Implementation of 16-QAM Alamouti Encoder on FPGA for MIMO Testbed

Mohd Aminudin bin Mohd Khalid

Faculty of Electrical Engineering
MARA University of Technology (UiTM)
Shah Alam, Malaysia

Dr. Nur Idora binti Abdul Razak

Faculty of Electrical Engineering
MARA University of Technology (UiTM)
Shah Alam, Malaysia

Abstract— ***This project is aimed at implementing Alamouti's scheme on Polarizone's FPGA-based software defined radio platform and also to conduct experiment with RF transmission. A 16-QAM encoder is implemented in this project successfully. The system manages to encode incoming data to Alamouti codes by performing baseband signal processing and schedule its timing according to Alamouti encoding scheme. Detail design specification is presented and future enhancement for decoding and channel estimation is proposed.***

Keywords: Software Defined Radio, FPGA, Space Time Block Coding, Alamouti's Encoding, Transmit Diversity, Orthogonal Codes, MIMO Testbed.

I. Introduction

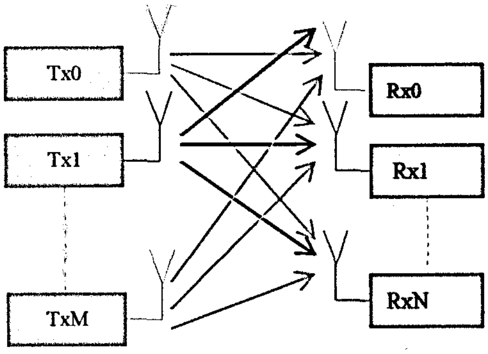


Figure 1: MIMO

In general, Multipe Input Multiple Output (MIMO) system is described as multiple antenna communication system as illustrated in *Figure* 1. There are M number of transmitter and N number of receiver directly connected for point to point communication. Many schemes can be implemented to enhance transmission reliability and capacity. In general, there are 2 general objectives for MIMO; i.e. spatial multiplexing and spatial diversity. The former is used to increase channel capacity and throughput by providing additional path for parallel transmission. The latter is used to form transmit diversity and receive diversity in which they can be used to combat fading by having more samples of transmission symbols per information; thereby increasing transmission reliability. Diversity gain in MIMO can be achieved by incorporating Space Time Block Code (STBC) encoder and decoder at transmitter and receiver respectively.

Alamouti [1] introduced encoding and decoding scheme with maximum two transmitting antenna and arbitrary number of receiving antenna. If only single antenna is used, by using Alamouti's scheme, the diversity gain is equivalent to that of two antennas at receiver with Maximum Ration Combining method; i.e. receiver diversity without transmit diversity. Alamouti's method can be generalized to N number of receiving antennas which results in 2N of diversity gain. As for decoding algorithm, the scheme assume receiver has channel state information either by using pilot symbol or other intelligent methods. The main contribution of Alamouti's scheme is the idea of having less number of antenna of at receiver and plurality of antennas at transmitter. This is suitable for mobile station application in which device size is a limiting factor. However, in this scheme the number of transmitting antenna is limited to two. For larger diversity gain, the only solution is to increase the number of receiving antennas. This concept has been enhanced by [2], [3] for greater number of transmit antennas. Other aspects of decoding has also been explored in [4] in term the application of artificial neural network in channel estimation and better decoding method.

This project is aimed to provide detail architecture of Alamouti Encoder specifically for Polarizone's Field Programmable Gate Array (FPGA) based Software Defined Radio (SDR) platform and also to verify the correctness of the implementation by monitoring baseband output with RF transmission. A 16-QAM encoder is implemented in this project successfully. The system manages to encode incoming data to Alamouti codes by performing baseband signal processing and schedule its timing according to Alamouti encoding scheme.

II. Literature Review

Alamouti encoder is implemented successfully in [5] and [6] by using Xilinx Virtex 4 FPGA. The MIMO-STBC Alamouti was designed based on 2x4 MIMO setup. The encoding and decoding algorithm are based on [1]. Their work started by performing simulation on MATLAB. Subsequently, the algorithm is implemented in hardware description language and hardware design simulation is performed and verified against MATLAB simulation result.

