A Heterogeneous Multi-device System for Synchronous Signal Capture and Processing

Alexander Kroh, Vu Hoang Thang Chau, Duc Dung Vu, and Ediz Cetin
School of Engineering, Macquarie University, Sydney, NSW, 2109, Australia
{alexander.kroh, vuhoangthang.chau, richard.vu, ediz.cetin}@mq.edu.au

***Abstract*—This paper presents a framework for extending synchronous GHz-range analog signal acquisition using multiple heterogeneous Radio Frequency System-on-Chip (RFSoC) platforms. Analog-to-Digital channels are aggregated and streamed to a centralized backbone via 100 Gbps Ethernet. A timestamping mechanism enables accurate signal reconstruction despite inter-device synchronization challenges. Two preprocessing methods, bit truncation and frame buffering, optimize throughput under bandwidth constraints. Experiments using AMD ZCU111 and RealDigital RFSoC 4x2 platforms demonstrate reliable intra-device synchronization and inter-device alignment within 6 ns. The results validate the feasibility of scalable, distributed RF signal capture and provide a foundation for future multi-board data acquisition systems systems.**

***Index Terms*—Heterogeneous systems, Radio frequency system-on-chip, Time synchronization, Data acquisition**

I. Introduction

The increasing complexity of Radio Frequency (RF) Digital Signal Processing (DSP) implementations in applications spanning defense, aerospace, and civil domains, has driven demand for flexible and scalable Digital Acquisition (DAQ) systems capable of handling diverse signal types, bandwidths, and sampling rates [1]–[4].

Field-Programmable Gate Arrays (FPGAs), particularly Radio Frequency System-on-Chip (RFSoC) platforms, offer integrated Analog-to-Digital Converters (ADCs), and high-speed processing capabilities suitable for RF DSP tasks. However, these platforms are typically constrained by limited ADC channel counts, posing challenges for applications requiring large-scale, synchronized data acquisition.

Prior work has explored various approaches to address these limitations. In [2], the authors demonstrated a heterogeneous architecture combining General Purpose Graphical Processors (GPGPUs), FPGA and analog capture devices to support a million-channel Fast Fourier Transformation (FFT) for radio astronomy. Works [5], [6] have shown the utility of AMD RFSoC ZCU111 platforms for prototyping and high-speed acquisition, though their homogeneous hardware configurations limit adaptability. In [7] an AXI C2C-based approach was proposed for multi-board integration, focusing on Dynamic Partial Reconfiguration (DPR) capability of AMD FPGA platforms rather than synchronized data capture.

While synchronization across homogeneous platforms has been investigated [1], [3], the evaluation only considered Digital-to-Analog Converter (DAC) channels. ADC channel synchronization presents additional challenges. Unlike DACs which allow for data flow control in the FPGA, ADCs begin streaming data immediately upon configuration, complicating alignment across devices.

This work explores the feasibility of using heterogeneous RFSoC platforms for distributed RF signal capture. We present a modular framework that enables synchronized acquisition across multiple boards and supports high-throughput data aggregation via 100 Gbps Ethernet link. Our system integrates a timestamping mechanism and preprocessing strategies to facilitate accurate signal reconstruction under bandwidth constraints.

To validate the framework, we employ two different RFSoC platforms – AMD ZCU111 and RealDigital RFSoC 4x2 – sampling RF signals at $1,024$ Msps. The proposed architecture streamlines development from FPGA design to software integration and provides a foundation for scalable, multi-device DAQ systems.

The key contributions of this paper are:

- A framework for aggregating ADC channels across heterogeneous platforms and streaming data to a centralized backbone via high-speed Ethernet.
- Characterization of timing relationships and synchronization challenges in heterogeneous multi-device environments, establishing baseline requirements for future cross-board implementations.

The rest of the paper is organized as follows: Section II describes the system architecture and methodology. Section III presents experimental results and analysis, while concluding remarks are given in Section IV.

II. Proposed system architecture

Our work considers systems comprising multiple heterogeneous front-end platforms, each integrating ADCs, FPGAs, and Central Processing Units (CPUs) connected to a backbone system via high-throughput Ethernet link. The high-level system architecture, illustrated in Fig. 1, features two exemplar front-end platforms – AMD RFSoC ZCU111 and RealDigital RFSoC 4x2 – which provide eight and four ADC channels, respectively. Each platform supports a 100 Gbps Quad Small Form-factor Pluggable (QSFP) Ethernet interface for communication with the backbone, enabling efficient data aggregation and transfer.

