A Novel Hardware Efficient Design for IEEE 802.11ax Compliant OFDMA Transceiver

Muhammad Aslam*, Xianjun Jiao, Wei Liu, Michael Mehari, Thijs Havinga, and Ingrid Moerman

***Abstract*— The introduction of Multi User (MU) communication in IEEE 802.11ax in the frequency domain via MU Orthogonal Frequency Division Multiple Access (MU-OFDMA) and in the spatial domain via MU Multiple Input Mulitple Output (MU-MIMO) enables the Access Point (AP) to serve up to 128 Stations (STAs) in a schedule. However, the MU functionality poses new challenges in the chip design. Existing MU Transceivers (TRXs) rely on the duplication approach wherein a dedicated hardware is utilized per user. The hardware footprint of such an approach increases proportionally with the number of users simultaneously served in the MU TRX. This paper introduces a novel hardware efficient design for an MU TRX. Unlike the duplication approach, the proposed design for the MU TRX has comparable hardware footprint of a Single User (SU) TRX regardless of the number of users being served, thanks to the hardware virtualization technique. The applicability of the design is initially validated for IEEE 802.11ax compliant MU-OFDMA transmitter on an FPGA of a modern SDR. The performance and hardware consumption is compared against the conventional duplication approach. The proof-of-concept implementation focuses on 20MHz, where maximum 9 STAs can be involved. Though we believe the design is extendable to support the maximum number of STAs in MU-OFDMA for IEEE 802.11ax standard. The experimental results show that the hardware virtualization based MU-OFDMA transmitter provides the same performance and consumes less than 13% hardware resources in comparison with the conventional duplication approach.**

***Index Terms*— FPGA, Hardware Virtualization, IEEE 802.11ax, Multiple-User Communication, OFDMA**

I. Introduction

IEEE802.11 standard has been constantly evolving to address the ever increasing demands [1]. The evolution of the earlier versions of the standard (i.e. IEEE 802.11a/b/g/n/ac) was mainly enhancing the data rate by exploiting many state-of-the-art technologies [2]. For instance, IEEE 802.11a/g increased the data rate to 54 Mbps by adopting Orthogonal Frequency Division Multiplexing (OFDM) technology. OFDM is a modulation technique used to mitigate inter symbol interference caused by multipath propagation and narrowband interference. IEEE 802.11n and IEEE 802.11ac achieved data rates till 600 Mbps and 6.9 Gbps, respectively, primarily by introducing Multiple Input Multiple Output (MIMO), increasing channel bandwidth and enhancing modulation and coding rates. MIMO is an antenna technology used to enhance either the throughput through spatial multiplexing or the reliability through spatial diversity. All of these standards however can only serve a Single User (SU) at a time, meaning that the wireless medium can only be occupied by a single Station (STA) either transmitting to or receiving from an Access Point (AP). Only when IEEE 802.11ac introduced the Down Link (DL) Multi User (MU) MIMO in the second wave, an AP was able to transmit to multiple STAs simultaneously.

The IEEE 802.11ax standard [3] (also known as High Efficiency (HE) WLAN), for the first time, introduced both DL and Up Link (UL) MU transmission. The main objective of the standard is to not only increase the data rate, but also improve the spectral efficiency especially in high-density public environments, like train stations, stadiums and airports. The key enablers for the MU features are Orthogonal Frequency Division Multiple Access (OFDMA) and MU-MIMO. OFDMA is the MU version of OFDM which allows simultaneous MU transmission in the same frequency band by assigning subsets of Subcarriers (SCs) named Resource Units (RU) to individual users. Though MU communication is recently introduced in IEEE 802.11ax, it already exists in other standards such as IEEE 802.16 (WiMAX), LTE, 5G NR [4], [5]. On the one hand, the MU communication improves the Wi-Fi performance in spectrum and spatial domains by exploiting MU-OFDMA and MU-MIMO, respectively. On the other hand, it also requires a complex Hardware (HW) design for the radio Transceiver (TRX) to implement these sophisticated techniques. In other words, more HW resources are required to implement such a complex HW design. Existing approaches [6], [7] employ a Hardware Duplication (HD) approach to design the Physical (PHY) layer of MU TRX (i.e., dedicated HW resources are assigned for each user). With such approaches, the consumption of the HW resources increases proportionally with the number of users in MU TRX. Consequently, a MU TRX realised by an HD approach occupies a larger die area of a chip, making it more expensive. The amount of resources consumed in an HD approach can be formulated using (1),

$$HW_{MU_{HD}} = N_{users} \times HW_{SU} \tag{1}$$

where, $HW_{MU_{HD}}$ reflects the total amount of HW consumed by the PHY layer of an MU TRX using HD approach, N_{users} denotes the number of users and HW_{SU} represents

*Corresponding author. Email: Muhammad.Aslam@UGent.be.

Authors are with IDLab, Department of Information Technology at Ghent University - imec, Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium. Email: Firstname.Lastname@UGent.be.

This research was funded by the Flemish FWO SBO S003921N VERI-END.com (Verifiable and elastic end-to-end communication infrastructures for private professional environments) project and the Flemish Government under the ”Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” program.

the HW consumed by the PHY layer of an SU TRX. In the worst case, MU-OFDMA TRX in IEEE 802.11ax supports up to 74 simultaneous users, meaning that an HD approach will consume 74 × HW_{SU} resources. For this reason, an alternative, more hardware-efficient design for MU TRX is highly desired, which is the focus of this paper.

Ideally, a HW efficient design should use the same HW_{SU} irrespective to the number of users in an MU TRX, which can potentially be achieved by resource sharing among different users. However, resource sharing in this circumstance is more challenging than sharing a processor amongst multiple software tasks. Because there is a tight timing constraint. Taking the Transmitter (TX) as an example, once the preamble of a packet is sent, the waveform needs to be continuous. One can of course generate the waveform of the entire packet up in front, however in real life situation, this means packets will be queued longer and the overall throughput would drop. Generating packet on the fly is hence the preferred approach. For OFDMA based MU TX, it means the OFDM symbol with all users' data allocated in different ranges of subcarriers, needs to be generated in a symbol time, which is less than 16 us for IEEE 802.11ax. Due to this stringent timing constraint, the HW resources for a SU cannot be shared with other users as a whole. Instead, the sharing needs to happen at much finer granularity with carefully designed multi-stage pipelining. Essentially we aim to let the HW for a SU appears to be serving MU simultaneously, which coins the technology named Hardware Virtualization (HV). We define HV as a HW resource sharing technique in which multiple logical instances are created from a single physical instance implemented on HW. It is worth mentioning that the hardware virtualization here is different than the term "hardware virtualization" used in cloud computing, where multiple virtual machines or containers run on a single physical machine.

