A Real-Time Experimentation Platform for sub-6 GHz and Millimeter-Wave MIMO Systems

Jesus O. Lacruz*
jesusomar.lacruz@imdea.org
IMDEA Networks Institute
Madrid, Spain

Rafael Ruiz Ortiz*
rafael.ruiz@imdea.org
IMDEA Networks Institute
Madrid, Spain

Joerg Widmer
joerg.widmer@imdea.org
IMDEA Networks Institute
Madrid, Spain

ABSTRACT

The performance of wireless communication systems is evolving rapidly, making it difficult to build experimentation platforms that meet the hardware requirements of new standards. The bandwidth of current systems ranges from 160 MHz for IEEE 802.11ac/ax to 2 GHz for Millimeter-Wave (mm-wave) IEEE 802.11ad/ay, and they support up to 8 spatial MIMO streams. Mobile 5G and beyond systems have a similarly diverse set of requirements.

To address this, we propose a highly configurable wireless platform that meets such requirements and is both affordable and scalable. It is implemented on a single state-of-the-art FPGA board that can be configured from 4x4 mm-wave MIMO with 2 GHz channels to 8x8 MIMO with 160 MHz channels in sub-6 GHz bands. In addition, multi-band operation will play an important role in future wireless networks and our platform supports mixed configurations with simultaneous use of mm-wave and sub-6 GHz. Finally, the platform supports real-time operation, e.g., for closed-loop MIMO beam training with low-latency, by implementing suitable hardware/software accelerators. We demonstrate the platform's performance in a wide range of experiments. The platform is provided as open-source to build a community to use and extend it.

CCS CONCEPTS

• **Networks** → **Network experimentation**; **Mobile networks**; • **Computer systems organization** → **Real-time system architecture**; • **Hardware** → **Digital signal processing**.

KEYWORDS

Millimeter Wave, Testbed, FPGA, MIMO, Phased Antenna Array, Multi-band

ACM Reference Format:
Jesus O. Lacruz, Rafael Ruiz Ortiz, and Joerg Widmer. 2021. A Real-Time Experimentation Platform for sub-6 GHz and Millimeter-Wave MIMO Systems. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24-July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3458864.3466868

*Co-Primary Authors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiSys '21, June 24-July 2, 2021, Virtual, WI, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8443-8/21/07…$15.00
https://doi.org/10.1145/3458864.3466868

1 INTRODUCTION

Wireless networks are evolving at a rapid pace to meet increasing data rate and latency requirements. Applications like virtual and augmented reality, remote surgery, vehicular connectivity, connected homes, and factory automation, require network performance that far exceeds that of currently deployed wireless systems. To address this, the capacity of wireless networks is constantly being increased through wider channels, including channel bonding and aggregation, multiple spatial streams in Multiple Input Multiple Output (MIMO) systems, and higher-order Modulation and Coding Schemes (MCSs). The recent IEEE 802.11ax standard [32] uses 160 MHz channels, up to 8x8 MIMO and raw data rates of up to 9.6 GBit/s. Significantly more bandwidth for even wider channels is available at Millimeter-Wave (mm-wave) frequencies, and wireless systems for this part of the frequency spectrum have appeared over the last few years. IEEE 802.11ay [33] has emerged as the successor to the IEEE 802.11ad standard [31], supporting up to 8 spatial streams, channel bonding up to 8 GHz, and high-order MCSs to reach theoretical data rates of 256 GBit/s [6]. At the same time, new 3GPP 5G NR standards [30] introduce wireless technologies with similar characteristics at sub-6 GHz and mm-wave frequencies to target high data rates and device densities, as well as low latency.

Multi-band operation is included in the latest 5G-NR standards, where mm-wave and sub-6GHz interfaces are present in the same system. Also WiFi APs typically support multiple bands, e.g., IEEE 802.11ad/ay and IEEE 802.11ac/ax. While it is possible to combine multiple separate platforms for multi-band research, having them on the same device (or even the same circuit board) is more efficient and helps to implement a variety of applications such as ultra-fast session transfer [13, 25, 29] or multi-band environment sensing and activity recognition using coherent channel state information from multiple bands. Furthermore, it is possible to share common processing blocks between the interfaces to save chip area and reduce power consumption.

For wireless systems research, experimentation is a vital part of the research process. Given the diverse and rapidly changing requirements of wireless systems, it is difficult to have platforms that adapt well to all of the constraints imposed by the different use cases. Testbeds of Commercial-Off-The-Shelf (COTS) devices are standard-compliant, easy to use, and inexpensive [23] and some even provide limited access to lower layer functionality [8, 28]. However, research requiring modifications to the MAC and physical layers is usually only feasible with FPGA-based Software Defined Radio (SDR) systems [10, 20, 21, 34]. Connecting multiple synchronized SDRs even allows to build large MIMO systems [18, 26], but they are often not suitable to be used as general purpose research platforms due to

their cost and highly optimized design. Flexible and affordable SDR-based testbeds for mm-wave only support narrow channels and low data rates, and are not compatible with current (let alone future) mm-wave standards [1, 19]. Two full-bandwidth systems based on more powerful FPGAs exist but do not support MIMO [11, 24]. [22] presents an early stage implementation of a fully-digital MIMO testbed using an FPGA with a custom-made daughter-board. No single FPGA platform caters to both to high-order MIMO sub-6 GHz research with narrow channels and mm-wave research with much wider channels and fewer RF chains. Furthermore, except for the expensive design in [24], the existing mm-wave platforms cannot operate in a *closed-loop* manner, where the devices switch between transmission and reception and can react to received packets. These features are crucial for research on advanced future MIMO systems.

To fill this gap, we design a MIMO Radio Platform for Heterogeneous wireless systems (*MIMORPH*), a highly flexible experimentation platform that supports configurations from mm-wave 2x2 MIMO with 4 GHz of bandwidth to sub-6 GHz 8x8 MIMO with 1 GHz of bandwidth, as well as *simultaneous* combinations thereof. *MIMORPH* is built on the Radio Frequency System on a Chip (RFSoC) platform [36] that integrates multiple AD/DA converters with giga-sampling rates, two multi-core processors and programmable logic. To ensure flexibility of the system and avoid the complexity of a bespoke implementation for each of the supported standards, *MIMORPH* follows a memory-based design [2] that transmits signals from and captures samples to the on-board memories for pre- and post-processing. *MIMORPH* can easily be adapted to different sampling frequencies and bandwidths, super sampling rates and sample resolutions. The system uses a buffer structure that enables different tradeoffs between the maximum bandwidth, quality of the signal, and maximum packet rate to stay within the overall DDR memory read/write speed limits. We provide simple functions to manage the system and tune the functionality according to the user's needs, without requiring advanced hardware-design skills. The designed blocks are connected using Advanced eXtensible Interfaces (AXIs), a widely adopted industry standard for on-chip communication, which simplifies the integration of new signal processing blocks. To support real-time experimentation, we implement hardware accelerators for packet detection, synchronization, channel estimation, and antenna reconfiguration, as well as transmission and decoding of control packets for closed-loop experiments.

We validate *MIMORPH* using the following configurations: 1) 60 GHz 4x4 MIMO with 2 GHz channels, 2) sub-6 GHz 4x4 MIMO with 160 MHz-wide IEEE 802.11ax signals, 3) 8x8 MIMO with 160 MHz over cable, and 4) a multi-band configuration with simultaneous use of mm-wave 2x2 MIMO with 2 GHz bandwidth and sub-6 GHz 4x4 MIMO with 160 MHz.[1] We also design the signal processing blocks for IEEE 802.11ay-compliant MIMO beam training with *in-packet* antenna reconfiguration [6, 33]. We validate the system and the signal processing blocks operating in a closed-loop manner for real-time channel estimation and beam training in mobile scenarios.

We believe our system can become a standard platform for high-performance MIMO research. Compared to other platforms, the design is very affordable for the performance it offers. With $9K for the Xilinx FPGA platform and $2.5K for a mm-wave RF front-end, a full 4x4 mm-wave MIMO node is below $20K, an order of magnitude less than the cost of an X60 Single Input Single Output (SISO) node [24]. We develop the system using standard interfaces and easy-to-use functions with the aim of creating a community to further develop and enhance the system, and make the implementation of *MIMORPH* available as open source.

[1]We lack RF front-ends supporting 4 GHz of bandwidth as well as the RF break-out boards for 8x8 MIMO over the air.

2 *MIMORPH* ARCHITECTURE

Since *MIMORPH* is designed as a general purpose platform for wireless experimentation, its main components need to easily adapt to diverse requirements. To this end, we choose a *memory-based* design for most of the processing architecture and forego a full hardware implementation for the sake of flexibility and scalability. In our design, I/Q samples from memory are fed to the DACs at the transmitter, sent over the air, and then the samples from the ADCs are fed back to memory at the receiver side. We include additional hardware processing blocks for certain real-time processing components. The system architecture is divided in: i) a Processing System (PS) with general purpose processors used for management and system control as well as implementation of non-time critical applications and ii) a Programmable Logic (PL) with the actual FPGA logic, AD/DA converters and memory. This architecture helps to modularize the system with clearly defined boundaries between processors and FPGA logic. At the same time, the overall system is tightly coupled by means of control and high-speed interfaces. In our design, we avoid custom interfaces and use AXI [3] for all inter-block communication as well as for the control and management from the PS. AXI allows to modularize the system and supports asynchronous transfers, flexible data widths, burst transactions, and hand-shake signalling, which makes it ideal for on-chip communication.

