

Introduction to the ARC 2025 Journal Track

This editorial introduces the special section on ARC 2025 Journal Track, comprising extended versions of five selected articles from the 2025 International Symposium on Applied Reconfigurable Computing (ARC), together with a perspective article commemorating the first 20 years of the symposium. The selected contributions illustrate the breadth of contemporary reconfigurable computing research and address key challenges in compilation, dependable acceleration, low-latency machine learning, on-chip communication, and event-driven edge intelligence. Specifically, they investigate control-flow-aware compilation for coarse-grained reconfigurable arrays; rapid detection and recovery from radiation-induced faults in FPGA-based AI accelerators; streamlined dataflow architectures for echo state networks; throughput-adaptive networks-on-chip for AI accelerators; and hardware-accelerated graph neural networks for event-based audio processing. The anniversary article complements these works by reviewing representative research themes developed across 2 decades of ARC and identifying emerging directions for the field. Collectively, the articles demonstrate the continuing importance of cross-layer approaches that combine algorithms, compilation, architectures, runtime mechanisms, and application-specific optimization to improve the performance, energy efficiency, programmability, and dependability of reconfigurable systems.

1 Introduction

The International Symposium on **Applied Reconfigurable Computing (ARC)** has long provided a forum for advancing reconfigurable architectures, design methods, systems, and applications. This special section presents extended versions of 5 articles selected from ARC 2025, held in Seville, Spain, together with a perspective article marking the first 20 years of the symposium. The five research articles were invited based on the quality and relevance of their ARC 2025 conference versions. Each of these submissions was substantially extended and underwent a separate journal review process in accordance with ACM TRETs standards. The five research articles address complementary bottlenecks that increasingly define the field: compilation for **Coarse-Grained Reconfigurable Arrays (CGRAs)** in the presence of complex control flow; maintaining the dependability of FPGA-based AI accelerators under radiation-induced faults; meeting stringent latency and energy requirements for time-series inference; preventing on-chip communication from limiting the performance of AI accelerators; and exploiting sparse event streams for near-sensor audio intelligence. The 20th-anniversary perspective is included to place these contributions within the longer trajectory of ARC and to show how today's challenges extend recurring themes of programmability, efficiency, adaptability, reliability, and application-driven architecture.

Taken together, these articles illustrate how reconfigurable computing has evolved beyond the optimization of individual computational kernels. Achieving efficient and dependable systems now requires coordinated advances across the entire stack: compiler infrastructures must expose and exploit parallelism; architectures and interconnects must adapt to changing workload characteristics;

ACM Reference format:

Roberto Giorgi and Mirjana Stojilovic. 2026. Introduction to the ARC 2025 Journal Track. *ACM Trans. Reconfig. Technol. Syst.* 19, 3, Article 39 (September 2026), 4 pages.

<https://doi.org/10.1145/3839105>



This work is licensed under Creative Commons Attribution International 4.0.

© 2026 Copyright held by the owner/author(s).

ACM 1936-7414/2026/9-ART39

<https://doi.org/10.1145/3839105>

runtime mechanisms must detect and recover from faults; and algorithms must be reformulated to take advantage of spatial execution, quantization, sparsity, and heterogeneous integration. The articles in this special section provide representative examples of these directions.

2 Overview of the Selected Articles

The special section opens with the 20th-anniversary article, “*A Perspective on the Topics from a Selection of Papers from the First Twenty Years of the International Symposium on Applied Reconfigurable Computing (ARC)*,” by João MP Cardoso and Walid Najjar. The article examines representative work published during ARC’s first 20 editions and relates it to more recent developments in the field. Its selection process combines citation impact, best-paper recognition, reviewer assessments, and nominations from members of the ARC community, resulting in a set of 26 articles, approximately 3.7% of the articles presented during the period considered. The selected contributions cover arithmetic acceleration, logic and high-level synthesis, CGRAs, instruction-set extensions, FPGA security, dependability, dynamic partial reconfiguration, runtime systems, and deep-learning acceleration. More than a retrospective, the article connects these contributions to current challenges and future research opportunities, including open and reproducible design infrastructures, domain-specific CGRAs, emerging AI workloads, and the need for better integration between architectures, tools, and applications.