HV however requires a Context Switching (CS) mechanism that is responsible for storing contexts or states of the current logical instance and restoring the stored contexts or states of the next logical instance upon switching between subsequent logical instances. Thus, HV based HW design for MU TRX need extra HW resources to perform CS in addition to HW_{SU}. The required HW resources in an MU TRX using the HV approach can be formulated using (2),

$$HW_{MU_{HV}} = HW_{SU} + HV_{overhead} \tag{2}$$

here, $HW_{MU_{HV}}$ reflects the total amount of HW consumed by the PHY layer of an MU TRX using HV approach, $HV_{overhead}$ represents the extra HW required to perform CS. The HW_{SU} is constant in (2), whereas $HV_{overhead}$ is dependent on N_{users}. In order to make HV benefitial, it is crucial to keep the $HV_{overhead}$ significantly smaller than the HW_{SU}, so that the overall impact of the increment of $HV_{overhead}$ on $HW_{MU_{HV}}$ is negligible.

In this work, we propose a novel HV based MU TRX design. In order to validate such a design experimentally, *openwifi* [8]—an existing open-source Wi-Fi implementation on modern Software-Defined Radio (SDR) platforms—is modified. An SDR is a radio communication system wherein radio components are either running on a general purpose processor (e.g., host computer) or implemented on a programmable hardware (e.g., Field Programmable Gate Array (FPGA)), in both cases the key functionalities of radio communication become programmable. Modern SDRs do not only offer flexibility to speed up the validation process, but also support high processing capabilities, which is why we selected SDR as the validation platform. The key contributions of this work are summarized below:

1) A high-level architecture of MU transceiver (TRX) for OFDMA and MU-MIMO is presented, where we indicate how the HV technique can be used to keep the MU HW footprint close to HW_{SU} irrespective to N_{users}.
2) A generic methodology for applying the HV technique to realise the MU TRX hardware design is described.
3) A detailed design of an IEEE 802.11ax compliant MU-OFDMA TX with multi-stage pipelining and dedicated state machines is described for meeting the stringent timing constraints.
4) The applicability of the design is experimentally validated on the FPGA of a modern SDR with quantitative analysis in terms of performance and hardware footprint.

The rest of the paper is organized as follows. Section II details the state-of-the-art work. The proposed design for the MU TRX is elaborated in Section III. Experimental validation is performed in Section IV. Lastly, conclusions and future work are discussed in Section V.

II. State of the Art

This section presents the state-of-the-art of radio TRXs, focusing on the realization of MIMO and OFDMA, where the HD approach is most commonly employed.

A. SU/MU-MIMO OFDM Transceiver

An IEEE 802.11n compliant 4×4 SU-MIMO OFDM TRX is implemented in Synopsis SAED90nm ASIC [9]. Although the SU-MIMO OFDM TRX serves a SU at a time, it is capable of achieving a throughput of 600 Mbps by transmitting 4 parallel streams in the spatial domain to a SU. The implementation uses 4 dedicated (De)Interleavers, (I)FFT and Guard Interval (GI) modules for each spatial stream, increasing the overall chip area to 13 mm^2. The authors in [9] indeed admit that more optimized approaches are needed to reduce the chip area. Another IEEE 802.11a based 4×4 SU-MIMO OFDM TRX is prototyped on an ASIC in [10]. The authors also addressed the silicon complexity of SU-MIMO OFDM and mentioned that the chip area is increased by a factor of 6.5 when going from SISO to SU-MIMO design using HD approach.

R. Carlos et al. [11] provide an important guide to implement massive MIMO for 5G wireless communication systems by presenting 8×8 MU-MIMO OFDM TRX. The work has significantly reduced the hardware footprint of the final design by reusing the same (I)FFT and GI modules for 8 streams, it however utilizes a dedicated (de)modulator (i.e., mapper, pilot & virtual carrier inserter) for each stream.

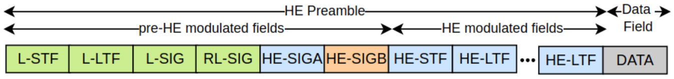


Fig. 1. The general structure of HE PPDUs in 802.11ax.

B. MU-OFDMA Transceiver

A virtex5 FPGA based Group Orthogonal (GO) OFDMA prototype for high speed variable bit rate broadcast services is presented in [12]. GO-OFDMA is a variant of OFDMA, optimised to reduce the Peak to Average Power Ratio (PAPR). In GO-OFDMA, a subset of OFDM SCs are assigned to different users. The FPGA based prototype of the GO-OFDMA in [12] consists of 32 points (I)FFT, of which the first 16 SCs are assigned to one user, and the remaining 16 SCs are equally divided between two other users. The corresponding data stream of each user is modulated, spread using an orthogonal code, and interleaved before reaching to IFFT. The processing module before the IFFT module is referred to as a HW chain, which is dedicated for each user, resulting a significant increase in HW resource usage as the number of users rise. Carrier Interferometry (CI) GO-OFDMA [13] is an improved version of GO-OFDMA, providing better performance in terms of PAPR and Packet Error Rate (PER). The virtex5 FPGA based CI-GO-OFDMA prototype is again using the HD approach.

C. Summary

In the prior arts concerning parallel data transfer in WLAN, the SU/MU-MIMO or MU-OFDMA are studied using simulation models or prototyped on FPGA/ASIC using the HD approach. The simulation work [14]–[17] helps to grasp the key concepts of an MU TRX, but is confined to non-real-time performance and does not discuss the hardware complexity needed to implement such kind of solutions.

We observe that existing HW implementations of the MU TRX heavily rely on the HD approach. As shown by (1), the HW utilization increases proportionally with the number of users in an MU TRX. Consequently, the MU TRX realised by a HD approach are less economical, as they lead to larger chip area in the final product. In this paper, HV approach is made use to improve the HW utilization efficiency in an MU TRX. The challenge is to keep the HW utilization of an MU TRX as close as possible to the hardware footprint of a SU TRX, at the same time the design must meet the real-time constraint, such as producing samples fast enough to keep the waveform continuous. To the best of our knowledge, this challenge is not yet addressed in the literature.

III. The Proposed Hardware Design for MU Transceiver

In this section, first, the PHY layer Protocol Data Unit (PPDU) formats and the OFDMA feature of IEEE 802.11ax in 20 MHz are briefly introduced. Next, the procedure for designing a MU TRX using HV technique is described. Finally, an IEEE 802.11ax compliant MU-OFDMA TX is implemented on a modern SDR platform using the proposed HV approach. Its performance and HW footprint are compared quantitatively against the conventional HD implementation.

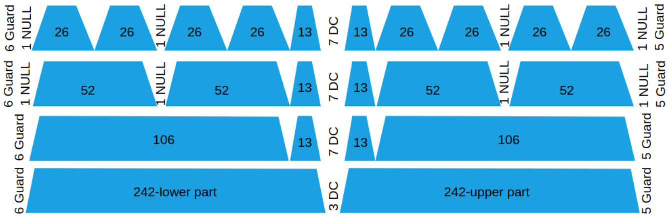


Fig. 2. IEEE 802.11ax Resource Units (RU) allocation in 20MHz.