A 2x2 MIMO-OFDM is implemented in [7] based on Sundance's Software Radio Development Kit. The paper focuses on MIMO testbed design which implements the whole suite of 2x2 MIMO systems which include a novel synchronization algorithm for multiple antenna system. The system also incorporated 802.11n protocol. Alamouti's 2 transmit antenna with 2 receive antennas encoding and decoding scheme is implemented in the testbed's FPGA. An indoor experiment was conducted between transmitter and receiver that reside in neighbouring room. It was reported that the system worked well and during the testing.

In [8] also implemented 2x3 Alamouti's STBC system and tested by using channel models implemented on FPGA. Their work focuses on the performance evaluation of 2x1, 2x2 and 2x3 Alamouti STBC with different type of modulation schemes; i.e. BPSK, QPSK and 16QAM. Their results, as expected confirmed that the 2x3 system performs the best for all modulation schemes tested.

In [10], Cattoni et. al. implemented a 4x4 MIMO-OFDM STBC system based on 802.16x framework. The FPGA implementation considered issues like computational complexity, interference and noise reduction, real time execution and efficient hardware resource management. Decoding technique is straight forward and did not include channel feedback mechanism. The STBC which is based on the extended Alamouti's code is simulated for 4x1 and 4x4 and 2x2 setup by using Matlab-Simulink. The result shows that the 4x4 setup is less robust at high SNR. This is because more antennas give higher interference. Furthermore, there is no intelligence implemented to monitor channel state information. Substraction combining method was chosen because of reduction in computational burden for multi-antenna system. FPGA bit steam has been generated and uploaded into FPGA. A co-simulation between Matlab-Simulink and FPGA has been conducted to verify the correctness of FPGA implementation.

In [11], a 2x1 and 2x2 STBC are implemented on WARP board [9] for testbed. An image transmission experiment by using joint source channel coding technique is conducted and the performance is evaluated by using 16-QAM. The effectiveness of incorporating of error correction code was tested as well. This work assumed that the receiver has full information of channel state. It is reported that the performance of 2x2 outperformed 2x1 by the assistance of 2/5 convolutional code.

III. 16-QAM ALAMOUTI'S SPACE TIME BLOCK CODING

In FPGA, a basic 16-QAM encoding is performed by using table lookup method. Each 16-QAM symbol is represented by a pair of mapping table for and I and Q value, whereby each entries are indexed from 0 to N-1; for N number of sample points as illustrated in *Figure* 2 for 100% amplitude and 45 degree phase. In our implementation. There are 16 pairs of I and Q values that represent each symbols in 16-QAM. Each amplitude is sent to Digital to Analog Converter (DAC) module as voltage scale to be converted to its equivalent analogue value for carrier modulation purposes.

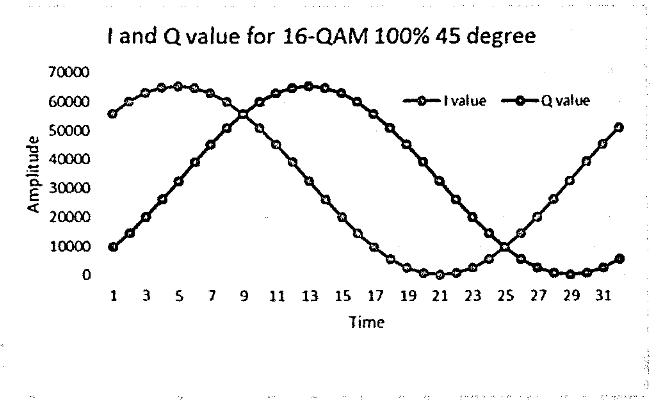


Figure 2: I and Q example

Figure 3, illustrates the basic Alamouti encoding scheme in the form of 16-QAM 2x1 setup; i.e. Data is transmitted over wireless medium from two antenna modules and received by single antenna module. Since the basic unit of data for original 16-QAM is four bits, for Alamouti encoding scheme it requires at least eight bits of input data from SOURCE. This is the smallest data size for Alamouti encoding scheme.

