Journal of Computer Science

Original Research Paper

Software Defined Radio Implementation of a QPSK Modulator/Demodulator in an Extensive Hardware Platform Based on FPGAs Xilinx ZYNQ

Sérgio Bimbi Junior, Vitor Chaves de Oliveira and Gunnar Bedicks Junior

Electrical Engineering Faculty, Mackenzie Presbyterian University, São Paulo/SP, CEP 01302-907, Brazil

Article history
Received: 08-02-2015
Revised: 23-04-2015
Accepted: 27-05-2015

Corresponding Author:
Vitor Chaves de Oliveira
Electrical Engineering Faculty, Mackenzie Presbyterian University, São Paulo/SP, CEP 01302-907, Brazil
Email: vitor_chaves_oliveira@yahoo.com.br

Abstract: Software Defined Radio (SDR) technology enables wireless devices deployment with support to multiple interfaces in modulation formats and it has gained importance due to the current wireless standards proliferation. In order to activate these functionalities it is necessary to apply SDR inside a reconfigurable hardware such as the Field Programmable Gate Array (FPGA). In this study, design procedures developments are presented resulting in a Quadrature Phase Shift Keying (QPSK) modulator/demodulator based on a hardware architecture of the Xilinx FPGA Zynq family. Also in the modem's conception it was employed the Xilinx Vivado-tool combined with Matlab, Simulink and development with ISE/EDK apparatuses. As for the reconfigurable hardware platform applied to this, it was utilized the ZedBoard along with the Analog Devices FMCCOMMS1 radio module. In addition, it is noteworthy to indicate that the entire software-hardware solution was initially carried out through a simulation that achieved the system scheme, including the hardware platform, before any hard implementation of the QPSK modem itself. The central goal in this investigation is to fully demonstrate how to construct a feasible radio system through software as close to the state of the art as possible.

Keywords: Broadcast, Digital Reconfigurable Systems, FPGA, QPSK Modem, Software Defined Radio, VHDL Development

Introduction

The Software Radio (SR) that later became defined by the term Software Defined Radio (SDR), was first introduced by Mitola III (1992) as an approach for the development of a radio reconfigurable platform programmable through software (Marpanaji *et al*., 2007). These SDR systems became evident after the development of a large number of wireless communication systems. So the terminal equipment of telecommunications carriers had to adapt to the standards evolution. These patterns require a transceiver capable of supporting multiple encryptions, modulations and digital signal processing techniques. In addition the demand for high Quality of Service (QoS) in all engineered communications is a growing concern (de Oliveira *et al*., 2015; Oliveira *et al*., 2013; Song and Yao, 2010; Sailaja *et al*., 2014).

As for the predominantly employed hardware in the SDR it is necessary to reference the Field Programmable Gate Array (FPGA). The FPGA platforms currently are of interest because of its excellent performance, low power consumption and total or partial reconfiguration capabilities (Kazaz *et al*., 2013). Another relevant characteristic is that the digital signal processing based on hardware implementation via FPGA has applications in numerous areas such as mobile communication systems, Voice Over IP (VoIP), multimedia, radio broadcasting, radar, satellite systems, etc. Furthermore, the motivation to use Phase Shift Keying (PSK) modulation is that it is widely used in existing wireless technologies and, nowadays important applications such as Long Term Evolution (LTE), LTE Advanced, IEEE 802.11b-1999 and IEEE 802.11g-2003 and IEEE 802.15.4 (Yano *et al*., 2014; 2013) modulation utilizes Quadrature Phase Shift Keying (QPSK) which consists of a quadrature PSK modulator, allowing a superior, i.e., cost effective, spectrum usage (Lalge and Bhandari, 2014; Violas *et al*., 2014).

General System Architecture

In the block QPSK Generator two successive bits in the sequence are grouped into a single symbol. At the input of the modulator a series of conversions are performed on the input bits, subsequent to this process a conversion from unipolar to bipolar signal is performed. After these processes it is necessary to filter the signals to avoid distortion of the received signal, which is characterized as Inter Symbolic Interference (ISI). After filtering, the modulated signal is ready for to be used. At the end of the process the modulated signal is generated in two separate parts shown in I and Q. These signals are summarized and translated into an analog signal characterized by an analog digital converter. The QPSK analog signal is characterized mathematically by Equations 1-3:

$$sQPSK = dI(t)\ cos(2\pi f0t) - dQ(t)\ sin(2\pi f0t) \quad (1)$$

$$dI(t) = \sqrt{\frac{2E}{T}} cos[(2i - 1)\frac{\pi}{4}] \quad (2)$$

$$dQ(t) = \sqrt{\frac{2E}{T}} sin\left[(2i - 1)\frac{\pi}{4}\right] \quad (3)$$

where, E is the energy of the symbol and can be viewed in Equation 4:

$$E = 2Eb \quad (4)$$

In block DFE for Radio TX raised cosine filters are used, which consists of a flat portion and a portion with a decay rate (roll off). Where the frequency f1 and the Nyquist range width B_0 are related by the roll off factor that can be seen in Equation 5:

$$\propto = 1 - \frac{f1}{B_0} \quad (5)$$

It is also placed an interpolator filter, which is being applied to increase the sampling frequency. In the DFE block for Radio RX its input has a decimator filter where it will reduce the sampling frequency and also perform a pulse shaping. Subsequent to this process, the symbols of recovery blocks are connected, where it is possible too b served spectrum and constellation.

FPGAs in SDR Applications

A Software Defined Radio elaboration (Priya *et al*., 2013) implies in a particular project and conventional procedures for a transceiver system. Therefore, initially, during the process of formatting a hardware platform, it is required to reach a compromise between desired flexibility, modularity, scalability and SDR system performance. Flexibility is the system's ability to suit air interfaces and protocols even if those protocols are still under definition and the system's modularity permits easy adaptation and/or modernization of the current system, targeting migration to new technologies. Whereas scalability which is related to modularity, allows in a general scope, the system to be increased with an outlook on capacity expansions, such as added channels in a base station. While the system's efficiency links to three features which can be quantized through energy consumption, system cost and computational magnitude (Lathi and Ding, 2009; Kostic, 1994; Wang *et al*., 2014).

Besides this, in a hardware platform, there are commonly used components in charge of formatting the hardware's digital portion, they can be listed as Digital Signal Processing (DSP), Application Specific Integrated Circuit (ASIC) and FPGA. Each of these components provides a certain degree of reprogramming, this is a prerequisite for an SDR construction. The DSPs are based on microprocessor architectures and are programmable in high level languages such as C/C ++. This aspect, briefly mentioned above, directly contributes for the system to gain a high flexibility in project design, but with this architecture systems are more likely to fall short in their processing consummation. In view of this element run its operations in a sequential manner, by highly complex systems, time efficacy becomes compromised. One way around this problem would be to use more elements, running routines in order to pursue a parallelism (Blech *et al*., 2010; Dowrla and Rani, 2013; Mukesh *et al*., 2014).

Other plausible solutions are the utilization of ASICs and FPGAs. ASICS enables an optimized system implementation regarding the use of silicon, watt hour consumption and overall performance, however they possess a high cost for initial operation. On the other hand, once compared to the DSPs (Cummings and Haruyama, 1999), the FPGAs allow a high reconfigure ability level as well as an elevated standard of parallelism for operations to be carried out. It is possible to deploy a large number of objects that can be ran simultaneously, there are also no problems with respect to numerical representations and variables accuracy and, in recent years, FPGAs tools obtained significant enhancements that enable systems development at a high level starting from a base model. This architecture provides to the user an easy development applicable in three different languages: Very High-Level Design Language (VHDL), Verilog and Hardware Description Language (HDL). Then, when choosing one of these three mentioned elements, corresponding to the digital hardware part, it is necessary to exert a compromise between flexibility, processing 'speed' and power consumption. Lately, appeared on the market hybrid integrated circuits as ZYNQ Xilinx Family, offering to

the developer a combination of such architectures (Dowrla and Rani, 2013). The primary purpose in these conjugated constructions is to achieve optimal energy efficiency preserving system requirements for performance. In applications with families in this framework, it is possible to construct of a fully reconfigurable system that has the ability to adapt to new hardware elements of the Front End architecture and to the Analog to Digital Conversion (A/D) and/or Digital to Analog Conversion (D/A) of an SDR. With the partial/dynamic reconfiguration it is possible to address energy and general performance issues. Such an important dynamic feature has not been addressed in the literature, which constitutes in a completely novel development with possibilities to a SDR with high performance in regarding cost and energy consumption and also possesses an available microcontroller architecture, the Advanced RISC Machine (ARM) for ZYNQ, that provides an easy competence to embed operating systems, user interfaces, peripheral connection, use of commercial libraries.

