



# LUND UNIVERSITY

## Implementation of Low-latency Signal Processing and Data Shuffling for TDD massive MIMO Systems

Malkowsky, Steffen; Vieira, Joao; Nieman, Karl; Kundargi, Nikhil; Wong, Ian; Öwall, Viktor; Edfors, Ove; Tufvesson, Fredrik; Liu, Liang

DOI:

[10.1109/SiPS.2016.53](https://doi.org/10.1109/SiPS.2016.53)

2017

*Document Version:*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Malkowsky, S., Vieira, J., Nieman, K., Kundargi, N., Wong, I., Öwall, V., Edfors, O., Tufvesson, F., & Liu, L. (2017). *Implementation of Low-latency Signal Processing and Data Shuffling for TDD massive MIMO Systems*. 260-265. Paper presented at IEEE International Workshop on Signal Processing Systems, Dallas, United States. <https://doi.org/10.1109/SiPS.2016.53>

*Total number of authors:*

9

*Creative Commons License:*

Unspecified

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00



# Implementation of Low-latency Signal Processing and Data Shuffling for TDD Massive MIMO Systems

Steffen Malkowsky<sup>1</sup>, Joao Vieira<sup>1</sup>, Karl Nieman<sup>2</sup>, Nikhil Kundargi<sup>2</sup>, Ian Wong<sup>2</sup>, Viktor Öwall<sup>1</sup>, Ove Edfors<sup>1</sup>, Fredrik Tufvesson<sup>1</sup>, and Liang Liu<sup>1</sup>

<sup>1</sup> Dept. of Electrical and Information Technology, Lund University, Sweden

<sup>2</sup> National Instruments, Austin, Texas, USA  
firstname.lastname@{eit.lth.se, ni.com}

**Abstract**—Low latency signal processing and high throughput implementations are required in order to realize real-time TDD massive MIMO communications, especially in high mobility scenarios. One of the main challenges is that the up-link and down-link turnaround time has to be within the coherence time of the wireless channel to enable efficient use of reciprocity. This paper presents a hardware architecture and implementation of this critical signal processing path, including channel estimation, QRD-based MMSE decoder/precoder and distributed reciprocity calibration. Furthermore, we detail a switch-based router implementation to tackle the stringent throughput and latency requirements on the data shuffling network. The proposed architecture was verified on the LuMaMi testbed, based on the NI SDR platform. The implementation supports real-time TDD transmission in a  $128 \times 12$  massive MIMO setup using 20 MHz channel bandwidth. The processing latency in the critical path is less than 0.15 ms, enabling reciprocity-based TDD massive MIMO for high-mobility scenarios.

**Index Terms**—Massive MIMO, TDD, Low-Latency, SDR

## I. INTRODUCTION

Modern communication systems often use the multiple-input multiple-output (MIMO) concept to enhance link performance and link reliability, a technique nowadays incorporated within Wi-Fi and LTE (Long Term Evolution) standards employing up to 8 antennas. Massive MIMO (MaMi) takes this concept further by radically increasing the number of antennas at the base station (BS) side. Theoretical results have shown that MaMi is capable of serving a multitude of user equipments (UEs) in the same time-frequency resource while simultaneously achieving high reliability, high spectral efficiency and high energy efficiency [1], [2].

MaMi also introduces many new challenges, especially if real-time operation is required. For example, it renders feedback of down-link (DL) channel estimates impractical, thus making time division duplexing (TDD) operation the more viable option [1]. However, in such a setup, the precoding turnaround time, *i.e.* the time between obtaining Channel State Information (CSI) on the up-link (UL) and transmitting precoded data on the DL, puts constraints on data shuffling

and processing latency that makes their design non-trivial. In particular, hundreds of interconnect paths among antennas and processing units need to be established and properly scheduled to ensure low-latency data shuffling with a minimum of overhead. Moreover, detection and precoding circuitry require special attention despite implementing relatively low-complex linear schemes due to the large-scale matrix operation required with massive antenna arrays.

In this paper, we detail the implementation of key signal processing blocks and data shuffling system for a  $128 \times 12$  orthogonal-frequency division multiplexing (OFDM) MaMi design. More specifically, we present (i) a high-throughput low-latency router capable of routing up to 8 inputs to up to 8 outputs, (ii) a run-time reconfigurable partially parallel Minimum Mean Square Error (MMSE) decoder/precoder implementation based on the QR-decomposition (QRD), and (iii) a distributed approach to perform reciprocity calibration. Our designs were tested and verified on field-programmable gate arrays (FPGA) in the LuMaMi (Lund University MaMi) testbed [3].

