ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research

Vol. 12, Issue 09, pp.11667-11674, September, 2025

https://doi.org/10.64485/ijramr.6108.09.2025

RESEARCH ARTICLE **CROSSREF** **OPEN ACCESS**

DESIGN AND IMPLEMENTATION OF 256-QAM TRANSCEIVER ON FPGA USING MATLAB/SIMULINK MODEL

***Rehan Muzammil**

Senior Member, IEEE, Electronics Engineering Department, ZHCET, Aligarh Muslim University, Aligarh, India

ARTICLE INFO

Article History

Received 24th June, 2025
Received in revised form
28th July, 2025
Accepted 24th August, 2025
Published online 30th September, 2025

Keywords:

256-QAM, FEC, FPGA, ARM, Model-Based Design.

****Corresponding author:***
Rehan Muzammil

ABSTRACT

This paper discusses the Software Defined Radio-based Digital Transceiver, focusing on the 256-QAM transceiver used exclusively in 5G. The transceiver discussed here is software-defined, quite unlike that used in 5G.The 5G transceiver switches to the 256-QAM Modulation scheme when the received power is high. This paper uses the SIMULINK model to discuss the 256-QAM Modulation scheme, where the whole transceiver is designed and implemented on an FPGA. The speed and reconfigurability of the FPGAs are the driving forces for such an implementation. This Model's VHDL and C codes are generated automatically using the HDL workflow advisor. This circuit was designed and implemented on ZEDBOARD, which has an SoC comprising the Zync-7000 FPGA and ARM processor. The whole Model runs such that a part runs on the FPGA, and the rest runs on the ARM processor.

Citation: Rehan Muzammil. 2025. "Design and Implementation of 256-QAM Transceiver on FPGA using MATLAB/SIMULINK Model", International Journal of Recent Advances in Multidisciplinary Research, 11, (09), 11667-11674.

INTRODUCTION

The need for very high-speed data transmission has resulted in the rapid growth of digital communication hardware.The mobile communication industry faces the problem of providing this technology which can support many services ranging from a low bit rate communication, such as voice communications of about 2 Mbps, to multimedia communications with a much higher bit rate of over 2 Mbps.ASIC design and implementation are not very flexible and require much time to market. Hence, Field-Programmable Gate Array (FPGA) is preferred over the ASIC.The FPGA hardware is programmable and does not involve a delay in the case of IC fabrication. (1, 7). The capacity of the multimedia communications network is rapidly growing with the rapid growth in mobile data services. To reach such a goal higher modulation scheme and FEC codes become necessary. This results in the design of the FPGA architecture and implementation of adaptive QAM modulation, which can support a high modulation scheme such as 256-QAM. (2). In wired and wireless communication circuits, FPGAs can provide excellent solutions, such as reprogramming the baseband circuit. Nowadays, FPGAs can offer accuracy in simulating, validating, testing, and implementing designs(3, 10).Still, a Major challenge faced in circuit design is implementing the 256-QAM (4). A wide variety of applications require high-speed and low-cost wireless communication circuits for data transmission, which is catered by rapid growth in technology. Higher Modulation schemes are used to increase the data rate, such as the 256-QAM modulation, which results from the bandwidth-limited channel because it can accommodate more data into one symbol.The Forward Error Correction schemes are necessary because such systems are used in wireless channels that observe random and burst errors. FPGA is a more effective transceiver design as the transmitter and receiver can be integrated into a single FPGA (5, 6). The current analog capacity channels are much slower than the High-speed wireless data communications in general and Multi-channel Multi-point Distribution Services (MMDS) in particular (8).New transmission channel methods are extensively used for providing more bandwidth as higher-order modulation types are limited (9). The low power consumption has enhanced the necessity for the evolution of communication technology from 3G to 5G. The 5G technology is now viable, which is necessary for the rapidly growing traffic demand and newer services (11). In the application of digital signal processing and digital communications, FPGAs have been exclusively adopted. They are also well suited for the technologies like software-defined radios, which are evolving fast. This is due to the programmability and reconfigurability of the FPGAs. The baseband functionalities of a transceiver are primarily implemented using FPGAs. The communication transceiver can use reconfigurable hardware such as FPGAs to perform digital signal processing tasks (12). Section II describes the System architecture. Section III describes the Forward Error Correction (FEC) used in this system. Section IV illustrates the QAM-256 Modulation. Section V discusses the QAM-256 Demodulation. Section VI describes the hardware implementation of the design onto the ZEDBOARD and the Real-time results Section VII draws on the conclusion.

System Architecture: The 256-QAM transceiver is implemented using the Model in MATLAB SIMULINK. The top Model of the system is shown in Figure 1.

