CEAS Space Journal (2020) 12:551–564
https://doi.org/10.1007/s12567-020-00321-9

ORIGINAL PAPER

Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space

Fredrik C. Bruhn[1,2,4] · **Nandinbaatar Tsog**[2] · **Fabian Kunkel**[1] · **Oskar Flordal**[1] · **Ian Troxel**[3]

Received: 19 November 2019 / Revised: 31 May 2020 / Accepted: 3 June 2020 / Published online: 15 June 2020

Abstract
The last decade has seen a dramatic increase in small satellite missions for commercial, public, and government intelligence applications. Given the rapid commercialization of constellation-driven services in Earth Observation, situational domain awareness, communications including machine-to-machine interface, exploration etc., small satellites represent an enabling technology for a large growth market generating truly Big Data. Examples of modern sensors that can generate very large amounts of data are optical sensing, hyperspectral, Synthetic Aperture Radar (SAR), and Infrared imaging. Traditional handling and downloading of Big Data from space requires a large onboard mass storage and high bandwidth downlink with a trend towards optical links. Many missions and applications can benefit significantly from onboard cloud computing similarly to Earth-based cloud services. Hence, enabling space systems to provide near real-time data and enable low latency distribution of critical and time sensitive information to users. In addition, the downlink capability can be more effectively utilized by applying more onboard processing to reduce the data and create high value information products. This paper discusses current implementations and roadmap for leveraging high performance computing tools and methods on small satellites with radiation tolerant hardware. This includes runtime analysis with benchmarks of convolutional neural networks and matrix multiplications using industry standard tools (e.g., TensorFlow and PlaidML). In addition, a ½ CubeSat volume unit (0.5U) ($10 \times 10 \times 5$ cm^3) cloud computing solution, called SpaceCloud™ iX5100 based on AMD 28 nm APU technology is presented as an example of heterogeneous computer solution. An evaluation of the AMD 14 nm Ryzen APU is presented as a candidate for future advanced onboard processing for space vehicles.

Keywords OBDP · Machine learning · GPU · Small satellites · Heterogeneous computing

✉ Fredrik C. Bruhn
f@bruhnspace.com; fredrikc.bruhn@unibap.com

Nandinbaatar Tsog
nandinbaatar.tsog@mdh.se

Fabian Kunkel
Fabian.kunkel@unibap.com

Oskar Flordal
oskar.flordal@unibap.com

Ian Troxel
ian@troxelaerospace.com

[1] Unibap AB (Publ.), Kungsängsgatan 12, 753 22 Uppsala, Sweden

[2] Mälardalen University, Box 883, 721 23 Västerås, Sweden

[3] Troxel Aerospace Industries Inc., 2023 NE 55th Blvd., Gainesville, FL, USA

[4] Bruhnspace AB, Rapphönsvägen 7B, 756 53 Uppsala, Sweden

1 Introduction

There are numerous studies and argumentation for increased onboard autonomy and data information processing to provide more efficient use of the relatively limited communication link bandwidth on small satellites [1–3]. Expanding on the needs of intelligent processing, it is especially relevant to study the rapidly evolving field Earth Observation driven by advances in sensor technologies. ESA's Φ-lab at the ESRIN facility has led several workshops in the context of artificial intelligence (AI) for Earth Observation (AI4EO) and written a European AI research agenda [4]. The agenda identifies a range of challenges and opportunities for ensuring European pooling of resources, talent supply, digital environment for rapid prototyping, and development of solutions to capture the opportunities. The landscape formed around the transformative AI technology is today dominated by United States and China. ESA have formulated several candidate

projects in the mid-technology readiness level (TRL) range within the General Study Technology Programme (GSTP) Element 1 "Develop" AI 2019 compendium [5]. These candidate projects are of strategic importance to current and future space systems and space exploration and cover both data exploitation and operations. The proposed developments are categorized in these areas:

- Smart payload data
- AI in data exploitation
- AI in operations
- Guidance, navigation, and control
- Edge/onboard AI

This paper covers architectural and software aspects of Edge/onboard AI and Smart Payload Data but uses toolchains common with cloud architectures on ground and hence AI in data exploitation and operations. The presented architecture can also be applied to guidance, navigation, and control (GNC). The commonality with hardware and software development environments on ground is important to simplify deployment of AI in space systems. It is furthermore important for cost and resource sharing reasons, where existing code from industry or consumer business can be reused and a wider access to talent is possible.

To make a difference in the information market it is important to provide an infrastructure and ecosystem that is generic while still offering specialization at the same time in order to minimize the size, weight, and power (SWaP) for small satellites. This is especially important, since small satellites is driving many new products and services [6].

The authors have explored edge computing and especially onboard AI data processing since 2013, leading up to a scalable radiation tolerant heterogeneous architecture first implemented using AMD 1st generation (28 nm) G-series System-on-Chip (SOC) paired with MicroSemi FPGA on an Input/output (IO) expanded industrial Qseven form factor board [6]. AMD denotes their SOCs as accelerated processing units (APUs). This paper expands on the previous work to include a full heterogeneous computer architecture also for AMD 2nd generation (28 nm) G-series SOC, AMD R-series (28 nm) SOC, and the latest AMD V1000 Series (14 nm) SOC [7, 9].

2 Related work

Due to increasing demands of onboard sensor and autonomous processing, research has long focused on high performance and reliability. The adoption of graphical processing units (GPU) in space is emerging rapidly due to the necessity of handling massive data in-orbit or in deep space. One example of a CubeSat with heterogeneous architecture is the NASA Hyperspectral Thermal Imaging (HYT) mission being integrated by University of Hawaii [10].

Processing capabilities on CubeSat has been limited due to available SWaP and novel computer architectures have been explored like hybrid and reconfigurable computing. George and Wilson present an overview of different architectures, methods, and alternatives for onboard space computing in an overview paper [8]. The authors also describe the radiation effects that are shared between all space computers including the presented architecture in this paper. The reconfigurable computing part is defined in a field programmable gate array (FPGA) while hybrid computing is synonym with heterogenous computing, i.e., the combination of CPU + GPU, CPU + FPGA on the same chip or board. Fault tolerant computing is needed for space computers due to the radiation background effects and uses a combination of techniques also common with the presented architecture. These include information redundancy exemplified by error detection and correction coding (EDAC), error correcting codes (ECC), cyclic redundancy check (CRC), algorithm-based fault tolerance (ABFT), and parity checking. Checkpoint and exception handling are prominent examples of software redundancy.

Adams et al. of University of Georgia have investigated a similar approach of hybrid processing as the authors with a combination of Nvidia Tegra TX2i and Microsemi SmartFusion2 [9]. It shares similar features with the architecture presented in this paper, including the physical form factor of PC/104, stacking connector, and standard protocols. However, there is a big difference in radiation performance behaviour between the Nvidia TX2 and the AMD SOC, which is further discussed below. Very similarly, Adams et al. use the SmartFusion2 as a trusted control node and watchdog of the larger CPU + GPU SOC. In the heterogeneous architecture the FPGA use is a bit expanded as it has redundant communication paths to the SOC and can have isolated hybrid compute tasks separate from the watchdog functionality as further described below.

ESA has investigated GPU for space applications through analysis of different low-end and high-end GPUs from radiation and power consumption perspectives in the GPU4Space project [10].