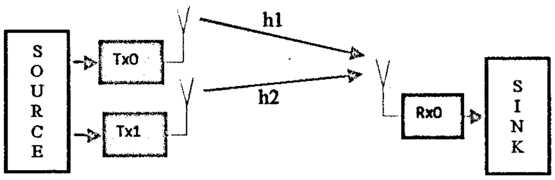


Figure 3: Basic Alamouti Scheme

A unit of eight bits input from SOURCE are separated into two registers; i.e. *BUFF1* and *BUFF0* for X_2 and X_1 respectively. These two sets of data are encoded according to Alamouti's encoding scheme as described in *Table 1*. A correct mapping for QAM symbol for *BUFF1* and *BUFF0* is conducted according to 16-QAM symbol table as described previously. For the first time slot Tx0 operates on BUFF0 *and Tx1 operates based on Tx1 whereby the* N number of points each for X_1 and X_2 are sent to DAC module for carrier modulation. *Tx0* transmits X_1 by processing the array of $X_1[n] = I_1[n] + jQ_1[n]$ for n = 0 to 31. At the same period of time, *Tx1* transmits X_2 by processing the array of $X_2[n] = I_2[n] + jQ_2[n]$ for n = 0 to 31. For the second time slot, *Tx0* operates based on *BUFF1* while *Tx1* operates based on *BUFF0* according to Table 1. At this period of time, *Tx0* transmits $-X_2^*$ while *Tx1* transmits X_1^* for each scale points describe in the corresponding 16-QAM mapping table. Note that the number of N depends on designer system design consideration and architecture.

Table 1: Alamouti's Encoding Scheme

	Time Slot 1	*Time Slot 2*
Antenna1 (Tx0)	$X_1[n] = I_1[n] + jQ_1[n]$	$-X_2{}^*[n] = -I_2[n] + jQ_2[n]$
Antenna2 (Tx1)	$X_2[n] = I_2[n] + jQ_2[n]$	$X_1{}^*[n] = I_1[n] - jQ_1[n]$

At the receiving side, two set of vectors are received in two time slots as listed in Table 2.These vector contains information regarding to X_1 and X_2. These two vectors must be detected and stored in registers for further processing according to the equation stated above

Table 2: Receiving vectors

	Time Slot 1	*Time Slot 2*
Antenna1 (Rx0)	$Y_1[n] = Ir_1[n] + jQr_1[n]$	$Y_2[n] = Ir_2[n] + jQr_2[n]$