We implement *MIMORPH* on the Xilinx RFSoC ZCU111 development board [36], which integrates multiple Giga-sampling rate AD/DA converters, DDR memory, powerful multi-core ARM processors and close to one million logic elements for the processing blocks. Nevertheless, our architecture and processing blocks can be ported to other FPGA devices, thanks to the standard interfaces and modularity adopted in the design of the system.

2.1 Macro-Channel Datapaths

The first design consideration for a multi-band system is related to the datapaths, given the different bandwidths and sampling frequencies. While sub-6 GHz architectures can be built with datapaths that process one-sample per clock cycle, the Giga-Samples per Second (GSPS) sampling frequencies of mm-wave systems require Super Sample Rate (SSR) datapaths and signal processing blocks that process multiple samples in parallel in a single clock cycle. This allows to process I/Q samples at GSPS rates with FPGAs having clock frequencies of only hundreds of MHz. To support concurrent datapaths with different configurations, we use the concept of *channelizers* that allow to separate the read/write sections of the datapaths from the actual signal processing blocks in the design. Making use of this strategy, only minor modifications are required to *morph* the functionality of the system.

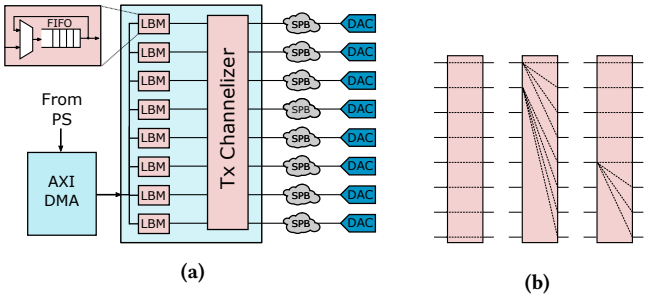


Figure 1: (a) Transmit datapath. (b) Different configurations of the Tx channelizer.

2.1.1 Transmitter Side. The transmit chain has 8 macro channels connected to the signal processing blocks and then to the DACs through a Tx Channelizer, as shown in Fig. 1a. The I/Q samples from the PS arrive via Direct Memory Access (DMA) and are distributed to the 8 Loop-Back Memory (LBM) blocks. These LBM are FIFO queues with loop-back logic that work as circular buffers to continuously transmit the samples they store. Each LBM has an SSR factor of 16, i.e., it outputs samples at a rate of 16 times the clock frequency of the transmit datapath. The output of the LBM passes to a Tx channelizer which *distributes* the I/Q samples to the signal processing blocks which finally pass them on to the DACs. To make efficient use of the memory, the transmit chain uses the on-chip block memory banks to store the independent I/Q samples for each of the macro channels, thus leaving all of the on-board DDR4 RAM to store the received samples. This design enables full-duplex without reducing the maximum bandwidth of the system.

To highlight the flexibility of this design, we provide three examples: a 4x4 wideband mm-wave system, an 8x8 sub-6 GHz system, and a 2x2 mm-wave + 4x4 sub-6 GHz multi-band configuration. Fig. 1b shows the configuration of the Tx channelizer for these examples. Wideband mm-wave requires the leftmost configuration in Fig. 1b, where the I/Q samples with SSR = 16 pass directly to the signal processing blocks and then to the converters. With the direct up-conversion mm-wave RF chains [27] we use, one LBM is used for the in-phase samples and another for the quadrature samples, allowing for 1 to 4 spatial streams. For an 8x8 sub-6 GHz system with 160 MHz of bandwidth, the Tx channelizer distributes the I/Q samples of two of the LBMs to all the DACs in the datapath, as shown in the middle graph in Fig. 1b. This configuration has an SSR = 2. With internal up-sampling and Numerical Controlled Oscillators (NCOs) sub-systems it is possible to operate the system in the 2.4 GHz and the 5 GHz band for IEEE 802.11ac/ax. The rightmost setup of the Tx channelizer in Fig. 1b shows a mixed configuration to support multi-band in a single node. The datapaths provide 2 spatial streams with separate I and Q that lead to the mm-wave front-ends, as well as 4 spatial streams for the sub-6GHz system. A total of 5 LBMs are required. The NCOs and up-sampling sub-systems are enabled only for the sub-6GHz spatial streams. With this design, different SSR factors can co-exist in the same system and only require configuration of the Tx channelizer and the signal processing blocks for each of the spatial streams. The architecture also scales to more powerful systems [37] that support 8 to 16 concurrent streams.

2.1.2 Receiver Side. The receiver datapath follows the structure of Fig. 2, with up to 4 macro channels to transfer the samples coming from the ADCs and signal processing blocks (SPB) to the DDR memory. We consider the same example configurations as for the transmitter datapath. The receiver datapath has the following main components. The *DDR4 memory* is accessed through a memory interface IP from Xilinx, which includes an AXI bus to communicate with the device. The write efficiency of DDR memory depends on how contiguous the memory accesses are and on the burst size used in the AXI transfer. We therefore design a highly optimized *AXI DMA* which performs continuous and contiguous writing of a configurable number of samples using the longest possible DDR burst size. With this configuration we reach 87% of writing efficiency, which is close to the values given in the reference manual [35]. Since the DDR4 is shared between all the active channels, a *channel multiplexer* is necessary to write the samples to memory. It reads I/Q samples sequentially from each macro channel for the duration of a burst before switching to the next channel. *FIFO* buffers are included to queue samples from the macro channels while the channel multiplexer is serving another channel and for clock domain crossing between the ADC/signal processing blocks and the DDR.

The *Rx channelizer* has the complementary purpose of that of the transmitter. The samples coming from the ADCs and signal processing blocks are grouped depending on the SSR factor of each stream to fit the width of the AXI datapath to the memory. The configuration in Fig. 2a is for 4x4 mm-wave systems where the ADCs are configured in pairs to carry the I and Q samples separately. Fig. 2b shows the configuration for the 8x8 sub-6 GHz system, where the SSR factor of 1 allows to group all the I/Q streams to a single macro channel, reducing the DDR write speed requirements. Finally, Fig. 2c shows the configuration for the multi-band system with 2x2 mm-wave + 4x4 sub-6 GHz streams, using three macro channels.

2.2 Storing the Received Samples

We consider the most demanding configuration of multiple wideband mm-wave streams, which easily extends to other, less demanding configurations. The maximum DDR write speed is 148 Gbps (at 87% efficiency). For an IEEE 802.11ad/ay system with a sampling frequency of 3.52 GHz (1.76 GSPS × 2 samples per symbol) and 16-bit I/Q samples, a single stream requires a DDR throughput of 112 Gbps. However, 2 and 4 streams require speeds of 225 and 450 Gbps, respectively, which exceeds the memory speed. Below we discuss two solutions for the DDR write speed limit: reduced sample resolution and a configurable inter-frame delay.

2.2.1 Reducing the Sample Resolution. To reduce the sample resolution, we integrate a bit-slicer and a sample concatenation block as shown in Fig. 4. The bit slicer takes the W-bit samples coming from the ADC with a certain SSR factor ($N = W \cdot SSR$) and removes a configurable number of least-significant bits of each sample to output W^*-bit samples with the same SSR factor. The sample concatenation block then groups the $W^* \cdot SSR$ bits to output samples in an N-bit wide AXI data path for the Rx channelizer. With this

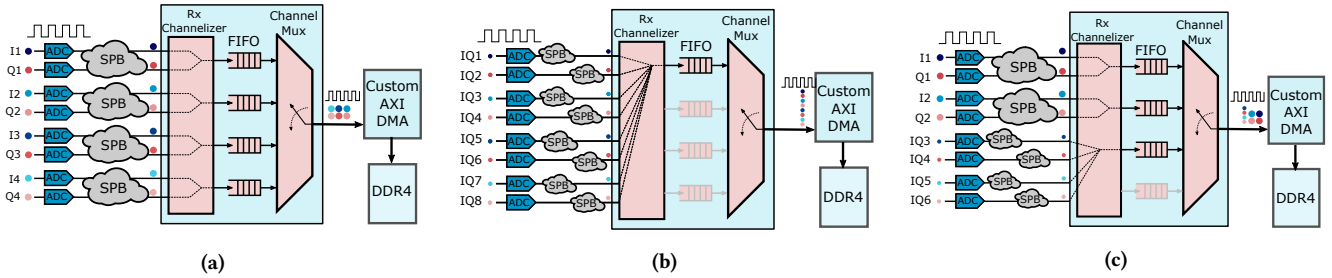


Figure 2: Different configurations of the receiver data path. (a) 4x4 mm-wave system. (b) 8x8 sub-6GHz configuration. (c) multi-band with 2x2 mm-wave + 4x4 sub-6GHz.

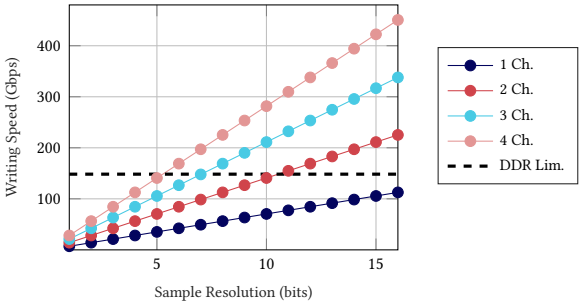


Figure 3: DDR write speed vs. I/Q sample resolution.