The remainder of this paper is structured as follows. In Section II we introduce the basic MaMi concepts before discussing architecture and interconnect implementation in Section III. Section IV details the decoder/precoder design and the distributed reciprocity calibration. In Section V we present the hardware utilization for a Kintex-7 FPGA and verification results. Finally, in Section VI we conclude the paper.

## II. TDD-BASED MASSIVE MIMO

The signal model of a TDD-based MaMi system for a particular OFDM subcarrier is illustrated in Fig. 1, where an  $M$ -antenna BS simultaneously serves  $K$  single-antenna UEs.

### A. Up-link Signal Model

Collecting the  $K$  simultaneously transmitted symbols in a vector  $\mathbf{z} \triangleq (z_1, \dots, z_K)^T$ , the received signals by the BS can

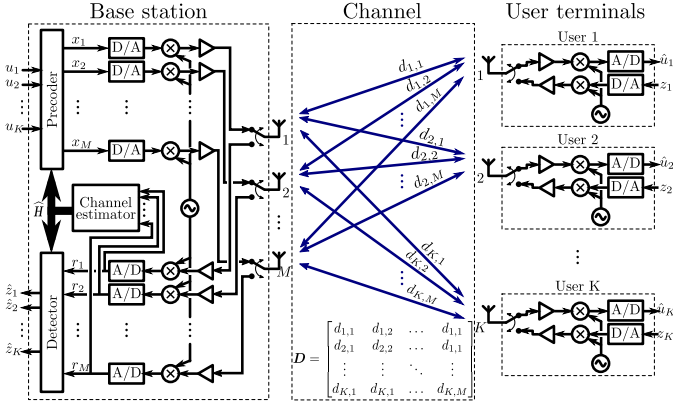


Fig. 1: A simplified massive MIMO system. Reciprocity for the propagation channel  $D$  is assumed.

be described as

$$\mathbf{r} = \mathbf{G}\sqrt{\mathbf{P}_{\text{ul}}}\mathbf{z} + \mathbf{n}, \quad (1)$$

where the matrix  $\mathbf{G}$  models the total UL channel (propagation channel and transceiver chains),  $\mathbf{P}_{\text{ul}}$  is a  $K \times K$  diagonal matrix containing the transmit power levels used by the  $K$  UEs and  $\mathbf{n}$  is a vector modeling UL noise. The estimated user symbols  $\hat{\mathbf{z}} \triangleq (\hat{z}_1, \dots, \hat{z}_K)^T$  can be obtained by linear filtering of the received signals  $\mathbf{r}$  as

$$\hat{\mathbf{z}} = \mathbf{f}_{\text{eq}}(\mathbf{G})\mathbf{r}, \quad (2)$$

where  $\mathbf{f}_{\text{eq}}(\cdot)$  constructs an appropriate equalization matrix.

### B. Down-link Signal Model

Let the vector  $\mathbf{x} \triangleq (x_1, \dots, x_M)^T$  model the precoded signals, which are transmitted in the downlink. We stack the signals received by each UE in the vector  $\hat{\mathbf{u}} \triangleq (\hat{u}_1, \dots, \hat{u}_K)^T$ , where  $\hat{u}_k$  is the received symbol at UE  $k$ . With that, the received signal vector is modeled as

$$\hat{\mathbf{u}} = \mathbf{H}\mathbf{x} + \mathbf{n}', \quad (3)$$

where the matrix  $\mathbf{H}$  models the DL channel (propagation channel and transceiver chains) and  $\mathbf{n}'$  models DL noise.<sup>1</sup>

It is well known that feedback of DL channel estimates is impractical in MaMi [1]. As a result,  $\mathbf{x}$  is based on a precoder designed from the UL channel matrix  $\mathbf{G}$ , given that a calibration step - dealing with the non-reciprocity of the channel - is performed first. This matter is addressed next.

### C. Reciprocity Calibration

The differences between the UL and DL channel can be seen by factorizing both channel matrices as

$$\mathbf{G} = \mathbf{R}_{\text{BS}}\mathbf{D}\mathbf{T}_{\text{UE}}, \quad \text{and} \quad \mathbf{H} = \mathbf{R}_{\text{UE}}\mathbf{D}^T\mathbf{T}_{\text{BS}}, \quad (4)$$

<sup>1</sup>Different nomenclature is used to reference the uplink and downlink channels, as they are not assumed to be reciprocal in this case of study.