The five extended ARC 2025 articles are introduced below in no particular order.

In “*An MLIR-Based Compilation Framework for Control Flow Management on Coarse Grained Reconfigurable Arrays*,” Yuxuan Wang, Cristian Tirelli, Giovanni Ansaloni, Laura Pozzi, and David Atienza address one of the principal barriers to the broader adoption of CGRAs: the difficulty of compiling complete applications rather than isolated dataflow kernels. Most existing CGRA compilation techniques focus on individual loops or dataflow graphs and either transfer control-flow management to a host processor or depend on dedicated reconfiguration hardware. These approaches can introduce substantial communication and reconfiguration overhead and restrict the range of applications that can be efficiently accelerated.

The authors present an end-to-end framework built on multi-level intermediate representation that represents both control and dataflow through a dedicated CGRA dialect. The framework combines control-flow-graph transformations, liveness analysis, modulo scheduling, and an integer linear programming-based mapping method. It supports arbitrary control flows on **Program Counter (PC)** controlled, homogeneous CGRAs without requiring specialized hardware and eliminates the need for reconfiguration during transitions between basic blocks. Across the evaluated benchmarks, the proposed framework achieves an average $2.12\times$ speedup over the closest PC-based control/dataflow graph mapper and also improves on processor-assisted and reconfiguration-based alternatives. This work illustrates how modern compiler infrastructures can broaden the applicability of spatial architectures without sacrificing their execution efficiency.

The reliability of reconfigurable AI systems is the subject of “*Fast SEU Detection and Recovery in FPGA-Based AI Accelerators*,” by Giorgio Cora, Eleonora Vacca, Corrado De Sio, Sarah Azimi, and Luca Sterpone. FPGAs offer the performance and adaptability required by AI acceleration in avionics and space systems, but their configuration memory is vulnerable to radiation-induced **single-event upsets (SEUs)** that can corrupt the implemented logic. Conventional mitigation approaches may impose substantial hardware overhead, incur high detection latency, or require full-device reconfiguration that interrupts execution and discards completed work. The authors address these limitations with **Reconfigurable Platform for AI Resilience (RePAIR)**, a heterogeneous platform that combines runtime fault detection, dynamic partial reconfiguration of the affected accelerator region, and execution recovery for FPGA-based AI inference.

RePAIR integrates an open-source RISC-V processor with a Tensor Processing Unit-like systolic-array accelerator and extends the accelerator instruction-set architecture with checksum-based runtime self-test. The testing mode reuses existing computational paths to check individual matrix multiplications, incurring only three additional clock cycles per multiplication and limited area overhead. Upon detecting a structural fault, the processor initiates dynamic partial reconfiguration and resumes execution from the last operation that completed successfully. Across two FPGA platforms, RePAIR reduces system downtime by up to three orders of magnitude compared with full-device reconfiguration, while limiting worst-case inference overhead to 30%. Fault-injection campaigns combined with space-environment modeling further assess the platform under representative orbital conditions. Overall, the article presents a practical approach to improving the availability of FPGA-based AI accelerators through application-aware SEU detection and localized recovery.

In “*Streamlined Dataflow FPGA Architectures for Echo State Neural Networks in Time-Series Prediction and Classification Applications*,” Atousa Jafari and Marco Platzner investigate high-frequency time-series workloads whose sampling rates and response-time requirements are difficult to satisfy using conventional embedded processors. **Echo state networks (ESNs)**, a form of reservoir computing, offer an appealing alternative to conventional recurrent neural networks because only their output layer must be trained. Nevertheless, efficiently translating these models into hardware requires careful treatment of quantization, nonlinear functions, resource allocation, and model generation.

The authors propose a fully unrolled, streamlined dataflow architecture in which all ESN layers are quantized and directly mapped to FPGA resources. Two architectural variants allow neuron computations to be mapped either to digital signal processing blocks or exclusively to **Look-Up Table (LUT)** logic. The accompanying tool flow performs model configuration, training, optimization, quantization, and automatic generation of the FPGA design. For a 200-neuron reservoir, the accelerator achieves 9.5 ns latency and 100 million samples per second throughput. In the evaluated classification tasks, 8-bit quantization generally keeps accuracy within approximately 1–3 percentage points of the floating-point reference, while the resulting hardware delivers orders-of-magnitude improvements in latency and throughput over embedded CPU and graphics processing unit implementations. The article offers useful insights into the benefits and resource costs of aggressively spatial architectures for applications where deterministic, extremely low-latency performance is more important than general-purpose flexibility.