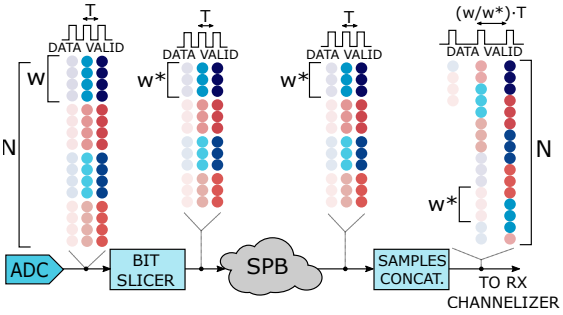


Figure 4: Blocks to reduce sample resolution

architecture, the Rx channelizer is independent of the sample resolution and thus does not need any redesign. As we can see from Fig. 3, to continuously capture samples with the 4 macro channels at a sampling frequency of 3.52 GSPS, at most 5-bit resolution samples can be captured. This value increases to 7 bits for three channels and 10 bits for two channels.

This reduced resolution system has advantages in scenarios where continuous long captures are necessary to measure certain channel effects that cannot be studied with short packets. Besides, the structure from Fig. 4 can be used to analyze low-resolution signal processing blocks that have important applications in the design of low-power mm-wave systems [5, 38]. Another advantage is that it is possible to use different configurations without having to design specific signal processing blocks for each application.

2.2.2 Configurable Inter-Frame Spacing. It is further possible to operate at full bandwidth without sacrificing sample resolution, by using a configurable inter-frame spacing and introducing a packet detection block in the receiver structure. To this end, in the transmitter datapath we set a configurable space between packets per LBM cycle. For MIMO, this functionality is implemented in a synchronous manner for all the spatial streams. Besides, in the receiver datapath, a packet detection block is required to determine the start of a frame. This works as follows: by default, the AXI output signal (`DATA VALID`) is set to low and samples are discarded. When a valid packet is detected, `DATA VALID` switches to high for a configurable number of clock cycles and samples are stored in the FIFO buffer. Since samples are saved only intermittently, the data rate to the DDR memory is reduced without sacrificing sample resolution.

Fig. 5 (left) shows the relation between the DDR write speed and the inter-frame spacing relative to the packet length, for a sampling frequency of 3.52 GSPS and 16-bit samples. With 1 channel there is no need for frame separation while for 2, 3 and 4 channels, the proportion of the packet length required for inter-frame space is 0.55, 1.3 and 2.1, respectively. As shown in Fig. 5 (right), inter-frame spacing can be used in conjunction with the reduced resolution configuration to trade off packet rate and signal quality.

2.3 System Management

MIMORPH includes a PS with a quad-core ARM Cortex-A53 and a real-time ARM R5 subsystem. The *MIMORPH* system is managed by a single ARM core, leaving the remaining processing resources free for user-specific mixed hardware-software applications. The PS is connected to the PL by means of AXI to access the LBMs, DDR memory, as well as the configuration registers for each one of the blocks of the system.

We include libraries to manage the configuration of the PL, specifically the AD/DA converters, NCOs and interpolation/decimation sub-systems, on-board clocks, PLLs, SD controller and the remaining peripherals on the board. These devices are configured at system start-up and their configuration can be updated at run-time. Furthermore, we include an SPI management block to send (receive) commands to (from) external devices connected to the GPIO ports on the board. This is particularly important to enable real-time reconfiguration of the 60GHz front-ends of *MIMORPH* to change the amplifier gain, switch between transmit and receive mode, and change the antenna beam patterns. Finally, we include a lightweight TCP/IP stack (*lwIP*) [17] to enable communication with a host PC via an Ethernet connection. With this, we can program simple commands in MATLAB for the most common functions needed to use *MIMORPH* as a research platform:

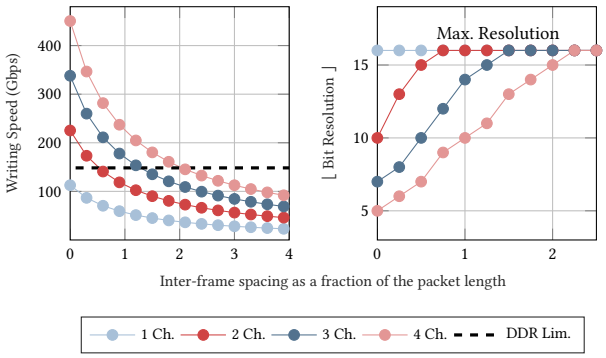


Figure 5: DDR write speed for different inter-frame spacing and bit resolution

- `system_init`: configures the TCP/IP stack in the host PC and opens a connection with *MIMORPH*.
- `configure_<X>`: is a generic function prototype used to configure a specific device or signal processing block in the system, whose name is indicated by `X` (e.g., packet detector, antenna control block, etc.). Each component requires its specific list of arguments, depending on the functionality.
- `transmit_<X>CH`: sends to *MIMORPH* the set of I/Q samples the user wants to transmit using the number of macro channels `X`, loads them to the corresponding LBMs and starts the transmission synchronously in all enabled channels.
- `capture_samples_<X>CH`: enables the continuous capture of I/Q samples from `X` number of macro channels. The captured samples can be sent via TCP/IP or saved on an SD card (if present). Internally, the function configures the channel multiplexer, resets the corresponding FIFOs, synchronizes the datapaths and enables DMA to capture the number of samples requested by the user.
- `capture_pkt_<X>CH`: is similar to the previous function, but instead of requesting a specific number of I/Q samples, it captures a certain number of packets. The packet detector needs to be included in the receiver datapath and properly configured using the `configure_PD` function. The captured packets are again sent to the host PC or saved on an SD card, depending on the function arguments.

Besides the commands mentioned above, *MIMORPH* includes several additional helper functions which are documented in the testbed files. It is possible to analyze the performance of signal processing blocks operating in real-time to verify their proper functioning, quantify the errors, and detect possible design and implementation issues. These simple functions allow to manage the entire system without requiring complex configuration and provide an important step towards a true *plug & play* experimentation platform.

2.4 Remarks

MIMORPH is unique in that it seamlessly supports both sub-6GHz, mm-wave, and multi-band experimentation with all their specific characteristics, simply using some configuration commands. A crucial design decision that enables this flexibility is the isolation of the writing/reading parts and the signal processing blocks in the datapath. Given the memory-based design, the platform can be used to transmit and receive signals with arbitrary structure, making it ideal for experimentation involving not only standard compliant signals but also new algorithms, packet formats, and waveform designs. The memory management and channelizer concepts can be extended to systems with more AD/DA converters. This imposes higher demands on the on-board shared memory access, which can be addressed using the concepts discussed in Section 2.2. *MIMORPH* makes efficient use of the FPGA resources, leaving enough logic elements and memory free to further extend the capabilities of the testbed. Specifically, the *MIMORPH* implementation using Xilinx Vivado 2019.1 requires less than 15% of Slices Registers, 65% of Block RAM resources (due to the memory-based design) and only 3% of DSP48 units.

3 *MIMORPH* REAL-TIME HARDWARE ACCELERATION BLOCKS

The general architecture of *MIMORPH* described in Section 2 meets the requirements of a general purpose platform for heterogeneous wireless experimentation. The user can flexibly configure the platform to operate in multiple different frequency bands and with different bandwidths. With the memory-based design of the platform, it can easily be used in an *open-loop* manner with the transmitter sending a certain number of packets while a receiver captures the frames and later streams them to a host PC for post-processing. While this functionality allows to perform a wide variety of experiments, the latency of a pure memory-based system with offline processing is too high for closed-loop experiments, where the devices interact and exchange frames in both directions. This is particularly true to the high processing requirements of mm-wave MIMO systems.

To fill this gap, we augment the functionality of *MIMORPH* by means of hardware accelerators that help to offload part of the frame processing overhead from the PS to the PL. This sufficiently reduces latency to allow for closed-loop communication with real-time feedback.

3.1 Transmitter blocks

• **Frame generation:** Our transmitter architecture allows to transmit moderate-length frames, given the available on-chip block RAM.[2] To transmit longer frames (without having to reallocate DDR memory for this purpose), it is convenient to generate parts of the frames online, in particular the preamble and similar repeated sequences present in the structure of a frame. A specific example are the frames of IEEE 802.11ay systems which can include training fields (TRN) appended after the data part to perform *in-packet* training [6]. TRN fields are composed of complementary orthogonal Golay sequences which can be sent concurrently via each one of the spatial streams in a MIMO configuration. Even small data packets with a high number of TRN fields may exceed the size of the LBM when using fully offline packet generation. To this end, we implement a transmitter block which builds the packet in real-time, using the LBM only to store the payload (i.e., the header and data fields) of the packet, whereas the remaining fields of the packet including the preamble with Short Training Field (STF) and

[2]For 4x4 mm-wave MIMO, the most demanding configuration, each LBM can store 64K samples, to leave enough memory resources to implement signal processing blocks.

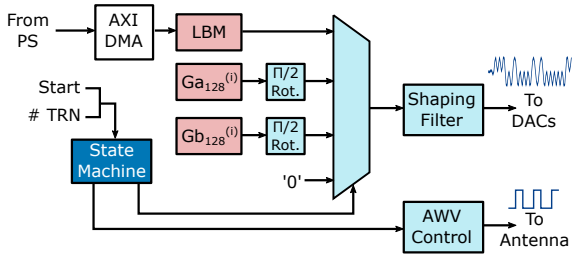


Figure 6: Transmitter blocks

Channel Estimation Field (CEF), as well as TRN fields are generated online by the PL. We further include a shaping filter (raised-cosine) which up-samples the signal by a factor of two. The filter coefficients are freely configurable for different roll-off factors. As can be seen from Fig. 6, we include a state machine that builds each one of the fields of the packets alternating between the Golay sequences stored in memory and the LBM storing the payload of the packet. The block starts sending a packet by means of an external trigger signal coming from the PS. Although the preamble and TRN fields are composed of Golay sequences, configuring the block to use different training sequences requires just minor modifications and allows for simple extension of the block to support different packet structures. This simple structure also saves PL area to facilitate future extensions with further hardware blocks. For example, storing a MIMO 4x4 packet with 64 TRN fields and a payload of 512 bytes (16QAM modulation and rate 3/4 LDPC encoded) would require 13.55 Mb of memory (exceeding the size of the LBM in the FPGA), while using the proposed partial online generation only requires to store 200Kb, a 65-fold reduction in memory resources of the FPGA.

