Rapid Prototyping and Performance Evaluation of a MIMO CDMA System using an FPGA-Based Hardware Platform

Mostafa Hefnawi
Department of Electrical and Computer Engineering
Royal Military College of Canada
Kingston, Canada
hefnawi@rmc.ca

***Abstract*— This paper investigates the rapid prototyping of a multiple input-multiple-output direct sequence-code division multiple access (MIMO DS-CDMA) system with rake receiver, implemented on a field programmable gate array (FPGA) based hardware platform. The hardware implementation is created using the Altera DSP builder– a MATLAB/Simulink based system-level design tool and the Stratix EP1S80 DSP development board from Altera. The hardware-in-the-loop (HIL) co-simulation and the Logic Analyzer are used with the physical FPGA board implementing the design to evaluate the system performance and to verify the functionality of the hardware implementation in the MATLAB/Simulink environment. Results show that, in general, the bit error rate (BER) of the hardware implementation fell within the confidence intervals of the simulated BER.**

***Index Terms*— Rapid prototyping, FPGA, Hardware-in-the-loop, DS-CDMA, MIMO, Space-time coding, Rake receiver.**

I. INTRODUCTION

Wireless standards are continuously evolving to support higher data rates by incorporating advanced baseband processing techniques such as multiple-input multiple-output (MIMO), space-time coding (STC), and adaptive modulation techniques. Thus, future wireless devices will need to support multiple air-interfaces and modulation formats as well as the capability to support a completely different standard. Software defined radio (SDR) is recognized as a key technology to enable such functionality by using a reconfigurable hardware platform based on Field-programmable gate arrays (FPGAs). On the other hand, it is well known that the design and development of current wireless systems is characterized by very short production cycles and employ a multidisciplinary approach. To address these challenges, FPGA based rapid prototyping methodology that requires minimal FPGA design skills and a very basic knowledge of hardware description languages has received much attention from designers worldwide [1]-[5]. It allows communication designers to quickly verify the performance of their systems and test them in a real-time and under real world conditions. MATLAB/Simulink based design tools such as System Generator from Xilinx [6] and DSP Builder from Altera [7] are becoming very popular in achieving these tasks. In this paper, a MIMO DS-CDMA system with rake receiver [8] is implemented in FPGA using the Stratix EP1S80 DSP development board from Altera which is a powerful prototyping tool that provides a close integration with Matlab / Simulink and Altera's DSP Builder [9]. MIMO technology constitutes a very important part in the implementation of modern broadband wireless communication systems. This technology is being utilized in new mobile and broadband wireless systems, including the 3G, 4G-LTE, and the new 4G-LTE-Advanced systems, to greatly improve both spectral and power efficiency. It will also be supported in the new cognitive radio based 5G systems to make significantly better use of available spectrum via space division multiplexing access (SDMA). Various FPGA implementation approaches that enable the rapid prototyping of MIMO systems have been considered [10]-[12]. However, in most of the cases, the work described in such papers mainly focuses on partial FPGA implementations that lack real-time operation (operate in off-line mode), assume perfect MIMO channel knowledge, or do not consider performance validation in presence of multipath MIMO channels. The implementation in this paper includes a space-time coded (STC) CDMA transmitter, a propagation channel, a channel estimator, a rake receiver, and an STC decoder, which is composed of a combiner and a maximum likelihood detector (MLD). Using DSP Builder, a design model of our system is created in the MATLAB/Simulink software to generate VHDL files for synthesis and compilation. The design is first verified in the Simulink environment then implemented in the EP1S80 FPGA development board using the hardware in the loop (HIL) co-simulation concept. The advantage of using this system-level approach is the possibility to test and integrate our hardware implementation within the simulation environment in the form of a Simulink block allowing us to test the system performance in different scenarios. The flexible hardware implementation realized in this paper can be viewed as a base of future MIMO implementation and can be easily redeployed in other configurations to meet specific MIMO designs.