A. 802.11ax PPDU Formats

IEEE 802.11ax supports four types of PPDU: HE SU PPDU, HE Extended Range (ER) SU PPDU, HE MU PPDU, and HE Triggered Based (TB) PPDU. HE SU PPDU and ER SU PPDU are used for UL/DL transmission to a single user, while HE ER SU PPDU is intended for outdoor scenarios. The remaining two types of *HE* PPDU are targeting MU communication, among which HE MU PPDU is for DL MU transmission and HE TB PPDU is for UL MU transmission. Fig. 1 shows the structure of all the PPDUs. Each HE PPDU consists of an HE preamble and an HE data part. The HE preamble is comprised of pre-HE modulated and HE modulated fields. The pre-HE modulated field is further composed of Legacy Short Training Field(L-STF), Legacy Long Training Field (L-LTF), Legacy Signal (L-SIG), Repeated L-SIG (RL-SIG), HE Signal A (HE-SIGA) and optionally HE Signal B (HE-SIGB). The HE modulated field further consists of HE Short Training Field (HE-STF) and HE Long Training Field (HE-LTF). All of these sub-fields in HE preamble part serve specific purposes defined in the IEEE 802.11ax standard. Note that the HE-SIGB sub-field only exists in HE MU PPDU. For the HE modulated preamble and data fields, IEEE 802.11ax uses a 256-point Fast Fourier Transform (FFT) to generate an OFDM symbol with a symbol duration of 12.8 μs and Guard Interval (GI) options of 0.8 μs, 1.6 μs and 3.2 μs. The pre-HE modulated fields are modulated using a 64-point FFT to form an OFDM symbol with a symbol duration of 3.2 μs and GI of 0.8 μs. The OFDM symbol duration including GI defines the time limit in which the samples containing MU's data must be generated.

B. OFDMA in 802.11ax

OFDMA is an OFDM-based multi-access technique, which allocates subsets of SCs (or tones) to different users, allowing simultaneous MU data transmission in the same OFDM symbol. In IEEE 802.11ax, each group of these SCs is referred to as a Resource Unit (RU). There are 26-tone RU, 52-tone RU, 106-tone RU and 242-tone RU in the 20 MHz channel. Each RU consists of pilot SCs, data SCs, and unused SCs. While data SCs carry the actual data, pilot SCs are the known modulated symbols used to perform channel estimation by the receiver. The purpose of the unused SCs is to avoid interference from adjacent channels or to reduce the leakage from adjacent RUs within the channel. Fig. 2 shows the RU allocation scheme in the 20 MHz channel. For instance,

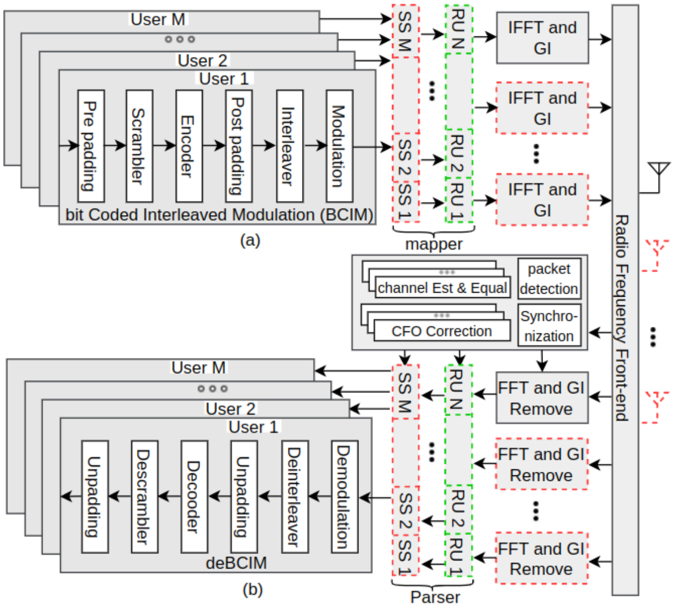


Fig. 3. The block diagram of an MU-OFDMA TRX (excluding red dotted modules), MU-MIMO TRX (excluding green dotted modules) and MU-OFDMA MIMO TRX for IEEE 802.11ax standard. Here, SS, RU and GI denote spatial stream, resource unit and guard interval respectively.

IEEE 802.11ax can simultaneously serve up to 9 users by using all 26-tone RUs in 20 MHz channel. OFDMA makes use of HE MU PPDU and HE TB PPDU for DL MU and UL MU transmissions, respectively. Note that all the RUs in a MU transmission have the same time allocation; i.e., the transmission in each RU ends at the same time. In a DL MU transmission, this is achieved by adding padding bits to the shorter packets. In UL MU transmission, the AP informs the STAs the packet length to be transmitted via the trigger frame.

C. Proposed Hardware Design for the MU Transceiver

The procedure to apply HV technique on an MU TRX primarily consists of three steps:

- Draw the modular design of an MU TRX.
- Identify modules for which the HW footprint increases with the number of users in the MU TRX.
- Apply HV technique on the identified modules to keep the HW footprint as close as possible to SU TRX.

Fig. 3 shows the block diagram of the PHY layer of MU TRX for IEEE 802.11ax standard. The Fig. 3-a (excluding modules in red dotted line) illustrates the modules constituting an MU-OFDMA TX. The first module on the TX side, Bit Coded and Interleaved Modulation (BCIM), performs padding (optionally), scrambling, encoding, interleaving and modulation on the data received from higher layers for each user. Next, the frequency mapping module maps users' data to different RUs, which are subsequently combined to form an OFDM symbol in the Frequency Domain (FD). Thereafter, Inverse Fast Fourier Transform (IFFT) is performed to convert it into Time Domain (TD) signal. Lastly, Guard Interval (GI) is appended to each OFDM symbol, the symbols are sequentially transmitted over the air via the Radio Frequency (RF) front-end. It is apparent that the number of BCIM module increases with the number of users in the MU TX, hence the BCIM module is where HV can be applied for a MU-OFDMA TX.

In MU-MIMO, BCIM also increases proportionally to the number of user, hence the same technique can also be applied for MU-MIMO. Fig. 3-a (excluding modules in green dotted line) illustrates the modules constituting the PHY layer of an MU-MIMO TX. Note that a single Spatial Stream (SS) is shown per user in Fig. 3-a, for simplicity. Though there is a significant difference between MU-MIMO and MU-OFDMA when applying HV on the BICM module. In MU-MIMO, the capacity of the system increases with the number of users, whereas in MU-OFDMA, the total capacity is fixed, each user only gets a share of the given capacity. Due to this difference, there is a need to increase the system clock speed proportionally to the number of users when applying HV on MU-MIMO TX. Overclocking is however not needed for MU-OFDMA. In addition to the BCIM module, IFFT and GI modules could also benefit from HV in the MU-MIMO TX. This is because for MU-MIMO, each user requires a different signal in the TD, whereas for MU-OFDMA a common TD signal is shared by all users.

