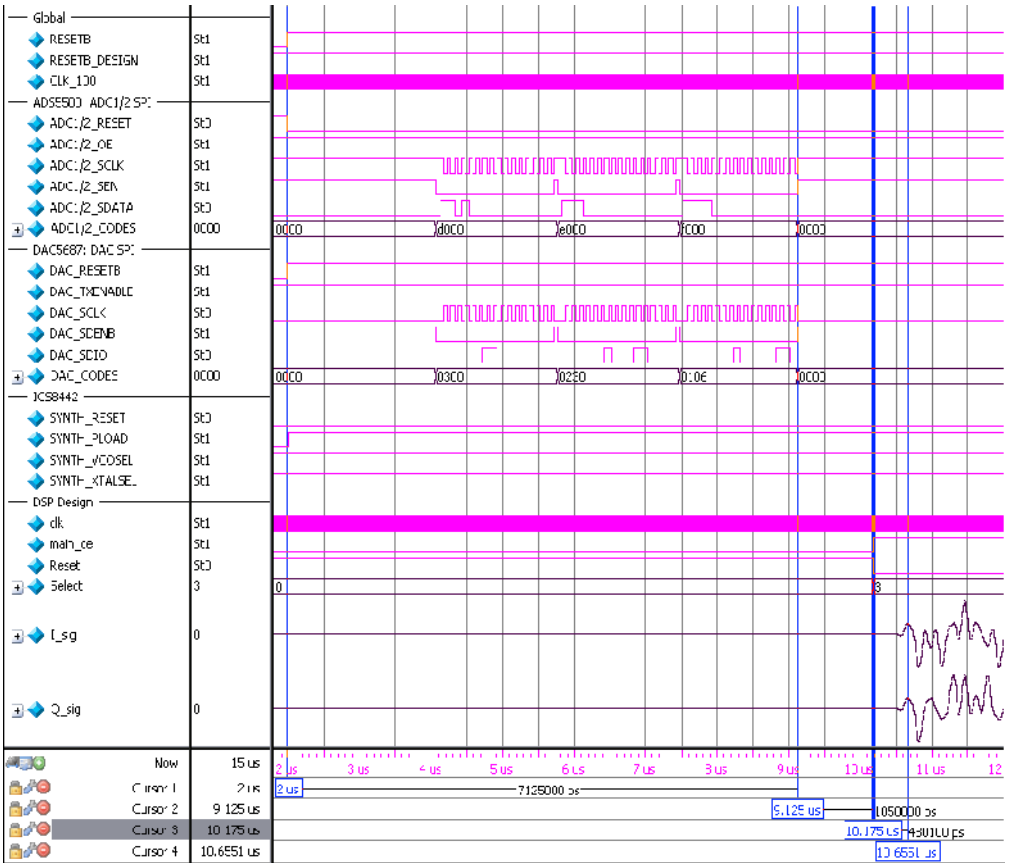


Figure (4) HDL Module of Integrated Design

2.3 HDL Synthesis

Although Xilinx ISE (Integrated Software Environment) has its own synthesis tool called XST (Xilinx Synthesis Technology), it can only synthesize HDL netlist generated from the System Generator. Therefore, the Synplify Pro software is used to perform logic synthesis for the HDL module of integrated design in two stages.

1. Logic compilation and optimization: Compile the Verilog HDL module of integrated design to Xilinx FPGA structural elements and then optimize the integrated design to make it as small as possible to improve circuit performance.
2. Technology mapping: Map the optimized integrated design to Xilinx FPGA logic components using architectural-specific technique.

Timing characteristics comprise another important factor that affects the performance of FPGA implementation. Thus, the estimated period (required path delay) for the FPGA element must not exceed the requested clock period. For this reason, the timing slack (requested period - estimated period) should have a positive value; otherwise, the integrated design has to be reworked. The clock frequencies used for ADC/DAC are set to 100 MHz for CLK_100 (ADC/DAC SPI process) and 80 MHz for LIO_CLKIN_1 (16-QAM SDR transmitter and receiver). The estimated timing report for the synthesized design meets the time constraints, as shown in Table 1.