Incoming RF signals are digitized by each platform's integrated ADC channels, managed by the `RFDC` block. A timestamping mechanism, implemented by the `Timestamping`

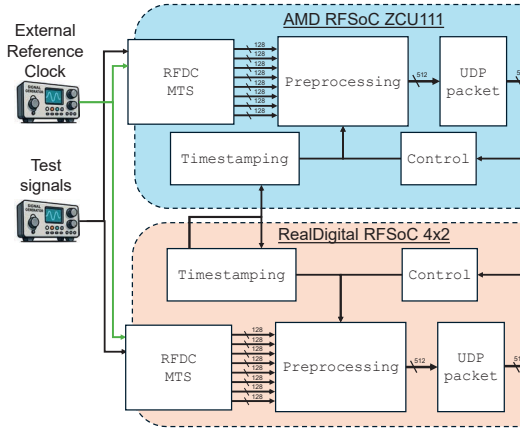

Fig. 1. High-level system architecture (Directional arrows i

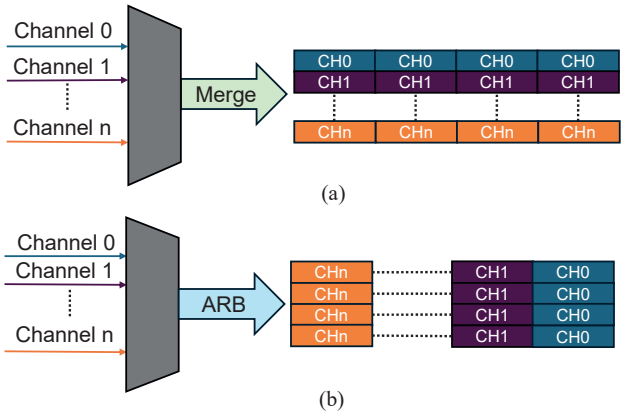

Fig. 2. Aggregation Methodologies. (a) Interleaving: each network packet contains interleaved samples from multiple channels. and (b) Arbitration: Each network packet contains samples from only one channel.

block, synchronizes time across the front-end [illegible] lowing for accurate signal reconstruction at the backbone. The `Preprocessing` module reduces data bit rates and transforms the continuous ADC data streams into frames. These frames are then encapsulated into User Datagram Protocol (UDP) packets by the `UDP Packet` module for transmission over the QSFP link.

The backbone collects the incoming network traffic and reconstructs the streams from each front-end for further processing. This may include Angle of Arrival (AoA) estimation for beamforming, signal encoding for Muiltile Input/Multiple Output (MIMO) systems, or local demodulation for data extraction. While the downstream applications are beyond the scope of this work, the architecture is designed to support such extensions.

The `Control` module allows the backbone to dynamically configure `Preprocessing` module parameters on each front-end platform. Although the initial setup is performed by the local CPU, runtime adjustments can be made centrally to optimize bandwidth usage across front-end systems.

The modularity of the framework overcomes the design challenges arising from the dual requirements of FPGA and software design for system control and data management [8], [9].

Key challenges addressed in this architecture are detailed in the following sub-sections.

A. Signal Channel Aggregation

The ADC channels must be carefully interleaved in order to efficiently utilize a single Ethernet interface. For channels operating at the same sample bit rate, data can be merged such that samples alternate within each network packet, as illustrated in Fig. 2(a). This interleaving approach minimizes the latency as corresponding samples from different channels arrive at the backbone at similar times.

For channels with different bit rates, data is buffered and encapsulated into independent packets per channel. These packets are then arbitrated at the network packet level to achieve interleaving as shown in Fig. 2(b). While this approach supports heterogeneous data rates, it introduces extra latency, as the backbone must receive all samples from $N-1$ channels before observing the first sample from the N^{th} channel.

B. Preprocessing

Preprocessing is essential to ensure that all channels adhere to the throughput limits of the Ethernet interface. Although AMD FPGAs allow the preprocessing method to be reconfigured at any time using DPR [7], this work focuses on two general runtime-switchable approaches that do not require DPR. These are: *Bit Truncation*, where the least significant bits of each sample is truncated to reduce data volume and *Frame Buffering*, which preserves full-resolution samples but transmits them periodically, tolerating temporary sample loss during buffer drain. The latter technique is particularly applicable to periodic signal acquisition tasks, where triggers such as peak detection or preamble identification are employed.

The preprocessing method and its parameters are provided to the backbone via an application-level packet header. The backbone uses this metadata to select and configure the appropriate approach to reconstruct the received signals.