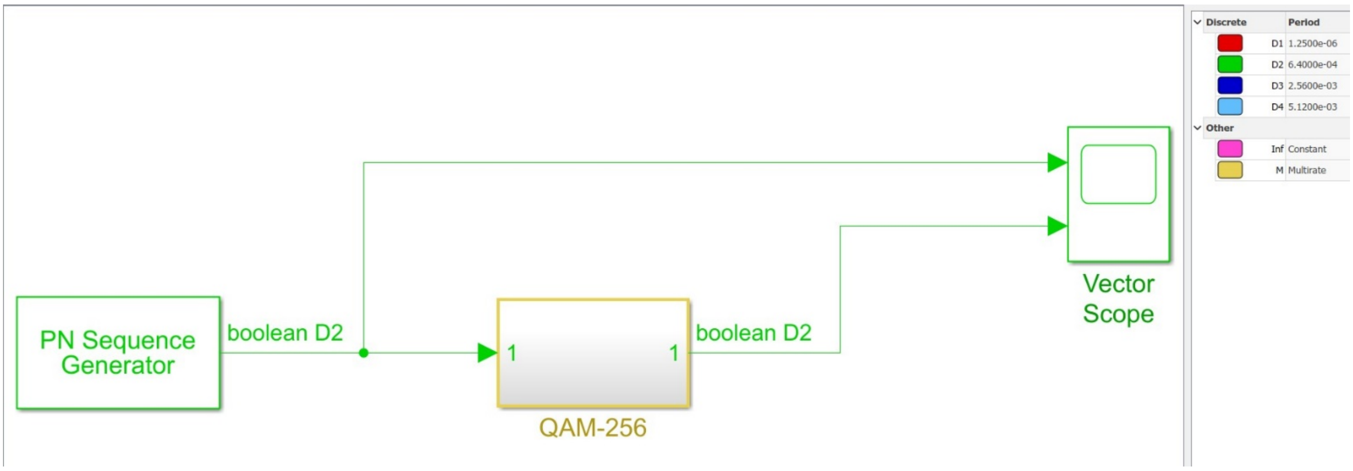


Fig. 1. Top Model of the System

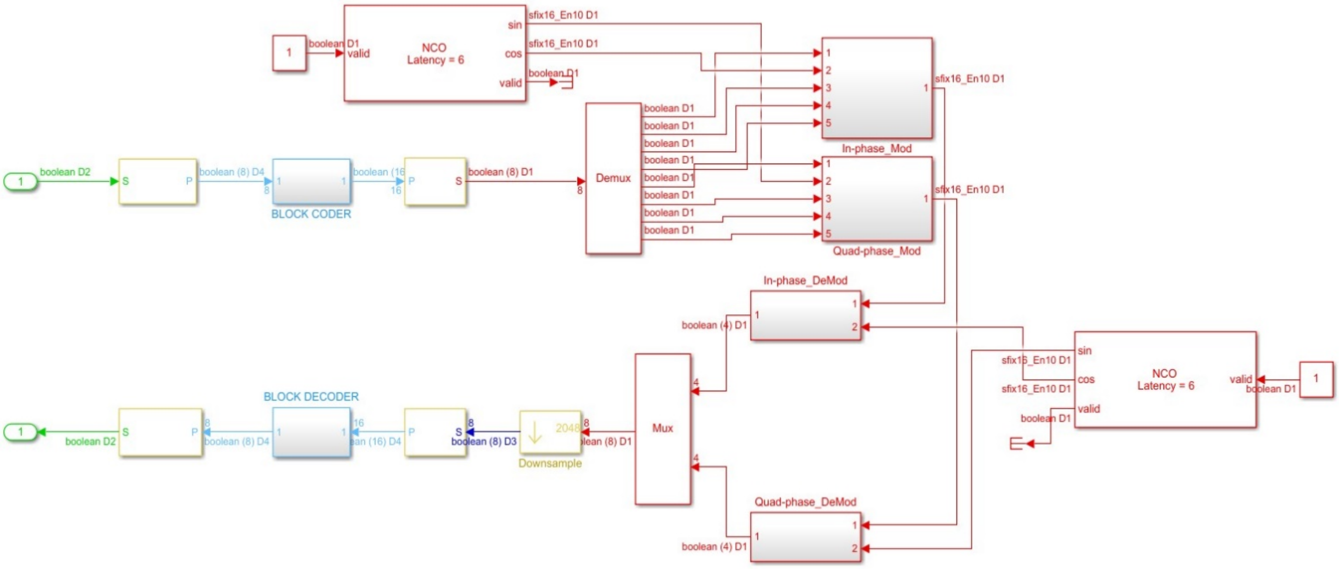


Fig. 2. QAM-256 Model of the Transceiver

In this Model, the QAM-256 subsystem contains the baseband model of the complete transceiver, shown in Figure 2. The main subsystems in the 256-QAM Model are Modulation/demodulation, Coding / Decoding, etc. The QAM-256 subsystem block runs on the FPGA, and the rest runs on the ARM processor onto the ZEDBOARD. The colors depict the sampling speeds at various connection points in the Model, with red color having the highest sampling speed and sky blue having the lowest sampling speed. The FEC used in this design is a more efficient scheme of the (7,4) Block Codes which is the (8, 4) Block Codes.

From the input bit stream to the Modulator, the circuit comprises the transmitter, and after this, the serial bit output from this subsystem is the receiver. The single serial bit stream input to the 256-QAM subsystem is fed to the serial-to-parallel converter, which outputs a parallel bit stream of 8-bit wide. This 8-bit stream is divided into four-bit streams and provided to the two (8,4) Block coders running in parallel. The output from the two Block coders is concatenated to give a bit stream of 16-bit width. This 16-bit parallel stream is passed through a parallel-to-serial converter which converts it into 8-bit words (because the QAM-256 has eight bits per symbol. This 8-bit word is again broken into two 4-bit words and passed on to the in-phase and quadrature-phase modulators, shown clearly in Figure 2.

Forward Error correction: The forward error correction is implemented using two (8, 4) block coders and decoders. The circuit of the block coder is given in Figures 3 and 4. The block decoder is shown in Figures 5 and 6.