Table (1) Estimated Timing Report

Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors
TS_CLK_100 = PERIOD TIMEGRP "CLK_100" 10 ns HIGH 50%	SETUP	2.412ns	3.588ns	0
	HOLD	0.542ns		0
TS_LIO_CLKIN_1 = PERIOD TIMEGRP "LIO_CLKIN_1" 12.5 ns HIGH 50%	SETUP	3.607ns	4.893ns	0
	HOLD	0.257ns		0

2.4 Virtex-5 FPGA Implementation

The final step in FPGA design is the implementation of design model on Xilinx Virtex-5 using ISE software tools. The synthesis output files generated by the ISE software in electronic design interface file (EDIF) and user constraints file (UCF) forms represent the optimized netlist of integrated design, timing constraints, and FPGA pin assignment that have been implemented into the FPGA development board by following the steps listed below:

- Translate: Convert the net list file of integrated design in EDIF format to native generic database (NGD) file, which contains logic description of hierarchical components and Xilinx primitives for the integrated design using NGD build program.

- Map: Perform logical DRC on the NGD file and map the design logic to slices and input-output (I/O) cells in FPGA to create native circuit description (NCD) file.
- Place and Route: The design in mapped NCD file is place and route into FPGA based on timing constraints using timing analysis tools with no errors found.

- Bit generation and program download: Bit generation is used to generate configuration bit-stream file in BIT format form, and subsequently downloaded into FPGA via JTAG cable using the iMPACT program.
- The timing requirement is satisfied as shown in the post-PAR (final) static timing report in Figure 5 and Table 2.

Table (2) Post-PAR Static Timing Report

Starting Clock	Requested Frequency	Estimated Frequency	Requested Period	Estimated Period	Slack
CLK_100	100.0 MHz	189.4 MHz	9.000	4.280	4.720
LIO_CLKIN_1	80.0 MHz	218.7 MHz	11.500	3.573	7.927
System	100.0 MHz	511.8 MHz	9.000	0.954	8.046

```
================================================================================
Timing constraint: TS_LIO_CLKIN_1 = PERIOD TIMEGRP "LIO_CLKIN_1" 12.5 ns HIGH
50%;

 222274 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)
 Minimum period is  11.035ns.
--------------------------------------------------------------------------------


All constraints were met.


Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)

Clock to Setup on destination clock CLK_100
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
CLK_100        |    4.979|         |         |         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock LIO_CLKIN_1
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
LIO_CLKIN_1    |   11.035|         |         |         |
---------------+---------+---------+---------+---------+
```

Figure (5) post-PAR (final) static timing report

The implementation steps shown in Figure 6 are read in the constraints file that consists of three major steps: translate, map, and place and route. The translate step essentially flattens the output of the synthesis tool into a large single netlist. A netlist in general, is a large list of gates which is compressed at this stage to remove any hierarchy. The map step groups the logical symbols in the flattened netlist into physical components, specific to the target device. The place and route step places each of these physical components onto the FPGA chip and connects them through the switch matrix and dedicated routing lines.

