Hardware Implementation of MIMO OFDMA Test Bed and its Application Towards Channel Characterization on Indoor Lab Test Environment

Sandip Das*, Suvra Sekhar Das*, Indrajit Chakrabarti†
*G. S. Sanyal School of Telecommunications,†Dept. of Electronics & Electrical Communication Engineering
*†Indian Institute of Technology Kharagpur, India.

***Abstract*—In this article, we present a hardware architecture of MIMO-OFDMA test bed. The testbed is designed in Field Programmable Gate Array (FPGA) in order to provide maximum flexibility in implementation. A modularized design approach is adopted which enables easy implementation, Integration, and testing algorithms of various physical layer modules of OFDMA. The entire physical layer design is successfully implemented in a single Virtex-4 FPGA processing core. The designed testbed architecture features up to 2x2 MIMO implementation with different MIMO modes and Frequency Domain Link Adaptation (FDLA). However, It is also discussed that the design can support up to 4x4 MIMO implementation depending on the choice of PHY design parameters such as FFT Size and FPGA processing frequency. The design is shown to be flexible and reconfigurable to work with different configurations using minimal PHY layer design alterations. The real-time over the air performance results are obtained using over-the-air transmission and reception of data traffic through the testbed. Finally, over-the-air performance results of the testbed are calibrated with the simulated performance over the SUI-3 channel. The implementation is performed on Wireless Open-Access Research Platform (WARP). LTE Release-10 parameters are adopted for designing the testbed.**

Keywords: Testbed, OFDMA, FPGA, Hardware Implementation, MIMO, Channel Characterization, LTE, SUI-3, real-time.

I. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO) are some of the promising techniques to improve data rate and reliability over wireless fading channels. Next-generation wireless technologies such as Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX) have adopted Orthogonal Frequency Division Multiple Access (OFDMA) in order to meet the exponentially growing data rate demand over the recent years [1]. OFDM, MIMO and Frequency Domain Link Adaptation (FDLA) are some of the important features of an OFDMA based systems.

OFDM is a multi-carrier modulation technique that provides better bandwidth utilization as compared to other traditional single carrier systems [2]. In OFDM, the large bank of oscillators can be replaced by an IFFT at the transmitter and FFT at the receiver for multicarrier modulation and demodulation respectively. Channel estimation and equalization are reasonably simple as the techniques of Least Square(LS) estimation and Zero Forcing (ZF) equalization work quite well. This technique reduces the hardware complexity in its real-time implementation.

978-1-5090-2361-5/16/$31.00

MIMO has been widely accepted as a promising technique to enhance the link capacity in OFDMA based systems. MIMO technology uses multiple antennas and intelligent signal processing technique at both transmitter and receiver to improve the capacity of the system [3]. However, the hardware realization of such systems requires large signal processing due to computation intensive algorithms which in turn consumes large resource and power. Moreover, as the order of the MIMO increases, the resource requirements also increases linearly. Obviously this produces a real challenge in real time implementation of MIMO-OFDMA systems.

There have been a significant amount of research performed over the recent years in real time implementation and verification of physical layer algorithms of OFDMA based systems. Test-beds allows easy implementation and testing of various physical layer algorithms. The real-time implementation of test bed also offers the opportunity to analyze the implementation issues, hardware complexity, and real-time performance. This procedure results in depth analysis of complexity vs. performance trade-offs. Therefore, there have been frequently reporting of real time or nonreal-time hardware test beds over decades. The nonreal-time test bed uses hardware for real-time transmission and reception of data that has been generated offline using software like MATLAB or similar environment. Examples of this kind of test bed include Viena MIMO testbed [4] which uses a combination of MATLAB and FPGA-based custom hardware, and Hydra testbed [5] which uses Ettus kits [6]. This type of test bed provides ease of implementation but lacks flexibility and in-depth complexity and performance analysis.

The real-time testbed uses hardware for both real-time processing of physical layer algorithms and real-time performance analysis using over the air transmission and reception using radios. There are many examples of this kind of testbeds ([7], [8], [9]) reported in the literature. However, these kinds of literature are largely concerned with algorithm and performance evaluation. Implementation aspects and complexity analysis are largely ignored in the literature. In [10] an efficient implementation of MIMO-OFDM test bed targeted for 802.11n Wireless LAN (WLAN) system is presented which provides in-depth analysis of implementation aspects

and complexity analysis. However, there are very few testbeds [11], which is targeted towards LTE systems and to the best of authors knowledge there is hardly any reporting of testbed with hardware implementation architecture and in-depth complexity analysis which adopts LTE specifications. In this paper, we present an efficient hardware architecture of real-time FPGA-based MIMO-OFDMA test bed targeted towards application in fourth generation systems.

