Ultra-wideband SDR architecture for AMD RFSoCs “using” PYNQ based GNU Radio blocks

Marius Šiaučiulis, Louise H. Crockett and Robert W. Stewart

https://github.com/marsiau

https://github.com/strath-sdr/

Contents

Introduction

StrathSDR

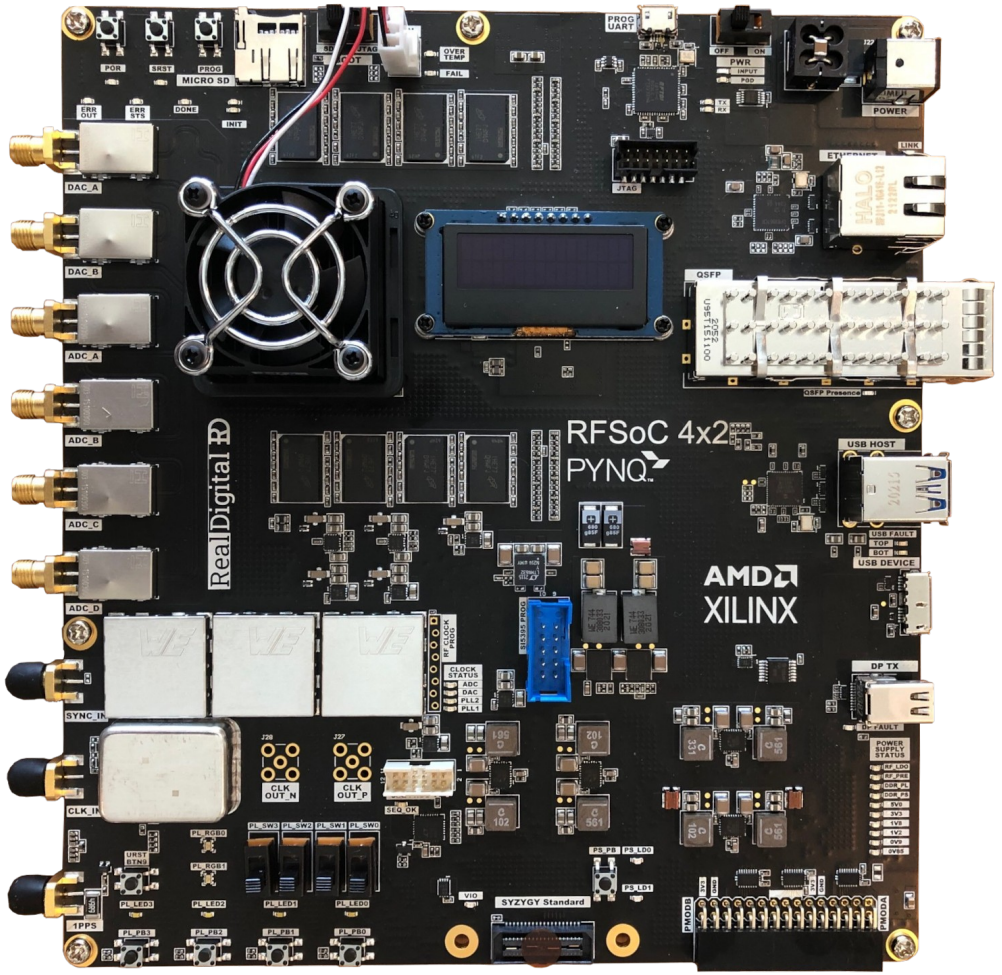
- Modern Software Defined Radio (SDR) systems are capable of multi-gigabit-per-second sampling rates, producing/consuming massive amounts of digitized RF data.
- Developing systems for SoC devices can be challenging for non-specialists, due to the learning curve of the design tools and processes.
- It is often convienient to generate signals using software frameworks before moving to hardware implementation.
- We present an open-source RFSoC design capable of Tx/Rx offload in a 100GBit/s network channel with minimum latency. This design uses the:
  - AMD PYNQ & RFSoC 4x2 platform – next couple of slides.
  - GNU Radio – open-source development toolkit for SDR applications.