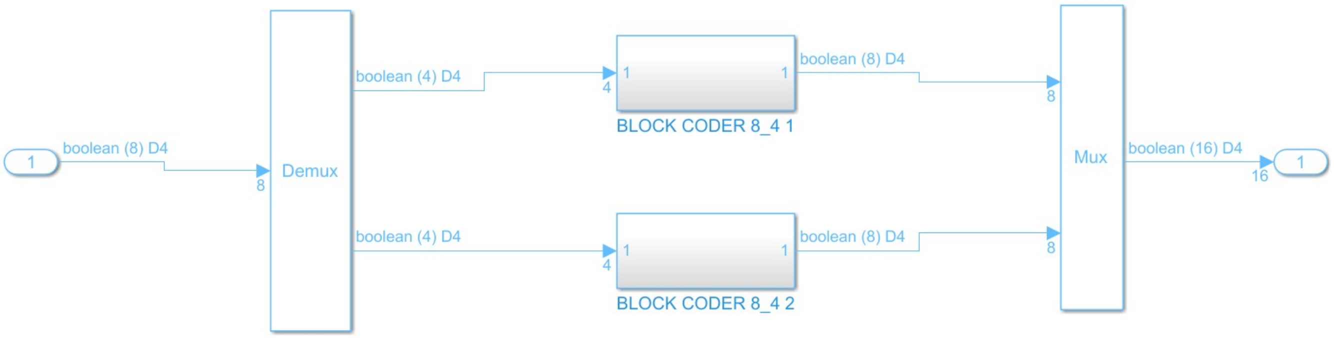


Fig. 3. The Block Coder Subsystem (two coders in parallel)

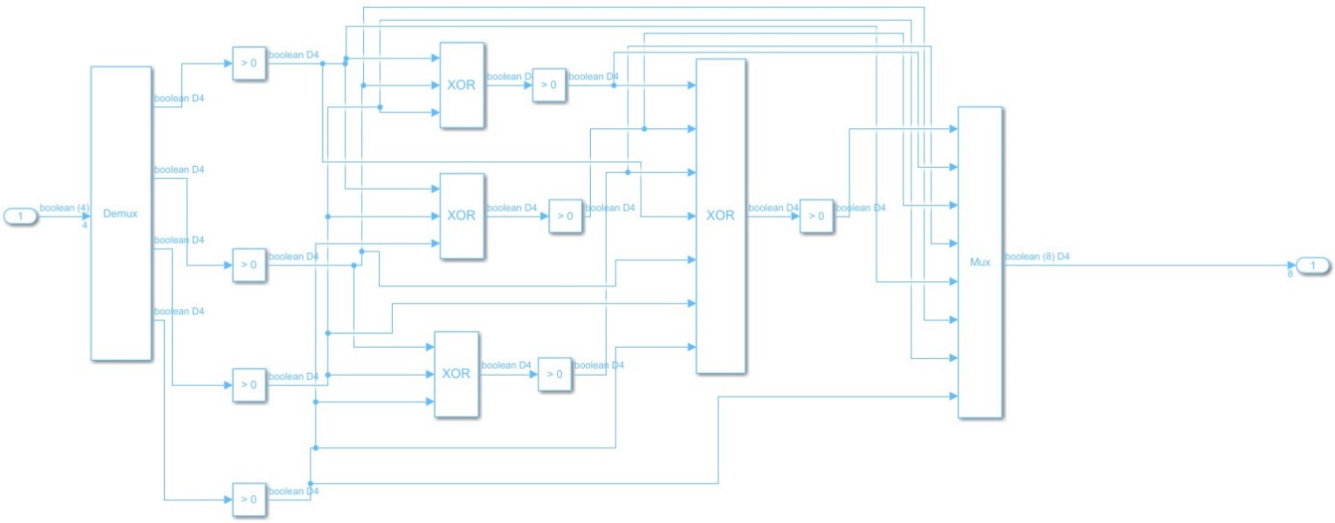


Fig. 4. The (8, 4) Block Coder architecture

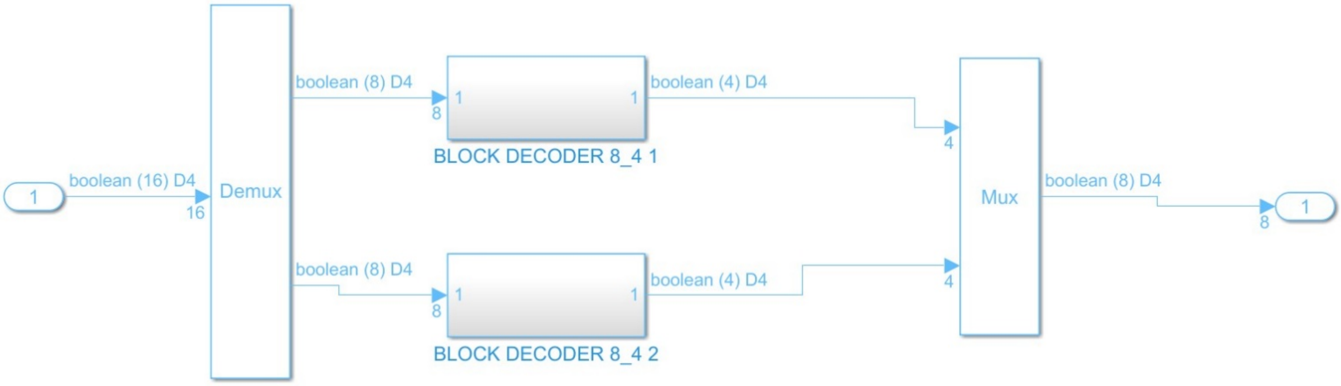


Fig. 5. The Block Decoder Subsystem (two decoders in parallel)

The algorithm for the (8, 4) block coder and decoder is well documented in the books and literature.The coder's output is 8-bit, which comprises the 4-bit data and 4-bit parity. These 8-bit words from the two coders are fed to the multiplexer, which joins this into a single 16-bit wordprovided to the parallel-to-serial converter. This P/S converter gives an output of 8-bit wide which is split into two 4-bit wide, fed to the In-phase and quadrature-phase modulators of the 256-QAM transmitter. On the receiver side, as seen in Figure 5, the 16-bit word is fed to the Demultiplexer, which breaks it into two 8-bit words. The 8-bit words are further processed by the two decoders running in parallel and give an output of 4-bit each, which are then concatenated to make it an 8-bit word which is the data. This is then passed through a P/S converter giving a single bit-stream output. This bit-stream is then fed to the vector scope, where it is compared with the input bit-stream to ascertain the errors in the transmission, which can be observed in Figure 1. The comparison results are shown in the Real-time result section, later passed onto the vector scope as shown in Figure 1.