SDR Based on FPGA ZYNQ

The SDR platform proposed uses the ZedBoard development kit (Avnet, 2015) and the radio module FMCOMMS1 (Analog Devices, 2015). As for the QPSK modulator/demodulator design it was conducted through Matlab/Simulink and System Generator. It is an SDR system, consisting in a full duplex QPSK modulator/demodulator (MODEM) in simulation utilizing the Matlab/Simulink tool. In this simulation runs a system capable of transmitting (modulate) and receiving (demodulate) for system perform its execution. The system was implemented using the Xilinx EDK tool with the codes developed directly in VHDL language (codes and instantiation). So it was developed a QPSK modulator/demodulator for proof of concept and testing according to Fig. 1.

For this application the FMC-LPC connector is accessed through a Micro Blaze microcontroller, emulated in software, where accesses are run through DMAs, Direct Memory Accesses and the Analog Devices FMCCOMMS1 radio module is accessed to generate the QPSK modulator/demodulator so that the system performance is analyzed. In a wireless digital transmitter it is employed a DUC, Digital Up Converter, in order to interpolate the band symbols for a faster sampling rate (Zeeshan and Khan, 2013) connected to the DAC, Digital to Analog Converter. Similarly, a receiver may employ a DDC, Digital Down Converter, to reduce the sampling rate from ADC, Analog to Digital Converter, baseband interface with respect to the Nyquist bandwidth theorem (Nyquist, 1924). The input signal is important in digital signal processing for providing a clear understanding of the sampling rates in the system. This experiment explores the Simulink symbol recovery model, synchronous to identify the sampling rates along the QPSK transmitter.

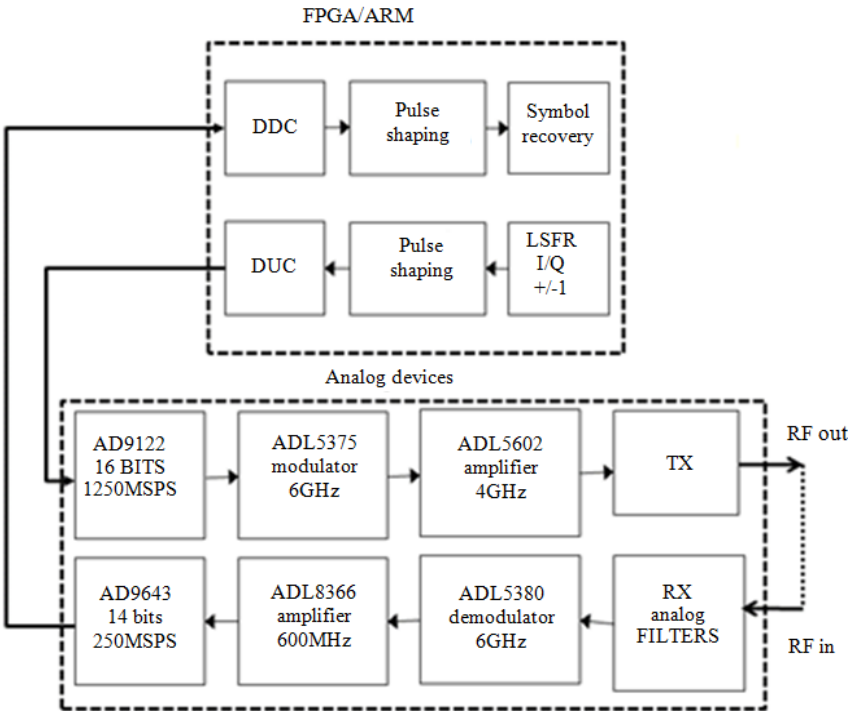

Fig. 1. QPSK Model based on FPGA ZYNQ

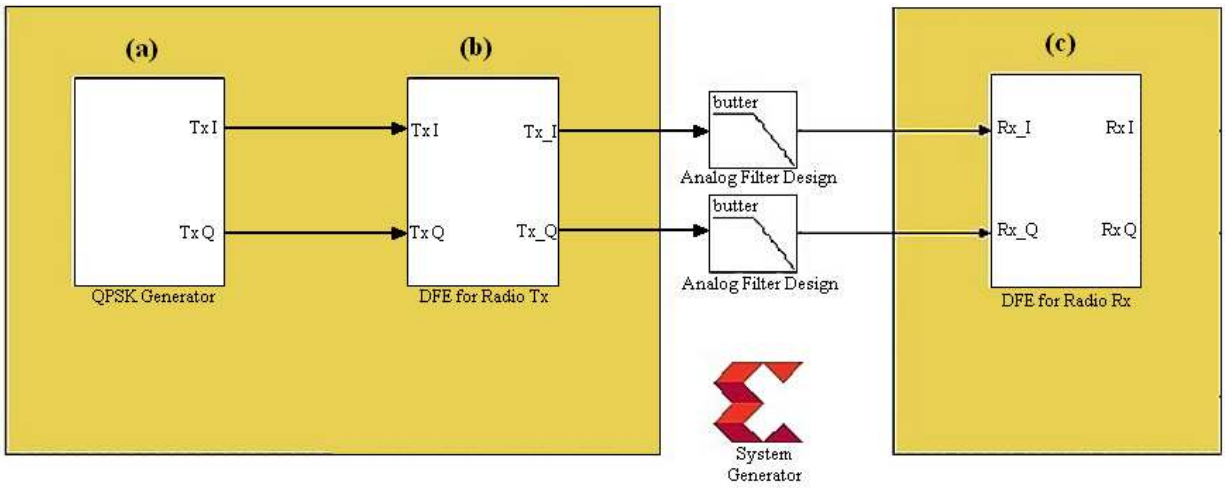

Fig. 2. QPSK Model-Simulink (a) QPSK generator with LFSR block (b) Transmission Filters (c) Reception Filters

The model is divided into four stages, namely: QPSK Generator, Digital Front End (DFE) for Radio TX, DFE for Radio RX and Symbol Timing Recovery. The QPSK generator stage is responsible for the QPSK symbol generation and UP Sample blocks implementation, in the DFE for Radio RX block it is enclosed the raised cosine filter and interpolators filters that precede the signal transmission stage. In the DFE for Radio RX block the decimator filter and the raised cosine filter are positioned and, finally, in the last stage the Symbol Timing Recovery is contained in the block capable of performing the symbol reconstruction. For the execution of the four blocks contained in the project are used settings parameters that describe the system behavior, which has a sampling frequency (Fs) that is D times higher than the symbol rate (Rs). In this project the Rs used is 1.5Msps and D = 8, resulting in an Fs of 12MHz. The described blocks can be seen in Fig. 2.

QPSK Generator

In Simulink, the signals are propagated from the source to the destination, from left to right. The source of the QPSK symbol generator is a pair of linear displacement registers (LFSR). The LFSRs generate pseudo-random bit patterns that are transformed into +1/-1 pulses as QPSK I and Q components. Where D is the decimation filter applied and Fs is the symbol frequency applied to the system. At the reception is contained a block with a decimator filter and a raised cosine filter, in the last one it is held the symbol reconstitution block. Within the QPSK generator stages are included the PN Sequence and QPSK blocks, as can be verified through Fig. 3.

In the PN Sequence are included the LFSRs blocks generating pseudo-random bit patterns that are processed at +1/-1 pulse components as QPSK I and Q symbols, shown in Fig. 4.

These blocks are parameterized according to patterns that characterize the symbols generation, the sequence employed is of a Fibonacci type (Lathi and Ding, 2009), which is characterized by shift registers dependent to bits amount with 20 shift registers, as seen in Fig. 5. In this method, the bits positions, allocated in Gate type XOR (exclusive or gate), affecting the next system state in a bit vector with a 2^{19-1} dimension (Number of bits in LFSR), with a hexadecimal value of $0\times47213_h$ and the mathematical representation of this system can be seen in Equation 6:

$$\propto^i = x^0 + x^4 + x^5 + x^6 + x^9 + x^{14} + x^{17} + x^{18} \quad (6)$$

The polynomial feedback represents the bits that influence the next state written in a hexadecimal form, commenced with the $0\times3F_h$ hexadecimal as the initial value. The reinterpret block is used to format the type of output signal, i.e., no signal, with signal, floating point or by specifying the binary point, keeping the original input value. In this case the reinterpret block is converting the Ufix_1_0 signal (no signal) to Fix_1_0 (complement of two). The Threshold blocks characterize the signals limits, that is, the block responds logic level -1 if the input is negative and otherwise it presents logic level 1, as shown in Fig. 6.

