POLITECNICO DI TORINO
Repository ISTITUZIONALE

POSTER: A Reliable Multi-FPGA RISC-V Based Cluster for Space AI Inference

Original
POSTER: A Reliable Multi-FPGA RISC-V Based Cluster for Space AI Inference / Cora, G., Duni, M., De Sio, C., Azimi, S., Sterpone, L.. - (2026), pp. 331-332. (23rd ACM International Conference on Computing Frontiers, CF 2026 Catania (ITA) May 19 - 21, 2026) [10.1145/3801487.3805604].

Availability:
This version is available at: 11583/3013172 since: 2026-07-16T09:37:51Z

Publisher:
Association for Computing Machinery

Published
DOI:10.1145/3801487.3805604

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

13 September 2026

Poster: A Reliable Multi-FPGA RISC-V Based Cluster for Space AI Inference

Giorgio Cora
Politecnico di Torino
Turin, Italy
giorgio.cora@polito.it

Morgana Duni
Politecnico di Torino
Turin, Italy
morgana.duni@polito.it

Corrado De Sio
Politecnico di Torino
Turin, Italy
corrado.desio@polito.it

Sarah Azimi
Politecnico di Torino
Turin, Italy
sarah.azimi@polito.it

Luca Sterpone
Politecnico di Torino
Turin, Italy
luca.sterpone@polito.it

Abstract

The growing interest in space oriented AI inference is rapidly increasing the demand for computing platforms that are both high-performance and reliable. In this context, SRAM-based FPGAs are widely adopted thanks to their flexibility, reprogrammability, and the ability to integrate heterogeneous computing modules within a single device. In this work, we propose a versatile FPGA-based clustered architecture composed of multiple interconnected computing nodes. Each node integrates a lightweight RISC-V processor responsible for inter-board management and communication, running FreeRTOS, and a TPU-like accelerator to enable efficient on-board AI inference. The nodes are interconnected through high-speed optical links using the Aurora protocol, enabling a fully connected cluster that can distribute inference workloads across the system while supporting reliability-oriented recovery via partial reconfiguration.

CCS Concepts

• **Computer systems organization → Reliability**.

Keywords

FPGA, Cluster, Reliability

ACM Reference Format:
Giorgio Cora, Morgana Duni, Corrado De Sio, Sarah Azimi, and Luca Sterpone. 2026. Poster: A Reliable Multi-FPGA RISC-V Based Cluster for Space AI Inference. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF '26), May 19–21, 2026, Catania, Italy*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3801487.3805604

1 Introduction

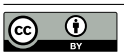
In recent years, the growing adoption of AI inference in space applications has driven the need for computing architectures able to deliver high performance under strict power and reliability constraints. Among the available solutions, Commercial Off-The-Shelf (COTS) SRAM-based FPGAs are widely used to accelerate compute-intensive workloads, including deep neural network (DNN) inference[3], thanks to their versatility, reconfigurability, and support for heterogeneous processing architectures. FPGA-based clusters[2] can further improve performance by distributing workloads across multiple interconnected nodes, enabling higher throughput and scalable inference architectures. However, the use of SRAM-based FPGAs in radiation environments raises significant reliability concerns, since radiation particles may affect the configuration memory and induce Single Event Effects (SEEs). Among them, Single Event Upsets (SEUs)[4] are particularly critical, as they can alter the implemented functionality and cause erroneous behavior.

In this paper, we present a four-node FPGA-based cluster. Each node adopts a processor–coprocessor architecture, combining a lightweight RISC-V core for communication and system coordination with a systolic-array accelerator for efficient DNN inference. The nodes are connected in a fully connected topology through Aurora links over optical fibers. The architecture also supports fast recovery through reconfiguration: when a processor or accelerator fails, the faulty node can be restored while the remaining nodes continue operating, reducing downtime and improving overall availability.

2 Proposed Architecture

The proposed clustered architecture is shown in Fig. 1. The system consists of four homogeneous nodes. Inter-node communication is implemented through a lightweight Aurora protocol over optical fiber channels. As a result, each node integrates three high-speed optical links, enabling a fully connected four-node cluster where any node can directly exchange data with the others. Each node embeds the same processor-coprocessor architecture, built around two main computing engines. First, a lightweight RISC-V processor, the NEORV32[6], that act as main controlling unit. To enhance robustness against radiation-induced faults, the NEORV32 subsystem is implemented with Triple Modular Redundancy (TMR), improving tolerance to transient faults affecting logic and state. The NEORV32 implements an AXI4-Lite interface to control the accelerator and an AXI-Stream interface to transmit and receive data through the Aurora links, enabling efficient inter-node exchanges.

The NEORV32 runs a lightweight version of FreeRTOS. The operating system is configured to execute dedicated tasks, including: an Aurora communication task responsible for data exchange with the other nodes, an accelerator management task handling command and data transfers to/from the TPU, and a synchronization task used to maintain alignment and coordination across the cluster. In case synchronization fails, a remote reconfiguration of the processor is requested through a dedicated unit, communicating with the other nodes over a classical communication channel. The



This work is licensed under a Creative Commons Attribution 4.0 International License.
CF '26, Catania, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2568-5/2026/05
https://doi.org/10.1145/3801487.3805604

