

Case-Study for Integration of COTS SoC Devices in Reliable Space Systems for On-Board Processing

Ivan Rodriguez-Ferrandez^{*†‡}, David Steenari[‡], Maris Tali[‡], Leonidas Kosmidis^{†*},
Ferdinando Tonicello[‡]

^{*}Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

[†]Barcelona Supercomputing Center (BSC), Barcelona, Spain

[‡]European Space Agency (ESA), Noordwijk, The Netherlands

Abstract—Recent trends in Data Handling Systems (DHS) include increased data-rates, in-orbit reconfiguration and the introduction of advanced On-Board Processing (OBP) methods to extract actionable information on-board, with low latency. At the same time, NewSpace industry has successfully deployed COTS-based DHS and processing equipment in space. In particular for OBP, COTS-based processors and FPGAs can offer higher computational performances than space qualified equivalents – which in turn can enable new applications through more advanced OBP algorithms. The individual component cost is also lower for COTS than space qualified components, which can allow overall cost-optimisations on mission level.

An internal working group at ESA has studied the concept of using “safety barriers” to ensure no propagation of failure from functions implemented with COTS to other on-board units. Thanks to ESA studies there is a path to use Class IV equipment on missions with higher class (Class I-III), through the use of hardware “safety barriers” to limit failure propagation from Commercial Off-The-Shelf (COTS) equipment.

In our work, we take the above mentioned concepts and apply them to reference implementations, targeting payload processing modules based on standard form factors (ADHA) and automotive-grade GPU Systems-on-Chip (SoCs). In addition, we make use of previous work on improving in-flight availability of complex COTS SoC processors, through system and software Fault Detection, Isolation and Recovery (FDIR) techniques.

Index Terms—Data Handling Systems, On-Board Processing, Space Systems, Reliable Systems

I. INTRODUCTION

Recent trends in DHS include increased data-rates, in-orbit reconfiguration and the introduction of advanced OBP methods to extract actionable information on-board, with low latency, using Artificial Intelligence (AI) algorithms [1] [2] [3] [4].

The European Space Agency (ESA) and European Space Industry have recognised the importance of ensuring the availability of mature and high-performance DHS equipment for institutional Earth Observation (EO) missions. To achieve this, the ADHA modularity concept has been introduced [5].

Meanwhile, the NewSpace industry has demonstrated successfully the deployment of Commercial Off-The-Shelf (COTS)-based DHS and processing equipment in space, which offer higher computational performance and lower individual component costs than space-qualified equivalents [6].

One identified issue with the use of COTS equipment is the possible failure propagation to external equipment. To

address this concern, the ESA COTS EEE Working Group has proposed the use of hardware *safety barriers* as a means of limiting any potential failures [7].

By isolating the COTS equipment from other mission-critical components through the use of safety barriers, the risks associated with using non-space qualified components can be mitigated, while still taking advantage of their lower cost and higher performance.

To explore the potential of these concepts in real-world scenarios, we present a case study that focuses on standard form factors for payload processing modules using automotive-grade embedded GPU SoCs.

Moreover, we propose the use of a *Software Safety Barrier* concept that utilizes a separate reliable processor with qualified software and error checking to limit application level error propagation and ensure system availability. In addition, we propose an extension to the ESA hardware safety barrier to manage single event latch-up (SEL) in COTS devices with a highly variable power consumption.

Finally, we apply the aforementioned solutions to three hardware form factors: 6U-ADHA, 3U-ADHA, and Cubesat. Our goal is to provide insights into the potential of using COTS-based systems and software/hardware safety barriers in the design of DHS equipment for future space missions.

II. BACKGROUND

A. Data Processing Units (DPUs) and OBP

Part of the DHS, Data Processing Units are often responsible for pre-processing, data reduction and compression in space missions. In certain cases, a Data Processing Unit (DPU) can be integrated with the Instrument Controller Unit (ICU). Usually, DPUs interface directly with one or multiple instrument front-ends, receiving raw payload data, process it, and transfer it on the Mass-Memory and Formatting Unit (MMFU).

In smaller missions such as micro-satellites or Cubesats, a single payload computer can integrate all the aforementioned functions: instrument interfacing, data processing and mass-memory storage.

We summarise some of the major challenges that DPU designers begin to face today:

- In-flight reconfigurability requirements of On-Board Processing algorithms

- High instrument data rates (ranging from tens of Mbit/s to multiple Gbit/s, depending on the mission size)
- Requirements for high energy efficiency, i.e. high performance within a limited power envelope

B. Advanced Data Handling Architecture (ADHA)

ESA has initiated the ADHA-2 studies (Advanced Data Handling Architecture) together with European Large Space Integrators (LSIs) and DHS equipment suppliers [5].

ADHA specifies the module form-factor and backplane definition. The backplane connector is based on Compact PCI Serial Space (cPCI-S-S), but with a revised pin-out definition, which adds redundancy features typically required in DHS equipment used in ESA missions. The backplane utilises Controller Area Network (CAN), SpaceWire and SpaceFibre interfaces for control and data transfer.

At module (board) level, two form-factors have been defined, equivalent to Eurocard-3U and Eurocard-6U. At unit level, several standard functions have been defined including: On-Board Computer (OBC) module, power module, mass-memory module, and Co-Processing Module (CPM). An ADHA-based DPU or payload computer can be built up by a system controller and multiple CPMs.