$$Y_1[n] = h_1[n](X_1[n]) + h_2[n](X_2[n]) + n1[n] \quad (1)$$

$$Y_2[n] = h_1[n](-X_2^*[n]) + h_2[n]X_1^*[n] + n2[n] \quad (2)$$

IV. TESTBED DESIGN

A. 16-QAM Alamouti Encoder RTL Design

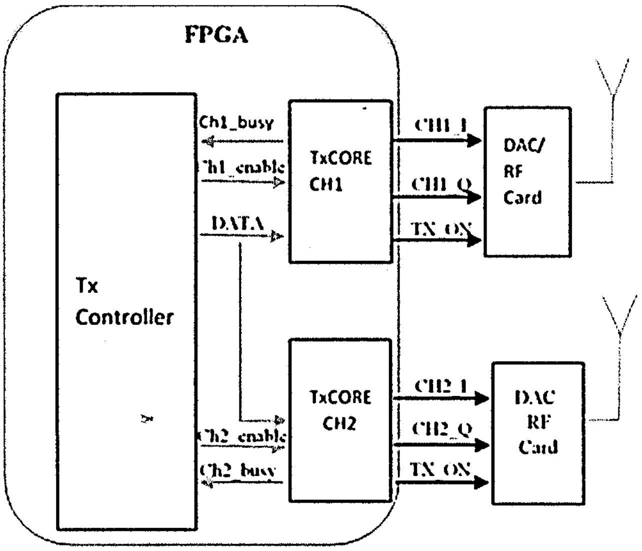


Figure 4: Block Diagram for Alamouti Encoder

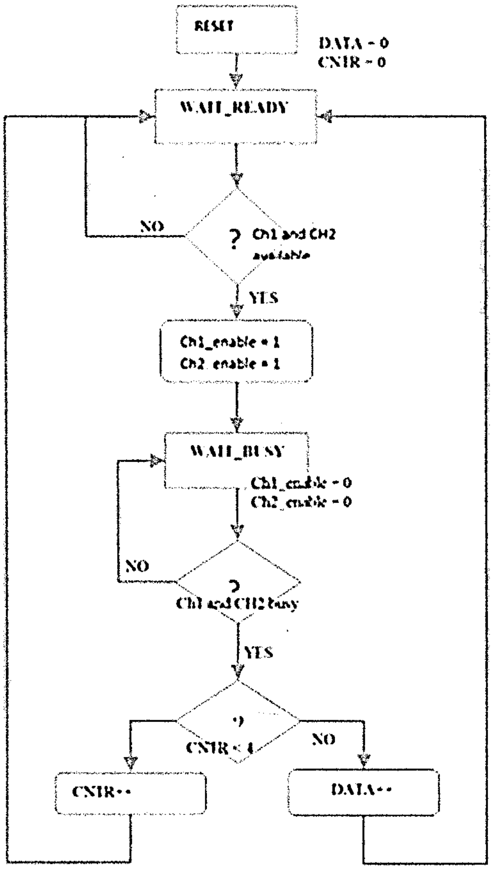


Figure 5: Algorithmic State Machine for TxController

As depicted in *Figure 4*, a dedicated FPGA that is connected to all units of antenna modules is provided inside the test-bed system for baseband processing. However for Alamouti Space Time Block Code, the system design is limited to two units of antenna modules although Polarizone MIMO provides four RF modules. The FPGA is connected to RF modules which consists of Digital-to-Analog Converter and antennas. The Register Transfer Level design for Alamouti encoding is implemented inside the FPGA fabric. The architecture is design in such a way that it is extendable in future for additional features such as the ability to fetch data from external system such as network interface, serial connection or via USB interface.

In this design, there are three main components inside the FPGA fabrics; i.e. Tx Controller, TxCORE for Channel 1 and TxCORE for Channel 2. Another important aspect in digital design is reset and synchronization between all modules in the digital circuit. For Alamouti encoding, it is important to ensure all modules can be controlled by reset signal and all are synchronized to the same clock signal. This is due to the fact that Alamouti encoding requires two time slots for RF transmission. Therefore, the timing for the 2 channels must be same. As of current design, TxControlller only functions as data provider to all channels in the form of eight bits counter which means the data supplied for Alamouti encoding is locally generated inside the FPGA itself. As described above, for enhancement purposes this module can be modified to support external input. TxCORE for Channel 1 and TxCORE for Channel 2 is dedicated baseband processor for each antenna module. It encodes data supplied from TxController and schedule its transmission in two time slots according to Alamouti scheme. To be precise, it performs 16 QAM symbol mapping from four bits binary data to its corresponding sinusoidal waveform. The amplitude of sample points in sinusoidal wave are sent to DAC module for carrier modulation. Different set of sampling points are sent for different bit pattern.

The behaviour of TxController is described in the form of Algorithmic State Machine (ASM) notation in *Figure 5*. At RESET state, system initializes with a known symbol; i.e. DATA = 0. This value can be changed according to system specification. At WAIT_READY state, the system waits for the readiness flag from Channel 1 and Channel 2. If both channels are busy or not ready, it will keep looping in the same state. When TxCORE for channel 1 and channel 2 flag its readiness, state transition occurs from WAIT_READY state to WAIT_BUSY state and TxControllers trigger enable signal for TxCORE activation. At WAIT_BUSY state, it waits until both channels are busy for state transition to WAIT_READY state, otherwise it loops at the same state. The system perform four sessions of known symbol transmission so that receiver is able to perform symbol timing synchronization and channel estimation.

Figure 6 specifies the behaviour of TxCORE for channel 1 in the form of ASM notation. The TxCORE for channel 2 also behaves the same with that of TxCORE for channel 1. TxCORE system initializes at INIT state for synchronization purposes which transmits 4 known symbols. Subsequently, it transits to READY state; waiting for enable signal from TxController. At this point of time, TxCORE for Channel 1 and TxCORE for Channel 2 are waiting for Ch1_enable and CH2_enable signal to arrive, respectively. Once detected, state transition occurs from READY state to BUSY state. BUSY state is a critical state whereby Alamouti encoding procedure takes place. At this point, both TxCOREs must operate according to Alamouti encoding scheme as described in the previous section. At time slot 1, TxCORE for Channel 1 modulates carrier signal by sending 32 sample points scale to DAC card, each for I and Q channel. The DAC card modulates carrier signal according the scale of the sample point receiv̆ed. Therefore the process at time slot 1 lasts for 32 clock cycles. At this state, TxCORE for Channel 1 and TxCORE for Channel 2 send *($I_1[n]+jQ_1[n]$)* and *($I_2[n]+jQ_2[n]$)* for Channel 1 and Channel 2 respectively. For time slot 2, TxCORE for Channel 1 and TxCORE for Channel 2 send *($-I_2[n]+ jQ_2[n]$)* and *($I_1[n] - jQ_1[n]$)* for Channel 1 and Channel 2 respectively.

