The Test Bench for FPGA-based QPSK and QAM Modulators in Software defined radio

O. Venkata Krishna[1]and B. Janardhana Rao[2]
[1] Department of EIE, CVR College of Engineering, Hyderabad, India
Email: venkatakrishna.odugu@gmail.com
[2] Department of ECE, CVR College of Engineering, Hyderabad, India
Email: janardhan.bitra@gmail.com

***Abstract*—In todays fast evolving mobile communications the requirements of higher data rates are continuously increasing, pushing operators to upgrade the back haul to support these speeds. A cost effective way of doing this is by using microwave links between base stations, but as the requirements of data rates increase, the capacity of the microwave links must be increased. The objective of the paper is the developing the next generation high speed microwave links for the E-band. In the research project there was a need for a testing system that was able to generate a series of test signals with selectable QAM modulations and adjustable properties to be able to measure and evaluate hardware within the project. The developed system was designed in a digital domain using an FPGA platform from Altera, and had the ability of selecting several types of modulations and changing the properties of the output signals as requested. By using simulation in several steps and measurements of the complete system the functionality was verified. The developed system can be used to test several different modulators in other projects as well and is easily extended to provide further properties.**

***Index Terms*—FPGA, Signal generation, Quadrature modulation, QPSK, Software defined radio, Testbench, Radio communication, High datarate.**

I. INTRODUCTION

As the demand from customers of Internet capacity within mobile networks increase, the operators need to improve the capacity of the entire network continuously. To connect sub networks and antennas to the backbone of the network, several linking methods exist to create a backhaul. The paper aims at developing a microwave link with increased data capacity to be used in a back haul. The application of the paper is to demonstrate a 10 Gbps radio system over E-band. The following text is cited from the project specification [1] and provides a general idea of the purpose of the application. A general knowledge in radio communications will ease the understanding. This paper work is aiming at solutions for spectrum efficient radio communication at the E-band 71-76, and 81-86 GHz, and the 120 GHz band. The E-band is now gaining an increased interest for telecom operators providing internet access over the mobile network where radio links are used for the backhaul. The driving force is the demand from customers to have mobile access to internet for streaming video etc. The E-band provide 2*5 GHz bandwidth, commercial radio links for the E-band often use the modulation format OOK, which is simple but not spectrum efficient. The goal of this work is to investigate solutions for spectrum efficient use of the E-band and higher frequencies like 120 and 220 GHz-band, and as a result increase the bit rate at least by a factor of four utilizing the same bandwidth. In this paper, several solutions will be implemented and tested. The first solution is based on D-QPSK modulation, which uses a precoder realized in FPGA. This solution is more complex than the OOK-based system but is more spectrum efficient, by a factor of two. Like the OOK, the D-QPSK is non-coherent i.e. the phase of the carrier is not needed to be retrieved. For higher spectrum efficiency, more complex modulation is required like QAM. In a QAM signal, the data is represented by both amplitude and phase of the modulated carrier, the carrier phase reference have to be retrieved in the receiver, and both the amplitude and phase have to be detected. Therefore D/A converters are needed in the modulator of the transmitter and A/D converters in the receiver. Some of the simpler modulation formats will be realized in hardware while QAM will be realized using software in FPGA's.

The relation between application and this paper work provides an understanding of the purpose. The project purpose is to investigate several different options and types of modulators and modulations, and the efficiency of these. To provide a flexible way of testing the transmission and modulation properties of products within the project, a proposal of developing a test bench
as a paper was laid out. This paper is the result of that proposal. The focus within paper was on implementation of a system to be used for testing products within the application. The system that was implemented provides a way of measuring properties of different options that are selectable in the project to be guidance in crossroads with focus on modulation type. Outside of the scope of the paper, the test bench will also be usable for other components in the project but this will not be considered during implementation due to time and resource limitations. Within the scope of the paper, all work that was done was based on information of existing technology although the actual implementation was very specific for the application. The existing technology includes the types of modulation and techniques for generation of test sequences.