Considering all the aspects listed above, a hardware realization of a flexible and reconfigurable architecture of MIMO-OFDM testbed has been proposed and implemented in FPGA. A modularized design approach is adopted for easy reconfiguration of the PHY layer of the test bed.

The rest of the paper is organized as follows. Section II presents the detailed design of MIMO-OFDM testbed. Section II-A presents the physical layer system model. Section III provides an overview of the testbed system implementation. Hardware platform used for real-time implementation of such module is presented in Section IV. Section V presents the hardware resource usage involved in the implementation. Results obtained by real time over the air performance of the implemented algorithms and channel characterization of indoor lab environment are presented in this section. Finally, conclusions are presented in section VI.

II. Design of MIMO-OFDMA Testbed

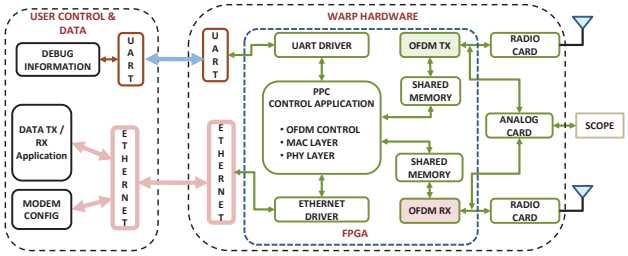

Fig. 1: Hardware System Architecture

Figure 1 gives a logical overview of the system respectively that has been implemented in WARP hardware. The testbed consists of custom OFDM transmitter (OFDM Tx) and receiver (OFDM Rx) chains implemented in FPGA. Power PC (PPC) embedded in the FPGA controls the parameters of PHY using 256 Kb of shared memory configured in the FPGA. A set of API functions is written in Power PC (PPC) to achieve data handshake between the application interface and firmware. A debug interface is developed using serial communication. Radio cards embedded on the board are used to transmit and receive RF signals over the air. The baseband signals before upconversion are passed to analog cards embedded on the board and viewed in Digital Storage Oscilloscope (DSO) for real-time monitoring of signals. Finally, a GUI based application is developed in MATLAB for sending and receiving data traffic and hardware configuration.

A. Physical Layer System Model

Figure 2 and Figure 3 presents the system model of The MIMO-OFDM test bed. Following the data flow model, the received signal for each antenna can be written as

$$r(n) = \exp\left(j\frac{2\pi n\varepsilon}{N}\right)\sum_{l=0}^{L-1} h(l)x(n-\theta-l) + z(n) \quad (1)$$

In equation 1, θ denotes the timing offset in multiple of sampling period T_s, and $\varepsilon = NT_s f_d$ denotes the CFO present in the received signal. f_d stands for the carrier frequency offset present in the received samples. $x(n)$ represents the time domain transmitted signal obtained from equation 2 using N-point DFT of input constellation points s(k). $h(l)$ denotes the l^{th} tap channel gain of an L-tap wireless fading channel. $z(n)$ is assumed to be zero mean complex-valued Gaussian noise.

$$x(n) = \sum_{k=0}^{N-1} s(k)\exp\frac{-j2\pi kn}{N},\ n\ =\ 0\ to\ N-1 \quad (2)$$

The objective of this work is to create a flexible and reconfigurable FPGA-based OFDM test bed. It is also important to realize the system within limited resources. Thus making an area efficient architecture is highly desired. In this work, the transceiver is made flexible enough to be configured up to 10 MHz of bandwidth with 4x4 MIMO configuration. Whereas all the analysis made corresponds to the case of 5 MHz bandwidth configuration.

