A 3.6TOPS/W, 550GFLOPS/W Adaptive Fault-Tolerant SoC for Satellite Onboard AI & Control

Yvan Tortorella*, Pierangelo Rapa*, Luigi Ghionda*, Maicol Ciani†, Riccardo Tedeschi†, Andrea Belano†, Alessandro Piccini†, Arya Baghche Saraei†, Michael Rogenmoser‡, Enrico Zelioli‡, Zexin Fu‡, Victor Isachi†, Chaoqun Liang†, Angelo Garofalo†‡, Davide Rossi*†, Francesco Conti†, Luca Benini†‡,
*Fondazione Chips-IT, Italy †University of Bologna, Italy ‡ETH Zürich, Switzerland

***Abstract*—We present Astral, a heterogeneous SoC for advanced satellite on-board control an AI functions in 12 nm technology, integrating a fault-tolerant general-purpose host and a programmable data-parallel accelerator with dynamically reconfigurable protection against transient and permanent faults, ultra-low-latency (32ns) fault recovery, and interconnect dataflow protection. Astral can be dynamically reconfigured to operate in fault-tolerant mode, or to exploit all on-chip processing units for parallel, irredundant computation, to achieve 630GOPS and 72GFLOPS performance in a 500mW power envelope.**

***Index Terms*—Fault-tolerance, Heterogeneous safety-critical SoC, Hardware Acceleration, Reliable Computing**

I. Introduction

Modern satellites produce vast volumes of telemetry and high-resolution payload data, often exceeding downlink bandwidth and limiting real-time responsiveness [1], [2]. To mitigate this bottleneck, onboard AI processing offers a promising solution. In addition, space processors must remain reliable under radiation and aging effects while managing mixed-criticality workloads through adaptive protection that can be traded off for performance when reliability constraints are relaxed [3]. To address the aforementioned challenges, we present Astral, a heterogeneous reconfigurable SoC featuring three main contributions: 1) a Linux-capable Host Subsystem based on a dual-core cache-coherent RV64 processor with programmable split/lock reliability scheme; 2) an Accelerator Subsystem based on a runtime-configurable Dual Modular Redundancy (DMR)/Triple Modular Redundancy (TMR) programmable accelerator with extremely fast 32 ns fault recovery and an Error-Correcting Codes (ECC)-extended memory interconnect providing dataflow protection; 3) programmable Tensor, Neural, and Softmax accelerators achieving up to 630 GOPS @ 3.6 TOPS/W @ 8b, 72 GFLOPS @ 550 GFLOPS/W @ FP16; 4) a Secure Subsystem providing 67MB/s and 143MB/s bandwidth on hardware-accelerated AES-GCM and SHA-256 respectively. Figure 2 shows the Astral architecture.

II. Reconfigurable Redundant Cache-Coherent Dual Core Processor

Astral's Host Subsystem is a dual-core, snoop-based MOESI cache-coherent cluster of 64-bit CVA6 cores based on the AXI-Coherence Extended (ACE) protocol. The Cache Coherency Unit (CCU) transparently forwards non-coherent traffic while arbitrating snoop requests and responses for coherent transactions. The core requests exiting the coherent domain converge into a unified bus connected to the system crossbar. The two CVA6 cores have a 6-stage in-order pipeline with 16kiB L1 instruction cache and 32kiB L1 data cache. The data cache contains 256 lines split into two ways. Lines are 128b wide, and are protected with Single-Error Correction, Double-Error Detection (SEC-DED) ECC with a granularity of 32b,

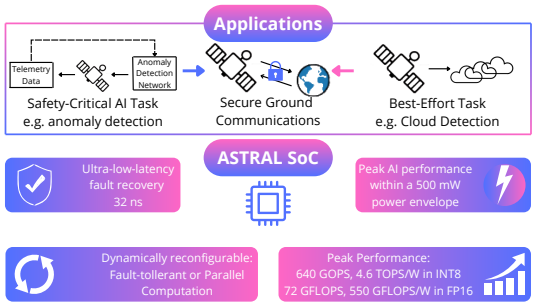


Fig. 1. Energy-efficient open-HW&SW designs support space and reliable demanding use-case scenarios and applications.

correcting up to four simultaneous errors within a single line. Additionally, a self-invalidation mechanism is triggered if multiple errors are detected within a single cache word. The two Host Subsystem cores are wrapped in a runtime-configurable DMR unit to split/lock the two cores in independent/DMR mode.