Communication efficiency is addressed by Mohamed Amine Zhiri, Hana Krichene, Chiara Sandionigi, and Sébastien Pillement in “*HT-NoC: A Throughput-Adaptive Multi-Dataflow Network-on-Chip for AI Accelerators*.” The authors propose HT-NoC, a dynamically reconfigurable **network-on-chip (NoC)** that increases communication throughput by reallocating underutilized router resources to active data transfers. Its support for multiple dataflows accommodates fully connected, convolutional, depthwise, and pointwise layers, providing flexibility across diverse **Deep Neural Network (DNN)** layer types.

HT-NoC dynamically switches between a conventional operating mode and full- or partial-high-throughput modes tailored for persistent, bandwidth-intensive traffic. A synthesized 12×12 implementation incurs a 12% LUT overhead over a baseline mesh NoC. For fully connected layers, HT-NoC accelerates weight propagation by $4\times$ and reduces its dynamic energy by $2.6\times$ on average. When integrated into an AI dataflow accelerator, it achieves up to a $3.2\times$ speedup for GPT-2 feed-forward network blocks and compares favorably with specialized systolic-array-based Transformer accelerators. The article demonstrates how lightweight reconfiguration can improve communication efficiency while retaining support for a broad range of DNN layer types.

The final article, “*Hardware-Accelerated Graph Neural Networks: An Alternative Approach for Event-Based Audio Classification and Keyword Spotting on SoC FPGA*,” by Kamil Jeziorek, Piotr Wzorek, Krzysztof Blachut, Hiroshi Nakano, Manon Dampfhofer, Thomas Mesquida, Hiroaki Nishi, Thomas Dalgaty, and Tomasz Kryjak, explores the combination of neuromorphic sensing, graph neural networks, and embedded reconfigurable hardware. Event-based audio sensors produce sparse asynchronous streams rather than uniformly sampled frames. Although this representation can substantially reduce unnecessary computation, exploiting it requires architectures capable of processing events continuously and managing the irregular memory accesses associated with graph construction and convolution.

The proposed system uses an artificial cochlea to transform audio signals into events and processes them through a hardware-aware event-graph neural network implemented on a system-on-chip FPGA.

A simplified graph-generation procedure, termed *skip-step connection*, limits neighbor searches and replaces computationally expensive graph features with hardware-efficient alternatives. On the Spiking Heidelberg Digits benchmark, the design improves classification accuracy over earlier FPGA implementations of hardware-aware spiking neural networks by 4.5 and 19.3 percentage points. The extended journal version also broadens the evaluation toward keyword spotting and continuous event-stream processing, strengthening the connection between event-based audio sensing and practical near-sensor inference. Its smallest configuration obtains competitive accuracy while using only 23% of the logic and 37% of the memory required by a previous hardware state of the art. By demonstrating continuous event-driven inference with competitive accuracy, low latency, and reduced resource requirements, the article points toward practical near-sensor intelligence that preserves the sparsity of the original signal throughout the processing pipeline.

Taken as a whole, the articles in this special section reflect both the breadth and the continuing coherence of ARC. Although they operate at different abstraction levels, they share a common objective: adapting the computing system more closely to the workload’s structure, dynamics, and dependability requirements.

Acknowledgments

The guest editors thank the authors for preparing these extended articles and for their careful responses during the review process. We are equally grateful to the reviewers, whose detailed assessments and constructive recommendations helped strengthen the technical content and presentation of the articles. We also sincerely thank the ACM TRETs editorial team, especially Vaughn Betz (Editor-in-Chief) and Clarissa Nemeth (Journal Administrator), for their guidance and assistance throughout the journal track process. We hope that readers will find this collection informative and inspiring, and that it will encourage further research at the intersection of reconfigurable architectures, compilation, dependable systems, and energy-efficient computing.

Roberto Giorgi

Department of Information Engineering and Mathematics, University of Siena, Siena, Italy

Mirjana Stojilovic

School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland