GPU4S: Embedded GPUs in Space Latest Project Updates

Leonidas Kosmidis*, Iván Rodriguez*, Álvaro Jover*, Sergi Alcaide*, Jérôme Lachaize†
Jaume Abella*, Olivier Notebaert†, Francisco J. Cazorla*,‡, David Steenari§

*Barcelona Supercomputing Center (BSC), Spain
†Airbus Defence and Space, France
‡Spanish National Research Council (IIIA-CSIC), Spain
§European Space Agency, The Netherlands

***Abstract*—Following the trend of other safety-critical industries like automotive and avionics, the space domain is witnessing an increase in the on-board computing performance demands. This raise in performance needs comes from both control and payload parts of the spacecraft and calls for advanced electronics systems able to provide high computational power under the constraints of the harsh space environment. On the non-technical side, for strategic reasons it is mandatory to get European independence on the used computing technology. In this project, we study the applicability of embedded GPUs in space, which have shown a dramatic improvement of their performance per-watt ratio coming from their proliferation in consumer markets based on competitive European technology. To that end, we perform an analysis of the existing space application domains to identify which software domains can benefit from their use. Moreover, we survey the embedded GPU domain in order to assess whether embedded GPUs can provide the required computational power and identify the challenges which need to be addressed for their adoption in space. In this paper, we describe the steps followed in the project, as well as a summary of results obtained from our analyses so far in the project.**

I. Introduction and Background

The space domain needs high performance, scalable processing solutions for the increased computational requirements of future missions to enhance the autonomy and for on-board data processing. Space presents a unique set of constraints with a small production volume compared to other safety critical industries; therefore, it is always keen to explore and adapt solutions from other domains in order to reduce non-recurrent costs if possible. The increased performance demand is required both for the *platform computers*, which are responsible for controlling critical functionalities, like guidance, navigation an power distribution as well as the *payload computers*, which are in charge of controlling the payload devices along with pre-processing of their data before they are transmitted to the ground. In the former case, more advanced capabilities are expected from future missions for i.e. increased autonomous navigation. In the latter case, the increase of the data generated by newer on-board scientific instruments which have higher resolutions and higher sampling frequencies requires more computational power to be accomplished.

Graphics processing units (GPUs) is a potential technology from another domain which can provide the higher performance needed. GPUs were originally used as an accelerator for visual applications, but nowadays they have evolved to general purpose high-performance accelerators, which in terms of performance and energy efficiency have surpassed the levels achieved by Central Processing Units (CPUs). The unprecedented level of performance per watt for very demanding computations led GPUs to become an integral part of high-performance computing. As a matter of fact, one quarter (132) of the supercomputers in the recent edition of the TOP500 list (as of November 2019) are based on GPUs, in addition to the Green500, which ranks the TOP500 supercomputers based on their performance per watt, 8 of the 10 top are using GPUs as accelerators for computing. For these promising reasons, past studies analysed the suitability of high-performance GPUs in space [1][2]. However, those studies found that although their energy efficiency is high, their power consumption is an order of magnitude greater than the tight power budget of a space system, which is limited to a couple of Watts and for this reason are not suitable for this domain. The reason for this is not only because of the limited on-board power supply, but also because of the constraints in heat dissipation in the vacuum, which limits cooling solutions to passive dissipators only.

Interestingly, GPUs entered in the embedded domain to satisfy the increasing demand for multimedia-based hand-held and consumer devices such as smartphones, in-vehicle entertainment systems, televisions, set-top boxes etc. These embedded GPUs were re-designed, compared to the high-performance desktop/server ones, to exhibit low-power requirements targeting battery power and thermally-constrained devices. Advances in transistor technology allowed these devices to achieve impressive performance that were only conceivable by high performance systems of the past decade [3].

Despite the lower performance ratio between GPU vs CPU in the embedded domain than in the high-performance devices, due to power and thermal constraints, mobile GPUs were progressively adopted for accelerating heavy workloads, for applications ranging from signal processing, to advanced driving assistance systems (ADAS) in cars, as well as prototype supercomputers for exascale [4][5].

Despite their promising characteristics, embedded GPUs

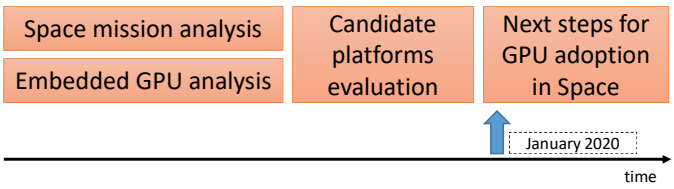

Fig. 1. Main activities. Box width does not represent activity duration

have not been examined for their applicability in this domain. This project's goal is bridging this gap by providing an initial assessment of existing embedded GPUs, as a first step of their further exploration and adoption in this domain. In particular, the project tries to address two questions: whether the algorithms used in on-board processing can be accelerated with GPUs, as well as whether embedded GPUs can provide the computational power required by future on-board processing, within the power limits of space systems. In addition, it is expected to scratch the surface regarding their reliability in the harsh space environment, so that it can be studied better in future projects, which will cover also other aspects that could not be completed in the limited time frame and scope of a pilot project.

The rest of the paper is organised as follows. Section II introduces the project and its main tasks. Section III analyses the performance demand of current and future space missions. Section IV details the outcome of the work on the analysis of the embedded GPU market. Section V presents the preliminary work performed so far, while Section VI provides the ongoing and future work until the end of the project. Section VII presents the challenges and difficulties experienced so far in the project. Finally Section VIII presents the main conclusions of our work so far.