TABLE I: Linear Precoding/Detection Schemes

Scheme	Decoding	Precoding
MRC/MRT*	$\mathbf{G}^H$	$\mathbf{H}_C^H$
ZF†	$(\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H$	$\mathbf{H}_C^H(\mathbf{H}_C\mathbf{H}_C^H)^{-1}$
MMSE‡	$(\mathbf{G}^H\mathbf{G} + \beta\mathbf{I}_K)^{-1}\mathbf{G}^H$	$\mathbf{H}_C^H(\mathbf{H}_C\mathbf{H}_C^H + \beta\mathbf{I}_K)^{-1}$

\* Maximum-Ratio Combining / Maximum-Ratio Transmission

† Zero-Forcing

‡ Minimum Mean Square Error

where  $\mathbf{D}$  represents the - reciprocal - propagation channel and the diagonal matrices  $\mathbf{R}_{\text{BS}}$ ,  $\mathbf{R}_{\text{UE}}$  and  $\mathbf{T}_{\text{BS}}$ ,  $\mathbf{T}_{\text{UE}}$  model the non-reciprocal hardware responses of the receivers and transmitters, respectively.

Let  $\mathbf{C} = \mathbf{T}_{\text{BS}}\mathbf{R}_{\text{BS}}^{-1}$  denote the, so called, calibration matrix which is assumed to be at hand for the time being. It was shown in [4], that  $\mathbf{C}$  can be used to re-establish the reciprocity assumption. In particular, there are two approaches to achieve this. The first approach is to calibrate the uplink channel  $\mathbf{G}$ . With that, the precoded signal  $\mathbf{x}$  is written as

$$\mathbf{x} = \mathbf{f}_{\text{pre}}(\mathbf{C}\mathbf{G})\mathbf{u}. \quad (5)$$

In (5), the vector  $\mathbf{u} \triangleq (u_1, \dots, u_K)^T$  contains the symbols intended for the  $K$  UEs, and  $\mathbf{f}_{\text{pre}}(\cdot)$  builds a precoding matrix. For latter use, we define  $\mathbf{H}_C \triangleq \mathbf{C}\mathbf{G}$ . The second approach is to apply the calibration in the precoded signal itself. With that,  $\mathbf{x}$  is written as

$$\mathbf{x} = \mathbf{C}^{-1}\mathbf{f}_{\text{pre}}(\mathbf{G})\mathbf{u}. \quad (6)$$

While a performance analysis of both approaches is performed in [5], we constrain our analysis and focus on the utilization of hardware resources during implementation.

### D. Linear detection & precoding schemes

TABLE I summarizes the precoding and detection schemes, i.e.  $\mathbf{f}_{\text{eq}}(\mathbf{G})$  and  $\mathbf{f}_{\text{pre}}(\mathbf{H}_C)$ , considered in our design. We remark that the precoding matrix addressed in (6) has the same form of the decoder and thus is omitted from the table. This is addressed later in the paper.

### E. System Parameter and Frame Structure

The signal processing is implemented for an OFDM-based MaMi TDD system, which main parameters are shown in TABLE II. Notice that the OFDM parameters follow LTE numerology. For illustration purposes, Fig. 2 shows the frame structure considered in this work.<sup>2</sup> Explained briefly, one radio frame of 10 ms length is divided into 20 slots, each having a length of 0.5 ms. The first slot is used for synchronization and control signaling and the rest of the slots for payload data transmission. UL channel estimation is performed during the first OFDM symbol of each slot giving a minimum supported channel coherence time of approximately 430  $\mu\text{s}$ . We remark

<sup>2</sup>However, we note that our frame structure is parametrizable, i.e. the OFDM symbols can be reordered in an arbitrary way.

