Real-time Wideband Software-defined Radio with Python Programmability based on RFSoC

Wei Cheng, Zhihui Gao, Tingjun Chen
Department of Electrical and Computer Engineering, Duke University

Abstract

Next-generation wireless networks necessitate large signal bandwidth to support the growing demands of high data rates, which poses significant challenges in the design of real-time radio platforms. We demonstrate SPEAR, a real-time wideband software-defined radio (SDR) utilizing the Xilinx RFSoC ZCU216 evaluation board. SPEAR leverages a customized "Streaming Direct Memory Access (DMA)" IP to address the latency issues associated with DMA control, thereby enabling high bandwidth data streaming in real-time. It also features a Python-based hardware configuration tool and signal processing framework incorporating an OFDM-based Physical layer. We showcase a real-time data link using the direct RF radio architecture between two RFSoC ZCU216 boards, achieving an error vector magnitude (EVM) of 3.2% for 256QAM across a bandwidth of 1.25 GHz.

CCS Concepts

• **Networks** → **Network experimentation**; • **Computer systems organization** → **Real-time system architecture**.

Keywords

Software-defined radios, RFSoC, hardware-software co-design

ACM Reference Format:
Wei Cheng, Zhihui Gao, Tingjun Chen. 2024. Real-time Wideband Software-defined Radio with Python Programmability based on RFSoC. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3636534.3698855

1 Introduction

Wideband wireless systems require higher sampling rates, challenging datapath designs with increased data rates. For example, 5G new radio (NR) accommodates a single channel up to 400 MHz in the frequency range 2 (FR2) [2] and IEEE 802.11ad supports a bandwidth of 2.16 GHz in the 60 GHz unlicensed band [1]. The growing need for larger bandwidth necessitates the design of high-speed datapaths on all software-defined radio (SDR) platforms.

Recently, Radio Frequency System-on-Chip (RFSoC) technology emerged as a promising solution to address the aforementioned challenges by providing built-in multi-GHz sampling rates RF converters and high-end FPGA fabrics for datapath routing. RFSoC has been used to build high-performance SDR platforms and fully digital beamformers [8, 15, 16, 18], MIMO radios in both FR1 and FR2 [6, 11–13], radars [19], as well as quantum control [7, 14], and analog computing platforms [17, 20, 21].

In this demonstration, we present SPEAR (**S**treaming-based **P**ython-**E**nh**A**nced **R**FSoC) [9], an SDR platform based on the Xilinx RFSoC ZCU216 evaluation board [5] capable of supporting a real-time bandwidth of up to 1.25 GHz. SPEAR consists of real-time streaming-based datapaths for the transmitter (TX) and receiver (RX), and a Python interface that enables users to design waveforms or experiment with digital signal processing (DSP) algorithms using the Python programming language. Overall, SPEAR offers flexibility for users who are interested in operating a real-time RF communication system with GHz+ bandwidth. Both the software and hardware design of SPEAR are open-sourced in [4].

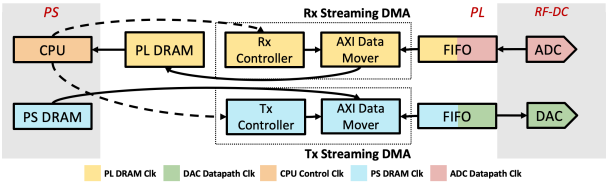

Figure 1: Hardware architecture of SPEAR. Both CPU and PS DRAM belongs to the Processing System (PS) while remaining components belong to the Programmable Logic (PL). DAC and ADC belong to RF Data Converters (RF-DC).

2 System Design of SPEAR

Fig. 1 depicts the dataflow of the hardware design of SPEAR, where the generated I/Q samples on the TX side are streamed from the processing system (PS) DRAM to the digital-to-analog converter (DAC), and the captured I/Q samples on

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3698855

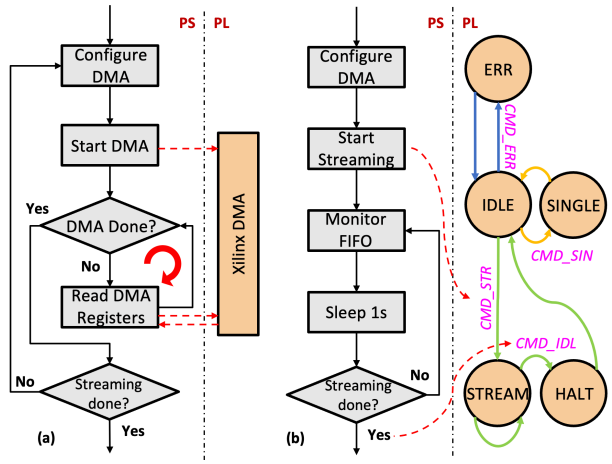

Figure 2: The software execution flow of DMA control with (a) CPU busy waiting, and (b) hardware-assisted streaming.

the RX side are streamed from the analog-to-digital converter (ADC) to the programmable logic (PL) DRAM. Our DRAM-based solution can hold samples as long as DRAM capacity permits, in contrast to other BRAM-based designs [3, 11], which only holds a limited number of samples. Additionally, this asymmetric data flow (streaming from/to PS/PL DRAM) prevents resource contention that may occur when reading and writing high-speed I/Q sample streams from/to the same piece of memory.

