04GRANT AGREEMENT NO.: 732174

Call: H2020-ICT-2016-2017
Topic: ICT-13-2016
Type of action: RIA

Orchestration and Reconfiguration Control Architecture

D7.4: Summary of Results of Second Open Call for Extensions

Revision: v.1.0

Work package	WP 7
Task	Task 7.2
Due date	31/10/2019
Submission date	04/11/2019
Deliverable lead	IMEC
Version	1.0
Authors	Wei Liu (IMEC), Ingrid Moerman (IMEC), Xianjun Jiao (IMEC), Seyed Ali Hassani (KUL), Roberto Bomfin (TUD), Ahmad Nimr (TUD), Walter Nitzold (NI)
Reviewers	Joao Santos (TCD)

Abstract	D7.4 collects the summaries and conclusions of the second Open Call for Extensions. It introduces the organization of the Call, the

	winners, and an overview of each extension, including their implementation, usage, and main challenges. This deliverable will be used for promoting the ORCA facility in WP8.
Keywords	Open Call, testbed, SDR, extensions.

Disclaimer

The information, documentation and figures available in this deliverable, is written by the ORCA (Orchestration and Reconfiguration Control Architecture) – project consortium under EC grant agreement 732174 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	R	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ORCA project and Commission Services	

EXECUTIVE SUMMARY

This deliverable reports the work done in the 2nd Open Call for Extensions of ORCA project. In this Open Call, four distinctive topics are defined, each of which possesses clear functional and technical requirements, as well as a validation scenario on the ORCA facility. This call is organized in a similar way as previous open calls: First, the potential candidates submit their proposals for receiving feasibility and relevance feedback from patrons, and then a final proposal is prepared based on the feedback. Next, external evaluators are contacted and polled for conflict of interests, and when all conditions are met, the final proposals are forwarded to independent external evaluators for evaluation. Finally, the winning proposals are selected based on the scores of the evaluation or consensus meeting if differences exist among the external evaluators. After introducing the Open Call's organization, the main targets and results of each of the extensions are introduced. The first extension was developed to realize partial FPGA reconfiguration on two distinctive SDR platforms, aiming to achieve run-time configuration/composition of processing chains on SDRs. The second extension consists in designing an Intellectual Property (IP) core of Polar decoder running on the FPGA of SDR devices. The third extension achieves the E-UTRA Dual Connectivity (DC) capability through two implementations: an ns-3 LTE based implementation of DC, and an implementation built on the open-source LTE/EPC software provided by Open Air Interface project. The last extension is developed to provide ORCA with IEEE 80211ad-compliant baseband processing blocks at a scaled-down frequency, and their integration with the GNU Radio & RFNoC framework, for being used in on-site and remote experiments in the laboratories from ORCA consortium.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AMP	Asymmetric Multiprocessing
BMA	Bare Metal Application
CIR	Channel Input Response
DC	Dual Connectivity
DDS	Direct Digital Synthesizer
EXT	Extensions
IP	Intellectual Property
OAI	Open Air Interface
LBT	Listen Before Talk
lwIP	Lightweight IP
OC	Open Call
OC2 EXP	Second Open Call for Experiments
PDCP	Packet Data Convergence Protocol
PL	Programmable Logic
PR	Partial Reconfiguration
PS	Processing System
REWIRE	The Run-timE firmWare reconfIguration controller
RLC	Radio Link Control
RTOS	Real-Time Operating System
SDR	Software Defined Radio
SC	Successive Cancellation
SCF	Successive Cancellation Flip
SCF	Successive Cancellation List
TFTP	Trivial File Transfer Protocol

1 INTRODUCTION

This deliverable provides a summary of the results obtained during the second Open Call for Extensions in ORCA project.

The remainder of this document is structured as follows:

- Chapter 2 includes an overview of the OC2 for Extension, with the call information and the winners of the extensions.
- Chapter 3 provides a high-level introduction of each Extension, in terms of the extension's objective, main challenges, results, conclusions, and feedback towards the ORCA consortium.
- Finally, in Chapter 4, we conclude the deliverable.

2 SECOND OPEN CALL FOR EXTENSIONS

2.1 Call Information

The OC2 EXT aims to attract external partners with dedicated expertise to develop functionalities for the ORCA facility.

The following Table 1 demonstrates basic Call information.

Project full name	ORCA - Orchestration and Reconfiguration Control Architecture
Project grant agreement No.	732174
Call identifier	ORCA-OC2-EXT
Call title	Second ORCA Open Call for Extensions
Submission deadline	Monday the 9th July 2018, at 17:00 Brussels local time
Feasibility & relevance check deadline	Monday the 2nd July 2018, at 24:00 Brussels local time

Table 1 : ORCA OC2 EXT Basic Call Information.

The extension topics in OC2 EXT include:

- EXT1 – Partial reprogramming of FPGA on SDR at runtime
- EXT2 – Polar Codes for FPGA
- EXT3 – Interfaces for 5G and LTE interworking
- EXT4 – Dynamic runtime composition of mmWave transceiver chains consisting of processing functions that are split between FPGA and CPU

The patrons of the four extensions are IMEC, TUD, NI, KUL respectively

In terms of financial information, the total budget for OC2 EXT: €300,000. Maximum budget per Extension: €75,000. It is defined that each project will have guaranteed support of €18,000, with an extra budget of typically €4,500 per Extension will be allocated to the ORCA consortium partner acting as Patron for guaranteed support.

2.2 Winners

An established, independent and impartial evaluation process, of confidential nature has been applied to filter and select the winning proposals. A brief summary of the process is as follows:

- Feasibility and relevance check
    - Candidates provide a first submission.
    - Patrons evaluate the feasibility and relevance of the proposals and provide feedback.
- Final submission and evaluation
    - Based on the feedback, the proposals passed the feasibility check will submit the final applications.

- Coordinator polls the external evaluators regarding conflict of interests of the proposals they are assigned to evaluate. In case there is conflict, the assignment will be adjusted.
- The external evaluators will receive and review the final applications based on a set of criteria and give scores.
- Consensus meetings between evaluators were held when necessary to agree on the winners.

The following projects have won the ORCA OC2 EXT:

- **EXT1: ReproRun** (Reprogramming FPGA devices at run time using partial reconfiguration in SDR platforms). Submitted by: CTTC – Centre Tecnològic de Telecomunicacions de Catalunya (Spain)
- **EXT2**: **EfficientPo** (Efficient Polar Encoding and Decoding for FPGAs). Submitted by: EPFL – École Polytechnique Fédérale de Lausanne (Switzerland)
- **EXT3: DALI** (Dual Connectivity Solution for ORCA). Submitted by: Universitat Politecnica de Catalunya (Spain)
- **EXT4: MISO** (Millimeter-wave SDR-based Open Experimentation Platform). Submitted by: IMDEA NETWORKS INSTITUTE (Spain)

3 SUMMARY OF RESULTS IN EACH PROJECT

3.1 EXT1

3.1.1 Goal of this extension

ReproRun provides a run-time partial reconfiguration (PR) framework for the FPGA devices of two SDR platforms. The on-field reconfiguration concerns both the functions running at the programmable logic (PL), as well as the processing system (PS) of FPGA devices. The PL partial bitstreams and PS firmware are fetched from a remote location.