C. Timestamping

For applications such as phased-array beamforming, precise synchronization of ADC channels within each front-end, and across all other front-end platforms is essential to ensure accurate phase alignment and signal integrity. While intra-device synchronization is facilitated by the Multi-Tile Synchronisation (MTS) feature available on RFSoC platforms, inter-device synchronization necessitates additional mechanisms beyond native hardware support.

The AMD ZCU111 and RealDigital 4x2 RFSoC platforms support external reference clock input, which is used to derive the ADC sampling clocks. This configuration enables synchronized sampling across boards. However, these channels lose synchronization when transferred over Ethernet to the backbone system. Consequently, a dedicated mechanism is

required to realign the data streams of each network packet to the relative time instance that they were originally acquired.

We introduce a timestamping mechanism that tags the time at which the first sample in each frame was received from the ADC. Given the known sampling rate, the backbone can calculate the time at which subsequent samples in the frame were received. A synchronization pulse is distributed to the Input/Output (IO) pins of each front-end device. We expect that the propagation delay of this pulse from its origin to the FPGA of all heterogeneous front-end boards to be asymmetric. However, we can compensate for asymmetric delays as long as they are consistent. Once the synchronization signal is received and the appropriate compensation delay is applied, an ADC sample counter begins to count from zero to form our sample timestamp.

The timestamp, along with preprocessing metadata is embedded in each network packet's application-level header. This not only facilitates signal reconstruction but also provides resilience against packet loss, enabling the backbone to detect and realign data streams based on timestamp gaps.

III. Results and Discussion

The primary objective of this evaluation is to assess the system's ability to reconstruct signals acquired from multiple heterogeneous front-end platforms. These signals may undergo different preprocessing methods, resulting in varying bit rates and potential sample loss. While the backbone could be an embedded device or another FPGA-based platform, we utilize a server-based system to facilitate flexible experimentation and performance analysis.

To validate the framework, we employed AMD ZCU111 and RealDigital 4x2 development boards.The ZCU111 features a first-generation RFSoC, while the 4x2 board incorporates a third-generation RFSoC with enhanced RF front-end capabilities. These platforms differ in their clock management architectures, which influence synchronization performance.

Although each RFSoC device supports up to eight ADC and eight DAC channels, the 4x2 board exposes only four ADC channels to the user. For evaluation purposes, we assume full channel availability but focus on a subset sufficient to demonstrate intra- and inter-device synchronization.

A shared external 10 MHz reference clock was used to synchronize ADC sampling rates across both boards. While sample rate matching is not strictly required by the framework, it enables precise timing analysis for signals requiring cross-board alignment.

To evaluate synchronization across all ADC channels and boards, we selected a chirp signal with two key properties: (1) high-frequency content for accurate alignment measurement, and (2) long-period variation to avoid ambiguity due to signal repetition. The chirp signal, generated using a Keysight M8190a Arbitrary Waveform Generator (AWG), sweeps from 16 kHz to 128 MHz over an 8 μs duration. RF splitters were used to distribute the signal and reference clocks to both boards.

ADC channels 1 and 5 of the 4x2 board (corresponding *ADC_A and ADC_C*) and channel 0 on the ZCU111 board (*ADC224_T0_CH0* on the XM500 daughter board) were driven by the chirp signal. This configuration enables evaluation of intra-device synchronization on the 4x2 and inter-device synchronization between the 4x2 and ZCU111.

All ADCs were configured to sample at $1,024$ MHz, resulting in 8 samples per clock cycle streams to the FPGA fabric operating at 128 MHz. If all eight channels were active simultaneously, the aggregate data rate would approach 131 Gbps. Given the 100 Gbps limit of the QSFP interface, preprocessing is required to reduce the data rate. To this end, as detailed in Subsection II-B, we evaluated two preprocessing strategies: *Bit Truncation*, where only the upper 8 bits of each sample is transmitted, reducing bandwidth while maintaining continuous data flow, and *Frame Buffering*, where full 16-bit resolution of each sample is preserved but data is transmitted periodically, introducing controlled sample loss.

Initial results, as shown in Fig. 3, confirmed successful reconstruction of ADC data streams. Intra-device synchronization was achieved with negligible error, while inter-device synchronization exhibited minor discrepancies. Amplitude variations between 4x2 and ZCU111 channels can be explained by the frequency response differences between the 4x2 and the XM500 daughter board of the ZCU111 board. In the following, we further analyze the effectiveness of intra-device signal synchronization, and evaluate the extent of inter-device synchronization error.