The majority of the MU RX is the reverse process of MU TX. Similar to BCIM in MU TX, the deBCIM in MU RX increases with the number of users, hence HV technique should be applied on the deBCIM module. Additionally, the MU RX requires synchronization, Carrier Frequency Offset (CFO) correction, channel estimation and channel equalization to mitigate the distortion caused by the channel impairments and mismatch in carrier frequencies at the TX and RX. The synchronization module serves to find the beginning of a packet, it is shared by all users, whereas CFO, channel estimation and equalization modules increase with the number of users in an MU RX. Therefore, HV technique should also be applied on these modules.

Till this point, we present the complete TRX design for both MU-OFDMA and MU-MIMO, and identify the modules requiring HV. As the procedure to apply HV is generic, it suffices to validate the applicability of the HV based design on one part of the TRX. In the subsequent section, we therefore focus on the OFDMA based MU TX: first for benchmarking purposes, the MU TX is designed using the conventional HD approach; next, the design is improved using the HV technique; Finally, the two design approaches, HD and HV, are rigorously compared in terms of HW utilization.

D. The MU-OFDMA TX using Hardware Duplication

Fig. 4-a illustrates the complete protocol stack and functional blocks of IEEE 802.11ax for MU-OFDMA TX. The MU-OFDMA TX makes use of a modified version of *openwifi*. *openwifi* offers lower MAC layer, radio driver, and IEEE 802.11a/g/n compliant PHY layer, but relies on Linux OS for the upper layer communication stack, i.e., application, transport, networking and upper MAC layers. Given that the current Linux kernel running on *openwifi* has no support for IEEE 802.11ax, the packet injection mode is then used to inject the MAC Payload Data Unit (MPDU) into the radio driver via

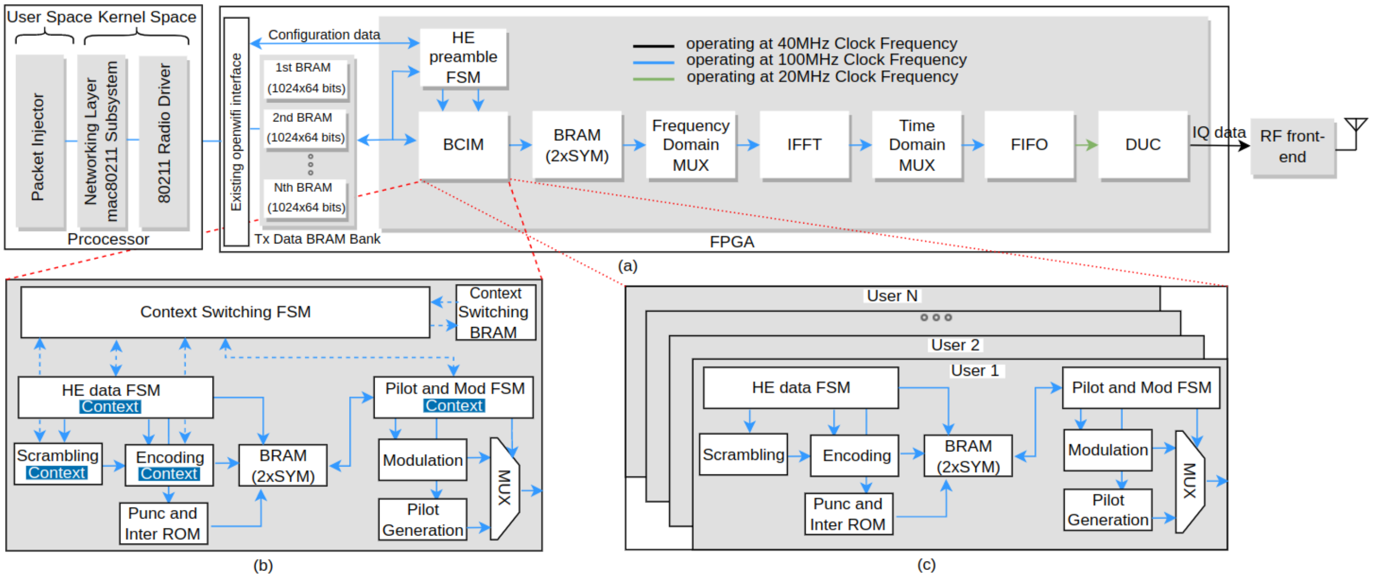
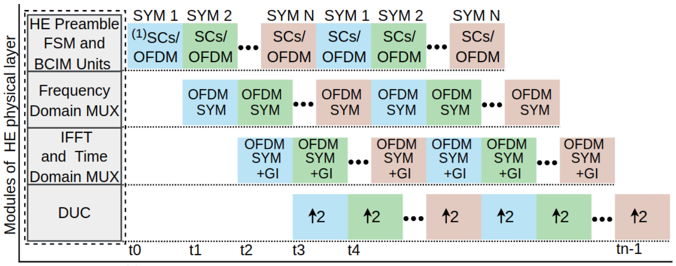


Fig. 4. The proposed design for (a) MU-OFDMA transmitter using (b) hardware virtualization and (c) hardware duplication approaches.

mac80211 subsystem, the rest of communication stack in the Linux OS is bypassed. The radio driver sends the MPDU to the PHY in FPGA via the existing interfaces in *openwifi*. The PPDU are then formed by the PHY layer, the type of the PPDU (i.e., HE SU, MU, TB PPDU) is configured via the driver. Since HE ER PPDU is a variant of SU PPDU, it is left out of this work for simplicity. Finally, the PHY layer transmits packets and updates the radio driver about the result of each transmission via interrupts. The following steps are needed to compose a PPDU and send it over the air, which is also illustrated in Fig. 4:

1) The HE preamble Finite State Machine (FSM) module reads data from the data Block Random Access Memory (BRAM) bank. The data BRAM bank has N numbers of BRAMs with each BRAM dedicated for storing data of a single user. The size of each BRAM is 1024×64 bits, i.e., each BRAM has 1024 locations with each location has a data width of 64 bits. The first few locations of each BRAM contains the configuration needed to form the HE preamble part including the supported coding type, RU size, HE Modulation Coding Scheme (MCS) value, packet length for each RU, GI value, HE-LTF types. Based on these configurations, the FSM formulates L-SIG, RL-SIG, HE-SIGA, HE-SIGB, HE-STF, and HE-LTF fields.
2) Next, BCIM module performs scrambling, encoding, interleaving, and modulation for the PPDU headers. Note that only the BCIM module of the first user is utilized to generate the PPDU header, as it is common for all users (see User 1 BCIM module in Fig. 4-c). In addition, HE-STF, HE-LTF fields skip the BCIM modules, as their modulated version in the frequency domain is stored in the Look Up Tables (LUTs) of FPGA. Till this point, the modulated SCs of the HE preamble part is generated, which is common for all users.
3) For the data part, the HE data FSM in the BCIM module reads the user specific data from the data BRAM bank. Then, the scrambling, encoding, interleaving, and modulation steps are performed. As packets' size can be different between users and MU transmission of all users end at the same time, HE data FSM is also responsible for padding if a user's packet cannot occupy all the OFDM symbols. Unlike the HE preamble, a dedicated BCIM module is used for each user's data, hence the number of BCIM modules used depends on how many users are in a MU transmission (see Fig. 4-c).
4) Next, Frequency Domain Multiplexer (FD-MUX) combines the modulated SCs (i.e., either from the common HE preamble part or from HE data part of different users), and unused SCs to generate a 64-point or 256-point OFDM symbol in the FD, depending on whether the symbol belongs to the pre-HE modulated fields.
5) Finally, the IFFT module generates OFDM symbols in the Time Domain (TD). The TD-MUX appends a GI for each OFDM symbol, and the Digital Up Converter (DUC) upsamples and transmits the signal over the air via the RF front-end. In our design, TD samples of L-STF and L-LTF fields of the pre-HE preamble part are stored in LUTs of the FPGA. The TD-MUX reads these LUTs first, and then uses samples generated by IFFT to form a complete HE PPDU packet.
6) Upon transmission of all the OFDM symbols, the PHY layer generates an interrupt for the radio driver and waits for the next packet.