C. Use of COTS in ESA Class IV Missions

Recently ESA started considering the use of COTS devices in Class IV missions. This class of missions includes the lower cost missions targeting micro-satellites or constellations. In order to allow bringing innovation to this class of missions, the ESA COTS EEE Working Group is working towards the standardisation of the use of these new COTS devices. As part of this process, it defines categories, guidelines about their testing and also their interactions with space qualified components, ensuring always that a COTS system will *not harm* other subsystems. The outcome of this effort is included in the update of the ECSS-QST-60-13 standard [8].

This effort includes the definition of Q1 and Q2 COTS component classes, for components that lack lot-traceability [9]. This new component classification, creates a framework for the use of NewSpace products based on COTS within institutional missions with limited budgets.

The principles for Q1 can be summarised as follows:

- 1) Use of automotive-grade components when possible
- 2) Perform proton radiation testing on component level for single event effects (SEE) characterisation, focusing on SEL and non-recoverable single event functional interrupt (SEFI)
- 3) Perform TID radiation testing on component or board level

D. Hardware Safety Barrier for COTS Equipment

Due to the rapid development of technology, the performance and cost gap between space qualified components and COTS devices has widened. This creates an interest to use these components in some part of the missions to increase mission return and reduce development costs. This is why ESA has

been working in a set of procedures to allow the use of these components in higher class missions, not only NewSpace ones (Class V). This allows first to use these components in higher class missions (Class IV) and in the future to increase their use in even higher class missions (Class I-III).

This work is carried out by the ESA COTS EEE components and modules Working Group [9] which developed the ESA COTS guidelines in order to define a set of rules to limit the failure propagation. This is achieved by the use of local power regulation in the power buses and external drivers for all interfaces, to limit failure propagation in communication interfaces. To accomplish this, reference designs are proposed for the power delivery to the COTS device and voltage clamps for low speed serial lines.

E. Automotive-grade COTS GPU-SoCs

Among all the hardware options for COTS devices, Automotive Embedded Graphics Processing Units (GPUs) have shown a great potential in high performance and safety features [6]. Driven by the autonomous driving concept, these devices have been designed with high performance in mind as well reliability features in order to meet the automotive safety requirements, defined by the automotive functional safety standard ISO 26262 [10].

However, whether GPU-SoCs can be used to accelerate space algorithms remained an open question until recently. Thanks to the recent ESA-funded activity GPU4S (GPUs for Space) [6] the OBPMark open-source benchmark suite [11] has been developed, which is currently the standard software for testing new payload processing computers at ESA. This set of benchmarks, which supports both sequential and parallel implementations both for multicore CPUs and GPUs, allows us to evaluate the performance benefits of GPUs targeting specific space-related algorithms. In particular, it has been shown that high performance can be achieved in these GPU platforms, with reasonable programming effort [12].

Despite these benefits, in order to use such devices in space, a set of challenges must be overcome like a) module integration in spacecraft systems, b) radiation characterisation, and c) latch-up protection.

From a general integration perspective, these high performance GPU SoCs are available as a System-on-Module. This allows for fast integration because the most challenging parts of the system like DRAM, booting sequencer or power distribution are already solved by the supplier. On the other hand, for space related use, this is a complication, because the auxiliary components cannot be decided by the designer. As a consequence, workarounds and deep study of the module is required to identify all of the possible failure modes and how to isolate the system from the rest of the spacecraft.

Regarding radiation testing, there is already a lot of effort to characterise such systems like E.J. Wyrwas et al. [13] with an NVIDIA TX2, the new generation of this class of embedded GPUs the NVIDIA Xavier [14] or Guertin et al. [15] with the Snapdragon 820 that was been used in the Mars Helicopter Ingenuity [16].

All of these tests have demonstrated really promising reliability data regarding such systems.

Finally, in Section III-C we address the problem of latch-up protection, proposing the use of a state-aware delatcher.

III. PROPOSED SOLUTIONS

In this Section we summarise our proposals for the solution of the aforementioned issues.

The proposed solutions can be summarised as:

- 1) Use of Q1 components.
- 2) Implement hardware safety barriers at module level.
- 3) Extend the hardware safety barrier system.
- 4) Implement software safety barriers at module level.

Figure 1 displays a general view of how the proposed solutions will work for the complete system.

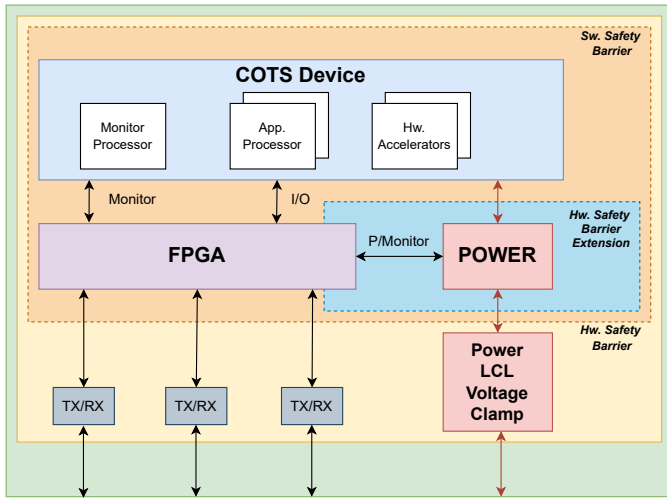


Fig. 1. General solution diagram

A. Use of Q1 Components

As outlined above, we propose to follow the component qualification guidelines of the upcoming Q1 component classification, as described in [9].

In case that proton SEE tests show that key components are susceptible to (destructive) SEL effects, or non-recoverable SEFI effects (such as corruption of boot memory), an iteration of the component selection and radiation testing needs to be performed in order to identify components with similar function and performance.

In rare cases, components with non-destructive SEL can still be utilised – given that the component has an application-enabling performance. In our case this is the computational performance or throughput. It is worth noting that this is entirely driven by the use-case.

B. Leverage Automotive Certifiable HW/SW

As discussed in Section II-E, these COTS embedded computers must include built-in reliability and safety features in order to comply with the ISO 26262 rules. Such systems exhibit

multiple hardware and software components to guarantee that the computer always provide valid and trustworthy outputs.

One of the safety systems included in automotive-grade components such as NVIDIA Xavier GPU SoC family, is the ARM Reliability, Availability, and Serviceability (RAS) feature [17]. RAS is a set of hardware and software diagnostic systems, which focuses on the reliability and error reporting of ARM-based CPU systems. In addition, it includes protocols for communication between the different safety systems with the high-level software, such as the operating system (OS). This system can be used to pinpoint errors in radiation conditions, as well to identify sensitive parts of these complex COTS System on Chip (SoC) [14] devices when they are studied under irradiation. In addition, starting ARM v8.2, RAS will be mandatory for all ARM processors compliant with this version of the standard. Therefore, this feature will be available in more ARM devices and will enable interoperability between different products.

Another feature to leverage is the use of real-time, safety certified processors for multiple parts of the subsystems. For example the more complex automotive SoCs like the NVIDIA Xavier/Orin, feature a set of dual lockstep ARM R5 processors [18] which are used during the boot sequence to guarantee the correctness of the bootloader, as well as other vital parts of the system, like power delivery and main memory.

Other manufacturers, in order to guarantee correctness and real-time operations, implement dual modular redundancy using lockstep processors such as the Aurix TriCore [19] or dynamic modular redundancy like the multicore ARM-78AE [20] which allows to use each core independently for more performance or in lock-step to guarantee correctness.

All these features are useful to ensure that the system can operate correctly under radiation conditions.

C. Hardware Safety Barrier Expansion

As described in the Background Section (II-D) the *hardware safety barrier* allows to add a COTS device and ensure that errors will not cascade to the rest of the spacecraft.

However, in our proposal we are interested into using complex COTS SoCs, featuring embedded GPUs. In these devices, there is high variability in current consumption [21] [22] depending on whether the GPU is used. Therefore, the basic implementation of the delatcher to prevent SEL is not sufficient, since it is unable to differentiate between the start of an execution of a demanding computation or a real latch-up.

Therefore, in order to guarantee the reliable use of these high performance COTS systems, a *state-aware* delatcher needs to be developed. This delatcher needs to have information of the state of the SoC main application processors, accelerators, state of computation and responsiveness of the system. With that information the delatcher will have sufficient information to decide whether an observed current spike is due to a standard task processing or a SEL.

The state-aware delatcher, has two different systems running in parallel, the *SoC state handler* and the *SEL detector*.

The *SoC state handler* is continuously reading from the *Software Safety Barrier*, the low level GPIO signals, the application processor watchdog and the control processor. With all this information, it decides whether the SoC is in any of the following states:

- *off*. The main Application processor and accelerator are off.
- *idle*. The main Application processor and accelerator are powered on but they are in an idle state.
- *computing*. The main Application processor and accelerator are powered on and running applications.
- *booting*. The main Application processor and accelerator are powered on but they are in booting state.
- *crash*. The SoC has suffered a crash, Software or SEE related.
- *unknown*. The SoC state cannot be decided and an intervention is needed to restore the system.

Based on the state information, other *SEL detector* takes the decision to power cycle the board or not, when a current spike takes place. The full flow is shown in the Figure 2.

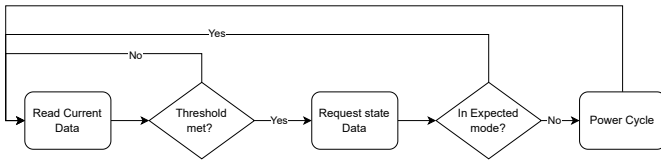


Fig. 2. Operation flow of the smart delatcher

Due to the fast nature of SEL, this full state awareness needs to be implemented in hardware, and be able to communicate with the software safety barrier, described in Section III-D, in order to decide whether the occurring current spike is related to a SEE or to normal computation. It is worth noting that this solution has a small drawback: in order to know fast enough the status of the main SoC, a near real-time communication needs to be established, increasing the base power consumption of the device and using some of the available computation capacity.

D. Software Safety Barrier

As some high-performance COTS processors require the use of closed source proprietary drivers and libraries i.e. in order to use the GPU, and they are only available for Linux – it is not feasible to run an entirely space-qualified software stack on these COTS platforms [23].

Therefore, to enable the use of such COTS processors, similarly to the hardware barrier described above, in this paper we propose to use also a *Software Safety Barrier* to limit possible failure propagation from the COTS processor to external equipment – through software/data/control interfaces.

The software safety barrier introduces a second layer of software within a module or unit, which is external to the COTS processor but embedded in the module/unit, and uses a fully space-qualified software stack. This additional software layer runs in a Fault-Tolerant (FT) CPU.

For control, external equipment will only interact with the qualified software running on the FT-CPU. High-speed data interfaces, which are not feasible to be managed via software without introducing unacceptable bottlenecks, are managed through dedicated hardware IP in FPGA.

The method can be summarised as:

- All control and data interfaces of the COTS processor are connected to an external FPGA or SoC, with known radiation effects and reliability
- The FPGA contains a FT-CPU soft-core (such as NOEL-V or LEON3FT) [24] [25]
- The FT-CPU runs a fully space qualified software stack, such as RTEMS [26]
- The FT-CPU monitors the COTS processor and collects and checks telemetries (TM) and status.
- All Telemetry and Telecommand (TM/TC) to external equipment is filtered through a qualified software node
- All high-speed data interfaces are monitored by dedicated IPs in the FPGA

IV. REFERENCE DESIGNS

Based on our proposals, we present reference diagrams for the integration of such systems in three different form factors for space.