The block's output is a two-bit number with signal. In the simulation for the QPSK block it is obtained rectangular shaped rhythms, modulated in pulses I and Q with 4 samples per symbol, as shown in Fig. 7.

Within the QPSK block's architecture are included the Up-Sample blocks with resampling rate (factor 4), effectively creating new samples, which can optionally be set to '0's between the existing input samples. This is known as zero padding. The block creates QPSK symbols with different return to zero for pulses with distinguished amplitudes, i.e., +1/-1 with a 6Msps sampling frequency, the logic states of the QPSK signal can be understood through Table 1.

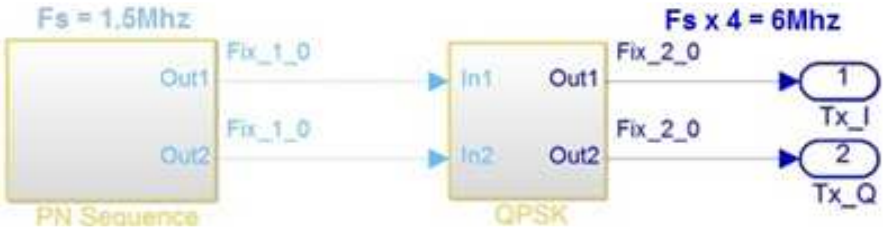

Fig. 3. QPSK symbol generator block

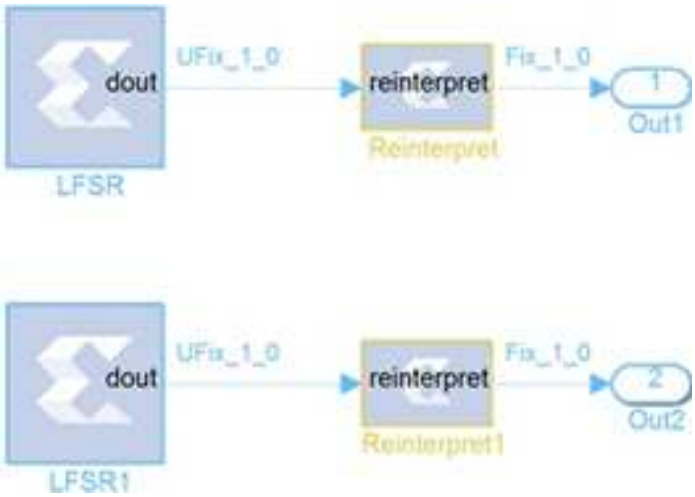

Fig. 4. LFSR Block

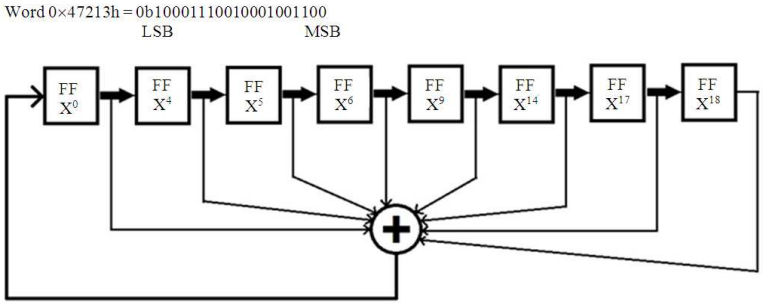

Fig. 5. Flip Flops Fibonacci sequence

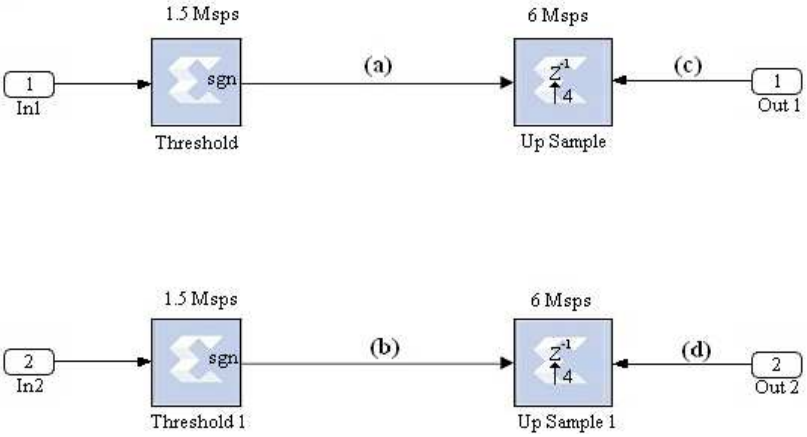

Fig. 6. Up-sample blocks

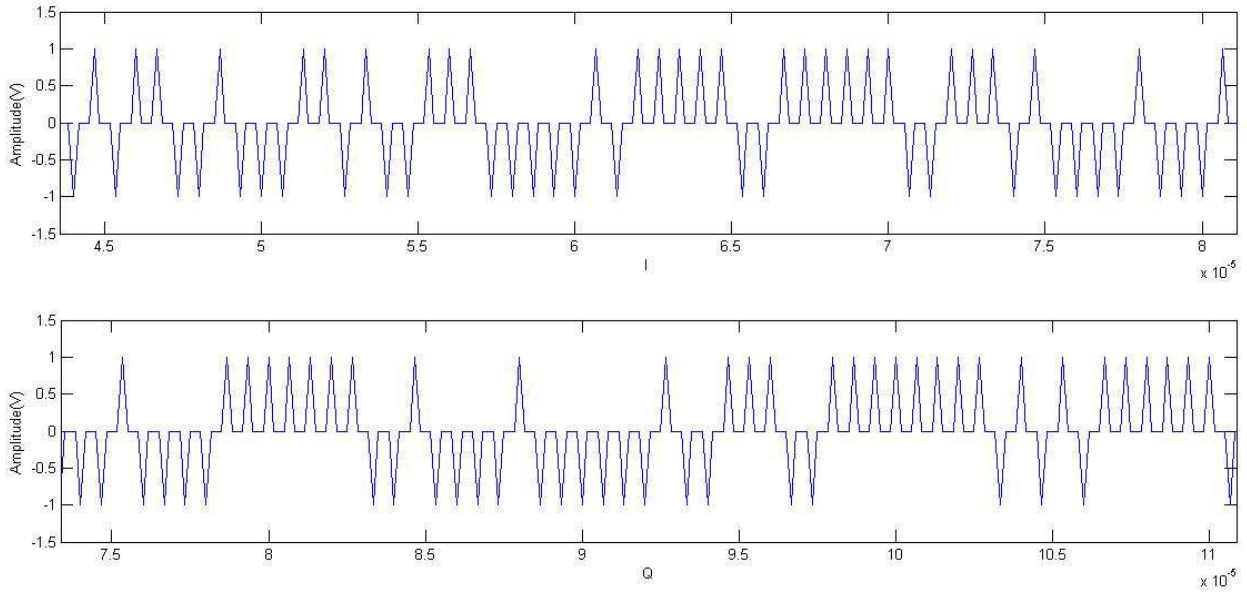

Fig. 7. Generated signals from the QPSK system

Table 1. QPSK States characterization

Bit 0	Bit 1	QPSK
0	0	+1i+1j
0	1	-1i+1j
1	0	-1i-1j
1	1	+1i-1j

Transmission Filters

In this point it is created a signal chain of Digital Up Converter (DUC)/Digital Down Converter (DDC) for the QPSK module with a 6Msps sampling frequency using 1,5MHz bandwidth. In the transmitter DUC interpolates the signal to the AD9122 dual-DAC 16-bit AD-FMCOMMS1@24Msps and, in order to cushion the QPSK pulses, filters are used with a limited bandwidth in 1.5MHz, the filter applied to the TX block is a low pass with bandwidth of 1.5MHz. These transmission filters blocks can be seen in Fig. 8.

The details inserted into the transmission filters blocks are demonstrated in Fig. 9.

In XILINX_RRC_FILTER_TX block is arranged a raised cosine filter, which is implemented with assistance from the FIR_Compiler_6.3.2 block. Its coefficients are calculated by filter builder, a Matlab tool, with the filter response type set to the Pulse Shaping mode. The coefficients are saved in rrc_coef_gui variable, with an order of 80 and 4 samples per symbol and a 0.25 roll off factor, besides working in Window mode type Finite Impulse Response (FIR), as shown in Fig. 10.