All the modules in the PHY layer are pipelined. A high level illustration of the pipelining is shown in Fig. 5, wherein the horizontal and vertical axis represent the time scale and modules of the PHY layer, respectively. At $t0$ time, HE preamble and BCIM modules begin processing the first OFDM symbol. After a tick, these modules start processing the second OFDM symbol and FD-MUX begins processing the first



(1)SCs/OFDM represents data and pilot subcarriers per OFDM symbol

Fig. 5. Applying pipelining at module level.

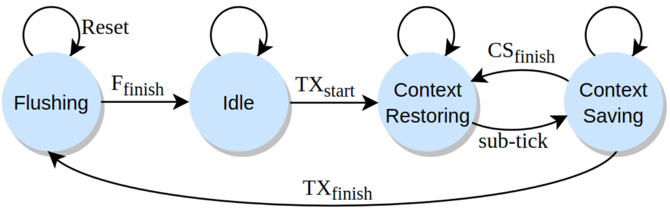
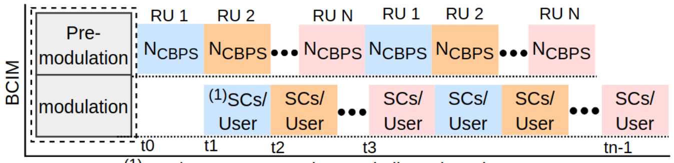


Fig. 6. State diagram of finite state machine used for context switching.



(1)SCs/RU represents data and pilot subcarriers per User

Fig. 7. Applying pipelining at sub-module level.

OFDM symbol. The tick is defined as a time unit required to process the Number of coded bits per OFDM symbol ($OFDM_{CBPS}$), which is equal to $\sum_{u=1}^{N_{users}} N_{CBPS}(u)$. In other words, N_{users} BCIM physical modules (in case of HD approach) or logical modules (in-case of HV approach) are required to generate a single $OFDM_{CBPS}$. The value of $N_{CBPS}(u)$ depends on the MCS value and RU size for each user. After the third tick (i.e., $t3$), all the modules are busy in processing the consecutive OFDM symbols. The BRAMs (one between the interleaver and modulation submodules of BCIM module, and the other between the BCIM and FD-MUX) have the capacity to store 2 OFDM symbols and behaves as a ping pong buffer. That is, the BRAMs can accept a new OFDM symbols while sending out the previous OFDM symbol. The pipeline based design does not only improve the latency by parallel running the modules, but it also allows to increase the speed of system clock at which the baseband processing is performed, which further increases the processing throughput.

E. The MU-OFDMA TX using Hardware Virtualization

The HV approach based MU-OFDMA TX uses the same communication stack and functional blocks shown in Fig. 4-a except for the BCIM module. Unlike the HD approach where a dedicated BCIM module is utilized for each user, the HV approach utilizes the same BCIM module for multiple users. This is achieved by using multitasking, pipelining and multi-clock domains. This subsection addresses only the BCIM module, as the remaining modules do not need to be duplicated, hence are already fully discussed in the previous subsection.

1) Multitasking: Multitasking commonly used in the field of Operating System (OS) enables OS to execute multiple tasks at a time by rapid context switching among tasks. During each Context Switch (CS), the contexts or states of the current task are saved and the states of the next task are restored, requiring extra HW or software resources to perform saving-restoring operations. We aim to apply the multitasking concept to utilize the same BCIM module for multiple users. However, it also inherits the context saving-restoring problem. Like a dedicated program and memory unit used in OS to perform context saving-restoring operation, a dedicated CS-FSM and an extra CS-BRAM is introduced in the proposed design (see Fig. 4-b).

The first step to design the CS-FSM is to identify the submodules in the BCIM module which need context saving-restoring operations. In principle, any submodule which has memory, delay, internal states or pipelines demands a CS. Four modules, which are HE data FSM (due to internal state), Pilot and Mod FSM (due to internal state), scrambling (due to memory), and encoding (due to memory) modules in Fig. 4-b, in the proposed design need context saving for the current user and restoring for the next user during each CS operation.

The state diagram for CS-FSM is drawn in Fig. 6. At the beginning of a packet transmission, the CS-FSM enters the $Flusing$ state in which CS-BRAM is reset. After $N_{users}\times$ 4 clocks, it enters into $Idle$ state and waits for the TX_{start} signal issued by the low MAC. In the CS-BRAM, 4 memory locations are assigned to each user, with each location being 64-bits wide and requires one clock's duration to perform read/write operations. These locations are used to store the internal states of HE data FSM, data scrambler, convolutional encoder, and the FSM used to insert pilots and perform modulation. When the TX_{start} is issued, the CS-FSM instructs the BCIM to begin the data processing and then enters the $ContextRestoring$ state. After a sub-tick, CS-FSM restores the states of the next user and enters into the $ContextSaving$ state. The sub-tick is defined as a time required to process $N_{CBPS}(u)$. In the $ContextSaving$ state, the CS-FSM stores the states of the current user to be used later. After four clock cycles, the CS-FSM generates CS_{finish} signal indicating that the CS is finished and re-enters into $ContextRestoring$ state. The CS-FSM also checks whether an iteration has been performed for all RUs in an OFDM symbol. If this condition is met, it increments the generated number of OFDM symbols, which is common for all users. Once all the OFDM symbols are generated, the CS-FSM generates TX_{finish} signal and goes back to the $Flusing$ state. The FSM always has four states, irrespective to the number of users in MU-OFDMA TX. Thus, the FSM design offers high efficiency in terms of hardware utilization.