II. The Project

The GPU4S (GPU for Space) project, funded by the European Space Agency (ESA) aims to explore the suitability of embedded GPUs for space covering both the software and hardware perspectives. The project is coordinated by the Barcelona Supercomputing Center (BSC), a leading institution in code parallelisation and optimisation, parallel programming models, critical systems and in particular embedded GPUs as well as an NVIDIA Center of Excellence. In addition it is home to two supercomputers in the actual TOP500 list, Marenostrum and Marenostrum P9 CTE. The latter is based on GPUs and it is currently in the 7th position of the most energy efficient supercomputers (Green500, November 2019). Airbus Defence and Space (ADS) in Toulouse, formerly Astrium, is Airbus' aerospace division, a world-leading primary spacecraft hardware and software supplier. The consortium partners have complementary expertise required to cover the wide range of activities required in the project.

In order to reach its main goal, the project is organised in 4 main work packages which are distributed in time as shown in Figure 1.

Space mission analysis: in this part, led by ADS, we study various space domains, targeting specific algorithms and applications that can benefit from the use of GPUs in space. BSC assists in this task by ensuring that the selected applications are suited for the GPU execution model. This is needed to take into account because there are crucial design differences between GPUs and CPUs which are mandatory to be taken into account to understand how certain algorithms used in space, e.g. applications with divergent path execution among different threads or the ones that do not use coalesced memory accesses, perform differently in embedded GPUs. In particular, certain space algorithms may not be suitable for embedded GPUs despite their high computational nature, and might require to be redesigned.

We provide a preliminary list of applications already validated as GPU-compatible in Section III. Furthermore, we characterize a list of criteria that can be used for the selection of GPU candidates depending on the mission profile.

Embedded GPU analysis: in this work package, which is led by BSC, in correlation with the previous activity, we review the available commercial-of-the-shelf (COTS) hardware and soft-IP (Intellectual Property) in terms of embedded GPUs. Our aim is to identify their characteristics, in order to select a candidate set of boards for evaluation in the next activities based on a set of requirements commonly defined with ADS. We focus primarily on European IP, which leads the market and can provide complete autonomy in the European space sector. To understand the broad spectrum of mobile GPUs, we conduct a taxonomy of existing products.

Besides the mission profile criteria, an important consideration when selecting a candidate board is energy performance and efficiency, because of the critical architectural distinctions between their designs and those of their high-performance counterparts, which must be exploited by the software. This fact has been studied only superficially so far [6], and many works in the field have overlooked these differences and treated them as equivalents. Another aspect that is no less important is the available software ecosystem, which is vital for its application in space. Furthermore, we identify software applications in the space domain that can take advantage of GPUs, also justifying their use. Finally, it is very important to ensure whether these applications and their associated algorithms comply with the GPU programming model.

Then, we study the potential market options to select the suitable candidates for evaluation. We begin by providing a taxonomy of available GPU choices by listing the features and differences between products from various GPU suppliers or families for each category, i.e., GPUs and soft-IP. The taxonomy and summary of our findings from the survey are discussed in Section IV.

Candidate Platform Comparison: based on the two initial steps, the identification of potential GPU candidate platforms follows. Then a comprehensive comparison among the selected embedded GPU candidates against current and upcoming processing devices for the space domain, enables to assess the potential benefits of embedded GPUs, particularly with

regard to high performance requirements for future missions, while meeting the power and temperature limitations of the space environment. For this, we use representative kernels extracted from existing space algorithms to the GPU. Additionally, performance and other data from previous mission and application-specific integrated circuit (ASIC) processes will be used for comparison by standardizing them in accordance to the current space technology node (65nm). This will allow us to select the most adequate GPU platform for space from the evaluated candidates. BSC is responsible for the benchmarking of the selected candidates, while ADS takes care of the benchmarking of existing space processors for the comparison.

Next Steps for GPU adoption: the final step of the study is to identify future steps in the adoption of GPUs in space by defining existing constraints as well as suggesting suitable solutions to address them. To determine the next stages of GPU adoption in the space domain with a system integrator approach, we are looking at the necessary steps for qualification of COTS systems by tackling their system-level reliability issues or the development of radiation-hardened components. In the first case, we are also delivering fault tolerance software solutions tailored specifically to these platforms, for the use of COTS embedded GPUs in space. Moreover, we port an algorithm provided by ESA to the GPU platform. ADS is responsible for defining the adoption roadmap and the use of existing reliability solutions from its experience with designing space systems, while BSC provides reliability solutions particularly developed for GPUs, as well as the implementation of the GPU demonstrator.

III. Space Application Survey

This section examines the domain of space applications in terms of the use of embedded GPUs. The efficiency demands of space missions steadily increase. To illustrate the **current missions**, the Gaia astrometry mission [7] (dedicated to measuring the positions of celestial bodies), launched in 2013 by the European Space Agency as a follow-up mission to the Hipparcos mission [8] in 1989, is 100 times more accurate than its predecessor and is intended to map 1.7 billion stars, 4 times more than Hipparcos. In addition, these scientific missions produce vast amounts of data, which need to be transferred. Transmitting this amount of information is challenging even for our current communication standards. The majority of scientific on-board instruments, including those used for monitoring, use sensor arrays that enable parallel computing. For instance, the International Space Station's (ISS) alpha magnetic spectrometer (AMS-02) generates data at 7 Gigabit/s [9]. Therefore, an array of 600 CPUs is used to reduce the amount of data by 3 orders of magnitude before the transmission [10].