In the XILINX 2X INTERPOLATION HALFBAND FILTER TX1 block is arranged an interpolation filter with half-band response, this filter is implemented with the support of FIR_Compiler_6.3_phase_0 block. Their coefficients are calculated by firhalfband Matlab tool with the type of filter response in Half Band mode. The coefficients are stored in the h variable, with an order of 18, besides working in the pass band mode type FIR. The filter's response and its settings can be seen in Fig. 11 and 12, respectively.

In the XILINX 2X INTERPOLATION HALFBAND FILTER TX2 block is arranged an interpolation filter with a half-band response, this filter is implemented with the support of the FIR_Compiler_6.3_phase_0 block. Their coefficients are calculated by firhalfband Matlab tool with the type of filter response in halfBand mode. The coefficients are stored in the h variable, with an order of 1, besides working in pass band mode type FIR. The filter's response and its settings can be seen in Fig. 13 and 14, respectively.

With specific filters the coefficients are employed to the raised cosine filter block using the Xilinx-FIR Compiler 6.3 block, where the coefficients are applied in its vector entry. The system contains a Roll off of 1, 1.5Msps Rs, four M states and decimator filter in the order of 8, with these data it is concluded that its bit rate can be determined through Equations 7 and 8:

$$Rb = Rs * \left(Log_2\left(M\right)\right) \tag{7}$$

Bit rate is 3 Mbps. So:

$$Bw = \frac{1}{2} * \left(1 + r\right) * Rs \tag{8}$$

Bandwidth is 1.5MHz.

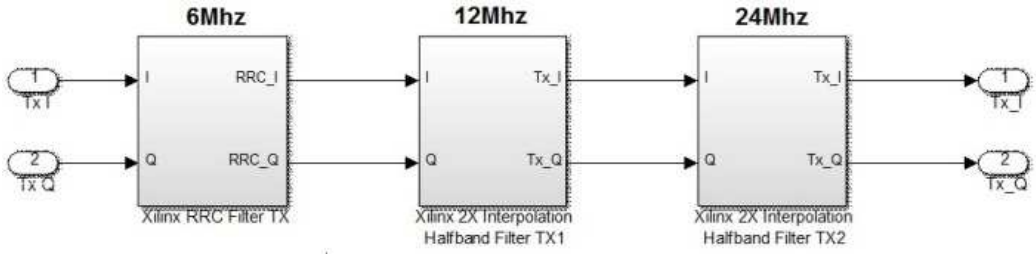

Fig. 8. Transmission filters blocks

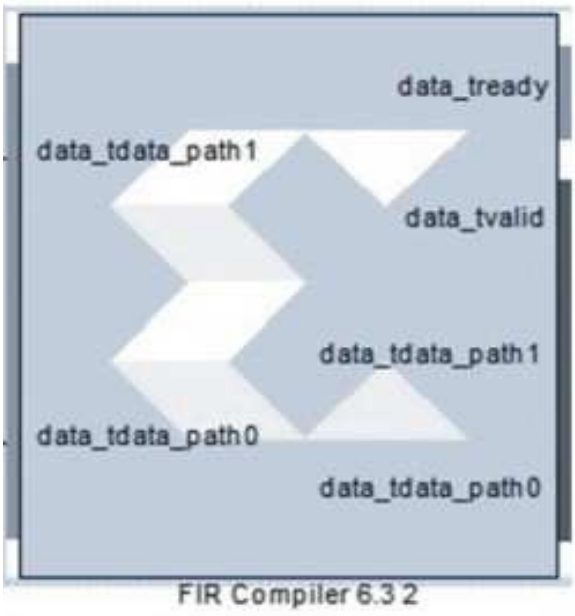

Fig. 9. Transmission filters details

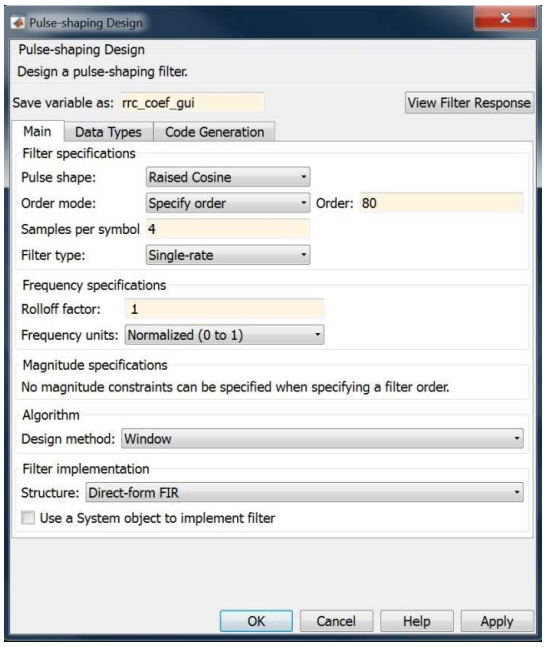

Fig. 10. Raised cosine filter configuration

Reception Filters

The reception filters blocks are demonstrated in Fig. 15, where the Xilinx 2X Decimation Half band Filter RX and Xilinx RRC Filter RX blocks are present. In the Xilinx 2X Decimation Half band Filter RX block is arranged a decimator filter with half-band response, this filter is implemented with the aid of FIR Compiler 6.3 block phase 0. Their coefficients are calculated by Matlab fir half band tool with the type of filter response in halfBand mode. The coefficients are saved in the same transmission coefficients, as well as working in pass band mode type FIR. In the Xilinx RRC Filter RX block is arranged a raised cosine filter, this filter is implemented with the support of the FIR Compiler 6.3.2 block. Their coefficients are calculated by Matlab filter builder tool with the type of filter response set to the Pulse Shaping mode. The coefficients are saved in the rrc_coef_gui variable, with an order of 80 and 4 samples per symbol, a roll off factor of 1, besides working in Window mode Finite Impulse Response (FIR) type.

Symbol Recovery

In the Symbol Timing Recovery block are arranged the blocks capable of performing the symbol recovery, it is relevant to point out that such block was not implemented in the FPGA, it was only used for simulation purposes. The input of the first block is the Real-Imag to Complex1, which is responsible for converting real or imaginary inputs to an output signal with a complex value. In this block, the inputs might have equal dimensions or one of them can be a matrix and the other a scalar input. If the block has a matricidal input, the output is a matrix of the same dimension. The Fractional Delay block, i.e., a block with fractioned interruptions, exercises control over of a more precise phase index for the Table Lookup algorithm, due to maintaining a whole as well as fraction of the address table. The Timing Error Detector and Loop Filter block, that retrieves 4 samples per symbol, has the purpose of recovering the synchronized symbol. In it, the sampling frequency is synchronized to the frequency of the received signal symbol.

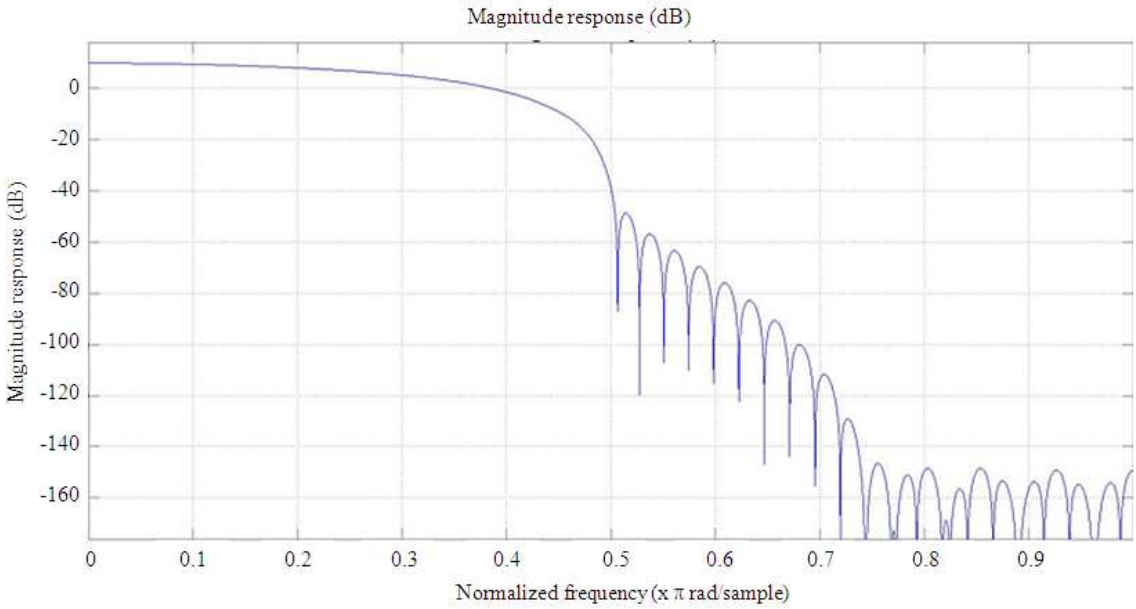

Fig. 11. Response of the XILINX 2X_INTERPOLATION HALFBAND FILTER TX1 filter block