- 1) 6U-ADHA Co-Processing Module
- 2) 3U-ADHA Payload Computer Module
- 3) CubeSat form-factor Payload Computer

A. 6U-ADHA Co-Processing Module

The reference design for the 6U module is based on the NVIDIA AGX Orin and the NVIDIA AGX Orin Industrial. Due to the large size of these devices, this is the only form factor that can host such system. This allows to have a fully qualified industrial module which hosts more high speed I/O, fully automotive qualified components and more low level control signals for the SoM. In Figure3 a block diagram of the proposed reference design is shown.

For this reference design we use the FPGA as intermediary between the COTS device and the rest of the spacecraft. This allows the COTS device to be connected to the spacecraft in a safe manner. Also as shown in the diagram the FPGA will serve as an agent for translating the commercial high speed interfaces like Ethernet or PCIe to the space standards like SpaceFibre and SpaceWire.

This arrangement of components allows the COTS system to be transparent to the OBC, since the FPGA is responsible for the communication and processing of the commands, and some them are forwarded to the COTS device.

B. 3U-ADHA Payload Computer Module

The reference design for the 3U module is based on the NVIDIA NX. The system is similar to the CubeSat design presented next, but uses the ADHA form factor. The use of this module will take up 25% of the available area, leaving enough space for the FPGA, power distribution and conditioning, backplane and frontend connections. Using this module,

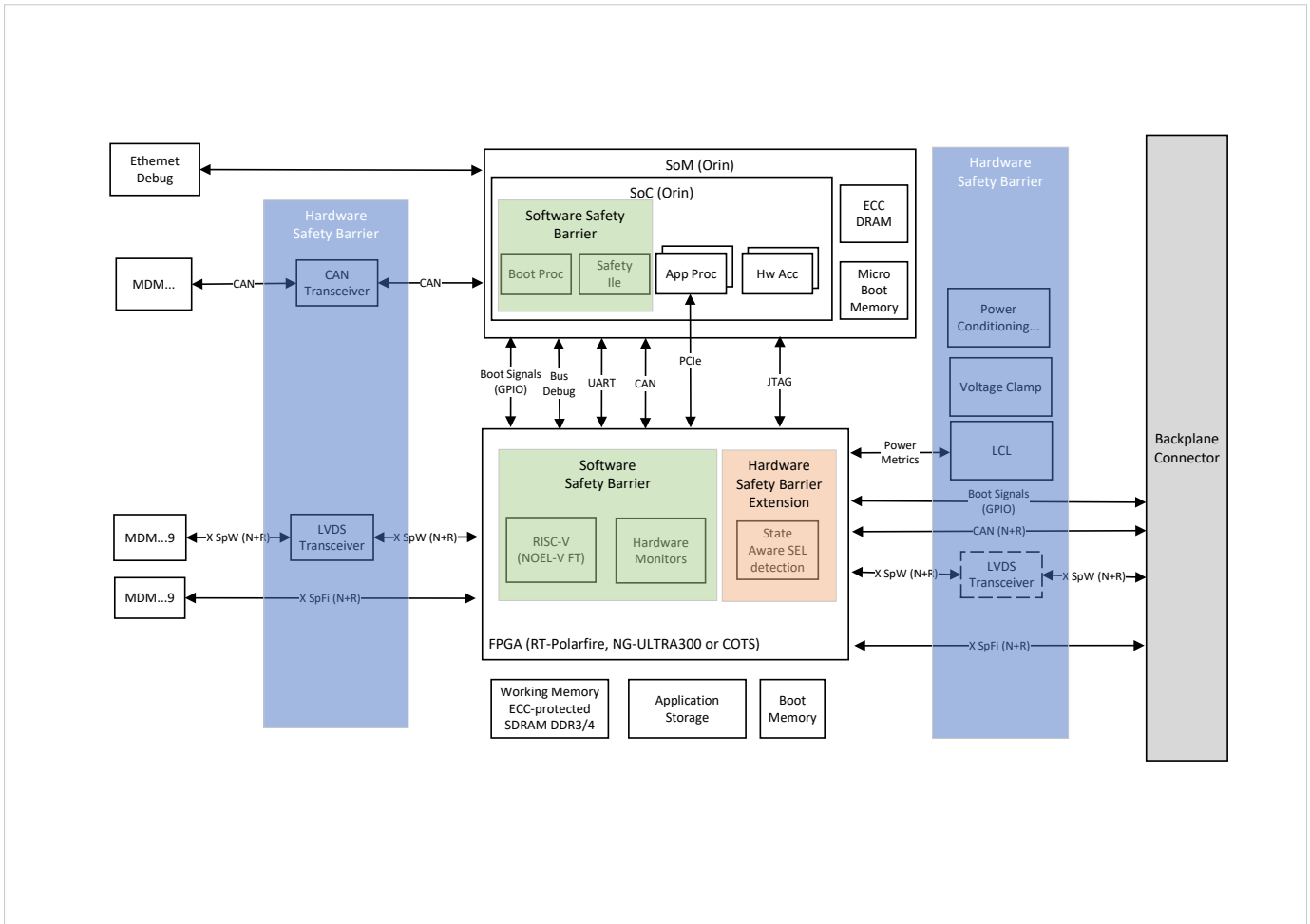


Fig. 3. Reference design architecture for the 6U-ADHA module

we can get high performance with a smaller form factor, sacrificing some of the I/O and low level control signals of the SOM. Moreover, while the NVIDIA NX can be ruggedized, it does not fulfil all the automotive grade qualifications, unlike the larger form factor NVIDIA Orin AGX and Industrial. In Figure 4 the reference block diagram can be seen.