Lastly, upcoming space missions, which include advanced space concepts like space-tug and the Active Debris Removal (ADR) system, will require major computational power to implement new features, specifically for autonomous guidance and navigation control (GNC) based on image processing and autonomous learning [11], to identify, approach, gather and eventually clear debris.

Likewise, on-board data processing for new generation missions will grow in almost all mission types [12][13][14]:

- For robotic exploration and research, high-performance data collection networks are required to satisfy the heavy data rate coming from on-board equipment (spectrometers, imagers, etc.).
- For Earth monitoring, the increase in sensor technology resolution, support for dynamic range and faster read rates have resulted in a drastic growth in sensor bandwidth and data volume, causing significant downlink capacity bottlenecks as well as creating a demand for very high on-board data processing capabilities, for both data processing and compression. Furthermore, emerging technologies, such as deforming mirrors, need significant on board processing. Streaming video recorded from space could also be a potential new application either instead of static observation images or as flight mission data for postmortem analysis. Moreover, image enhancement is another possible direction.
- For next-generation launchers, the considerable increase in data speed and on-board computing power will allow worthwhile applications, like the increase in user telemetry transmission channels. Among them, video data transmission for monitoring agile and critical launch maneuvers, currently not feasible with data transfer rates of up to 400 Mbits/s at a compression ratio of 20.
- The navigation equipment for future missions, such as the space tug concept for the Active Debris Removal (ADR) project, which can shift the orbit of non-cooperative spacecraft, will rely on computer vision.
- In the telecommunications field, upcoming developments go beyond typical geosynchronous equatorial orbit (GEO) based telecommunication systems, including missions such as machine-to-machine communication from low-Earth orbit with medium to large spacecraft constellations or spectrum monitoring.
- It is also expected that radar processing will undergo a similar technological evolution and will be synergistic with the telecommunications domain, as both demand high-performance state-of-the-art signal processing.
- The only domain where performance requirements are not expected to increase is launchers. In the case of scalable launchers, high performance processing requirements are not rapidly evolving, as the need for in-flight video transmission (mainly for post-mortem analysis) requires a strong data compression ratio: in the order of 20. This is already met and the need is not expected to evolve rapidly, remaining in a range similar to that of current Earth Observation needs. Nevertheless, with reusable launchers, the re-entry/landing phases alone require high-availability, high-performance processing devices, especially for vision-based positioning and landing guidance. For this study, such application is considered similar to

that of Planetary Approach and Landing already covered by the exploration domains. Therefore, the launcher domain is not considered as candidate for GPU acceleration in this project.

- Last but not least, new missions will need considerable agility and autonomy provided by sophisticated robotics for in-orbit operation and space exploration, from robotic navigation to landing. Also, the use of artificial intelligence applications is expected to increase in on-board devices due to the high complexity of some space exploration tasks.

The development of complex computing architectures embedding many processing cores on single device with low power consumption as GPUs is a real opportunity to significantly enhance on-board processing power and meet the performance needs of future space applications.

TABLE I
REPRESENTATIVE SPATIAL DOMAINS AND ALGORITHMS SUITABLE FOR GPU ACCELERATION.

Use Case type	Algorithm name	Algorithm description
Science/Image processing	GAIA VPU	Complex Image Processing chain (49 algorithms pipelined)
Science/Image processing	Euclid NIR	Near-Infra-Red image sensor data processing
Optical Observation	GO3S	High resolution image processing algorithm
Compression	ESA CCSDS 122 Image data compression	Generic Image data compression algorithm
Radio Observation/Signal Processing	ADS-B – OCE	Automatic dependent surveillance – broadcast (ADS–B) algorithm-surveillance technology in which an aircraft determines its position via satellite navigation
Vision Based Navigation	GENEVIS (Solver), OpenCV	Computer Vision library for space targets
Neural Network Processing	Image Classification, Tensor Flow	Object detection and classification

We have selected a first set of applications that fit into the areas identified above and are presented in the Table I. Some of these highly computationally intensive algorithms have already been chosen for the processing of high quality data in present and upcoming missions. We believe that all of the representative algorithms in the table may be well suited for GPU acceleration. The reason for this is that the majority of GPU microarchitectures experience a severe throughput penalty when software behaves unpredictably, such as not accessing consecutive memory locations (memory coalescing) as well as taking different paths (branch divergence). Preliminary studies of these applications indicate that they are free of branch divergences, while their memory is accessed in a regular way.

The knowledge already gained and the leveraging of those available and operational software extracts in past and recent parallel studies ensure both a good foundation for this GPU work and a potential starting point for performance and porting effort benchmarking. To finalize the survey and guarantee that as many possible domains are covered, an internal meeting will be organized with all Airbus programs to pinpoint other algorithms that may be attractive for GPU implementation.

The study of space application domains also involves defining a set of criteria for eligible GPU candidates, supported by the specific features of certain application and mission domains. As an example, Low Earth Orbit (LEO) missions have lower availability requirements that permit the use of COTS components, particularly if such components have reliability characteristics as pre-qualified products for use in industrial or safety-critical domains.