In independent mode, the two cores act as independent hardware threads providing up to 408DMIPS with no error detection. When configured for DMR execution, the interfaces of the two cores are constantly compared by a DMR checker, detecting if a soft error has corrupted one of the two cores. In case of error detection, the DMR checker prevents the cores output from being propagated to the system, while the DMR FSM flushes the two cores. The cores then restore the execution from a safe checkpoint, consisting of the Register File and CSR content, while the data caches remain consistent and correct thanks to the SEC-DED ECC scheme. The DMR checker is implemented as a bitwise XOR of the two cores buses. In case of inconsistency, the XOR-ed bus feeds an OR tree, signaling an error occurrence. Each ACE bus is 561b wide, thus a single, high fan-in OR tree for error detection would limit the maximum achievable frequency. Instead, the Host Subsystem DMR checker implements dedicated checkers for each ACE channel, reducing the maximum fan-in. The Host Subsystem also features a 128 kiB Last-Level Cache (LLC) consisting of 256 lines, 512b each. The LLC is also protected by SEC-DED ECC scheme with a granularity of 32b, allowing for correction of up to 16 errors within a single cache line, and cache line invalidation in case of multiple errors detected within a single 32b word.

We evaluated the Astral Host Subsystem performance running the SPLASH3 benchmark on Linux/RTEMS in both independent and DMR mode. The Host achieves up to 480MIPS and 255MIPS in independent and DMR configurations respectively, with a DMR slowdown of just 1.88× compared to the independent mode, while ensuring reliable execution.

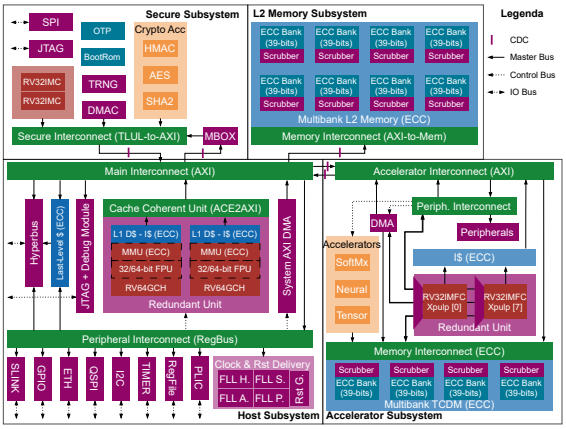


Fig. 2. Astral SoC architecture: Host Subsystem (bottom left); Accelerator Subsystem (bottom right); global L2 Memory (top right); Secure Subsystem (top left).

III. Runtime-Configurable Redundant Accelerator

Figure 3 provides details of the Astral Accelerator Subsystem architecture and its reliability features. A 128kiB Tightly-Coupled Data Memory (TCDM) is split into 16 banks storing 32b information + 7b correction for SEC-DED ECC, in addition to dedicated programmable scrubbers. The initiator agents access the TCDM via a reliable memory interconnect protected with ECC Encoders (ENCs) and Decoders (DECs), each consisting of a Hsiao encoder/decoder pair. During a memory request, the corresponding ECC encoder generates check bits for metadata (address, byte enable, write enable) and payload. For accelerators, the encoding occurs in the accelerators load/store unit via a single Hsiao encoder. For cores, the encoding happens after the crossbar via dedicated ECC encoder. On the accelerators' side, an additional ECC decoder operates exclusively on metadata, enabling the routing from a single 32b address, 288b wide data channel to memory-bank-compatible requests, while downstream ECC encoders restore metadata protection. The resulting ECC-extended memory transaction is arbitrated between accelerators and RISC-V cores with identical protection schemes. Decoding and correction of the metadata happens before reaching the TCDM banks. The 39b payload is stored during write requests, and traverses the fault-tolerant interconnect being decoded upon arrival either on the cores or accelerators side during reads.