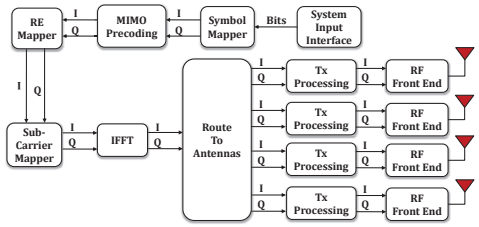

Fig. 2: System Model Transmitter

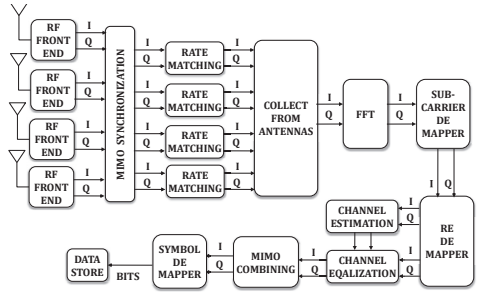

Fig. 3: System Model Receiver

1) Frame Structure: Figure 4 illustrates the time-frequency domain frame structure adopted in the design. Each transmission burst of 1 millisecond consists of 12 OFDM symbols in which first two OFDM symbols are identical preambles

dedicated for synchronization and CFO estimation and correction at the receiver. The reference signals are mapped in the subframe according to the figure 4. These reference signals are used at the receiver for channel gain estimation and equalization. In this figure, the reference signals distribution is shown for a two transmit antenna system.

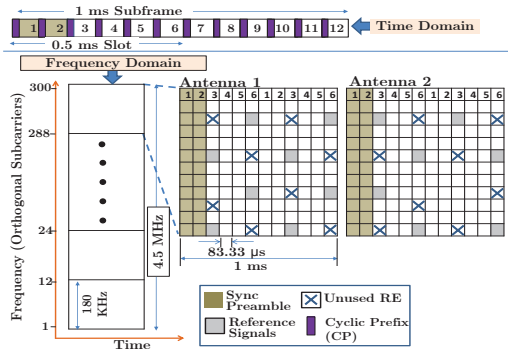

Fig. 4: Time-Frequency domain frame structure

III. Implementation Overview

In order to implement the design in FPGA with limited resources, A mixed serial and parallel architecture is adopted. In the transmitter side, we use a serial processing architecture for Symbol Mapper, MIMO Precoding, Resource Element Mapper (RE Mapper) and Subcarrier Mapper. Parallel architecture is used for Transmitter Processing, which resamples the data using the sampling frequency provided in the specifications (7.68 MHz for 5 MHz case). The module Route to Antennas takes the serial output from its previous block and routes data to a number of parallel MIMO branches for transmitter resampling process. Prior to this module, each module processes the data of all the MIMO branches serially. This type of processing is possible as the processing clock of the system is set to run at much higher frequency i.e. 40 MHz, nearly eight times faster than the sampling clock for the mentioned case.

At the Receiver side, the similar idea follows. After Synchronization and Rate Matching operates on parallel MIMO branches and resamples the data at processing clock of the system. Collect Form Antennas takes the data from parallel processing branches and merges into a single chain for further serial processing. All the successive blocks such as FFT, Subcarrier De-mapper, Resource Element De-mapper (RE De-mapper), Channel Estimation and Equalization, MIMO Combining and Symbol De-mapper processes the data of all the MIMO branches serially.

By adopting this type of architecture one most noticeable advantage that comes out, is the all the computationally intensive blocks (blocks those having very high logic and memory requirements) needs not to be replicated for processing of parallel MIMO branches. This architecture saves a significant amount of resources while implementing it on a platform with limited resources. The detailed algorithm and implementation of some of the important system components are discussed below.

A. FFT/IFFT

At the transmitter side, IFFT modulates the constellations s(n) using orthogonal subcarriers and adds CP of length N_{CP}. FFT at the receiver side demodulates the signal and removes CP. In this work, Xilinx's reconfigurable FFT & IFFT core is used in the design. In order to make this design area effective, pipelined processing of data for different MIMO branches is performed at the FFT and IFFT in a time division processing fashion. Hence, the number of parallel branches that can be processed using this architecture is limited by the burst size and the processing clock frequency as shown in the figure 5.

B. Synchronization

Synchronization is the first task that is performed at the receiver. It detects the start of the frame and compensates the effect of time and frequency offset present in the received packet. In this work, the synchronization is performed in the following steps. Firstly, the packet detection is performed using the ratio of two sliding window energy detection. The energy detection peak triggers the autocorrelation block that computes the ML estimates of symbol timing offset (STO) (or the beginning of the symbol ($\hat{\theta}_{BOS}$)) by maximizing the lagged autocorrelation of the received signal provided in equation 3.