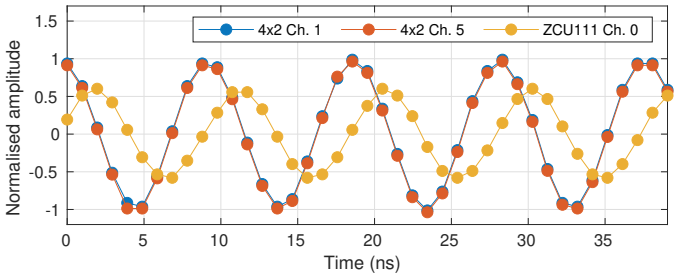

Fig. 3. Inter- and intra-device signal reconstruction.

A. Intra-device Synchronization

To evaluate intra-device synchronization, we analyzed the alignment of chirp signals captured by channels 1 and 5 of the RFSoC 4x2 development board. These channels were selected to span different ADC tiles, thereby testing the effectiveness of the MTS feature of the RFSoC.

We performed $1,000$ iterations of the experiment, reprogramming the FPGA and LMK clock management devices on the 4x2 board before each run. Cross-correlation analysis of the reconstructed signals consistently showed maximum peak at zero sample delay as shown in Fig. 4. Given the chirp signal's 8 μs period, this result implies precise alignment, confirming that the MTS feature reliably synchronizes ADC channels within a single device.

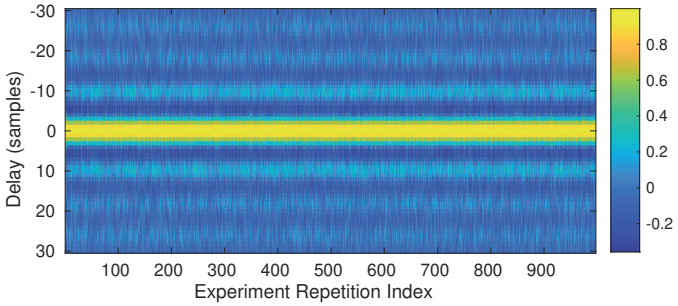

Fig. 4. Heatmap of cross-correlation of 4x2 channel 1 and channel 4 over 32k samples per channel, per iteration over 1000 iterations showing dominant and consistent 0 sample delay.

B. Inter-device Synchronization

To assess cross-device synchronization, we repeated the experiment using channel 1 of the RFSoC 4x2 and channel 0 of the ZCU111. Of $1,000$ iterations, 997 yielded valid timestamp matches, with three discarded due to automation errors.

Cross-correlation results, depicted in Fig. 5, revealed a dominant peak near zero sample delay, but with slight variability across iterations, implying a good synchronization between different boards, but with minor errors. Delay values ranged from -6 to $+3$ samples, as can be observed from Fig. 6, corresponding to a timing error of -6 ns to $+3$ ns. With 8 samples per clock cycle, these discrepancies are attributed to minor mismatches in the $1,024$ Msps ADC sampling clocks, rather than the timestamping mechanism itself.

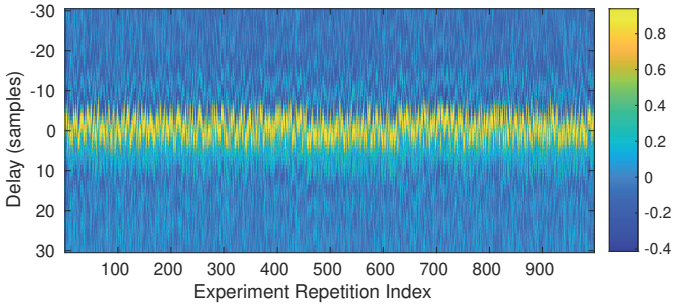

Fig. 5. Heatmap of cross-correlation between 4x2 channel 1 and ZCU111 channel 0 over 32 k samples per channel per experiment with 997 experiment repetitions showing dominant but inconsistent delays near 0.

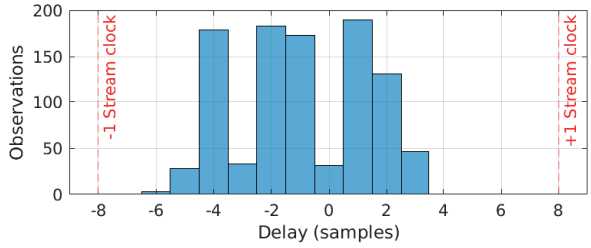

Fig. 6. Histogram of sample delays observed between 4x2 channel 5 and ZCU111 channel 1 across 997 experiment repetitions.

C. Timestamp Delay Balancing

Our timestamp mechanism includes support for delay compensation to account for asymmetric propagation of synchronization pulses through the IO across heterogeneous platforms. In practice, we found that the delay introduced by the 128 MHz data stream clock was sufficiently balanced, enabling accurate timestamp generation without requiring additional calibration. This result demonstrates the robustness of the timestamping approach, even in the presence of hardware-level asymmetries.