II. System Model

In this implementation we consider a 2x2 MIMO DS-CDMA system with Rake receiver as depicted in Figure 1. At the mobile user transmitter, the binary information is fed to a BPSK modulator, which generates constellation symbols. Using the Alamouti's space-time block coding (STBC) scheme [13, 14], the symbols are grouped in pairs and each pair of symbols is transmitted simultaneously from antenna #0 and antenna #1 as per Table 1, where T is the sampling period, and (*) represents the complex conjugate. This allows a single antenna to decode each symbol with a diversity order of two. The output of each branch of the STBC block is then spread using a direct pseudo-random sequence.

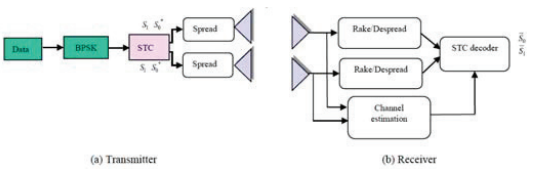

Figure 1: 2x2 MIMO DS-CDMA with Rake despreader

Table 1
Alamouti's transmit diversity scheme

Time	*TX antenna #0*	*TX antenna #1*
t	s_0	s_1
$t+T$	$-s_1^*$	s_0^*

The received signals during the first and second signalling intervals for each receiving antenna are given in Table 2, where L is the number of faded paths and h_{ij}^l is the fading channel gain of the l^{th} path from the i^{th} transmitting antenna to the j^{th} receiving antenna. The receiver estimates the channel coefficients and uses them with the rake despreader to perform a coherent combination of the signals coming from different paths by providing a separate correlation receiver for each of the multipath signals [15].

Table 2
Space-time received signals

Time		***RX antenna #0***		***RX antenna #1***
t	$r_0 =$	$\sum_{l=1}^{L} h_{11}^l s_0 + h_{21}^l s_1 + n_0$	$r_2 =$	$\sum_{l=1}^{L} h_{12}^l s_0 + h_{21}^l s_1 + n_2$
$t+T$	$r_1 =$	$\sum_{l=1}^{L} -h_{11}^l s_1^* + h_{21}^l s_0^* + n_1$	$r_3 =$	$\sum_{l=1}^{L} -h_{12}^l s_1^* + h_{21}^l s_0^* + n_3$

The resulting symbols at the rake despreader output are passed into a combiner. The combiner takes the first and the second symbols and outputs the following two symbols:

$$\begin{aligned} \tilde{s}_0 &= g_{11}^* r_0 + g_{21} r_1^* + g_{12}^* r_2 + g_{22} r_3^* \\ \tilde{s}_1 &= g_{21}^* r_0 - g_{11} r_1^* + g_{22}^* r_2 - g_{12} r_3^* \end{aligned} \quad (1)$$

where g_{ij} is the complex combined paths obtained at the output of the rake receiver. The estimated $\tilde{s}_0$ *and* $\tilde{s}_1$ are then used in the maximum likelihood detector to decide which symbol is being transmitted.

To estimate the channel a traditional pilot-assisted method with an all-ones sequence is used. The received signals, when a pilot signal is transmitted, can be expressed as:

$$\begin{aligned} r_0 &= r(t) = h_0 s_0 + h_1 s_1 + n_0 \\ r_1 &= r(t+T) = -h_0 s_1^* + h_1 s_0^* + n_1 \end{aligned} \quad (2)$$

For the pilot signals $s_0 = s_1 = 1$, Eq. (2) can be reduced to

$$\begin{aligned} r_0 &= h_0 + h_1 s_1 + n_0 \\ r_1 &= -h_0 + h_1 + n_1 \end{aligned} \quad (3)$$

where h_0 and h_1 are complex terms that can be expressed as

$$\begin{aligned} h_0 &= a_0 + jb_0 \\ h_1 &= a_1 + jb_1 \end{aligned} \quad (4)$$

Finally, the real and imaginary part of h_0 and h_1 can be expressed as:

$$\begin{aligned} a_0 &= \frac{\mathrm{Re}(r_0) - \mathrm{Re}(r_1)}{2}, \quad a_1 = \frac{\mathrm{Re}(r_0) + \mathrm{Re}(r_1)}{2} \\ b_0 &= \frac{\mathrm{Im}(r_0) - \mathrm{Im}(r_1)}{2}, \quad b_1 = \frac{\mathrm{Im}(r_0) + \mathrm{Im}(r_1)}{2} \end{aligned} \quad (5)$$

III. FPGA Implementation Using DSP Builder

The implementation of the transmitter using DSP builder blocks is shown in Figure 2. The binary data stream is generated using a "pattern" block with the last two pilot bits set to "1"s for channel estimation. The data stream enters a BPSK modulator and the resulting symbols are applied to the STBC block created using the subsystem of Figure 3.

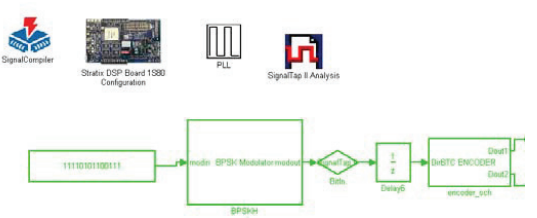

Figure 2: Implementation of the Transmitter

The STBC block generates two binary data streams. The first 12 least significant bits (LSB) are the information bits and the last two most significant bits (MSB) are the two pilot bits

set to "1"s. The subsystem also uses "complex conjugate", "complex addition", and "delay" blocks to generate the symbols from each branch according to the Alamouti's scheme described in Table 1. The two generated STBC signals are then spread by performing an XOR with a pseudo-noise spreading code (PNSC). The sampling rate of the data stream from the pattern is set to 2 Mega samples per second (Mbps) and the PNSC sequence sampling rate is set to 20 Mbps which results in a data rate at the spreader output of 20 Mbps.

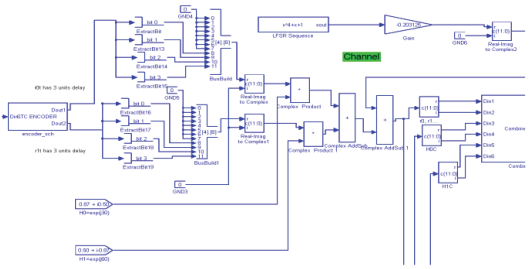

Figure 3: STBC Transmitter using DSP Builder blocks

In order for the system to support multiple data transmission rates, a PLL block is added to the model with the input frequency set to 80 Mbps, and the two output frequencies set to 2 Mbps and 20 Mbps.

The BPSK spread spectrum signals from both branches are passed into a two-ray fading channel modeled using delays and gain multiplication as shown in Figure 4.

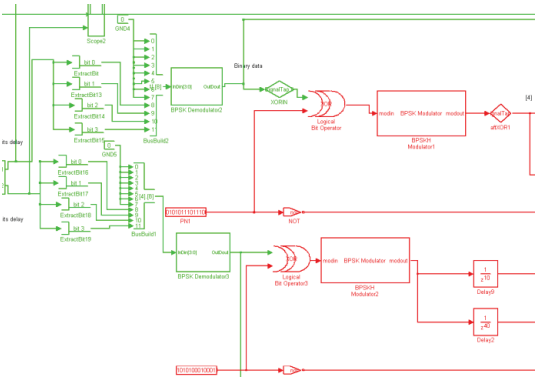

Figure 4: STBC DS-CDMA with two-ray fading channel

Figure 5 shows the implementation of the receiver, which consists of two receive antennas and each antenna is equipped with two fingers rake processing. To recover the original signal, a block "Tsamp" is used at the output of the rake despreader to downsample the signal from 20 Mbps to 2Mbps.