A) Software defined radio

The general idea of constructing communication components in software by using FPGAs or MCUs has increased in popularity as prices decrease and performance

increase. The technology is referred to as SDR, Software Defined Radio. This technology provides a way of changing properties of a radio transmitter or receiver within reconfigurable blocks, such as modulation, frequencies, amplitudes and algorithms amongst others. SDR technology opens up new ways of adaptive communication and a higher spectral efficiency according to Tuttlebee [2]. Since the parameters can be controlled by software, a higher performance and greater spectral efficiency can be achieved since the communication channel can be adjusted dynamically depending on the environment, and properties such as modulation type can be adjusted to minimize the number of errors while keeping the data rate at a maximum. The interest in SDR continuously increases, but some problems exist that still require parts of the radio system to be in hardware and this is due to the limited performance of A/D and D/A converters and requirement of faster performance of the SDR [2], especially as frequencies increase. In the paper, a SDR block was used as a main component in the test bench that was developed.

B) QAM MODULATION

A commonly used modulation scheme is QAM, Quadrature amplitude modulation. According to Ergen [4], this is one of the most widely used modulation methods and is used in most of todays digital communications. As the name suggests, it is an implementation of amplitude modulation. By using two separate carriers phase shifted to each other the two carriers are amplitude modulated separately by different data, and then combined into a single carrier by simple addition. The two carriers are named I and Q, in-phase and quadrature, and are usually a cosine (I) and sine (Q) wave due to their 90^{o} phase shifted relation to each other. The finite number of amplitude steps that the I and Q can be adjusted to implements different types of QAM modulations. By having two selectable amplitudes for each carrier, a constellation diagram such as the leftmost in Figure 1 is obtained and as seen, where each point in the constellation diagram represents two bits of data. This comes from each of the two carriers having two selectable amplitude levels and can thus represent one bit each, and together two bits. If the number of amplitude steps are increased to four the data represented by each point is four bits. By increasing the number of selectable amplitude steps, and thus the number of constellation points, more data can be represented by each point and thus a higher transmission rate can be achieved and the loss of SNR. The number of constellation points dictates the modulation type, described as M-QAM where M is the number of points in the constellation.

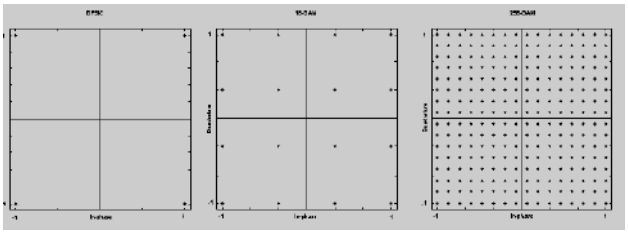

Figure 1: QPSK, 16-QAM and 256-QAM constellation diagrams

There are several different implementations of QAM modulation such as differential, circular and rectangular. In this paper only rectangular QAM was implemented. When transmitting a QAM signal as a radio wave, a modulator is used to apply the different amplitude levels on a carrier wave and thus modulating it. The method of directly applying amplitude levels on an input carrier wave is called direct conversion and due to the cost-effective implementation of this it is a common choice in today's digital communication systems. However, according to Schenk and Linnartz in RF imperfections in high-rate wireless systems [5].

II. IMPLEMENTATION

The main focus of the paper was in the implementation phase since the main objective of the paper was to produce a hardware unit usable in the MODEM project.To reach the goal of constructing a hardware unit with flexible attributes, the platform to be used in the project was chosen as an Field Programmable Gate Array (FPGA). This hardware component consists of logic elements with programmable paths in between, providing the possibility to create different hardware behavior by programming the FPGA. The programming is a result of code written in a hardware description language. In this paper System Verilog was used.The code used to describe the desired hardware behavior in an FPGA is written in a language type known as hardware description language. SystemVerilog, used in this project, is an extension to the widely used Verilog, providing additional data types and more flexible code construction than the original.