Introduction – PYNQ

- AMD PYNQ is an open-source project that exposes Intellectual Property (IP) cores within the FPGA fabric as a collection of Python objects.
- Provides Python APIs for:
  - Loading and managing FPGA bitstreams (called overlays)
  - Interacting with memory mapped interfaces
  - Handling PL interrupts (through asyncio)
  - Direct Memory Access (DMA) IP
- RFSoC-PYNQ Python wrappers for RFSoC specific functionality:
  - xrfclk - to configure external RF reference clocks
  - xrfdc - controlling and interacting with the RF Data Converters (RFDCs)
  - xsdfec - drivers for Soft-Decision Forward Error Correction (SD-FEC) block

Introduction – RFSoC 4x2

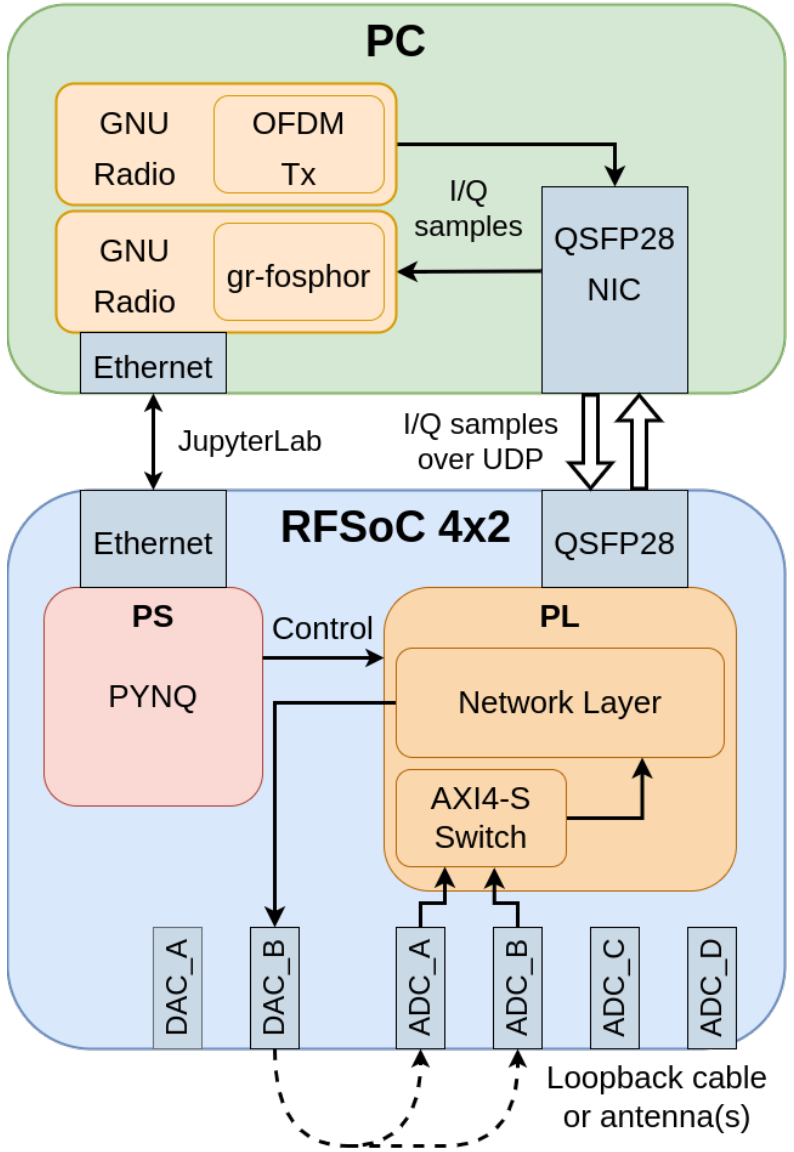
- RF Digital to Analog Converters (RF-DACs)
  - 2x
  - 14-bit
  - 9.85Gsp/s
- RF Analog to Digital Converters (RF-ADCs)
  - 4x
  - 14-bit
  - 5Gsp/s
- 100Gbit/s QSFP network interface
- 64-bit Quad core Arm CPU running PYNQ Linux
- Programmable Logic (PL / FPGA)