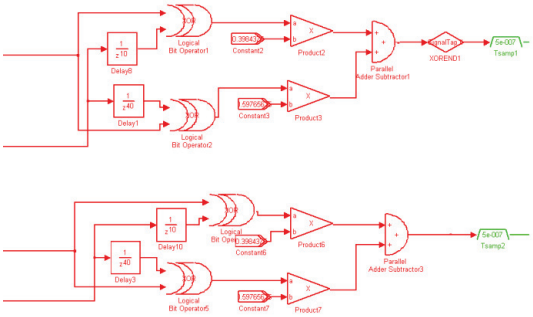

Figure 5: Two fingers RAKE receiver implementation

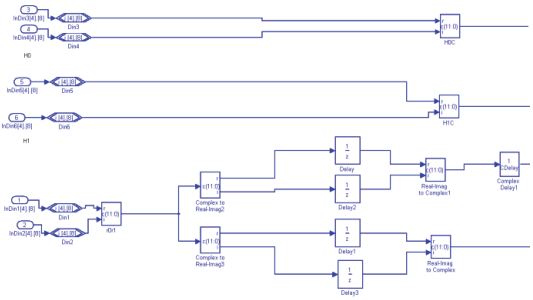

Figure 6: STBC Combiner subsystem – Stage 1

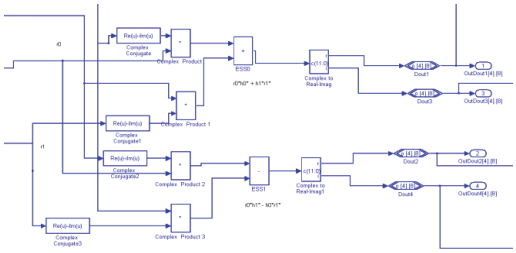

Figure 7: STBC Combiner subsystem – Stage 2

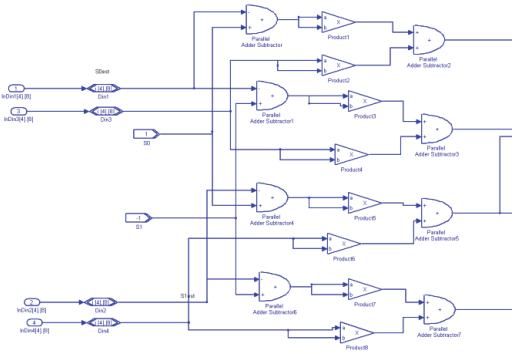

Figure 8: Maximum Likelihood detector implementation

The rake receiver output from each antenna is applied to the STBC decoder that consists of a combiner (Figure 6 and 7) and a MLD (Figure 8). The first stage of the combiner (Figure 6) uses two "delay" blocks with the clock phase selection set to 01 and 10 to extract r_0 and r_1, respectively. The second stage (Figure 7), uses "complex conjugate", "complex addition", and "multiplication" blocks to determine $\tilde{s}_0$ *and* $\tilde{s}_1$ as per Equation 1. The combined signals are then sent to the maximum likelihood detector of Figure 8, which uses the decision rule to decide if a 1 or –1 is being received. The decision scheme consists of calculating the minimum Euclidian distance between the received signals, (s_0, s_1) and (-1, 1) using two branches. The output of the two decision branches is then connected to a MUX block to combine both signals into a serial data stream. Figure 9 on the other hand, shows the block that implements the channel estimation. At the input of the channel estimation block (CEB), a data stream of 14 samples is received. The first 12 samples represent the received data and the last 2 samples represent the received pilot. In order for the CEB to extract the last pair of samples, (r_0, r_1), which is generated from the last pair of the transmitted pilot symbols $(s_0 = 1, s_1 = 1)$, the received signal is split into two branches, each with a one bit delay block, to extract r_0 and r_1 separately. The clock phase selection of the delay block is set to 000100000000 for r_0 and to 00001000000000 for r_1. Since the upper branch is one bit period ahead of the lower branch, the algorithm requires that both branches be synchronized. So, another delay with one bit period is added to the upper branch. Once the pair (r_0, r_1) is extracted, Equation 6 can be used to calculate the estimated channel parameters h_0 and h_1 which are then used by the STC decoder.

