Reconfigurable Processor-Centric Accelerators for Safety-Critical Applications

Luis Waucquez*and Alfonso Rodríguez

Centro de Electrónica Industrial, Universidad Politécnica de Madrid (UPM), Madrid, Spain

Abstract

The growing complexity of modern electronic systems operating in harsh environments with strict safety requirements necessitates robust mechanisms to ensure reliability and fault tolerance. These systems must function efficiently under challenging conditions, addressing issues such as single-event upsets and common mode failures in applications including IoT, aerospace, and the automotive industry. This paper introduces a versatile processor-centric accelerator platform designed for safety-critical applications. Built on the RISC-V ISA architecture, the platform's cores support four operational modes: Single, Triple-Core LockStep (TCLS), Dual-Core LockStep (DCLS), and DCLS with classic staggering. To evaluate the platform's performance and fault tolerance, a small fault injection campaign was conducted, demonstrating its capability to maintain reliable operation while delivering the necessary computational power.

Introduction

System on Chip (SoC) complexity continues to grow due to advancements in semiconductor manufacturing, which allow for higher levels of integration. However, as technology nodes shrink, the susceptibility to soft errors caused by radiation and interference increases. These errors can lead to critical failures, resulting in economic losses or even endangering lives. This issue is particularly relevant for devices operating in harsh environments, such as space, where strict design and fabrication rules are necessary to ensure reliability. However, space is not the only domain affected—industries like automotive and aerospace also require compliance with stringent safety standards, such as ISO 26262 and DO-254.

To improve reliability and robustness, different approaches can be adopted. One method is to harden the technology node by using advanced semiconductor processes that provide intrinsic resistance to radiation. Another approach is to introduce architectural modifications, such as implementing redundancy techniques at the hardware level, including Double Modular Redundancy (DMR), Triple Modular Redundancy (TMR), or N-way Modular Redundancy (NMR). Additionally, Error Correction Code (ECC) can be applied to memory components to maintain data integrity. A third strategy focuses on system-level modifications, incorporating redundancy at the module level rather than at the gate level, which avoids altering the original microarchitecture.

*This work has been supported by the Secretaría de Estado de Telecomunicaciones e Infraestructuras Digitales, Ministerio para la Transformación Digital y de la Función Pública, funded by NextGenerationEU, in the context of the Programme for the creation of Chip University-Business Chairs, Project TSI 069100-2023-0016, Chip Chair UPM-INDRA (CAUPIME)

While technology and architectural modifications offer greater reliability, they also introduce significant overhead in terms of performance, power consumption, and chip area. In contrast, system-level approaches provide a more cost-effective solution with lower impact on performance, though they are less precise in detecting and recovering from errors. This trade-off makes system-level techniques particularly appealing for emerging markets such as the Internet of Things (IoT) and the NEW SPACE paradigm, where Commercial Off-The-Shelf (COTS) components are increasingly used in lower orbits.

The proposed reconfigurable processor-based accelerator has been incorporated into X-HEEP [1] Figure 1, an open-source SoC platform developed for low-power applications using a RISC-V processor. Experimental results show that the architecture and mechanisms provide fault tolerance and recovery features, effectively handling single-bit errors and common-mode failures, as confirmed through error injection during simulation.

Reconfigurable Processor-Centric Safe Accelerator

The platform offers a processor-centric computing offloading system for safety-critical applications, designed as a memory-mapped accelerator with run-time reconfigurable mechanisms for adaptive fault-tolerant execution. It is built on open-source IPs from the PULP and X-HEEP ecosystems and enhanced with non-intrusive hardware-based configuration and control modules. A block diagram illustrates the overall system Figure 2.