RFSoC 4x2
http://www.rfsoc-pynq.io/rfsoc_4x2_overview.html

Hardware Setup Overview

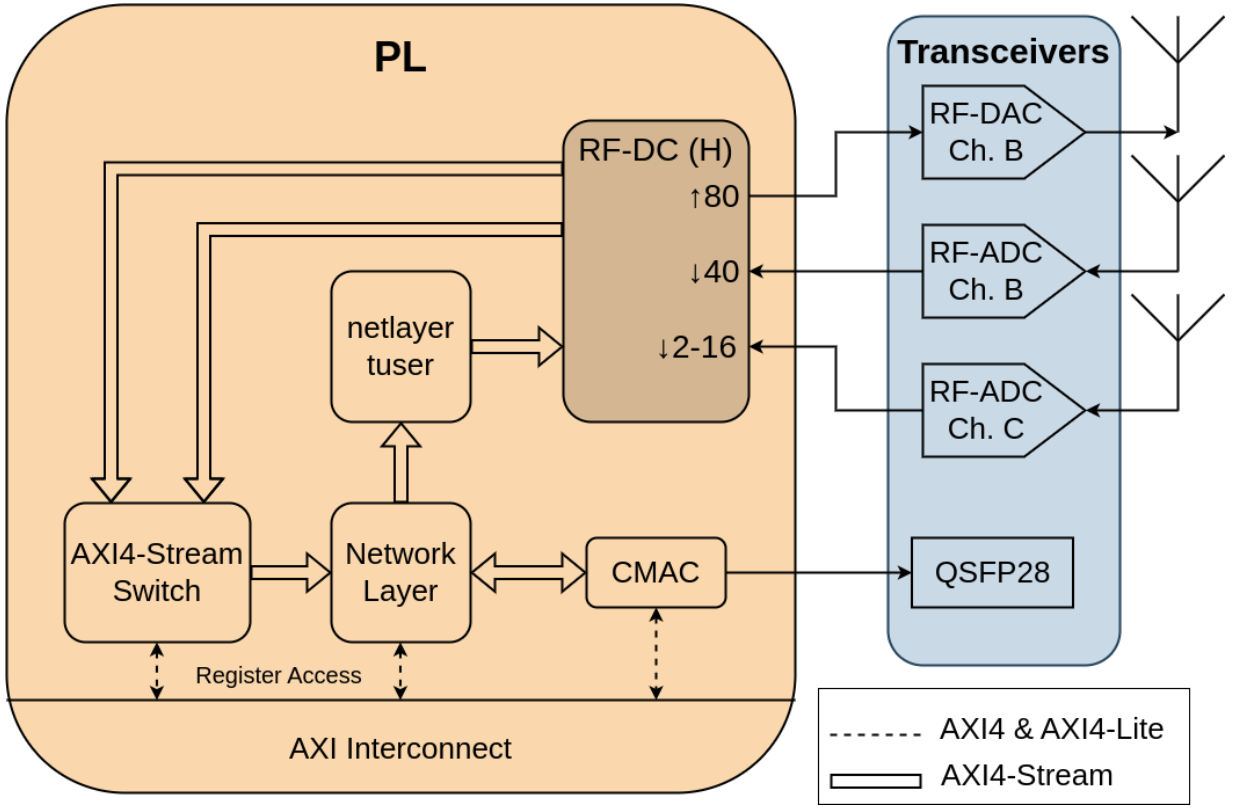
- GNU Radio flograph is used to generate OFDM signals at 61.44 MSPS.
- I/Q samples are sent/received over a 100Gbit QSFP Network.
- GNU Radio with gr-fosphor is utilized for Graphics Processing Unit (GPU) accelerated, real-time spectrum plotting.
- An Ethernet connection is used to access a JupyterLab interface running on the board and remotely control the RF-DC parameters.



Design overview diagram.