The RISC-V cores in the Astral Accelerator Subsystem communicate with a hierarchical instruction cache split into 4kiB shared and 512B private portions. The cache is extended with data and tag parity bits for cache line invalidation and replacement upon error detection, providing reliability and 85.7% less overhead compared to SEC-DED ECC schemes. The RISC-V cores communicate with the system via a dedicated logic that allows for Hybrid Modular Redundancy (HMR) runtime reconfiguration schemes. The HMR unit can be configured in software to let the cores run in Independent mode if a low-reliability, high-performance task needs to be executed. In case of safety-critical tasks execution, the HMR logic can be programmed to run the RISC-V cores in DMR or TMR mode. While DMR provides the best performance/recovery latency ratio for soft error mitigation, the TMR mode can also filter stack-at faults happening in the core logic.

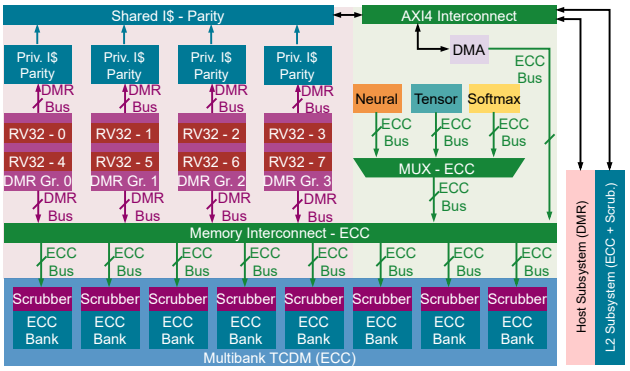


Fig. 3. Accelerator Subsystem architecture, with highlights on memory interconnect protection with ECC down to the memory banks with dynamic scrubbers, I$ protection with parity, RISC-V cores grouped in Hybrid Modular Redundancy logic with DMR example grouping, Application-Specific Accelerators and DMA controller.

Both DMR and TMR schemes share a Quick Recovery hardware extension consisting of ECC-protected replicas of the status registers (Register File, Program Counter, CSRs) of the cores used to store their voted or checked state. The cores status bus consistency is ensured by a dedicated checker with a pipeline stage, preventing high fan-in logic from limiting design frequency. In case of failure, the status is not stored in the Quick Recovery region. The cores are flushed, and the content of the ECC-protected Quick Recovery state registers is used to safely restore the state of the cores. The Quick Recovery requires a fixed amount of 24 clock cycles (38.6ns at 620MHz) to restore a consistent operating state, achieving fast real-time deterministic fault mitigation. Performance-wise, the DMR and TMR configurations introduce a $1.88\times$ and $2.86\times$ slowdown respectively over the irredundant configuration.

The Accelerator Subsystem of the Astral SoC is extended with dedicated accelerators boosting AI workloads which do not require hardwae-based fault protection. The Accelerator Subsystem includes a CNN accelerator based on 16 engines with 32 8b-MAC each, providing up to $6\times$ speedup compared to the RISC-V cores; a Tensor accelerator based on a 12×4 systolic array of FP16 FMAs (up to $28\times$ faster than the RISC-V); and a 16 units special function engine capable of autonomously executing SoftMax, GELU and other non-linear operators in transformers with up to $10.8\times$ speedup over software implementation. While the accelerators are not internally protected, the Accelerator Subsystem interconnect is fully ECC-protected to prevent fault-induced deadlocks and other system-level error conditions. Figure 6 shows that 21.6% of the SoC area is constituted of ECC-protected SRAMs and interconnects, while 12.4% is protected by other mechanisms. The remaining surface is dedicated to accelerators for workloads that do nor require hardware fault tolerance.