2) Pipelining: It is crucial that we iterate through all users in an OFDM symbol fast enough, so that the samples are generated within an OFDM symbol time, which ensures a continuous transmission. For this purpose, the HV approach leverages pipelining at two different levels. One pipelining operates at the OFDM symbol rate, which is shown in

TABLE I
SUPPORTED FEATURES FOR MU-OFDMA TRANSMITTER

PPDU type	x-tone RU size	MCS values	Max RUs
HE SU PPDU	242	0-6	1
HE TB PPDU	242, 106, 52, 26	0-6	1
HE MU PPDU	242, 106, 52, 26	0-6	9

TABLE II
HARDWARE UTILIZATION COMPARISON: HD BASED MU-OFDMA TRANSMITTER VS HV BASED MU-OFDMA TRANSMITTER.

N_{RU}	Resource	HW_{HV}	HW_{HD}	$\eta_{\%}$	η_n
1	LUTs	59611	59611	0	100
	Registers	15768	15768	0	100
	BRAMs	16	16	0	100
	DSPs	21	21	0	100
2	LUTs	59982	112468	46.67	53.33
	Registers	16152	24423	33.87	66.13
	BRAMs	18	29.5	38.98	61.02
	DSPs	21	25	16.00	84.00
4	LUTs	60020	218182	72.49	27.51
	Registers	16394	41733	60.72	39.28
	BRAMs	18	56.5	68.14	31.86
	DSPs	21	33	36.36	63.64
9	LUTs	60499	482467	87.46	12.54
	Registers	16771	85008	80.27	19.73
	BRAMs	18	124	85.48	14.52
	DSPs	21	53	60.38	39.62

Fig. 5 and explained in the previous subsection, the other pipelining shown in Fig. 7 works inside the BCIM module and is managed by the CS-FSM. Based on the processing load in a single CS operation, the submodules of the BCIM module are categorized into pre-modulation and modulation. The pre-modulation part consists of HE data FSM, scrambling, encoding, and interleaving submodules; whereas modulation, pilot generation, and FSM responsible for pilot insertion and modulation are the parts of the modulation category (see Fig. 4-b). In each CS operation, the pre-modulation part needs to process $N_{CBPS}(u)$ of an user, while modulation needs to modulate all SCs (i.e., data and pilot SCs per user) in the assigned RU. Thus, a BRAM is used to decouple these two parts and it has a capacity to store two $OFDM_{CBPS}$ data. The value of $N_{CBPS}(u)$ depends on the MCS and RU size of a given user, whereas the value of SCs/RU relies only on the RU size. At t0 time, all the submodules of pre-modulation part begin the data processing of the first user. After a sub-tick (i.e., time equivalent to process $N_{CBPS}(u)$), the pre-modulation part switches to the second user, meanwhile the submodules of the modulation part are enabled for the first user. The BRAM acts as a ping pong buffer, i.e., it accepts the output of pre-modulation while providing input to the modulation submodules. At the same moment, pre-modulation and modulation submodules are working concurrently, but for different users.

3) Multi-clock domains: The design make uses of three different clocks to meet the timing constraints required by MU-OFDMA TX (see Fig. 4-a). All the modules up to TD-MUX operate at 100 MHz, whereas DUC reads data at 20 MHz and generates IQ samples at 40 MHz. A First-in First-out (FIFO) is placed between DUC and TD-MUX to mitigate metastability. Metastability is a phenomenon which happens when data crosses multiple clock domains and causes the output to be unpredictable.

It is worth noting that, independent on the approach (HD or HV) the BCIM module of the MU-OFDMA TX operates at the same 100 MHz clock speed. It does not comply to the fundamental theory behind the virtualization concept, which demands overclocking in order to generate multiple virtual instances from a single physical instance. In other words, the BCIM module should theoretically operate at $N_{users}\times 100$ MHz. This can be explained as follows, given the fixed total BW in MU-OFDMA TX, the data rate of each RU decreases, as the number of RUs rises. For the BCIM module, the amount of clocks required for processing depends on a user's MCS value. The higher the MCS, the more information bits are coded into a symbol. This work leverages the original *openwifi*'s modulation module, which only supports up to MCS 6. In the worst case scenario, 9 26-tone RUs in 20 MHz channel with each RU having MCS 6 requires ($N_{RU} \times N_{CBPS}(u) = 9 \times 144$) 1296 clocks to generate a single OFDM symbol in the FD, while a single 242-tone RU in a 20 MHz channel with MCS 6 demands ($N_{RU} \times N_{CBPS}(u) = 1 \times 1404$) 1404 clocks. Thus, the number of clocks to process multi users' data could be less than a single user. The in-depth understanding of the OFDMA operation allows us to design a MU-OFDMA TX using HV without overclocking.

4) Timing overhead: As introduced in previous section, the context switching activity requires 4 clock cycles to write to memory for saving context, and another 4 clock cycles to read from memory for restoring context, for every user's each OFDM symbol. This means an timing overhead of 8 clock cycles, which is 80 ns at 100 MHz clock speed. This amount of time is trivial when compared with the time costed for processing $N_{CBPS}(u)$, typically in the order of hundreds of clock cycles depending on the exact RU and MCS configuration of a user. Thanks to the marginal timing overhead, and the pipeline architecture at very fine granularity, we successfully meet the stringent timing constraint to generate waveform continuously on the fly.

IV. RESULTS AND DISCUSSIONS

In this section, the HW utilization of the HV based MU-OFDMA TX is benchmarked against the HD based counterpart. Additionally, the HV based MU-OFDMA TX is validated for a 20 MHz channel and three types of HE PPDUs: i.e., the HE SU, HE TB, and HE MU PPDU. The supported features of the validation are listed in Table. I and are applicable for both HD and HV approaches. As the original *openwifi* supports up to MCS 6, this limitation also applies to our Proof of Concept (PoC). The experimental results in terms of Error Vector Magnitude (EVM) under various PPDU configurations are presented.

A. Comparison of Hardware Utilization

The proposed design is validated on Zynq Ultra-Scale+MPSoC ZCU102 Evaluation Kit [18]. The Zynq MP-SoC is composed of Programmable Logic (PL or FPGA) and

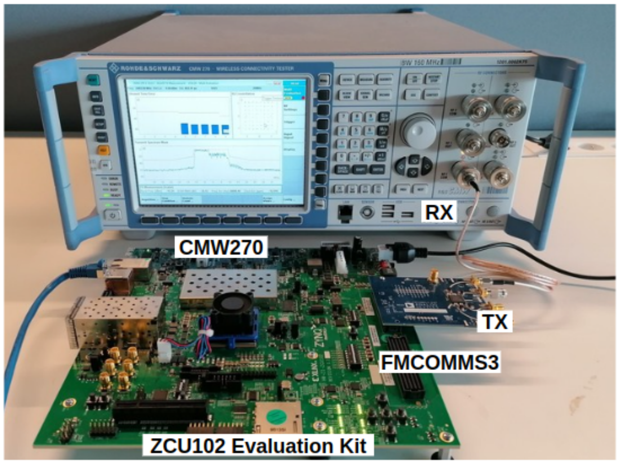


Fig. 8. The experimental setup for validation of MU-OFDMA transmitter.