NASA has conducted several studies from a radiation perspective on different GPUs including from both Nvidia and AMD [11, 12, 14]. Notable, Salazar et al. have conducted radiation testing on five COTS graphic cards, of which two AMD GPUs and three Nvidia GPUs, aiming for application on the International Space Station (ISS) in low earth orbit (LEO) radiation environment [14]. Top three among five GPUs were chosen to test under the total dose of 6 krad. However, 6 krad is very low and ISS is not a representative environment for most missions. An expanded description of radiation effects is discussed in Sect. 4. None of the cards

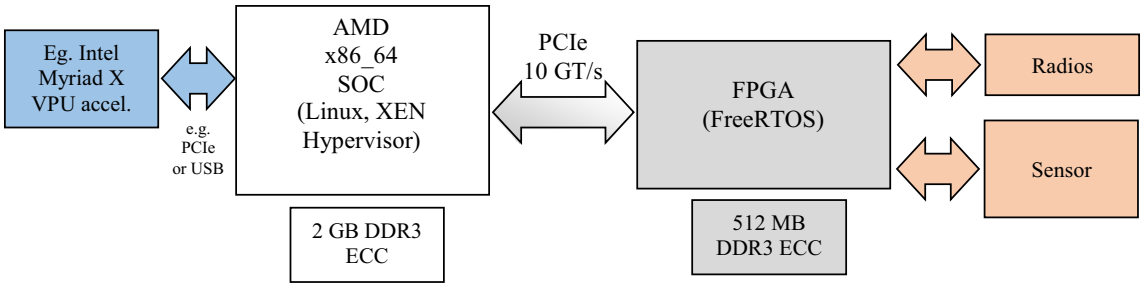

Fig. 1 Illustration of the heterogeneous compute architecture as implemented on the Qseven industrial form factor compute board

failed in a permanent failure, all the cards have several failures, i.e., functional interrupts which are required a reboot or power cycle to get the control again. MSI HD6450 employed with AMD's GPU has performed the best and recorded 43.1 days of MTTFI (the mean time to functional interrupt).

An important rationale for the use of GPUs in space is the energy efficiency for a computing task. Kosmidis et al. and Tsog et al. have shown that GPUs can have a significantly higher power efficiency compared to CPU for the same computation [13, 14].

2.1 Heterogeneous computing architecture overview

Building on the initial Qseven standard derived heterogeneous design described in refence 7, Unibap AB and Troxel Aerospace Industries, Inc have coordinated to develop a next-generation onboard heterogeneous/hybrid computing platform for intelligent processing, e.g., Big Data analytics and Artificial Intelligence (AI) processing to address the need of onboard data processing using the AMD V1000 Series 14 nm embedded family of SOCs and the Microchip PolarFire FPGA.

The initial compute architecture laid the foundation to the presented × 86 embedded computer using SOC/APUs from AMD. The SOC devices are from the FT3/FT3b footprint compatible 1st and 2nd generation G-series SOCs featuring multi-core 64-bit CPU cores and integrated Graphical Processing Unit (GPU). and paired with a Microchip/Microsemi SmartFusion2 FPGA, which includes an ARM Cortex M3 Microcontroller and high-speed IO. The industrial standard Qseven interfaces are supported together with a wide range of IO expanded through the FPGA. Figure 1 illustrates the heterogeneous/hybrid architecture combining × 86 SOC, ARM-based FPGA and optionally additional accelerators (e.g., Intel Movidius Myriad ASICs).

Figure 1 illustrates the initial heterogenous architecture as described above. From a raw theoretical performance view, the AMD G-series SOCs have up to 87 GFLOPS GPU FP32, single precision performance. Common space interfaces such as SpaceWire, SpaceFibre and RapidIO can be supported through the FPGA or external circuits. The data rate limitation in the heterogeneous SOC-FPGA link is 10 Giga Transfer per second (GT/s) (bidirectional) over 2 lanes PCIexpress generation 2. The DDR3 memory support Error Correction Code (ECC) on both the AMD SOC and the FPGA and operate at 1066 or 1333 MHz on the AMD and 667 MHz on the FPGA. To simplify integration of new functions in the FPGA, Unibap developed a custom Direct Memory Architecture (DMA) for the heterogeneous computing architecture interaction between the AMD SOC and the FPGA over PCIexpress. Theoretically using two lanes of PCIe, an actual real data flow of 8 Gigabit/s (Gbps) is theoretically possible without the protocol overhead. Unibap has demonstrated a sustained heterogeneous bandwidth of 5.7 Gbps (i.e., 720 MB/s) using DMA over the PCIe interface.

The heterogeneous PCIe link between the AMD SOC and the FPGA is used in the NASA HYTI mission by integrating a DMA Camera Link sensor interface and providing DMA interfaces to S- and X-band radios [10].

For the purpose of demonstrating a real implementation of the architecture, a Qseven compute solution in a ½ CubeSat volume unit (0.5U) ($10 \times 10 \times 5$ cm^3) called SpaceCloud™ iX5 is presented as an example of a heterogeneous computing solution suitable for spaceflight that provides advanced onboard processing for space systems.

2.2 High performance computing tools in space

The AMD V1000 Series SOC and AMD R-Series SOCs advances the concept of heterogeneous computing by integrating hardware features for rapid IO memory translation (IOMMU) and instructions from Heterogeneous System Architecture (HSA) standard led by HSA Foundation [15].

Supporting HSA has significant benefits to the compute architecture as the V1000 and R-Series can be made to leverage AMD's high-performance computing (HPC) software stack called Radeon Open Compute (ROCm) [16]. A particularly interesting aspect of the ROCm stack is that is can convert and execute Nvidia CUDA code and hence provide an avenue for radiation tolerant execution of CUDA code. This is also of interest, since large algorithm investments

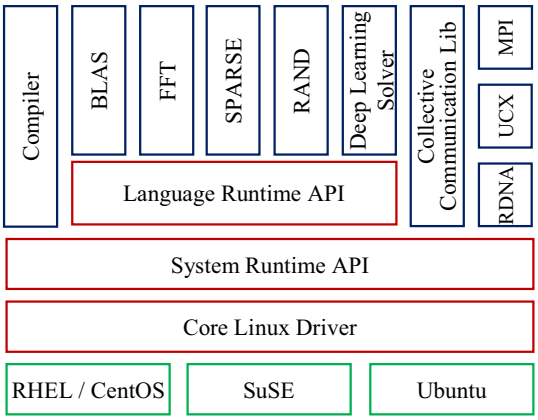

Fig. 2 Overview of the ROCm HPC software stack

have been made in the CUDA framework and ROCm offer an avenue to leverage these investments on an open source platform.

ROCm is an open source high performance computing platform for GPU accelerated platforms. The open source aspect of the ROCm stack is important from a software radiation hardening perspective as it allows for injection of time and executing state monitoring in the code, as well as review and modifications of choice. The main development of ROCm is done by AMD and currently it is targeting mainly Linux operating systems. As illustrated in Fig. 2, ROCm provides a complete software stack, from the Linux kernel driver, to compiler support and common libraries and software for machine learning. The core Linux driver of ROCm is accepted upstream in the Linux kernel. Currently, the latest release of ROCm is version 3.3.0 which have upstream support in Linux kernel 4.15 and 5.3 respectively for R-series and V-series. ROCm supports important features for heterogeneous computing, including:

- multi-GPU coarse-grain shared virtual memory,
- process concurrency and pre-emption,
- large memory allocations,
- HSA signals and atomics,
- user-mode queues and DMA,
- standardized loader and code-object format,
- dynamic and offline-compilation support,
- peer-to-peer multi-GPU operation with RDMA support,
- profiler trace and event-collection API,
- systems-management API and tools

ROCm is the first HSA-compliant HPC software stack and is designed to allow other hardware vendors to adopt and develop their drivers to extend the ROCm ecosystem. The aim of HSA generally is to decrease the development complexity of applications on heterogeneous processing units (e.g., CPU, GPU, FPGA, etc.) for developers. Moreover, it allows to handle coherent shared memory through the entire heterogeneous processing units. For example, by allowing this, developers do not need to care about the different memory structures of CPU and GPU. Furthermore, the processing units see data in coherent shared memory in the same way. It reduces the mechanical data copying process between the memories of different processing units.