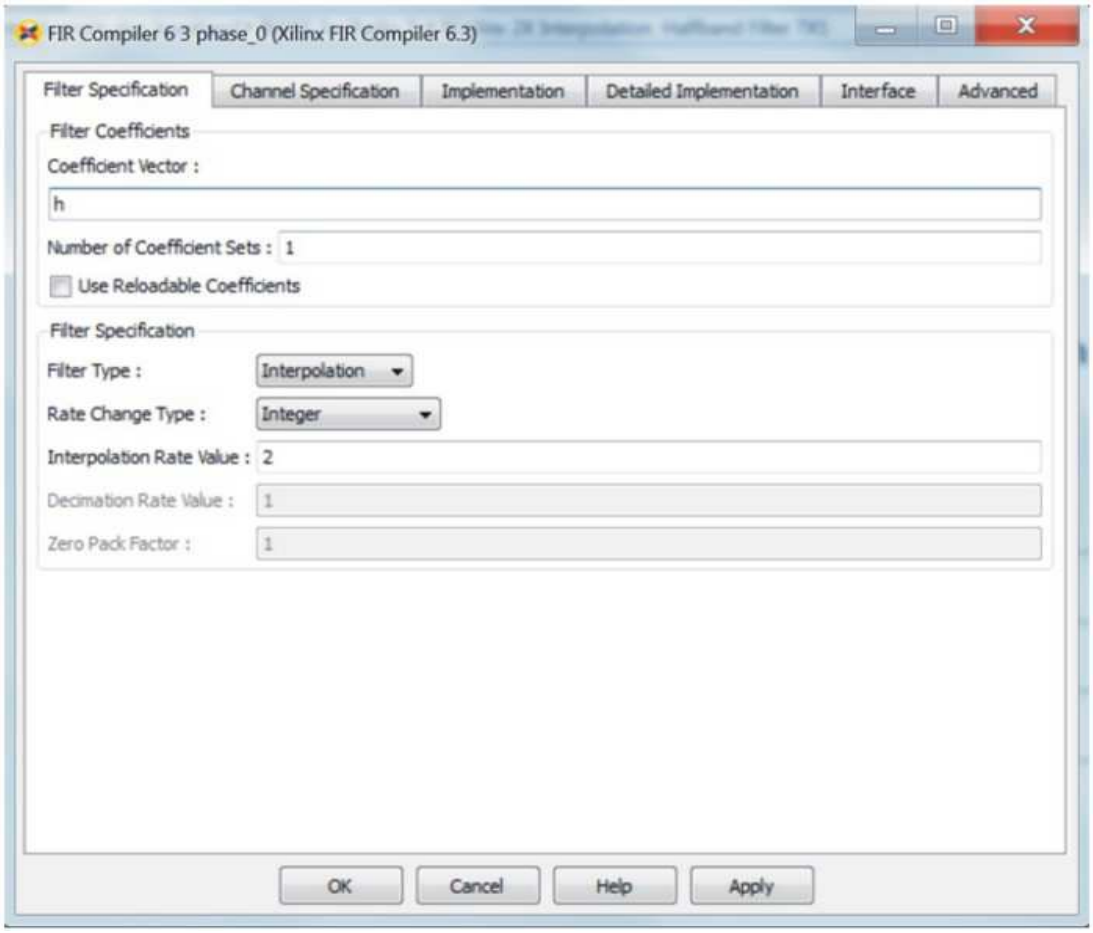

Fig. 12. Configuration of the XILINX 2X_INTERPOLATION HALFBAND FILTER TX1 filter block

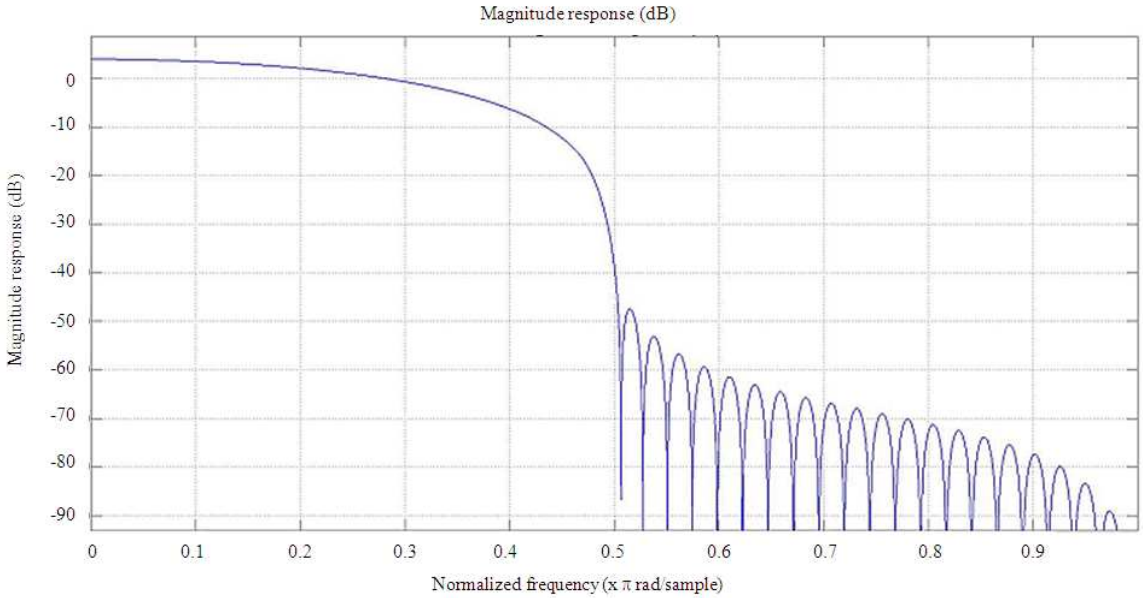

Fig. 13. Response of the XILINX 2X_INTERPOLATION HALFBAND FILTER TX2 filter block