• **AWV control:** Most of the RF front-ends with phased antenna arrays include phase shifters to enable analog beamforming. Phase shifter configurations corresponding to different beam patterns are stored in so-called Antenna Wave Vectors (AWVs). Real-time antenna reconfiguration (and specifically beam training) plays a very important role for mm-wave communications, since antenna misalignment may significantly reduce the effective SNR and thus the throughput. This requires a closed-loop experimentation platform to support mobile experiments. The 60 GHz development kit from Sivers [27] allows fast antenna reconfiguration by means of pulses sent to the GPIO ports. The antenna has an index vector to configure the different AWVs, with pointers that can move through the index vector by means of the GPIO pulses. Three different pulses are supported: i) `INC` advances the index vector position by one; ii) `RTN` moves to a configurable position on the index vector and, when a new `INC` pulse is received, moves back to the *previous* position that was active before receiving the `RTN` pulse and continues incrementing from there; iii) `RST` moves the pointer to the default position in the index vector. We use this functionality to design an AWV Control Block (Fig. 6) that implements a state machine to generate these pulses for the 60GHz front-end to perform real-time antenna reconfiguration with very low latency. The block can be used to configure different beam patterns for each of the TRN sub-fields at the end of an IEEE 802.11ay frame. The implemented functionality meets all of the requirements for IEEE 802.11ay MIMO beam training. The block can be triggered from both, transmitter and receiver datapaths, enabling the implementation of joint transmitter/receiver beam training mechanisms [33].

3.2 Receiver blocks

The receiver blocks allow to detect, synchronize and perform channel estimation over a received frame, providing the necessary functionality to implement a wide variety of real-time experiments. Since the main focus of the paper is to present a memory-based design with specific hardware accelerators for timing-critical functions, the hardware blocks process part of the frames in real-time, while the actual data payload of the packet is stored in the on-board DDR for post-processing. Fig. 7a presents the hardware accelerators designed for the receiver chain. As can be seen, the memory-based spirit of the design coexists well with real-time hardware accelerators, with the samples going to the Rx Channelizer, the AXI DMA, and then to the DDR4 memory.

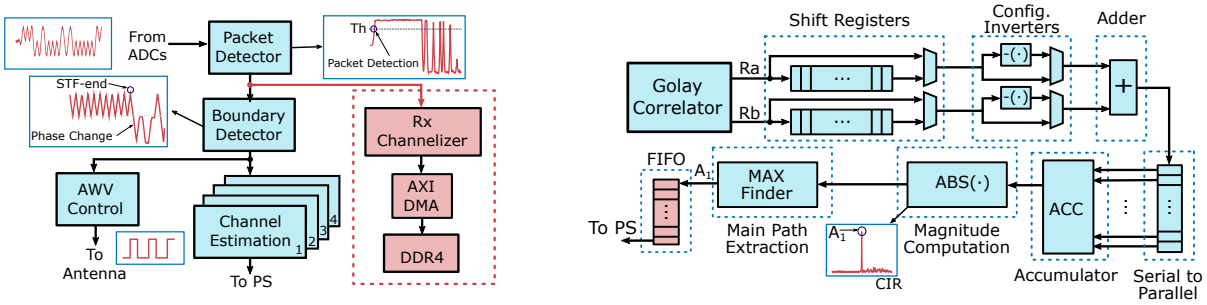
• **Packet detection:** This block enables not only efficient use of the DDR memory in a purely memory-based approach (as discussed in Section 2), but also allows to trigger the receiver signal processing blocks in the receiver chain, as shown in Fig. 7a. We implement a configurable packet detector using a normalized auto-correlation algorithm [11, 14] which detects the periodic sequences used for packet preambles [31, 33]. The block is flexible to be used with different preambles with different periods of repetitive sequences, and different SSR factors for different sampling frequencies. The block outputs a packet detection flag (`PD_FLAG`) which trigger the functionality of the rest of the blocks in the receiver datapath.

• **Boundary detection:** The boundary detection algorithm estimates the position of the last sample of the preamble, providing a coarse synchronization for the rest of the signal processing blocks in the receiver chain. The block follows the approach from [9, 15] to implement Eq. 1, which takes advantage of the inverted sign of the last Golay sequence in the preamble of IEEE 802.11ad/ay frames to track a phase inversion in the correlation pattern, as shown in Fig. 7a, to determine the last sample of the preamble.

$$\mathbf{BAC}[i] = \sum_{n=0}^{31} \mathbf{r}[i-n] \times \mathbf{r}[i-n-64] + \mathbf{r}^*[i-n-128] \quad (1)$$

When a successful estimation is performed, a `BD_FLAG` is sent to the next blocks in the datapath.

• **Channel estimation:** Real-time channel measurements are a critical feature for wireless experimentation. MIMO precoding, beam training mechanisms, rate adaptation, and interference mitigation techniques are just a few examples of applications that require accurate and timely knowledge of channel conditions. While there are many different methods to perform channel estimation in MIMO systems, the TRN fields composed of complementary Golay sequences facilitate simple time-domain channel estimation using fast Golay correlators [16]. Taking this into account, we modify the design from [16] to match the weight values of the correlator to the ones defined in the IEEE 802.11ay standard for the different spatial streams [33]. This block is able to generate the correlation against the Ga_{128} and Gb_{128} using the same structure. Besides, it only requires registers and adders without complex multipliers, which helps to save hardware resources.



(a) Receiver-side blocks **(b) Channel Estimation Block**

Figure 7: High-level receiver block diagram (a) and structure of the channel estimation block (b).

TRN fields are composed of complementary Golay sequences with the following structure:

$$\left\{+\mathrm{Ga}_{128}^{(i)}; -\mathrm{Gb}_{128}^{(i)}; +\mathrm{Ga}_{128}^{(i)}; +\mathrm{Gb}_{128}^{(i)}; +\mathrm{Ga}_{128}^{(i)}; -\mathrm{Gb}_{128}^{(i)}\right\} \quad (2)$$

where (i) corresponds to each spatial stream. Specifically for MIMO 4x4 systems, a TRN field comprises 2 time slots, where the second time slot is sent with the same sign for streams 1 and 2, and with the opposite sign for streams 3 and 4.

The high-level structure of the designed block is presented in Fig. 7b. It performs the Channel Impulse Response (CIR) estimation over each individual complementary sequence and averages the results to reduce the noise. To this end, we include configurable inverters which handle the sign differences of the sequences. We also include configurable shift registers to align the $\mathrm{Ga}_{128}^{(i)}$ and $\mathrm{Gb}_{128}^{(i)}$ sequences before adding them.

After adding the complementary sequences, we include a serial to parallel converter which prepares the samples to be accumulated during the TRN field. We further include a magnitude computation block which precedes the main path extraction block. The main path amplitude value is stored in a FIFO memory which is read by the processor to extract the values corresponding to the different TRN fields in the packets. It is worth to mention that beside the magnitude computation, the user can include a phase computation block which can run in parallel to this block. Besides, although we only extract the main path using a block to find the maximum, the block can be easily extended to extract a larger number of paths (e.g., for equalization purposes), using L-Max finder structures like the ones from [12].

For a transmitter with 4 spatial streams, 4 structures like the one from Fig. 7b are required to compute the MIMO channel (one per stream). It is important to remark that since all the block is hardware implemented, the latency for a full MIMO CIR computation is as extremely low, around 1.47 μs, measured from the moment the first sample of the TRN enters the Golay correlator until the CIR values are stored in the FIFO memory.

As can be seen from Fig. 7a, it is possible to add signal processing blocks to the datapath without affecting the system's modularity or requiring modifications to the Rx channelizer. This maintains the advantages of a memory-based design while augmenting functionality to support certain real-time operations.

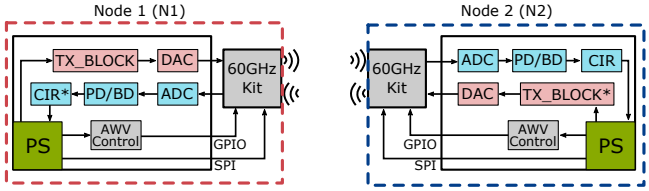


Figure 8: Hardware blocks for the closed-loop configuration

3.3 Transceiver design

It is possible to combine the transmitter and receiver hardware accelerators in a single design, to have full transceiver capabilities as shown in Fig. 8. One of the nodes (N1) from Fig. 8 can be set up to periodically transmit frames while the second node (N2) is listening in receiver mode. Note that N1 can use the AWV Control block to setup different antenna configurations to transmit the packet, e.g., when transmitting packets containing TRN fields. Once N2 detects and synchronizes with the received frame, it sends a command to the PS, using interrupts, to read the channel measurements and take a decision based on the values read from the decoder in the PL. After that, N2 is able to reconfigure its RF front-ends as transmitter using the SPI interface described in Section 2.3. Once the reconfiguration is executed, N2 is able to transmit (feedback) packets back to N1 triggering the TX BLOCK in Fig. 8. Repeating the process above, allows to establish a closed-loop communication link between the two MIMO nodes, for example to reconfigure the system in real-time based on the channel measurements. These features turn *MIMORPH* into a powerful experimentation platform that is suitable for a wide variety of experiments, for example to test beam training mechanisms, multi-band location systems, and rate adaptation algorithms, among others.