TABLE II: High-level system parameters

Parameter	Variable	Value
Bandwidth	$W$	20 MHz
Carrier frequency	$f_c$	1.2 GHz - 6 GHz
Sampling Rate	$f_s$	30.72 MS/s
FFT Size	$N_{\text{FFT}}$	2048
# Used subcarriers	$N_{\text{SUB}}$	1200
Slot time	$T_s$	0.5 ms
Frame time	$T_f$	10 ms
# UEs	$K$	12
# BS antennas	$M$	128

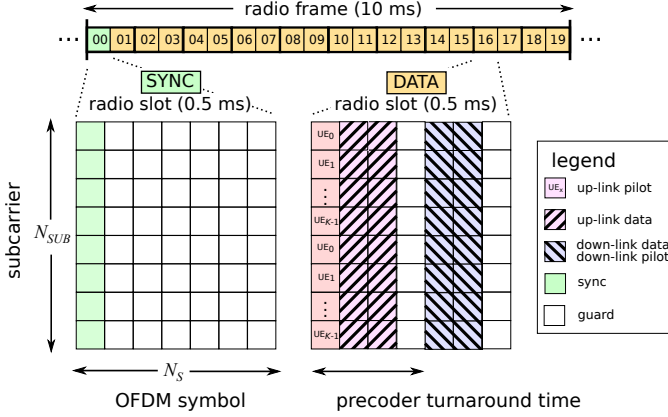


Fig. 2: Example radio frame for an OFDM based MaMi system. The first subframe is used for synchronization and control signaling while the following 19 are used for data transmission. Down-link data transmission is shared with down-link pilots to perform reciprocity calibration on the UE side.

that by supporting such channel coherence times, UE speeds up to 80 km/h can be supported in a carrier operating in the 3 GHz band, with satisfactory performance.

### III. PROCESSING ARCHITECTURE AND INTERCONNECTION NETWORK

MaMi requires coherent processing of a large number of transmit and receive signals at the BS. To achieve this, the system is build of 64 Remote Radio Heads (RRHs) Software Defined Radios (SDR) (USRP-2943R) [6], four FPGA co-processors (FlexRIO PXIe-7976R) [7] and a NI PXIe-8135 host computer [8] which are all interconnected through a PCI Express network to allow inter-FPGA as well as FPGA-host connections as shown in Fig. 3. Using the system parameter from TABLE II, the system processes 30.72 MS/s per channel  $\times$  128 channels  $\times$  4 bytes per sample (two for I- and two for Q-component) = 15.7 GB/s from the antennas in UL and DL direction.

To lower the pressure on the interconnect network, the per-antenna OFDM processing is distributed over the 64 RRHs

(each having 2 RF-chains) and the baseband sample width is decreased to 3 bytes per sample. This allows to lower bidirectional rate between the RRHs and the FPGA co-processors to be reduced to 6.5 GB/s. Due to the subcarrier independence in OFDM the MIMO processing can be distributed in frequency, allowing processing to be performed on a number of parallel FPGAs. Thus, to further balance throughput limitations, the MIMO processing for 128 channels is distributed over four FPGA co-processors each containing a Kintex 7 410T FPGA which leads to a final in/out rate of 1.62 GB/s per MIMO processor.

To manage the finite number of interconnect paths each FPGA provides, an intermediate aggregation/disaggregation stage is required. In order to limit the number of interconnect paths between the RRHs and FPGA co-processors, eight RRHs are grouped together utilizing two RRHs for data aggregation/disaggregation per group.

Data aggregation/disaggregation functionality is required on both, the RRHs and FPGA co-processors. On the UL, data from the 16 antennas within a RRH group is combined and splitted into four subbands, one for each FPGA co-processor (Antenna Combiner / BW Splitter). Next, the FPGA co-processors combine data arriving from all eight RRH groups to capture data from all 128 antennas. The DL performs the reverse operation (Antenna Splitter / BW Combiner). Data aggregation/disaggregation is implemented using a reconfigurable hardware router with it's conceptual block diagram shown in Fig. 4.

The router multiplexes a sample sourced from one of the  $N_{\text{in}}$  input FIFOs and demultiplexes this sample to one the the  $N_{\text{out}}$  output FIFOs depending on the source and destination listed in the route table. It then advances its pointer to the route table depending on route success. This ensures that no data is lost within the system and that individual samples are routed to their associated processing resource. One 64-bit sample can be routed per clock cycle and the design allows to run at a clock frequency of 200 MHz. Thus, a maximum throughput of 1.6 GB/s is achieved.

Three bitfiles are compiled for the FPGAs based on the signal flow requirements and limitations of the FPGA endpoints. The resources and configurations of each of the routers in the system are summarized in TABLE III.

The system is designed to support up to 12 UEs which are hardcoded by a dedicated orthogonal pilot pattern within the up-link pilot slot while the supported bandwidth is up to 20 MHz. Due to the modular design the number of BS antennas is scalable from 2 to 128.