QAM-256 Modulation: The 256-QAM Modulation Blocks (in-phase and quadrature-phase) are shown in Figure 7. Both the blocks are identical,so only one subsystem is described here.

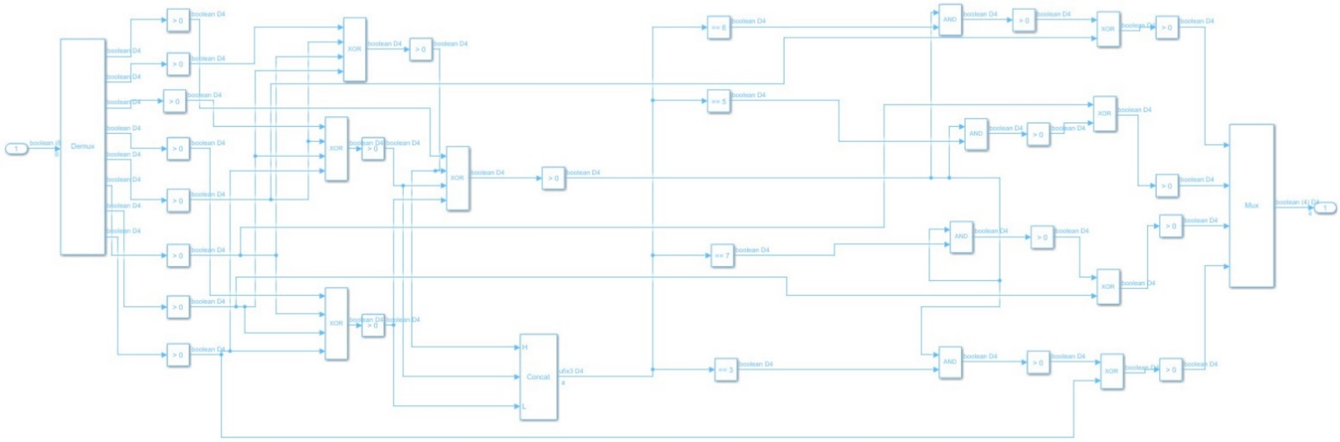


Fig. 6. The (8, 4) Block Decoder architecture

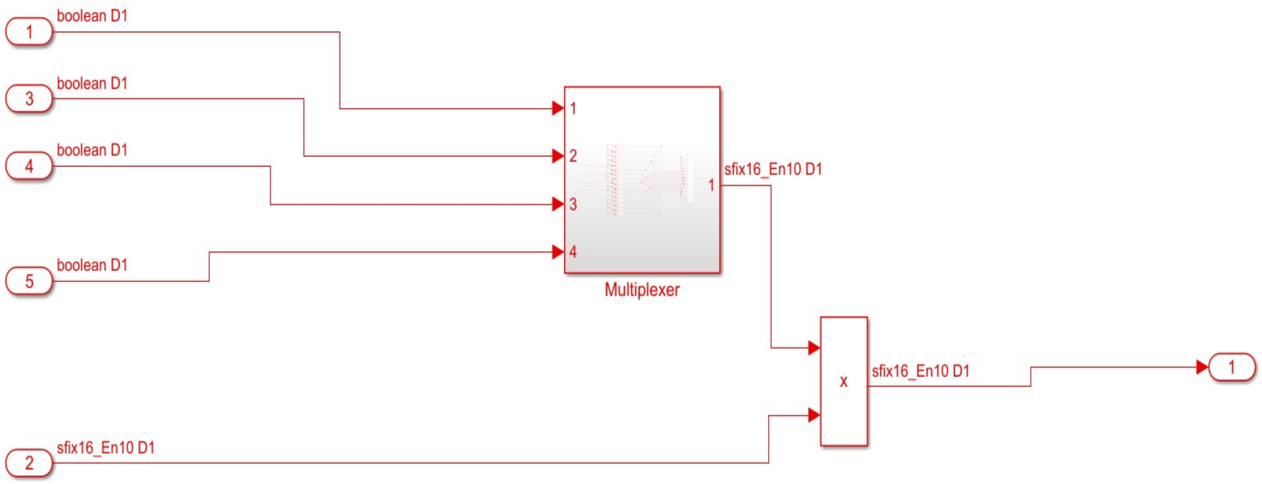


Fig. 7. QAM-256 Modulator

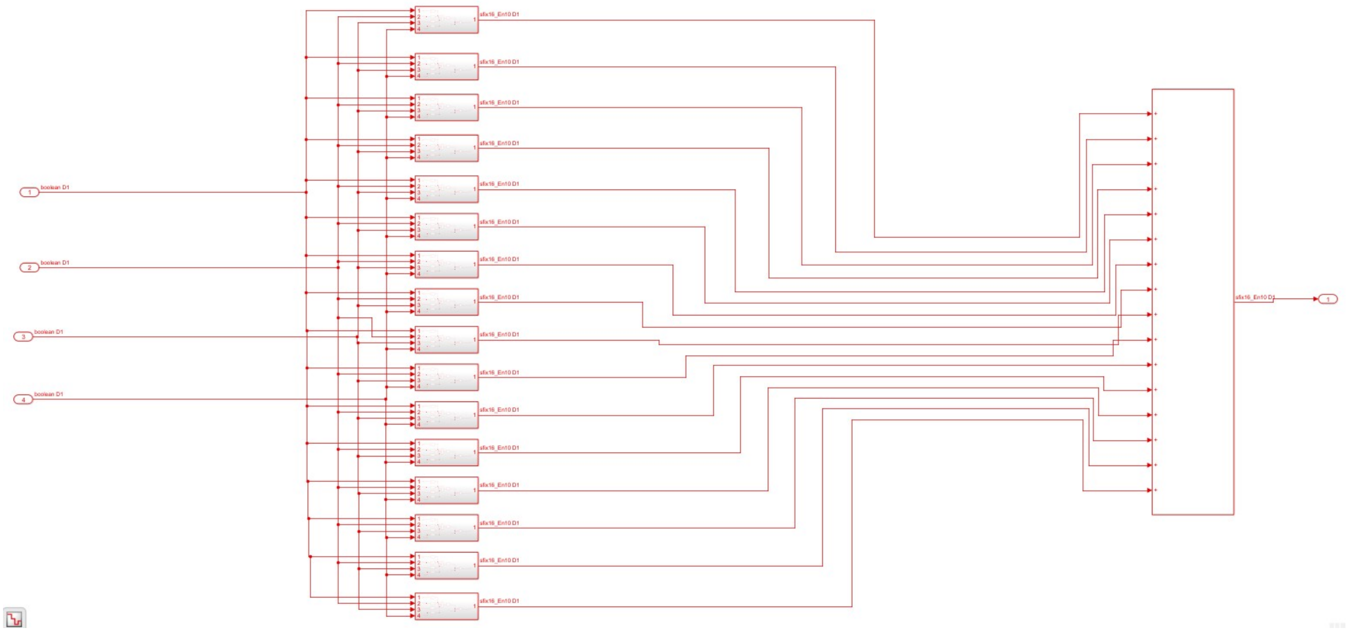


Fig. 8. The QAM_256 Multiplexer

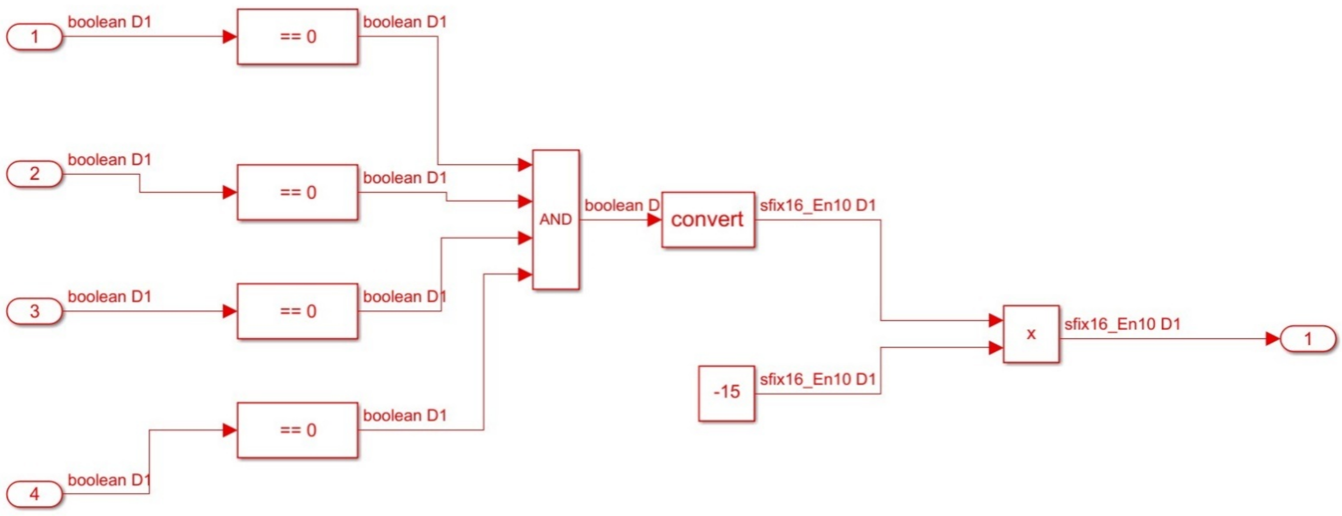


Fig. 9. One Subsystem in the QAM-256 Multiplexer

Table 1. Multiplexing Table (Real and Imaginary parts of symbols)

4-bit word (Gray Coded)	Multiplexed Symbol Value
0000	-15
0001	-13
0011	-11