IV. End-to-End Application Measurements

The RISC-V cores configured in DMR mode expose checked and safe bus transactions both towards the protected instruction cache and the protected memory interconnect and TCDM, providing a hardware configuration capable of running reliably safety-critical AI workloads. In Figure 4 we demonstrate this capability by executing a safety-critical anomaly detection Temporal Convolutional Network (TCN) [1], [4] used to reliably analyze satellite telemetry to 1) take action isolating malfunctioning satellite components and 2) securely communicate the result to the ground station. The Accelerator

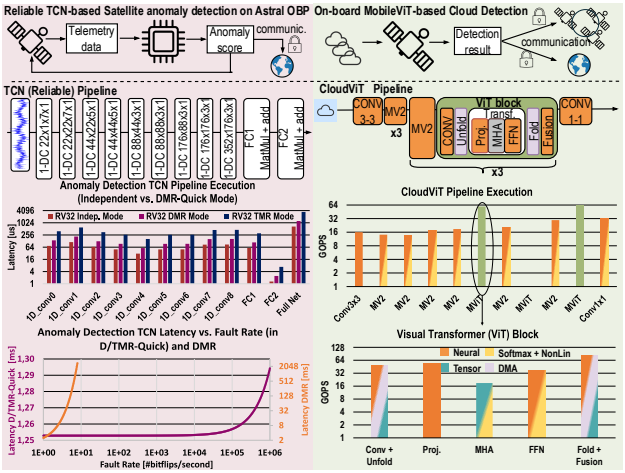


Fig. 4. Application mapping of a TCN for anomaly detection executed on the DMR cores, and a cloud detection transformer executed on the accelerators.

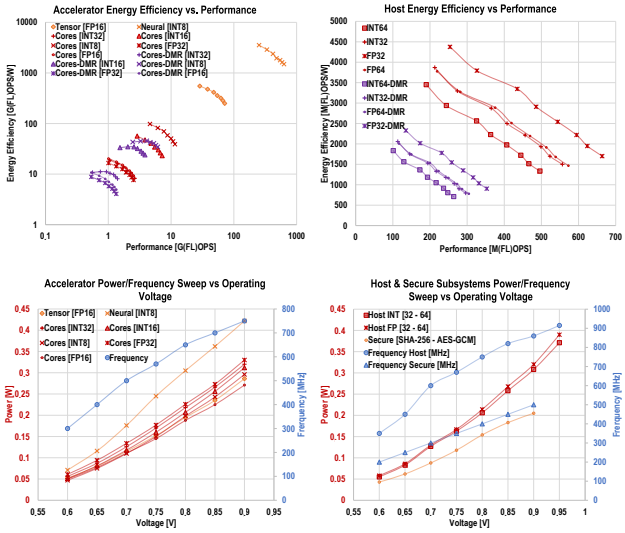
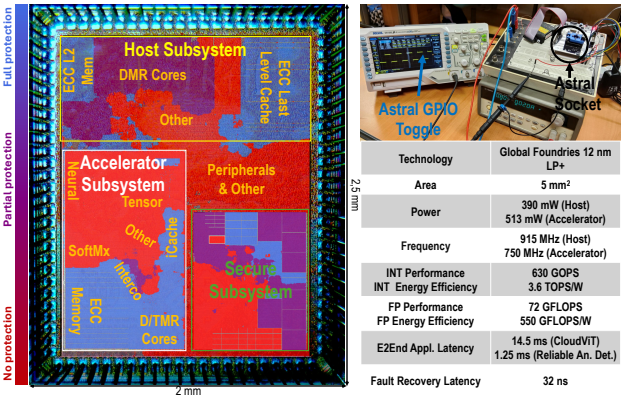


Fig. 5. Astral power, performance, and frequency measurements on the Host and Accelerator Subsystems. All operating points are obtained by sweeping the supply voltage between 0.6 V and 0.9 V in 0.05 V increments. In the top plots, the performance and energy-efficiency values are computed at the maximum frequency corresponding to each operating point shown in the bottom plots.