This design shares a lot of commonalities with the 6U-ADHA design. In particular, some of the front-end control signals are removed, as well as the JTAG interface since it is not available on the NVIDIA NX device. This also means that both systems can have similar software and FPGA hardware design, with the only change to be in the I/O.

C. CubeSat form-factor Payload Computer

The smallest reference design which targets CubeSats is based on the NVIDIA Orin NX. Again it uses the NX form factor from NVIDIA, which allows to have the same hardware and software interface with other NX modules like Orin Nano, as well as with previous models of the same SoC family like the Xavier NX and other compute modules compatible with this form factor like the Mixtile Core 3588E [27].

Widely used in the NewSpace industry, CubeSats only standardise the outer dimensions of the satellite in order to be compatible with multiple launchers. However, the internal systems like the onboard computer (OBC) or power systems use different interconnections or form factors due to the lack of standardisation. For this reason, in our reference design we choose to follow the *PC/104 space standard*, which is widely used in industry and uses two 52 pins connectors to deliver power and control signals between the different modules.

This PC/104 connector has a limitation in contrast to the compact PCI connector on the ADHA design, which is that the connector is not able to carry high speed low voltage differential signalling (LVDS), like PCI express or gigabit Ethernet. For this reason, in our reference design we took inspiration from the PCIe/104 specification [28] to use the express connector to carry LVDS signals and between the different modules. The proposed reference block diagram is shown in Figure 5.

As it can be seen on the block diagram, the control signals (CAN and I2C) of the module are connected to the PC/104 as well some low level control signals to handle the power of the module. This is similar to the ADHA reference design,

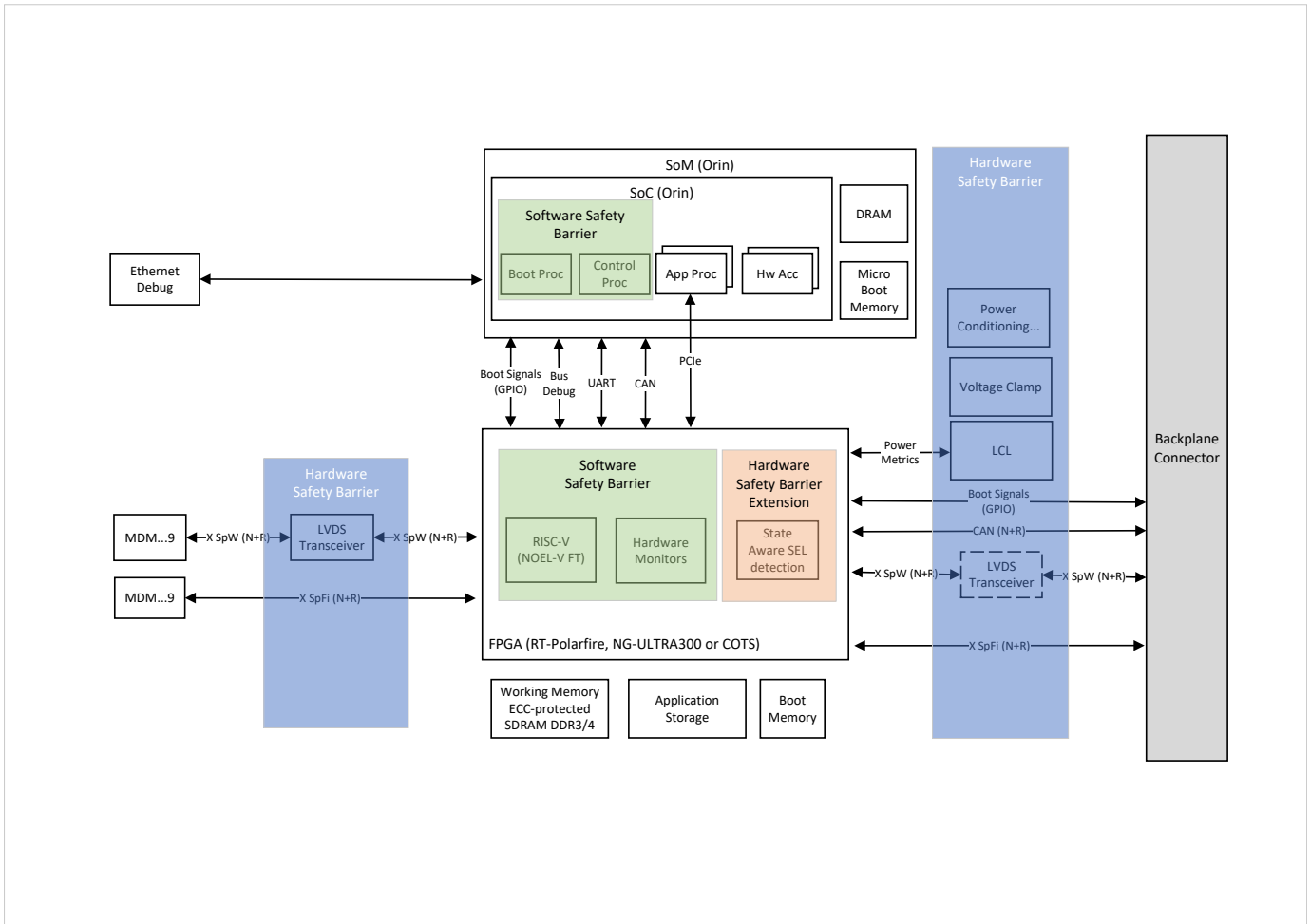


Fig. 4. Reference design architecture for the 3U-ADHA module

in which only the CAN connection is exposed and protected with the hardware safety barrier. However, in contrast to the previous designs, the rest of the connections goes without that control, which is a trade off of the cubesat design due to the limit space of the PC/104 form factor.