$$R_{r,r}(n) = \sum_{m=0}^{W-1} r^*(n+m)r(n+m+P) \tag{3}$$

Since the first two OFDM symbols are identical preambles, in equation 3, the window size (W) as well as lag (P) is taken as $(N + N_{CP})$. Hence, the estimates of STO ($\hat{\theta}_{BOS}$) can be obtained as

$$\hat{\theta}_{BOS} = \underset{n}{\operatorname{argmax}}\{R_{r,r}(n)\} \tag{4}$$

Finally, the CFO estimate ε can be obtained using the autocorrelation peak according to equation 5.

$$\hat{\varepsilon} = \frac{1}{2\pi}\angle R_{r,r}(\theta_{BOS}) \tag{5}$$

C. Channel Estimation

In this work, a distributed pilot based channel estimation and equalization are used. The pilot distribution over the subframe depicted in Fig 4, is adopted from the LTE Release 10 specifications. The channel estimate over the entire subframe is obtained using the following phases, Firstly, the channel estimates $H_p(k_p, m_p)$ at the pilot subcarrier positions (k_p, m_p) are extracted using Least Square (LS) estimation. Then the channel estimates for all the subcarriers at pilot carrying OFDM symbol (m_p) is found using frequency domain interpolation provided in equation 6.

$$H(n, m_p) = \begin{cases} H(n-1, m_p) + \frac{\left(H_p(k_{p_{i+1}}, m_p) - H_p(k_{p_i}, m_p)\right)}{N_{gk}}, \\ \qquad\qquad \text{for } n = 1, 2, ...N, n \notin K_p \\ H_p(n, m_p), \quad \text{for } n \in K_p \end{cases} \tag{6}$$

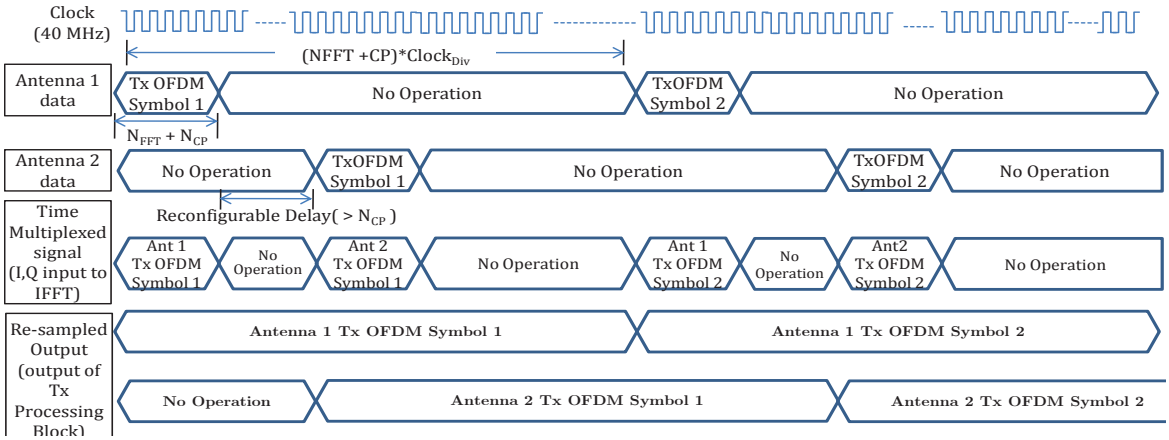

Fig. 5: Timing Diagram

In equation 6, $K_p = \{k_{p_1}, k_{p_2}, ...k_{p_K}\}$ represents the pilot subcarriers in m_p^{th} OFDM symbol. $i = floor(n/N_{gk}) + 1$ and $N_{gk} = k_{p_{i+1}} - k_{p_i}$ signifies the subcarrier gap between two consecutive pilot subcarriers.

Finally, time domain interpolation is applied to get the channel estimates in non-pilot carrying OFDM symbols. This produces the channel estimates over the entire subframe. The estimation and interpolation process is depicted in figure 6

$$H(n,m) = \begin{cases} H(n,m-1) + \frac{(H(n,m_{p+1}) - H(n,m_p))}{N_{gm}}, \\ \qquad \text{for } n = 1,2,...N, \quad \& \ m \notin M_p \\ H_p(n,m), \quad \text{for } n = 1,2,...N, \quad \& \ m \in M_p \end{cases} \tag{7}$$

In equation 7, $M_p = \{m_1, m_2, ...m_P\}$ represents the indices of pilot carrying OFDM symbol. $p = floor(m/N_{gm}) + 1$ and $N_{gm} = m_{i+1} - m_i$.