Some of the parameters that we have identified from the mission profile are: the long term availability of the technology/equipment to match the long lifespan of space missions, the feasibility of the implementation of the algorithms on the GPU, the power and thermal limitations of the mission, the I/O interfaces, the one-time cost of developing radiation tests and mitigation technologies, the recurrent expense for the number of pieces of equipment, in addition to the flexibility to use various algorithms at specific stages of the mission schedule and the update of the algorithm for correction or extension.

IV. GPU TAXONOMY

This section presents a classification of mobile GPUs, shown in Fig 2, in order to elaborate on the different potential options available for the space domain.

The first category classifies GPUs as embedded or high-performance. This project only focuses on the embedded ones, since the high-performance ones were already covered in previous works [1][2].

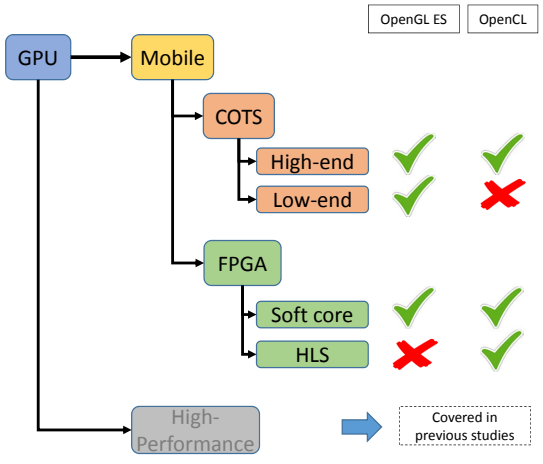

Fig. 2. GPU Taxonomy

Unlike the desktop GPU market where 3 major players dominate (NVIDIA, AMD and Intel), the embedded GPU

market has many major vendors, among which Europe stands out as a major force in which some of the most important embedded GPU IP companies began to emerge: ARM, Imagination Technologies or Bitboys, between others. The latter started the development of Imageon, now re-branded to Adreno by Qualcomm in Finland, after being acquired by ATI and AMD. In a similar way, the Broadcom's VideoCore of the European Raspberry Pi educational computer was initially developed by Alphamosaic (UK). In addition, ThinkSilicon, which specialises in GPUs for IoT devices, is based in Greece.

Another relevant point of our work is that thanks to previous works [6] [15][16], we were able to broaden the scope of the study past the evident choice of OpenCL supported devices (see Fig 2), enabling wider market coverage of existing embedded GPUs while also providing a wider range of choices for the right choice of embedded GPUs. It is important to note that the completion of the survey, particularly with regard to the potential IP acquisition of embedded GPUs to manufacture a rad-hard chip, requires contacting various suppliers, which translates into a lengthy administrative process that is underway.

Mobile GPUs can be split into two categories: FPGA implementations and COTS GPUs. In the first category, FPGA implementations support circuit reconfiguration to ease hardware development. This is useful when a product is not completely specified or additional needs are introduced at a later stage. In the latter category we find mobile GPUs that are implemented as ASICs (Application-Specific Integrated Circuits), usually as part of a larger SoC (System-On-Chip) that often contains CPUs and other devices such as embedded memory, DSPs, etc. Both of these categories are interesting for the space domain.

A. COTS GPUs

COTS designs deliver low recurring costs and improved efficiency compared to FPGA implementations using the same design methodology. COTS components are made using the latest generation of advanced silicon technologies, such as 10nm FinFET, as they are directed at mainstream consumer devices. However, the reliability in these markets with respect to both temporary and permanent failures has not been tackled, as these devices are not used in critical systems and users are generally expected to replace them after every 2-3 years with newer ones with more advanced capabilities. Consequently, an extra effort is needed to protect these components from the effects of radiation on both software and hardware.

The COTS Embedded GPU category can be divided into Low-End and High-End products.

A.1. Low-end GPUs

Low-end GPUs only support graphics APIs such as OpenGL ES 2 and hold a large niche in the mobile industry [17][18]. The simplified architectural design of these products makes the support of OpenCL impossible, however, thanks to this design approach, the power consumption is considerably reduced. ARM's Mali-4XX is the most significant example, which accounts for 20% of the mobile phone sector's market share [19], excluding other widely produced mass-market devices such as single board computers, set-top-boxes, TV-sets, automotive systems and FPGAs. An advantage of such a long-standing and widely used device is the maturity of the technology and the development tools accompanied usually by a well defined documentation.

Other devices from this category are Qualcomm's Adreno 2xx, Broadcomm's VideoCore IV, Imagination Technologies PowerVR SGX and ThinkSilicon's NemaPico and NemaTiny. While OpenCL support is not provided on these low-end devices, it is possible to program efficient solutions for general purpose computations employing general purpose computations over embedded graphics [15][16].

One of the main issues of both high and low-end embedded GPUs is the lack of public domain architectural designs, which prevents researchers to certainly know the full execution pipeline of a device under a specific program execution, which also does not allow observability for space qualification. Vendors only supply high-level implementation details and all of the development tools are proprietary and close source. The only GPU with open specifications to date is Broadcom's VideoCore IV, but the limited number of development tools available work at the assembly level. This results in low productivity and high complexity, while no debugging or profiling methods are available.