Streaming DMA based on hardware FSM. The Streaming Direct Memory Access (DMA) is the most important hardware component that distinguishes SPEAR from previous works [11–13]. It consists of a customized controller and Xilinx's AXI Data Mover IP, as depicted in Fig. 1. This controller internally functions as a finite state machine (FSM), whose transition diagram is shown in Fig. 2(b).

Starting in the IDLE state, the system moves to the SINGLE state upon receiving a CMD_SIN command, initiating a single DMA transfer for specified samples to/from memory. The transition to the STREAM state occurs through a CMD_STR command, initiating continuous back-to-back DMA transfers between DAC/ADC and memory. A CMD_IDL command must be issued to transition the controller from a STREAM state to a HALT state, effectively stopping data streamin.

Software architecture and DMA control method. For both TX and RX datapaths, it is essential to design a DMA that can efficiently and repeatedly transfer samples between memory (PS and/or PL DRAM) and the DAC/ADC to ensure real-time performance. As shown in Fig. 2, in contrast to the control of DMA of existing designs, SPEAR offloads the inner control loop to the hardware FSM. Other RFSoC-based platforms [11, 12] rely on the DMA control flow shown in Fig. 2(a) and cannot ensure real-time performance due to the incurred latency caused by the polling of DMA registers, as indicated by the red circle in Fig. 2(a). In our design, CPU only wakes up for FIFO monitoring tasks, which monitors

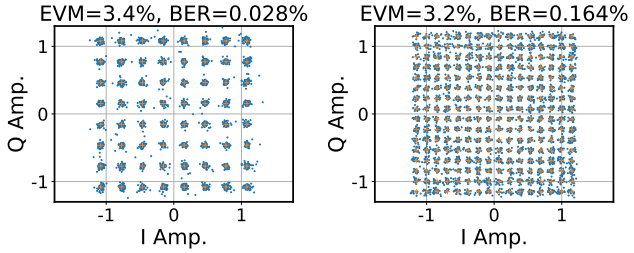

Figure 3: Constellation diagrams of 64QAM and 256QAM for a 1.25 GHz bandwidth link.

real-time performance by checking the remaining samples within the FIFO. If the TX FIFO is never empty and RX FIFO is never full, real-time streaming is guaranteed and can be verified at the circuit level.

DSP pipeline employing an OFDM PHY. We validate the functionality of our platform by using a general orthogonal frequency-division multiplexing- (OFDM-) based DSP pipeline implemented in Python [10]. For each measurement, we generate and analyze 102 OFDM symbols, where the first and the last OFDM symbols serve as the pilot symbols for channel state information (CSI) estimation and calibration, and the remaining 100 OFDM symbols carry data using 16QAM, 64QAM, 256QAM, and 1024QAM modulations. We focus on (*i*) signal-to-noise ratio (SNR), (*ii*) error vector magnitude (EVM), and (*iii*) bit error rate (BER) and use them as metrics to evaluate the quality of transmission (QoT) for the link.

3 Demonstrations

We demonstrate a real-time data link between two RFSoC boards supporting a bandwidth of 1.25 GHz. The DAC on one RFSoC board (TX) is configured at 2.5 GSaps sampling rate with 2× interpolation, and the ADC on another RFSoC board (RX) is configured at 2.5 GSaps sampling rate with 2× decimation. The DAC and ADC operate in the I/Q-to-real and real-to-I/Q mode, respectively, in a direct RF radio architecture. This configuration ensures that the 1.25 GHz bandwidth channel centered at carrier frequency of 625 MHz fully occupies the 1st Nyquist zone between 0–1.25 GHz. SPEAR's Python programmability allows users to easily generate, capture, and visualize customized signal waveforms. The QoT of the received signal after DSP, including the EVM and BER, is also shown in Fig. 3, where the data link with 1.25 GHz bandwidth and 256QAM achieves an EVM and BER of 3.2% and 0.2%, respectively.

Acknowledgments

This work was supported in part by NSF grants CNS-2128638, CNS-2211944, and AST-2232458, and the Center for Ubiquitous Connectivity (CUbiC), sponsored by Semiconductor Research Corporation (SRC) and Defense Advanced Research Projects Agency (DARPA) under the JUMP 2.0 program.

References

[1] 2012. IEEE Standard for Information Technology - IEEE 802.11ad.

[2] 2020. 5G NR Base Station (BS) Radio Transmission and Reception. Technical Specification (TS) 38.211. 3rd Generation Partnership Project (3GPP).