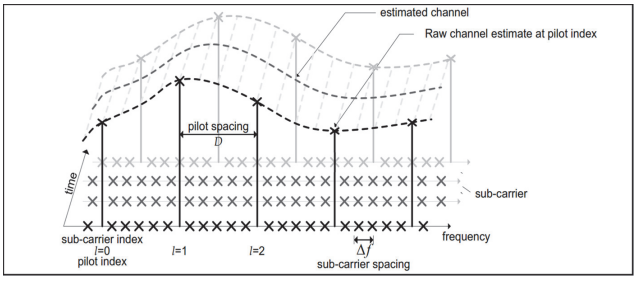

Fig. 6: Interpolation of channel estimates in time-frequency

D. MIMO Combining

MIMO combining plays an important role in efficient reception and interpretation of received signals. In this work, various types of MIMO combining is used at the receiver namely Zero Forcing (ZF), Maximum Ratio Combining (MRC), Space Frequency Block Codes (SFBC) and Spatial Multiplexing (SM).

E. Testbed Features

Table I presents the physical system parameters that are adopted while designing the system. The system is configurable up to 10 MHz of bandwidth. However, all the results presented in the subsequent sections are corresponding to 5 MHz bandwidth configuration as highlighted in the table. The test bed features different MIMO modes discussed in section III-D. The testbed also features Frequency Domain Link Adaptation (FDLA) which is discussed in [12].

TABLE I: LTE Physical Layer parameters

Access Scheme	OFDMA (Downlink)			
Channel Bandwidth (MHz)	1.4	3	5	10
Transmission Bandwidth (MHz)	1.08	2.7	4.5	9
IFFT (Tx) / FFT (Rx) Size	128	256	512	1024
Used Subcarrier (Nused)	72	180	300	600
Number of Resource blocks	6	15	25	50
Sampling Rate (Msps)	1.92	3.84	7.68	15.36
Modulation	QPSK, 16-QAM, 64-QAM			
No. of Symbols per subframe	12 (Extended CP)			
Cyclic Prefix length (μs)	16.6667 (l=0,1,2,.5), Extended CP			
Sub-frame / Slot Duration	1 ms Subframe / 0.5 ms Slot			
Error Correction Coding	Convolution Encoder, Viterbi Decoder			

IV. FPGA Design Platform

The WARP hardware [13] provides a platform for implementing and testing physical layer algorithm in real time as it is equipped with a Virtex 4 FPGA having 42 K logic slices, 160 Xilinx DSP-48 blocks and four radio cards which are capable of operating at 2.4 GHz frequency with a bandwidth of 40 MHz. It also provides the facility of varying transmitter receiver gains which are very helpful while testing the performance of any algorithm in real time.

WARP Design Flow:

Fig. 7 shows detailed design flow for the hardware implementation of the testbed. The design contains several phases of implementation.

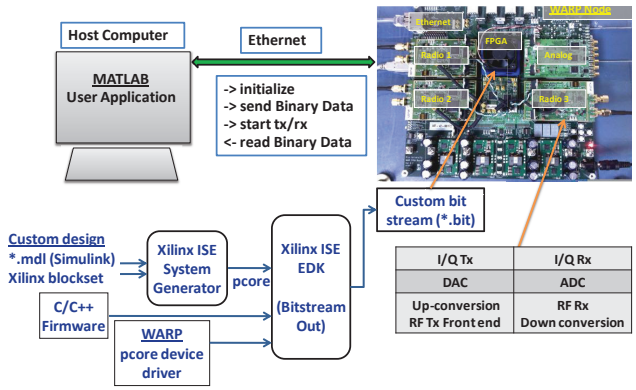

Fig. 7: Design Flow in WARP

- Physical layer processing is implemented in FPGA processing unit using the system generator tool provided by Xilinx. It uses simulink workspace for development environment and Xilinx library blocks are used for the RTL logic design. finally a IP core is generated upon compilation.
- The system generator IP-core is imported to Xilinx Embedded Development Kit (EDK) tool. Here the IP core generated in system generator is interfaced with other hardware specific driver IP-cores
- A firmware program is written into power PC which is a microprocessor, resides into the FPGA in order to control the physical layer module. This program is written in C++.
- An user application is written in MATLAB in order to send and/or receive data traffic to/from the hardware through ethernet interface.