A.2. High-end GPUs

High-end embedded GPUs support both graphics and computing APIs, such as OpenCL or NVIDIA's proprietary GPU programming model, CUDA. These architectures are the ARM Mali T6xx-T8xx and G7X families, the latest SGX and Rogue families from Imagination, Adreno 3xx and above from Qualcomm, NemaSmall from ThinkSilicon and the latest Vivante GC series. While these architectures may, in principle, have the ability to support OpenCL runtime, GPU providers do not always release drivers. In fact, Google recently dropped OpenCL use in Android for this same reason. As a result, GPU manufacturers are less interested in developing OpenCL drivers for mobile GPUs unless there is a clear interest from large companies or other sectors, such as supercomputing. On the other hand, although NVIDIA supports OpenCL in its high-performance GPUs, in its embedded GPUs like the Jetson Series, composed by K1, X1, X2 and Xavier, only supports CUDA. Thus, the choice of a high-end GPU has to entail a deeper analysis than just the review of the providers' product sheets.

As the majority of embedded GPUs are primarily aimed at consumer markets, they do not explicitly comply with safety requirements. The exceptions here are Imagination's PowerVR 6XT (GX6650) GPU, which is found in ASIL-B certified automotive platforms like the R-Car H3 from Renesas, and the latest series of Furian GPUs from Imagination and Xavier from NVIDIA, which are designed to be ASIL-D certified, the highest level of safety for automotive applications.

A.3. Machine Learning and other Accelerators

In the previous section, we identified the increased autonomy as an emerging on-board requirement, which requires neural network processing to be implemented. These algorithms are very computationally demanding and GPUs have been identified as suitable architectures for their acceleration. However, almost all embedded GPU design companies have introduced their own machine learning accelerators together with custom software stacks, which can provide higher performance neural network processing, mainly inference, in a more energy efficient way. Similarly, they introduce other special purpose accelerators eg. for vision workloads.

For example, the Project Trillium from ARM includes the Machine Learning Processor, ARM's Object Detection processor, as well as ARM's NN (Neural Network) software. Moreover, ARM's neural network software allows interoperability of machine learning operations on the CPU, GPU or the specific accelerators that can be found in their SoC. Similarly, Imagination Techologies provide Series2NX, a family of neural network accelerators with variable low arithmetic precision from 16 bit down to 4 bits and a software stack with various machine learning frameworks. Think Silicon offers the NEMA xNN power efficient inference accelerator which features 8 bit operations as well as approximate computations. Finally, Nvidia includes in its Xavier SoC Tensor Cores, which is a matrix multiplication accelerator for machine learning workloads, NVDLA Nvidia's Deep Learning Accelerator which is released as open source IP and can accelerate various deep learning algorithms, as well as the PVA (Programmable Vision Accelerator) an image processing VLIW-based (Very Long Instruction Word) accelerator. The latter is similar to Intel's Myriad X VPU (Vision Procesing Unit), another european developed technology from Modivious in Ireland, before it was acquired by Intel.

Each vendor provides its own software stack for these accelerators together with optimisation tools and support for popular machine learning frameworks like Tensorflow, Pytorch etc. However, usually these accelerators are tied to the GPU of the corresponding vendor. Moreover, the machine learning frameworks are constantly evolving due to the popularity of this domain. For these reasons the experimental evaluation and fair comparison of such accelerators is very complicated. Therefore it is considered out of the scope of our project and may be probably investigated in a future ESA activity. Instead, we focus on running the machine learning workloads on the embedded GPUs, which might not be as energy efficient as the NN accelerators, but they are quite efficient for such workloads. More importantly GPUs are more flexible than ASIC accelerators and they are programmed with more general purpose APIs.

B. FPGA Solutions

FPGA solutions provide inferior throughput, however, the underlying COTS FPGA device can be radiation hardened and qualifiable to be used in space, such as the V5QV from Xilinx. Furthermore, research into emerging silicon technologies has demonstrated higher levels of reliability than existing technology used in space (65 nm), increasing expectations concerning the use of these technologies in space.

The Xilinx report about Failures in Time (FIT) in the Xilinx FPGA [20] demonstrates a technology improvement which will be verified with the next generation of FPGA. These solutions can provide further benefits for long-term interplanetary missions. In particular, the configurability of the FPGA can shorten the time-to-launch of a several year mission, even if the desired hardware accelerator is not fully built. Alternatively, in the event that a new, more efficient image compression algorithm is developed, a hardware accelerator can be repurposed to support it, thereby decreasing the size of the data consequently reducing the time for downlink communications. For the aforementioned explanations, both of these potential approaches have their unique strengths, and are thus included in our study.

This category can be subdivided in two levels:

B.1. Soft GPU cores

Soft embedded GPU cores are implemented in RTL (Register Transfer Level) by using a hardware-description language such as VHDL or Verilog. The design is then synthesized into the FPGA, which can be used seamlessly by the software, either through graphics or compute APIs.

Certain GPU design companies offer evaluation solutions for specific FPGA devices, such as Think Silicon with its high-end NEMA GPU and the low-end Think2.5D products for Xilinx's Zynq platforms. But the feedback we have gathered from all the commercial GPU IP providers to date is that most high-end embedded GPUs outperform existing FPGAs, and only small configurations of these designs can be accommodated on very costly ($\sim$50,000 euros) FPGA development boards. Moreover, the IP providers claim that the achieved target frequencies of such designs on the FPGA are very low, which in combination with the reduced configuration of the designs in terms of cores and cache sizes in order to fit in the FPGA, results in very low performance in comparison to their ASIC implementation. Hence, we do not recommend the use of commercial GPU cores for FPGA implementations, but only for rad-hard ASICs.