A) Development environment

The choice of environment when developing for FPGAs is tightly bound to the hardware used, since each hardware manufacturer provides the environment for their own product. In this paper, the hardware was supplied from Altera and thus the most convenient development environment was Quartus II which has features especially useful for development on Altera hardware. Several advanced hardware units can easily be generated by the program, providing means of utilizing several advanced hardware features such as Phase-Locked Loops (PLL) and Fast Fourier Transforms (FFT).

To get a better understanding of the algorithms and methods to be implemented, simulations in Mathworks Matlab were performed prior to each implementation task. By breaking down and dividing complex operations into smaller steps, a better understanding of how an implementation could be done was gained. As an example, dividing the cross correlation operation into basic mathematical operations such as multiplication and addition provided a way of understanding the needed components. The smaller steps were then combined to perform the original task, and compared with built-in functions of Matlab for verification. A major part of the development was simulation of the written code, as

DOI:10.32377/cvrjst0715

examining the programmed FPGA simply would not provide enough information of internal states and is also a very cumbersome task due to compilation and programming times. Since Quartus II has a good integration with Modelsim, also provided by Altera, the choice of simulation software was obvious. With the help of Modelsim, the expected behavior of the written hardware descriptive code could be examined by simulation.

As an FPGA is only a chip, more hardware was necessary to utilize the chip. To load the compiled code into the FPGA, a programmer was needed and connections to the FPGA had to be established. Also input and output possibilities were required to make use of the system along with power supplies, reference clocks, buses and other peripherals. There were several boards equipped with these units, along with an FPGA, to make development easier.

To observe the analog output from the conversion board, an Agilent Infiniium 54854A oscilloscope with X/Y-plot was used. This provided the ability to observe the constellation points produced in the analog domain, which is explained more in detail in the following sections.

B) Project work flow

The different tools specified all have their part in the work flow, and to illustrate the dependencies of each other and to visualize the work flow Figure 2.illustrates when the different tools were used. As the work flow describes, a hardware design was the starting point of the work. As soon as a design had been made, it was implemented in Matlab and Quartus II in two separate trails. The Matlab implementation resulted in immediate simulation and could be used find errors and provide a better understanding of the design early in the work flow. The Verilog code implemented in Quartus II resulted in HDL code that could be simulated using Modelsim, and by comparison with the Matlab simulation a first verification could be made before going deeper into implementation and synthesis of actual hardware.

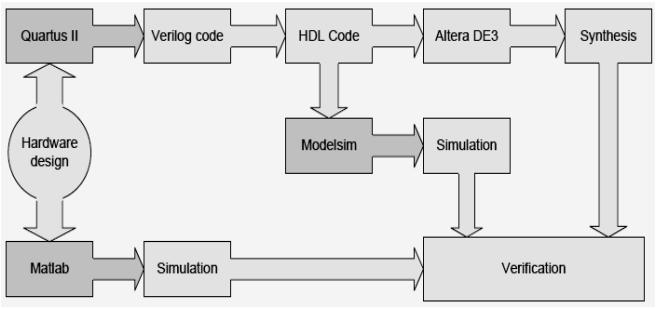

Figure 2: Project workflow.

When the Matlab simulation, Modelsim simulation and hardware synthesis all are completed, these can together perform a very genuine and reliable verification of the implemented hardware design.

C) Pseudo Random Bit Sequence (PRBS)generator and detector

The first hardware unit to implement was a Pseudo Random Bit Sequence generator and a detector for such a sequence. A PRBS can be seen as a sequence of bits with a specific length where the bits appear to be random, and the purpose of a PRBS generator is to produce such a sequence. Due to the random appearance, two rotated copies of the same PRBS does not have many bits in common except for when the two sequences match up exactly. This is useful when examining the input and output relation of a unit under test which is described more in detail later. The goal of this first step was to produce a PRBS generator along with a detector to be used in test transmissions.