Finally, it is also worth mentioning that because the PCIe lanes are going over the connector, this functionality can be upgraded in the future with more modules attached to the PCIe/104 connector.

V. ON-BOARD SOFTWARE ECOSYSTEM

In addition to the hardware reference designs described in the previous Section, we have put significant effort in terms of implementing end-user application software for on-board data processing - primarily focusing on payload data, since the targeted use case is for DPUs.

The baseline code used for the on-board payload data processing tasks comes from the reference, parallel software implementation of OBPMark and OBPMark-ML benchmarks reference in OpenMP and CUDA [29], in order to exploit the multicore and GPU features of the embedded COTS GPU device.

For optical imagers the following payload processing stages are available:

- 1) Image pre-processing corrections and calibrations (flat-field correction, bad pixel correction, radiation scrubbing, stacking, binning, etc) [30] [29].
- 2) Cloud screening, using Convolutional Neural Network methods [29].
- 3) Object detection, e.g. ship detection [29].
- 4) Image and data compression: CCSDS 121, CCSDS 122 [31] [29]. Support for CCSDS 123 is on-going.
- 5) Data encryption [29].

In addition, for synthetic aperture radar (SAR) instruments – image formation (i.e. range-Doppler algorithm) has been also implemented in parallel software [32] within the OBPMark benchmark suite.

As all implementations mentioned above are available as parallelised implementations both in OpenMP and CUDA – it is possible to utilize both the embedded CPU multicore and the embedded GPUs at the same time (e.g. for different stages of a processing pipeline) in order to ensure the utilization of the available computational elements.

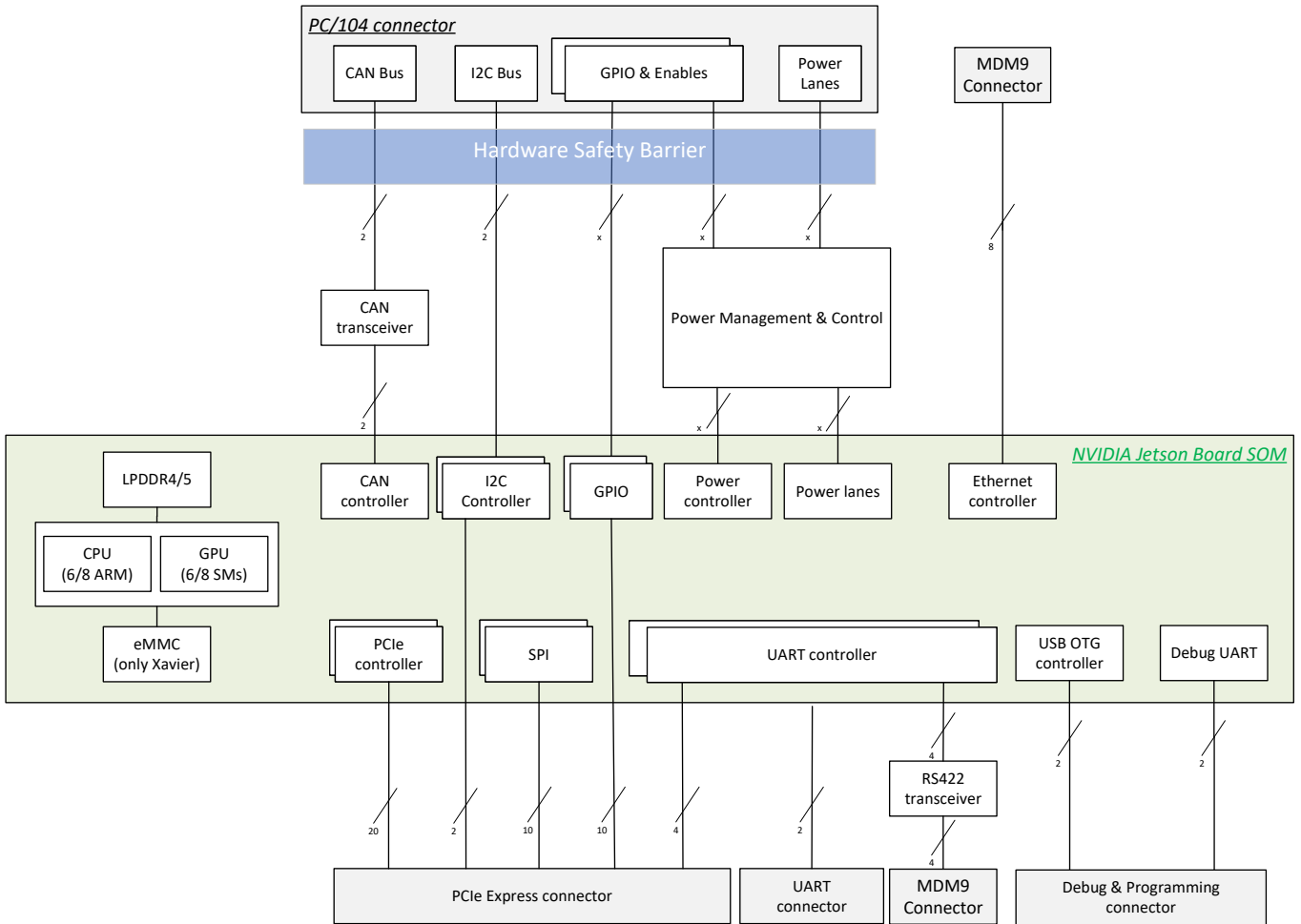


Fig. 5. Reference design architecture for Cubesat form factor payload computer.