Open source research-oriented soft GPU cores are also available, like MIAW [21] and FlexGrip [22], which implement AMD and NVIDIA as GPU microarchitectures or FGPU [23] which is different from any commercially available GPU. Unfortunately, these cores do not have energy efficient GPU microarchitectures such as tiled and deferred rendering architectures [6]. These cores support only a subset of the instruction sets (typically limited to integer instructions) and offer support only for the compute APIs, but not graphics. On the other hand, these projects come with development tools of limited functionality, and without debugging or profiling capabilities, whereas most of them are no longer maintained or supported in any way. Aside these problems, the most significant obstacle to the use of these designs in space is their licensing terms. Having a GPL license in most cases,

would mean releasing the RTL code for the complete platform implemented in the FPGA. For this same reason, open source GPU designs are not appropriate for this domain.

B.2. High-Level Synthesis

Modern FPGAs also support OpenCL using High-Level Synthesis (HLS). These products convert OpenCL into custom circuits, which are shaped to run in the FPGA network. While this is not a GPU solution, it is based on OpenCL, which delivers the same application interface as a high-end embedded GPU or a soft GPU core. All of these components can be found in both Xilinx and Altera, including Intel's recent HARP prototypes, with both a CPU and an FPGA on the same chip. This reconfigurable solution has the advantage that the FPGA hardware resources can be used more efficiently across multiple algorithms in comparison to a fixed soft-GPU design solution. However, FPGA reconfiguration (flashing) requires much more time than running different kernels on a GPU, and whilst FPGAs are now used in space missions, this feature has never been used before.

High-level synthesis in OpenCL makes the development and debugging effort significantly easier versus a hardware description language. However, the performance of the generated circuits from the high-level synthesis needs to be evaluated.

In addition, existing space-qualified FPGAs, like the V5QV from Xilinx, are not supported for high-level synthesis. Lastly, we have found that existing HLS tools cannot obtain the same OpenCL code and execute it unchanged on an FPGA, because they need extra code for interfacing between the host and accelerator sides. Moreover, the kernel code needs to undergo major modifications and annotations, so that the generated hardware is efficient, to a degree that will not have anything in common with the original OpenCL code. Because of this, the conclusion is that further research on this path is required, but this is beyond the scope of our study, which focuses solely on embedded GPUs, and is more amenable to a future project related to the design of ASIC hardware for the space.

C. GPU Survey Summary

From the four classes of embedded GPUs that we have defined in our taxonomy, we have determined that the FPGA path should no longer be explored during the project to implement commercial COTS GPU designs or open source GPU designs. HLS has potential, but it was considered outside the scope of this project.

Hence, both high-end and low-end products can be used, offering different trade-offs in terms of throughput, power consumption, programming interfaces, complexity, debugging and development tools, open design and functional safety. Based on the information we have collected from vendors, we have chosen a set of on-board GPUs to assess and compare with existing embedded devices as described in the following Section.

V. Preliminary GPU Evaluation Results

Based on the above presented surveys on space applications and existing embedded GPUs, we have narrowed down the

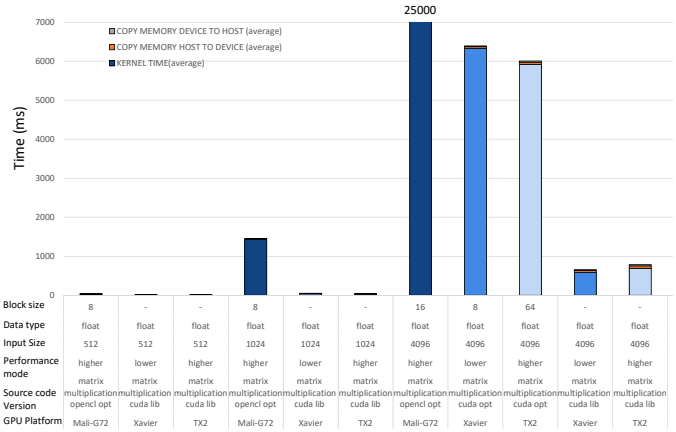

Fig. 3. Performance comparison of various configurations of Matrix Multiplication in the 10W TDP performance mode of the 3 GPU platforms.

GPUs to be evaluated. For cost and effort reasons the selection of the platforms to be evaluated experimentally had to be limited to 3 COTS GPUs, with additional GPU platforms to be evaluated using their published results. The initial selection included 3 embedded GPUs, two European which can be used either as IP or COTS and the latest non-European GPU which has the highest theoretical performance among all GPUs in the embedded market. In particular, the selected GPUs are: the latest ARM Mali GPU G72 found in HiSilicon's HiKey 970, the Imagination Technologies PowerVR Series 6 (GX6650) GPU found in the Renesas RCAR H3 automotive board compatible with ASIL-B functional safety level and the latest embedded NVIDIA platform Xavier which is designed for autonomous driving and certifiable up to the highest automotive safety level, ASIL-D. All platforms have a 10W TDP performance mode which has been determined as an upper limit for on-board systems based on our analysis and two of them, the RCAR H3 and NVIDIA Xavier are automotive-grade products, designed to comply with functional safety certification, therefore they include hardware reliability features.