D. Real-time Preprocessing Mode Switching

We further evaluated the system's ability to switch preprocessing modes during runtime. An 8 MHz sinusoidal signal was applied to channel 4 of the ZCU111 board, with the signal amplitude set to 3.5% of the ADC's full-scale range to highlight quantization effects.

Initially, the system operated in *bit truncation* mode, transmitting only the upper 8 bits of each sample. A rapid transition to *frame buffering* mode was triggered via two consecutive commands issued from the ZCU111 CPU. The buffering mode preserved full 16-bit sample resolution but introduced periodic sample loss due to bandwidth constraints.

Results depicted in Fig. 7 show that we were able to transition between modes without sample loss, but samples were indeed lost as required by the frame buffering mode. The timestamp embedded in each packet enabled accurate reconstruction of the signal following the switch. This capability is particularly useful for capturing periodic signals with high resolution while maintaining efficient network utilization.

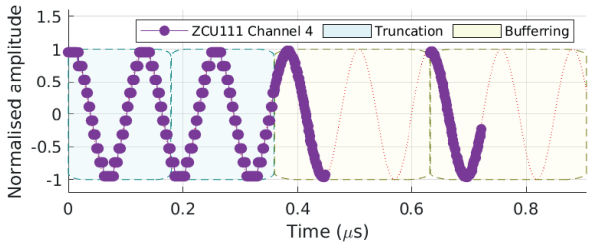

Fig. 7. Real-time preprocessing mode switching from bit truncation to frame buffering.

IV. Conclusion

This study presented a scalable framework for extending ADC capabilities using multiple heterogeneous RFSoC platforms. Through timestamping and preprocessing, the system achieves accurate signal reconstruction and synchronization across devices. Intra-device synchronization is reliably maintained, while inter-device synchronization errors remain within 6 ns. These findings support the feasibility of distributed RF signal acquisition for applications requiring high temporal precision. Future work will explore integration with real-time signal processing pipelines.

REFERENCES

[1] O. Chu, "The Future of Radar Technology–Integrating RFSoC with Reconfigurable Computing for Reconfigurable Radar Systems," in *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2024, pp. 609–615.

[2] G. Grutzeck, I. Krämer, M. Ciechanowicz, N. Reyes, C. König, A. Baryshev, S. Yates, and B. Klein, "A MKID-readout based on a heterogeneous, closely coupled architecture," *32nd IEEE International Symposium on Space THz Technology (ISST 2022)*, 2022.

[3] C. Karle, M. Neu, B. Nuss, L. Witte, A. Scheder, E. Waldner, E. Shkurtaj, T. Harbaum, and J. Becker, "Scalable Multi-Level Synchronization Technique of Distributed Multi-RFSoC-Server Systems for 6G," in *2024 IEEE 37th International System-on-Chip Conference (SOCC)*, 2024, pp. 1–6.

[4] A. Coskun, S. D'Addio, R. Jansen, D. Levacq, and P. Angeletti, "112-Gbps High Speed Serial Links to Interconnect Direct RF Sampling ADCs and Digital High Throughput Satellite Processors," in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2024, pp. 431–434.

[5] K. K. Guner, T. O. Gulum, and B. Erkmen, "FPGA-based Wigner–Hough transform system for detection and parameter extraction of LPI radar LFMCW signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[6] R. Toledo-Moreo, C. Pérez-Lemus, D. Pérez-Lizán, F. J. Toledo-Moreo, and I. Villó-Pérez, "An Acquisition System Based on an RFSoC Board: Demonstration for a Ground-Based Astronomical Experiment," *IEEE Instrumentation & Measurement Magazine*, vol. 28, no. 4, pp. 60–68, 2025.

[7] M. Fuchs, H. Krause, T. Muscheid, L. Scheller, L. E. Ardila-Perez, and O. Sander, "Cross-Chip Partial Reconfiguration for the Initialisation of Modular and Scalable Heterogeneous Systems," *IEEE Transactions on Nuclear Science*, vol. 72, no. 3, pp. 727–734, 2024.

[8] P. Coussy, A. Baganne, and E. Martin, "A design methodology for integrating IP into SOC systems," in *Proceedings of the IEEE 2002 Custom Integrated Circuits Conference (Cat. No. 02CH37285)*, 2002, pp. 307–310.

[9] D. D. Vu, S. Biswas, A. Kan, and E. Cetin, "A systematic methodology for time-multiplexing algorithms on a reconfigurable system-on-chip," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025, pp. 1–5.