Hardware Architecture

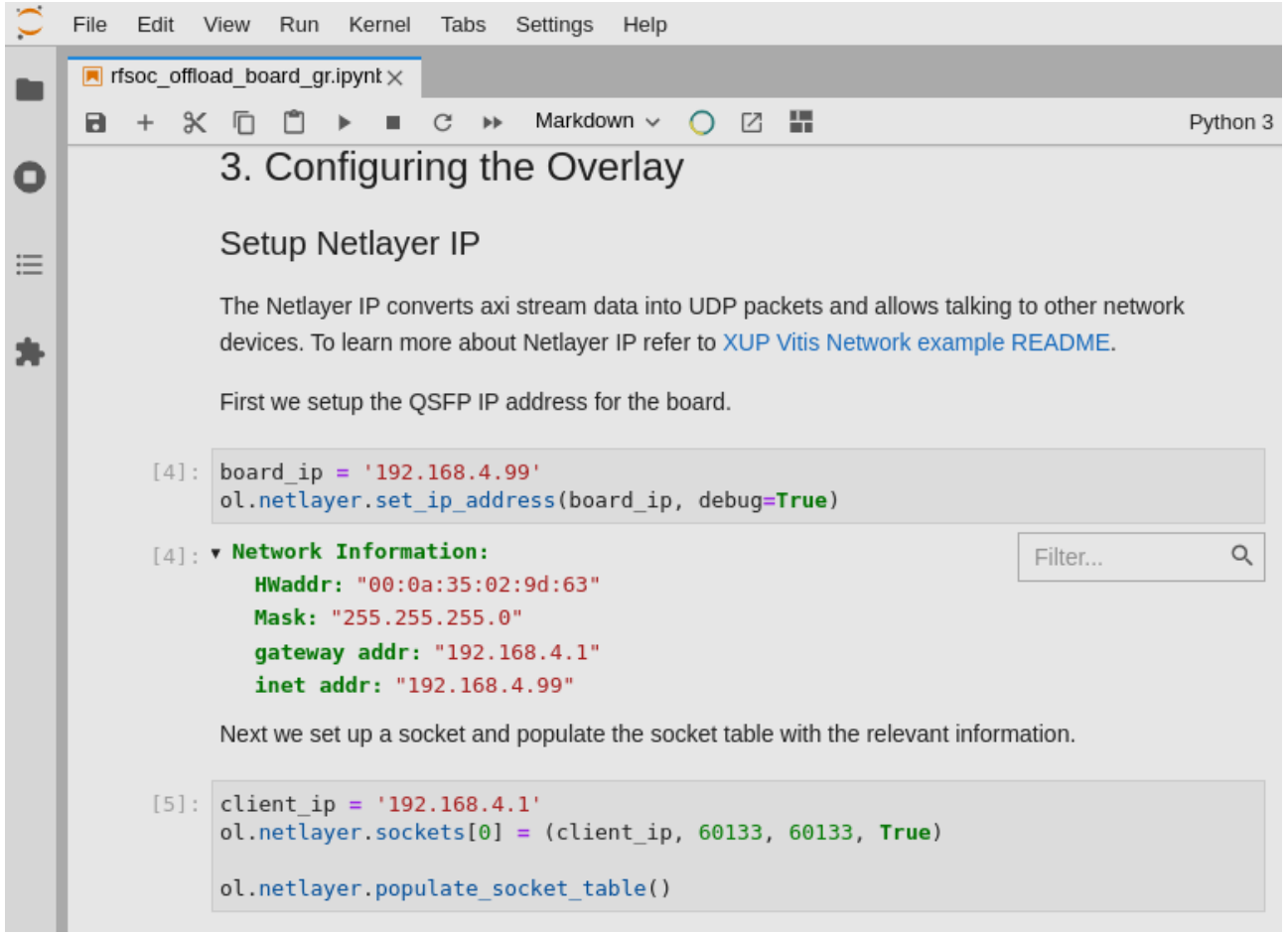
- The high-speed network architecture is implemented entirely on the Programmable Logic (PL), enabling maximum network bandwidth and minimum latency.
- User Datagram Protocol (UDP) is used for data transfer over the QSFP28 network.
- RF-DAC Ch. B Digital Up-Converter (DUC) interpolates the signal by 80.
- RF-ADC Ch. B Digital Down-Converter (DDC) decimates by 40.
- RF-ADC Ch. C setup with flexible decimation of 2-16.



High level overview of the hardware system.