Comparing the 1st and 2nd generation G-series SOCs to the V1000 series reveal a significant performance uplift in performance, partially due to HSA but mostly because of a new manufacturing process and new CPU + GPU architecture. The performance uplift and use of ROCm HPC software stack is demonstrated with benchmarks in this paper. Overall GPU compute performance of the V1000 family in 16 bit (half) floating point (FP16) is up to 3.7 TFLOPs and the SOC support up to 8 CPU threads execution using simultaneous multithreading (SMT) on quad × 86 CPU cores from the AMD ZEN microarchitecture.

3 Stacking interface for modularity and form factor

The SpaceCloud™ iX5 is modularized by providing a core compute board and a common stacking interface based on the Samtec LSHM-150-04.0-L-V-A-S-K-TR connector [17]. The physical outline form factor of the printed circuit board (PCB) is aligned to the Pumpkin PCB Specification [18] with all PC-104 related connectors removed and replaced. Figure 3 shows a photograph of the Unibap e2160 heterogeneous compute module and the iX5 CORE carrier board on the left. On the right the stacking connectors are highlighted. The system is designed to operate on 12 V DC voltage and fit within a 0.5 U (10 × 10 × 5 cm) volume.

Figure 4 shows a photograph of the iX5 compute module, CORE carrier board and EXTENSION board stacked together.

The signal partitioning and capabilities in the stacking connectors on the iX5 CORE module are defined in Table 1.

4 Single-event effect mitigation middleware (SMM)

A brief discussion of radiation effects is required to understand the value of Troxel Aerospace's single-event effect mitigation middleware (SMM) and its relevance to enabling the use of COTS processors in space applications. Several types of radiation effects have the potential to damage or create incorrect operating conditions in electronics, e.g., processors, while operating in a space environment. Total ionizing dose (TID) can be thought of as a build-up of absorbed radiation over time that changes the

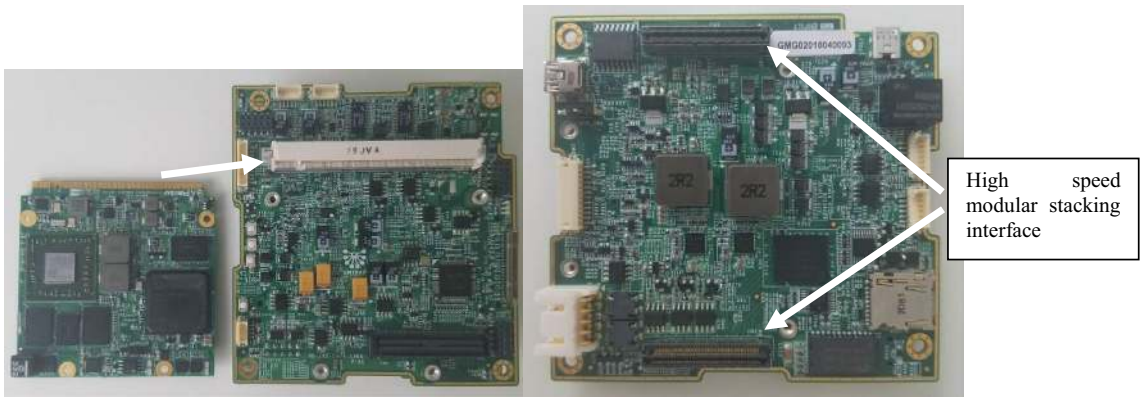

Fig. 3 Photograph of SpaceCloud™ iX5 core components. Left, IO expanded Qseven compatible compute core (Unibap e2160) with CORE-1000 carrier board. Right: Photograph of SpaceCloud™ iX5 CORE module with high speed stacking interface for expansion

Fig. 4 Photograph of SpaceCloud™ iX5 CORE and EXTENSION module stacked with compute module attached at the top

electrical characteristics of transistors. A transistor's ability to effectively switch without increasing leakage current degrades as dose increases to a point, where the transistor will either no longer switch and/or becomes stuck in a closed or open state. TID effects are a combination of numerous particles strikes and/or gamma irradiation over time. Other effects occur based on a single particle strike and are generally grouped into the category of single-event effects (SEEs). Within this category of radiation effects, single-event latchup (SEL) is a destructive event, whereby a single particle, typically a heavy ion, concentrates enough charge within a transistor to cause a charge path between two of the three contact points of the transistor causing an un-designed current to flow between them. If the current flow is sufficiently large or flows in an inappropriate direction, the transistor suffers permanent damage such that it becomes a current short or in some other way no longer functions properly. Other types of SEE, such as single-event upsets (SEUs) and single-event functional interrupts (SEFIs), are non-destructive events caused by a single particle (typically a proton, neutron, or heavy ion) that either causes a memory bit to "flip", i.e., change from 1 to 0 or 0 to 1, or cause the device to enter an incorrect state of operations, respectively.

Radiation hardened processors (rad-hard processors) are designed with various techniques at the basic silicon transistor layer to provide some level of immunity to the radiation effects previously described. Typical TID immunity levels exceed 100 krad up to over 1 Mrad and SEL immunity is typically above 75 MeVcm^2/mg (Si). Non-destructive SEEs are designed to be so rare in these devices that they typically occur only once in 20 years. Rad-hard processors such as the BAE RAD750 [19], BAE RAD5545 [20], Cobham/Geisler LEON3FT [21], and Moog Broad Reach BRE440 [22], form the basis of many satellite control systems that require a high degree of radiation effects immunity, especially large/expensive spacecraft, human-rated vehicles, and exoplanetary missions like the Mars rovers. However, there is a large performance price paid for such radiation immunity with rad-hard processors being typically tens to hundreds of times less capable in processor performance compared to modern COTS processors [23]. If chosen carefully and validated through extensive radiation testing, COTS processors can be selected that have favourable destructive radiation effect characteristics—indeed, the AMD processors mentioned in this paper have been shown to have favourable TID and SEL characteristics [24]. However, all COTS processors exhibit high rates of non-destructive SEEs compared to rad-hard processors and thus typically require frequent rebooting to mitigate these effects. The frequency of time between reboot vary greatly based on the underlying technology and the radiation environment in which the processor is operating. In benign environments such as the International Space Station, the time between reboot can be weeks to months while in more stringent environments such as polar orbits, GEO stationary, MEO, or HEO orbits, or exoplanetary missions, SEFI rates, and time between reboots, can be multiple per day. For many missions, particularly

Table 1 Summary of the electrical interfaces in the high-speed expansion stacking interface (level 1 and level 2)

Interface	LSHM-150-06.0-F-DV-S-K-TR (100 pin, height 12 mm)			
module (level)	DD-iX5 CORE (level 1)		DD-iX5 EXTENSION (level 2)	
	Signal definition	Device	Signal definition	Device
CORE Module Stacking Connector Extension	16×LVDS pair @ 700 Mbps	FPGA	2×I^2C	FPGA
	CAN v2.0b	FPGA	2×USB v2.0	AMD SOC
	2×I^2C	FPGA	SPI	FPGA
	SPI	FPGA	I^2C	AMD SOC
	2×SERDES (10 Gbps)	FPGA		
	12×GPIO 3.3 V	FPGA		
	PCIexpress×1 (5 GT/s)	AMD SOC		
CORE Module Stacking Connector Base	3.3 V	DC	3.3 V	DC
	3.3 V on/off	FPGA	5 V	DC
	5 V	DC	5 V on/off	FPGA
	5 V on/off	FPGA	12 V	DC
	12 V	DC	12 V on/off	FPGA
	12 V on/off	FPGA	Reset	FPGA
	Reset	FPGA	12×GPIO	FPGA
	6×GPIO 3.3 V	FPGA	PCIexpress×1 (5 GT/s)	AMD SOC
	2×SATA v3	AMD SOC	PCIexpress×4 (20 GT/s)	AMD SOC
	PCIexpress×4 (20 GT/s)	AMD SOC		
	2×USB v3	AMD SOC		
	2×USB v2	AMD SOC		
	I^2C	AMD SOC		
	Ground			

expensive or human-rated ones mentioned previously, where rad-hard processors are typically used, such reboot rates, and moreover, any reboot at all is unacceptable. Additionally, the observed SEFI susceptibility of COTS processors has been increasing as their feature sizes have decreased (i.e., moving from 32 to 28 nm to 14 nm, etc.) reducing times between reboot for a given mission.