### IV. IMPLEMENTATION OF KEY SIGNAL PROCESSING BLOCKS

To achieve real-time processing of 128 channels at 30.72 MS/s, all of the key signal processing blocks are implemented on FPGAs using LabVIEW FPGA. Each FPGA co-processor

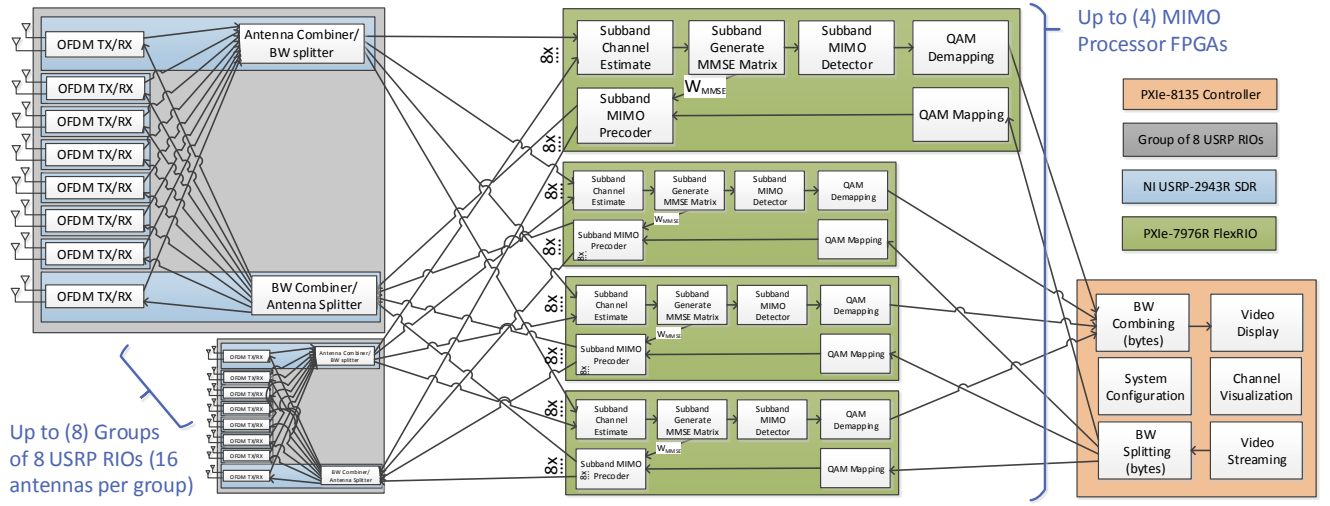


Fig. 3: 128 channel Massive MIMO basestation showing interconnections for data routing, splitting into subbands, and distributed subband MIMO processing.

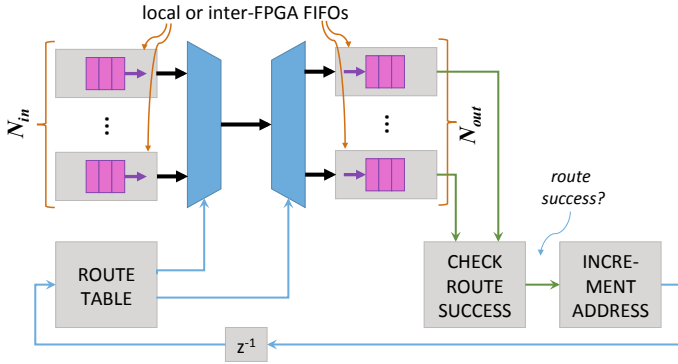


Fig. 4: Router that dynamically routes samples from  $N_{in}$  input FIFOs to  $N_{out}$  output FIFOs according to the pattern stored in route table memory.

TABLE III: Router resources

FPGA bitfile	$N_{routers}$		UL		DL	
	UL	DL	$N_{in}$	$N_{out}$	$N_{in}$	$N_{out}$
Antenna Combiner	1	0	8	4	—	—
Antenna Splitter	0	1	—	—	4	8
MIMO Processor	2	2	8	1	1	8

shown in Fig. 3 performs channel estimation, MIMO detection, and MIMO precoding on a subband (5 MHz, 300 subcarriers) of the 20 MHz bandwidth (1200 used subcarriers). Channel Estimation for each of the 12 UEs is performed on orthogonal subcarriers, employing zero-order hold in time and frequency between two consecutive estimates.

