A NUMERICAL MODEL FOR I/Q MODULATION/DEMODULATION FOR AN FPGA-ENABLED SDR PLATFORM

Bianca L.M. BOTA, Rareș BUTA, Botond S. KIREI, Calin A. FĂRCAȘ, Sorin A. HINTEA
Technical University of Cluj-Napoca, Cluj Napoca, Romania
Corresponding author: Bianca BOTA (e-mail: Bianca.Bota@bel.utcluj.ro

Abstract: This paper presents a numerical model for I/Q modulation/demodulation implemented in Matlab. This numerical model serves as a starting point for the HDL/FPGA implementation of a baseband processor for a previously developed SDR platform. The numerical model verification is carried out by performing 4, 8, 64, and 256-point QAM transmission over an additive White Gaussian Noise Channel and by comparing numerical and theoretical bit error rates. A partial HDL implementation of the baseband processor is given: the baseband processor generates a 1MHz tone signal that is mixed by the RF fronted, generating a double sideband modulated signal.

Keywords: *Matlab, SDR, IQ Modulator, BER, QAM*

I. INTRODUCTION

One of the fields that currently registers a significant development worldwide is the vast field of telecommunications. The possibility of communication without borders and access to the Internet, are fundamental elements for modern society, which is why companies and institutions invest in expanding infrastructure and telecommunications services to cover all areas. To keep pace with the fast development of novel wireless communication technologies, the use of rapid prototyping platforms, such as software-defined radios (SDRs) [1], is mandatory. A myriad of low-cost SDR platforms is now available on the market. Without the intention of completeness, some of these SDR platforms are: HackRF One [2], Adalm Pluto [3], and the USRP product line [4]. These platforms have in common that the baseband processing is carried out in a field programmable gate array (FPGA). Another common characteristic is the bandwidth, which is typically 20MHz.

In our previous work [5] we propose a RedPitaya [6] based large-bandwidth low-cost SDR architecture. The components of this architecture are chosen such that the transmit/receive chain bottleneck will be the Red Pitaya board. This SDR platform lacked a proper baseband processor. The modulated/demodulated signal must be generated/processed in a numerical environment, such as MATLAB. RedPitaya I/O functions were used to feed the RF front-end.

In this paper, we develop a numerical model for I/Q modulation/demodulation. This numerical model is a starting point for the baseband processor implementation in HDL/FPGA. The paper is organized as follows. In Section II. our previous work is presented, describing the proposed SDR platform. Section III. describes the numerical model developed in Matlab. The numerical model validation is carried out by performing 4, 8, 64, and 256-point QAM transmission over an additive White Gaussian Noise (AWGN) Channel and by comparing numerical and theoretical bit error (BER) rates. Section IV. presents the partial implementation of the baseband processor: it generates a tone, that is mixed by the RF fronted up to 800MHz.

II. PREVIOUS WORK. THE SDR PLATFORM

The proposed SDR platform is presented in *Figure 1*, which can be seen as a Zero-IF transceiver. The Red Pitaya board comprises the baseband processor in the FPGA and the Analog/Digital conversion stages. The RF output bandwidth of the RED Pitaya is approximately 40Mhz, but the onboard ADC support 125Msps. The I/Q modulator stage is carried out with a demo board 2391A [7] optimized for evaluating the LTC5589 low power I/Q modulator up to 4.5GHz. The device is configurable through SPI interface allowing the user to set the gain, optimum center frequency, I-offset, Q-offset and sideband suppression. Besides the 1dB step digital gain control, there is an analog gain control.

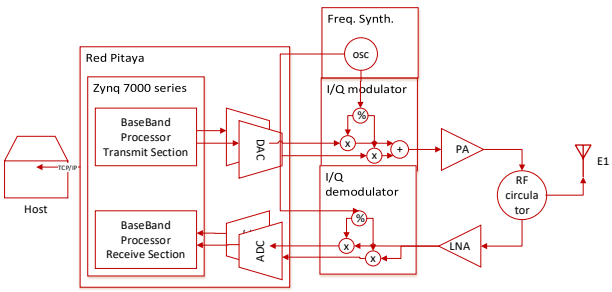


Figure 1. High-bandwidth low cost SDR platform [5]

The power amplifier (PA) stage could be carried out with a low-cost TRF37x75 EVM [8] dev-board. The TRF37x73 RF Gain Block Amplifiers are internally matched to 50Ω and provide a flat gain response and excellent OIP3 up to 6GHz. These parameters concord with the mixing stage output impedance and frequency range.