Processing Subsystem (PS or ARM Cortex-A53). The PHY layer of the MU-OFDMA TX is realized on FPGA, which consists of three parts:

- the logical parts, mapped to LUTs and Registers,
- the memory part, placed on dedicated BRAMs,
- the signal processing part (e.g., multiplication, division), mapped to Digital Signal Processing (DSP) blocks.

Thus, the hardware utilization efficiency is calculated in terms of LUTs, Registers, BRAMs and DSPs, based on the FPGA compilation reports for both approaches, as shown in Table II. The efficiencies (η in Table II) are calculated using (3) and (4).

$$Improved\ Efficiency(\eta_{\%}) = \left(\frac{HW_{HD} - HW_{HV}}{HW_{HD}}\right) \times 100 \tag{3}$$

$$Normalized\ Efficiency(\eta_n) = \left(\frac{HW_{HV}}{HW_{HD}}\right) \times 100 \tag{4}$$

where, HW_{HD} and HW_{HV} represent the HW (i.e. LUTs, Registers, BRAM or DSPS) consumed by HD and HV approaches, respectively. On the selected Zynq SoC, a BRAM has the capacity of 36Kb and a DSP block—DSP48E2 consists of 27 bits pre-adder, 27 ×18 bits signed multiplier and 48 bits datapath multiplexer. The number of BRAMs and DSPs in the Table. II represents the amount of used 36Kb BRAMs and DSP48E2 blocks, respectively.

In general, Table II shows that HV based approach reduces HW consumption for a given N_{RU}, and the benefit is more visible for larger N_{RU}. This is in line with our expectations, as HV involves extra logic and memory for context switching. Fortunately, the majority of the HW overhead remains constant as the number of users increase in MU-OFDMA TX, resulting higher HW efficiency for more users. Thus, the best efficiency is achieved when maximum numbers of users are present, among the supported scenarios this is the case of 9 users, where the HV based design consumes $\approx 13\%$ LUTs, 20% Registers, 15% BRAMs and 40% DSPs as compared to the HD based design.

TABLE III
THE EVM PERFORMANCE OF MU-OFDMA TRANSMITTER.

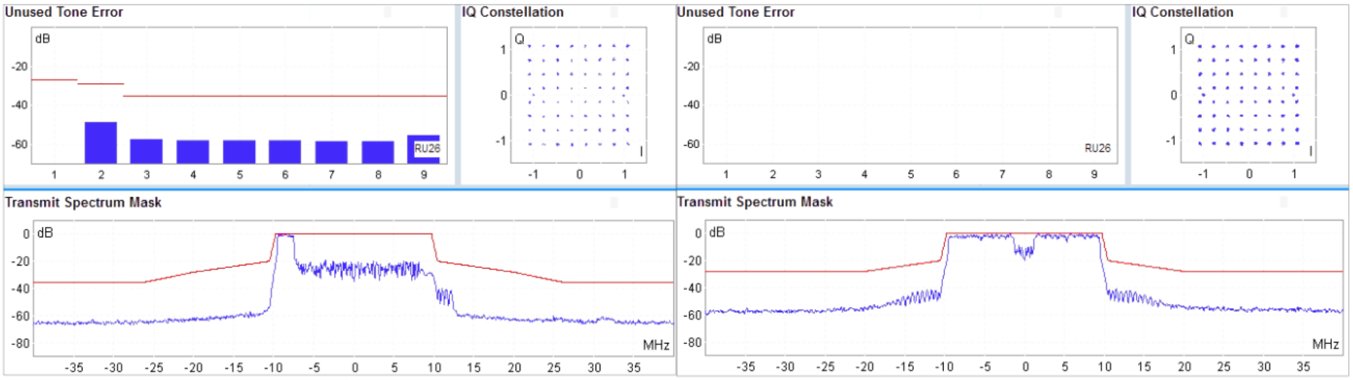
MCS	HE SU/MU PPDU		HE TB PPDU	
	EVM_S(dB)	EVM_M(dB)	EVM_S(dB)	EVM_M(dB)
0	-5	-38.92	-27	-40.26
1	-10	-38.83	-27	-40.36
2	-13	-38.69	-27	-41.05
3	-16	-38.8	-27	-41.43
4	-19	-38.86	-27	-41.02
5	-22	-38.98	-27	-40.87
6	-25	-38.31	-27	-40.69

B. Experimental Validation

The experiment setup (shown in Fig. 8) consists of an SDR and a Rohde & Schwarz CMW270 [19] wireless connectivity tester. The SDR is composed of the ZCU102 evaluation kit and an analog RF front-end board FMCOMMS3 [20]. The MU-OFDMA TX running on the SDR is configured through the modified *openwifi* driver to send IEEE 802.11ax compliant packets over the air, as explained in Section III-D. On the other side, the CMW270 tester, set in non-signaling IEEE 802.11ax compliant mode, decodes the signal transmitted by the SDR and measures EVM, unused tone error, and spectral mask.

1) Error Vector Magnitude: EVM measures of how accurately symbols are transmitted within the constellation diagram. According to the IEEE 802.11ax standard, for HE MU, SU, and TB PPDUs, EVM is measured by receiving a minimum of 20 PPDUs with each at least 16 symbols long (if the RU size is greater than the 26-tone RU) or 32 symbols long (if the RU size is equal to 26-tone RU). Table III's left part shows the measured EVM (EVM_m) values against the allowed EVM (EVM_s) defined in the standard for HE SU/MU PPDU. The measured EVM_m values of MU-OFDMA TX transmitting SU PPDU and MU PPDUs are obtained using the Rohde & Schwarz CMW270 wireless tester. In addition, various possible combinations of RUs are transmitted using the RU allocation vector defined in the IEEE 802.11ax standard (see Table. 27-26 in [3] for more details) and measured. In general, the measured EVM is rather consistent under different MCS and RU configurations, the worst level observed is -38.17 dB. A typical measurement is shown in Table. III, obtained when MU-OFDMA TX is transmitting an MU PPDU with a RU Allocation vector equal to 112; i.e., 4 52-tone RU, each RU assigned with MCS 6. It is evident from the Table. III that the PoC meets the EVM requirements defined in the standard.

The EVM in TB PPDU is measured for both occupied and unoccupied RUs. The EVM of unoccupied RUs is termed as relative EVM (or unused tone error), which is explained in the next paragraph. The right half of Table. III depicts the EVM of the occupied RUs for HE TB PPDU. The worst EVM_m for TB PPDU is -40.03 dB. Again we observe very stable EVM performance irrespective to the RU and MCS configurations The EVM_m shown in Table. III for TB PPDU are measured when the MU-OFDMA TX is transmitting a TB PPDU on the first 26-tone RU1 using MCS 6 in the 20 MHz channel. Such kind of information (i.e., RU index, channel width) are sent by an AP to a STA using the trigger frame before the actual UL transmission happens.



(a) (b)

Fig. 9. Performance of the MU-OFDMA TX in terms of EVM, unused tone error, and spectral mask measured by a Rohde & Schwarz CMW270 tester when transmitting (a) HE TB PPDU containing a single 26-tone RU (RU1) and (b) HE MU PPDU containing 4 52-tone RUs.