All design based on the previously described system architecture is implemented in VHDL manually. The design is simulated by using Xilinx ISim HDL simulator. Subsequently, the design is integrated with Polarizone framework to prepare for on-chip encoding experiment with RF transmission. Subsequently all integrated modules for encoder has been synthesized to FPGA-based digital logic gates for hardware implementation.

During experiment, Xilinx ChipScope with built-in logic analyser core is being used extensively for baseband data acquisition and signal inspection. At the same time, spectrum analyser is used to observe the spectrum and perform transmit power measurement. The discussion on the acquired output is analysed in the next section.

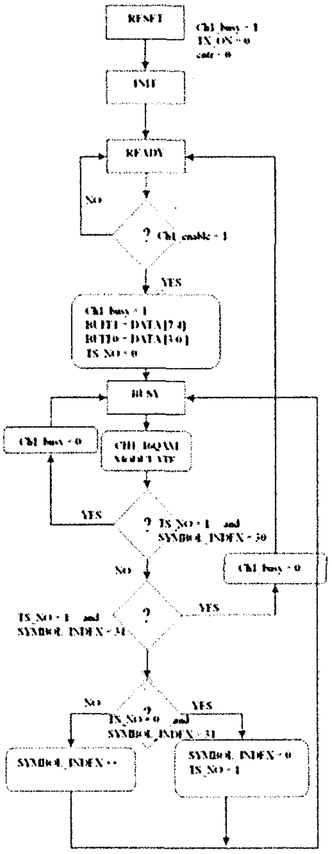


Figure 6: Algorithmic State Machine for TxCORE1

V. Discussions

RTL simulation and actual RF transmission is conducted to verify the design specification as mentioned in the previous section. The detail behaviour of the designed encoder is described detail along with its features. The feature should be used in the design of decoder for real time decoding.

A. Encoder Behaviour

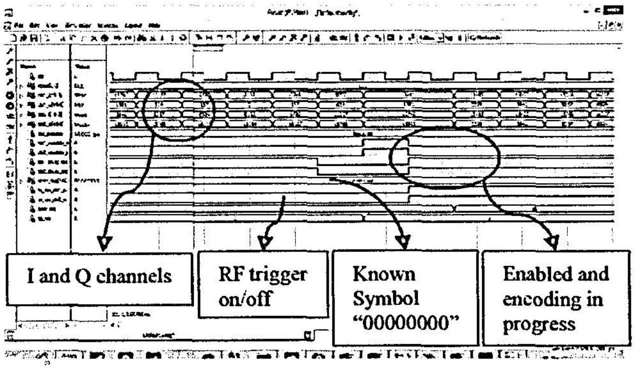


Figure 7: Pre and Post-Initialization Process and Known Symbol Transmission

Figure 7 shows the simulation result during initialization of Alamouti encoder. This is the first instant that both channels are turned on for known symbol transmission. Prior to this particular clock cycle, it can be seen that the I and Q channel for Channel 1 and Channel 2 also produces values but the RF modules triggering signal are turned off. This to ensure the RF modules are in ready state, so for safety margin, around 64 clock cycle are allowed just for this purpose. Also for initialization purposes, a set of known data with value "00000000" is sent out via the RF modules. This data is encoded by Alamouti encoder and the process repeated four times. This is to include synchronization feature for integration with receiver. With this initialization procedure, receiver will be able to perform symbol timing synchronization and channel estimation prior to unknown symbol transmission.