4 EVALUATION

In this section we show a range of example experiments with *MIMORPH* to demonstrate the capabilities of the system. All experiments were carried out in an indoor 7.5m×6m laboratory environment with walls, windows and furniture, as shown in Fig. 9. We also tested the 60 GHz front-end in outdoor environments and reached distances close to 50m for low MCS values.

First, we test the capabilities of *MIMORPH* using the general structure from Section 2, i.e., with offline generation and decoding



Figure 9: Setup showing two 4x4 MIMO mm-wave nodes.

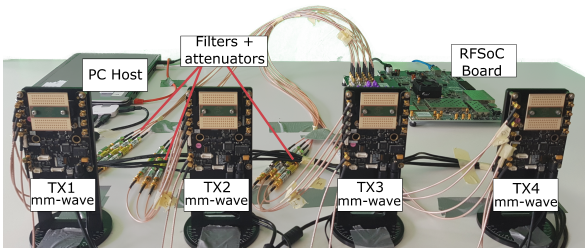


Figure 10: Node with a 4x4 mm-wave MIMO configuration.

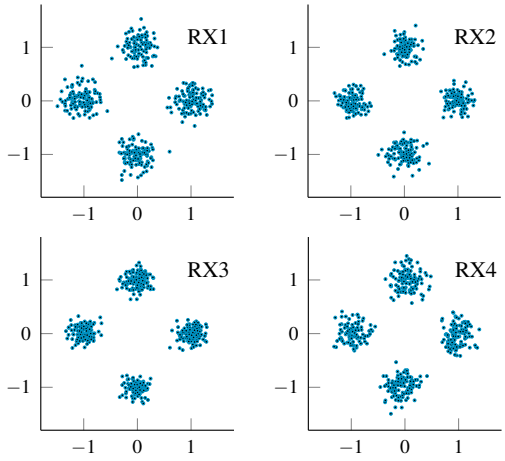


Figure 11: I/Q constellations for 4x4 MIMO using $\pi/2$ QPSK with rate 13/16 LDPC coding

of frames, using different configurations for the macro channels, showing the testbed capabilities for heterogeneous wireless MIMO experimentation.

4.1 mm-wave 4x4 MIMO setup

This configuration imposes the highest load on the system, not only in writing speed to the DDR memory but also for the signal processing blocks integrated in the design. The evaluation is performed for the two schemes of Section 2.2, using either inter-frame spacing or reduced sample resolution. Both systems use the configuration from Fig. 2a with all 4 macro channels enabled. The 60GHz RF front-ends of a node are synchronized using an external 45MHz clock source. We use independent clock sources for the antennas at transmitter and receiver (i.e., the nodes are not synchronized). This clock is independent from the clock of the RFSoC board. The sampling frequency of the AD/DA converters is set to 3.52 GHz to transmit and capture IEEE 802.11ad/ay compliant frames using two samples per symbol. The transmit datapath uses a 220 MHz clock frequency with an SSR factor of 16 and the receiver datapath uses a 440MHz clock and an SSR factor of 8.

Multiple frames are generated offline using MATLAB for a subset of the Single Carrier (SC) MCSs of the IEEE 802.11ay standard [33]. We further use the frame structure that includes an Enhanced Directional Multi Gigabit (EDMG) CEF as defined in the IEEE 802.11ay standard, which is necessary for equalization and symbol detection. The frames are sent to *MIMORPH* using the functions explained in Section 2.3. The antenna arrays are deployed as shown in Fig. 10 with a separation of 15 cm between each of the 60 GHz antenna arrays. We vary the power of the transmit amplifiers on the EVK06002 kits to obtain different SNRs for a Line-of-Sight (LOS) link. The separation between the transmitter node and the receiver is around 3 m (Fig. 9).

For the first experiment we conservatively set the inter-frame separation to be 2.1 times the length of the frame. Fig. 11 shows the constellation of the received I/Q samples for MCS9 frames, after detection and equalization using a MATLAB software model. The constellation points can be easily distinguished. Having access to raw I/Q samples to measure MIMO channels opens ample opportunities for researchers working on efficient channel estimation, hybrid precoding schemes, MIMO beam training, among others.

We perform Bit Error Rate (BER) analysis for the full 16-bit resolution and a reduced resolution of 5 bits, the maximum resolution without inter-frame spacing. The results are presented in Fig. 12 (top) and Fig. 13 (top), respectively. As expected, there is a performance loss for the lower resolution system, but communication with lower MCSs is possible. These graphs allow to estimate MCS vs. Signal-to-Noise Ratio (SNR) for a target BER / Packet Error Rate (PER), depending on the sample resolution used in the design.

Figs. 12 (bottom) and 13 (bottom) show the raw throughput (ignoring header overhead and medium access delay) for the two different sample resolutions. Note that despite the higher BER, the low resolution system can reach higher raw throughput for some MCS configurations since it does not have inter-frame spacing. Besides BER and throughput, we also compare estimated CFO and CIR for the 16-bit and 5-bit sample resolution. Fig. 14a shows the Cumulative Distribution Function (CDF) of the absolute difference between the CFO computed with the full resolution and the 5-bit I/Q samples for high (18dB), medium (10dB) and low SNR (3dB) scenarios. As can be seen, the differences are below 13 kHz in all cases, with 90% of the values below 8.5 kHz. Such small differences do not impact CFO compensation performance for 60 GHz carrier frequencies. We also analyze the CIR estimation with full and 5-bit resolution in Fig. 14. The values are very similar for both cases with only small differences in the amplitude values. The relative CIR error between both approaches is less than 12% in 90% of the cases for different SNR values.

There is ample room to improve the implementation by moving further processing blocks from software to the FPGA datapath, which reduces the rate requirements for the DDR memory. For low resolution systems, more sophisticated signal processing techniques specifically designed for low-resolution samples can be implemented to improve the BER for the high MCS cases shown

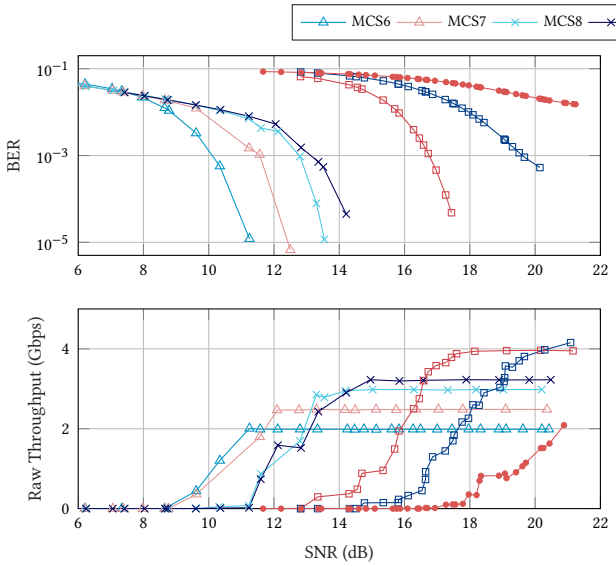


Figure 12: BER / throughput analysis for 4x4 mm-wave MIMO (16-bit samples)

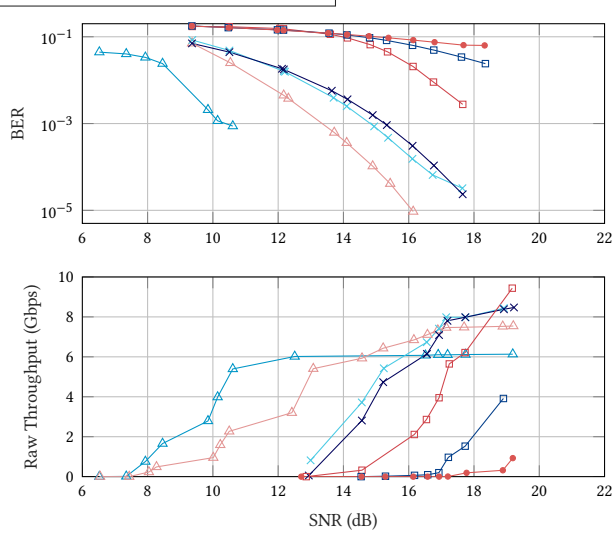
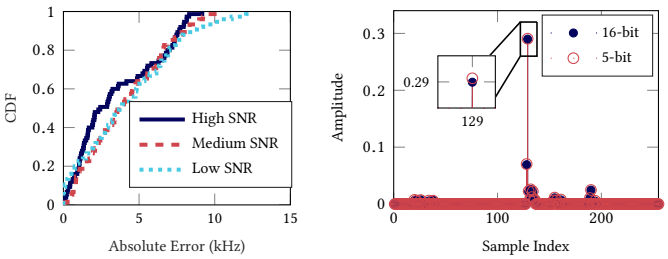


Figure 13: BER / throughput analysis for 4x4 mm-wave MIMO (5-bit samples)



(a) CFO error for 5-bit samples **(b) CIR comparison**

Figure 14: Performance with low sample resolution

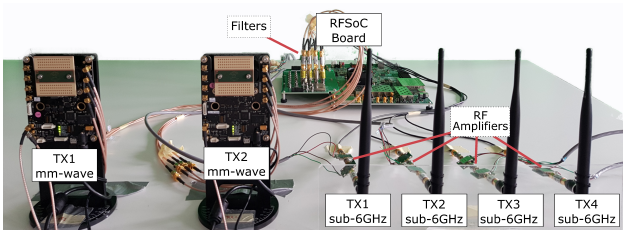


Figure 15: Node with a multi-band 2x2 MIMO mm-wave + 4x4 MIMO sub-6 GHz configuration.

in Fig. 13. Finally, it is possible to implement mixed resolution designs with low-resolution samples for some signal processing blocks where this only marginally degrades performance and high-resolution samples for other blocks. This allows to reduce area requirements and increase the speed of the processing blocks.