Subsystem demonstrates 1.9, 2.9, and 5.3 average GOPS during the TCN execution in TMR, DMR, and Independent mode respectively, with higher performance coming at the cost of lower fault reliability capabilities. We also analyse the latency introduced by fault recovery in DMR and TMR mode with the Quick Recovery mechanism and compare it with simple DMR with cluster reset after each fault detection. In case of DMR with Quick Recovery, the latency of the TCN execution remains within the range of 1.25ms even with a fault rate equal to 106faults/s, which is 108× higher than the low-orbit rate in case of solar flares [5], [6]. On the contrary, the latency introduced by the reset-based fault mitigation in the DMR case reaches 16s (10000× the expected execution time) at less than 1fault/s exposure.

The Accelerator Subsystem is also used to run AI tasks that are not safery-critical such as image cloud detection. We rely on a Visual Transformer (ViT) network trained for in-orbit



Technology	Global Foundries 12 nm LP+
Area	5 mm²
Power	390 mW (Host) 513 mW (Accelerator)
Frequency	915 MHz (Host) 750 MHz (Accelerator)
INT Performance INT Energy Efficiency	630 GOPS 3.6 TOPS/W
FP Performance FP Energy Efficiency	72 GFLOPS 550 GFLOPS/W
E2End Appl. Latency	14.5 ms (CloudViT) 1.25 ms (Reliable An. Det.)
Fault Recovery Latency	32 ns

Fig. 6. Astral SoC micrograph picture, testing and measurement setup, and resume table.

cloud detection [7] and demonstrate the use of all the available accelerators to execute it, achieving 31.6 average G OPS for the entire network execution, and 64 G OPS for the ViT layer. The Astral Host Subsystem orchestrates the AI workloads execution safely running a space-qualified Real-Time OS (RTEMS) and Linux in a reliable DMR configuration.

Before downlink transmission, Astral leverages the Secure Subsystem to protect the telemetry data and the satellite reconfiguration commands produced by the Accelerator Subsystem during TCN and cloud detection networks. The Secure Subsystem provides authenticated and confidential satellite-to-ground and satellite-to-satellite communication through hardware-accelerated AES-GCM and SHA-256 [8], [9], combined with secure boot and protected key storage. Access to the Secure Subsystem is strictly mediated through a mailbox-based interface. Other subsystems can only issue requests via message passing, triggering the Secure Subsystem to retrieve the required data and execute the corresponding security operations internall, ensuring strong isolation of sensitive assets, and a trusted execution environment. The hardware-accelerated AES-GCM and SHA-256 functions achieve 67MB/s and 143MB/s bandwidth, respectively, with 27.5× and 26.8× speedup over the Host Subsystem and a power envelope of 205mW. Figure 5 shows the performance, power, and frequency results of Astral's physical characterization.

Table I compares Astral with 6 state-of-the-art embedded processors, microcontrollers, and heterogeneous SoCs for embedded AI and mixed-criticality systems, including several reliable designs. Astral combines leading-edge capabilities in terms of SEU protection and recovery with state-of-the-art AI acceleration capabilities to support satellite onboard AI workloads, delivering leading efficiency of 550G FLOPS/W in FP16 and second-best of 3.6T OPS/W in INT8. Astral is protected with ECC or architectural redundancy 33% of the system, with the Host Subsystem delivering up to 255MIPS in DMR and the Accelerator Subsystem providing up to 7.2GOPS in protected mode, with 32ns real-time fault recovery. This enables the Astral SoC to serve in a vast diversity of space scenarios characterized by variable levels of fault tolerance. The cooperation among the subsystems of the Astral chip enables complex heterogeneous workloads, including real-time or full-fledged OS operation, AI acceleration, and cryptographic functions, within a 1W power envelope, achieving high performance, security, safety, and real-time fault recovery under strict power budgets.

TABLE I
STATE OF THE ART COMPARISON TABLE.