The goal of a PRBS generator is to produce a sequence of a determined number of bits by using a generator polynomial implementation. There are several polynomials that can be used, and these polynomials generates sequences with the length 2^{n-1} where n is the degree of the polynomial. In this project, a decision was made that the length of 127 would be sufficient, thus the degree would be n=7. To implement a generator polynomial in hardware, n registers are used in combination with XNOR gates to produce a linear feedback shift register (LFSR).When put together, the degree of the polynomial decides the length of the bit sequence and the number of registers to be used. The polynomial provides information on where to place XNOR gates on the registers to create taps that implement a LFSR, whose output will be a PRBS. According to application note XAPP 052 [6, page 5] using XNOR from registers 6 and 7 will implement a maximum length LFSR. This implementation is derived from the generator polynomial:

$$y = x7 + x6 + 1$$

The hardware implementation with the XNOR taps is described by the circuit diagram in Figure 3.

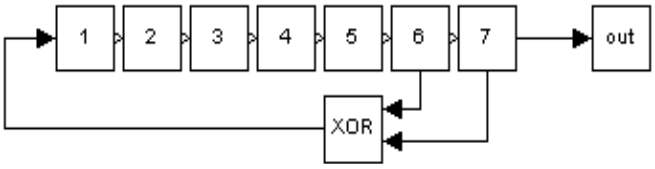

Figure 3: Linear feedback shift register

To utilize the PRBS generator in a test transmission, a tool for detecting the reception of the signal was necessary and this is where the PRBS detector was used. The purpose of the detector was to find a PRBS within a chosen bit stream and provide a way of measuring the integrity of the found sequence. A reference signal to be located in the bit stream was needed during the detection and therefore the detector had a PRBS generator built in with the same properties as of the transmitting generators. As cross correlation is an effective method for comparison of signals and thus detection, the method was used in this implementation. To perform the Fourier transform, a fast Fourier transform (FFT, which is a more efficient way of

calculating the transform) was used both for the transform and the inverse transform. The development environment Quartus II supplied tools.

The main task of the project was to produce a baseband signal in the FPGA and, by using the attached conversion board, convert these digital signals into an analog set of signals to be fed into a modulator for testing. The signals to be fed to the modulator was the In-phase signal and the Quadrature signal. The QPSK modulator was one of the first components to be implemented and therefore the signal design was revised when the unit was used in later stages. The main difference between this implementation and the one used later is the usage of an IF wave. This modulator was constructed to output I and Q signals as a 75 MHz modulated wave.

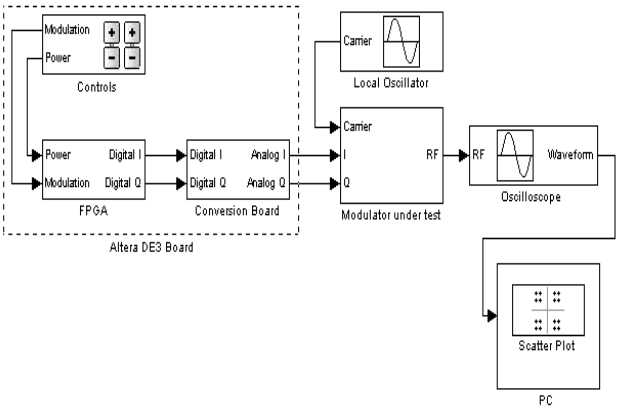

Figure 4: Measurement system overview

III. MEASUREMENT SYSTEM

The purpose of the measurement system was to provide a hardware unit with selectable modulations and signal properties to provide a way of measuring the performance of different modulations and the I/Q imbalance of these in a modulator. This was to be measured and be of assistance when deciding the modulation type to focus on in the research project. The different modulation units that were implemented previously in the thesis were used as building blocks for this system, but they were all modified to suit the purpose better. It is important to note that the hardware target to be measured upon is a modulator constructed within the MODEM-project, and internally inside the FPGA several modulators are implemented to produce the test signals for the hardware modulator.

These are two separate items which are not to be confused with each other although they are referred to by the same name. Since not all modulations were equally interesting from the research project point of view, a decision was made to only implement QPSK and 256-QAM. Along with this, an ability to adjust the voltage of the I and Q signals individually was also needed. An overview of the system and the environment is pictured in Figure 4.All of these functionalities had to be selectable and adjustable while the system was running, and the entire system also needed to be independent in a way that no computer would be needed during testing. Since an FPGA is a volatile device2 the system had to be reconfigured at each power-up. To avoid the need of a computer to do this, an on board ash memory served as the source for programming the FPGA upon startup.

