Orchestration and Reconfiguration Control Architecture

DALI
Dual Connectivity Solution for ORCA

Open Call partner
Universitat Politecnica de Catalunya

Patron
National Instruments

OBJECTIVES

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.

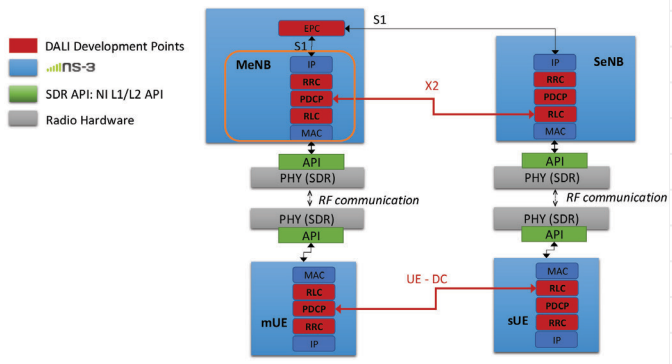
CHALLENGES

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.

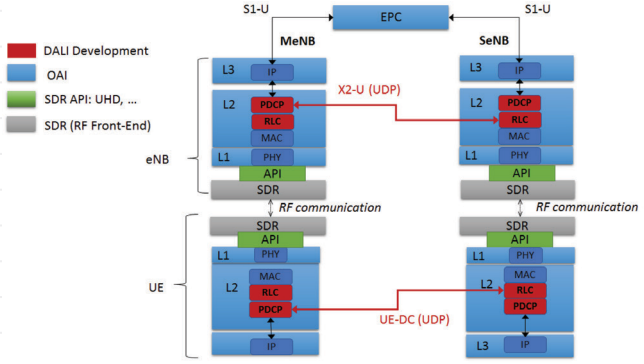
CONCEPT

DALI Dual Connectivity setup (Figures X and Y) 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 stablished to communicate PDCP and RLC layers of different eNB nodes through X2 interface and a newly implemented DALI UE DC interface for communication between UEs.

As shown in Figures X and Y, 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 distro of Linux with 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.v



Architecture of DALI solution with ns-3 based real-time LTE devices



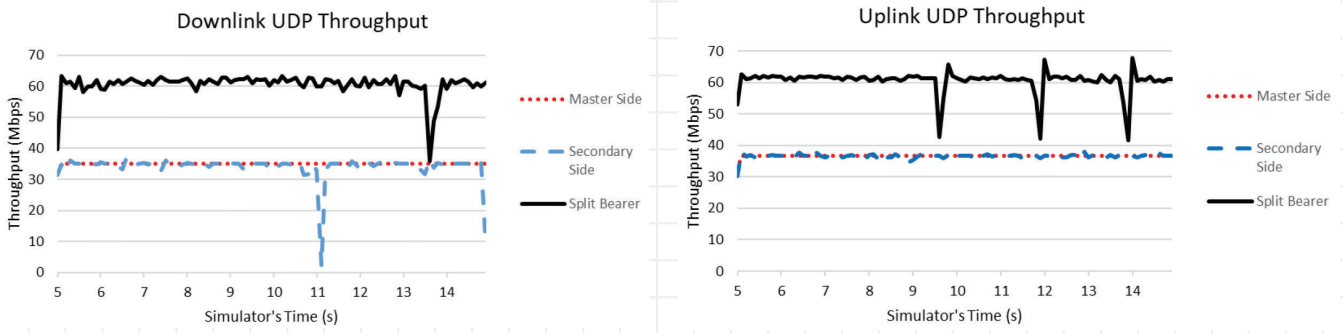
Architecture of DALI solution with OAI-based real-time LTE devices