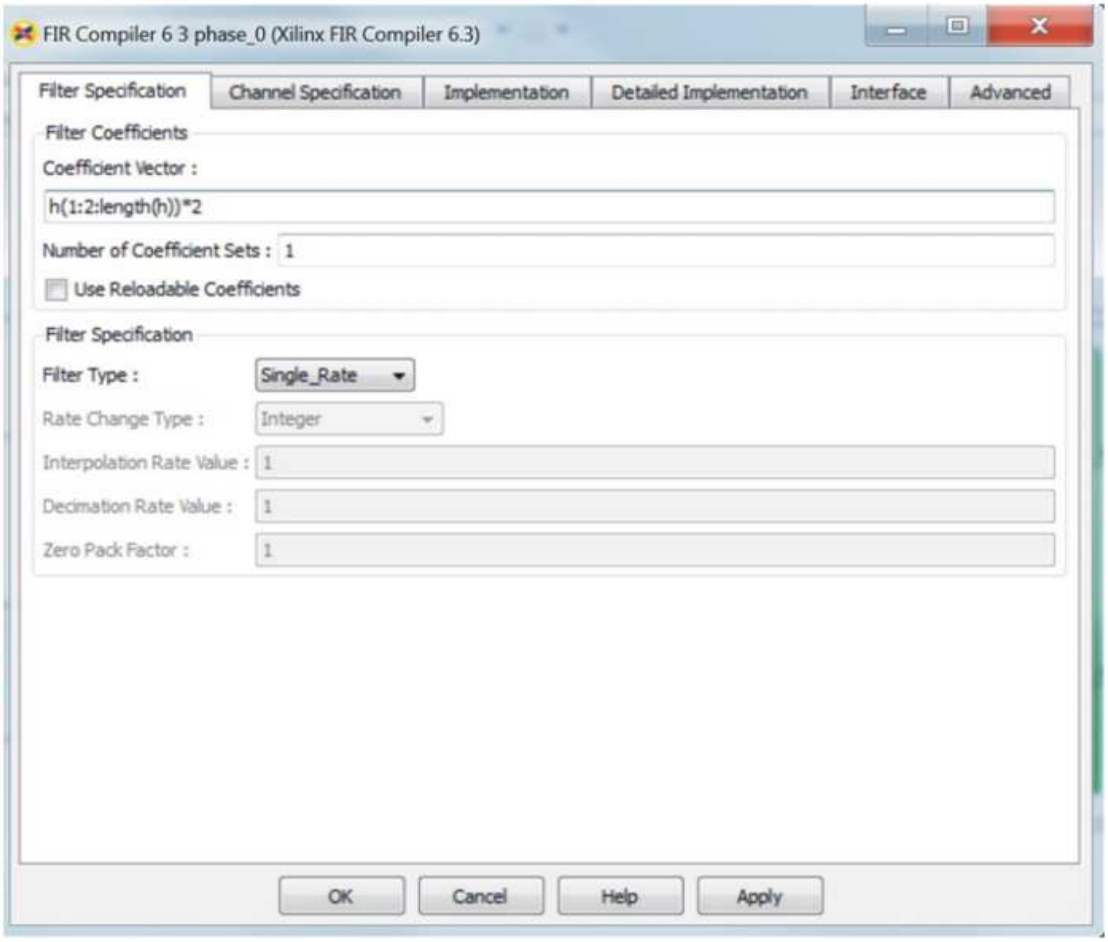

Fig. 14. Configuration of the XILINX 2X_INTERPOLATION HALFBAND FILTER TX2 filter block

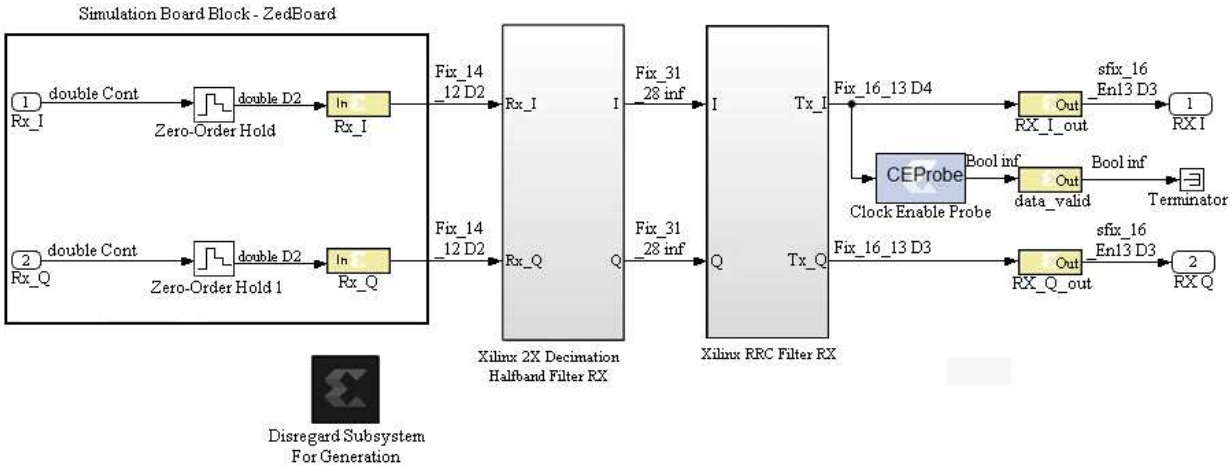

Fig. 15. Reception filter blocks-Xilinx 2X Decimation Halfband Filter RX

Results

Inside the proposed system, signals are acquired in the frequency and constellation domain coherent to a structure based on a QPSK modulator/demodulator. The system has as a resource the Analog Devices ADI IIO Scope tool that is incorporated into its architecture. With the embedded Linux this tool runs in order to verify the signals received by the system, working along with the project in a Loop Back scheme. The Fig. 16 and 17 show, respectively, the received signals relating to the spectrum and constellation shown in Analog Devices ADI IIO Scope tool.

With the generated files, the system is started via the SD Card, this process lasts for approximately 30 seconds, this process completed the system is fully started and in operation (transmitting and receiving). After the system has been fully started it is possible to connect the USB interface, mouse and keyboard for operations, in addition to the Linux graphical interface to execute user interaction. To select the operating frequency it is necessary to use Analog Devices ADI IIO Scope, it was selected the 371UHF channel in the center frequency of 611 MHz (reserved for radio astronomy), with it, the system transmits and receives (modulator/demodulator) the QPSK signal generated. The system will automatically open the Analog Devices ADI IIO Scope tool, where it sets the frequency that the QPSK digital signal transmits. With the digital channel configured, the system will transmit and receive digital QPSK signals. In order to perform more accurate testing it was built a system using Linux and GNU-Radio (Abirami *et al*., 2013), associated with a Digital TV Dongle for the reception of the transmitted spectrum and to display its constellation. The Model 1, executes the reception of the transmitted spectrum obtaining the 1.5 MHz bandwidth in baseband as shown in Fig. 18. The second model, built on GNU-Radio, features the system constellation reception, representing the demodulator, as shown in Fig. 19.

Discussion

The system presents efficiency once compared to ideal theoretical values (Divya, 2013), with the immense advent of dynamic reconfiguration, that is, to be able to reconfigure modulus transmitter and receptor, with this is possible to improve the efficiency of the system with the project already implemented.

It can be seen that the Bit Error Rate (BER) of the transmitted and received simulation signals are aligned to the theoretical rate of a QPSK (4-Quadrature Amplitude Modulation (QAM)), as shown in Fig. 20.

Conclusion

The article demonstrates a methodological and constructivist view on the implementation of a half-duplex QPSK modulator/demodulator utilizing an SDR with FPGAs devices of the ZYNQ family. In light of the foregoing hardware and software applied to FPGAs with a QPSK modem fully implemented in VHDL along with Analog Devices ADI IIO Scope and, once comparisons are made to the current state of the art, the following advantages are identified: Main processor modules simplification, network interface and Fast Fourier Transform (FFT) modules streamlined. These last two in traditional solutions are implemented into isolated blocks, in the designed construction they are entirely developed in the FPGA software architecture. Within the ZYNQ family system advantages held in its architecture, there is an ARM processor running any application peripherals with Linux, increasing system performance when communicating with the radio module Analog Devices FMCOMMS1 and

specific functions using the part that concerns FPGA for high speed applications. As a future proposal it is aimed the construction of a modulator/demodulator configured in a partial manner in order to minimize the FPGA utilization area envisioning a higher performance SDR regarding to the cost-benefit ratio and power consumption.

