DOI: 10.35598/mcfpga.2025.032

GNU Radio in SDR Systems: History, Architecture and Integration with FPGA for High-Performance Signal Processing

Oleksandr Vorgul
Scientific adviser
ORCID 0000-0002-7659-8796
dept.Microprocessor Technologies and System
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Maksym Skorbatiuk
dept.Microprocessor Technologies and System
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
maksym.skorbatiuk@nure.ua

***Abstract*– This report explores the role of the GNU Radio platform and the USRP (Universal Software Radio Peripheral) in the development of software-defined radio (SDR) systems using FPGAs. The article discusses the historical aspects of GNU Radio development , the USRP architecture, and the mechanisms for interacting with programmable logic. Particular attention is paid to the advantages of offloading digital signal processing (DSP) to FPGAs for tasks requiring high performance and low latency. Practical aspects of implementing SDR systems are analyzed, including support for wideband communication protocols and cognitive radio. The results demonstrate the effectiveness of the GNU Radio-USRP-FPGA bundle as a universal platform for scientific research and engineering development.**

Keywords— GNU Radio, USRP, SDR, FPGA, Signal Processing, Software-Defined Radio, Ettus Research, Rfsoc.

I. Introduction

The development of software-defined radio (SDR) has revolutionized the design of wireless systems. Key to this process was the emergence of the open source GNU Radio framework and specialized hardware USRP (Universal Software Radio Peripheral), developed by Ettus Research [1]. The combination of the flexibility of software signal processing with the performance of programmable logic integrated circuits (FPGAs) has created the basis for implementing complex real-time communication protocols. This paper explores the architectural synergy of GNU Radio , USRP, and FPGAs, as well as their application in modern telecommunications and research systems.

II. Historical context and technological foundations

2.1 Evolution of GNU Radio

The GNU Radio project , initiated by Eric Blass in 1998 and incorporated into the GNU ecosystem in 2001, was a response to the need for an open source toolkit for SDR development [2]. Its graphical environment (GNU Radio Companion (GRC) and Python /C++ language support have abstracted the complexity of DSP algorithms, making SDR technologies accessible to a wide range of researchers.

2.2 USRP Architecture

USRP Platform Introduced by Ettus Research in 2004, became the hardware basis for GNU Radio [3]. Conceptually, USRP is a heterogeneous system combining:

- Radiofrequency frontends (RF transceivers) with a wide frequency range;
- High speed ADC/DAC ;
- Data transmission interfaces (Ethernet , PCIe);
- Field programmable logic arrays (FPGAs) for signal pre-processing.

A key innovative aspect of the USRP was the use of FPGAs to offload DSP operations (filtering, decimation, interpolation, FFT), which relieved the host processor and allowed for wideband signal processing [4].

III. Integration of FPGA into the SDR pipeline

3.1. The Role of FPGA in Signal Processing

Modern USRPs (X3xx, N3xx, E310/E320 series) use Xilinx FPGAs Zynq or Intel Cyclone , implementing:

- Digital frequency converters (DDC/DUC);
- Polyphase filters for decimation/interpolation;
- Demodulation/modulation of basic signals (QPSK, QAM);
- Adaptive processing (e.g. for cognitive radio) [5].

An example of deep integration is the Xilinx platform Zynq UltraScale+ RFSoC , where the ADC/DAC are directly connected to the programmable logic, minimizing latencies [6].

3.2. GNU Radio as a control environment

GNU Radio communicates with the USRP FPGA via :

1. UHD Driver (USRP Hardware Driver), which provides configuration of RF parameters and transmission of IQ data.
2. RFNoC (RF Network on Chip) for developing custom FPGA IP cores controlled from GRC [7].

3. (Source / Sink) blocks for data streaming.

This architecture allows for task distribution: time-critical operations are performed in the FPGA, and higher-level protocols are performed in Python /C++.

IV. Practical applications

4.1. Academic and industrial cases

- 5G/6G Prototyping : Massive MIMO and OFDM Implementation on USRP X410 with FPGA [8].
- Cognitive Radio : Dynamic frequency hopping with real-time spectrum analysis.
- Radio astronomy : Correlation processing of signals in interferometers (CASPER project).
- Cybersecurity : Analysis of vulnerabilities of wireless protocols (WiFi , LoRaWAN).

4.2. Advantages of architecture

- Performance : Processing of streams up to 200 MS/ s in FPGA [3].
- Flexibility : Quick change of functionality through reprogramming of FPGA and GRC graphs.
- Cost-effectiveness : Lower development costs compared to ASIC.

V. Conclusions

The symbiosis of GNU Radio , USRP and FPGA has formed a de facto standard for SDR development. The ability to offload calculations to programmable logic solves the problem of the "bottleneck" in processing wideband signals, while maintaining the flexibility of software configuration. Further development of RFSoC platforms and high-level synthesis (HLS) tools for FPGA will strengthen the position of this ecosystem in the implementation of 6G and IoT communication systems.

References

[1] Blossom , E. GNU Radio : Tools for Exploring the Radio Frequency Spectrum // Linux Journal . – 2004. – Vol . 2004, No. 122. – P. 4.

[2] GNU Radio Project . About GNU Radio [Electronic resource]. – URL: https://www.gnuradio.org/about/ (date of access: 02.07.2025).

[3] Ettus , M. Universal Software Radio Peripheral (USRP) Family of Products // Proceedings of the SDR Forum Technical Conference . – 2008.

[4] Tennina , S. et al . IEEE 802.15.4 and ZigBee as Enabling Technologies for Low Power Wireless Systems // Springer , 2013. – 198 p .

[5] Ettus Research . USRP X310 Product Sheet [Electronic resource]. – URL: https://www.ettus.com/wp-content/uploads/2019/01/X310-DS.pdf (date of access: 02.07.2025).

[6] Xilinx . Zynq UltraScale+ RFSoC Data Sheet [Electronic resource]. – URL: https://www.xilinx.com/support/documentation/data_sheets/ds889-rfsoc-overview.pdf (accessed: 02.07.2025).

[7] Braun , M. et al . RF Network-on-Chip : An Open Source Framework for FPGA-Based SDR // IEEE Access . – 2020. – Vol . 8. – P. 226066–226077.

[8] Niknam , S. and al . Massive MIMO for 5G: Prototyping and Testing // IEEE Communications Magazine . – 2020. – Vol . 58, No. 2. – P. 88–94.