www.orca-project.eu

DALI
Dual Connectivity Solution for ORCA

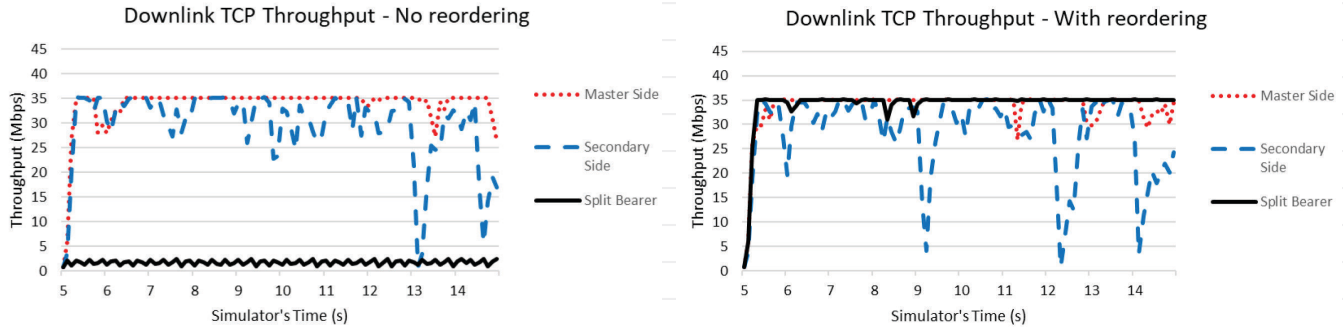
MAIN RESULTS

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 Z. 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.



UDP Throughput: Downlink (up) and Uplink (bottom)

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



TCP Throughput for Downlink without (up) and with (bottom) with PDCP reordering function

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.

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.

Thanks to the ORCA facility that we were able to implement and validate our LTE Dual Connectivity solution within a realistic experimentation setup.

Orchestration and Reconfiguration Control Architecture

MISO

Millimeter - Wave Open Experimentation Platform

Open Call partner
IMDEA Networks Institute

Patron
KU Leuven

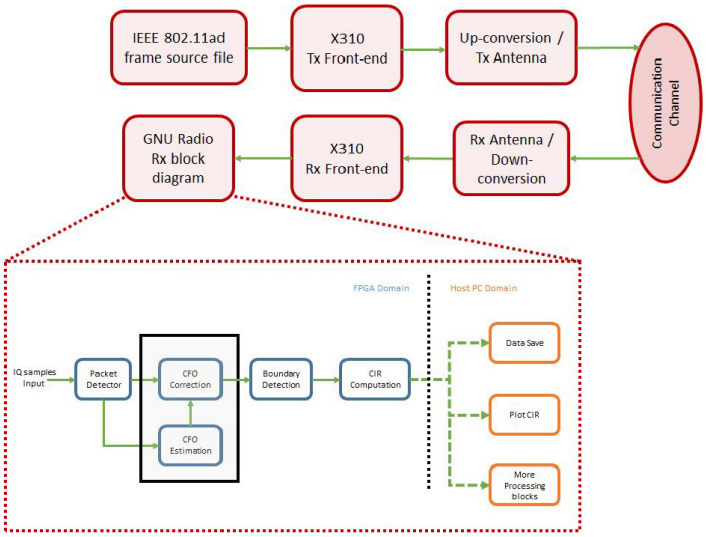
OBJECTIVES

In this project we proposed a mixed hardware-software design for mm-wave experimentation on software-defined radios (SDR). Specifically, we proposed to design and implement the hardware blocks to decode the preamble of single carrier (SC) IEEE 802.11ad compliant frames at a 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.

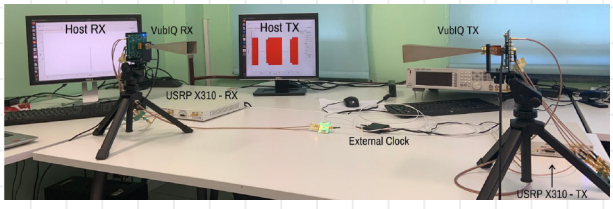
CHALLENGES

Reducing the sampling frequency of a 60GHz communication system is a big challenge since the imperfections do 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, meet timing constraints and leave space to further upgrading of the design.

CONCEPT



Block diagram of the developed mixed hardware-software system



Example experiment setup for on-site experiments.

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.

Tx side is implemented including IEEE 802.11ad compliant single carrier frames from files which are outputted from the X310 USRP devices using a basic Tx 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).