	This Work	ICECS23 [10]	CICC25 [11]	GR740 [12]	ISSCC19 [13]	TCSI25 [14]	ISSCC23 [15]
Technology	*GF12LP+*	TSMC28	22nm	ST 65nm RADHARD	28nm SG-MONOS	Intel16	GF22
Area [mm^2]	*5*	1.9	2.35	N.A.	94.3	16	18.7
Cores	*2×RV64GCH (Host)* *8×RV32IMFCXflexv (Accelerator)*	3×RV32IMC	16×RV32IM(V)	4×SparcV8 32-bit	4×RISC32	2×RV64GCH 3×RV32RT 12×RV32IMFCXflexv	1×RV32IMFC 16×RV32IMFCXpulpNN
ML Accelerators	*FP/INT (Reliable/Scalar)* *12 × 4 FP16 (Tensor)* *16 × 32 INT8 (Neural)* *BF16 (Softmax)*	✗	INT8 Vector	✗	✗	INT Reliable Scalar FP Vector	9 × 32 INT8 (Neural)
Power Envelope [mW]	*1000 (SoC)* *55 - 390 (Host)* *47 - 513 (Accelerator)* *43 - 205 (Secure)*	15 - 300	3.73 - 50.1 (Multicore) 7.73 - 110 (Engine)	42 - 1161	500	1200 (SoC) 100 - 700 (Host) 50 - 747 (Integer) 29 -600 (Vector)	12-8 - 128
Frequency [MHz]	*350 - 915 (Host)* *300 - 750 (Accelerator)* *200 - 450 (Secure)*	100 - 300	123 - 541 (RISC-V) 340 - 1225 (Engine)	250	600	435 - 1000 (Host) 225 - 825 (Integer)	100 - 420
Peak INT Perf. [GOPS]	*8 / 16 / 32-bit* ***630 (Neural)** / 7.2 / 3*	8 / 16 / 32-bit - / - / 0.13	31 (RISC-V (INT8) 527 (Engine - INT8)	N.A.	N.A.	8 / 16 / 32-bit 78.5 / 25 / 10	8 / 16 / 32-bit 90 (RBE) / - / -
Peak INT E.E. [GOPS/W]	*8 / 16 / 32-bit* *3547 (Neural) / 57 / 19*	8 / 16 / 32-bit - / - / 5.6	1900 (RISC-V) 17000 (Engine)	N.A.	N.A.	8 / 16 / 32-bit 412 / 200 / 50	8 / 16 / 32-bit 1900 (RBE) / - / -
Peak FP Perf. [GFLOPS]	*16 / 32-bit* ***72 (Tensor)** / 2.5*	✗	✗	0.023	N.A.	16 / 32-bit 61.5 / 31.3	16 / 32-bit 6.9 / -
Peak FP E.E. [GFLOPS/W]	*16 / 32-bit* ***550 (Tensor)** / 16.5*	✗	✗	N.A.	N.A.	16 / 32-bit 457.8 / 197.8	16 / 32-bit 207 / -
End2End Application latency [ms]	***14.5 (INT8/FP16 CloudViT)*** ***12.1 (INT8 MobileNetV2)*** ***1.25 (INT8 Reliable Anom. Detect.)***	✗	✗	✗	✗	✗	48 (ResNet-18) 28 (MobileNetV2)
Redundancy Scheme	*Host: DMR* *Accelerator: DMR+ TMR (Quick Recovery)*	TMR	DMR	RadHard	DMR	Safe: TMR Cluster: DMR+ TMR (Rapid)	✗
Runtime-Configurable Redundancy	*Host* ✓ *Accelerator:* ✓	✓	✗	✗	✗	Safe: ✗ Cluster: ✓	✗
Recovery Latency[ns]	*32*	1210000	N.A.	N.A.	N.A.	30	✗
ECC + Dynamic Scrubbing]	✓	✓	✗	✗	✗	✗	✗
ECC + Dynamic Scrubbing]	✓	✗	✗	✗	✗	✗	✗

V. CONCLUSIONS

We presented Astral, a highly reconfigurable heterogeneous SoC achieving 30ns real-time fault recovery, up to 550G FLOPS/W in FP16 and 3.6T OPS/W in INT8, and integrated hardware-accelerated encryption within a 1W power envelope, targeting a wide range of space applications with varying criticality levels while supporting multiple configurations trading off performance for fault tolerance.

