SafeSU-2: a Safe Statistics Unit for Space MPSoCs

Guillem Cabo[†], Sergi Alcaide[†,‡], Carles Hernández[*], Pedro Benedicte[†],
Francisco Bas[†,‡], Fabio Mazzocchetti[†], Jaume Abella[†]
[†]Barcelona Supercomputing Center (BSC) [‡]Universitat Politècnica de Catalunya (UPC)
[*]Universitat Politècnica de València (UPV)

***Abstract*—Advanced statistics units (SUs) have been proven effective for the verification, validation and implementation of safety measures as part of safety-related MPSoCs. This is the case, for instance, of the RISC-V MPSoC by CAES Gaisler based on NOEL-V cores that will become commercially ready on FPGAs by the end of 2022. However, while those SUs support safety in the rest of the SoC, they must be built to be safe to be part of commercial products.**

This paper presents the SafeSU-2, the safety-compliant version of the SafeSU. In particular, we perform a Failure Mode and Effect Analysis (FMEA) for the SafeSU for relevant fault models, and implement fault detection and tolerance features needed to make it compliant with the requirements of safety-related devices in general, and of space MPSoCs in particular.

I. Introduction

Functional safety is needed for some space, avionics and automotive systems among others. Different safety standards and guidelines exist for electronic systems describing the development process for those systems (e.g. ECSS-Q-ST-60-02C for the space domain [4]), which include specific processes for the design, verification and validation (V&V) of the system thereof. In particular, the architecture of the system must include safety measures for fault detection and/or tolerance, as well as support to ease V&V in the form of observability and controllability features.

A RISC-V multicore by CAES Gaisler has been recently unveiled [5] offering enhanced capabilities for space missions. It will be released commercially on FPGA by 2022 [3], and is intended to include the SafeSU statistics unit [1], [2], which implements a number of features for the observability and controllability of multicore interference, as needed to guarantee that real time requirements of space applications will be met in accordance with space safety regulations.

However, while the SafeSU provides safety measures, as well as V&V means for the processor, it must also include fault detection and/or tolerance mechanisms so that its robustness is in pace with that of the rest of the multicore, specially given the harsh environmental conditions for space operation.

This paper presents the SafeSU-2 (read as "saves you too"), which extends the SafeSU with suitable fault detection and tolerance features able to deal with at least single event upsets (SEUs), in line with the rest of the SoC.

II. SafeSU-2 Architecture

The SafeSU-2 is a statistics unit part of a space-graded SoC. The schematic of the relevant parts of the SoC and the main SafeSU-2 components is shown in Figure 1. As shown, the SoC includes 4 cores with their respective first level instruction and data caches, connected through an AMBA Advanced High-performance Bus (AHB) to a shared second level cache which, in turn, is connected to a memory controller that serves as bridge to the off-chip SDRAM memory. Although not shown in the picture, the SoC includes an AMBA Advanced Peripheral Bus (APB) connected to the AHB where all I/O and debug interfaces are connected.

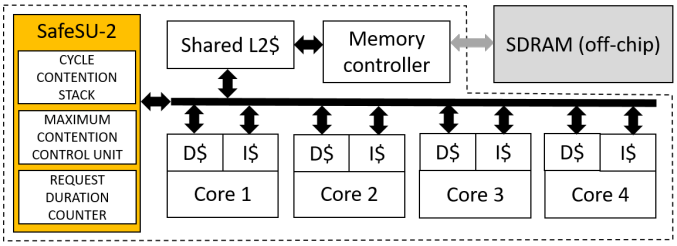

Fig. 1. High-level schematic of the relevant parts of the SoC with the SafeSU/SafeSU-2.

The SafeSU-2 has the same functional components as the original SafeSU, namely, a Cycle Contention Stack (CCS), a Maximum Contention Control Unit (MCCU), and a set of Request Duration Counters (RDCs). The CCS allows collecting statistics of the contention experienced by each core to access the bus broken down by contender core, hence easing application optimization during design phases, and providing accurate diagnosis information during operation in case of deadline overruns. The MCCU is a contention quota mechanism that allows setting how much interference each core is allowed to cause on each other, raising an interrupt when any quota is exceeded. Hence, the MCCU allows implementing safety measures to limit inter-core interference. Finally, the RDCs provide statistics of the maximum duration of different types of events, hence allowing estimating bounds on maximum contention expected during verification phases. Also, the RDCs can be pre-programmed with a specific latency per event and raise an interrupt if a higher latency is observed, hence allowing to implement additional safety measures to detect whether any component is abnormally kept busy by a single request.

The internal architecture of the SafeSU (and SafeSU-2) is as shown in Figure 2. It includes the following modules:

- **SafeSU2_ahb**. AHB interface and SafeSU-2 configuration registers.
- **Crossbar**. Registered outputs and externally programmable MUXes to interface SafeSU-2.
- **SafeSU2_raw**. Interface-independent SafeSU-2 module with RDC and MCCU configuration signals.
- **CCS counters**. CCS contention counters. They are configuration-independent (depend only on the core count).
- **RDC**. RDC module with internal registers, but excluding configuration signals.
- **MCCU quotas**. MCCU module with programmable quotas, but excluding configuration signals.
- **Overflow reporting**. CCS, RDC and MCCU counters overflow reporting.

 http://dx.doi.org/10.23919/DATE54114.2022.9774515

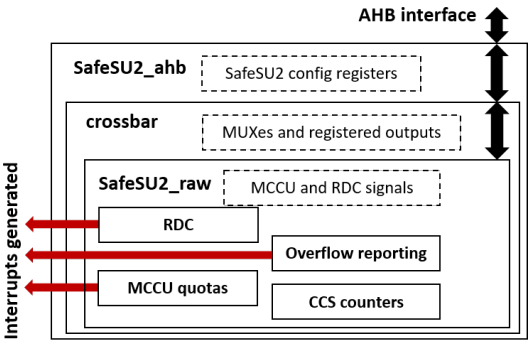

Fig. 2. Internal components of the SafeSU/SafeSU-2.