Unfortunately the RCAR H3 production has been canceled by Renesas after its selection, which has not permitted the procurement of this device for benchmarking. For this reason, we have replaced it by another COTS GPU from NVIDIA, the Jetson TX2, which has a product variant specifically designed for industrial applications.

Based on the space application survey we have designed a GPU benchmark suite for space, which consists of kernels extracted from several space domains. Initially we planned to use Brook Auto [16] which can facilitate the certification/qualification of the GPU software and allows the programmability of low-end GPUs, however since the selection of the GPU platforms included only the latest high-end GPUs we have used their corresponding native programming APIs, OpenCL and CUDA. More information about the designed benchmark suite is going to be provided in a future publication which is currently under peer review, and the source code will be released as open source after the end of the project.

In Figure 3 and Figure 4 we present some preliminary results for various implementations of two widely used kernels in space algorithms, the matrix-matrix multiplication and the Fast Fourier Transform (FFT).

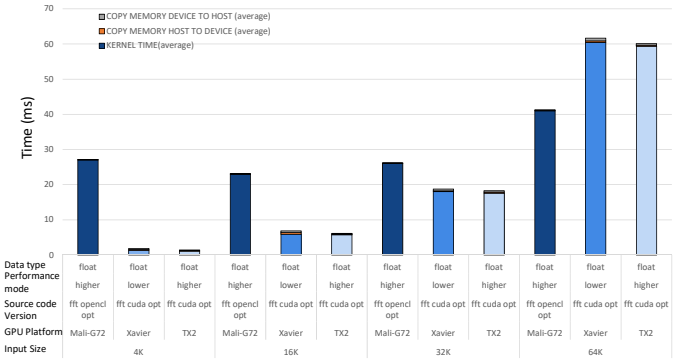

Fig. 4. Performance comparison of various configurations of Fast Fourier Transform (FFT) in the 10W TDP performance mode of the 3 GPU platforms.

In the matrix multiplication benchmark (Figure 3) we notice that for small sizes the 3 GPUs have similar performance, however when the size of the input is increased, the NVIDIA GPUs outperform the ARM one. Although this is partially due to the fact that there is no ARM optimised library for this operation, a comparison of the same hand-written and optimised implementation in all platforms shows the same trend. Moreover, an interesting observation is that for the handwritten implementation the TX2 is faster than the Xavier, while when the NVIDIA library is used (CUBLAS), the Xavier is faster.

On the other hand, for the FFT (Figure 4) we observe that again for small input sizes the Mali is slower than the NVIDIA GPUs. However, for the large input the hand-written and optimised version performs significantly better in the ARM than in the NVIDIA GPUs.

In short, the preliminary comparison results of the 3 candidate GPU platforms under the same power budget of 10W indicate that there is no embedded GPU platform among them that is definitely better than the others. On the contrary, the results vary based on the actual input sizes, the algorithm and its implementation. For this reason, the entire experimental evaluation has to be completed and all the results to be analysed before reaching a conclusion for the most appropriate GPU platform.

VI. Current and Future Project Work

A. Further GPU Benchmarking

Recent results on radiation testing of several embedded GPUs [24] show that embedded GPUs from AMD have good reliability properties. However, AMD embedded GPUs had not been considered in the initial phases of the project for two reasons. The main reason for this decision was that AMD GPUs do not consitute european technology, which was a primary focus of the project. On the other hand, NVIDIA's GPUs were also not european technology, but they were included as the highest performance embedded GPUs available in the market. Second, in the early project stage when the candidate GPUs were identified, the newest embedded GPUs from AMD which could provide equivalent performance capabilities to the Xavier were not available in the market. Consequently, the detailed comparison among the platforms has been extended in order to include the evaluation of an AMD Embedded Ryzen GPU. This task is currently on-going and it is expected to be completed in some months.

In addition to the evaluation with kernels which we extracted from space applications, we have ported an ESA provided application to CUDA, as a demonstrator that space applications can be parallelised to the GPU programming model. The algorithm we have ported is the Euclid NIR (Near Infrared) image processing algorithm used in ESA's Euclid NIR mission [25]. Our preliminary experimental results on the NVIDIA Xavier platform have been published in [26] and show that the GPU implementation of the algorithm is 3.5 times faster than the sequential CPU version on the Xavier platform for a standard image size. However for higher resolutions the speedup is 15 times. A comparison with two representative processors used in space, such as LEON 2 and the PowerPC 750 shows an improvement of 806 times and 128 times respectively. Currently we are working on porting the application to OpenCL, so that we can provide a better comparison among the platforms we have considered in the project.

B. Reliability Considerations and Space Adoption Roadmap

In addition to the performance evaluation and benchmarking, we have started working towards the last main task, which is defining the next steps for the adoption of GPUs in space. In this task, we are mainly focusing on performing some preliminary analysis of reliability solutions for embedded GPUs. Recall that GPU4S is the first ESA project on embedded GPUs, so its purpose is not to address all problems related to embedded GPUs in space. Instead it rather aims to assess in a first stage whether they can be beneficial for the space domain and identify potential issues that need to addressed for their adoption, providing some guidelines in this direction.