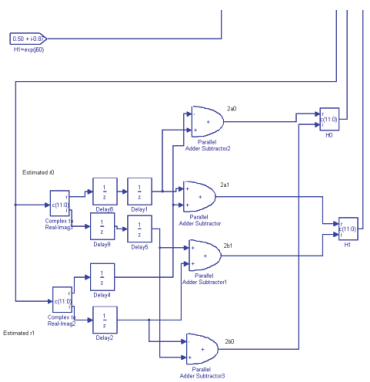

Figure 9: Channel estimation block

To remove the two pilot bits, the subsystem called "bit14to12" is used as shown in Figure 10.

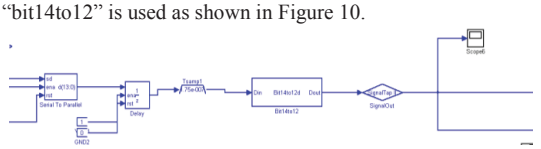

Figure 10: Pilot bits removal

Since the DAC needs a 14 bits unsigned bus type, the 12 bits signed bus type is used to build a 14 bits unsigned bus as shown in Figure 11. Basically, the 12-bit bus is used as the most significant 12 bits and the least significant 2 bits are set to 0.

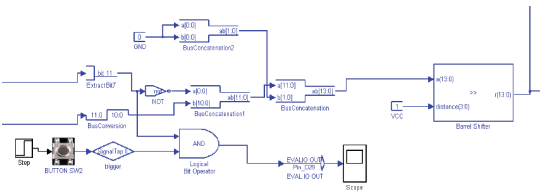

Figure 11: 14 bits bus formation

To allow the co-simulation with the physical FPGA board implementing the design, the Hardware in the Loop (HIL) block is added to the Simulink model as shown in Figure 12. A simple JTAG interface between Simulink and the FPGA board links the two environments. The HIL block also makes available to the hardware a large Simulink library of sinks and sources, such as noise generator, scope, and bit error rate measurement.

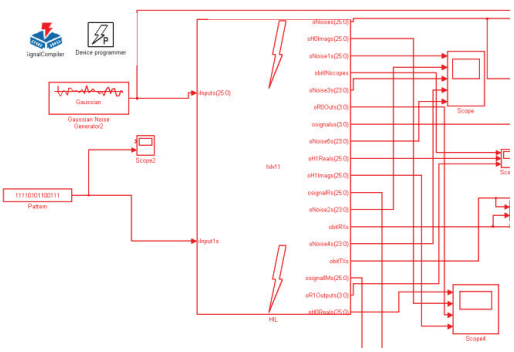

Figure 12: HIL co-simulation

IV. RESULTS

Figure 13 shows the BER performance comparison of the hardware implementation and the Simulink simulation. It is shown that the hardware BER, generally, fell within the confidence intervals of the simulated BER. The slight disagreement is likely due to the quantization process and the additional noise in the hardware systems.

In addition to the HIL based BER performance, a Logic Analyzer block called "Signal Tap II" is added to the design in order to verify the functionality of the hardware implementation. After a design is downloaded to the board, the Logic Analyzer can be used to capture the signal by inserting a "signal Tap II" block at the node of interest. The different captured signals are shown in Figure 14. Aside from a few errors and a delay representing the processing latency at both ends of the link, the STBC encoder output, XORIN, is identical to the rake receiver output, XOREND, and the ST decoder output, BitOut, is identical to the input bit stream, BitIn.