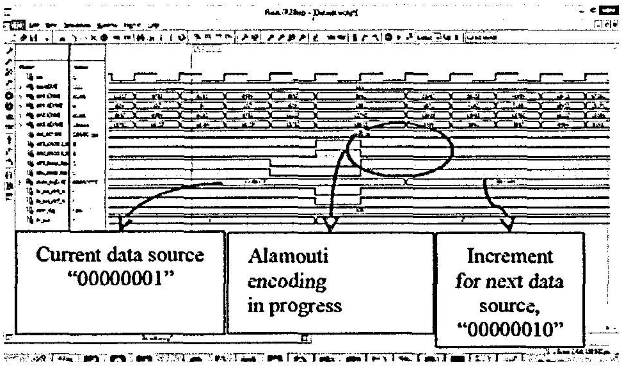

Current data source "00000001"
Alamouti encoding in progress
Increment for next data source, "00000010"

Figure 8: Start to transmit incremental data source

After known symbol transmission is completed, TxController starts to increase its data source one by one. As shown in *Figure 8*, data "00000001" is encoded to Alamouti encoded I and Q value. At this point, TxController trigger CH1_enable and CH2_enable and subsequently it can be seen that CH1_busy and CH2_busy are triggered with tx_on_ch1_o and tx_on_ch2_n are switched on. This indicates that Alamouti encoding is in progress. 64 clock cycles are required to complete a single Alamouti encoding scheme since a unit of 16-QAM symbol is designed with 32 sample points.

B. Encoded Output Verification

Table 3: Example expected output when X_1 = "0001", X_2 = "0000"

1st Time Slot	2nd Time Slot
I_1 = (75%, 22.5°), Q_1 = (75%, 292.5°)	$-I_2$, Q_2
I_2 = (25%, 45°), Q_2 = (25%, 315°)	I_1, $-Q_1$

For example, an input data from TxController is fixed as "00000001"; i.e. X_1 = "0001" and X_2 = "0000". The expected waveform is according to Alamouti rectangular 16-QAM mapping as in *Table 3*. Output value for I and Q value are recorded during HDL simulation and graph plots in *Figure 9* and *Figure 10* confirms that the output is as expected. The graphs plots show correct encoding for X_1 and X_2 for the first time slot at Channel and Channel 2. It is possible to cross check whether or not the encoding for second time slot is encoded correctly. Time Slot 1 for Channel 1 can be cross checked with Time Slot 2 for Channel 2 for symbol X1. It can be seen that the Q portion of X_1 at Time Slot 1 at Channel 1 is flipped at Time Slot 2 at Channel 2 due to complex conjugate property. Similarly for X_2, the I portion at Time Slot 1 for Channel 2 is flipped at Time Slot 2 at Channel 1.

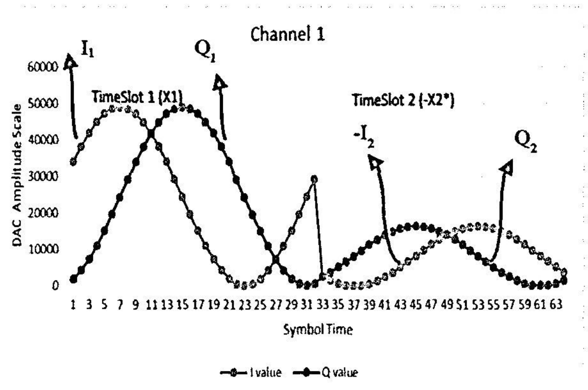


Figure 9: Channel 1 output in RTL simulation (DAC scale versus symbol time)

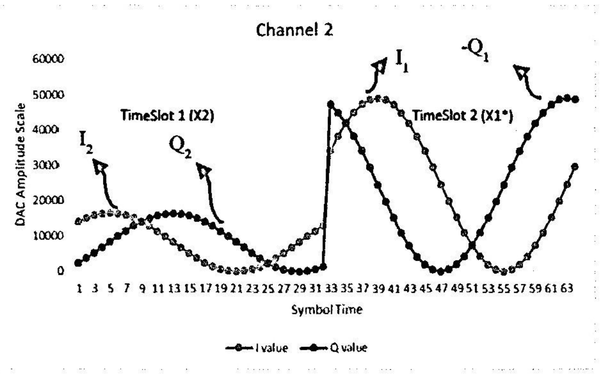


Figure 10: Channel 2 output in RTL simulation (DAC scale versus symbol time)