#### A. QRD-based MMSE Decoder/Precoder

MIMO processing is performed using linear MIMO decoding and encoding. The linear decoding matrix  $\mathbf{W}_{MMSE}$  can be

solved for efficiently in hardware using the QR decomposition [9], [10] where

$$\mathbf{B} = \mathbf{G}^H \mathbf{G} + \sigma^2 \mathbf{I} = \begin{bmatrix} \mathbf{G} \\ \sigma \mathbf{I} \end{bmatrix} = \mathbf{Q} \mathbf{R} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \mathbf{R} \quad (7)$$

$$\mathbf{W}_{MMSE} = (\mathbf{B}^H \mathbf{B})^{-1} \mathbf{G}^H = \mathbf{R}^{-1} \mathbf{Q}_1^H = \mathbf{Q}_2 \mathbf{Q}_1^H / \sigma.$$

To achieve the required matrix throughput of one matrix every 12 subcarriers, the  $\mathbf{W}_{MMSE}$  throughput must be  $16.8 \times 10^6$  subcarriers/s/12 =  $1.4 \times 10^6$  Matrices/s. To achieve this throughput, the MIMO processing is split into four FPGAs as aforementioned. The QR decomposition is formulated into a partial parallel implementation employing a systolic array, calculating four columns of the  $128 \times 12$  UL channel estimate matrix  $\mathbf{G}$  in parallel with a new row input each clock cycle. Each column is processed using the discrete steps of the modified Gram-Schmidt algorithm. The total execution time for this formulation is  $3 \times (128 + 12) = 420$  clock cycles. The core is clocked at 200 MHz such that four running in parallel are able to meet the  $1.4 \times 10^6$  Matrices/s throughput.

The end computation of  $\mathbf{Q}_2 \mathbf{Q}_1^H / \sigma$  is similarly formulated, where the matrix-matrix multiply is performed using four parallel length-12 vector dot products with a real multiply to scale by  $1/\sigma$ . The logic in the MIMO processor can be reconfigured so that the same hardware resources that provide  $\mathbf{W}_{MMSE}$  can also provide the ZF and MRC decoders. Taking advantage of channel reciprocity and distributed reciprocity calibration, the downlink precoder is simply the transpose of the decoder matrix. This allows the same core to generate the uplink linear detector and the downlink linear precoder matrices as discussed in the next section.



TABLE IV: FPGA Utilization for the routers

Target	Registers	LUT	RAMs	Instances
RRH	12418 (2.4%)	8578 (3.4%)	55 (6.9%)	1
Co-processor	7686 (1.5%)	4073 (1.55%)	22 (2.75%)	4

### B. Distributed Reciprocity Calibration

Reciprocity calibration is required to utilize the reciprocity property of the propagation channel  $\mathbf{D}$  and for the downlink precoding to work. Ideally, each antenna needs to be calibrated using a complex reciprocity weight for each subcarrier. However, tests have shown [4] that for our RRH transceivers, the weights are fairly constant over a 20 MHz bandwidth. This allows averaging over the whole bandwidth to produce a single weight that can be applied to all subcarriers which in turn scales down the required memory by a factor of 1200.

If the reciprocity calibration weights are directly applied to the UL channel matrix  $\mathbf{G}$  as given in (5), the processing has to be performed centrally as shown in Fig. 5a. This approach requires to multiply  $\mathbf{G}$  with the reciprocity calibration weights to generate  $\mathbf{H}_C$ . Then  $\mathbf{W}_{\text{MMSE,DL}}$  is generated by performing a QRD on  $\mathbf{H}_C$ . Since two *Subband Generate MMSE Matrix* blocks are necessary, area utilization and latency for the MIMO processing is doubled.

To remedy these two disadvantages the reciprocity calibration formulation as given in (6) is used in this design. As shown in Fig. 5b the reciprocity weights are applied on the RRHs for each antenna separately before performing the OFDM processing, e.g.  $\mathbf{C}_{11}^{-1}$  for antenna 1. This approach greatly reduces area utilization and lowers the latency of the critical precoding turn-around signal path as the result from the QRD can be reused. Another interesting feature, is that distributed reciprocity calibration allows to perform the calibration inside the groups of RRHs, such that no traffic between the FPGA co-processors and the RRHs is required which relaxes bandwidth pressure on the bus.