Manuscript received March 1, 2023; revised June 21, 2023

The modulator stage is also fed by the carrier signal, generated by a low-cost MAX2870 [9] 23.5-6000MHz RF Signal Generator development board. The board can maintain excellent phase noise and spurious indicators and provides dual-channel differential output drivers, which can be individually set to provide output power from -4dBm to +5dBm.

The demodulator stage is carried out by an LTC5586 demo board [10]. Circuit 2349A's two switchable single-ended RF inputs are optimized for 700MHz to 6GHz. Its single-ended LO input is programmable from 300MHz to 6GHz operation via SPI bus.

The receive side baseband processor of the transceiver is solved with the RedPitaya board and its low-RF input section. The nominal bandwidth of the ADC is 50 MHz, but the sampling rate is 125 Mbps. It is worth mentioning that the baseband processing could be carried out using the Eclypse Z7 dev board bundled with a Zmod AWG 1411 and Zmod Scope 1410-105 [11].

III. THE NUMERICAL MODEL FOR I/Q MODULATOR/DEMODULATOR

An I/Q modulator is a device that converts baseband information into low-RF signals. [12] As can be seen in *Figure 2*, the IQ modulator is based on two mixers whose outputs are combined.

In RF applications, quadrature signals or so-called I/Q signals are most often used, these two signals being the basis of complex RF signal modulation and demodulation. Two periodic signals are known to be in "quadrature" when their phase differs by 90 degrees. Thus, the most well-known quadrature signals are the sine and cosine signal.

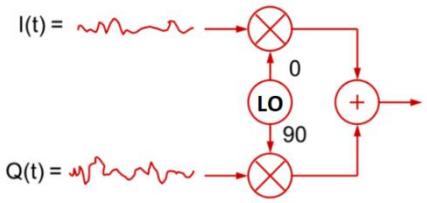


Figure 2. IQ Modulator architecture.

The two modulating signals, called in-phase (I) and quadrature (Q) signals, are mixed with a local oscillator (LO) with the "Q" component out of phase. The I/Q signals are then summed up to form a combined output signal. Summing the I/Q signals results in the desired amplitude and phase of the combined output signal.

The Matlab platform was used to implement the architecture of the numerical I/Q modulator presented in *Figure 3*. The implementation itself, as can be seen in *Figure 4a and Figure 4b*. consists of modeling four Matlab scripts, namely: *plotBER.m*, *rcos.m*, *berawgn.m*, and *transmission.m*.

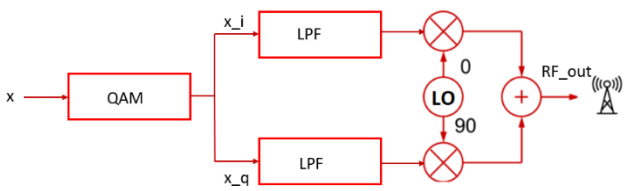
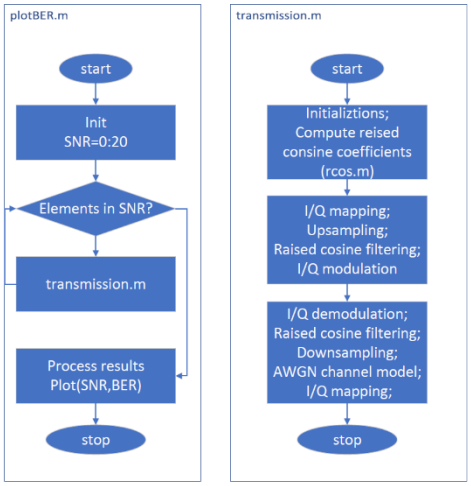


Figure 3. Numerical IQ Modulator architecture

The Matlab script *plotBER.m* plots the theoretical BER and practical BER for a constellation of M points. The flowchart of the script is presented in Figure 4.a. The constellation size M can be changed by the user, with the values 4, 16, 64, and 256 representing the typical constellation order values.



