Enhancing Safety with RISC-V-based SPIDER Autonomous Robot: A Use-Case from the ECSEL FRACTAL Project

Joaquim Castella Triginer[1], Markus Postl[1], Mikel Fernandez[2], Feng Chang[2], Sergi Alcaide[2], Ramon Canal[2,3], Jaume Abella[2]

[1]Virtual Vehicle Research GmbH, Austria
[2]Barcelona Supercomputing Center (BSC), Spain
[3]Universitat Politècnica de Catalunya (UPC), Spain

Abstract

The SPIDER use case of the ECSEL FRACTAL project demonstrates the safety capabilities developed in the project on a RISC-V platform. The use case demonstrates the integration of a diverse redundancy service and a multicore interference monitoring service. These services allow the SPIDER co-execution of safety-relevant and machine learning tasks, while implementing fail-operational capabilities on a single computing device.

Introduction

The automotive market is demanding for machine learning (ML) techniques for computer vision and decision-making. Such techniques are implemented in features used for developing advanced driver-assistance systems and autonomous driving. CUDA systems or multi-core CPUs are used to accelerate ML tasks and other computationally intensive algorithms. However, it is difficult to find approaches capable of executing safety-critical tasks and compliance with safety automotive standards.

This article shortly describes the ECSEL FRACTAL project and outlines its main scope. Then, it briefly introduces the *SPIDER autonomous robot* use case. Finally, it describes the safety services developed specifically for the project and their integration to the SPIDER.

FRACTAL Project Description

The FRACTAL [1] project with the title *'A Cognitive Fractal and Secure EDGE-based on a unique Open-Safe-Reliable-Low Power Hardware Platform Node'* aims to create a reliable computing node to realize a Cognitive Edge under industrial standards. It scales from low-performance to high-performance edge nodes to become a building block for the Internet of Things. The nodes will have capabilities to adapt according its surrounding and improve their behaviour in terms of performance, safety, security, power, or efficiency [2].

The strategic objectives of the project are:

1. To design and implement an open-safe-reliable hardware platform.
2. To guarantee extra-functional properties of FRACTAL nodes (dependability, security, timeliness and energy-efficiency).
3. To evaluate and validate data analytics with AI to identify the largest set of working conditions that allow preserving safe and secure operations.
4. To integrate fractal communication and remote management features into the nodes.

The project has a total of eight use cases in the domains of automotive, railway, and smart cities.

SPIDER Autonomous Robot Use-Case

The Smart PhysIcal Demonstration and Evaluation Robot (SPIDER) [3] is a mobile hardware-in-the-loop (HiL) platform allowing for reproducible testing of perception systems, vehicle software and control algorithms under real world conditions.

The SPIDER serves as vehicle demonstrator for the automotive domain in the FRACTAL project. It validates the applicability of performing computational intensive vehicle functions at the edge while still being able to guarantee extra-functional properties for preserving safety operational behaviours.

The objectives of the use case are:

1. Co-execution of safety-relevant and ML tasks.
2. Implement fail-operational capabilities within a single computing device even in the presence of common-cause faults.

The use case composes two core functions: the safety-critical collision avoidance function, preventing collisions with surrounding objects to avoid damage and human harm; the reinforcement learning approach based path tracking function to follow a predefined trajectory while evading obstacles on the track.

The functions are implemented at a FPGA running the RISC-V SoC from the SELENE project [4]. The FRACTAL nodes are embedded in ROS2 [5] nodes running on a Debian based Linux. The Robot Operating System (ROS) is a set of libraries and tools for building robot applications.

Safety Services

Two safety services are being integrated in the SPIDER use case, namely software-only support for diverse redundancy [6], and performance monitoring based on the SafeSU multicore interference statistics unit [7].

Diverse redundancy service. Safety-critical functions reaching ASIL-D in accordance with ISO 26262 must be realized with some form of diverse redundancy to avoid the so-called *common cause failures* (CCFs). Those failures arise upon a fault that generates an undetected error despite redundancy. If redundant elements (e.g., computing cores) have identical state simultaneously, a fault affecting both analogously (e.g., in the clock or power signals) could produce identical errors, which would not get noticed (i.e., a CCF). To mitigate CCFs, diverse redundancy is used so that, despite the fault can lead redundant items to error, such error will differ across redundant elements and hence, will be detectable.

In the context of automotive systems, the usual solution consists of using lockstep cores that execute the same instruction flow (i.e. redundantly) with some time staggering so that the electrical state of the cores differs at any point in time. This setup guarantees diversity in front of random hardware faults. Since our target SoC (the SELENE SoC [4]) lacks native lockstep cores, we have realized a software-only library providing such service: the SafeSoftDR library [6]. While it has some limitations (e.g., does not support code interfacing with I/O devices), it provides alternative means to guarantee that any cores running redundant code trail the conventional core by a guaranteed number of instructions. This enforces the required time staggering.

Multicore interference monitoring service. Multicores, like the SELENE SoC, are subject to timing interference across tasks running simultaneously in different cores, which may make them violate their deadlines, and hence, fail some safety requirements.

The SafeSU [7] provides capabilities to measure how much each core or accelerator interferes each other, and even programming interference quotas. These capabilities are particularly interesting upon a deadline overrun, since these statistics provide detailed information on how much each core has interfered each other.

Summary

The SafeSoftDR library is being integrated in the SPIDER use case by enabling its service for the collision avoidance function to guarantee error detection, even in front of CCFs. In particular, the library is invoked calling the monitor service, which launches the indicated function redundantly in two cores, and monitors progress making sure that the head process stays sufficiently ahead of the trail one.

The SafeSU is used in the SPIDER use case by resetting its counters prior to the execution of the collision avoidance function, and collecting results after its execution, hence obtaining key information on what core(s) or accelerator(s) is (are) the offending one(s) if the task under analysis overruns its deadline.

Integration of this services allow the SPIDER co-execution of safety-relevant and machine learning tasks on a single computing device.

References

[1] ECSEL FRACTAL consortium. *FRACTAL Research Project*. `https://fractal-project.eu/`. 2020.

[2] Aizea Lojo et al. "The ECSEL FRACTAL Project: A Cognitive Fractal and Secure edge based on a unique Open-Safe-Reliable-Low Power Hardware Platform". In: *2020 23rd DSD*. 2020, pp. 393–400.

[3] Virtual Vehicle Research GmbH. *SPIDER*. `https://www.v2c2.at/spider/`. 2023.

[4] H2020 SELENE consortium. *SELENE RISC-V open source hardware platform*. `https://gitlab.com/selene-riscv-platform`. 2021.

[5] Steven Macenski et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (2022).

[6] F. Mazzocchetti et al. "SafeSoftDR: a Library to Enable Software-Based Diverse Redundancy for Safety-Critical Tasks". In: *FORECAST: Functional Properties and Dependability in Cyber-Physical Systems Workshop (held with HiPEAC conference)*. 2022.

[7] G. Cabo et al. "SafeSU: an Extended Statistics Unit for Multicore Timing Interference". In: *IEEE ETS*. 2021.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Italy, Austria, Germany, France, Finland, Switzerland". In Austria the project was also funded by the program "IKT der Zukunft" of the Austrian Federal Ministry for Climate Action (BMK). BSC authors' work is also part of the project PCI2020-112010, funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR. The publication was written at Virtual Vehicle Research GmbH in Graz and partially funded within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Digital and Economic Affairs (BMDW), the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency (SFG).