In addition, the devices include several hardware accelerators, such as: AI/ML processing for quantized neural networks, JPEG image compression and H.264/H.265 video encoding.

If higher performance is required for tasks that are not easily parallelisable, such as e.g. some compression or encryption algorithms, additional hardware accelerator IPs can also be added to the on-board FPGA. For example, we have observed that the encoding stages of CCSDS compression algorithms are more efficiently implemented in FPGAs in terms of performance per power. Hence, the pre-processing stage (e.g. predictor in CCSDS 121/123 or the discrete wavelet transform in CCSDS 122) [33] can be easily implemented in the embedded GPUs, which can be combined with the encoding stage implemented in the FPGA.

VI. CONCLUSIONS

In this paper we have outlined methodologies for the use of complex COTS System-on-Chip in reliable data handling systems, focusing on on-board payload processing applications.

Finally, we have presented the new concept of using a *Software Safety Barrier* in complex SoC with embedded GPUs

as a possible solution for mitigating error propagation on the software and data handling interfacing level and an extension of the ESA COTS EEE Hardware Safety Barrier to be used in state-of-the-art complex COTS embedded GPUs.

We have presented three different hardware/software conceptual designs for payload computers on standard form-factors: 6U-ADHA module, 3U-ADHA module and Cubesat. Finally, we have presented an overview of the software ecosystem in such a design.

ACKNOWLEDGEMENTS

This work was supported by ESA through the 4000136514/21/NL/GLC/my co-funded PhD activity "Mixed Software/Hardware-based Fault-tolerance Techniques for Complex COTS System-on-Chip in Radiation Environments" and the GPU4S (GPU for Space) ESA-funded project.

Moreover, it was supported by the European Community's Horizon Europe programme under the METASAT project (grant agreement 101082622). In addition, it was partially supported by the Spanish Ministry of Economy and Competitiveness under grants PID2019-107255GB-C21 and IJC-2020-045931-I (Spanish State Research

Agency / Agencia Española de Investigación (AEI) / <http://dx.doi.org/10.13039/501100011033>) and by the Department of Research and Universities of the Government of Catalonia with a grant to the CAOS Research Group (Code: 2021 SGR 00637). Leonidas Kosmidis was partially supported by the Spanish Ministry of Economy and Competitiveness under grant IJC2020-045931-I (Spanish State Research Agency / <http://dx.doi.org/10.13039/501100011033>).