The measurement system was constructed by using several building blocks previously written, along with a power control unit and a control system. The following sections describe the inner workings of these.

The system was built to test direct modulators and thus there was no need for an IF signal to be produced since only the amplitude levels of the I and Q signals were needed. The direct modulators internally produces a phase shift to be used on the Q signal from an input IF signal. Because of this, the VCO that resided in each of the modulators could be removed and instead the multiplication that previously was made with a sine or cosine wave could now be performed by multiplication with a constant factor. This constant was selected differently in each of the modulators to utilize the full range of the D/A converter and are described in table 2.4. The difference between the modulations were the number
of amplitude steps within the same output range. Since the signals from all different modulators later were to be adjusted in voltage, the conversion into unsigned signals had to be moved outside of the modulators and into the power control unit. Therefore the output was changed
from 14 bits unsigned info 14 bits signed within each of the modulators.

To avoid several clock domains, the PLL that each modulator was equipped with internally was also removed and instead the clock signal was fed into the modulator from a PLL that was global for the entire measurement system. The clocks for the D/A conversion board was also moved in the same way, reducing the complexity of each modulator unit.

A requested functionality was to be able to adjust the output voltage of the I and Q signals individually while the system was running. Since the available signals utilize the full range of the D/A conversion board, the voltage of the output signals could not be increased but only decreased, and this was performed by a power control unit that would attenuate the signals within a certain range. Since the signals should be scaled proportionally on each amplitude levels, a division with a voltage level variable had to be done. The desired division variable would be in the range of 0.5-1.0, but since division with decimal point numbers is a rather complex operation for an FPGA this was implemented using a different approach.By performing a multiplication with 214=x where x is the desired division variable and taking the high 14 bits of the 28 bit result, a much more effective division has been made since

 DOI:10.32377/cvrjst0715

multiplication between two numbers is a much simpler task than division for an FPGA. The maximum output from the D/A Conversion board was around 500 mV as measured previously, and without any attenuation the generated signals can at some stage reach this level. Eight stages of voltage levels were implemented since this was easily represented by the eight LEDs on the Altera DE3 board, and an estimate of the attenuation levels of these can be seen in table 2.5. As seen in table 2.5, the steps of maximum output amplitude are adjusted in steps of 15 mV. Another functionality of this unit was to convert the resulting signed 14 bit signals into unsigned 14 bits since this unit was removed from the modulators and the D/A Conversion board does not handle signed signals. This was simply done by adding 214=2 to the resulting signal, changing the range from -8192 to 8192 into 0 to 16383.

As this system was to be used stand-alone, the need for a computer had to be eliminated. The programmed FPGA looses all information once it is turned off and therefore the program had to be stored on a ash memory on the Altera DE3 which could reprogram the device every time it was powered up. To control the system and get information about applied settings, the on-board switches, pushbuttons, LEDs and HEX display was used. Using the switches, different modulations could be selected and displayed on the HEX display, see table 2.6. Pushbuttons were used to adjust the output voltage in eight stages, and to display the current setting the on-board LEDs were used. These were RGB LEDs therefore it was natural to use two separate colors for the different channels.

IV. RESULTS

All of the implemented systems were measured and verified. The results of these along with comments and discussions are provided in this section.Each of the hardware units implemented required different types of measurements for verification and thus different setup.

A) PRBS generator and detector

Since the PRBS units did not provide any output to be measured upon, all measurements were made by observing the internal states of the FPGA and comparing the output with simulations from Modelsim and Matlab. These produced enough information to prove the functionality of the system. To verify the design, observations of the internal states and registers were needed and for this task, a built-in program in Quartus II known as SignalTap was used. This program had the ability to observe a selected number of registers on the actual FPGA during runtime and download them to the development environment for verification. This method was used to observe the value of the cross correlation during runtime when correct or distorted PRBS were sent to the detector.