[3] 2024. Designing with the Zynq UltraScale+ RFSoC. https://learningcatalog-amd.netexam.com/Certification/46089/designing-with-the-zynq-ultrascale-rfsoc.

[4] 2024. SPEAR GitHub. https://github.com/functions-lab/SPEAR.

[5] 2024. Zynq UltraScale+ RFSoC ZCU216 Evaluation Kit. https://www.xilinx.com/products/boards-and-kits/zcu216.html.

[6] Hoda Barkhordar-Pour, Jin Gyu Lim, Mohammed Almoneer, Patrick Mitran, and Slim Boumaiza. 2023. Real-time FPGA-based implementation of digital predistorters for fully digital MIMO transmitters. In *Proc. IEEE IMS'23*.

[7] Rodolfo Carobene, Alessandro Candido, Javier Serrano, Alvaro Orgaz-Fuertes, Andrea Giachero, and Stefano Carrazza. 2023. Qibosoq: an open-source framework for quantum circuit RFSoC programming. *arXiv preprint arXiv:2310.05851* (2023).

[8] Tingjun Chen, Prasanthi Maddala, Panagiotis Skrimponis, Jakub Kolodziejski, Abhishek Adhikari, Hang Hu, Zhihui Gao, Arun Paidimarri, Alberto Valdes-Garcia, Myung Lee, Sundeep Rangan, Gil Zussman, and Ivan Seskar. 2023. Programmable and open-access millimeter-wave radios in the COSMOS Testbed: Design, deployment, and experimentation. *Computer Networks* 234 (2023), 109922.

[9] Wei Cheng, Zhihui Gao, and Tingjun Chen. 2024. SPEAR: Software-defined Python-Enhanced RFSoC for Wideband Radio Applications. In *Proc. ACM WiNTECH'24*.

[10] Zhihui Gao, Zhenzhou Qi, and Tingjun Chen. 2024. Mambas: Maneuvering analog multi-user beamforming using an array of subarrays in mmWave networks. In *Proc. ACM MobiCom'24*.

[11] Jesus O Lacruz, Rafael Ruiz Ortiz, and Joerg Widmer. 2021. A real-time experimentation platform for sub-6 GHz and millimeter-wave MIMO systems. In *Proc. ACM MobiSys'21*.

[12] Alphan Şahin, Mihail L Sichitiu, and İsmail Güvenç. 2023. A millimeter-wave software-defined radio for wireless experimentation. In *Proc. IEEE CNERT'23*.

[13] Marius Šiaučiulis, David Northcote, Josh Goldsmith, Louise H Crockett, and Šarūnas Kaladė. 2023. 100GBit/s RF sample offload for RFSoC using GNU Radio and PYNQ. In *Proc. IEEE NEWCAS'23*.

[14] Leandro Stefanazzi, Kenneth Treptow, Neal Wilcer, Chris Stoughton, Collin Bradford, Sho Uemura, Silvia Zorzetti, Salvatore Montella, Gustavo Cancelo, Sara Sussman, et al. 2022. The QICK (quantum instrumentation control kit): Readout and control for qubits and detectors. *Rev. Sci. Instrum.* 93, 4 (2022).

[15] Kyle A Steiner and Mark B Yeary. 2023. A 1.6-GHz sub-Nyquist-sampled wideband beamformer on an RFSoC. *IEEE Trans. on Radar Syst.* 1 (2023), 308–317.

[16] David Volz, Andreas Koch, and Bastian Bloessl. 2023. Software-defined wireless communication systems for heterogeneous architectures. In *Proc. ACM MobiCom'23*.

[17] Mingran Yang, Zhizhen Zhong, and Manya Ghobadi. 2023. On-fiber photonic computing. In *Proc. ACM HotNets'23*.

[18] Renjie Zhao, Timothy Woodford, Teng Wei, Kun Qian, and Xinyu Zhang. 2020. M-cube: A millimeter-wave massive MIMO software radio. In *Proc. ACM MobiCom'20*.

[19] Renjie Zhao, Timothy Woodford, Teng Wei, Kun Qian, and Xinyu Zhang. 2022. M-cube: an open-source millimeter-wave MIMO software radio for wireless communication and sensing. In *Proc. ACM MobiSys'22 Demo*.

[20] Zhizhen Zhong, Mingran Yang, Jay Lang, Dirk Englund, and Manya Ghobadi. 2023. First demonstration of real-time photonic-electronic DNN acceleration on SmartNICs. In *Proc. ACM SIGCOMM'23 Demo*.

[21] Zhizhen Zhong, Mingran Yang, Jay Lang, Christian Williams, Liam Kronman, Alexander Sludds, Homa Esfahanizadeh, Dirk Englund, and Manya Ghobadi. 2023. Lightning: A reconfigurable photonic-electronic SmartNICs for fast and energy-efficient inference. In *Proc. ACM SIGCOMM'23*.