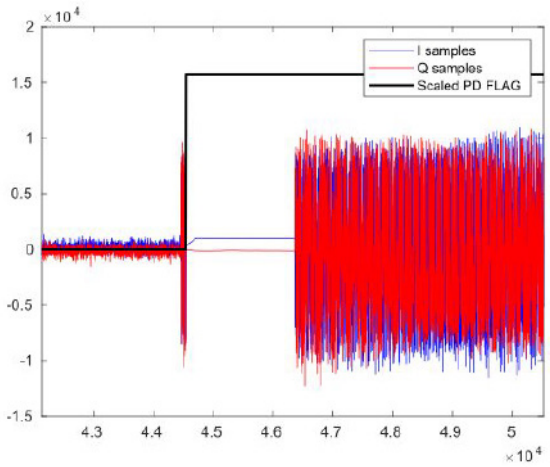
At the receiver side, after down conversion of the signal to baseband, it is fed to the USRP device using a basic Rx daughterboard. At the FPGA side, the signal passes through the RFNoC blocks implementing the preamble processing blocks, as can be seen in Figure 28. After CIR computation, 128 complex 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.

www.orca-project.eu

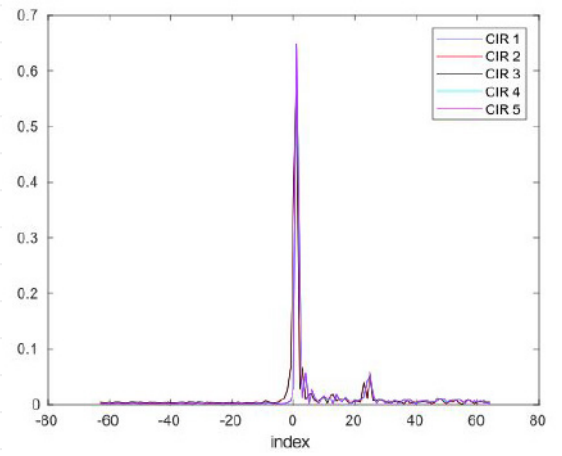
MISO
Millimeter - Wave Open Experimentation Platform

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.



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



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

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.

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.
Thanks to the ORCA project we were able to design and implement a highly flexible platform for local and remote millimeter-wave experimentation on lost cost software-defined radios.

Orchestration and Reconfiguration Control Architecture

POLAR CODE

Efficient Polar Encoding and Multi-Mode Decoding for FPGAs

Open Call partner
École Polytechnique Fédérale de Lausanne

Patron
Technische Universitaet Dresden

OBJECTIVES

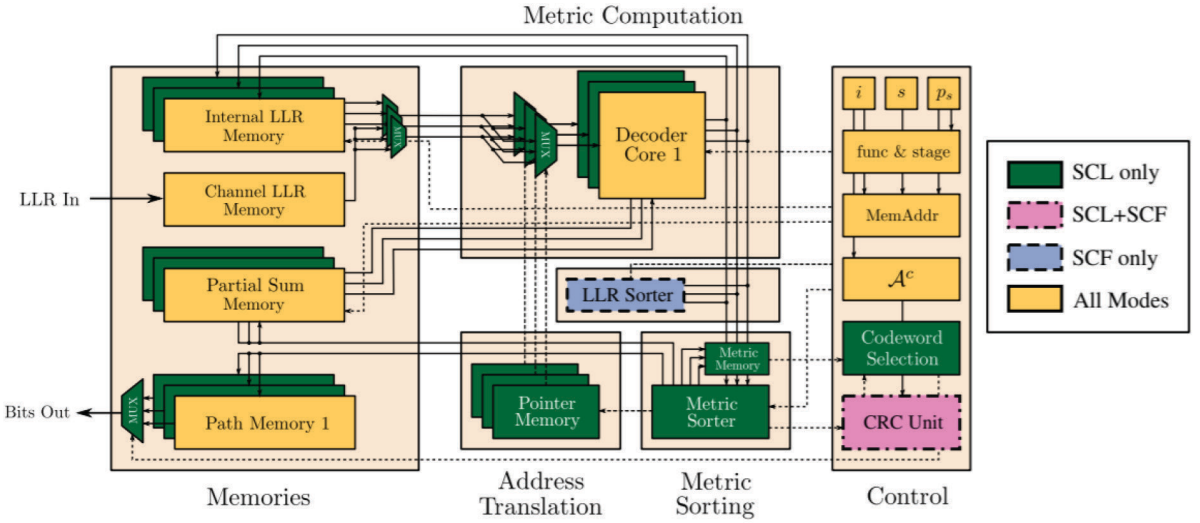
The first goal of this project was to develop an FPGA Polar decoder to add Polar coding capabilities to ORCA. This is interesting due to their superior error-correction performance and due to the timely relevance as Polar codes have been established as part of 5G-NR. Our second goal was to provide a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