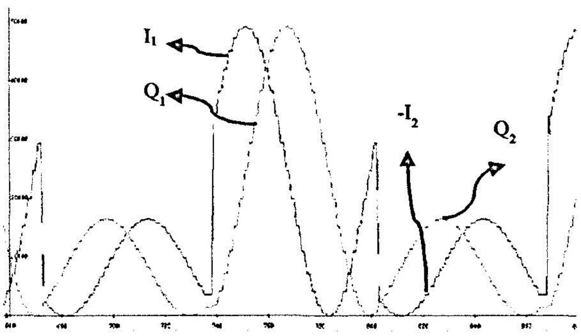


Figure 11: Channel 1 Output from Xilinx ChipScope

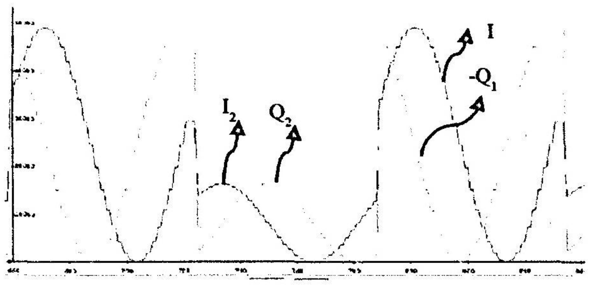


Figure 12: Channel 2 Output from Xlinx ChipScope

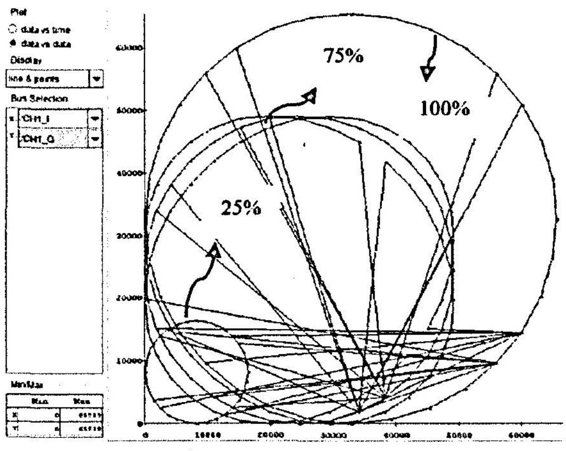
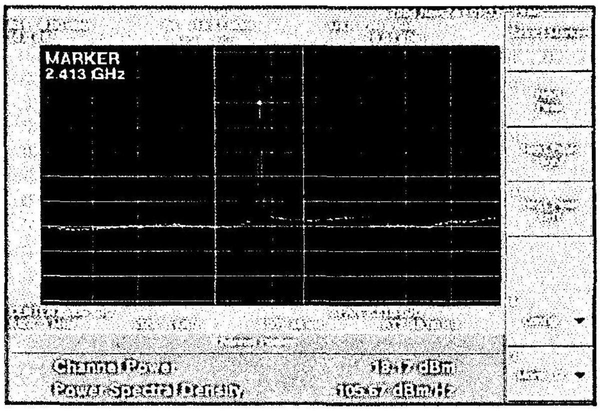


Figure 13: I vs Q sample plot

The design is executed on Polarizone SDR Platform with RF transmission. Baseband signal data acquisition is performed by using ChipScope and data is collected and analysed. It is can be observed that the baseband signal produced by the test-bed was similar to that of VHDL simulation as in *Figure 11* and *Figure 12*.

Figure 13 shows the constellation plot resulted from continuous execution with incremental input data. This is an example of instantaneous plot of I and Q value for Channel 1. The three main circles corresponds to the 3 level of amplitude scale (25%, 75% and 100%) for 16-QAM. The plot depends on the number data that can be stored in the internal buffer and the pair of points is the constellation points that are detected and stored in internal buffer.

Figure 14 shows the spectrum captured during RF transmission. It shows that the spectrum is centred around 2.4GHz as per system specification. The transmitter power is -18.17 dBm.