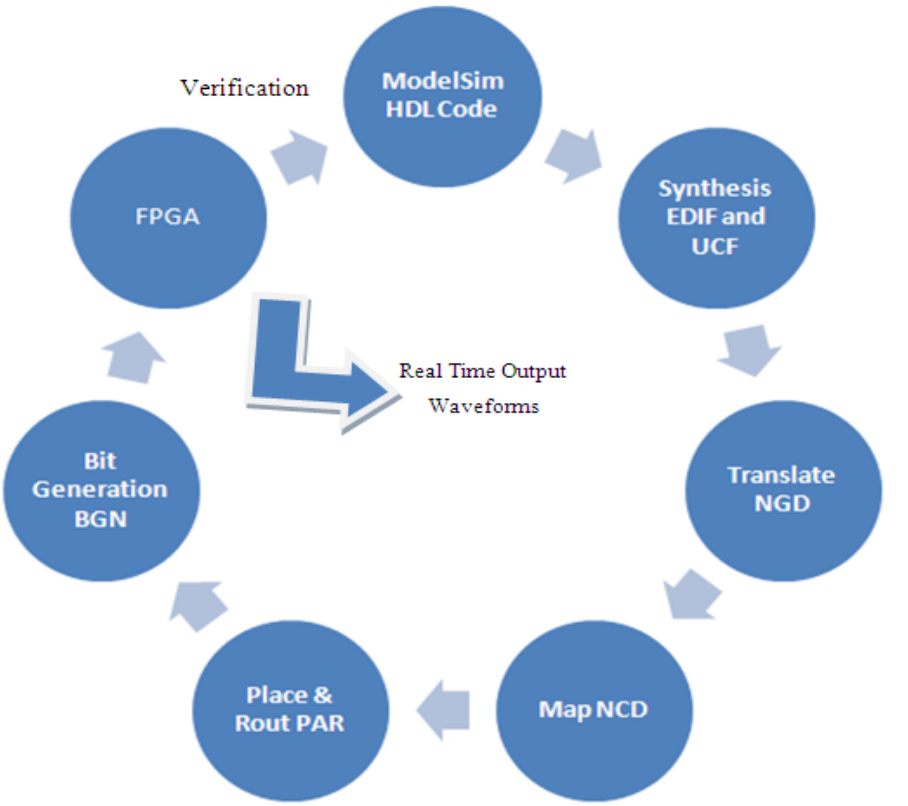


Figure (6) FPGA Implementation steps

3. Results

Figure 7 shows the input and output signals of the proposed SDR model. The difference between the original and the recovered message bits is zero, and the difference between the original and the recovered symbol integers is also zero. This indicates that the timing synchronization between the transmitter and the receiver is optimized and the symbol remapping removes the noise caused by AWGN using the tolerance range of threshold in decision-making technique. In this technique, the threshold level is limited by three values (0.667, 0.333, and 0.125) that represent the tolerance range in I and Q channels.

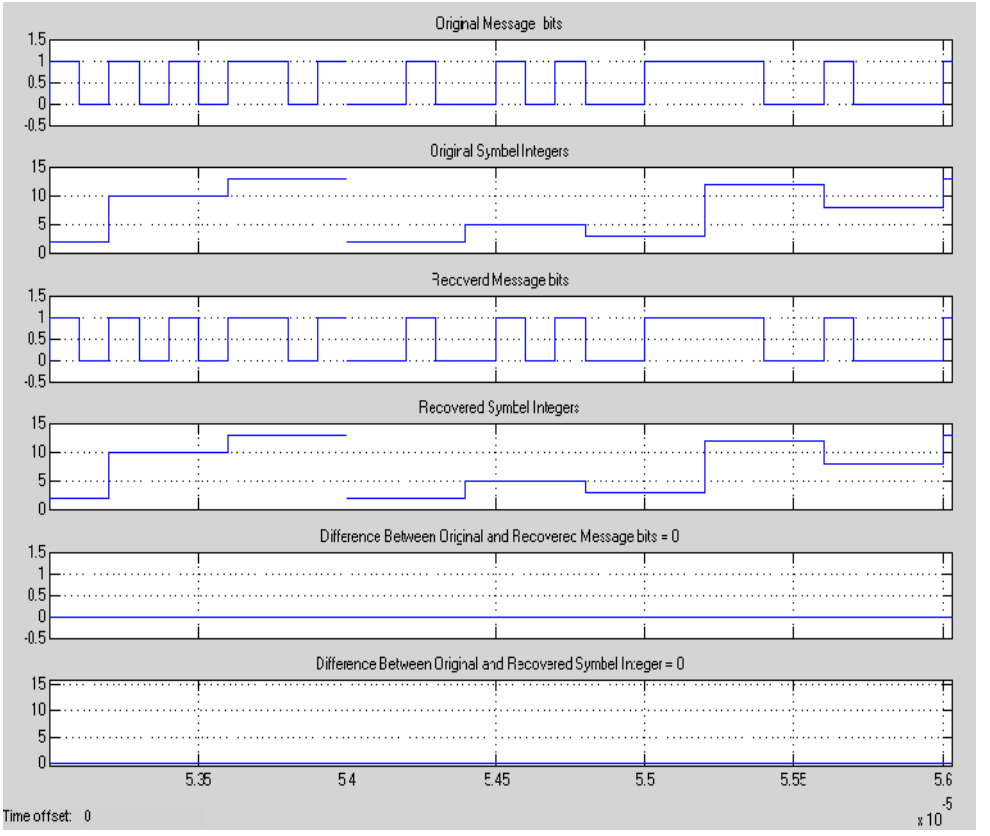


Figure (7) Input and output signal of proposed software defined radio

Figure 8 shows the information about the completion of placing and routing of the system in the hardware. The figure demonstrates the successful completion of placement routing and timing. All constraints are addressed for the FPGA transmitted board, and the FPGA receiver board and all signals are completely routed. Therefore, the placing, routing, and timing are completed with no errors found.

```
All constraints were met.

Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 34 secs
Total CPU time to PAR completion: 31 secs

Peak Memory Usage:  269 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 2
Number of info messages: 0

Writing design to file P240_1D_QAM_Tx_3.ncd


PAR done!
```

```
All constraints were met.

Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 6 mins 37 secs
Total CPU time to PAR completion: 6 mins 34 secs

Peak Memory Usage:  416 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 2
Number of info messages: 1

Writing design to file P240_1D_QAM_Rx_1.ncd


PAR done!
```