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 tradeoffs. Combining specifically SC, SCL, and SCF decoding allows for effective reuse of resource- hungry core components. The sequential SCF decoding is specifically designed to allow runtime tuning between performance and complexity.

CONCEPT

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 tradeoffs 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. The provided figure below illustrates the components of the decoder.



Multi-mode polar decoder architecture

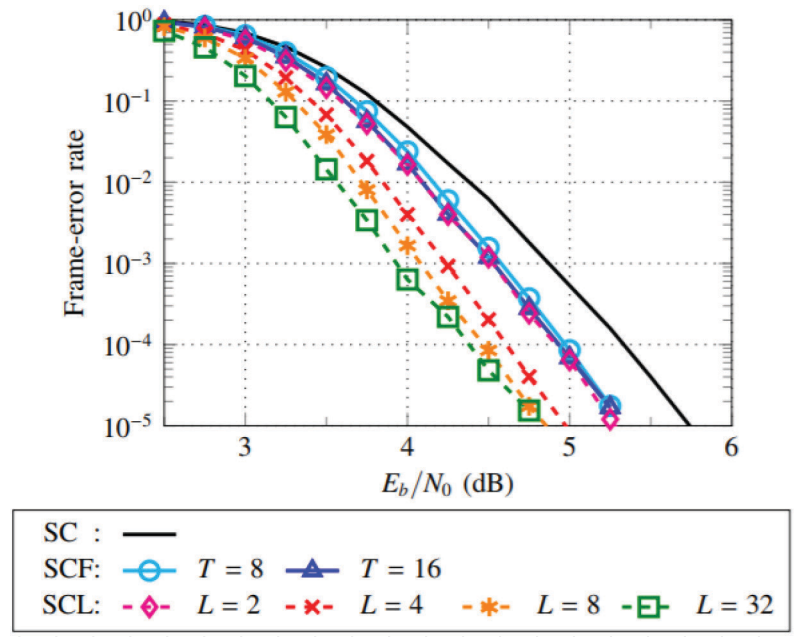
www.orca-project.eu

POLAR CODE

Efficient Polar Encoding and Multi-Mode Decoding for FPGAs

MAIN RESULTS

The decoder developed in this project supports several modes that allow tradeoffs between performance and complexity. The provided figure illustrates these tradeoffs. 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.



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.

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.

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.
Thanks to the ORCA facility, we were able to provide a re-usable IP core for testing and integrating Polar codes into a real-world test environment.

Orca

Orchestration and Reconfiguration Control Architecture

ReproRun

Reprogramming FPGA devices at run time using partial reconfiguration in SDR platforms

Open Call partner
Centre Tecnològic Telecomunicacions Catalunya

CTTC

Patron
imec

imec

OBJECTIVES

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.

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.

CONCEPT

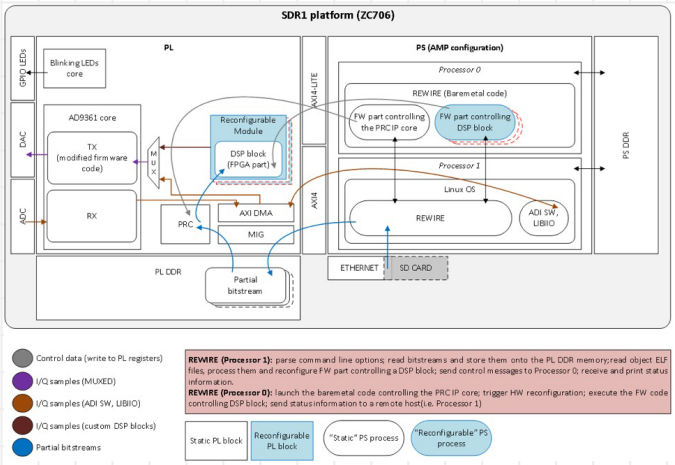
ReproRun is able to seamlessly reconfigure at run-time part of the PL area of an FPGA device with a partial bitstream and also the firmware running at PS embedded in the same FPGA device (i.e., hardwired or soft microprocessor). The firmware among other functions includes a software bare metal application (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. 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 (objet files) from the TFTP server running at a remote host (TFTP runs on top of the 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 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 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 SDR2 respectively). In SDR1 the partial bitstreams are a DDS function and a LTE waveform playback function. In SDR2 the partial bitstream is the CORDIC function of the siggen RFNoC example.