B) Modulator

All modulator implementations were tested during development to verify the functionality and signal behavior. The output waveforms were observed by using An oscilloscope and measured upon to determine the maximum output voltages. The first implementation, the QPSK modulator, was also compared with a Matlab simulation during development to provide knowledge on how the output waveforms should appear if fully functional. The following implementations did not include this step as the expected waveform simply had an increase in the number of attainable amplitude levels as seen in the oscilloscope measurements. The only reason the Matlab simulation was used was to determine that the mathematical operation of the modulator matched the hardware, and the same operation was used in all later implementations. Since each of the modulator implementations was a complete hardware block, the Altera DE3 had to be reprogrammed between each of the verifications. The waveforms were obtained using a 1 Gsps sample rate in the oscilloscope to properly sample the waveforms. The usage of SignalTap software was not possible due to hardware limitations when using the conversion board, therefore observations using an oscilloscope was the method of verification.

To obtain an eye diagram, the oscilloscope was set to keep the sampled waveforms for 500 ms while sampling new waveforms. This results in several samples being presented on the screen on top of each other, providing a possibility to see the amplitude levels at the same time in the shape of an eye.

i) QPSK

Since this was the first modulator to be implemented, initial simulations with Matlab and Modelsim was done to perform a verification of the output waveform correctness. As seen in Figure.5 the two different simulation results produce the same output, although the actual phase shift occurs slightly later in the Matlab simulation. the eye diagram of the qpsk is shown in Figure.6 .

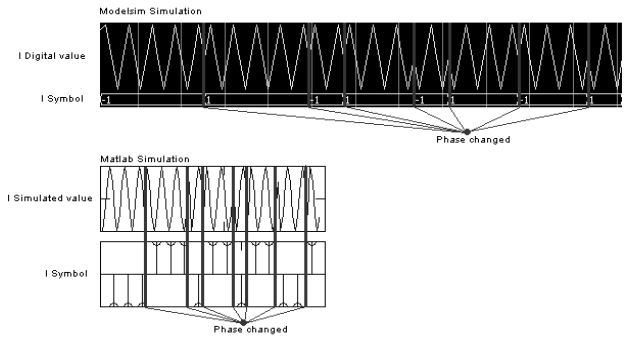

Figure 5: QPSK simulations in Modelsim and Matlab

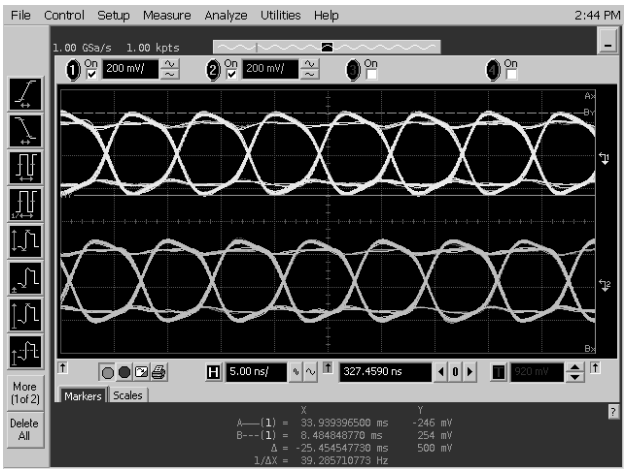

Figure 6: QPSK eye diagram

ii) 256-QAM

Increasing the number of amplitude levels to 32, this modulation puts high demands on the conversion board being able to provide a good enough resolution between the levels. Observing Figure.7 it is hard to tell how well the resolution is due to the actual resolution of the screen on the oscilloscope The utilization of the range on the conversion board was almost used at maximum in this modulator, using 15 as the maximum multiplication value and having a resolution of 10 bits on the sinus and cosine waves the expected maximum output voltage was 468.75mV.