0010	-9
0110	-7
0111	-5
0101	-3
0100	-1
1100	1
1101	3
1111	5
1110	7
1010	9
1011	11
1001	13
1000	15

This symbol value then modulates the real and imaginary parts of the incoming symbol of the eight bits (4-bit each for the real and imaginary parts), as shown in Figure 9. From Table 1 it can be seen that the eight bits are split up into two 4-bit chunks and gray-coded for the real and imaginary parts. The range of symbol values taken by each symbol for these real and imaginary parts are ranging from +15 to -15.

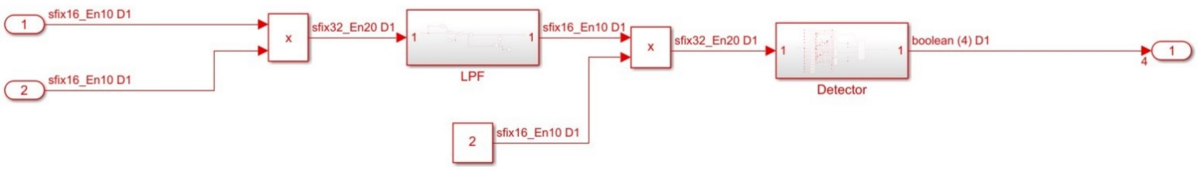


Fig. 10. The QAM-256 Demodulation Subsystem for the in-phase and quadrature-phase components