The CPU System has three CV32E20 [2], with flexi-

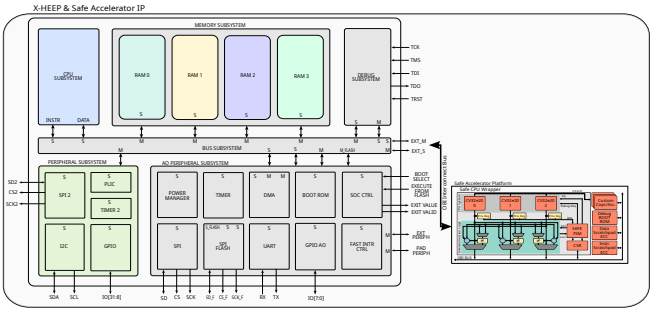

Figure 1: *X-HEEP [1] & Safe Accelerator IP*

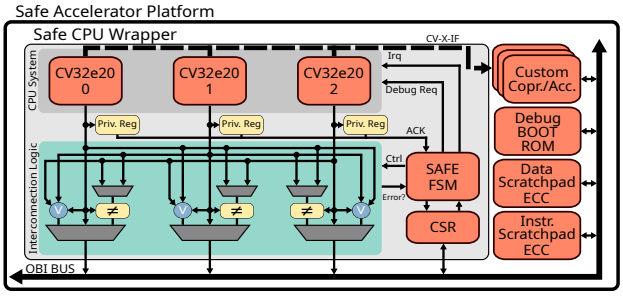

Figure 2: *Safe Accelerator Platform*

bility for different processors. The Safe CPU Wrapper adds fault-tolerant features, while the System Bus uses OBI [3] for parallel memory access. It includes two 32 KiB scratchpad memories and a debug/boot ROM. Accelerators offer enhanced functionalities by working alongside the CPUs, either in a memory-mapped configuration or as tightly coupled coprocessors for instruction extensions, and are also safeguarded by fault-tolerant mechanisms.

Safe CPU Wrapper

The *Safe CPU Wrapper* is the central unit, that enables cores to operate in both safe and non-safe modes through split-lock capabilities. It consists of several key components: the Safe CSR Unit and Private Core Registers, which manage the configuration settings and synchronization mechanisms between the wrapper and the cores; the Safe Interconnection Logic, which connects the cores to the system bus and allows dynamic switching between operation modes, including isolating cores and injecting specific instructions; and the Safe FSM Unit, a hierarchical control system responsible for managing the different configurations of the Safe Interconnection Logic, monitoring errors, and coordinating the cores during operations. The FSM unit activates different configurations based on control registers, with separate state machines for each core to facilitate synchronization and recovery tasks.

Synchronization & Fault Recovery Mechanism

The accelerator platform supports four operational modes: Single, Triple-Core LockStep (TCLS), Dual-Core LockStep (DCLS), and DCLS with classic staggering, which can be configured after reset or during execution, with an initial *"sleep"* state for all cores. The start sequence involves the Safe FSM triggering a debug request to the cores, which then load their program counter from the Debug/boot ROM and begin execution. Synchronization across cores is achieved by storing the master core's context, then loading it into the shadow cores via a synchronization routine in the Debug/boot ROM. In DCLS recovery, when an error is detected by the comparator, the Safe FSM isolates the faulty cores, injects a *wfi* instruction, and prevents error propagation by isolating the data bus and emulating a handshake sequence. The FSM then triggers a recovery routine that restores the checkpoint state to the cores. In TCLS recovery, the TMR configuration prevents the need for isolation, and when an error is detected by the voter, an interrupt request is triggered. The cores store and load their context via the voter.

Conclusion & Futures Lines

A dynamically reconfigurable platform that can switch between non-safety and safety modes while appearing as a single core to the user is introduced. It allows for cores to be turned on or off during different operation phases, providing flexibility for both critical tasks and low-power operations. The platform supports four redundancy modes: Single, TCLS, DCLS, and DCLS with temporal redundancy, which can be selected during runtime or configuration. Fault recovery strategies are primarily software-based to avoid impacting the core's interfaces and architecture, making it adaptable to other cores with the debug extension. Fault injection tests show the platform's ability to recover from single-bit errors and common-mode errors using a staggered solution. Future development plans include conducting more robust fault injection tests on simulations and FPGA, implementing ECC for scratchpad memories, and evaluating redundancy for accelerators or coprocessors with instruction extension interfaces.

References

[1] Simone Machetti et al. *X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller for the Exploration of Ultra-Low-Power Edge Accelerators*. 2024. arXiv: 2401.05548 [cs.AR].

[2] OpenHW Group. *cve2 Repository - CV32E20 Core Design*. 2024. URL: https://github.com/openhwgroup/cve2.

[3] OpenHW Group. *OBI User Guide*. Version 1.6.0. 2022. URL: https://github.com/openhwgroup/obi/blob/072d9173c1f2d79471d6f2a10eae59ee387d4c6f/OBI-v1.6.0.pdf.