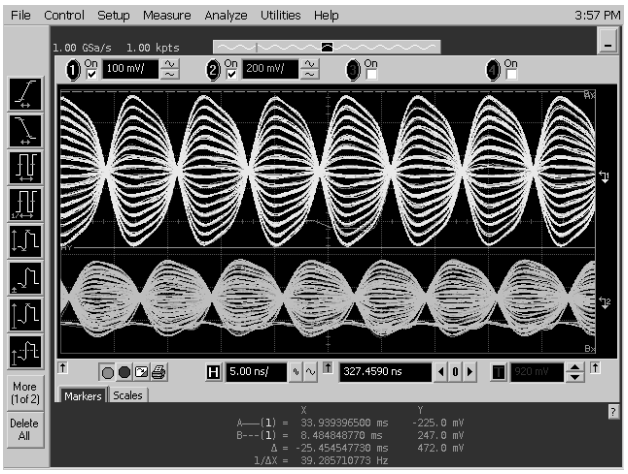

Figure 7: 256-QAM eye diagram

As the same range was used in all of the implemented modulators, the exact same maximum output value was expected in all cases. The Figures 8 and 9 prove that the same maximum out value was measured in all cases.

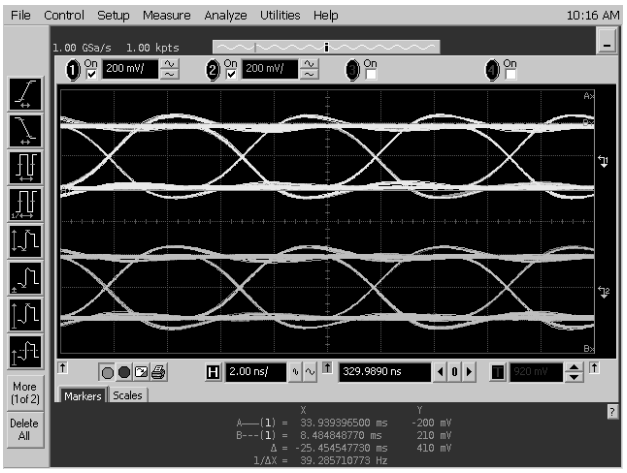

Figure 8: QPSK maximum output level

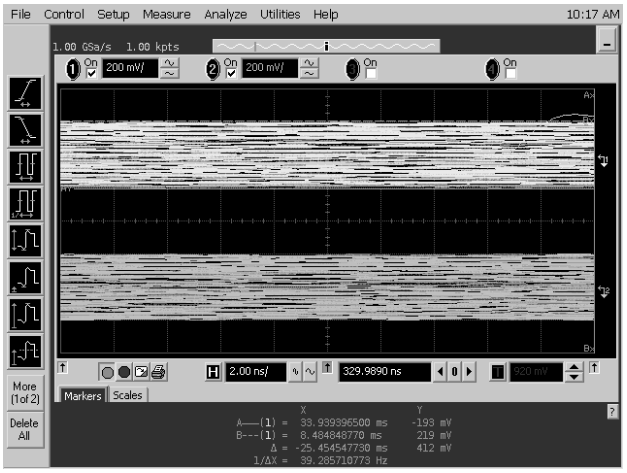

Figure 9: 256-QAM maximum output level

Since this constellation only consists of a total of four points, the distance between points is the same as the maximum and minimum amplitudes.

The Figure.10 provides information about the maximum and minimum output levels of QPSK : The maximum distance between points was 372 mV and The minimum distance between points was 291 mV. This constellation of256-QAM consists of 256 points, each theoretically representing 8 bits of data. The results were measured in Figure.11. The maximum distance between points was 25 mV and the minimum distance between points was 19 mV.

The maximum output voltage was measured to 500 mV at a 50ohm load which represents the maximum range of the bit digital value supplied to the conversion board ; the distance between the digital values 0 and 16383.