To overcome this limitation and provide a means to make COTS processors viable for space applications that require both improved processing capability and reduced radiation susceptibility (i.e., less frequent reboots) Troxel Aerospace developed an SEE Mitigation Middleware that greatly improves non-destructive SEE upset rates. Troxel Aerospace's SEE Mitigation Middleware (SMM) provides core-, device-, and system-level fault tolerance by implementing multicore checking in the background in Linux. This robust middleware for heterogeneous multicore processors provides resource-aware configuration and execution management, and fault detection and mitigation. The SMM is designed to operate as either a background "scrubbing" task or as an interactive fault correction mechanism directed by missions software. The middleware software layer primarily resides between the application layer and the Operating System (OS), with extensions into and below the OS, to provide intelligent resource, fault, and power management. The middleware provides a consistent computing environment and application programming interface (API) for fault management that allows mission software to be largely agnostic to the specific underlying hardware, thereby reducing development and integration cost, complexity, and schedule.

A functional block diagram illustrating where the middleware resides within a multicore processor software stack is shown in Fig. 5. The SMM provides an abstraction layer on which mission software, be it command and data handling

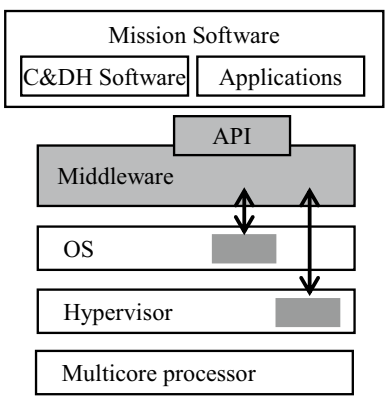

Fig. 5 Proposed middleware software architecture

(C&DH) software or applications, execute to increase portability and fault tolerance. The SMM is largely processor-agnostic and supports multiple processor architectures with core management functions fully portable across processor and Linux variants. A relatively small portion of the middleware is required to be OS- and processor-specific to support resource and fault status collection, and to execute commands to manage resources, deploy applications, and mitigate faults through processor-specific interfaces and technology. The technology-agnostic central management features of the SMM communicate to the technology-specific components (i.e., OS and hypervisor kernel extensions) through standard interfaces allowing the design to be common across processor and OS architectures.

The SMM has been deployed on a homogenous quad-core ARM processor, the Unibap e2000 and e2100 family featuring AMD 1st Gen SOC fm. "eKabini" and 2nd Gen SOC fm. "Steppe Eagle" Series GPU, the Unibap V1000 series AMD, and a Digital Signal Processor (DSP), demonstrating the middleware's flexibility across platforms. Through the completion of a NASA JPL SBIR Phase II program, the SMM implementations on the Steppe Eagle and DSP were irradiated with 32 h of heavy ions using Texas A&M's Cyclotron in November and December 2019 and demonstrated SEU (bit-flip) and SEFI immunity for all error events observed demonstrating a 720× increase in non-destructive SEE susceptibility. The 720× increase takes a conservative approach by assuming that the next test observation would have resulted in an uncorrectable error, which is possible but very unlikely. Even so, this is a dramatic increase in upset rate. As mentioned in the radiation discussion above, this improvement provides a varied benefit depending on the mission orbit. To provide two examples, if the processor would otherwise suffer a SEFI (reboot) every 3 days, i.e., a relatively harsh mission, the system would instead suffer a SEFI once every 5.9 years with Troxel Aerospace's SMM enabled. In another mission scenario, if the processor would suffer a SEFI every 30 days, i.e., a moderately harsh mission, the system would instead suffer a SEFI once every 59 years with the SMM enabled. These results demonstrate a substantial improvement in SEFI rate and would make these processors viable for a wide range of otherwise inappropriate missions such as autonomous operations, docking, exoplanetary landings, and other missions described in Sect. 5.

5 Mission scenarios and application

The latest available NASA crosscutting technology roadmap lists key avionics goals to include improved reliability and fault tolerance, increased autonomy, reduced size, weight, and power (SWaP), and commonality across spaceflight and ground processing systems [25]. Long-duration crewed missions, space-based observatories, and solar system exploration will require highly reliable, fault-tolerant systems. Communication delays, the challenging orbital dynamics of Near-Earth Asteroids (NEAs), and extreme science missions require increased autonomy for on-board decision infrastructures [26]. Future robotic missions will involve greater complexity and reactivity, which will require increased reliance on autonomy (i.e., advanced onboard processing). Deep-space missions that target active, dynamic, or time-varying phenomena will need robots that can adaptively adjust their configurations and behaviour to changing circumstances, and robustly handle uncertainty. Robotic missions to NEAs will require the decision-making and monitoring processes—currently performed by ground control—to be performed by onboard autonomous systems [27]. Advanced avionics technologies and approaches are needed to support these challenging missions.

Subsection TA11.1.1 of the Chief Technologists Office Technology Roadmap lists the three areas of flight computing that are critical to next-generation needs for science and exploration to include processors, memory, and high-performance flight software [25]. Scalable, multicore processors, co-processors, and memory that have a range of capabilities for fault tolerance and recovery are needed for use in radiation fields to support an increasingly software-intensive onboard environment. Flight software, called on to perform a range of functions, including increasing autonomy, will require techniques for state-based design and verification techniques to manage complexity at design time and ensure reliability and safety in operations. Historically, flight computing has focused on tight-loop operations.

Onboard experiments with intelligent onboard processing on CubeSats took a significant step forward in 2013 when the IPEX CubeSat was launched as a secondary payload. IPEX validated a range on board instrument data-processing algorithms and autonomy [28, 29].

ESA's Earth Observation directorate have been pushing AI for small satellites through the Φ-Sat-1 satellites. The Φ-Sat-1 mission was formulated in response to an ESA challenge and consists of two 6U CubeSats. The mission will demonstrate on-orbit image filtering using AI of hyperspectral images [30]. This mission represent the comprehensive approach ESA is taking to identify and deploy AI on space mission as discussed in the introduction [4, 5]. Varile et al. have explored Convolutional Neural Networks (CNN) for autonomous image analysis [31].

Future trends show generalization toward varied requirements for flight computing, including hard real-time, mission-critical calculations that often involve vision-based algorithms such as those for entry, descent, and landing; high-data-rate instrument throughput imperatives, such as those for hyper-spectral and synthetic aperture radar; and the increasing use of model-based reasoning techniques like

those for mission planning and fault management. Future flight computing systems must provide heterogeneous architectural support across this spectrum of computational drivers, including uncertainty, distribution, concurrency, and operations. As more capable science instruments observe and capture larger volumes of data, there is a need to develop methods for data reduction and triage at the point of collection. The introduction of intelligent machine-learning algorithms onboard is a critical technology area that is important for helping to address the entire end-to-end observing path in data-driven environments. Furthermore, the need to respond to and update observation plans is a critical part of moving towards more autonomous operations. This paradigm shift will require new onboard capabilities as demands for computation, storage, and software continue to grow to enable more autonomous operations coupled with onboard data services.