3.1.2 Main challenges

The main challenge of ReproRun was that the run-time partial reconfiguration framework had to be developed for two different SDR platforms featuring different FPGA devices. The combined PL and PS reconfiguration added novelty to the project but increased the development, testing and validation complexity.

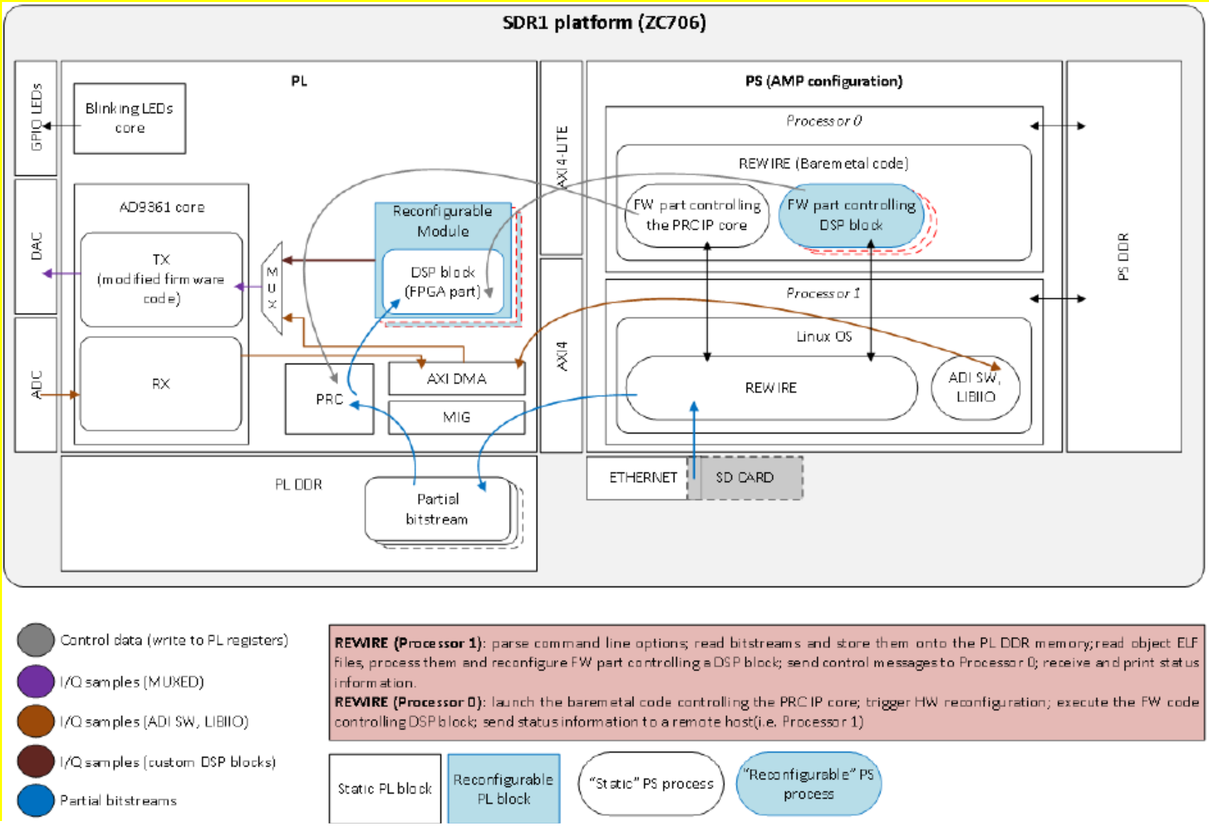
3.1.3 Description of concept of extension

ReproRun is able to seamlessly reconfigure at run-time part of the PL area of an FPGA device with a partial bitstream and also the firmware running at PS embedded in the same FPGA device, i.e., hardwired or soft microprocessor. The firmware among other functions includes a software bare metal application (BMA) that programs parameters of the PL-based function/application. The concept of ReproRun was applied in the two SDR platforms selected by the Patron. First SDR platform is the ZYNQ SDR, comprised of Xilinx ZC706 board and Analog Devices FMCOMMS2 board, hereafter referred to as SDR1. The second SDR platform is the Ettus X310 USRP, hereafter referred to as SDR2. Despite some key differences in the two SDR platforms, the core functionality of the developed framework remains the same. The Run-timE firmWare reconfiguration contRollEr (REWIRE) was developed to handle the PS-based run-time firmware reconfiguration.

In SDR1, part of REWIRE runs at the Processor 1, i.e., CPU1 of the ARM processor running embedded Linux, while the other part of REWIRE runs at Processor 0, i.e., CPU0 of the ARM processor running bare metal applications. An asymmetric multiprocessing (AMP) design framework was adopted towards this end. In SDR2, part of the REWIRE functionality runs in the host processor (desktop Linux, Processor 1) and another part runs at a MicroBlaze soft microprocessor (bare metal application, Processor 0). Processor 0 in SDR2 includes a library that fetches the partial bitstreams and BMAs (object files) from the Trivial File Transfer Protocol (TFTP) server running at a remote host (TFTP runs on top of the lightweight IP (lwIP) framework in Processor 0). REWIRE is able to parse command line options; read bitstreams and store them in the PL DDR memory; read object ELF files, process them and reconfigure firmware part controlling a DSP block; send control messages to Processor 0; receive and print status information. REWIRE also signals the partial reconfiguration controller (Xilinx IP core) to apply the partial reconfiguration. The architecture mentioned above is illustrated in Figures1 and 2.

The use of the SDR2 implied an integration of the ReproRun framework with the RFNoC code. For this reason, a separate Ethernet connectivity was dedicated for the ReproRun and RFNoC frameworks, while the access to SDRAM was shared. In addition, the ZPU soft processor of the RFNoC firmware was not used for the development goals of ReproRun, in order to decouple the software processing (avoiding as such the use of an Real-time Operating System (RTOS)). In both SDR platforms, the FPGA devices are configured with a static bitstream, which has a simple application in free run mode (blinking hard or soft LEDs in SDR 1 and SDR 2 respectively). In SDR1 the partial bitstreams are a Direct Digital Synthesizer (DDS) function and a LTE waveform playback function. In SDR2 the partial bitstream is the CORDIC function of the *siggen* RFNoC example. The concept and experimental setup of the two

SDR platforms can be seen in the figures that follow. Unfortunately, after thorough investigation, the partial reconfiguration via ICAP interface used by the Xilinx PRC on SDR2 is not feasible, due to a hardware limitation, i.e. the presence of a resistor that is soldered on the board and prevents the FPGA chip to be configured via this interface. However, the partial configuration can be achieved via JTAG interface. We are in contact with Ettus research to confirm this conclusion and possible work around.



