A three-perspective analysis of RISC-V design tools for safety and security architectures

Javier Mora de Sambricio[1], Alejandro García Gener[1] and Gonzalo Salinas Hernando[1]

[1]Collins Aerospace ART - Ireland

Abstract

Aerospace industry requirements for SoC design demand high reliability and fault tolerance techniques while maintaining high standard performance in harsh environments and autonomous complex applications. The emergence of RISC-V architectures and toolsets offers a new paradigm for ASIC design, promising to accelerate and facilitate the design, verification, and testing of ASICs, reducing design time and costs. This work aims to demonstrate the feasibility of designing safe and secure architectures under the umbrella of open-source tools based on RISC-V. The safety and security constraints are analysed from three perspectives: hardware solutions and architecture definition at design level, the introduction of external IPs in the design with redundant techniques, and fault detection and mitigation techniques from external factors. The results indicate that RISC-V-based tools can be utilized to develop reliable and secure aerospace systems.

Introduction

Aerospace industry has stringent safety and security requirements that make embedded system design more challenging than most other industries in regard to System On Chip (SoC) design. The nature of the aerospace industry demands that all hardware and software systems operate with high reliability and fault tolerance in harsh environments. Moreover, aerospace industry is evolving towards better SWaP-C factor computing systems due to the trend of more electric-powered autonomous and intelligent systems. These systems will require additional computing power to run simultaneously multiple, complex, and AI-based applications on the same integration systems, which further exacerbates the challenges faced in meeting these requirements [1]. The failure of an embedded system in an aerospace application can result in catastrophic consequences, which is why safety and security constraints must be taken seriously.

In a parallel line, a new paradigm of ASIC design has emerged in recent years with the advent of RISC-V, promising to accelerate and facilitate the design, verification, and testing of ASICs, reducing design time and costs. Moreover, the open nature of the ISA makes the offer in terms of implementations and workflows very extensive, overcoming the obstacle of general-purpose commercial off-the-shelf (COTS) processors, which are not designed for safety-security critical applications, becoming impractically certifiable as its complexity arises.

Objectives

This works aims to demonstrate the feasibility of designing safe and secure architectures under the umbrella of open-source tools based on RISC-V, from architecture design based on Chisel/Scala up to its verification on programmable devices. This is achieved analysing the safety and security constrains from three different perspectives. First one relates to hardware design and architecture definition. In this stage, we used Chipyard ecosystem to add a hardware isolation for multicore platforms on top of the Rochet-chip architecture to reduce contention of shared components and adding hardware security and checksum mechanisms. The second level relates to the introduction of external legacy and parametrizable memory mapped devices into existing architectures, being able to provide Dual Modular Redundancy (DMR) or Triple Modular Redundancy (TMR) over I/O modules written in Verilog, System Verilog and VHDL. The third one aims to explore safety constrains from external factors that may change memory information such as ionizing particles or a circuit degradation. In this stage different ECC techniques over memory modules are analysed under the scope of RISC-V related tools.

Architecture design level

Although Rocket-chip architecture provides several benefits in terms of flexibility and performance, it is still limited in terms of safety. For that purpose, we decided to design a SoC architecture based on Rocket-chip but fully oriented to safety and security aerospace applications. This new architecture overcomes contention problems of complex architectures ensuring time determinism for certification purposes. It also adds a hardware security layer of isolation between subsystems, creating a physical separation between cores so data is not accessible even if one of the cores is under transversal attacks. In this example we based the architecture in the lockstep principle. These systems are fault-tolerant architectures that run the same set of operations at the same time in parallel.

Although there are already lockstep implementations based on RISC-V [2], we introduced a hardware communication mechanism between cores that is able to perform secure checksums through custom instructions so that, if some malfunction is detected, the whole SoC can be reset.