Additionally, new paradigms for fleet management and sustainment, such as the Digital Twin, which are enabling to extended autonomous operations, amplify the need for robust onboard computing [32]. Pinpoint landing, hazard avoidance, rendezvous-and-capture, and surface mobility are directly tied to the availability of high-performance space-based computing. In addition, multicore architectures have significant potential to implement scalable computing, thereby lowering spacecraft vehicle mass and power by reducing the number of dedicated systems needed to implement onboard functions. These requirements are equally important to space science and human exploration missions. In addition, power-efficient, high-performance, radiation-tolerant processors and the peripheral electronics required to implement functional systems could also benefit commercial aerospace entities and other governmental agencies that require high-capability spaceflight systems. Advances in middleware to support cooperative processing in combining high-performance multicore general-purpose processors (GPPs) and niche co-processors, such as the robust middleware proposed by Troxel Aerospace, and heterogeneous computing architectures by Unibap, is required to achieve planned mission performance requirements.

5.1 Applications

There are many mission's scenarios and applications, where massive onboard processing is critical as discussed earlier in the paper. Some mission are prime candidates for advanced onboard computing, including the following types:

- Autonomous rendezvous and docking
- Quick react, low latency science observations, where human time scales are not enough to react
- Exo-planetary avionics and science missions, where message latency is too long
- Downlink bandwidth limited missions (high rate sensors), where intelligent data reduction is required

An example of a bandwidth limited mission that leveraging onboard radiation tolerant heterogenous ×86 computing is the NASA Hyperspectral Thermal Imaging, HyTI mission, due for launch in 2021 [10]. The HyTI mission is a 6U CubeSat that will demonstrate spectral thermal imaging from Low Earth Orbit (LEO) orbit with onboard science data product generation.

6 Software overview

The heterogeneous architecture allows for software portioning over different compute nodes in the heterogenous architecture (i.e., multi-core CPU, GPU, and the FPGA in this case). It is possible to extend the heterogenous architecture with more compute nodes using the available peripherals such as PCIe or USB, e.g., Intel Myriad X Vision Processing Units (VPU) with 3 TOPS as illustrated in the figure.

For the purpose of benchmarking and demonstrating AI software in this paper, the software configuration listed in Table 1 was used with either CPU support or both CPU and GPU support. The tools clpeak [33] and mixbench [34] was used to verify the GPU performance of 87 GFLOP for the AMD G-series SOC and 2 TFLOPs for the AMD V1605B SOC from the V1000 family.

It is important to note that the AMD HPC software stack ROCm is not possible to run on SOC/APUs after version 1.7 without modification. The official APU support has been removed from the packages. Hence, it was needed to recompile the entire stack to enable support for the AMD embedded series of devices. Bruhnspace corporation and Mälardalen University performed the ROCm patching and Bruhnspace provide an experimental software build online [35] while Unibap has patched the latest ROCm v3.3.0.

To illustrate the use of the "hipify" tool we convert a simple squaring CUDA code and execute it.

This example uses a simple squaring example, square.cu[1] to demonstrate the simplicity of using CUDA on AMD ROCm. However, it should be noted that "hipify" cannot parse CUDA assembler which need to be manually converted to AMD GPU assembler.

$ hipify-perl **square.cu**>square.cpp // ROCm "Hipify" Nivida CUDA example code to generic cpp code.

$ hipcc square.cpp -o square_hip // Compile the cpp code with AMD "hip compiler".

[1] https://raw.githubusercontent.com/ROCm-Developer-Tools/HIP/master/samples/0_Intro/square/square.cu.

$./square_hip // and finally run it on ROCm stack for AMD APU devices.

info: running Square CUDA example on device **AMD Ryzen Embedded V1605B with Radeon Vega GFX.**

info: allocate host mem (7.63 MB) info: allocate device mem (7.63 MB).

info: copy Host2Device info: launch 'vector_square' kernel info: copy Device2Host.

info: check result PASSED!

7 Intelligent data processing performance evaluation

To demonstrate the next-generation intelligent processing capabilities of the heterogeneous compute platform, six experiments have been conducted in this paper and executed on AMD A10-8700P (codename "Carrizo") R-series SOC and AMD V1605B part of the V1000 family of SOCs using the ROCm HPC stack (v2.6.0 and v3.3.0).

7.1 Evaluation environment

The experiments are performed on two refence platforms featuring V-Series V1605B and A10-8700P APUs from AMD. V1605B APU includes gfx902 (Vega) GPU with 1.1 GHz (15 W TDP setting) clock rate and Ryzen CPU with 2 GHz (15 W TDP setting) clock rate [36]. A10-8700P APU consists of Excavator CPU and gfx801 GPU that is employed in Acer E15 E5-552-T99R model notebook [37]. The clock rates of CPU and GPU in A10-8700P APU are 1.67 GHz and 0.8 GHz, respectively. The software used are defined in Table 2.

7.2 Experimental design

Artificial intelligence (AI) enabled applications are one of the concepts that should be employed for intelligent onboard data processing. TensorFlow[2] is explored as machine learning platform/framework in the experiments. Using TensorFlow, matrix multiplication has been performed for Experiment A on both CPU and GPU with the different sizes of the arrays. TensorFlow is an open source machine learning platform involving tensor computations. Matrix multiplication is the fundamental of neural network, hence, we selected it in this experiment. The aim of this experiment is to discuss how the platform gains computing performance using GPU for the advanced parallel algorithms compared to CPU. Furthermore, this experiment indicates the performance of Tensorflow framework. Necessary parameters for Experiment A are described in Table 3

Then, in Experiment B, we consider the optimal implementations of matrix multiplication provided by vendors (ROCm) as well as a well-known library (BLAS[3]) to evaluate the performance capabilities of the platforms. We use a code written in C++ for HIP compiler for GPU computing and sgemm from BLAS for CPU cores. For comparison reason, we use the program that used in Experiment A as well (Table 4).

In Experiment C, we evaluate the inference performance when running two different convolutional neural network (CNN) across both CPU and GPU on the v1605b and the A10-8700P. The networks are from the TensorFlow object detection model zoo which are good candidates for transfer learning when running detection networks for earth observation on a satellite. A resolution of 512×512 is used, where multiple overlapping images can be used to cover the typical large sensor sizes seen on satellites.

In Experiment D we benchmark the compute throughput and bandwidth of ROCm 3.3.0 running on AMD V1605B.

In Experiment E we evaluate the possibility to do training on the platform. There are cases when it is impractical to get data to ground and where online learning can be done on self-supervised data such as anomaly detection on sensor readouts. Included is an experiment, where we train a simple Long short-term memory (LSTM) autoencoder on a time series anomaly detection dataset. Given the algorithmic advancement in where classification workloads have reduced in FLOP count by $2\times$ every 16 months [38] workloads that is efficient to train on ground today is likely to become easier to train efficiently in orbit during the platforms lifetime.

In Experiment F we test the Intel Movidius Myriad X as neural network accelerator that can be used to offload calculations from the platform. Networks run in FP16 precision compiled through the Intel OpenVINO framework.[4] Given the drop in precision and separate implementations, a slightly different result is given for the numbers quoted for the Myriad X is on a different network.

7.3 Results

Experiment A Tables 5, 6, and 7 present the processing time of matrix multiplication on CPU and GPU with respect to edge size of matrices in the reference machines V1605B with TensorFlow 1.14.1, V1605B with TensorFlow 2.0.0, and A10-8700P with TensorFlow 2.0.0, respectively. Tables include minimum, maximum, average and median values of

[2] https://www.tensorflow.org/.

[3] BLAS – Basic Linear Algebra Subprograms https://www.netlib.org/blas/.