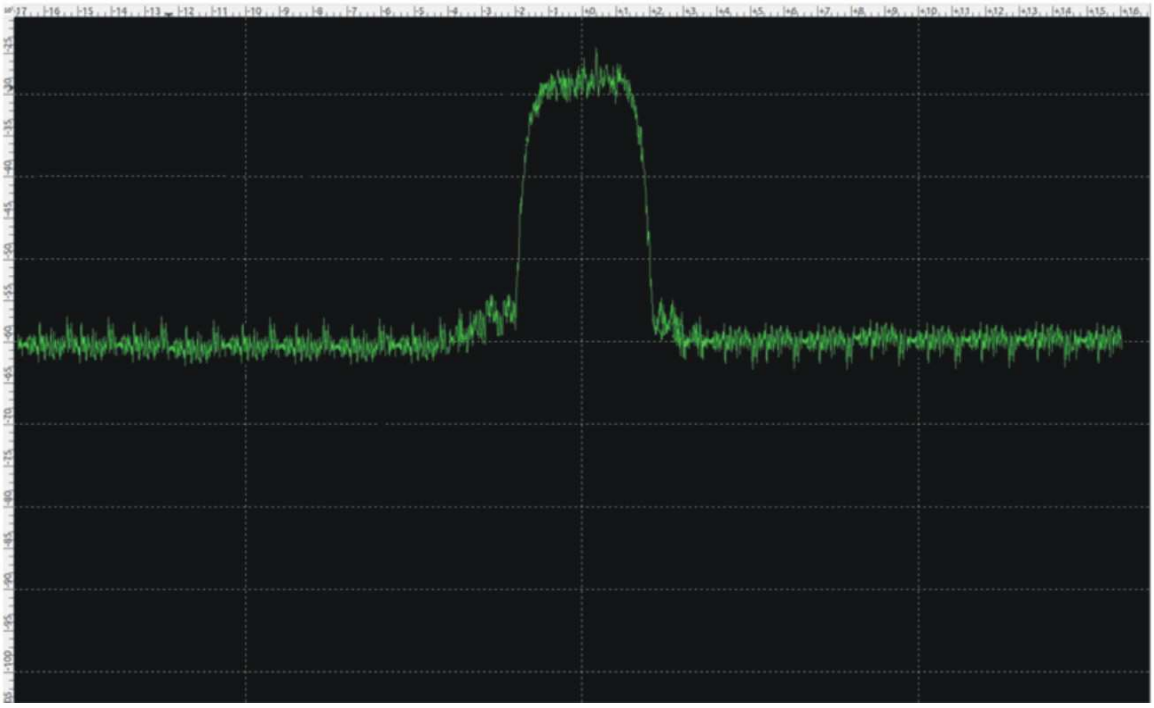

Fig. 16. Signals in the frequency domain-QPSK

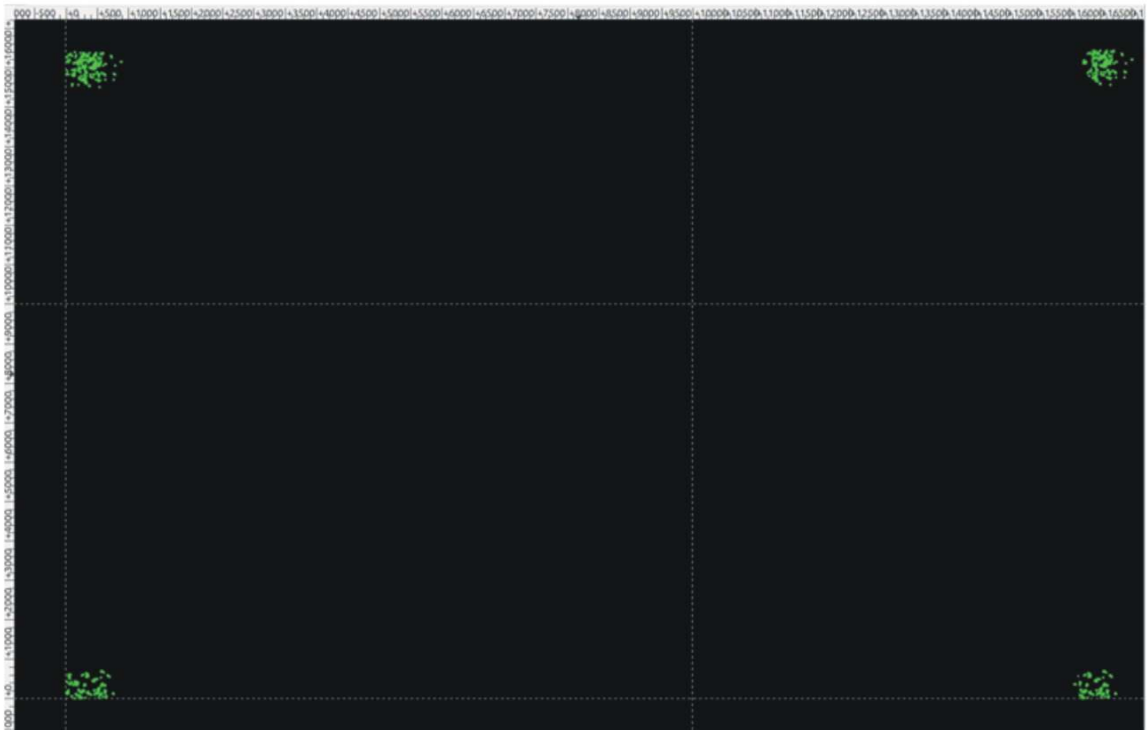

Fig. 17. Constellation obtained in the system receiver-QPSK

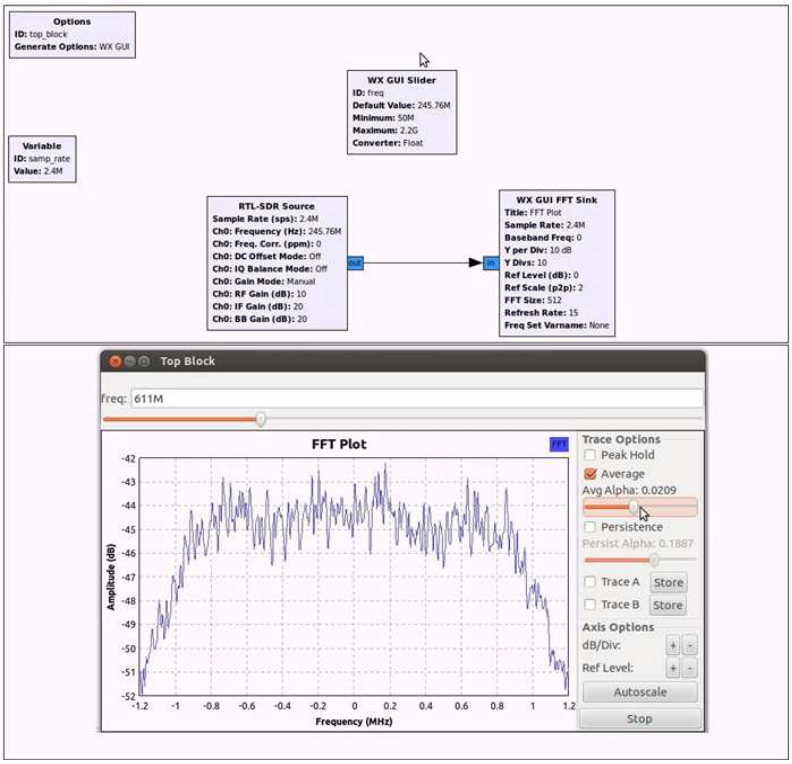

Fig. 18. GNU-Radio Model-Signal transmitted spectrum reception

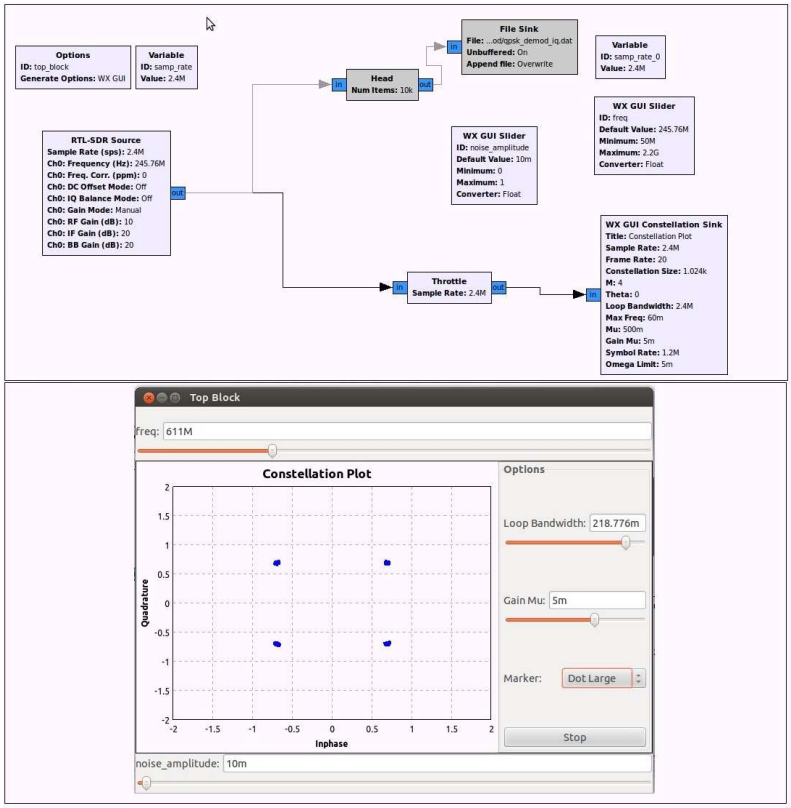

Fig. 19. GNU-Radio Model-Signal transmitted constellation reception

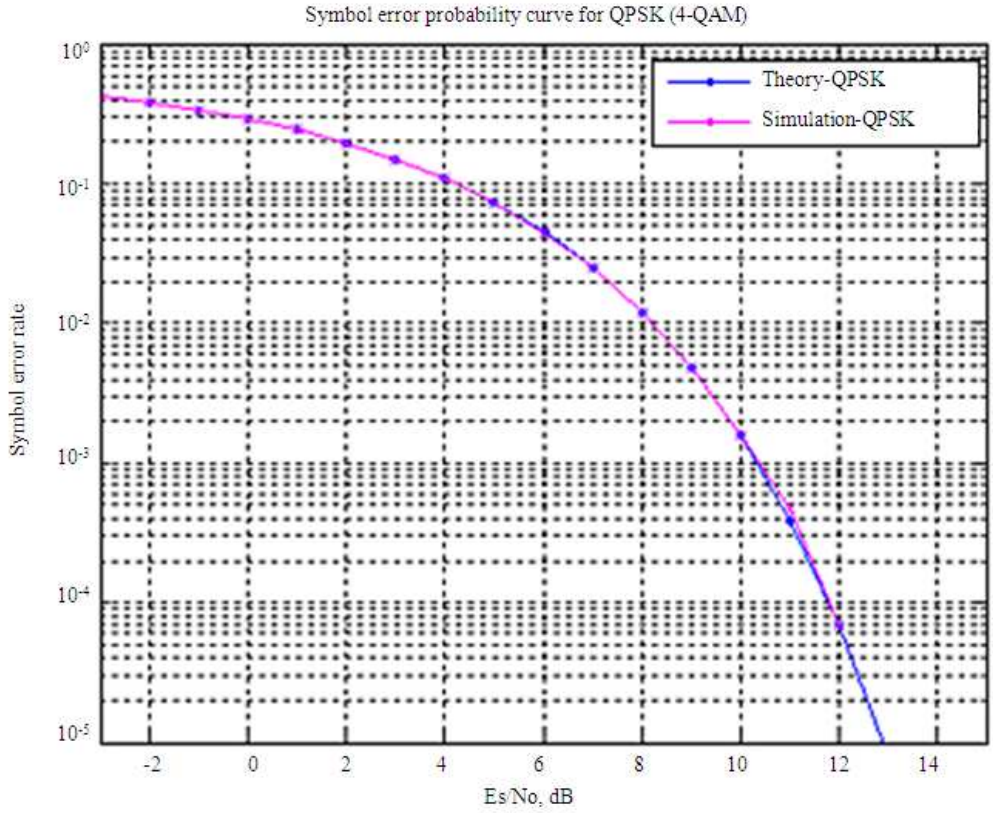

Fig. 20. Ber-Comparison of the theoretical curve with curve obtained in the system

Acknowledgement

The authors would like to thank Mackenzie Presbyterian University and their colleagues from the Radio Transmissions and Digital TV Laboratory.

Funding Information

The authors would like to thank Mackenzie Presbyterian University for sponsoring this project with scholarships and project funding.

Author's Contributions

Sérgio Bimbi Junior and Vitor Chaves de Oliveira: Research about state of art in SDR applications; Research about state of the art in modulation/demodulation applied in SDR applications; Research about hardware requirements for the project implementation; Implementation of SDR model in Matlab tool; SDR model Simulation in Matlab tool; SDR base model definition; Tests and conclusions about SDR Model; Model creation for tests, using GNU Radio (Receive QPSK signal); Models creation for digital filter (transmission/receiver).

Gunnar Bedicks Junior: Hardware specification (CPU and Radio); Project Orientation; Equations definition and appropriate math condition to implement SDR model; Performance evaluation and tests; Data analysis for sender and receiver signals, Establishment of best technical and tools for the project (Spectrum Analyzer and others models); Research to establish appropriate equations and mathematical models to apply on the blocks.

Ethics

The authors declare that they have no conflicting interests.

References

Abirami, M., R. Gandhiraj and K.P. Soman, 2013. Performance analysis of real time OFDM based communication system using GNU radio and USRP. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 1584-1590.

Analog Devices, 2015. AD-FMCOMMS1-EBZ functional overview. Analog Devices, Inc.

Avnet, 2015. ZedBoard Product Briefs. Avnet, Inc.

Blech, M.D., A.T. Ott, P. Neumeier, M. Moller and T.F. Eibert, 2010. A reconfigurable software defined ultra-wideband impulse radio transceiver. Adv. Radio Sci., 8: 67-73. DOI: 10.5194/ars-8-67-2010

Cummings, M. and S. Haruyama, 1999. FsPGA in the software radio. IEEE Communi. Magazine, 37: 108-112. DOI: 10.1109/35.747258

de Oliveira, V.C., G.B. Junior and C. Akamine, 2015. Methodology to asses internet protocol connection availability: A prerequisite for feasible video stream through content delivery networks. J. Comput. Sci., 11: 426-437. DOI: 10.3844/jcssp.2015.426.437

Dowrla, S.J. and K.J. Rani, 2013. VLSI implementation and design of digital modulation using xilinx. Int. J. Res. Comput. Communicat. Technol., 2: 733-737.

Kazaz, T., M. Kulin and M. Hadzialic, 2013. Design and implementation of SDR based QPSK modulator on FPGA. Proceedings of the 36th IEEE International Convention on Information and Communication Technology Electronics and Microelectronics, May 20-24, IEEE Xplore Press, Opatija, pp: 513-518.