REFERENCES

- [1] G. Furano, G. Meoni, A. Dunne, D. Moloney, V. Ferlet-Cavrois, A. Tavoularis, J. Byrne, L. Buckley, M. Psarakis, K.-O. Voss *et al.*, "Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 44–56, 2020.
- [2] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. Van Dijk, M. Esposito, J. Hefele, N. Verduyssen *et al.*, "The Φ -Sat-1 Mission: The First On-board Deep Neural Network Demonstrator for Satellite Earth Observation," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- [3] F. Ouallouche, K. Labadi, Y. Mohia, M. Lazri, and S. Ameur, "Artificial Intelligence for Satellite Image Processing: Application to Rainfall Estimation," in *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*. Springer, 2023, pp. 165–174.
- [4] F. Ortiz, V. Monzon Baeza, L. M. Garcés-Socarras, J. A. Vázquez-Peralvo, J. L. Gonzalez, G. Fontanesi, E. Lagunas, J. Querol, and S. Chatzinotas, "Onboard Processing in Satellite Communications Using AI Accelerators," *Aerospace*, vol. 10, no. 2, p. 101, 2023.
- [5] O. Mourra, "ADHA 2022 Workshop (Advanced Data Handling Architecture)," nov 2022. [Online]. Available: <https://indico.esa.int/event/427/>
- [6] L. Kosmidis, I. Rodriguez, A. Jover-Alvarez, S. Alcaide, J. Lachaize, O. Notebaert, A. Certain, and D. Steenari, "GPU4S: Major Project Outcomes, Lessons Learnt and Way Forward," in *Design Automation and Test in Europe Conference (DATE)*, 2021.
- [7] M. Nikulainen and F. Tonicello, "Utilization of COTS in ESA Missions," NASA Electronic Parts and Packaging (NEPP) Electronics Technology Workshop, 2021, accessed: March 10, 2023. [Online]. Available: https://nepp.nasa.gov/workshops/etw2021/talks/17-JUN-21_Thur/1045_Nikulainen_Tonicello-Utilisation-of-COTS-in-ESA-Missions.pdf
- [8] European Cooperation for Space Standardization (ECSS), "ECSS-Q-ST-60-13C Rev.1 – Commercial electrical, electronic and electromechanical (EEE) components," European Space Agency, ECSS Standard ECSS-Q-ST-60-13C Rev.1, 2022, accessed: March 10, 2023. [Online]. Available: <https://ecss.nl/standard/ecss-q-st-60-13c-rev-1-commercial-electrical-electronic-and-electromechanical-eee-components-12-may-2022/>
- [9] F. Tonicello, "ESA Position And Activities Related to COTS Usage in Space," in *ACCEDE 2022 Workshop on COTS Components for Space Applications*, oct 2022.
- [10] International Organization for Standardization, *ISO/DIS 26262. Road Vehicles – Functional Safety*, 2018.
- [11] D. Steenari, L. Kosmidis, I. Rodriguez-Ferrandez, A. Jover-Alvarez, and K. Forster, "OBPMark (On-Board Processing Benchmarks)-Open Source Computational Performance Benchmarks for Space Applications," in *European Workshop on On-Board Data Processing (OBDP)*, 2021.
- [12] L. Kosmidis, I. Rodriguez, A. Jover-Alvarez, S. Alcaide, J. Lachaize, O. Notebaert, A. Certain, and D. Steenari, "GPU4S: Major Project Outcomes, Lessons Learnt and Way Forward," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 1314–1319.
- [13] E. Wyrwas, "Proton Testing of nVidia Jetson TX2," jun 2019. [Online]. Available: https://nepp.nasa.gov/files/30370/NEPP-TR-2019-Wyrwas-NEPPweb-TR-19-021_InVidia-Jetson-TX2-2019June02-TN72754.pdf
- [14] I. Rodriguez-Ferrandez, M. Tali, L. Kosmidis, M. Rovituro, and D. Steenari, "Sources of Single Event Effects in the NVIDIA Xavier SoC Family under Proton Irradiation," in *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2022, pp. 1–7.
- [15] S. M. Guertin and M. Cui, "SEE Test Results for the Snapdragon 820," in *2017 IEEE Radiation Effects Data Workshop (REDW)*. IEEE, 2017, pp. 1–6.
- [16] B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, and D. Zhu, "Mars Helicopter Technology Demonstrator," in *AIAA Atmospheric Flight Mechanics Conference*, 2018.
- [17] ARM. (2020) Arm Architecture Reference Manual Supplement Reliability, Availability, and Serviceability (RAS), for Armv8-A. [Online]. Available: <https://developer.arm.com/documentation/ddi0587/latest>
- [18] —, *Cortex R-5*, mar 2018. [Online]. Available: <https://www.arm.com/products/silicon-ip-cpu/cortex-r/cortex-r5>
- [19] A. Hayek and J. Börcsök, "Safety Chips in Light of the Standard IEC 61508: Survey and Analysis," in *2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*. IEEE, 2014.
- [20] ARM, *Cortex-A78AE*, mar 2022. [Online]. Available: <https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a78ae>
- [21] E. Aragon, J. M. Jiménez, A. Maghazeh, J. Rasmusson, and U. D. Bordoloi, "Pattern Matching in OpenCL: GPU vs CPU Energy Consumption on two Mobile Chipsets," in *Proceedings of the International Workshop on OpenCL 2013 & 2014*, 2014, pp. 1–7.
- [22] I. Rodríguez Ferrandez, "An On-board Algorithm Implementation on an Embedded GPU: A Space Case Study," Master's thesis, Universitat Politècnica de Catalunya, 2021, <https://upcommons.upc.edu/handle/2117/344892>.
- [23] L. Kosmidis, M. Solé Bonet, I. Rodriguez-Ferrández, J. Wolf, and M. M. Trompouki, "The METASAT Hardware Platform: A High-Performance Multicore, AI SIMD and GPU RISC-V Platform for On-board Processing," in *European Data Handling and Data Processing Conference for Space (EDHPC)*, 2023.
- [24] Frontgrade Gaisler, *NOEL-V Processor*. [Online]. Available: <https://www.gaisler.com/index.php/products/processors/noel-v>
- [25] —, *Leon3 Processor*. [Online]. Available: <https://www.gaisler.com/index.php/products/processors/leon3>
- [26] J. Seronie-Vivien and C. Cantenot, "RTEMS Operating System Qualification," in *Data Systems in Aerospace (DASIA)*, vol. 602, 2005.
- [27] mixtile, "Mixtile core 3588e," 2023, accessed: August 21, 2023. [Online]. Available: <https://www.mixtile.com/core-3588e/>
- [28] PC/104 Consortium, "PCIe/104," 2021, accessed: March 10, 2023. [Online]. Available: <https://pc104.org/hardware-specifications/pcie104/>
- [29] D. Steenari, L. Kosmidis, I. Rodriguez, S. Muret, M. Bargholz, D. M. Tali, and L. Mansilla, "OBPMark and OBPMark-ML - Computational Benchmarks for On-Board Data Processing and Machine Learning," in *European Data Handling and Data Processing Conference for Space (EDHPC)*, 2023.
- [30] I. Rodriguez, L. Kosmidis, M. M. Trompouki, F. J. Cazorla, and D. Steenari, "Evaluating the Computational Capabilities of Embedded Multicore and GPU Platforms for On-Board Image Processing," in *European Data Handling and Data Processing Conference for Space (EDHPC)*, 2023.
- [31] A. Jover-Alvarez, I. Rodríguez, L. Kosmidis, and D. Steenari, "Space Compression Algorithms Acceleration on Embedded Multi-Core and GPU Platforms," *Ada Lett.*, vol. 42, no. 1, dec 2022.
- [32] M. Solé, I. Rodriguez-Ferrandez, D. Steenari, and L. Kosmidis, "Acceleration of Synthetic Aperture Radar for On-board Space Systems," *IEEE High Performance Extreme Computing (HPEC)*, 2023.
- [33] Y. Barrios, A. J. Sánchez, L. Santos, and R. Sarmiento, "SHyLoC 2.0: A Versatile Hardware Solution for On-board Data and Hyperspectral Image Compression on Future Space Missions," *IEEE Access*, vol. 8, pp. 54 269–54 287, 2020.