Figure (8) Command window showing the progress of implementation tools

The transmitter and receiver project status and device utilization summary are reported by the ISE program, as shown in Table 3. This table provide the total number of slices and LUTs used in this design, which represent the total area used in FPGA. Depending on the number of devices used in FPGA, the total power consumption in the SDR model implementation can be seen, according to the number of slices and LUTs. The total LUP for proposed design is 6.908 with 5.187 Slices as illustrated in utilization

summary. The real-time QAM baseband modulated signals are the analog output signals give in from the FPGA implementation via DAC is shown in Figure 9. The real-time I or IQ baseband modulated signals are verified by evaluated the measured and simulated values. The percentage of used logic elements to the available logic elements is calculated as follow:

Utilization % = $\frac{\text{used logic elements}}{\text{available logic elements}} X\ 100$ **, for example,**

Utilized number of LUTs = (6,908/30,720) x 100 = 22%

Table (3) FPGA utilization summary

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	10,345	30,720	30%	
Number of 4 input LUTs	6,908	30,720	20%	
Logic Distribution				
Number of occupied Slices	5,187	15,360	35%	
Number of Slices containing only related logic	5,187	6,187	90%	
Number of Slices containing unrelated logic	0	6,187	0%	
Total Number of 4 input LUTs	8,291	30,720	26%	
Number used as logic	6,908			
Number used as a route-thru	180			
Number used for Dual Port RAMs	30			
Number used as Shift registers	160			
Number of bonded IOBs	85	448	18%	
Number of BUFG/BUFGCTRLs	3	32	10%	
Number used as BUFGs	3			
Number used as BUFGCTRLs	0			
Number of DSP48s	3	192	1%	
Number of RPM macros	1			
Total equivalent gate count for design	181,238			
Additional JTAG gate count for IOBs	4,272			

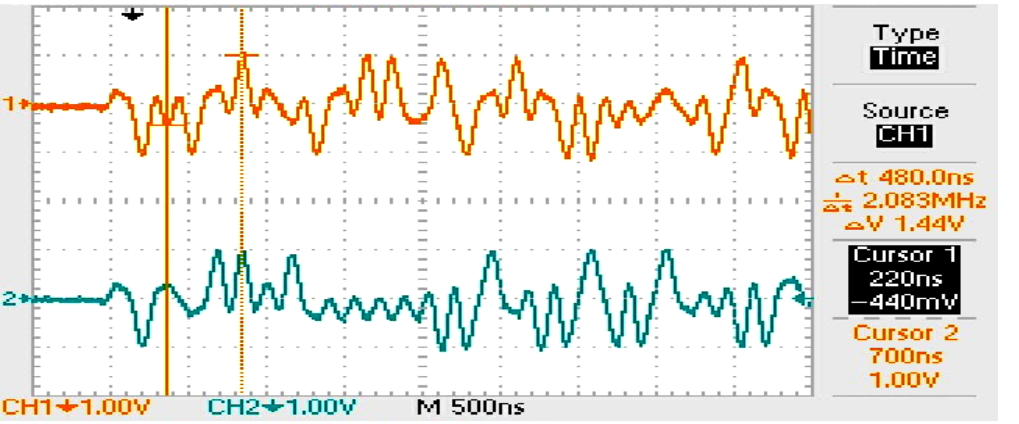


Figure (9) Real-time Results of QAM modulator

4. Conclusions

In this paper, a capable trail to design of Virtex-5 FPGAs has been developed and improved to work with 4G system and software defined radio. A new technique has been used to reduce the LUT and Slices in FPGA area therefore, decrease the power consumption and size in future implementation of FPGAs. The proposed technique paths is incorporate with MATLAB/ simulink and system generator. The Synplify pro synthesis and ISE software is also used to synthesis the HDL code and generate the bit stream to download it to FPGA board. Results obtained from the implementation of this design, has shown that the proposed algorithm reduces the FPGA logic utilization by nearly 20 % in terms of FPGA elements of Slices and LUTs, compared to conventional designs. In addition, a design and implementation of software defined radio model was done to verify the design performance under channel noise.

5. References

[1] Changrui, Kong, Xie Shige and Huizhi1, “Design and FPGA Implementation of Flexible and Efficiency Digital Down Converter”, ICSP Proceeding 978-1-4244-5900-1/10 IEEE, PP.438-441, 2010

[2] Durke, “Field Programmable Gate Array Based SDR Design” phD. Thesis, Naval postgraduate school, Department of Electrical and Computer Engineering,California, 2009

[3] V. B. Alluri, J. R. Heath, and M. Lhamon, “ A new multichannnel , coherent amplitude modulated, gate array technology implementation,” *IEEE Transaction on Signal Processing*, vol. 58, pp. 5369-5384, October 2010.