REWIRE (Processor 1): parse command line options; read bitstreams and store them onto the PL DDR memory; read object ELF files, process them and reconfigure FW part controlling a DSP block; send control messages to Processor 0; receive and print status information.

REWIRE (Processor 0): launch the baremetal code controlling the PRC IP core; trigger HW reconfiguration; execute the FW code controlling DSP block; send status information to a remote host(i.e. Processor 1)

Figure 1: Main architecture of EXT1 realized on ZYNQ SDR.

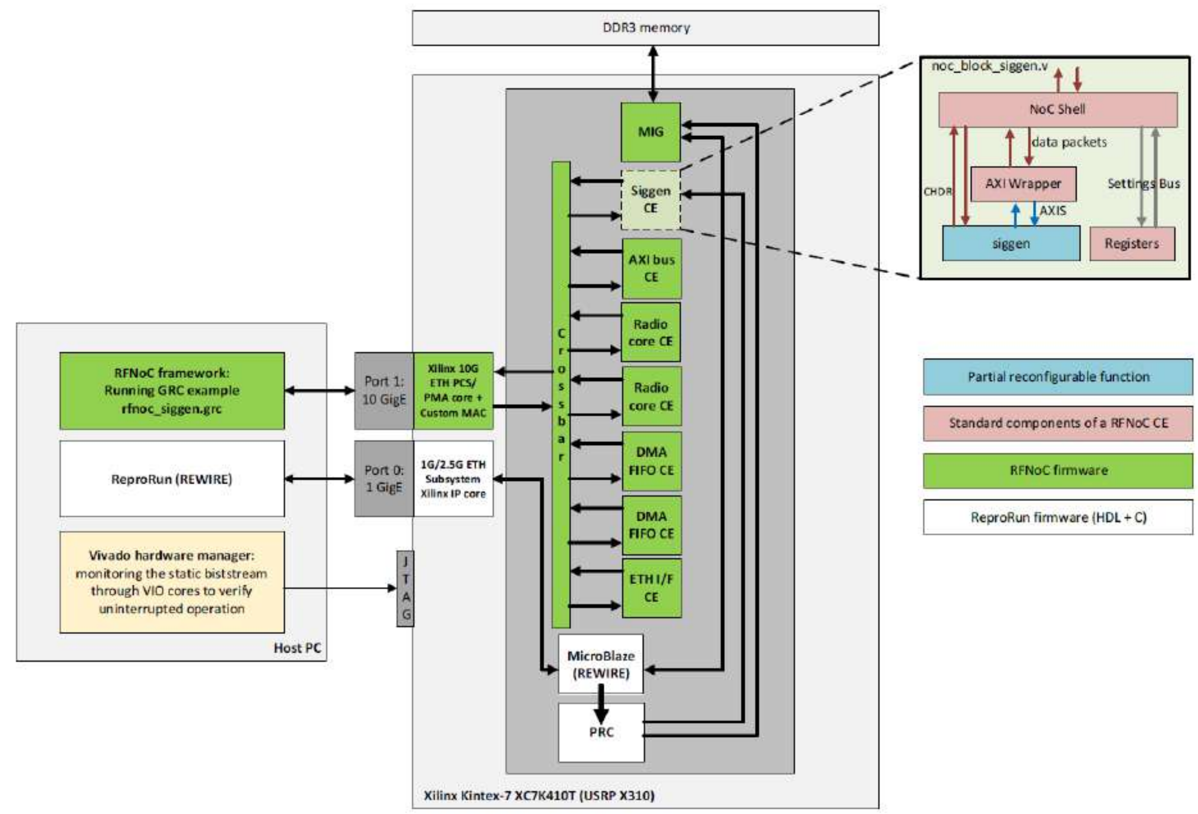

DDR3 memory
noc_block_siggen.v
NoC Shell
data packets
CHDR
AXI Wrapper
Settings Bus
AXIS
siggen
Registers
MIG
Siggen CE
AXI bus CE
Radio core CE
Radio core CE
DMA FIFO CE
DMA FIFO CE
ETH I/F CE
Crossbar
RFNoC framework: Running GRC example rfnoc_siggen.grc
ReproRun (REWIRE)
Vivado hardware manager: monitoring the static biststream through VIO cores to verify uninterrupted operation
Host PC
Port 1: 10 GigE
Xilinx 10G ETH PCS/ PMA core + Custom MAC
Port 0: 1 GigE
1G/2.5G ETH Subsystem Xilinx IP core
JTAG
MicroBlaze (REWIRE)
PRC
Xilinx Kintex-7 XC7K410T (USRP X310)
Partial reconfigurable function
Standard components of a RFNoC CE
RFNoC firmware
ReproRun firmware (HDL + C)

Figure 2: Main architecture of EXT1 realized on USRP X310 SDR.

3.1.4 Main results

The waveform output is changed by loading different partial bitstream, as shown in Figures 3 and 4.

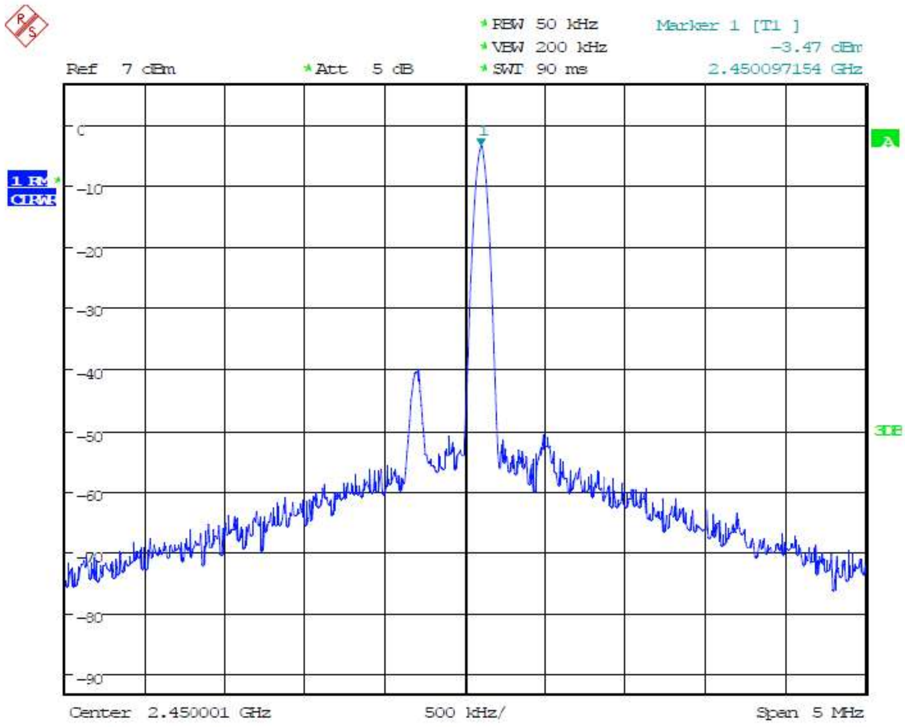


Figure 3: The output of the AD-FMCOMMS-2/3 when REWIRE configures the DSP_DDS.bin partial bitstream.