Channel Power -18.17 dBm
Power Spectral Density -105.67 dBm/Hz

Figure 14: Spectrum Analyzer

C. Waveform design at Transmitter

In this design, each 16-QAM symbol is designed with 32 points by considering the input to DAC module. The DAC module in this MIMO platform is 16-bits, which means the sets of amplitude value for scaling range between 0 to 21^6-1; i.e. from 0 to 65535. For QPSK, 32 points is enough to support its 4 constellation points since each constellation points are separated by other 7 points. For 16-QAM with 32 points, each 16-QAM constellation points is separated by one or two points. This can be improved by creating a sample points with higher resolution; i.e. 64 points or 128 points. This way better separation between 16-QAM symbols can be made for better constellation. However, increasing the number of points will result in longer clock cycles required to complete a single Alamouti encoding; i.e. 64-points symbols requires 128 clock cycles. For a higher level modulation scheme such as 256-QAM, higher resolution of symbol points are required for data encoding and for better signal constellation. This is the trade-off between the number of bits per symbol versus number of clock cycles required versus better signal separation.

Figure 13 also shows two clean circles that are plotted for amplitude scale 25% and 100%. For amplitude scale 75%, there are 3 types or circles and this may affects the quality of signal produced at RF end. Increasing the number of points per symbol may help to improvise the constellation plot and this is subject to further investigation.

D. Future Work

It also important to note that the Polarizone SDR platform was not design to operate with external central clock (i.e. Function Generator) for synchronization purposes. Therefore, the synchronization must be built-in inside the low level protocol between transmitter and receiver. A common method is to transmit a series of known symbol to indicate the starting point prior to incoming data encoding at transmitter. This feature has been implemented in the encoder and it should be used in the decoder design in future. A known symbol transmission can be used to synchronize transmitter and receiver as well as performing channel estimation in order to determine current H matrix.

VI. Conclusions

A method to implement Alamouti encoder in the form of algorithmic state machine is presented in detail in which the behavior of each sub-system is described. Design verification is conducted during HDL simulation and the expected I and Q values are verified. Baseband output from ChipScope Logic Analyzer is verified against HDL simulation result. It can be concluded that the encoder is able to encode data stream correctly.

It is suggested to use the feature introduced in the proposed encoding system to build a complete Alamouti decoder with symbol synchronization and channel estimation module. With this a complete RF transmission experiment can be performed by changing the transmit power and real time on-chip calculation of symbol error rate.

References

[1] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Selected Areas in Communications,* vol. 16, no. 8, 1998.

[2] H. J. Vahid Tarokh, A.R. Calderbank, "Space Time Block Codes from Orthogonal Designs," *IEEE Transactions on Information Theory,* vol. 45, no. 5, 1999.

[3] H. Jafarkhani, "A Quasi-Orthogonal Space-Time Block Code," *IEEE Transactions on Communications,* vol. 49, no. 1, 2001.

[4] K. K. S. Samar Jyoti Saikia, "ANN based STBC-MIMO set-up for Wireless Communication," *International Journal of Smart Sensors and Ad Hoc Networks,* vol. 1, no. 3, 2012.

[5] M. T. I. Mostafa Wasiuddin, Norbahiah Misran, "Implementation of Alamouti Encoder Using FPGA for MIMO Testbed," *Internal Conference on Advanced Computer Control,* 2009.

[6] M. W. N. Mohammad T. Islam, Norbahiah Misran, "Design and Implementation of Alamouti Encoder for 4G Wireless System," *EUROCON 2009, EUROCON '09. IEEE,* 2009.

[7] N. C. Jian Sun, Dongfeng Yuan, "Implementation of a 2x2 MIMO-OFDM Real-time System on DSP/FPGA Platform," *International Conference on Communications and Mobile Computing,* 2011.

[8] D. B. S. P. Sindhu, K. Hari Kishore, "Implementation of Alamouti 2x3 Code on FPGA Board," *International Journal of Computer Applications,* 2013.

[9] "WARP: Wireless Open Access Research Platform (http://warpproject.org/)."

[10] Y. L. M. Andrea F. Cattoni, Claudio Sacchi, "Efficient FPGA Implementation of a STBC-OFDM Combiner for an IEEE 802.16 Software Radio Receiver," *Telecommunications Systems Journal,* 2013.

[11] S. S. Shreya Kaushal, "A 2x2 FPGA based WARP Tested for Colored Image Transmission in MIMO Systems," *International Journal of Science and Modern Engineering,* vol. 1, no. 6, 2013