[4] Dick, and Hwang, “FPGAs: A Platform-Based Approach to Software Radios in Software Defined Radio, England: John Wiley & Sons Ltd, ISBN: 978-0-470-86770-1, Ch. 10, pp. 235-272, 2004

[5] J. Mitola, “ Software radios: survey, critical evaluation and future directions,” *IEEE Aerospace and Electronics Systems Magazine*, vol. 8, pp. 25-36, April 1993.

[6] Jeffry, “Software Radio a modern approach to radio engineering”, Prentice Hall PTR, A division of person education Inc, ISBN 0-13-081158-0, pp.33-123, 2002

[7] M. Palkovic, P. Raghavan, M. Li, A. Dejonghe, L. Van der Perre, and F. Catthoor, “ Future Software-Defined Radio Platforms and Mapping Flows," *IEEE Signal Processing Magazine,* vol.27, no.2, pp.22-33, March 2010.

[8] J. Kim, H. Seungheon, and S. Choi; , "Implementation of an SDR system using graphics processing unit," *IEEE Communications Magazine* , vol.48, no.3, pp.156-162, March 2010.

[9] Y.Lu, E. Oruklu, and J. Saniie, “ Fast chirplet transform with FPGAbased implementation,” *IEEE Signal Processing Letters*, vol. 15, pp. 577-580, December 2008.

[10] A. Pal, "FPGA-based (Xilinx) Embedded System Design," *Workshop on Microcontroller and FPGA-based Embedded System Design*, Department of Computer Science & Engineering, IIT Kharagpur, India, July 2007.

[11] Xilinx ISE 9.2i Software Manuals: Constraints Guide, and Development System Reference Guide, Xilinx, Inc., 2007.

[12] Xilinx, " System Generator 12.1 User's Manual", Xilinx. Inc., 2010.

[13] The MathWorks, Inc. DSP – Digital Signal Processing& Communications. [Online]. July 10, 2009,http://www.mathworks.com/applications/dsp_comm

[14] Xilinx, Inc., *DSP Design Flows in FPGA Tutorial Slides*, 2003..

[15] D. Benson, " The Design and Implementation of a GPS Receiver Channel, An FPGA plus DSP Creats a Powerful Combination", *DSP Magazine*, October 2005.

[16] Xilinx Corp., Virtex-4 Xtreme DSP Design Considerations User Guide,2008

[17] Xilinx Corp., System Generator User Guide,2008

[18] Xilinx Corp., Xilinx ISE 10.1 Software Manuals, 2008.

تَصميم فعال لمصفوفة البوابات المبرجة (إف بي جي أي) يعتمد التعديل الرباعي باستخدام مولد النظام وبرنامج ماتلاب

د.ماجد صلال نغمش*- د.خليل جديع حمادي*- د.علي مهدي حمادي*

المستخلص

هذه الدراسة تقدم ،تَصميم وتنفيذ مصفوفة البوابات المبرمجة بالاعتماد على التعديل الرباعي باستخدام مولد النظام وبرنامج ماتلاب. للسَماح لكُلّ أنظمة الإتصال الرقميةِ لتبني تقنيات تحميل وتشفير اكثر تعقيدا ِ، يَجِبُ أنْ يُستَعمل نمط إس دي آر بشكل ملحوظ لتامين الطلبات المتزايدةِ مِنْ إتصال اللاسلكي والصناعة النقَّالة. لذلك، تعتمد هذه الدراسةِ، على الطرق الأقصر والكفوءة لتَصميم مصفوفة البوابات المبرمجة من خلال استخدام برنامج ، مولّد نظام ، موديلسم Synplify , ماتلاب وآي إس إي (بيئاتَ البرامج المتكاملة) و أدوات برامج مُقَدَّمةُ. تصميم تمثيل قيم الاعداد في برنامج ماتلاب يحول إلى قِيَم الثابتةِ باستخدام البرنامج الأكثر شيوعا وحداثة المتمثل بقاعدة المعالج الرقمي للاشارة . ان حاملات الاشارة ورموزها يمكن اكتشافها بالاعتماد على دائرة قفل الطور . النتائج التي تم الحصول عليها تؤكد على امكانية تحسين وتقليل عدد الجداول والشرائح في تصميم مصفوفة البوابات المبرمجة مما يعني استهلاك اقل للقدرة.

* هيئة التعليم التقني - كلية تقنيات الهندسة الكهربائية والالكترونية