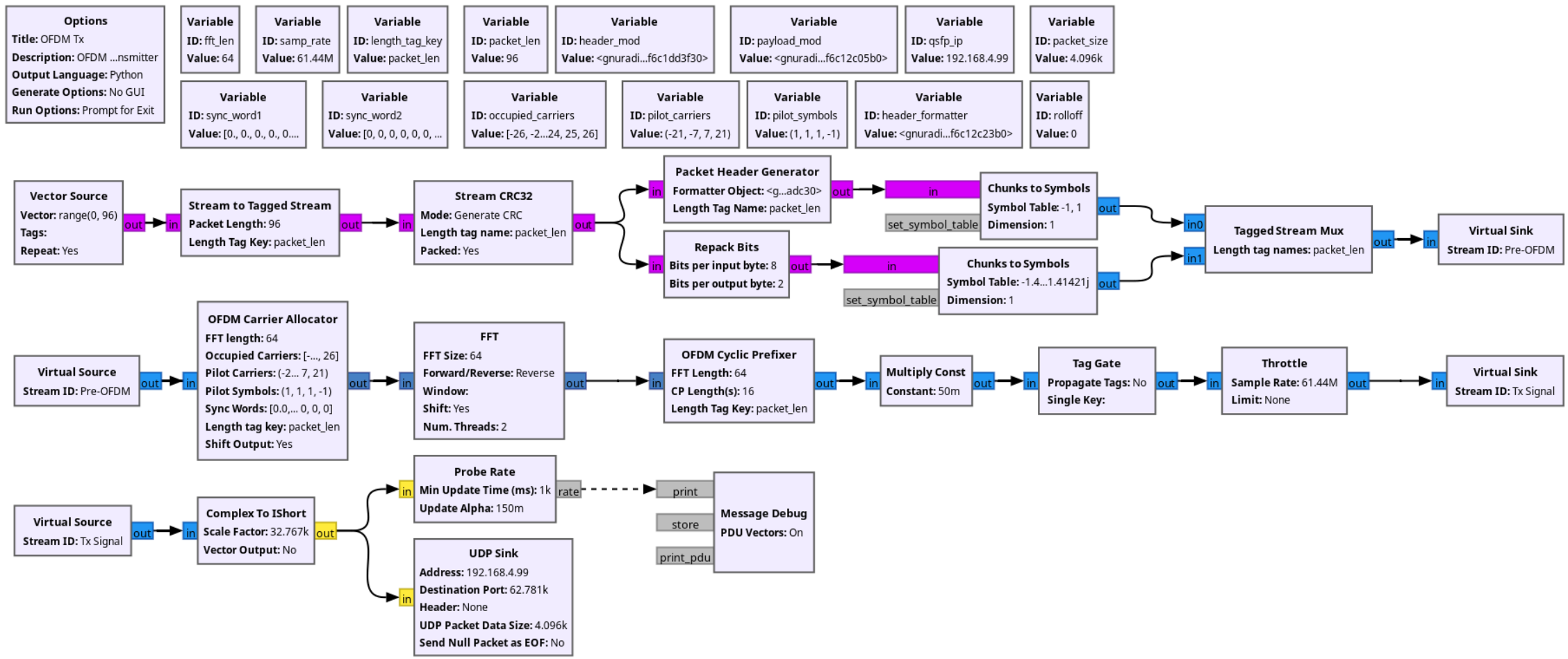
Software Architecture - Board

- JupyterLab provides an interactive Python session for dynamic adjustment of software and hardware parameters.
- PYNQ framework provides Python APIs for interacting with the Programmable Logic (PL), controlling parameters of the RF-ADCs, RF-DACs, and programming the internal reference clocks.
- Starts an XML-RPC server for remote commands.