REFERENCES

[1] Y. Wang, Y. Wu, Q. Yang, and J. Zhang, "Anomaly Detection of Spacecraft Telemetry Data Using Temporal Convolution Network," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Glasgow, United Kingdom, 2021, pp. 1–5.

[2] M. Villano, G. Krieger, and A. Moreira, "Onboard Processing for Data Volume Reduction in High-Resolution Wide-Swath SAR," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 8, pp. 1173–1177, Aug 2016.

[3] H. Tan, S. Sun, T. Cheng, and X. Shu, "Transformer-Based Cloud Detection Method for High-Resolution Remote Sensing Imagery," *Computers, Materials and Continua*, vol. 80, no. 1, pp. 661–678, 2024.

[4] Y. He and J. Zhao, "Temporal Convolutional Networks for Anomaly Detection in Time Series," *Journal of Physics: Conference Series*, vol. 1213, p. 042050, 2019.

[5] Y. Bentoutou and E. H. Bensikaddour, "Analysis of Radiation Induced Effects in High-Density Commercial Memories On-Board Alsat-1: The Impact of Extreme Solar Particle Events," *Advances in Space Research*, vol. 55, no. 12, pp. 2820–2832, 2015.

[6] M. Sajid, N. G. Chechenin, F. S. I. Torres, U. A. Gulzari, M. U. Butt, Z. Ming, and E. U. Khan, "Single Event Upset Rate Determination for 65nm SRAM Bit-Cell in LEO Radiation Environments," *Microelectronics Reliability*, vol. 78, pp. 11–16, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026271417303554

[7] B. Zhang, Y. Zhang, Y. Li, Y. Wan, and Y. Yao, "CloudViT: A Lightweight Vision Transformer Network for Remote Sensing Cloud Detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[8] L. Crocetti, L. Fanucci, L. Baldanzi, and F. Falaschi, "A scalable and configurable architecture for hardware authenticated encryption modules compliant with the ccsds security specifications," in *2024 Security for Space Systems (3S)*, 2024, pp. 1–8.

[9] "Sace Data Link Security Protocol," https://ccsds.org/Pubs/355x0b2.pdf.

[10] M. Rogenmoser and L. Benini, "Trikarenos: A Fault-Tolerant RISC-V-Based Microcontroller for CubeSats in 28nm," in *2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Istanbul, Turkiye, 2023, pp. 1–4.

[11] Q. Cheng *et al.*, "A 22nm Resource-Frugal Hyper-Heterogeneous Multi-Modal System-on-Chip Towards In-Orbit Computing," in *2025 IEEE Custom Integrated Circuits Conference (CICC)*, Boston, MA, USA, 2025, pp. 1–3.

[12] "Quad Core LEON4 SPARC V8 Processor, GR740, Cobham Gaisler," https://download.gaisler.com/products/gr740/doc/Product%20Brief_GR740_20230908_rev1.pdf, accessed: 2025-11-10.

[13] S. Otani *et al.*, "A 28nm 600MHz Automotive Flash Microcontroller with Virtualization-Assisted Processor for Next-Generation Automotive Architecture Complying with ISO26262 ASIL-D," in *2019 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2019, pp. 54–56.

[14] A. Garofalo *et al.*, "A Reliable, Time-Predictable Heterogeneous SoC for AI-Enhanced Mixed-Criticality Edge Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2025.

[15] F. Conti *et al.*, "A 12.4TOPS/W @ 136GOPS AI-IoT System-on-Chip with 16 RISC-V, 2-to-8b Precision-Scalable DNN Acceleration and 30% Boost Adaptive Body Biasing," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2023, pp. 21–23.