Note that the latter 3 modules can raise external interrupts if (1) a pre-programmed maximum latency is exceeded, (2) a quota is exhausted, or (3) a counter experiences overflow respectively.

III. SafeSU-2 Safety Measures

A Failure Mode and Effect Analysis (FMEA) has been conducted on the SafeSU to determine the safety extensions needed to obtain the SafeSU-2. Such process requires analyzing each individual component of each individual module, considering all relevant faults, and analyzing whether and how they propagate and what effects they may have on the overall SafeSU-2. Due to space constraints, we illustrate this process only for two components of the *SafeSU2_ahb* module in Table I omitting some low level details such as register names and the like for clarity.

TABLE I
A subset of the FMEA for the *SafeSU2_ahb* module.

Failure mode	SEU in configuration registers
Effect	May cause complete unit failure
Impact	High
Safety measure	Protect each register with SECDED codes

Failure mode	SEU in AHB protocol state machine
Effect	May cause misbehavior regarding the AHB protocol, and also internal updates
Impact	High
Safety measure	Due to the few signals and state bits, protect with triple modular redundancy (TMR)

Overall, the different failure modes have been addressed with TMR for small, yet critical, components involving combinatorial logic; SECDED for critical registers; parity for lowly critical registers; and with no safety measure at all for negligible faults (e.g. missing to increment a cycle contention counter). In the case of parity, which is used, for instance, for the CCS counters, upon a fault detection, an interrupt is triggered to allow software layers reset the unit as needed. Interrupts are also raised for 2-bit errors with SECDED, or 3 different outputs with TMR. Note that such an approach is acceptable based on the assumption that having two independent faults occurring in a very short timeframe (e.g. some microseconds) has negligible probability even in harsh environments like outer space.

IV. Evaluation

We have synthesized both the SafeSU (non fault-tolerant) and SafeSU-2 with the Vivado 2018 Toolchain and target the FPGA present in the Xilinx Ultra-Scale KCU105, which is a relevant development FPGA platform for space systems, for a target frequency of 100MHz. The number of Look-up Tables (LUTs) and Flip-Flops (FFs) used by both configurations are shown in Table II. As shown, the increase in terms of FPGA resources is relatively low for SafeSU-2 w.r.t. the non fault-tolerant counterpart (SafeSU).

TABLE II
FPGA resources used.

Target	**FPGA**		**ASIC**		
	LUTs	**FFs**	**500MHz**	**750MHz**	**1GHz**
SafeSU	6,075	2,734	87,283	106,365	106,365
SafeSU-2	8,350	3,010	218,372	282,607	285,156
Increase	37.4%	10.1%	150.2%	165.7%	168.1%

We have further synthesized both, the SafeSU and SafeSU-2 for TSMC 65nm technology for different frequencies and estimated the area (in μm^2). Results are shown in Table II. As shown, the area increase for an ASIC implementation is much larger than for an FPGA implementation. This occurs because, while fault detection and tolerance features do not impact the critical path in the case of the FPGA, they do in the ASIC implementation. Hence, more expensive standard cells are selected in the case of the ASIC implementation for the SafeSU-2 to meet timing requirements, which ultimately leads to much larger area requirements.

Finally, the SafeSU and SafeSU-2 have been proved to be functionally equivalent with formal verification. In particular, both units have passed equivalence checking with a formal proof of depth 25 with SymbiYosis [6].

V. Conclusions and Future Work

Our implementation of SafeSU-2, the fault-tolerant version of SafeSU, allows integrating it in the commercial space-graded SoC by CAES Gaisler to be available in 2022. Such product, which will reach the market in the form of a SoC on an FPGA, will enjoy SafeSU-2 with low additional cost w.r.t. SafeSU. Part of our future work is performing an extensive fault injection campaign of SafeSU-2 in the complete SoC before reaching the market, including scenarios such as mission-critical applications, as well as improving the ASIC version of the unit to reduce its relative cost.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 871467. This work has also been partially supported by the Spanish Ministry of Science and Innovation under grant PID2019-107255GB-C21/AEI/10.13039/501100011033.

References

[1] BSC. SafeSU gitlab. https://gitlab.bsc.es/caos_hw/hdl_ip/bsc_pmu/.
[2] G. Cabo et al. SafeSU: an extended statistics unit for multicore timing interference. In *IEEE European Test Symposium (ETS)*, 2021.
[3] De-RISC Consortium. Dependable Real-Time Infrastructure for Safety-critical Computer. https://derisc-project.eu/.
[4] ECSS. *ECSS-Q-ST-60-02C, ASIC and FPGA development*, 2008.
[5] G. Wessman et al. De-RISC: the first RISC-V space-grade platform for safety-critical systems. In *IEEE Space Computing Conf. (SCC)*, 2021.
[6] YosysHQ. SymbiYosis. https://github.com/yosyshq/symbiyosys.