[4] https://docs.openvinotoolkit.org/.

the processing time. For the comparison study of CPU and GPU, we focus on median values of the processing time. We confirm that the processing time on CPU increases rapidly, while edge size of matrix increases. Under edge size of 100, the usage of CPU could be better than the usage of GPU. On the other hand, we see that the processing time (median) on GPU is better than CPU when the edge size is more than 200.

Furthermore, we can see about 3.1 times improvements between the processing time for the edge size of 5000 on CPU, as it is 1633.21–1639.05 ms in V1605B and 5077.05 ms in A10-8700P. This result explains that the next generation platform employs a much powerful CPU based on AMD ZEN architecture. In the case of the GPU, we can see 2 times improvements between the GPUs employed in V1605B (about 704 ms for the edge size of 5000) and A10-8700P (1464.71 ms). In addition, we do not confirm big difference between the different versions of TensorFlow used for the experiments in V1000.

Experiment B Table 8 presents the comparison study of processing times of the different implementations of matrix multiplication. TF-GPU and TF-CPU describe a matrix multiplication code using TensorFlow 2.0.0 on GPU and CPU, respectively. HIP means a code provided in ROCm software stack and is implemented for GPU using HIP compiler. BLAS describes a matrix multiplication code for CPU computation using BLAS library. We consider both HIP and BLAS as optimized codes, since they are provided by vendors or a well-known benchmarking library. TF is our target framework in this paper, and we evaluate it by conducting the comparison study with the optimized codes. Naïve is a naïve implementation of matrix multiplication for CPU written in C. As a note, both HIP and BLAS are written in C/C++, and TF-CPU and TF-GPU are written in Python.

Matrices with edge sizes larger than 128, we see that TF-GPU performs better than HIP for both reference machines. Moreover, TF-CPU on A10-8700P leads BLAS on A10-8700P as well. Only TF-CPU on V1605B performs less compared to BLAS on V1605B. Since BLAS on V1605B leads BLAS on A10-8700P, it can be concluded that TensorFlow 2.0.0 is not optimized well for Ryzen CPU in V1605B. Although GPU in A10-8700P has less performance capability compared to GPU in V1605B, HIP on A10-8700P performs better than HIP on V1605B. This could be explained that the optimization of ROCm for A10-8700P is better than V1605B, since A10-8700P is one of the oldest platforms started with the ROCm development. In other words, there are room for more improvement in ROCm for gfx902 (Vega) GPU. The results of matrices with edge sizes smaller than 64 are less informative. This is, because, the different programming languages use, and their time measurement methods are slightly different. Hence, we can explain the overhead time influences on the results a lot in these cases. As a conclusion of this experiment, we can emphasize the optimization of TensorFlow fits well with our reference machines.

Experiment C Two pre-trained models, Model A and Model B, are considered in this experiment. Model A is a mobilenet with ssd[5]and Model B is a resnet50 with faster rcnn.[6] The experiment is run on randomized data across 256 images split into 16 batches and on 32-bit floating point (single precision). The APU is set to do automatic thermal management (to the threshold 12 W or 15 W TDP) to get a balanced overall system performance. Adding additional priority to the GPU can yield faster inference as indicated in the GPU high column but with CPU clocks dropping to 400 MHz and the total power increasing to maximum TDP (Table 9).

Experiment D Figure 6 show that the maximum throughput of 2.2 TFLOP is reach at 9 GB/s bandwidth and 2 TFLOP throughput at 16.5 GB/s bandwidth is measured using mixbench (HIP, alt mode) for the AMD V1605B embedded APU in single memory configuration. The V1605b support dual memory configuration.

Experiment E The training benchmark is run using a 128 LSTM run on a single dimensional temperature dataset. While this is limited in scope this mirrors the usefulness of, e.g., monitoring sensors on board and finding out when adjustments needs to be done to various instruments. This training is compared to a typical server as found on the ground, in this case a 24 core AMD ThreadRipper with a Nvidia 2080 RTX GPU. Given the limited size of the network this perform similarly on a server class cpu and gpu and the difference to the embedded platform for this type of workload is smaller making the case for training directly on the V1605B platform stronger.

Model	V1605B		Ground-based server	
	CPU	GPU	CPU	GPU
Time series anomaly detection (steps/s)	10	50	175	280

Experiment F The Myriad X experiment is run over USB 3.0 on a Mobilenet SSD (depth multiplier 1.0) with only 1 output category and 16-bit floating point. The network has a fixed input resolution of 480×384.

Using both execution slots and queuing up 32 jobs the average rate that can be processed is 29,8 fps

[5] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.mdssd_mobilenet_v1_coco.

[6] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.mdfaster_rcnn_resnet50_coco.

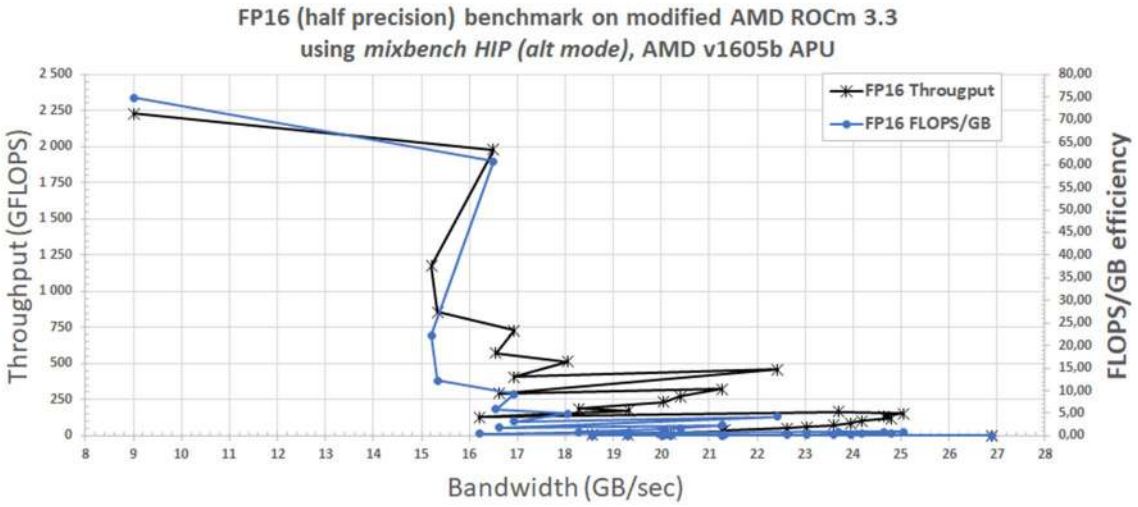

Fig. 6 Throughput vs bandwidth benchmark of ROCm v3.3.0 on AMD V1605b using mixbench

Table 2 Verified software AMD G-series SOC, AMD R-series AMD V1000

Software name	G-series	R-Series/V-series	V-series
(L)Ubuntu Operating system	18.04.4 6 AMD64	18.04.4 AMD64	18.04.4 AMD64
Linux kernel	5.4.28	5.0.0	5.4.28
AMD gpu kernel driver	amdgpu	amdgpu	Amdgpu
AMD IOMMU driver	–	IOMMU2	IOMMU2
AMD HSA driver	amdkfd	amdkfd	Amdkfd
AMD ECC memory kernel driver	AMD64 EDAC	–	–
Unibap DMA kernel driver	1.0	–	–
GCC	7.2	8.1	7.2
cmake	3.11	3.11	3.16
LLVM	10.0.0	6.0.0	11.0-git
Mesa, patched by Unibap	20.1-devel	18.2	19.2
Libclc, patched by Unibap	2020–02-22	–	–
ROCm, patched by Unibap	–	2.6.0	3.3.0
OpenCL	1.2	2.0	2.0
OpenGL	4.6	4.6	4.6
Vulcan	1.2	1.2	1.2
Theano	1.0.0	–	–
Caffe	1.0	–	–
OpenCV	3.3.1	4.1.1	4.1.1
Robot Operating System (ROS)	1.12.13 (Kinetic)	–	–
TensorFlow	1.4	1.14.1/2.0	1.15.2/2.2
pyTorch			1.6a
PlaidML		0.6.4	0.6.4
Clpeak [19]	2019–09-05	2019–09-05	2019–09-05
Mixbench (HIP) [20]			2020–05-19

8 Conclusions

A radiation tolerant CubeSat compatible onboard information processing architecture have been prototyped and evaluated. Evaluation of AMD 28 nm and 14 nm embedded products with multicore CPU and GPU have shown significant benefits in acceleration of radiation tolerant and potentially radiation hardened compute tasks.