(a) (b)

Figure 4a, 4b. Flowcharts of the main Matlab scripts.

Repeated transmissions are made to calculate the practical BER, and the erroneous bits are counted, after which a statistic is made at the end of the communications. The transmission itself is modeled within the *transmission.m* script. The filter used is a pulse-shaping filter, namely the Root Raised Cosine filter. The coefficients of this filter are calculated using the Matlab *rcos.m* script. To obtain the ideal BER or the theoretical BER, consider the script *berawgn.m*, script that handles the *berawgn* Matlab function. This function returns the BER and symbol error rate (SER) in an AWGN channel for unencoded data using different modulation schemes, such as QAM-type modulation.

To implement the *transmission.m* script that models the bit transmission itself, the first step is to generate a sequence of uniformly distributed pseudo-random integers, then increase the sampling rate by an integer factor. To generate the pseudo-random sequence, Matlab's *randi* function is used, which generates a sequence of integers between 1 and M, where M represents the order of the modulation, in the present case, M is chosen 64. The obtained sequence is to be modulated in quadrature, using QAM with the modulation order M = 64, the resulting signal in the modulation trace being called the QAM modulated signal with modulation order 64. Given the fact that we are working with a QAM modulation, the obtained modulated signal has two parts, namely a real part or an in-phase branch I and an imaginary part corresponding to the quadrature branch Q. Once the IQ signals are obtained, according to the diagram in *Figure 3.*, the next step is their filtering. This is necessary to limit the band. A Square Root Cosine filter is used, as it can avoid inter-symbol interference.

So, the global filtering of the QAM signals, necessary for limiting the band of the modulated signal, is performed with a high cosine RC characteristic that ensures zero ISI at the sampling times. This characteristic is equally

distributed between transmission and reception for optimal behavior in the presence of noise. As mentioned before, the coefficients of this filter are calculated within the *rcos.m* script. Once the two filtered signals are obtained for the I and Q branches, they will be mixed. To obtain the mixing, the two signals must be multiplied as follows: the signal corresponding to I branch is multiplied by the sinusoidal signal LO, and the signal corresponding to the Q branch by a sinusoidal signal out of phase by 90 degrees, i.e., a co-sinusoidal signal. The two signals corresponding to the local oscillator, the sinusoidal signal and the co-sinusoidal signal, were generated with a frequency of 15 MHz.

Once the two signals corresponding to I/Q branches are mixed with the LO signals, they will be passed through an adder. The modulated signal obtained at the output of the adder can be written to a *.coe* file, which file can then be loaded into the ROM memory of the FPGA. Currently, the ROM memory serves as an interface between the user and the onboard ADC convertors, which are fetching the signal samples from the ROM. It is for seen to replace the ROM memory block with RAM, thus a baseband processor can place signal samples to be fetched by the ADCs.

To verify the correctness of the implementation of the IQ modulator, the reverse operation of the demodulation was implemented, namely the IQ demodulator; the block diagram of the demodulator circuit can be seen in *Fig. 5*.

The modulated signal previously obtained from the output of the adder is considered in this case as the input signal or received signal for the demodulator circuit. To obtain the two signals corresponding to the in-phase and quadrature branches, the received signal is multiplied by the corresponding LO carrier signal. For branch I, in phase, the received signal is multiplied by the sinusoidal signal, and for the quadrature branch it is multiplied by its corresponding signal (the co-sinusoidal signal). The next step is filtering each branch, and the QAM demodulation.

Figure 5. Numerical IQ Demodulator architecture

After the execution of the *transmission.m* script, we can obtain a measure for the theoretical BER and the practical BER, considering the signal *x* transmitted and the signal resulting from the *x_receive* demodulation, the representations of the two BER signals being possible after running the *plotBER.m* script.