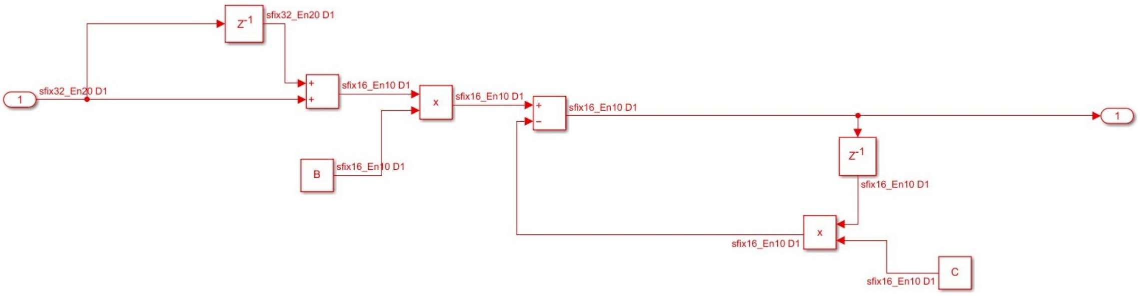


Fig. 11. The LPF subsystem

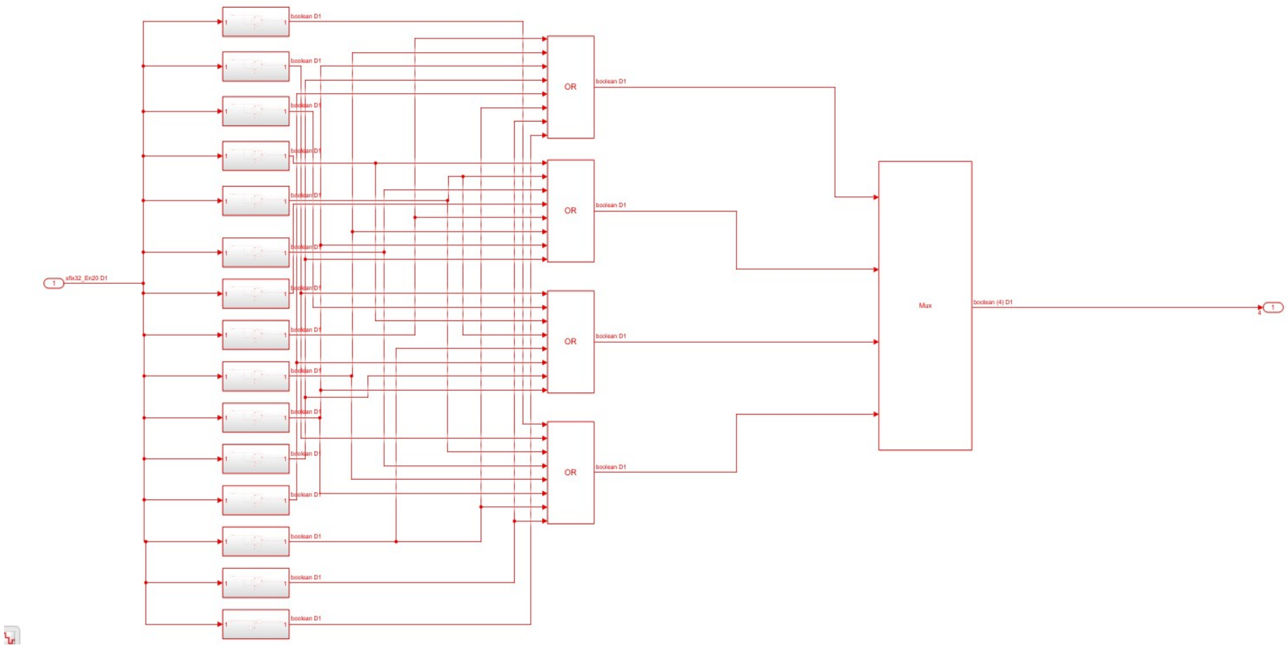


Fig. 12. The Detector Subsystem for in-phase and quadrature-phase demodulators

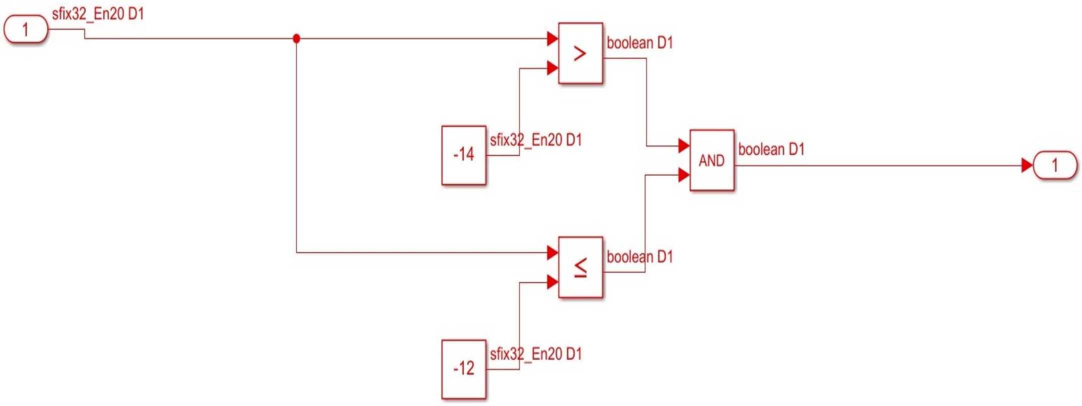


Fig. 13. One Subsystem in the Detector Subsystem

QAM-256 Demodulation: Figure 10 illustrates the 256-QAM demodulation for the real and imaginary parts of the 256-QAM symbols. The coherent demodulation is employed here, and the carrier generated in the transmitter is phase synchronized with the carrier generated in the receiver. This happens automatically as the numerically controlled oscillators generate the two carriers on the same FPGA. The output of the multiplier is fed to the LPF, as shown in Figure 10. This LPF then cuts off the high-frequency part of the received signal and passes only the DC part of the signal, which we are interested in. This DC part contains the information about the symbol transmitted. This DC part is then decoded, as shown in Table 1, to give the transmitted symbols for the real and imaginary parts. This LPF has been designed and implemented by converting the analog LPF into digital LPF using the Bilinear Transformation technique. The filter thus realized is an IIR Digital Filter, as seen in Figure 11.