4.2 Multi-band Evaluation

Multi-band technologies are being considered for 5G-NR systems and are a very compelling use case for *MIMORPH*. To this end, we configure a 2x2 mm-wave MIMO setup with 60 GHz front-ends together with a 4x4 sub-6 GHz system. The system components for this configuration are shown in Fig. 15. The only external components used for the sub-6GHz system part are simple power amplifiers and band-pass filters, besides the antennas. The transmitter and receiver part of the platform was configured as shown in Fig. 1b (right) and 2c, respectively.

For the mm-wave system, we generate samples with the same structure as the ones used in Section 4.1. For the sub-6 GHz system we generate IEEE 802.11ax frames carrying 48000 information bits for 160 MHz channel bandwidth and a carrier frequency of 2.4 GHz. Different carrier frequencies, bandwidth and frames can be configured in this setup without requiring further modifications to the system. Following the results from Fig. 5 and considering that we use three macro channels, we set an inter-frame spacing of 1.3 times the length of the IEEE 802.11ax frames. Fig. 17 shows an example of the received I/Q samples for the 6 spatial streams. Due to the different data rates, multiple mm-wave frames are captured in the mm-wave MIMO link while receiving a single sub-6GHz frame.

We perform over-the-air transmissions with the multi-band configuration and analyze the BER and data throughput for different MCS and SNR values. Results for the MIMO 4x4 2.4GHz interface are shown in Fig. 16 for the BER (top) and throughput (bottom). It can be seen that for a sufficiently high SNR the BER drops to very low values which translates into 2 GBit/s of actual data throughput for MCS9 frames.

We tested further configurations but leave out the results due to space constraints. For example, it is possible to set up an 8x8 MIMO configuration with direct up/down conversion using the on-chip NCOs. We tried this configuration over-the-wire with up

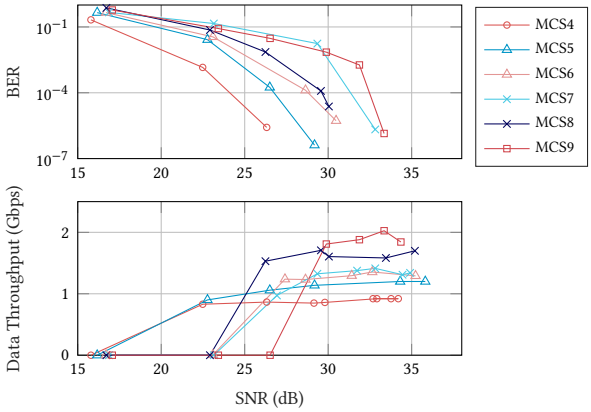


Figure 16: BER / Throughput analysis for 4x4 IEEE 802.11ax MIMO frames (160 MHz bandwidth)

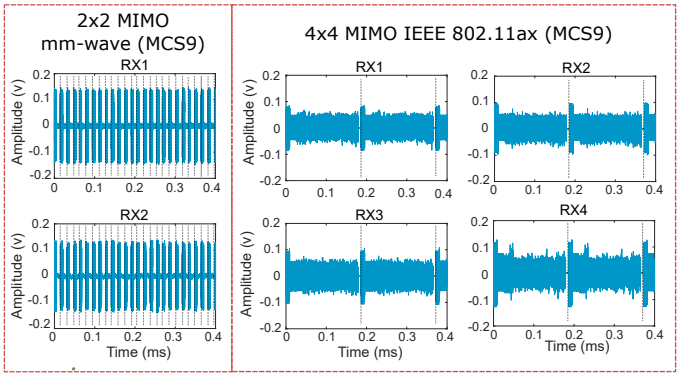


Figure 17: Received I/Q samples for the multi-band 2x2 mm-wave MIMO + 4x4 sub-6GHz MIMO system.

to 256-QAM configurations, since the daughter board attached to the RFSoC does not support 8 streams for over-the-air transmission.

4.3 Real-time closed-loop evaluation

To showcase the closed-loop processing capabilities of the platform, we implement a real-time mm-wave MIMO beam alignment mechanism, which is able to align two *MIMORPH* devices using a single packet in mobile scenarios. To this end, we deploy a pair of *MIMORPH* nodes, each one with 4 spatial streams to form a 4x4 mm-wave MIMO link, as shown in Fig. 10. One node (*N1*) is configured to send IEEE 802.11ay BRP-TX/RX like packets every 250 ms, each one including 32 TRN fields. Note that the time between packets and the number of TRN fields are configurable from the host controller processor. BRP-TX/RX packets [33] with TRN fields can be used for *simultaneous* training of a transmitter and a receiver. First, a sub-set of TRN fields are used to perform transmit training while the receiver is listening in a fixed configuration. Then, for the remaining TRN fields, the transmitter keeps a fixed antenna configuration, while the receiver changes through different antenna configurations.

For this experiment, we choose 16 AWVs from the transmitter and receiver codebooks to cover the range from $[-25^o, 25^o]$. The real-time beam alignment mechanism can be summarized in the following steps:

• **N1→ N2:** Node N1 is configured to transmit BRP-TX/RX packets each 250 ms using a configurable trigger from the PS. Once a transmission request is triggered, N1 starts sending the packet changing the AWV for the first 16 TRN fields of the packet. After that, it returns to the directional AWV selected by the processor, to continue the packet transmission. Once it ends, the processor uses the SPI interface to change from transmit to receive mode and waits for an ACK frame from N2. If N1 does not receive a packet after 250 ms, it starts the transmission of a new BRP-TX-RX packet.

• **N2 processing:** Node N2 is continuously listening to the channel using the packet detector block. Once it receives the packet, the boundary detection blocks synchronize with the end of the preamble, sending a trigger to the channel estimation block to start processing the TRN fields. At the same time, it sends a trigger to the AWV control block to start changing the receive beam patterns for the second half of the TRN fields of the packet. The CIR values $h_{i,j}\ \forall i = 1, ..., 4,\ j = 1, ..., 4$ for each packet are stored in the output FIFO memory for each channel estimation block, as shown in Fig. 7b. They are then read by the PS by means of an interruption routine triggered once the CIR processing is done. In this routine, the processor computes the best AWV configuration to be used by each one of the streams in both N1 and N2 nodes.

• **N2→ N1:** Node N2 configures its antennas from receive to transmit mode and updates the corresponding registers to use the *best* AWVs computed not only to transmit the ACK packet but also to receive the next training packet. After that, N2 sends an ACK packet (via all spatial streams) that includes the best AWV configuration to be used by N1 for the next transmission. The structure of this packet will be explained next. Finally, N2 changes from transmit to receive mode in order to wait for the next packet from N1.

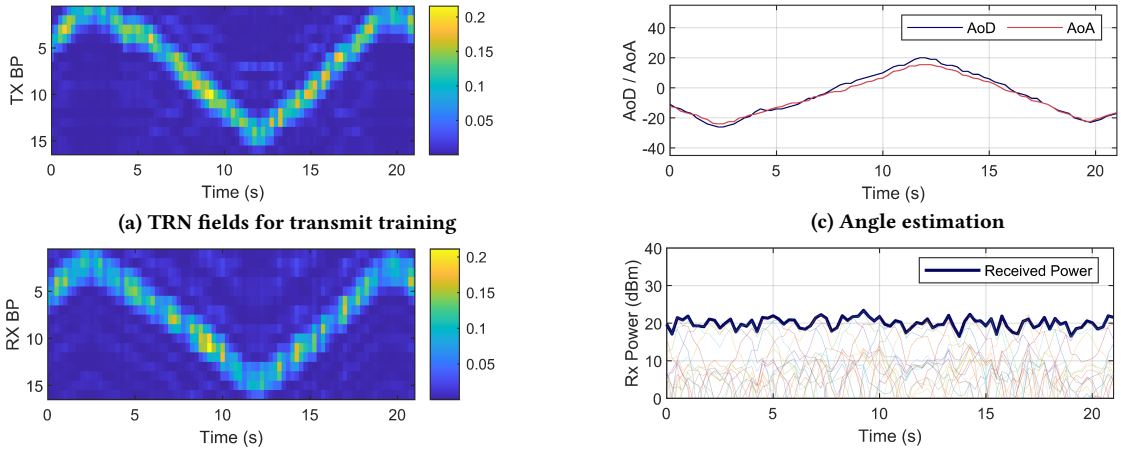
• **N1 processing:** Once N1 successfully detects and decodes the ACK packet, it updates the AWV that will be used to transmit and receive the next packet. After that, it changes from receive to transmit mode for all antennas and then it waits for the configured time (250 ms) before transmitting the next packet.

This process allows to align both *MIMORPH* nodes with a single training packet, using the real-time capabilities of the platform. Since N1 simply triggers the packet transmission periodically, it is able to recover from an ACK loss, e.g., due to blockage.