2) Unused Tone Error: The unused tone error is a new performance metric introduced in IEEE 802.11ax to quantify the performance of unoccupied RUs in TB PPDU. The main incentive to introduce the unused tone error is to minimize the interference between adjacent RUs of TB PPDU. It is calculated with respect to the average power of data subcarrier in the occupied RU(s). Fig. 9-a shows a screenshot of the CMW 270 while a TB PPDU at the first 26-tone RU is being transmitted by the SDR using MCS 6. The blue bars in the top left graph of Fig. 9-a is the measured unused tone error in the unoccupied RUs, whereas the red line is the upper bound defined by Equation 27-131 in the standard [3]. The shape of the blue bars visually explains the form of staircase, which is well below the upper limit. Note that the $Unused\,Tone\,Error$ graph for HE MU PPDU in Fig. 9-b is empty, because the metric is only valid for HE TB PPDU.

3) Spectral Mask: The spectral mask is defined to limit out-of-band emission on adjacent channels. It is provided in the units of Decibel relative (DBr) (i.e., dB relative to maximum spectral density of the signal). The bottom graphs in Fig. 9 show the measured spectral masks in blue line versus the allowed spectral mask in red line. The HE MU PPDU in Fig. 9-b is transmitting on 4 52-tone RUs with each 52-tone RU carries different users' data. The dip in Fig. 9-b corresponds to the location of the central 26-tone RU, which is not occupied. Similarly, the HE TB PPDU in Fig. 9-a is transmitting data on 26-tone RU1 and the remaining eight 26-tone (i.e., RU2 to RU9) are unused. It is evident from the spectral mask of Fig. 9 that the MU-OFDMA TX meets the requirements defined in IEEE 802.11ax standard.

V. Conclusions

This paper introduces a hardware efficient design for MU TRX. Unlike the traditional duplication approach where a dedicated physical layer is allocated for each user, the proposed design creates multiple logical physical layers based on a single physical layer by leveraging the hardware virtualization technique. To validate the proposed design, an IEEE 802.11ax compliant MU-OFDMA transmitter is implemented on an SDR platform. The MU-OFDMA transmitter is compliant to the IEEE 802.11ax standard and supports HE SU PPDU, HE MU PPDU and TB PPDU. The performance of the MU-OFDMA transmitter has been quantified in terms of spectral mask, EVM and unused tone error by a Rohde & Schwarz CMW 270 tester. The measurements prove that the MU-OFDMA transmitter well surpasses the requirement of the IEEE 802.11ax standard, at the same time it saves $\approx$ 87% LUTs and $\approx$ 80% Registers hardware resources for the case of 9 RUs. Hence the design could effectively reduce the die area of a final commercial chip.

The proposed design is validated with the IEEE 802.11ax compliant MU-OFDMA transmitter, however it can be applied to any transceiver or standard involving MU communication such as IEEE 802.16, LTE, and 5G. In the future, we are planning to extend work towards the IEEE 802.11ax compliant MU-OFDMA receiver and also toward the MU-MIMO, resulting a "total duplication-free" 802.11ax chip design.

References

[1] K. O. M. Salih, T. A. Rashid, D. Radovanovic, and N. Bacanin, "A comprehensive survey on the internet of things with the industrial marketplace," *Sensors*, vol. 22, no. 3, p. 730, Jan. 2022.

[2] *802.11-2020 - IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 2020.

[3] *802.11ax-2021 - IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE, 2021.

[4] *802.16-2017 - IEEE Standard for Air Interface for Broadband Wireless Access Systems*, IEEE, 2017.

[5] (2022) The mobile broadband standard. 3GPP. [Online]. Available: https://www.3gpp.org/

[6] K. Nishimori, R. Kudo, N. Honma, Y. Takatori, and M. Mizoguchi, "16x16 multiuser mimo testbed employing simple adaptive modulation scheme," in *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*, Spain, 2009, pp. 1–5.

[7] M. Wenk, P. Luethi, T. Koch, P. Maechler, N. Felber, W. Fichtner, and M. Lerjen, "Hardware platform and implementation of a real-time multi-user mimo-ofdm testbed," in *2009 IEEE International Symposium on Circuits and Systems*, Taiwan, 2009, pp. 789–792.

[8] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source ieee802. 11 sdr implementation on soc," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Belgium, 2020, pp. 1–2.

[9] T. H. Tran, Y. Nagao, M. Kurosaki, B. Sai, and H. Ochi, "Asic design of 600mbps 4×4 mimo wireless lan system," in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, South Korea, 2012, pp. 360–363.

[10] D. Perels, S. Haene, P. Luethi, A. Burg, N. Felber, W. Fichtner, and H. Bolcskei, "Asic implementation of a mimo-ofdm transceiver for 192 mbps wlans," in *Proceedings of the 31st European Solid-State Circuits Conference, 2005. ESSCIRC 2005.*, France, 2005, pp. 215–218.

[11] R. Carlos, G. Rodolfo, D. Luís, H. Akram, and C. Rafael F.S., "Multi-gigabit/s ofdm real-time based transceiver engine for emerging 5g mimo systems," *Physical Communication*, vol. 38, p. 100957, 2020.

[12] A. Agarwal, V. K. Sinha, R. Palisetty, P. Kumar, K. C. Ray, K. Kumar, and T. Pandey, "Performance analysis and fpga prototype of variable rate go-ofdma baseband transmission scheme," *Wireless Personal Communications*, vol. 108, no. 2, pp. 785–809, 2019.

[13] V. Mukati, A. R. Khan, B. V. Reddy, and P. Kumar, "A variable rate multi-user ci/go-ofdma scheme over frequency selective channel," in *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, India, 2014, pp. 1–5.

[14] Y. Daldoul, D.-E. Meddour, and A. Ksentini, "Performance evaluation of ofdma and mu-mimo in 802.11ax networks," *Computer Networks*, vol. 182, p. 107477, 2020.

[15] D. Magrin, S. Avallone, S. Roy, and M. Zorzi, "Validation of the ns-3 802.11 ax ofdma implementation," in *Proceedings of the Workshop on ns-3*, USA, 2021, pp. 1–8.

[16] O. Seijo, J. A. Lopez-Fernandez, and I. Val, "w-sharp: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3651–3662, 2021.

[17] *802.11ax Waveform Generation*, MATLAB, 2021. [Online]. Available: https://nl.mathworks.com/help/wlan/ug/802-11ax-waveform-generation.html

[18] *Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit*, Xilinx, 2021. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html

[19] *CMW270 wireless connectivity tester*, Rohde and Schwarz, 2021. [Online]. Available: https://www.rohde-schwarz.com/us/products/test-and-measurement/wireless-tester-network-emulator/rs-cmw270-wireless-connectivity-tester_63493-9552.html

[20] *AD-FMCOMMS3-EBZ: An AD9361 Software Defined Radio Board*, Analog Devices, 2021. [Online]. Available: https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcomms3-ebz.html