The values of B and C constants shown in Figure 11 are calculated and stored in the Model properties. They can be easily varied in the Model properties section depending upon the cut-off frequency of the LPF. Moreover, the detector part is shown by the subsystem in Figure 12. In the detector, one subsystem can be seen in Figure 13. As seen from this figure, the value -13 falls between -12 and -14, which are the thresholds. In the same way, -11 falls between -12 and -10, the two thresholds in the detector. There are 15 thresholds in all for the system. They are

-14, -12, -10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10, 12, 14.

A value above 14 is taken as +15 was transmitted, and a value below -14 is assumed to be -15 was transmitted. This is done for the real and imaginary parts. These values obtained are then decoded into 4-bit values for the real and imaginary parts, as illustrated in Table 1.

Hardware Implementation and Real-Time Results: HDL Workflow Advisor is used to implement the system in the hardware.This HDL Workflow Advisor is illustratedin Figure 15. From this figure, we observed thateach process must be passed on successfully to move on to the following procedure. All the functions from the first to the last must be passed successfully for the system to be run onto the FPGA, which is the ZEDBOARD.

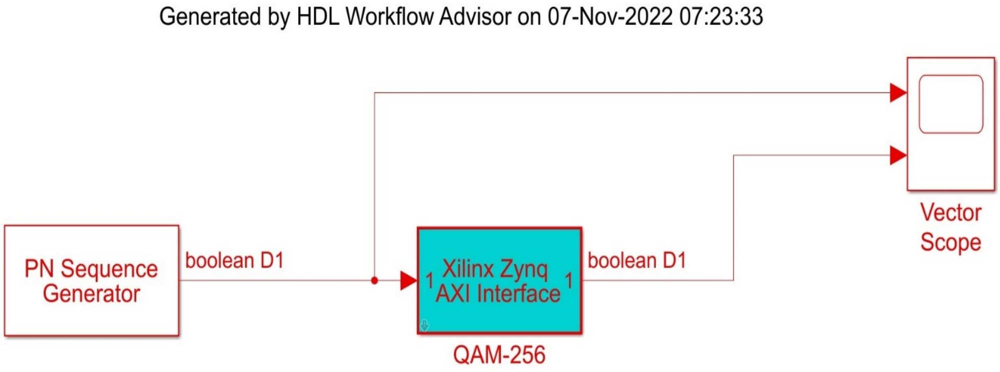
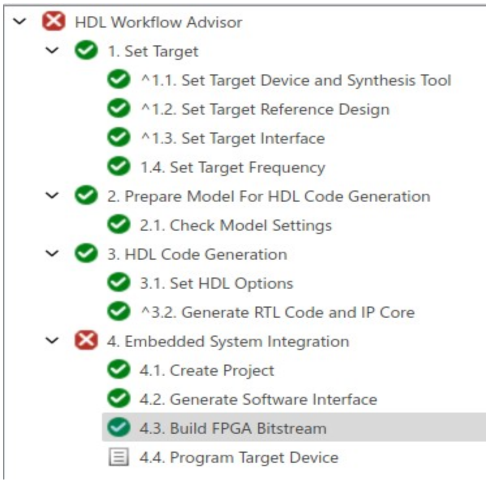

Generated by HDL Workflow Advisor on 07-Nov-2022 07:23:33
PN Sequence Generator
boolean D1
1
Xilinx Zynq AXI Interface
1
boolean D1
QAM-256
Vector Scope

Fig. 14. Generated Software Interface Model

The HDL Workflow Advisor generates the entire system's VHDL codes and the Software Interface Model. All the codes generated by the HDL Workflow Advisor are perfect if the system is perfect without errors. After generating the VHDL codes, the Advisor also developed the ".bit" file for downloading onto the FPGA. The C-codes for the ARM processor are generated on the go when we run this Software Interface Model. This C-code is then compiled into a ".exe" file to download onto the ARM processor. Figure 14 illustrates the Software Interface Model generated automatically by the Workflow Advisor. When we hit the Run button on the Software Interface Model, the Model runs in real-time, whose results are depicted in the next section. Figure 15 illustrates the completed processes for this design. The last function, the "Program Target Device", downloads the ".bit" file onto the FPGA on the ZEDBOARD. The Software Interface Model of the QAM-256 transceivers is generated in process 4.2. Finally, the FPGA bit-stream is generated in process 4.3, as seen in Figure 15.The real-time results for the 256-QAM are illustrated in Figure 16.The figure shows that the upper bit-stream is the transmitted bit-stream, and the lower bit-stream is the received bit-stream. Also, we can observe from this figure that apart from a slight latency, the sent and the received bit-streams are identical.