Safety integration of external IPs

SoCs are becoming increasingly complex and require the integration of numerous IP blocks to perform high-performance tasks, which can be either developed in-house or sourced from third-party vendors. Legacy IPs are pre-existing designs that have already been developed and tested and can be integrated into new designs to reduce development time and costs. The integration and compatibility of legacy IPs when designing custom SoCs is crucial, more so when there are specific safety requirements to be met. Chipyard already supports the introduction of external Verilog source code interacting with the Rocket-Chip architecture in the form of black boxes. However, there is no support for the most extended IP standard (IP-XACT) nor safety mechanisms.

For that reason, we built on top of the Rocket-chip architecture a set of Chisel classes to decipher IP-XACT files to obtain information about register size, addresses and shape. These classes read from XML files creating the necessary diplomatic nodes and interfaces to connect the Verilog code and the Rocket-Chip architecture.

In order to introduce safety capabilities, the aforementioned Chisel classes have been parametrized to give the design engineers the possibility to duplicate or triplicate the black box IPs introducing DMR and TMR techniques. The system automatically adds a voter module to perform redundancy mechanisms directly in hardware to not overload the processor.

We validated our results introducing the ARINC 429 interface, which is the most widely used avionics communication standard for commercial aviation, with a triple redundancy mechanism in a dual core Rocket-Chip architecture.

Fault-tolerant and error detection platforms

Some applications – especially those operating in harsh environments, such as satellite applications – are vulnerable to single-event upsets (SEUs) caused by ionizing particles, i.e., the unintended change of state in a memory element such as a flip-flop or block RAM (BRAM). The consequences of these changes can go from data corruption to a complete change of flow of the application.

In order to mitigate these faults, forward error correction (FEC) techniques are commonly used, such as error correction codes (ECCs). These techniques not only allow detecting faults in a circuit, but also correcting them so that the circuit can continue to operate normally even in the presence of faults.

Chipyard includes the possibility of adding ECC to the BRAMs in its caches and scratchpad memory components, which permit detecting and automatically correcting these faults without stopping the normal operation of the system [3]. Furthermore, it provides the possibility of triggering interrupts via the Bus Error Unit (BEU) so that the system can take action when a certain number of errors have been detected (which may indicate a sudden change in environmental conditions that should be countered) [4]. However, it does not include any fault injection mechanism for validation purposes.

This feature has been assessed in an FPGA implementation of the Rocket-chip design by emulating BRAM faults through the use of dynamic partial reconfiguration (DPR) capabilities of Xilinx FPGAs and MPSoCs. In order to emulate a SEU, the memory content is read through the internal configuration access port (ICAP) of the FPGA/MPSoC, one or several bits are flipped, and the result is written back. This methodology allows injecting faults in memory independently from the rest of the circuit. The ECC system responded as expected, automatically correcting the fault and triggering an interrupt right after it has been injected.

Discussion

Thanks to this work we demonstrated that the flexibility of the RISC-V ecosystem provides ample opportunities for safe and secure architectures, specifically for the aerospace industry. This is reinforced by the fact that the community has demonstrated a strong commitment on these architectures, providing further confidence in the platform's ability to support secure systems.

References

[1] Woodrow Bellamy. “Avionics Industry Advances Toward DAL A Multicore Adoption”. In: aviationtoday.com (2020).

[2] C. Rodrigues, I. Marques, S. Pinto, T. Gomes and A. Tavares, "Towards a Heterogeneous Fault-Tolerance Architecture based on Arm and RISC-V Processors," IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 2019, pp. 3112-3117, doi: 10.1109/IECON.2019.8926844.

[3] A. Dörflinger, Y. Guan, S. Michalik, S. Michalik, J. Naghmouchi and H. Michalik, “ECC memory for fault tolerant RISC-V processors”, Architecture of Computing Systems – ARCS 2020, pp. 44-55, 2020.

[4] SiFive, Inc. “SiFive U74-MC Core Complex Manual”, Chapter 12. (2021).