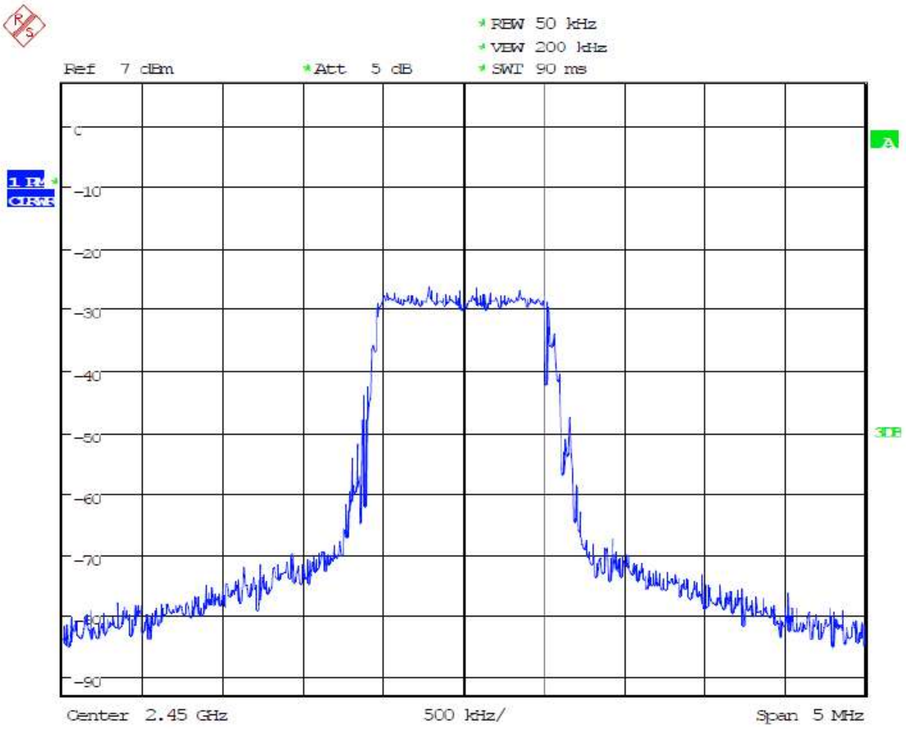


Figure 4: The output of the AD-FMCOMMS-2/3 when the REWIRE configures the DSP_LTE.bin partial bitstream.

3.1.5 Conclusions

ReproRun provides a seamless run-time reconfiguration framework of PL-based partial bitstreams together with their corresponding PS-based firmware, for two popular SDR platforms featuring two different FPGA devices. This combined on-the-field reconfiguration of hardware-accelerated and firmware functions is a novel top-up feature for 5G enabling technologies and other end-applications.

3.1.6 Feedback

This Extension perfectly aligns with the scientific roadmap of CTTC's team and opens up new opportunities for research synergies, collaborations, dissemination and hopefully exploitation. The communication with the Patron was fluent and the support satisfactory. Targeting only one SDR platform would have helped us to develop more advanced partial reconfiguration features.

3.2 EXT2

3.2.1 Goal of this extension

The first goal of this project was to develop an FPGA Polar decoder to add Polar coding capabilities to ORCA. This is interesting due to their superior error-correction performance and due to the timely relevance as Polar codes have been established as part of 5G-NR. Our second goal was to provide a decoder that can switch between different algorithms, which provides a means to compare them in a full system.

3.2.2 Main challenges

The first challenge lies in providing a single decoder core that can perform different decoding algorithms and can be tuned to different complexity-performance trade-offs. Combining specifically Successive Cancellation (SC), Successive Cancellation List (SCL), and Successive Cancellation Flip (SCF) decoding allows for effective reuse of resource-hungry core components. The sequential SCF decoding is specifically designed to allow run-time tuning between performance and complexity.

3.2.3 Description of concept of extension

For this project, we use a decoder architecture that builds on a variable number of SC decoder core instances, extended with a list processing unit. The number of instances scales the error-correction performance, but also the hardware complexity of the decoder. By adding a simple sorter and a modified controller, the SCF decoder allows to also explore trade-offs between throughput and error correction performance at run-time. The corresponding SCF decoder re-uses hardware resources for the SC/SCL decoder, which allows to add the SCF functionality with almost no hardware overhead. Figure 5 illustrates the components of the decoder.

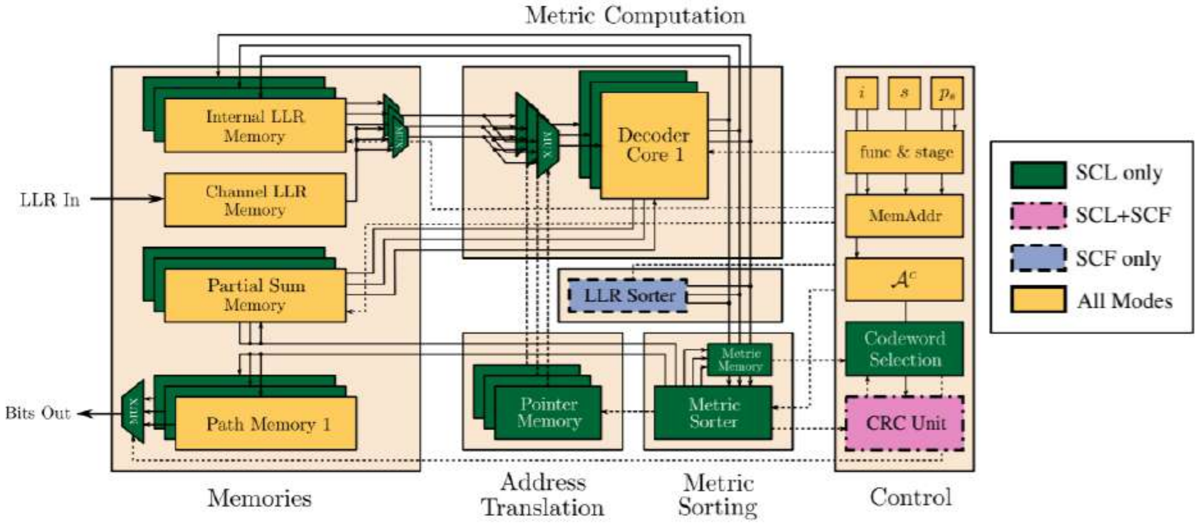


Figure 5: Multi-mode polar decoder architecture.

3.2.4 Main results

The decoder developed in this project supports several modes that allow trade-offs between performance and complexity. The provided figure illustrates these trade-offs. Since the decoder is primarily intended for the control channel (as specified in 5G-NR), it is designed to keep FPGA resource utilization below 20% (for a block size of N=1024). At the testbed specified clock frequency of 100MHz, it achieves 20Mbps throughput in SCL mode. The encoder throughput reaches up to 100 Mbps.