Using the Matlab *biterr*() function, the unsigned binary representation of the elements of the transmitted *x* signal is compared with the elements of the *x_receive* signal obtained after demodulation, the function returning the number of bits that differ, thus the total number of errors. Knowing the total number of errors as well as the total number of symbols transmitted, the practical BER is therefore calculated. To calculate the theoretical or ideal BER, the *berawgn* function is used. The *berawgn* function BER and SER in an AWGN channel for unencoded data using several modulation schemes, in this case, the modulation scheme QAM. The first input argument, *EbNo*, is the ratio of bit energy to noise power spectral density in dB (*Eb/N0*). The values in the BER and SER output vectors correspond to the theoretical error rate at the specified *Eb/N0* levels for a gray-coded signal constellation. In the figures below, Fig. 6. and Fig. 7., it can be seen two BER representations:

Considering the two representations – *Fig. 6.* and *Fig. 7.* where on the *y*-axis we can see BER and on *x*-axis we can see SNR - for modulation of order M = 4, respectively 64. It can be observed that in the case of modulation of small order, M = 4, the bit error rate is low, and there are no more erroneous bits in the simulation. We should run more simulation time to have data for the rest of the figure.

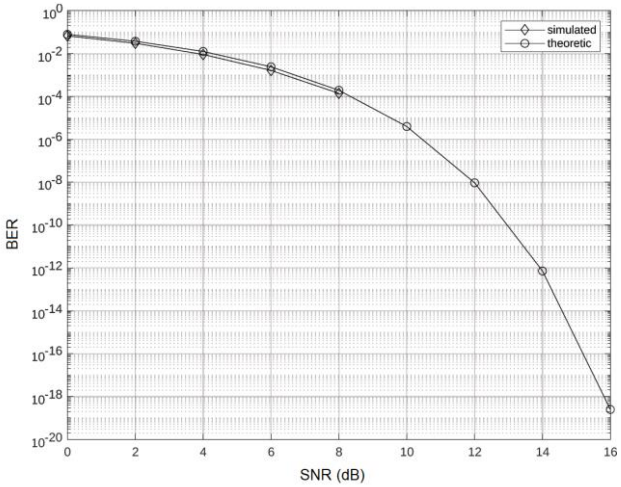


Figure 6. BER representation for M = 4

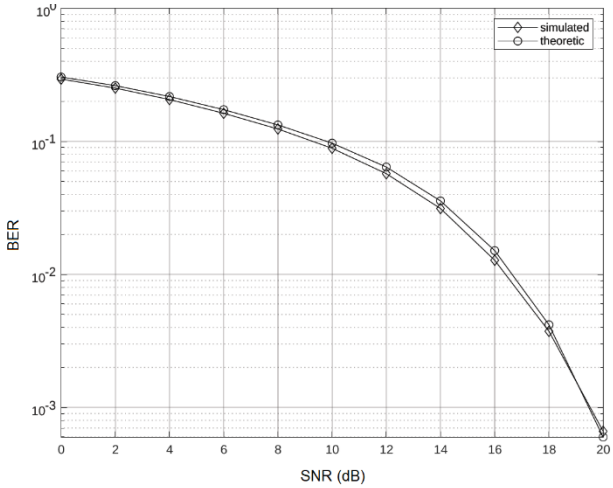


Figure 7. BER representation for M = 64

IV. EXPERIMENTAL RESULTS

This section presents the generation of a tone signal of 1 MHz that is mixed in the RF fronted and upconverted to 800MHz.

The Direct Digital Synthesis (DDS) compiler provided by the Vivado suite is used to generate the tone signal. First, a block design is created that contains a Zynq-7000 processing system, together with the DSS compiler. To configure a clock circuit according to user requirements, the LogiCORE™ IP Clocking Wizard is used. In this project, the IP is configured to generate a system clock of 100 MHz.

Next, as seen in *Figure 8*, two DDS systems were used: one used to generate the sinusoidal signal and one for

generating its phase. To generate the sinusoidal signal, *dds_compiler_0* is configured as a sinusoidal generator, and its phase generation is done by the *dds_compiler_1* block. To generate a 1 MHz carrier, the phase increment is:

$$\theta = \frac{f_{out}\,2^{B}}{f_{clk}} = \frac{1MHz \cdot 2^{16}}{100MHz} = 65536 \qquad (1)$$

where θ represents the phase increment, f_{out} is the desired output frequency, f_{clk} is system clock frequency, B is the working base of the DDS.