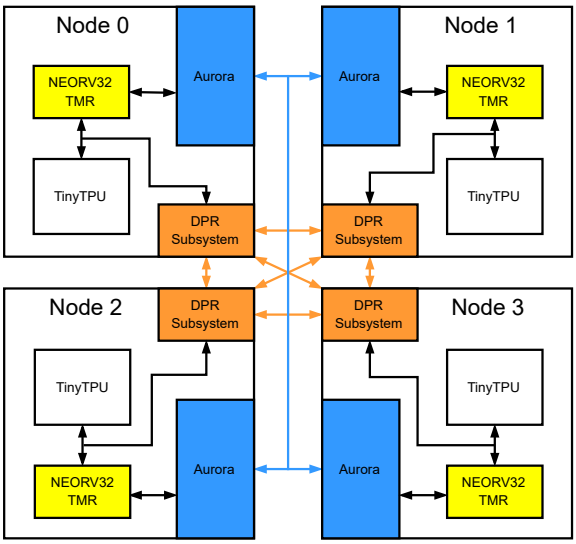


Figure 1: Overview of the proposed multi-node platform

coprocessor is a TPU-like accelerator, namely tinyTPU[5], implemented in VHDL and designed for DNN inference. The size of the systolic array (SA) can be configured from a minimum of 6×6 MAC units up to 14×14, while minor hardware modifications have to be implemented to support bigger SA matrices. It executes a custom ISA with 80-bit instruction-level parallelism, enabling compact and efficient command encoding for inference workloads.

Reliability is addressed using a combination of redundancy, lightweight runtime checking, and recovery through Dynamic Partial Reconfiguration (DPR)[1]. In particular, DPR is employed to repair the RISC-V subsystem when communication with the other nodes is disrupted or when the node fails to provide correct control-level behavior. In addition, when the TPU is idle, a small predefined test sequence is executed to verify the correctness of the accelerator datapath. If the produced signature differs from the expected one, the TPU region is also restored through partial reconfiguration, minimizing downtime while preserving the operation of the rest of the cluster. All the logic managing partial reconfiguration, either local or remote, is embedded in the DPR subsystem.

3 Experimental Results

The proposed platform has been implemented on four AMD ZCU102 development boards. Table 1 reports the resource utilization for a single node. Overall, the resource utilization remains limited, leaving space for integrating additional accelerators or increasing the compute capability of each node. In particular, the low resource utilization of the RISC-V and the communication subsystem simplifies adapting the node to alternative AI accelerators with minor design changes. Moreover, the cluster can be re-targeted to different workloads, potentially also at run time, by leveraging partial reconfiguration.

The design operates at 150 MHz, currently limited by the TPU datapath. System performance is evaluated in terms of downtime and reconfiguration capabilities. Fig. 2 reports the measured reconfiguration time.

As expected, the TPU reconfiguration time scales linearly with the SA size. It is important to note that the reported reconfiguration

Table 1: Resource utilization of a single node on ZCU102.

Platform Modules	LUTs	FFs	BRAMs	DSP
TMR NEORV32	6267	5832	21	0
Aurora Link	1958	4814	3	0
TPU (tinyTPU)	3960	7012	83	210
DPR Logic	1185	989	0	0
Glue Logic	2477	4432	3	0
Total [%]	5.8%	4.2%	12.1%	8.3%

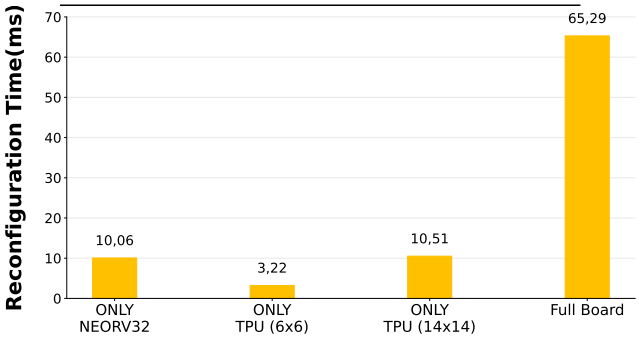


Figure 2: System's reconfiguration times

times account only for configuration-memory programming, assuming the bitstream is uploaded in the DDR memory. In practice, however, storing a full bitstream in DDR is often impractical due to its size, usually much bigger if compared to partial bitstreams. For this reason, full bitstreams are commonly stored in non-volatile memories, such as external flash or SD cards. This approach increases the effective reconfiguration latency due to interface bandwidth limitations, thereby amplifying the benefits of the proposed DPR-based approach in real deployments compared to full-board reconfiguration.

4 Conclusions and future works

The proposed system implements a reliable multi-FPGA architecture, enabling scalable AI workload distribution across multiple computing nodes. Reliability is enhanced through a combination of TMR on the control processor and recovery via dynamic partial reconfiguration, reducing overall downtime. Future works include improvement of the TPU fault detection mechanisms and overall improvement to the error correction procedure, as well as implementation of workload remapping features.

References

[1] AMD, Vivado Design Suite User Guide: Dynamic Function eXchange (UG909) 2025. https://docs.amd.com/r/en-US/ug909-vivado-partial-reconfiguration
[2] Ludovica Bozzoli et al. 2023. EuFRATE: European FPGA Radiation-hardened Architecture for Telecommunications. (2023). doi:10.23919/DATE56975.2023.10137035
[3] Giorgio Cora et al. 2025. RePAIR: Reconfigurable Platform for AI Resilience Within RISC-V Ecosystem. doi:10.1007/978-3-031-87995-1_5
[4] Kyle W. Gear et al. 2022. An analysis of FPGA configuration memory SEU accumulation and a preventative scrubbing technique. Microprocessors and Microsystems 90 (2022), 104467. doi:10.1016/j.micpro.2022.104467
[5] Fuhrmann, J.: Implementierung einer Tensor Processing Unit mit dem Fokus auf Embedded Systems und das Internet of Things Germany (2018). http://hdl.handle.net/20.500.12738/8527
[6] Nolting, S., et al.: The NEORV32 RISC-V Processor. Zenodo 2023. doi:10.5281/zenodo.8260609