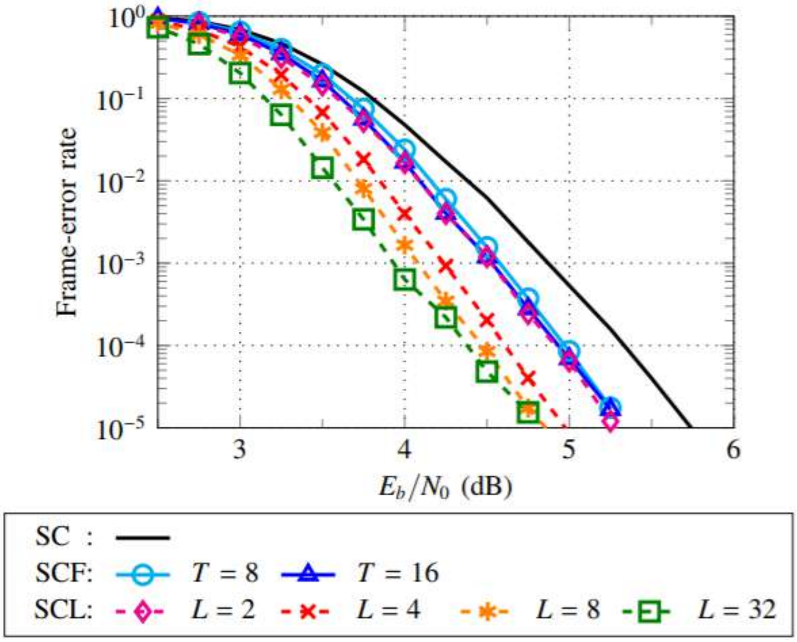


Figure 6: Multi-mode polar decoder error-correction performance. The FPGA design with 20% resource utilization allows for a list size of L=2. Larger list sizes require more resources.

Full curves are related to channel occupancy when the Listen Before Talk (LBT) system transmits whatever the sensed power is. The doted curve shows the impact on Wi-Fi characterized by its throughput. “High threshold” is set to get a secondary emission at each opportunity, i.e. at the end of the sensing period, whatever the primary system power is. “Low threshold” is set to get a secondary emission based on the sensing status.

3.2.5 Conclusions

The project provides the ORCA testbed with a multi-mode Polar encoder and decoder. By supporting different modes, it enables experiments that compare the impact of different decoding efforts on the system level, including the impact of the variable decoding latency of the SCF algorithm. The decoder is mainly targeted for decoding of control channel packets, as it is optimized for resource consumption rather than for throughput.

3.2.6 Feedback

The collaboration with the ORCA team at TUD was very efficient. Clear specifications were provided by the patron (TUD) and the integration of the delivered VHDL IP module into the test and verification environment provided by TUD was straightforward. The test and debug process itself was carried out in collaboration with TUD and did not require a local LabView setup. The latter might have helped in the final stages of the debug process but was also not required.

3.3 EXT3

3.3.1 Goal of this extension

DALI extends the ORCA facilities with the E-UTRA Dual Connectivity (DC) capability through two implementations: i) an ns-3 LTE based implementation of DC, and ii) DC implementation built on the open-source LTE/EPC software provided by Open Air Interface (OAI) project.

3.3.2 Main challenges

Although the complexity of OAI has been an important challenge, we could successfully implement our extension within OAI through rigorous efforts. The other challenge has been the CPU race conditions created between the ns-3 functionalities used and the NI API, which is also overcome through a collaborative effort with our patron NI.

3.3.3 Description of concept of extension

DALI Dual Connectivity setup (Figures 7 and 8) consists of two eNB nodes and two UE nodes (a Master and a Secondary in each case) and an EPC node. In line with 3GPP specifications for E-UTRA DC functionality, a link is established to communicate Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers of different eNB nodes through X2 interface and a newly implemented DALI UE DC interface for communication between UEs.

As shown in Figure 7 and Figure 8, in DALI, the eNB and UE pairs communicate through an SDR (alternatively, via simulated channels), and the LTE stack is implemented in ns-3 and OAI, respectively. In ns-3 DALI implementation, the ns-3 LTE stack runs on Linux RT, which interfaces the SDRs through the generic NI L1-L2 API. In OAI-based DALI implementation, the OAI LTE stack runs on Ubuntu distribution of Linux with a low-latency kernel, with the SDR interfacing is achieved through the generic Linux drivers/toolsets of the common SDR platforms, such as UHD for Ettus USRP SDR devices, Lime Suite for LimeSDR. In OAI implementation, the baseband processing is done on the Generic Purpose Processor (GPP) Linux machine.

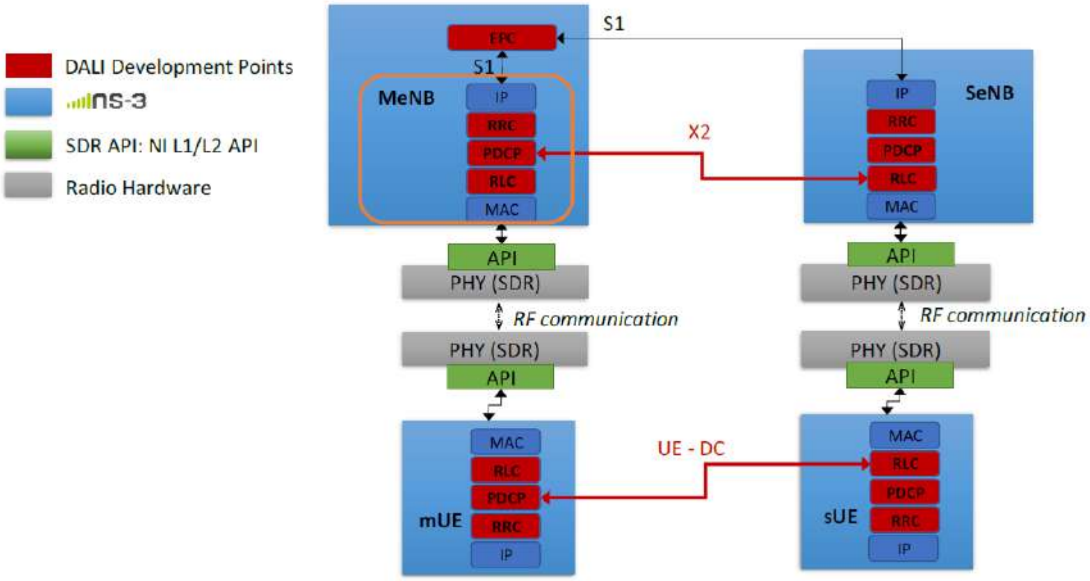


Figure 7: Architecture of DALI solution with ns-3 based real-time LTE devices.