To simplify ACK decoding, we encode the AWV information in the ACK message (per spatial stream) using Golay sequences. Let X be the 6-bit message encoding the AWV configuration. We encode the k-th bit x_k of X using a pair of complementary Golay sequences as

$$x_k^{(i)} \rightarrow \left\{ \mathrm{Ga}_{128}^{(i)};\ (-1)^{x_k} \cdot \mathrm{Gb}_{128}^{(i)} \right\},$$

where i corresponds to the spatial stream. This way, at the receiver, the correlation of the complementary sequences has a high-peak for a 1 and a low-peak for a 0. Although simple, this encoding is robust thanks to the excellent auto-correlation properties of the Golay sequences and allows to reuse the channel estimation block from Section 3.2 (with minor modifications) to decode the ACK packet, instead of having to implement a full real-time decoder.



(a) TRN fields for transmit training

(b) TRN fields for receive training

(c) Angle estimation

(d) Receive power

Figure 18: Real-time MIMO beam alignment for lateral mobility

This experiment uses mobility, with node N1 moving back and forth laterally while node N2 remains in place. The results are shown in Fig. 18. We include heatmaps in Figs. 18a and 18b showing how the maximum MIMO CIR values ($h_{i,i}\ \forall\ i = 1, ..., 4$) for the corresponding 16 TRN subfields of the packet change over time. We only show results for one of the spatial streams, since all of them behave similarly. As expected, the heatmap plots for TRN fields corresponding to the transmit training (Fig. 18a) and the ones corresponding to receive training (Fig. 18b) for the different beam patterns change over the duration of the experiment according to the movement.

In addition, using the measured CIR values for the different beam patterns we can compute the Angle of Arrival (AoA) and Angle of Departure (AoD) for each one of the spatial streams using a correlation-based approach similar to the one used in [11, 28]. As can be seen in Fig. 18c, the results are similar for both angle values and match the trajectory followed during the experiment. This capability can be used to implement more sophisticated low-overhead beam selection algorithms based on the estimated angles. It also serves as baseline for the implementation of localization and environment sensing systems. In order to quantify the angle estimation accuracy, we perform stationary experiments for different angles and positions and observe a median angle accuracy below 1° and a maximum error of 4°.

Finally, Fig. 18d shows the receive power for each one of the training packets with the selected antenna configuration (dark line). For comparison, we also include the power for the individual AWVs corresponding to a fixed beam configuration. As can be seen, our system consistently chooses the best beam pattern pairs to communicate for all the spatial streams. Furthermore, we observe that the receive power is not affected due the movement, demonstrating that the platform rapidly adapts in the dynamic environment.

An important aspect for closed-loop systems like *MIMORPH* is the capability to quickly react to received packets and reconfigure the system to transmit an ACK packet. Our system is built using development kits (both for baseband and the RF front-ends), which are designed for *proof-of-concept* systems. Specifically, apart from the GPIO interface used for fast AWV updates, the remaining antenna commands to implement the closed-loop system are sent via the slower SPI interface. Besides, due to signal integrity issues and limited GPIO pins availability on the RFSoC board, it is necessary to configure the antennas one at a time. In total, we need to execute a minimum of 3 commands per RF-front end to update the AWV pointer, change from receive to transmit mode and update the AWV to return after TRN processing. Besides, the CIR values are read by the PS one by one using AXI-lite interfaces. Finally, the beam selection algorithm is implemented in the PS in a sequential manner. Due to these factors, our platform achieves a turn-around time of 700 μs. While this latency is two orders of magnitude higher than the SIFS value defined in the IEEE 802.11ad/ay standard (3 μs), our system provides an important step forward towards a standard compliant experimentation platform. Latency values can be reduced by means of adding a GPIO daughter card to the RFSoC board to concurrently reconfigure all RF front-ends, improving the signal integrity, and increasing the SPI clock frequency. In addition, moving more functional blocks to hardware, such as the beam selection algorithm, would allow to get closer to the SIFS values defined by the standards.

5 RELATED WORK

While a significant number of experimental platforms for sub-6 GHz and mm-wave research have been proposed, it is difficult to find solutions that cover the diverse requirements of today's and future systems. For sub-6GHz research, [10, 34] provide useful platforms with fully implemented physical and MAC layer designs for real-time operation (at limited bandwidths and for low-order MIMO). Despite the enormous design effort, their flexibility is limited and any modifications require a substantial redesign because they use custom interfaces. To the best of our knowledge, there are no flexible implementations with standard interfaces available for mm-wave systems, the main reason being that the design complexity scales with the bandwidth and architectures suitable for sub-6GHz systems require a full redesign to be used for high bandwidth mm-wave systems. USRP devices gained high popularity in

Table 1: Comparison between existing wireless experimentation platforms

Platform	MIMO	Sub-6GHz	mm-wave	Real-time	Closed-loop	Free/open	Baseband architecture	Cost
Open-WiFi [10]	✗	✓	✗	✓	✓	✓	SoC	$5.5K
Tick [34]	✓	✓	✗	✓	✗	✗	FPGA + PC	$4K
MiRa [1]	✗	✗	✓	✗	✗	✗	USRP + PC	> $10K
X60 [24]	✓	✗	✓	✓	✓	✗	Multiple FPGA + Processor Cards	> $200K
mm-FLEX [11]	✗	✗	✓	✓	✗	✓	FPGA + Processor Cards	$40K
m-cube [39]	✓	✗	✓	✗	✗	✓	SoC	$15K
MillimeTera [22]	✗	✗	✓	✗	✗	✓	RFSoC	$29K
MIMORPH	✓	✓	✓	✓	✓	✓	RFSoC	$20K

the research community thanks to their affordability and ease of use. These devices have also been widely deployed in laboratories that provide remote access to perform large-scale network-level experiments, e.g., in ORBIT [23]. A single device can be used to implement a low-order MIMO system in the sub-6GHz band and they can be stacked for higher-order MIMO [26]. However, stacking introduces additional complexity and imposes hard restrictions on the back plane and the controllers that manage the system, due to the amount of data that needs to be exchanged among the nodes of the system. Although USRPs have been used for mm-wave experimentation [1, 19], their usefulness is limited due to their severe bandwidth limitations that prevent operation with bandwidths anywhere near those used by current mm-wave standards.

For mm-wave systems, experimentation has focused mostly on SISO channels due to the lack of suitable hardware for MIMO systems. The X60 system is based on a commercial platform implementing physical and MAC layers, integrating either horn antennas or small phased arrays. It was designed for SISO channels but has also been used for simple low-order MIMO experiments [7]. The mm-FLEX platform [11] includes fast antenna reconfiguration and full IEEE 802.11ad capability, but a potential extension of that system to multiple spatial streams would require multi-FPGA synchronization. Using it for fast *in-packet* training required by IEEE 802.11ay is difficult due to the appreciable latency between the converters and the FPGA of that specific system.

There are very few works on experimentation platform suitable for mm-wave MIMO systems [4, 22, 39]. The m-cube platform [39] makes use of an FPGA-based baseband processor attached to phased antenna arrays from commercial devices. Control commands to configure the antennas are sent from a dedicated FPGA. It has only been tested with reduced bandwidth baseband processors [2] and it requires to build custom-made bridge boards which add complexity to the system. In contrast, *MIMORPH* is built around off-the-shelf components that can be easily acquired by the potential users of the platform.

The work in [4] discusses a multi-FPGA platform for low-order 2x2 MIMO systems. However, its scalability is limited due to the synchronization and calibration overhead needed for the multiple stacked FPGAs. The MillimeTera testbed from [22] is a fully digital MIMO platform based on an RFSoC FPGA system. It integrates a custom designed daughter board including a phased antenna array. The platform is showcased using a simple example design from Xilinx, limiting the scenario to very short channel measurements. While the platform is interesting for the future once it is fully developed, it does not support analog/hybrid beam forming and thus is of limited use for IEEE 802.11ad/ay research. We summarize the key differences between different platforms from literature and our *MIMORPH* system in Table 1. As can be seen, the features of *MIMORPH* combine the advantages of several of the other platforms and thus make it a highly interesting solution for wireless experimentation. Its functionality can be further extended by integrating additional hardware accelerators.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented *MIMORPH*, a highly flexible and performant experimental platform for mm-wave and sub-6 GHz MIMO research. The system is easy to use and can be configured for a wide range of different use cases. We demonstrated several of such use cases by means of a series of testbed experiments. Our platform is ideally suited for IEEE 802.11ax and IEEE 802.11ay as well as 5G-NR research and has unique capabilities not offered by any other MIMO research platform. Our system specification and implementation are made available as open-source, to foster a community platform for cutting-edge wireless networking research.

For future work, we intend to implement further IEEE 802.11ay functionality, and in particular implement fully standard-compliant IEEE 802.11ay MIMO beam training using the hardware blocks presented in this paper. The joint mm-wave and sub-6 GHz configuration also allows for a highly efficient implementation of fast session transfer, rapidly moving GBit/s data streams from one interface to the other, depending on the instantaneous changes in the channels. Finally, we intend to port the platform to the new 16-channel RFSoC system [37] for even higher performance.

ACKNOWLEDGMENTS

This research work was sponsored in part by the European Union's Horizon 2020 research and innovation programme under Grant No. 871249 (LOCUS), the Spanish Ministry of Science and Innovation (MICIU) grant RTI2018-094313-B-I00 (PinPoint5G+) and the Region of Madrid through TAPIR-CM (S2018/TCS-4496).

REFERENCES

[1] Omid Abari, Haitham Hassanieh, Michael Rodriguez, and Dina Katabi. 2016. Poster: A Millimeter Wave Software Defined Radio Platform with Phased Arrays. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 419–420. https://doi.org/10.1145/2973750.2985258

[2] Narendra Anand, Ehsan Aryafar, and Edward W Knightly. 2010. WARPlab: a flexible framework for rapid physical layer design. In *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students*. 53–56.