JupyterLab interface

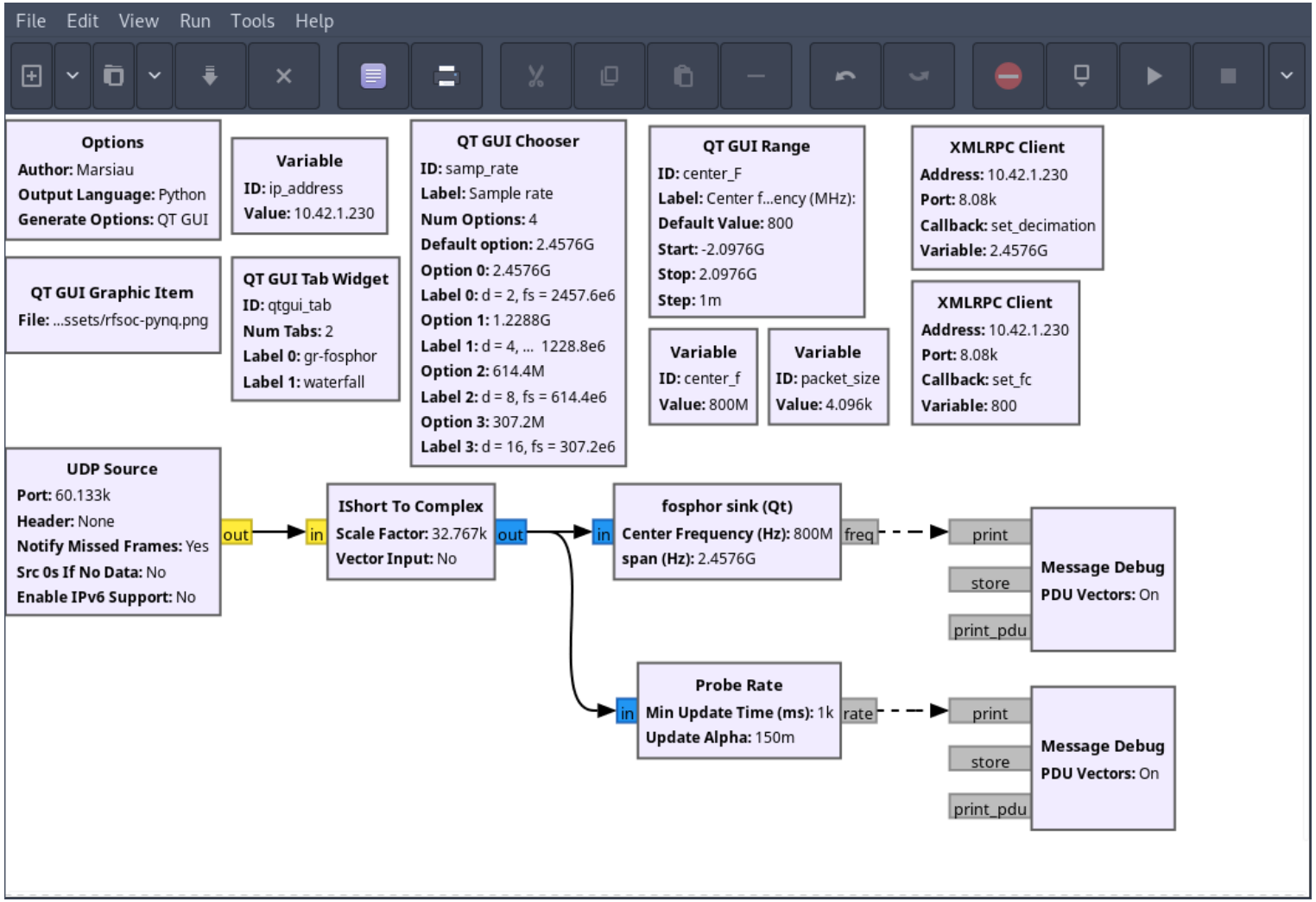
GNU Radio Tx flowgraph



GNU Radio Tx flowgraph

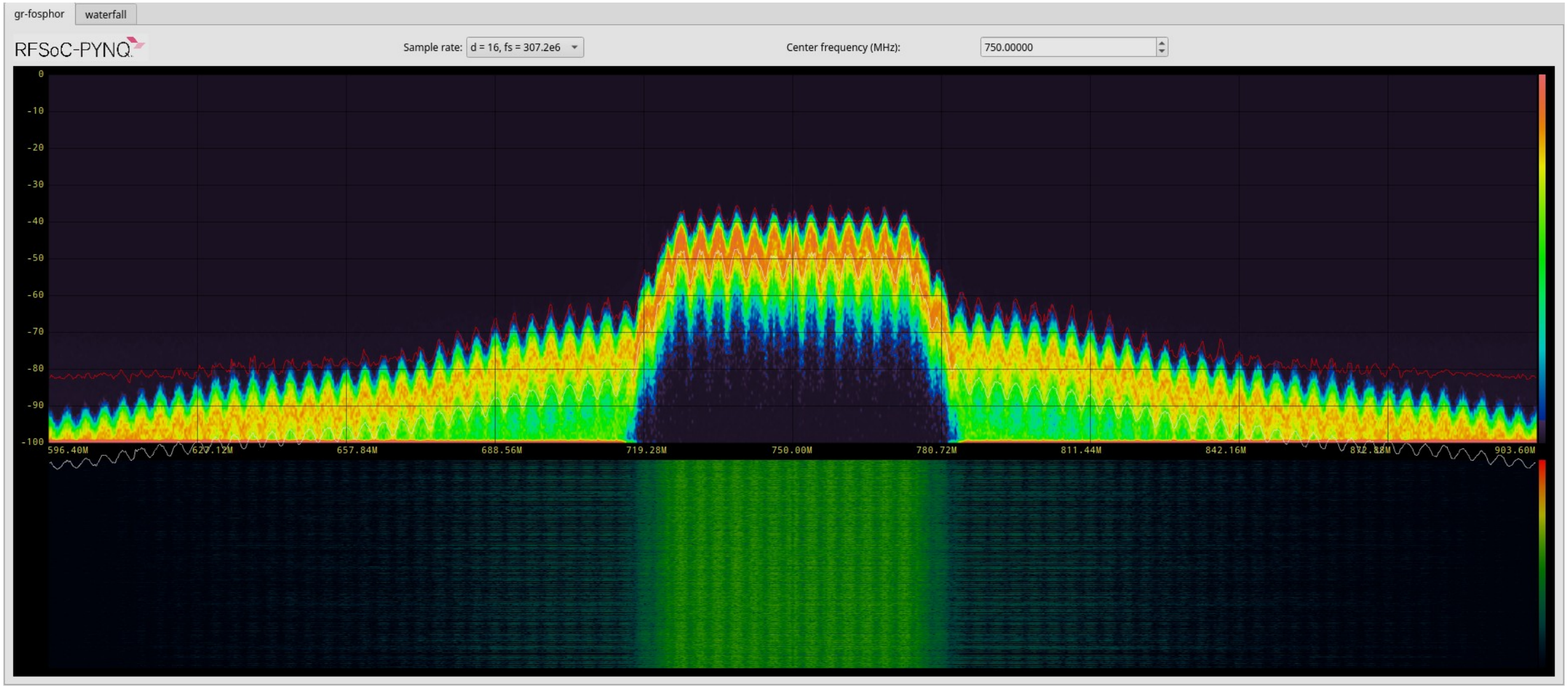
GNU Radio Rx flowgraph

- XMLRPC blocks enable adjustment of remote board parameters.
- Variable Blocks are used to make the design parametric.
- QT GUI Blocks facilitate the user input options.



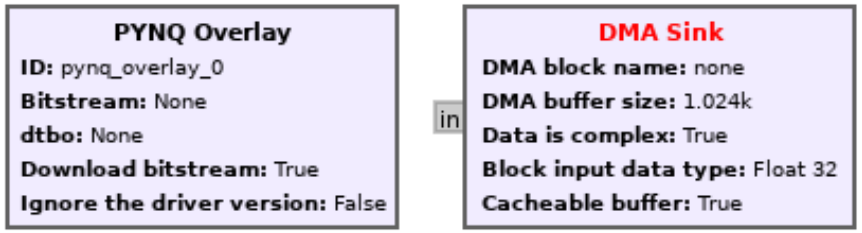
GNU Radio Rx flowgraph

gr-fosphor


gr-fosphor
waterfall
RFSoC-PYNQ
Sample rate:
d = 16, fs = 307.2e6
Center frequency (MHz):
750.00000
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
596.40M
627.12M
657.84M
688.56M
719.28M
750.00M
780.72M
811.44M
842.16M
903.60M

gr-pynq blocks

Block Name	Function	Status	Test Status
Overlay	Loads & manages FPGA bitstreams	Done	Done
DMA Sink	PS to PL data transfer	Done	Done
DMA Source	PL to PS data transfer	Done	WIP
xrfclk	Controls external RF clocks	WIP	WIP
xrfdc tile	Controls RF-DC tiles	WIP	WIP



Example gr-pynq blocks

Summary / Conclusions

- Presented open-source RFSoC design capable of Tx/Rx offload in a 100GBit/s network channel with GNU Radio integration.
- Remote radio control can be achieved using XML-RPC blocks.
- The architecture opens a UDP sockets allowing for arbitrary data transfer, making it applicable not only to RF data and communications but also to various instrumentation applications.
- With the increasing capabilities of SDR technology, offloading to/from power-limited edge devices from/to datacenters will become increasingly important, especially for spectrum monitoring and disaggregated radio applications.

StrathSDR

Thanks for listening!
Engage with us:

https://sdr.eee.strath.ac.uk

@strathSDR

github.com/strath-sdr

University of Strathclyde Glasgow