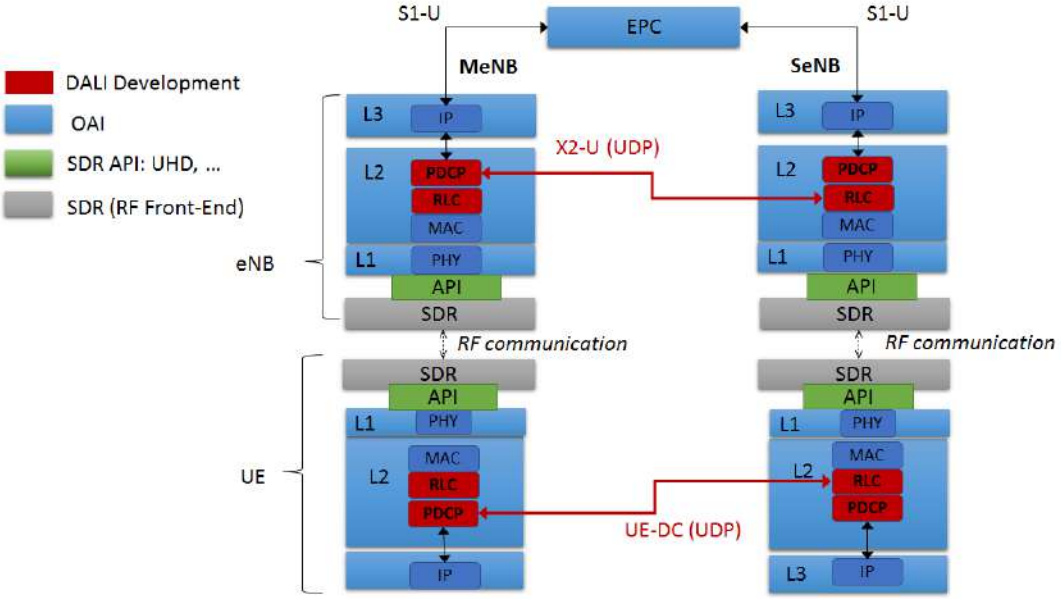


Figure 8: Architecture of DALI solution with OAI-based real-time LTE devices.

3.3.4 Main results

The validation of DALI has been done for both UDP-based and TCP-based applications. For ns-3 based implementation, the UDP-based application evaluation results are shown in Figure 9. It is clearly seen that split bearer option enabled by Dual Connectivity almost doubles the throughput for both downlink and uplink compared to that of a single bearer option at Master eNB or Secondary eNB.

However, TCP-based applications' throughput suffers from the out-of-order packets in case of dual connectivity as shown in Figure 10. For this, DALI implements the reordering function implemented in DALI improves the throughput significantly as shown in the figure.

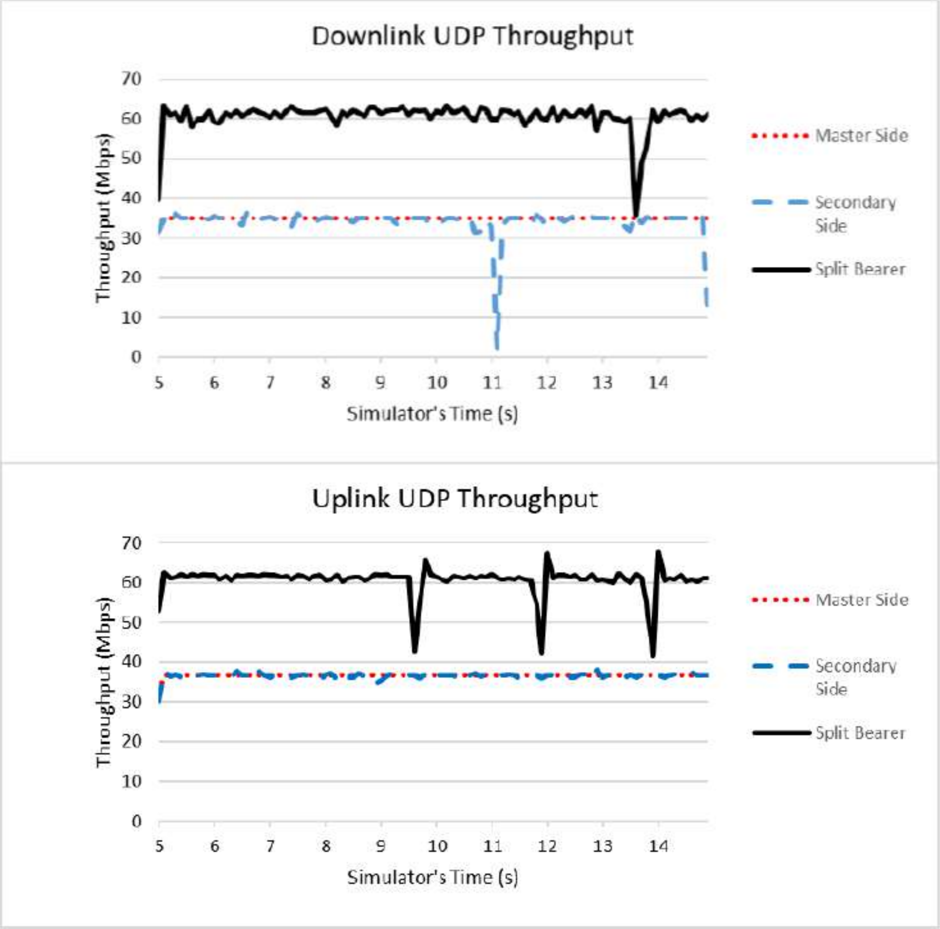


Figure 9: UDP Throughput: Downlink (up) and Uplink (bottom).

3.3.5 Conclusions

The DALI solution has been validated through a five-stage testing process, starting with in-house tests with simulated channels and finally experimenting with SDRs at ORCA testbeds. The DALI project provides an easily reproducible solution, since it uses open-source software and standard interfaces, and paves way to rapid development and realistic testing of further DC extensions through ORCA.

3.3.6 Feedback

ORCA was significantly useful due to the large variety of hardware and software resources it offers to conduct experiments. Access to these resources allows us to focus in the development of the solutions rather than setting up a testbed, which would bring technical and economic challenges. Additionally, the robustness of the testbed made possible to test the solutions under different conditions that were not initially planned which helped us to get better results.

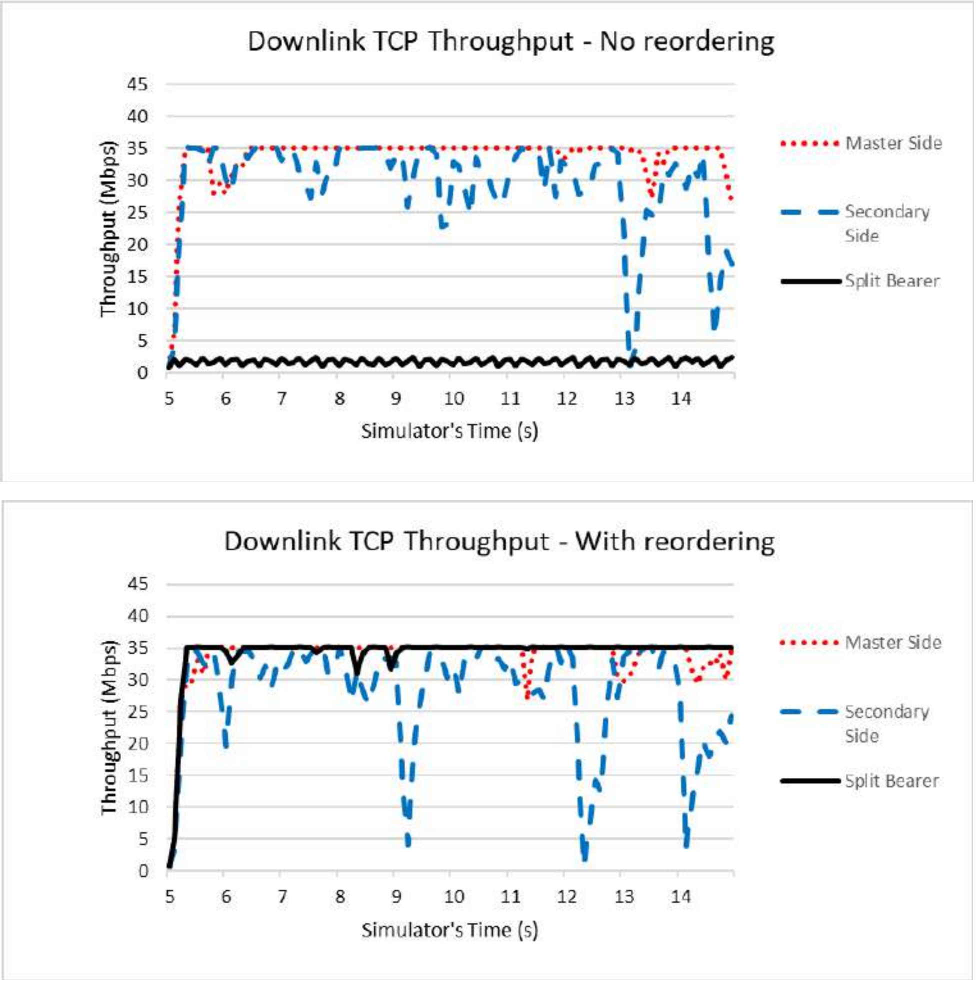


Figure 10: TCP Throughput for Downlink without (up) and with (bottom) with PDCP reordering function.

3.4 EXT4

3.4.1 Goal of this extension

In this project we proposed a mixed hardware-software design for mm-wave experimentation on SDRs. Specifically, we proposed to design and implement the hardware blocks to decode the preamble of single carrier (SC) IEEE 802.11ad compliant frames at an scaled-down frequency and their integration in the GNU Radio + RFNoC framework to be used in on-site and remote experiments in the laboratories from ORCA consortium.

3.4.2 Main challenges

Reducing the sampling frequency of a 60GHz communication system is a big challenge since the imperfections does not scale as well. Besides, the integration of multiple high-speed signal processing blocks while using the maximum bandwidth of USRP devices require efficient hardware implementations to fit into the FPGA device, met timing constraints and leave space to further upgrading of the design.

3.4.3 Description of concept of extension

The designed system can be used in on-site experiments (using USRP devices and 60 GHz RF front-ends) and also in any of the remote laboratories provided by the ORCA consortium with X310 devices available and RFNoC framework.

The transmitter side is implemented including IEEE 802.11ad compliant single carrier frames from files which are outputted from the X310 USRP devices using a basic transmitter daughter board (to output baseband complex IQ samples). The baseband signal is fed to a 60GHz RF front-end from the remote laboratory or one available from IMDEA Networks facilities (VubIQ 60GHz kit, for example).

In the receiver side, after down conversion of the signal to baseband, it is fed to the USRP device using a basic receiver daughterboard. In the FPGA side, signal passed through the RFNoC blocks implementing the preamble processing blocks, as can be seen in Figure 11. After CIR (Channel Input Response) computation, 128 complex samples (corresponding to the computed CIR) per detected frame are sent to the host PC. (local or remote, depending on the experiment) to save the data of the experiment, plot it or continue processing the frame using software blocks.

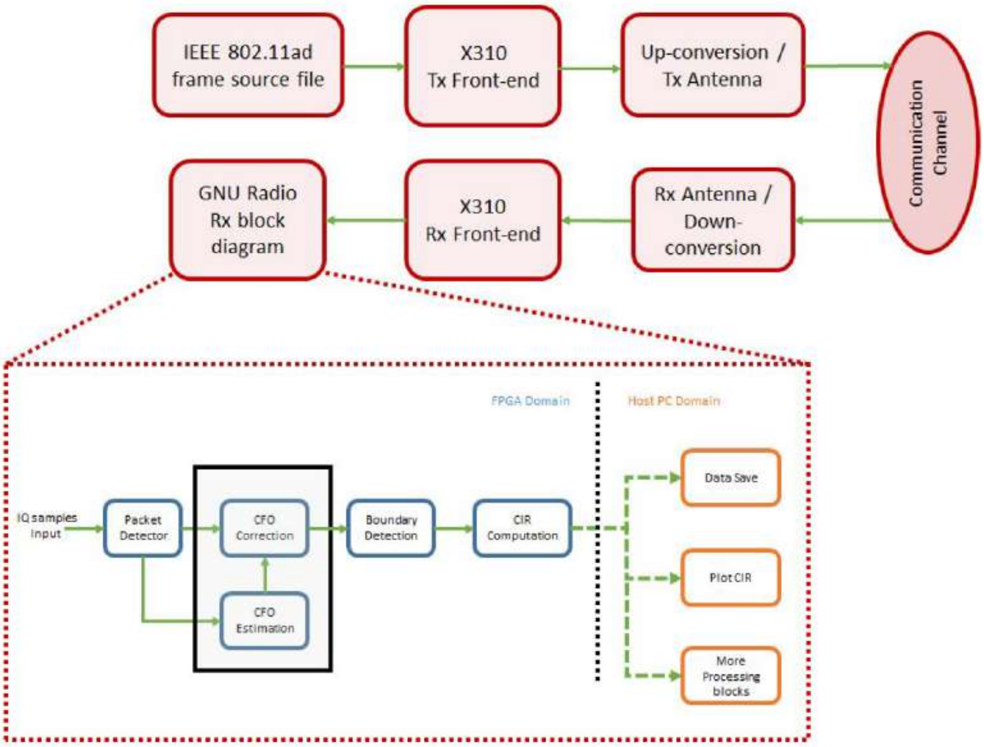


Figure 11: Block diagram of the developed mixed hardware-software system.

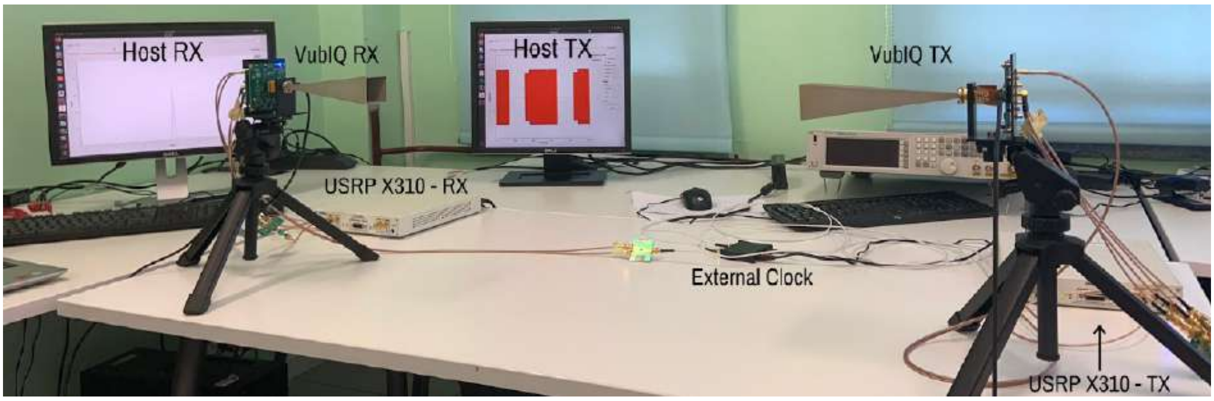


Figure 12: Example experiment setup for on-site experiments.

3.4.4 Main results

We designed and implemented the main preamble processing blocks to decode frames with the same structure of IEEE 802.11ad compliant frames but with a scaled-down sampling frequency. Implemented blocks were wrapped with an AXI interface and integrated in the RFNoC framework to be used with X310 USRP devices in the GNU Radio + RFNoC framework. Designed system is area/timing efficient allowing the use of the maximum available bandwidth of the USRP device while leaving FPGA space to include more processing blocks in future extensions of the project. The design have been validated in on-site experiments using 60GHz RF front-ends available at IMDEA Network facilities as well as in remote laboratories from the ORCA consortium.

The main results displayed below are (i) a successful detection of incoming packet in Figure 13, and (ii) the Channel Impulse Response of 5 consecutive frames

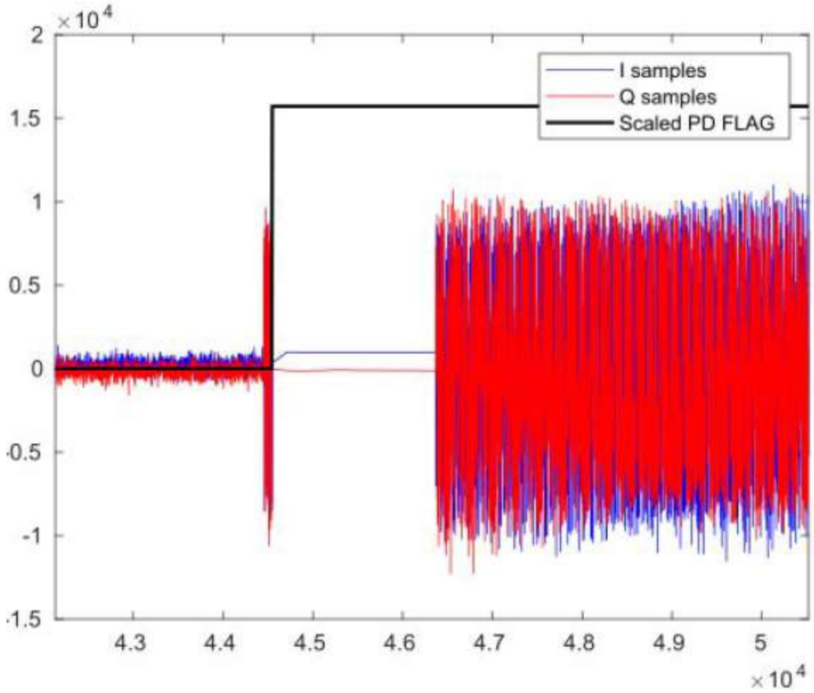


Figure 13: Packet Detector Output showing the flag and output switching when a valid frame is detected.

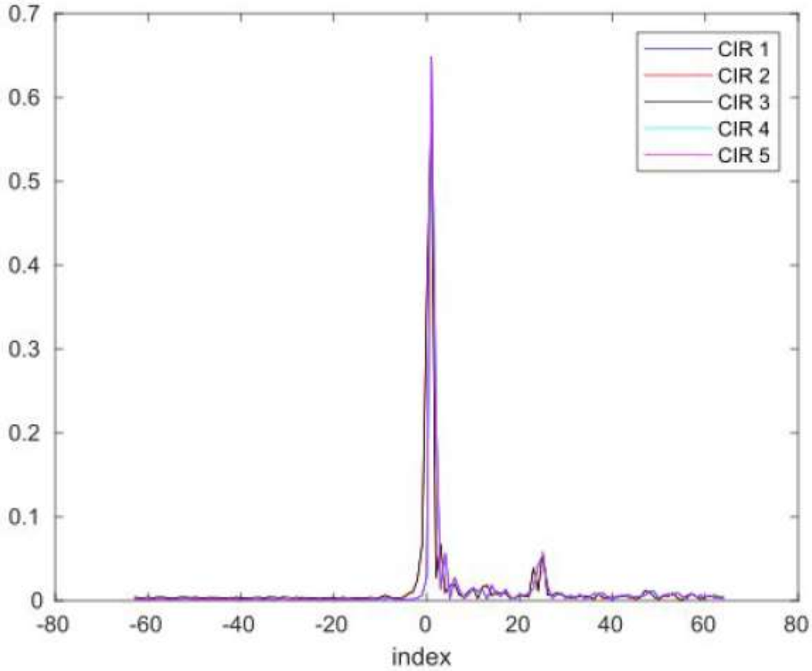


Figure 14: Output of the Channel Impulse Response (CIR) block, showing the CIR of five consecutive frames.

3.4.5 Conclusions

MISO project provide the basis for mm-wave experimentation using USRP devices which are cost efficient to build massive remote laboratories to expand the research and academic activities involving mm-wave communication systems to small research groups and universities around the world without the need of having expensive equipment which is not always possible.

3.4.6 Feedback

Overall, we were very satisfied with this project. The work and tasks were extremely well aligned with our interests and expertise, the overhead for running the project was very manageable so that we could fully focus on research and implementation tasks, and the resources provided by the project allowed to implement a low cost system that both will be used internally for our research as well as serves as a very useful testbed platform for others.

4 CONCLUSIONS

This deliverable reports on the work done during the 2nd Open Call for Extensions of the ORCA project. First, the organization of the Call were introduced, and then the results of each extension topics were given. In this Open Call, four different topics were defined, as listed below.

- EXT1 – Partial reprogramming of FPGA on SDR at runtime
- EXT2 – Polar Codes for FPGA
- EXT3 – Interfaces for 5G and LTE interworking
- EXT4 – Dynamic runtime composition of mmWave transceiver chains consisting of processing functions that are split between FPGA and CPU

The winning proposals were selected by scores given by independent external evaluators. The extensions were then implemented by qualified 3rd parties and validated on at least one ORCA facility. All the third parties have met the technical and functional requirements specified in this call.