 DOI:10.32377/cvrjst0715

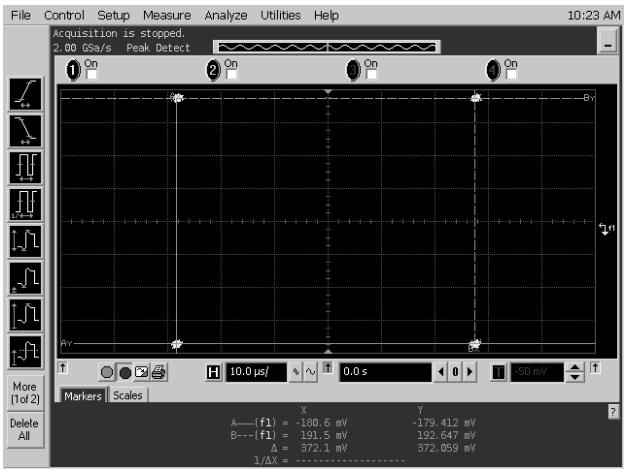

Figure 10: QPSK maximum distance between points

256-QAM.

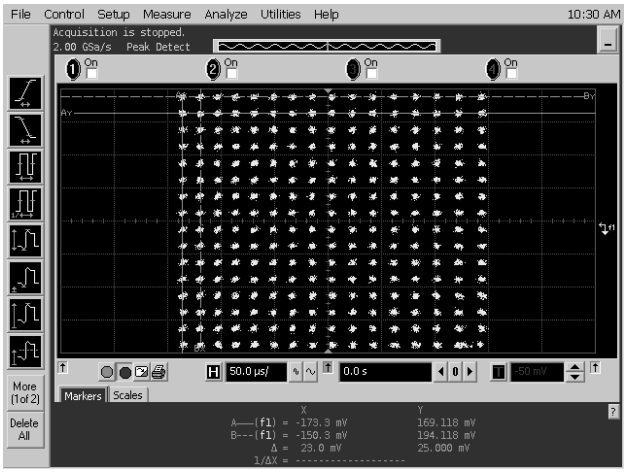

Figure 11: 256-QAM maximum distance between points

CONCLUSIONS

The work was set out to produce the following three tasks; 1. Generation of pseudo random data sequence. 2. Generation of I/Q symbol signals for an E-band I/Q modulator for QPSK and 256-QAM 3. Interfacing with existing modulator from Gotmic. The pseudo random data sequence was implemented first of all, with a detector accompanying it. All of the required modulators and several more have been implemented in a complete dynamic system to be used with any direct modulator. Interfacing with the existing modulator could not be performed due to hardware not being available, but the system is ready to be connected to any direct modulator that is required to be measured upon. The furnished system is delivered as a standalone system programmed onto the Altera DE3 board, as well as source code and programming files.

As stated earlier in the paper, there is room for extensions of the system to provide additional tools of evaluation. The two major tracks of extended functionality are Phase correction and Pre-distortion. Another track is to design the opposite module, the demodulator. In certain circumstances one might want to adjust the individual constellation points with different amplitude adjustments. One situation would be when there is an imbalance in the modulator which only affect certain constellation points and therefore are hard to adjust using the dynamic power or phase adjustments. With the help of a pre-distortion block, the individual constellation points can be adjusted before conversion into the analog domain. The individual adjustment could easily be implemented using two matrices with adjustment data for the I and Q signals respectively, and adjusting these matrices using a computer since there is limited abilities of user input on the Altera DE3 board.

REFERENCES

[1] 10 Gbps radio system. Project Specification, 04 2010. Internal project document.

[2] Wally H.W. Tuttlebee. Software defined radio: enabling technologies. John Wiley and Sons, 2002.

[3] J. Das. Review Of Digital Communication. New Age International, 1988.

[4] Mustafa Ergen. Mobile Broadband: Including WiMAX and LTE. Springer, 2009.

[5] Jean-Paul Linnartz Tim Schenk. RF imperfections in high-rate wireless systems. Springer, 2008.

[6] Peter Alfke. Efficient shift registers, lfsr counters, and long pseudo-random sequence generators. Xilinx XAPP 052, July 1996.

[7] Andrew Metcalfe Paul S. P. Cowpertwait. Introductory Time Series with R. Springer, 2009.