Table 3 Parameters for Experiment A

Parameters	Values
Edge size of matrix	10, 20, 50, 100, 200, 500, 1000, 2000, 5000
Experiments number	100 times

Table 4 Parameters for Experiment B

Parameters	Values
Edge size of matrix	8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096
Experiments number	10 times

AMDs high performance computing software stack ROCm have been patched to enable embedded devices and shown to support machine learning software like TensorFlow and execution of CUDA code in a radiation tolerant/hardened silicon.

The experiments show that the theoretical compute throughput is reached in benchmarks but real applications using TensorFlow can be further optimized.

It has been shown that deep learning training can efficiently be performed in orbit and that neural networks tuned for Earth observation applications can be used for near real-time onboard information processing and onboard training.

The heterogeneous architecture is tested by expanded the AMD SOC with an Intel Movidius Myriad X neural accelerator which can significantly increase the AI processing speeds but at lower bit resolution.

Table 5 Processing time on both CPU and GPU in V1605B (TF 1.14.1) with respect to edge size of matrix

Edge size	On CPU (ms)				On GPU (ms)			
	Min	Max	Average	Median	Min	Max	Average	Median
10	0.12	0.20	0.15	0.15	0.39	1.77	0.63	0.58
20	0.12	0.16	0.13	0.12	0.39	1.89	0.66	0.58
50	0.13	1.74	0.24	0.19	0.36	3.82	0.58	0.50
100	0.21	0.92	0.31	0.27	0.35	3.17	0.78	0.68
200	2.43	10.64	4.62	4.05	0.32	8.72	0.64	0.53
500	30.71	49.57	36.23	35.57	1.38	32.31	2.10	1.77
1000	199.05	278.22	213.13	211.69	6.30	142.80	8.29	6.98
2000	96.27	125.78	103.78	104.17	34.82	541.33	46.17	41.36
5000	1566.19	1691.06	1631.44	1633.21	639.78	3804.77	733.65	704.74

Table 6 Processing time on both CPU and GPU in V1605B (TF 2.0.0) with respect to edge size of matrix

Edge size	On CPU (ms)				On GPU (ms)			
	Min	Max	AVERAGE	Median	Min	Max	Average	Median
10	0.13	0.33	0.16	0.16	0.37	2.56	0.77	0.65
20	0.13	0.63	0.24	0.21	0.38	2.35	0.61	0.55
50	0.12	0.21	0.13	0.13	0.40	3.54	0.62	0.56
100	0.23	1.56	0.45	0.39	0.36	3.64	0.54	0.47
200	2.90	14.41	6.49	6.65	0.36	8.93	0.65	0.51
500	17.89	73.60	45.14	45.27	1.14	32.49	1.71	1.31
1000	15.60	342.25	259.27	273.90	5.85	139.83	7.86	6.46
2000	99.04	2075.00	487.88	107.33	35.27	525.53	46.10	41.39
5000	1573.95	7671.68	1696.15	1639.05	653.73	3839.39	733.60	704.28

Table 7 Processing time on both CPU and GPU in A10-8700P (TF 2.0.0) with respect to edge size of matrix

Edge size	On CPU (ms)				On GPU (ms)			
	Min	Max	Average	Median	Min	Max	Average	Median
10	0.29	0.96	0.48	0.44	0.99	2.36	1.27	1.17
20	0.37	1.15	0.46	0.42	0.92	2.44	1.29	1.22
50	0.41	0.89	0.50	0.46	0.85	2.31	1.20	1.11
100	0.45	1.15	0.63	0.56	0.78	1.69	1.04	0.99
200	1.32	2.69	1.63	1.56	0.68	2.18	1.12	1.07
500	6.46	9.92	7.61	7.48	1.86	5.69	2.68	2.66
1000	37.06	41.46	38.80	38.75	11.32	27.72	13.48	13.39
2000	273.60	322.30	283.77	282.65	85.70	124.89	90.36	89.88
5000	4336.91	5259.34	4948.74	5077.05	1458.28	1770.54	1472.15	1464.71

Table 8 Comparison of processing times on both CPU and GPU in V1605B and A10-8700P

Edge size	Processing time on V1605B (ms)					Processing time on A10-8700P (ms)				
	GPU		CPU			GPU		CPU		
	TF-GPU	HIP	TF-CPU	BLAS	Naïve	TF-GPU	HIP	TF-CPU	BLAS	Naïve
8	0.88	3.257	0.3	0.025	0.004	0.94	0.16	0.46	0.034	0.004
16	1.06	2.935	0.33	0.027	0.035	0.96	0.161	0.53	0.049	0.031
32	0.87	2.262	0.36	0.042	0.27	1.25	0.178	0.48	0.077	0.247
64	0.94	3.238	0.45	0.133	1.957	1.06	0.298	0.5	0.188	1.562
128	0.91	4.175	1.09	0.564	14.967	1.05	1.23	0.92	1.007	17.425
256	0.96	13.307	6.36	2.909	120.50	1.29	2.815	2.26	4.428	116.73
512	2.82	72.74	39.34	21.141	1066.4	2.7	12.926	7.7	19.411	3899.4
1024	15.72	314.85	306.98	110.22	70.457 s	12.58	56.747	42.87	126.08	45.277 s
2048	100.2	1491.3	2109.1	574.4	364.58 s	87.22	319.16	303.18	1034.3	396.45 s
4096	774.51	6738.6	1742.0	4692.9	-	607.42	2100.3	2560.7	8286.7	3339.6 s

Table 9 Comparison results of Model A and Model B

Model	V1605B			A10-8700P	
	CPU	GPU	GPU high priority mode	CPU	GPU
A. Mobilenet SSD (512×512 pixel images per second)	7.7	6.2	5.3	8.3	5.4
B. Resnet50 faster rcnn (512×512 pixel images per second)	0.28	0.28	0.44	0.20	N/A

Acknowledgements Open access funding provided by Mälardalen University. This work was sponsored in part by the Swedish National Space Agency under contracts 272/16 and 267/18. We thank AMD corporation for hardware donations and specifically Mr. Mazda Sabony of AMD for driver support.

References

1. Richardson G., et al.: Small satellite trends 2009–2013, 29th Annual AIAA/USU Conference on small satellites, 2015, Logan, Utah, USA.
2. https://www.linkedin.com/pulse/why-did-google-dump-skybox-tom-segert/. Accessed 14 Nov 2019
3. Hicks, M.T., Niederstrasser C.: Small Sat at 30: trends, patterns, and discoveries, 30th Annual AIAA/USU conference on small satellites, 2016.
4. Mathieu, P.-P., Loekkenetal, S.: Towards a European AI4EO Research and Innovation Agenda. ESA Φ-lab workshop proceedings, 28 September 2018, Issue 1, Rev 3, Towards a European AI4EO R&I Agenda.