## V. IMPLEMENTATION AND VERIFICATION RESULTS

In this section the FPGA resource utilizations and a latency analysis of the precoder turnaround time are presented.

TABLE IV details the resource utilizations for the routers on the RRHs and FPGA co-processors. The routers on the RRH require more resources as they route either 8 inputs to 4 outputs or 4 inputs to 8 outputs whereas the routers on the FPGA co-processor only perform a 4 to 1 or 1 to 4 routing. Note, that not every RRH requires a router but only two in each RRH group.

In TABLE V the FPGA resource utilizations for the implementation of the QRD, the decoder and the precoder blocks are detailed. As can be seen, the QRD occupies most resources

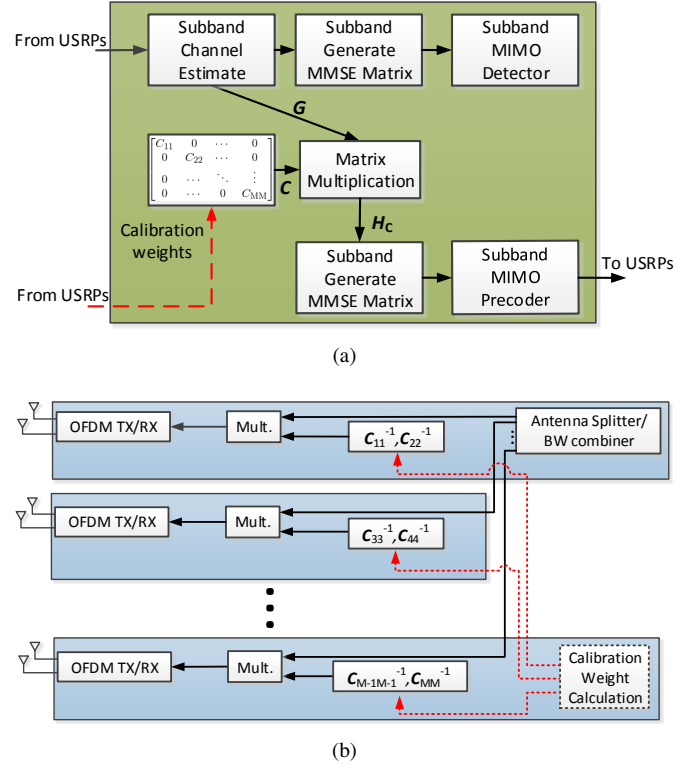


Fig. 5: Applying the reciprocity weights: (a) centrally on the FPGA co-processor and (b) distributed on the RRHs.

TABLE V: FPGA Utilization per function

Function	Registers	LUT	RAMs	DSP48
QRD	46470 (9.1%)	49315 (20.3%)	171 (21.5%)	596 (38.7%)
Decoder	27142 (5.3%)	8844 (3.5%)	13 (1.7%)	313 (20.3%)
Precoder	14379 (2.8%)	10106 (4%)	4 (0.5%)	193 (12.5%)
Total	87991 (17.2%)	68265 (27.8%)	188 (23.7%)	1102 (71.5%)

followed by the decoding and precoding. The DSP48 usage is relatively high with almost 72% of the overall available 1540 DSP48 blocks occupied.

To further analyze the DSP48 usage TABLE VI details a breakdown to the subfunction blocks in the MIMO processor FPGAs. The total DSP usage for the data path is 1109 DSPs. This number differs from the previously presented once, as synthesis tool might infer some DSP48 slices for control signaling. We also included the channel estimation here, which uses a least-square implementation. Main contributor to the DSP48 usage is the actual QRD using the modified Gram-Schmidt algorithm. It is also visible, that decoding occupies almost twice as many DSP48 blocks as precoding due to the higher dimensionality of the vector in the matrix-vector multiply.