Kostic, M.I., 1994. Digitalni Telekomunikacioni Sistemi. 1st Edn. Naucnaknjiga Beograd. Ivo M. Kostić, ISBN-10: 8623421124, pp: 626.

Lalge, A.M. and S.U. Bhandari, 2014. QPSK system implementation on FPGA. Int. J. Emerging Trends Technol., 1: 139-143.

Lathi, B.P. and Z. Ding, 2009. Modern Digital and Analog Communication Systems. 4th Edn., Oxford University Press, New York, ISBN-10: 9780195331455, pp: 1004.

Marpanaji, E., B.R. Trilaksono, A.Z.R. Langi, A. Kurniawan and A. Mahendra *et al*., 2007. Experimental study of DQPSK modulation on SDR platform. J. ICT Res. Applic., 1: 84-98. DOI: 10.5614%2Fitbj.ict.2007.1.2.2

Mitola, J. III, 1992. Software radios-survey, critical evaluation and future directions. Proceedings of the National Telesystems Conference, May, 19-20, IEEE Xplore press, Washington, pp: 13/15-13/23. DOI: 10.1109/NTC.1992.267870

Mukesh, M., L. Abhishek and R.R. Bhambare, 2014. QPSK modulator and demodulator using FPGA for SDR. Int. J. Eng. Res. Applic., 4: 394-397.

Nyquist, H., 1924. Certain factors affecting telegraph speed. Bell Syst. Technical J., 3: 324-346. DOI: 10.1109/T-AIEE.1924.5060996

Oliveira, V.C.D., I.H. Yano, A.D.A. Mota and L.T.M. Mota, 2013. Feasibility of desktop virtualization per software services and local hardware based on the network throughput. J. Comput. Sci., 9: 827-837. DOI: 10.3844/jcssp.2013.827.837

Priya, A., N. Rajesh and R. Muthaiah, 2013. Carrier synchronization in software defined radio for 8PSK. Int. J. Eng. Technol., 5: 81-87.

Sailaja, C., D. Rajaramesh and M.S. Rao, 2014. Software Defined radio for underwater applications using FPGA. IOSR J. VLSI Signal Processing, 4: 01-04.

Song, W. and Q. Yao, 2010. Design and implement of QPSK modem based on FPGA. Proceedings of the 3rd IEsssEE International Conference on Computer Science and Information Technology, July 9-11, IEEE Xplore Press, Chengdu, pp: 599-601. DOI: 10.1109/ICCSIT.2010.5564000

Violas, M., J. Lourenço, A. Silva and A. Gameiro, 2014. FPGA implementation of an LTE based OFDM transceiver for 2×1 MIMO. Recent Res. Electrical Eng., 40: 105-110.

Wang, K., Z. Song, X. Qi, Q. Yan and Z. Tang, 2014. FPGA-based implementation of all-digital QPSK carrier recovery loop combining costas loop and maximum likelihood frequency estimator. Int. J. Reconfig. Comput., 2014: 1-15. DOI: 10.1155/2014/502942

Yano, I.H., V.C. de Oliveira, E.V. de Araujo, A.G. Campagnuci and B. Fabiano *et al*., 2014. Wireless sensor networks for measuring the consumption of save water taps. Am. J. Applied Sci., 11: 899-905. DOI: 10.3844/ajassp.2014.899.905

Yano, I.H., V.C.D. Oliveira, E.A.D.M. Fagotto, A.D.A. Mota and L.T.M. Mota, 2013. Predicting battery charge depletion in wireless sensor networks using received signal strength indicator. J. Comput. Sci., 9: 821-826. DOI: 10.3844/jcssp.2013.821.826

Zeeshan, M. and S.M. Khan, 2013. Robust sampling clock recovery algorithm wideband networking waveform of SDR. Int. J. Communicat. Netw. Inform. Security, 5: 10-17.

Divya, M., 2013. Bit error rate performance of BPSK modulation and OFDM-BPSK with RAYLEIGH multipath. Int. J. Eng. Adv. Technol., 2: 623-626.