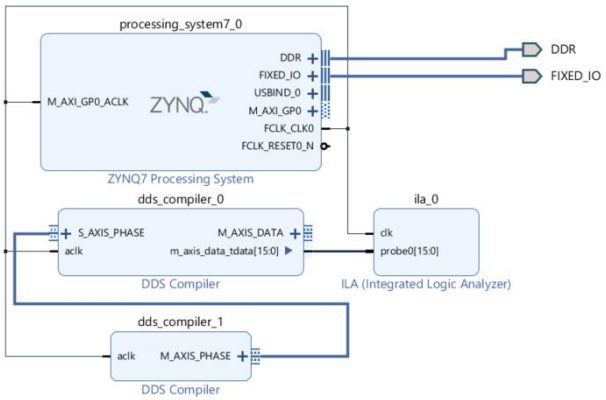
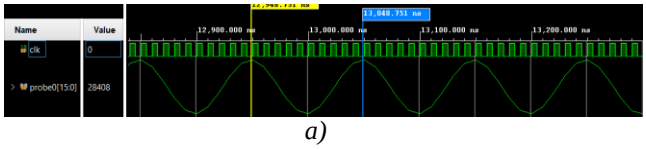


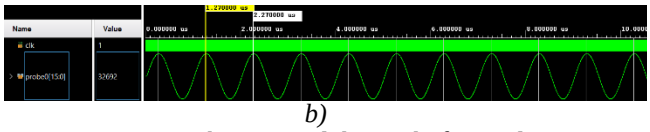
Figure 8. Block diagram of the system generating the tone signal (Xilinx Vivado snapshot)

With the help of the Integrated Logic Analyzer (ILA) block, the waveform obtained by using the DDS Compiler can be visualized. After running the simulation, one can see the generated signal, which has the desired frequency, i.e., 1, respectively 10 MHz. In *Figure 9a* and *9b*, with the help of cursors, the period of the tone signal (1 us, respectively 100 ns) was measured and highlighted.

Since the goal is to display and interpret the sinusoid using the ADALM PLUTO tool, the next step is to convert the digital signal to an analog signal. This is done with the help of a Digital-to-Analog Converter, DAC. The Zmod DAC 1411 Low-Level Controller is intended for use as a stand-alone IP in projects that directly interface with the Zmod DAC. The controller initializes the hardware and multiplexing two input channels onto a single channel with Double Data Rate (DDR). The IP is designed to generate two 100 MHz clock signals for the Zmod DAC 1411, one used to qualify input data, used as a sample clock, and to generate output data at 200MSPS 14-bit. The two SDR input data channels of the IP core are synchronous to the system clock input (100MHz) that the IP core requires.



a)



b)

Figure 9. The sinusoidal signal of period is a) 100 ns and b) 1 us

The partial baseband processor was tested using the setup presented in *Figure 10*. The baseband processor generating a tone signal was deployed in a Red Pitaya development board. An Adalm Pluto device is used for generating a local oscillation and to inspect the RF signal generated by the transmitter chain.

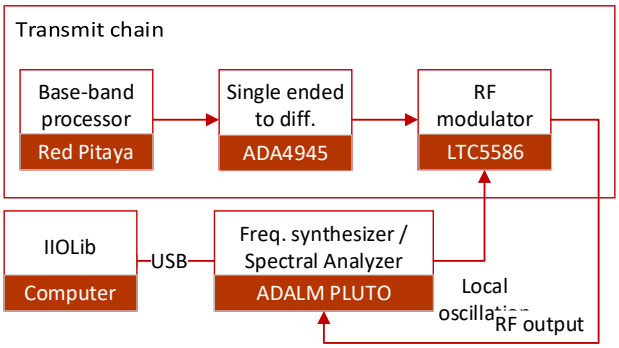


Figure 10. Setup for transmit chain testing

A double sideband (DSB) modulation was carried out to test the baseband processor: a local oscillation of 800 MHz is generated by the Adalm Pluto [13]. The baseband processor generates a 1 MHz tone. As the carrier and the tone signals are mixed, the resulting DSB-modulated signal is presented in *Figure 11*. The center frequency of the spectrum analyzer is set to 800 MHz. A carrier of 10 dBm can be clearly seen at 805 Mhz. Due to the 1 MHz modulating signal, two peaks at 804 and 806 MHz are present.