V. Implementation Results

A. Hardware Resource Usage

TableII shows the detailed module level resource utilization of the implemented FPGA-based MIMO-OFDM. The use of shared memory blocks in the design enables configuring the PHY layer modules with different parameter from the firmware level as well as from application level. This implementation makes the testbed robust in nature by providing maximum flexibility in terms of reconfiguration of physical layer parameters. The channel estimates are read to third party workspace such as MATLAB for further processing. This two reason leads to certain increase in Block RAM (BRAM) in the design.

B. Performance Evaluation Results

C. Channel Characterization

The performance of the developed testbed is tested for its real-time performance in GSSST lab environment. The over the air performance results is calibrated with the simulated

TABLE II: Resource Utilization in Hardware (FPGA)

Hardware Logic Utilization					
Module Name	Slice Reg	LUT	LUT RAM	BRAM / FIFO 16	DSP 48
IFFT and CP Add	10055	9551	2066	3	4
MIMO Sync	7921	10607	735	74	29
FFT and CP remove	9833	9397	1993	3	4
Channel Estimation	2287	3486	274	60	3
MIMO Combining	3592	9463	294	2	18
Other Modules	5031	7021		18	28
Resource Available	84352	84352		376	160
Total Resource used	38719	54887		274	86
Percentage Usage	46%	65%		72%	53%

BER performance on Stanford University Interim (SUI) channel [14]. Six different SUI channel namely SUI1 - SUI6 are considered in this work. These are mainly rician channel model with different 'K' factor and varrying doppler to various User Equipment (UE) mobility. This is because under small cell environment such as macrocell and femtocell, the behavior of the channel is similar to rician fading with rounded Doppler spectrum.

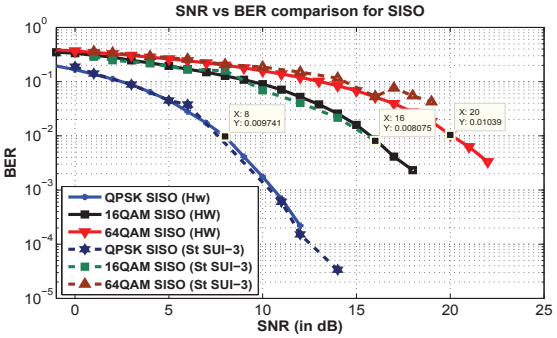

Fig. 8: BER performance for SISO in hardware (compared with simulated results for SUI-3 channel)

The six different realization of the SUI channel as described in [14], widely known as SUI1 - SUI6, are rician channel models with different 'K' factor and varying doppler to support for various UE mobility. The simulated BER performance in this channel is obtained by using Link Level Simulator (LLS) which is an intellectual property of GSSST. Simulation is carried out by calibrating parameters of LLS with the specifications used for designing the LTE transceiver and by replacing the channel part with standard SUI channel models. The simulation results obtained from LLS is calibrated with the one obtained from over the air performance the transceiver in LAB test environment. Figure 8 illustrates that the over the air performance results matches with the simulation results obtained for the SUI-3 channel. Figure 9 depicts the BER vs. SNR performance comparison for different MIMO modes. Table III gives the specifications of the SUI-3 channel used in this work. Figure 10 presents the experimental setup for the

demonstration and the performance evaluation of the test bed.

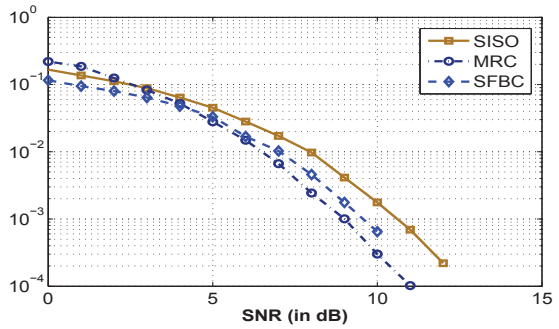

Fig. 9: BER performance comparison of SISO Vs. MRC in hardware

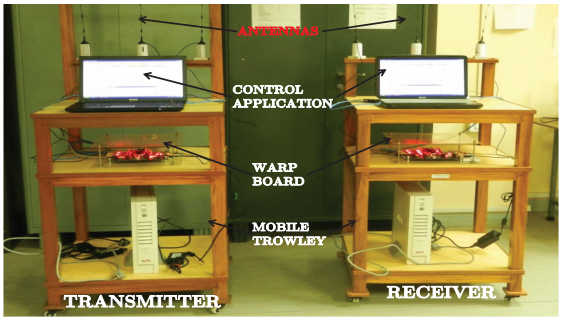

Fig. 10: Experimental Setup

From the calibration of results obtained by using LLS and over the air performance of transceiver, it can be inferred that the LAB test environment is similar to the channel models configured as SUI-3 standard channel models.