TABLE VI: DSP resources per MIMO processor FPGA

Function	Subfunction	DSP48s used	$N$ instances	Total
LS channel estimate	$\hat{h} = p^* y$	4	4	16
$QR(\hat{G})$	$\frac{v}{\ v\ }$	13	12	156
	$u - (u \cdot v)v$	10	20	200
compute $W_{MMSE}$	$\frac{1}{\sigma} Q_2 Q_1^H$	200	1	200
MIMO decode	$W_{MMSE} y$	312	1	312
	unbias $W_{MMSE}$	33	1	33
MIMO precode	$W_{MMSE}^T s$	192	1	192
total DSP48s implemented				1109

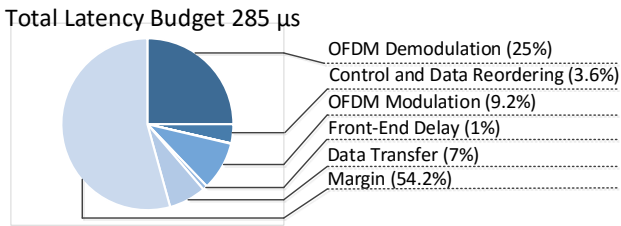


Fig. 6: Pie Chart of the different parts contributing to the latency for the precoding turnaround time.

Fig. 6 shows a breakdown for the latency in the precoding turnaround path. The Overall latency is about 132  $\mu s$  whereas the available budget is 285  $\mu s$ . The channel estimation and precoding parts are too low to be visible in the pie chart. Analog front-end as well as control and data reordering have the lowest contribution. OFDM demodulation occupies a whole OFDM symbol as its processing speed is limited by the sampling rate as opposed to OFDM modulation which is performed at highest clock rate. Due to its nature, a deterministic timing analysis for data transfer over the PCI bus is not feasible such that worst-case timing analysis with the maximum number of 4 hops over the bus was done. This path is from an RRH that only has an OFDM chain over the RRH performing the antenna combining and bandwidth splitting to the MIMO processor and back on the transmit path. Interestingly, the precoding turnaround time putting a hard constraint on overall signal processing and data shuffling latency has more than two OFDM symbols (54.2%) margin, *i.e.* our implementation can support even higher UE mobility than previously stated.

The presented signal processing was fully verified on the LuMaMi testbed running with 100 antennas and 12 UEs in real-time using the previously discussed frame structure. Verification was performed by transmitting pseudo-random sequences and comparing them, plotting a subset of the constellations on the host computer and even transmitting video streams on the uplink and downlink.

## VI. CONCLUSION

In this paper we presented a low-latency implementation of key signal processing blocks in an OFDM-based MaMi base station operating in TDD mode. By distributing processing over several FPGAs, the data shuffling and processing requirements are relaxed. Using a high-throughput low-latency router for inter- and intra FPGA communication and a modified QRD-based Gram-Schmidt algorithm we are able to implement the whole signal processing and communication chain distributed over 4 Kintex 7 FPGAs. Utilizing the property that reciprocity calibration may be performed in a distributed fashion over the RRHs we efficiently lower the processing latency. Latency analysis shows that our implementation is suitable for high mobility scenarios with a precoding turnaround time of less than 0.15 ms.

## ACKNOWLEDGMENT

This work was funded by the Swedish foundation for strategic research SSF, VR, the strategic research area ELLIIT, and the E.U. Seventh Framework Programme (FP7/2007-2013) under grant agreement n 619086 (MAMMOET).

## REFERENCES

- [1] T. L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, November 2010.
- [2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, Jan 2013.
- [3] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I. Wong, V. Öwall, O. Edfors, and F. Tufvesson, "A flexible 100-antenna testbed for Massive MIMO," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 287–293.
- [4] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, L. Liu, and F. Tufvesson, "Reciprocity Calibration for Massive MIMO: Proposal, Modeling and Validation," *ArXiv e-prints*, 2016.
- [5] W. Zhang, H. Ren, C. Pan, M. Chen, R. C. de Lamare, B. Du, and J. Dai, "Large-Scale Antenna Systems With UL/DL Hardware Mismatch: Achievable Rates Analysis and Calibration," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1216–1229, April 2015.
- [6] "NI USRP-2943R Data Sheet," <http://www.ni.com/datasheet/pdf/en/ds-538>, 2014, Online; accessed 29 June 2016.
- [7] "NI FlexRIO 7976R Data Sheet," <http://www.ni.com/pdf/manuals/374546a.pdf>, 2014, Online; accessed 29 June 2016.
- [8] "NI PXIe 8135 Manual," <http://www.ni.com/pdf/manuals/373716b.pdf>, 2013, Online; accessed 29 June 2016.
- [9] Y. Rao, "Implementing modified QR decomposition in hardware," Dec. 1 2015, US Patent 9,201,849.
- [10] Y. Rao, "Software tool for implementing modified QR decomposition in hardware," Nov. 3 2015, US Patent 9,176,931.