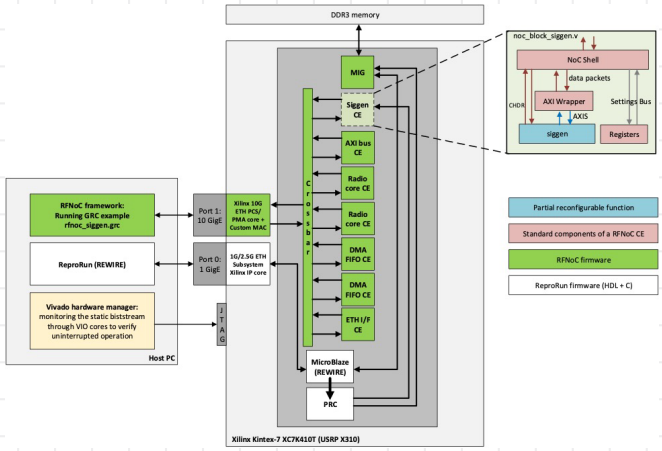
ReproRun

Reprogramming FPGA devices at run time using partial reconfiguration in SDR platforms

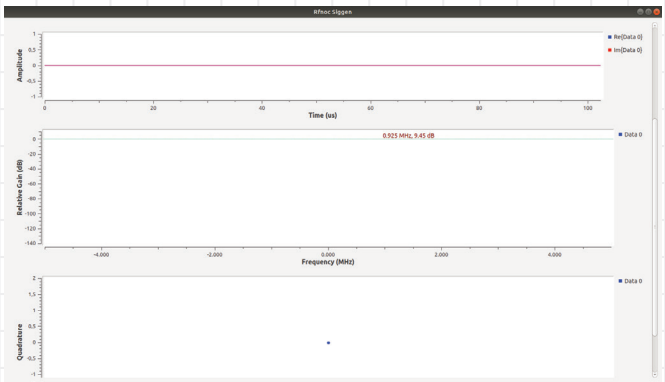
CONCEPT

The concept and experimental setup of the two SDR platforms can be seen in the figures that follow:

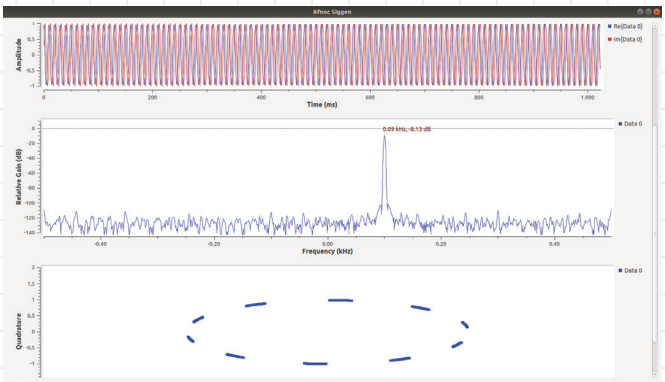




MAIN RESULTS



The Siggen RFNoC GRC example is not working when only the static bitstream is configured.



After configuring the partial bitstream, the siggen RFNoC example produces the correct/expected output.

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.

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.

Thanks to the ORCA facility we were able to develop a run-time partial reconfirmation framework for FPGA devices featured in two popular SDR platforms, as well as develope a flexible SDR framework featuring run-time reconfigurable FPGA-accelerated DSP functions"