[3] ARM. 2019. *AMBA AXI and ACE Protocol Specification*. https://static.docs.arm.com/ihi0022/g/IHI0022G_amba_axi_protocol_spec.pdf.

[4] S. Blandino, G. Mangraviti, C. Desset, A. Bourdoux, P. Wambacq, and S. Pollin. 2019. Multi-User Hybrid MIMO at 60 GHz Using 16-Antenna Transmitters. *IEEE Transactions on Circuits and Systems I: Regular Papers* 66, 2 (2019), 848–858.

[5] Sourjya Dutta, C Nicolas Barati, David Ramirez, Aditya Dhananjay, James F Buckwalter, and Sundeep Rangan. 2019. A case for digital beamforming at mmWave. *IEEE Transactions on Wireless Communications* 19, 2 (2019), 756–770.

[6] Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly. 2017. IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi. *IEEE Communications Magazine* 55, 12 (2017), 186–192.

[7] Yasaman Ghasempour, Muhammad K. Haider, Carlos Cordeiro, Dimitrios Koutsonikolas, and Edward Knightly. 2018. Multi-Stream Beam-Training for MmWave MIMO Networks. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom âĂŹ18)*. Association for Computing Machinery, New York, NY, USA, 225âĂŞ239. https://doi.org/10.1145/3241539.3241556

[8] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review* 41, 1 (2011), 53–53.

[9] Y. Huang, W. Liu, and S. Jou. 2011. Design and implementation of synchronization detection for IEEE 802.15.3c. In *Proceedings of 2011 International Symposium on VLSI Design, Automation and Test*. 1–4.

[10] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman. 2020. openwifi: a free and open-source IEEE 802.11 SDR implementation on SoC. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 1–2.

[11] Jesus O. Lacruz, Dolores Garcia, Pablo Jiménez Mateo, Joan Palacios, and Joerg Widmer. 2020. mm-FLEX: An Open Platform for Millimeter-Wave Mobile Full-Bandwidth Experimentation. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (MobiSys âĂŹ20)*. Association for Computing Machinery, New York, NY, USA, 1âĂŞ13. https://doi.org/10.1145/3386901.3389034

[12] J. O. Lacruz, F. Garcia-Herrero, M. J. Canet, and J. Valls. 2016. Reduced-Complexity Nonbinary LDPC Decoder for High-Order Galois Fields Based on Trellis MinâĂŞMax Algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24, 8 (2016), 2643–2653. https://doi.org/10.1109/TVLSI.2016.2514484

[13] Yao-Yu Li, Chi-Yu Li, Wei-Han Chen, Chia-Jui Yeh, and Kuochen Wang. 2017. Enabling seamless WiGig/WiFi handovers in tri-band wireless systems. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. 1–2. https://doi.org/10.1109/ICNP.2017.8117571

[14] Chun-Yi Liu, Meng-Siou Sie, Edmund W. J. Leong, Yu-Cheng Yao, Chih-Wei Jen, Wei-Chang Liu, Chih-Feng Wu, and Shyh-Jye Jou. 2017. Dual-Mode All-Digital Baseband Receiver With a Feed-Forward and Shared-Memory Architecture for Dual-Standard Over 60 GHz NLOS Channel. *IEEE Transactions on Circuits and Systems I: Regular Papers* (2017).

[15] W. Liu, T. Wei, Y. Huang, C. Chan, and S. Jou. 2015. All-Digital Synchronization for SC/OFDM Mode of IEEE 802.15.3c and IEEE 802.11ad. *IEEE Transactions on Circuits and Systems I: Regular Papers* 62, 2 (2015), 545–553.

[16] Wei-Chang Liu, Fu-Chun Yeh, Ting-Chen Wei, Ching-Da Chan, and Shyh-Jye Jou. 2013. A Digital Golay-MPIC Time Domain Equalizer for SC/OFDM Dual-Modes at 60 GHz Band. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS* 60, 10 (2013), 10.

[17] lwIP developers group. 2020. *lwIP: open-source TCP/IP stack designed for embedded systems*. https://www.nongnu.org/lwip/2_1_x/.

[18] Steffen Malkowsky, Joao Vieira, Liang Liu, Paul Harris, Karl Nieman, Nikhil Kundargi, Ian C Wong, Fredrik Tufvesson, Viktor Öwall, and Ove Edfors. 2017. The worldâĂŹs first real-time testbed for massive MIMO: Design, implementation, and validation. *IEEE Access* 5 (2017), 9073–9088.

[19] Adriana Moreno, Jesus O. Lacruz, and Joerg Widmer. 2020. Open Source RFNoC-Based Testbed for Millimeter-Wave Experimentation Using USRP Software Defined Radios. In *International Symposium on Circuits and Systems (ISCAS-2020)*.

[20] P. Murphy, A. Sabharwal, and B. Aazhang. 2006. Design of WARP: A Flexible Wireless Open-Access Research Platform. In *Proceedings of EUSIPCO*.

[21] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. 2014. OpenAirInterface: A Flexible Platform for 5G Research. *SIGCOMM Comput. Commun. Rev.* 44, 5 (Oct. 2014), 33âĂŞ38. https://doi.org/10.1145/2677046.2677053

[22] Michele Polese, Francesco Restuccia, Abhimanyu Gosain, Josep Jornet, Shubhendu Bhardwaj, Viduneth Ariyarathna, Soumyajit Mandal, Kai Zheng, Aditya Dhananjay, Marco Mezzavilla, James Buckwalter, Mark Rodwell, Xin Wang, Michele Zorzi, Arjuna Madanayake, and Tommaso Melodia. 2019. MillimeTera: Toward A Large-Scale Open-Source MmWave and Terahertz Experimental Testbed. In *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNetsâĂŹ19)*. Association for Computing Machinery, New York, NY, USA, 27âĂŞ32. https://doi.org/10.1145/3349624.3356764

[23] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. 2005. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In *IEEE WCNC*.

[24] Swetank Kumar Saha, Yasaman Ghasempour, Muhammad Kumail Haider, Tariq Siddiqui, Paulo De Melo, Neerad Somanchi, Luke Zakrajsek, Arjun Singh, Roshan Shyamsunder, Owen Torres, et al. 2019. X60: A programmable testbed for wideband 60 GHz WLANs with phased arrays. *Computer Communications* 133 (2019), 77 – 88. https://doi.org/10.1016/j.comcom.2018.09.005

[25] Omid Semiari, Walid Saad, Mehdi Bennis, and Merouane Debbah. 2017. Performance Analysis of Integrated Sub-6 GHz-Millimeter Wave Wireless Local Area Networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–7. https://doi.org/10.1109/GLOCOM.2017.8254610

[26] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. 2012. Argos: Practical many-antenna base stations. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. 53–64.

[27] SIVERSIMA. 2020. *EVK06002 Development Kit*. https://www.siversima.com/product/evk-06002-00/.

[28] Daniel Steinmetzer, Daniel Wegemer, Matthias Schulz, Joerg Widmer, and Matthias Hollick. 2017. Compressive millimeter-wave sector selection in off-the-shelf IEEE 802.11 ad devices. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. 414–425.

[29] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 28âĂŞ41. https://doi.org/10.1145/3117811.3117817

[30] 3GPP TS38.300. 2018. NR and NG-RAN Overall Description; Stage 2 (Release 15). *V15.2.0 (2018-06)* (2018).

[31] IEEE 802.11 working group. 2012. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. *IEEE Standard 802.11ad* (2012).

[32] IEEE 802.11 working group. 2017. IEEE P802.11ax/D2.0 Standard for Information Technology - Telecommunications and Information Exchange between Systems Local and Metropolitan Area NetworksâĂŞ Specific RequirementsâĂŞ Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)SpecificationsâĂŞAmendment 6: Enhancements for High Efficiency WLAN. *Standard IEEE 802.11ax* (2017).

[33] IEEE 802.11 working group. 2019. IEEE Draft 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: Enhanced Throughput for Operation in License-Exempt Bands Above 45 GHz. *IEEE P802.11ay/D3.0* (2019).

[34] Haoyang Wu, Tao Wang, Zengwen Yuan, Chunyi Peng, Zhiwei Li, Zhaowei Tan, Boyan Ding, Xiaoguang Li, Yuanjie Li, Jun Liu, and Songwu Lu. 2017. The Tick Programmable Low-Latency SDR System *(MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 101âĂŞ113. https://doi.org/10.1145/3117811.3117834

[35] Xilinx. 2020. *UltraScale Architecture-Based FPGAs Memory IP v1.4*. https://www.xilinx.com/support/documentation/ip_documentation/ultrascale_memory_ip/v1_4/pg150-ultrascale-memory-ip.pdf.

[36] Xilinx. 2020. *Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit*. https://www.xilinx.com/products/boards-and-kits/zcu111.html.

[37] Xilinx. 2020. *Zynq UltraScale+ RFSoC ZCU216 Evaluation Kit*. https://www.xilinx.com/products/boards-and-kits/zcu216.html.

[38] Jiayi Zhang, Linglong Dai, Xu Li, Ying Liu, and Lajos Hanzo. 2018. On Low-Resolution ADCs in Practical 5G Millimeter-Wave Massive MIMO Systems. *IEEE Communications Magazine* 56, 7 (2018), 205–211. https://doi.org/10.1109/MCOM.2018.1600731

[39] Renjie Zhao, Timothy Woodford, Teng Wei, Kun Qian, and Xinyu Zhang. 2020. M-Cube: A Millimeter-Wave Massive MIMO Software Radio. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom âĂŹ20)*. Association for Computing Machinery, New York, NY, USA, Article 15, 14 pages. https://doi.org/10.1145/3372224.3380892