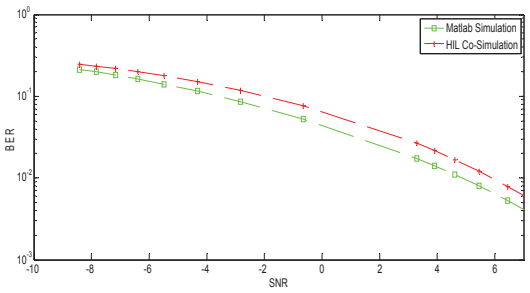

Figure 13: BER performance: Hardware implementation vs simulation

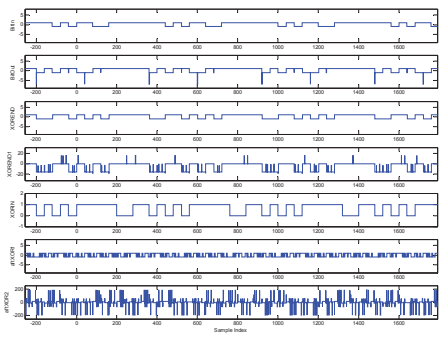

Figure 14: Signal Tap II results

V. CONCLUSION

Using a combination of Simulink and the Altera DSP Builder blocks a MIMO DS-CDMA system with Rake receiver was successfully converted from Simulink simulation to fully implemented digital hardware systems. It was noted that the Stratix EP1S80 DSP development board from Altera is a powerful and suitable developing tool for such tasks. The BER performance measurement within hardware was achieved using the HIL approach. Results show that, in general, the BER of the hardware implementation fell within the confidence intervals of the simulated BER. It was also shown that, based on the "Signal Tap II" Logic Analyzer, the captured transmitted and received signals looks highly correlated.

ACKNOWLEDGMENT

The author thanks the Canadian Microelectronics Corporation (CMC) for providing the prototyping station that includes the system-level design tools and the Stratix EP1S80.

REFERENCES

[1] Xiaoying, L., Fuming, S., and Enhua W. 2006. A Simulink-to-FPGA Co-Design of Encryption Module. *IEEE Asia Pacific Conference on Circuits and Systems*, 2008 – 2011.

[2] L. Bélanger, S. Roy, and T. Saïdi, Prototyping a MIMO W-CDMA system using a system-level approach , White Paper, Lyrtech Signal Processing, 2002.

[3] Ruan, J. Coulton, P., and Khirallah, C., " Using Simulink to Facilitate a Fixed-Point DSP Implementation of VBLAST Receiver," Proc. 4th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, pp. 251-253, 2003.

[4] M. Guillaud, A. Burg, L. Mailaender, B. Haller, M. Rupp, and E. Beck, " From Basic Concept to Real-Time Implementation: Prototyping WCDMA Downlink Receiver Algorithms – A Case Study," *Conf. on Signals, Systems, and Computers*, 2000.

[5] M. Hefnawi, and G. Gai, "Simulink Implementation of a CDMA MIMO-Beamforming System," IEEE International Symposium on Antenna Technology and Applied Electromagnetic, 2005.

[6] Xilinx 2008. System Generator for DSP, Release 10.1 Mar. 2008. www.xilinx.com.

[7] Altera Corporation 2011. DSP Builder User Guide. www.altera.com

[8] Hefnawi, M. and Gai, G. 2006. Adaptive Beamformer-Rake Receiver for Space-Time Coded DS-CDMA Systems. *IEEE International Symposium on Computational Intelligence and Intelligent Informatics.*

[9] Altera Corporation 2004. The Stratix EP1S80 data sheet, Dec. 2004, ver. 1.3. www.altera.com.

[10] M. S. Khairy, M. M. Abdallah, S. E.-D. Habib "Efficient FPGA Implementation of MIMO Decoder for Mobile WiMAX System," IEEE International Conference on Communications, pp. 14-18, 2009.

[11] P. J. Green and D. P. Taylor "FPGA Implementation of a Real Time Maximum Likelihood Space-Time Decoder on a MIMO Software Radio Test Platform," IEEE International Symposium on Electronic Design, Test & Applications, pp. 139-143, 2010.

[12] M. Vestias "Design and Test of a MIMO Receiver Based on the Alamouti Scheme in FPGA," IEEE International Conference on Consumer Electronics, pp. 107 – 111, 2012.

[13] Alamouti ,S. M.. A Simple Transmit Diversity Technique for Wireless Communications. *IEEE Journal on Selected Areas in Communications*. vol. 16, no. 8, pp. 1451-1458,1998.

[14] Tarokh, V. and Jafakhani, H., "Space-Time Block Coding for Wireless Communications: Performance results," IEEE Journal on Selected Areas in Communications, vol. 17, no. 13, 1999.

[15] T. S. Rappaport, *Wireless Communications Principles and Practice*, Prentice Hall, 2003.