V. CONCLUSIONS

In this paper, a numerical model for an SDR baseband processor was developed in MATLAB. The model consists of an up-sampling operation on the transmit chain, raised cosine filtering, I/Q modulation, and an adder stage. The receive chain comprises the reciprocal operations carried out in transmission. The numerical model was validated using repeated QAM transmissions and comparing the results with the theoretical ones. The model served as a starting point for a hardware implementation of the baseband processor. It was implemented partially (the transmit chain was deployed in an Eclypse Z7 dev board), and a single sideband modulation was achieved using a previously developed low-cost high bandwidth SDR.

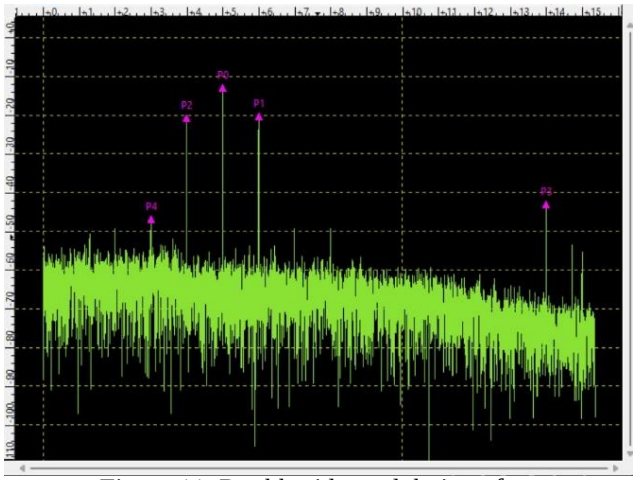


Figure 11. Double side modulation of an 1Mhz sine wave

REFERENCES

[1] Travis F. Collins, Robin Getz, Di Pu, Alexander M. Wyglinski, *Software-Defined Radio for Engineers*, Artech House, ISBN-13: 978-1-63081-457-1
[2] (14.01.2023), Great Scott Gadgets website, https://greatscottgadgets.com/hackrf/
[3] (14.01.2023), Analog Devices website, https://ro.mouser.com/datasheet/2/609/ADALM-PLUTO-Product-Highlight-1633770.pdf
[4] (14.01.2023), National Instruments website, https://www.ni.com/pdf/product-flyers/usrp-software-defined-radio.pdf
[5] (18.01.2023), Red Pitaya based low-cost SDR platform, https://go.redpitaya.com/hubfs/Red%20Pitaya%20based%20low-cost%20SDR%20platform%20.pdf
[6] (14.01.2023), Welcome to the Red Pitaya documentation, https://redpitaya.readthedocs.io/en/latest/index.html
[7] Linear Technology Corporation, DEMO MANUAL DC2391A - 700MHz to 6000MHz LOW POWER I/Q MODULATOR, 1630 McCarthy Blvd., Milpitas, CA 95035-7417
[8] Texas Instruments, "TRF37x73 and TRF37x75 EVMSLAU546A User's Guide", March 2014, Revised October 2021
[9] Maxim Integrated, "23.5MHz to 6000MHz Fractional/Integer-N Synthesizer/VCO", 19-6250; Rev 6; 4/17
[10] Linear Technology Corporation, "DEMO MANUAL DC2349A - LTC5586 6GHz High Linearity I/Q Demodulator with Wideband IF Amplifier", 1630 McCarthy Blvd., Milpitas, CA 95035-7417
[11] 14.01.2023, Eclypse Z7 Hardware Reference Manual - Digilent Reference, https://digilent.com/reference/programmable-logic/eclypse-z7/reference-manual
[12] Bruce Hemp, Sunny Hsiao, "*Low Power IQ Modulator for Digital Communications*", LT Journal of Analog Innovation, Vol. 25, No. 3, November 2015.
[13] Analog Devices Inc, ADALM-PLUTO SDR - Active Learning Module, Otl-Aicher-Str. 60-64, 80807 München, Germany, 2017, BR15418-.1-2/17