☒ HDL Workflow Advisor
- ✓ 1. Set Target
  - ✓ ^1.1. Set Target Device and Synthesis Tool
  - ✓ ^1.2. Set Target Reference Design
  - ✓ ^1.3. Set Target Interface
  - ✓ 1.4. Set Target Frequency
- ✓ 2. Prepare Model For HDL Code Generation
  - ✓ 2.1. Check Model Settings
- ✓ 3. HDL Code Generation
  - ✓ 3.1. Set HDL Options
  - ✓ ^3.2. Generate RTL Code and IP Core
- ☒ 4. Embedded System Integration
  - ✓ 4.1. Create Project
  - ✓ 4.2. Generate Software Interface
  - ✓ 4.3. Build FPGA Bitstream
  - ☰ 4.4. Program Target Device

Fig. 15. The HDL workflow advisor processes passed

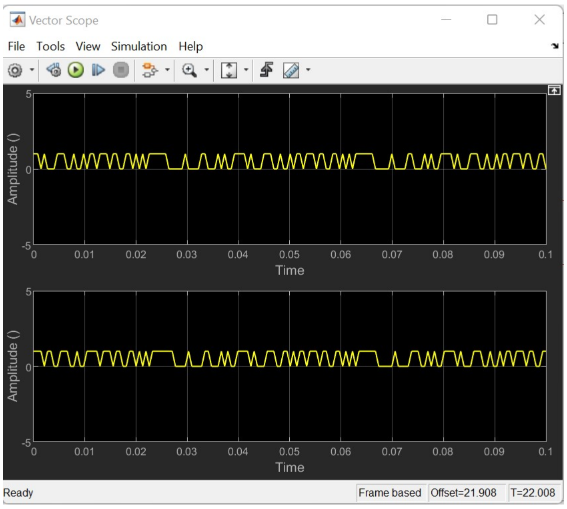


Fig. 16. Real-Time Results

CONCLUSION

The 256-QAM transceiveris designed and implemented using MATLAB / SIMULINK. For the different subsystems, the circuit is designed and tested in real-time and found to be working satisfactorily.The (8,4) Block Coding Decoding is working fine. The Modulator and Demodulator subsystems are designed and tested in real-time and are found to be working satisfactorily. The Detector and the LPF Subsystems were developed and tested, and were found to be working satisfactorily. The VHDL and the C-codes for the entire system are generated automatically by the HDL Workflow Advisor, and the system runs in real-time and works fine. The codes generated are perfect without requiring any debugging. The transmitted and the received bit-streams are seen to be identical except for a small latency, which is obvious. The Model-Based Design approach used here has saved considerable time in reaching the final implementation stage.

Ethical Approval- No human or animal studies were done in this manuscript.

Availability of supporting data- no supporting data is there for this manuscript.

Competing interest- The author has no conflict of interest to declare and has no financial interest to report. The author certifies that the submission is original work and is not under review at any other publication.

Funding- No funding has been there for preparing this manuscript.

Author’s Contribution- the corresponding author has prepared the whole manuscript. No other author is there.

Acknowledgments- Not Applicable.

REFERENCES

1. Dinh A.V. and R.J. BoltonJ., "A High-speed Baseband Transceiver for Wireless Communications Using QAM", 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233), 26-28 August 2001, pp. 631 – 634.
2. James Bishop, Jean-Marc Chareau, and Fausto Bonavitacola, "Implementing 5G NR Features in FPGA", 2018 European Conference on Networks and Communications (EuCNC), 18-21 June 2018, pp. 373 – 377.
3. Stefano Chinnici, Carmelo Decanis, Andrea Quadrini, and Dan Weinholt, "HIGH ORDER M-QAM TRANSCEIVER FOR GIGABIT RADIO MICROWAVE TRANSMISSION: FPGA TEST CHIPSET AND ASIC DESIGN", 2011 IEEE 9th International New Circuits and systems conference, 26-29 June 2011, pp. 297 – 300.
4. Satoshi Iino, Masaharu Ito, Yasushi Wada, Toshihide Kuwabara, and Tsunehisa Marumoto, "120 GHz 256QAM GaAs Transceiver Module for Over-10Gbps Wireless Communications", 2015 IEEE Compound Semiconductor Integrated Circuit Symposium (CSICS), 11-14 October 2015.
5. Wen-Chih Kan and Gerald E. Sobelman, "MIMO Transceiver Design Based on a Modified Geometric Mean Decomposition", 2007 IEEE International Symposium on Circuits and Systems, 27-30 May 2007, pp. 677 – 680.
6. Nobuhiko Kikuchi, Takashi Yano, and Riu Hirai, "FPGA prototyping of Single-Polarization 112-Gbit/s Optical Transceiver for Optical Multilevel Signaling with Delay Detection", Journal of Lightwave Technology (Volume: 34, Issue: 8, 15 April 2016), pp. 1762 – 1769.
7. Akash Mecwan and Dhaval Shah, "Implementation of OFDM Transceiver on FPGA", 2013 Nirma University International Conference on Engineering (NUiCONE), 28-30 November 2013.
8. Anh Dinh, R.J. Bolton, R. Mason, and R. Palmer, "Multi-channel Multi-point Distribution Service System Transceiver Implementation", 1999 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368), 22-24 August 1999, pp. 242 – 245.
9. Md. Abdul Latif Sarker and Moon Ho Lee, "FPGA-BASED MIMO TESTBED FOR LTE APPLICATIONS", 2011 Eighth International Conference on Wireless and Optical Communications Networks, 24-26 May 2011.
10. Dhanasekar Subramaniyam and Ramesh Jayabalan, "FPGA Implementation of Variable Bit Rate OFDM Transceiver System for Wireless Applications", 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), 03-04 February 2017, pp. 343 – 346.
11. Noriaki Tawa and Tomoya Kaneko, “A 950MHz RF 20MHz Bandwidth Direct RF Sampling Bit Streamer Receiver based on an FPGA”, 2017 IEEE MTT-S International Microwave Symposium (IMS), 04-09 June 2017, pp. 594 – 597.
12. Zhuan Ye, John Grosspietsch, and Gokhan Memik, "An FPGA Based All-Digital Transmitter with Radio Frequency Output for Software Defined Radio", 2007 Design, Automation & Test in Europe Conference & Exhibition, 16-20 April 2007, pp. 1 – 6.

****