5. GSTP Element 1 "Develop" compendium 2019, ESA-TECT-PL-015679, 28th of October 2019, Issue 1.http://emits.sso.esa.int/emits-doc/ESTEC/News/GSTPAICompedium2019.pdf
6. Behrens, J.R., Lal, B.: Exploring trends in the global small satellite ecosystem. New Space **7**(3), 126–136 (2019). https://doi.org/10.1089/space.2018.0017
7. Bruhn, F., et al.: Introducing radiation tolerant heterogeneous computers for small satellites. 2015 IEEE Aerospace Conference, Big Sky, MT, 2015, pp. 1–10. doi: 10.1109/AERO.2015.7119158
8. AMD V1000 Series SOC. https://www.amd.com/en/products/embedded-ryzen-v1000-series. Accessed 3 Nov 2019
9. Microchip PolarFire FPGA, https://www.microsemi.com/product-directory/fpgas/3854-polarfire-fpgas. Accessed 4 Nov 2019
10. Wright, R. et al.: HYTI: thermal hyperspectral imaging from a CubeSat platform, 29th Annual AIAA/USU Conference on Small Satellites 2019, Logan, Utah, USA. Associated presentation, https://digitalcommons.usu.edu/cgi/viewcontent.cgi?filename=0&article=4369&context=smallsat&type=additional. Accessed 18 Nov 2019
11. George, A.D., Wilson, C.M.: Onboard processing with hybrid and reconfigurable computing on small satellites. Proc. IEEE **106**(3), 458–470 (2018). https://doi.org/10.1109/JPROC.2018.2802438
12. Adams, C., Spain, A., Parker, J., Hevert, M., Roach, J., Cotten, D.: Towards an integrated GPU accelerated SoC as a flight computer for small satellites. 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1–7. doi: 10.1109/AERO.2019.8741765
13. Kosmidis, L., Lachaize, J., Abella, J., Notebaert, O., Cazorla, F.J., Steenari, D.: GPU4S: Embedded GPUs in Space, 2019 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 2019, pp. 399–405, doi: 10.1109/DSD.2019.00064.
14. George A.S., Glen. F.S.: Commercial off-the-shelf (COTS) graphics processing board (GPB) radiation test evaluation report, Johnson Space Center, Houston, Texas, December 2013. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140004427.pdf
15. Wyrwas, E., LaBel, K. A., Campola M., O'Bryan, M.: Guidance on Standardizing GPU Test Approaches, 2018 IEEE Radiation Effects Data Workshop (REDW), Waikoloa Village, HI, 2018, pp. 1–4. doi: 10.1109/NSREC.2018.8584282
16. Wyrwas, E..: Body of knowledge for graphics processing units (GPUs). NEP-BOK-2018, NASA Electronics Parts and and Packaging Program (NEPP), NASA Goodard Spaceflight Cenger (2018)
17. Kosmidis, L., Rodriguez, I., Lachaize, J., Abella, J., Notebaert, O., Cazorla, F., Steenari, D.: Embedded GPU benchmarking for high-performance on-board data processing. European Workshop on On-Board Data Processing (OBDP2019), 29 February 2019. ESTEC, Nordwijk
18. Tsog, N., Behnam, M., Sjödin, M., Bruhn, F.: Intelligent data processing using in-orbit advanced algorithms on heterogeneous system architecture, 2018 IEEE Aerospace Conference, Big Sky, MT, 2018, pp. 1–8. doi: 10.1109/AERO.2018.8396536
19. HSA Foundation, https://www.hsafoundation.com/. Accessed 14 May 2020
20. AMD ROCm Platform, https://rocmdocs.amd.com/en/latest. Accessed 14 May 2020
21. https://suddendocs.samtec.com/catalog_english/lshm_dv.pdf. Accessed 14 Nov 2020
22. https://www.cubesatkit.com/docs/CSK_PCB_Spec-A5. Accessed 14 Nov 2020
23. Haddad, N.F. et al.: Second generation (200MHz) RAD750 microprocessor radiation evaluation. 2011 12th European conference on radiation and its effects on components and systems, Sevilla, 2011, pp. 877–880. doi: 10.1109/RADECS.2011.6131320.
24. BAE Systems RAD5545 data sheet, https://www.baesystems.com/en/download-en/20190327203103/1434571328901.pdf. Accessed 29 May 2020
25. Sturesson, F., Gaisler J., Ginosar R. and Liran T.: Radiation characterization of a dual core LEON3-FT processor. 2011 12th European conference on radiation and its effects on components and systems, Sevilla, 2011, pp. 938–944. doi: 10.1109/RADECS.2011.6131334.
26. Schaefer, J.J., Troxel, I.A., Gruber, M., Conger, C., Schaf J., Narveson, K.: Heavy ion test results for the MOOG Broad Reach BRE440 Processor. 2019 IEEE Radiation Effects Data Workshop, San Antonio, TX, USA, 2019, pp. 1–6, doi: 10.1109/REDW.2019.8906285
27. Troxel, I.: Radiation tolerant technology: enabling new mission capabilities. Proc. Hardened Electronics and Radiation Technology (HEART) Conference, Albuquerque, NM, March 19–22, 2013.
28. Troxel, I.: Radiation tolerant technology: enabling new mission capabilities. Proc. Hardened Electronics and Radiation Technology (HEART) Conference, Albuquerque, NM, March 19–22, 2013
29. Office of the Chief Technologist, "2015 NASA Technology Roadmaps," NASA, July 2015.
30. Montgomery, B. G.: NASA avionics architectures for exploration (AAE) and fault tolerant computing. Fault-tolerant spaceborne computing employing new technologies 2014, 16–19 June 2014, Sandia Laboratories, Albuquerque, New Mexico
31. Moore, C.: Technology development for NASA's asteroid redirect mission, IAC-14-D2.8-A5.4.1. In: 65th International Astronautical Congress, 29 September–3 October 2014. Toronto
32. Thompson, D.R., Altinok, A., Bornstein, B., Chien, S.A., Doubleday, J., Bellardo, J., Wagstaff, K.L.: Onboard machine learning classification of images by a cubesat in earth orbit. AI Matters **1**(4), 38–40 (2015)
33. Chien, S., Doubleday, J., Thompson, D.R., Wagstaff, K., Bellardo, J., Francis, C., Baumgarten, E., Williams, A., Yee, E., Stanton, E., Piug-Suari, J.: Onboard autonomy on the Intelligent Payload EXperiment (IPEX) CubeSat mission. J. Aerosp. Inf. Syst. (JAIS) (2016). https://doi.org/10.2514/1.I010386
34. Esposito, M., Carnicero D., Bernardo, Pastena, M., Vercruyssen, N., Conticello, S., Dijk, C., Manzillo, P., Koeleman, R.: Highly integration of hyperspectral, thermal and artificial intelligence for the ESA PHISAT-1 mission. International Airborne Conference 2019, Washington D.C.
35. Varile, M., Feruglio L., Franchi, L.: Convolutional Neural Network for Automous Onboard Image analysis. 10th European CubeSat Symposium, 2018.
36. Iacopino, C., Harrison, S., Brewer, A.: Mission planning systems for commercial small-sat earth observation constellations. 9th International Workshop on Planning and Scheduling for Space (IWPSS), Buenos Aires, Argentina 2015
37. https://github.com/krrishnarraj/clpeak. Accessed 1 Nov 2019
38. https://github.com/ekondis/mixbench. Accessed 3 Nov 2020
39. https://bruhnspace.com/rocm-apu. Accessed 2 Nov 2019
40. https://www.amd.com/en/products/embedded-ryzen-v1000-series. Accessed 30 May 2020
41. https://www.cpu-world.com/CPUs/Bulldozer/AMD-A10-Series%2520A10-8700P.html. Accessed 30 May 2020
42. https://openai.com/blog/ai-and-efficiency/. Accessed 26 May 2020