TABLE III: SUI -3 channel parameters

SUI - 3 Channel				
	Tap1	Tap2	Tap3	Units
Delay	0	0.5	1	μs
Power (omni ant.)	0	-5	-10	dB
K Factor (omni ant.)	1	0	0	
Power (30^{o}ant.)	0	-11	-22	dB
K Factor (30^{o}ant.)	3	0	0	
Doppler	0.4	0.4	0.4	Hz
Antenna Correlation:	ρ_{ENV} = 0.4			
Gain Reduction Factor:	GRF = 3 dB			
Normalization Factor:	F_{omni} = -1.5113 dB, F_{30^o} = -0.3573 dB			

VI. Conclusion

In this paper, hardware efficient architecture of MIMO-OFDMA test bed is presented. From the synthesis results it is clear that this architecture can be adopted to implement up to 4x4 MIMO given the parameters for 5 MHz LTE Release-10 specifications and at processing clock frequency of 40 MHz. The limitation on implementing a higher order of MIMO lies on the FFT size/bandwidth configuration and sampling frequency. In this work, up to 2x2 MIMO has been successfully tested with the different configuration such as SISO, MRC, SFBC, and SM. A modularized and robust design approach is adopted to provide maximum reconfigurability. The test bed is reconfigurable from the application layer by providing configuration parameters on the fly. It is shown that because of the robustness and reconfigurability property of the test bed, it uses a bit extra hardware resources. However, the entire test bed can still be accommodated in a single virtex4 FPGA chip. Finally, the test bed is used for channel characterization in indoor LAB test environment. It is found that the real-time performance results (BER vs. SNR)obtained from the over the air transmission and reception of data traffic through the test bed matches with the simulated performance in LLS using SUI-3 power delay profile. This successfully calibrates the testbed performance.

References

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast, 2013-2018," February 2014.

[2] M. Speth *et al.*, "Optimum receiver design for wireless broad-band systems using OFDM," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1668–1677, Nov 1999.

[3] A. Paulraj, D. GORE, R. Nabar, and H. Bolcskei, "An overview of mimo communications - a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, Feb 2004.

[4] S. Caban *et al.*, "Vienna MIMO Testbed," *EURASIP Journal on Applied Signal Processing,2006*, vol. 2006, pp. 1–13, 2006.

[5] Mandke, K. and others, "Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, April 2007, pp. 1896–1900.

[6] M. Ettus, "USRP Users and Developers Guide," *Ettus Research LLC*, 2005.

[7] van Zelst, V. and Schenk, T.C.W., "Implementation of a MIMO OFDM-based wireless LAN system," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 483–494, Feb 2004.

[8] H. Sampath, S. Talwar, J. Tellado, V. Erceg, and A. Paulraj, "A fourth-generation MIMO-OFDM broadband wireless system: design, performance, and field trial results," *IEEE Communications Magazine*, vol. 40, no. 9, pp. 143–149, Sep 2002.

[9] C. Dubuc, D. Starks, T. Creasy, and Y. Hou, "A MIMO-OFDM prototype for next-generation wireless WANs," *Communications Magazine, IEEE*, vol. 42, no. 12, pp. 82–87, Dec 2004.

[10] S. Haene, D. Perels, and A. Burg, "A Real-Time 4-Stream MIMO-OFDM Transceiver: System Design, FPGA Implementation, and Characterization," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, pp. 877–889, August 2008.

[11] D. Bates, S. Henriksen, B. Ninness, and S. R. Weller, "A 4 X 4 FPGA-based wireless testbed for LTE applications," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, Sept 2008, pp. 1–5.

[12] S. Das, I. Chakrabarti, and S. Das, "Hardware implementation of Frequency Domain Link Adaptation for OFDMA based systems," in *Twenty First National Conference on Communications (NCC), 2015*, Feb 2015, pp. 1–6.

[13] Rice University, "Wireless Open-Access Research Platform (WARP)," 2015.

[14] IEEE 802.16 Broadband Wireless Access Working Group, "Channel Models for Fixed Wireless Applications," july 2001.