As already mentioned, approaches for other domains such as the automotive are usually employed by the space domain in order to reduce non-recurrent costs. Regarding COTS GPUs we propose the use of a solution that we previously developed for the automotive sector, based on dual-modular software-based diverse redundancy [27]. This solution exploits the parallelism of the GPU hardware in order to execute two copies of the same GPU kernel, while ensuring that they are executed on different hardware resources and with a time slack between them, so that a soft error cannot be manifested in both copies in the same way. The comparison of the results of the two copies is performed by a safety microcontroller, such the one already included in automotive-grade platforms like Xavier.

For the GPU IP solutions, since their hardware design can be modified, we propose a hardware solution, again from our

previous work in the automotive domain [28]. This solution applies the same concept with [27], however in this case it is the hardware which guarantees the spatial and temporal diversity. In particular, we present two hardware scheduler modifications. In the first policy we partition the GPU so that each kernel version do not use the same resources, while in the second we follow a round-robin dispatch of threads of the kernel, but again we guarantee that the two copies are not executed in the same hardware elements.

It is worth noting that most IP GPUs are available in RTL form and are not tied to any particular technology or library implementation, although some companies such as ARM provide also cell libraries optimised for different purposes, like high-performance, low-power or area. This means, that these GPU IPs can be synthesized with appropriate cell libraries for space such as radiation hardened or the naturally immune to latch-up ST Microelectronics' 28nm FDSOI european technology, currently used in the ARM-based DAHLIA H2020 project which targets the space domain. However, it is not possible to make any trustworthy estimations about how GPU designs will perform in such techonology nodes or older space technologies such as 65nm. We expect that in that case, the obtained frequency of the embedded GPUs with older technologies will be slower, but this might not be an issue since space electronics already need to be operating in lower frequencies in order to keep power and thermal dissipation within the on-board specification limits. Moreover, the area is inevitably expected to be larger and the energy efficiency to be lower than the ones obtained for free with the latest technology nodes, thanks to Moore's law and Dennard scaling. However, we should not forget that embedded GPUs have been in the market for long time now, as we have mentioned in Introduction, so there is enough evidence of embedded GPUs implemented in such older manufacturing processes and still provide benefits over their CPU counterparts in the same SoCs.

VII. Challenges and Difficulties

In this section, we describe the challenges and the difficulties we have faced in the project so far. The most important challenge we had to face was the limited information provided by the GPU IP companies. In particular, semiconductor industries are very protective about the details of their designs. The situation is even more complicated with embedded GPU designs, whose internals are very rarely provided, even in GPUs with available open source drivers such as Broadcom's VideoCore IV and AMD GPUs. For this reason, we had to perform our initial hardware survey considering only the publicly available information found in marketing material. The next step was to establish communication channels with all the GPU vendors, explain the purpose of our analysis and sign several non-disclosure agreements (NDAs) with each of them, in order to obtain non-public information about their products. However, this is a very long administrative process which requires the involvement of the legal departments of large companies and institutions and has caused long, unexpected delays in the project. In addition, although we obtained such exclusive information which helped us further for choosing the most promising embedded GPU candidates for benchmarking, not all rationale behind this selection can be disclosed in documents such as this publication.

Another limitation when dealing with semiconductor industries, is that no price information or internal data related to performance, power and area is provided, until the end user signs a customer agreement and specifies exactly the requirements of the design eg. which IP would like to license, its configuration parameters etc. However, this was not possible in our case and prevented us to make a comprehensive selection of GPU IP including a comparison of these properties for different available IPs and to be able to perform an accurate normalisation of projected power, performance and area of these designs on older space technologies.

Another important difficulty we faced was the unavailability of some candidate embedded GPU platforms as we already commented. In particular, the initially selected RCAR H3 from Renesas was out of stock at the moment we placed a purchase order. After one year of waiting time, the manufacturer decided to cease its production and cancel all pending orders. This change did not allow Imagination Techonologies' GPU, one of two european IPs which was selected for evaluation to be represented in our benchmarking, caused delays in our evaluation phase and forced us to look for an alternative platform to keep a minimum of 3 boards for experimental evaluation. Similarly, the unavailability of platforms based on the latest embedded AMD GPUs at the time of platform selection kept initially AMD products out of consideration.

Finally, another challenge we had to cope with was the budget limitation, for a project that we extended well beyond its narrow initial scope. Planned as a pilot project with an initial budget of 150K euros, the number of possible GPUs to be purchased and experimentally evaluated was limited to 3. Moreover, this further constrained the effort that could be devoted to conduct the various surveys including communication and legal agreements with vendors, platform setting up, code development and optimisation in order to perform a fair comparison and a demonstrator, as well the preliminary reliability assessment and definition of the adoption roadmap. Despite this limitation, we managed to extend the project outcome significantly beyond its limited scope, for example covering much more hardware devices that the four classes of embedded GPUs we identified in our taxonomy: high-level synthesis with OpenCL on FPGAs and machine learning accelerators. Moreover we proposed additional GPU-oriented reliability solutions inspired from the use of embedded GPUs in other safety critical systems, instead of relying only on legacy solutions used for general purpose COTS hardware in space.

VIII. Summary

In this paper we provide an overview of the objectives and the results we have obtained so far in the GPU4S project, which explores the suitability of embedded GPUs in the space domain.

Based on the results of our first survey of space applications and domains, we have since identified that embedded GPUs are suitable for a variety of multi-domain algorithms such as vision-based navigation, image processing, neural network processing, and signal processing, and that, depending on the specific mission, several characteristics such as reliability and thermal requirements are relevant to the selection of candidate GPUs.

In terms of the domain of embedded GPUs, there are several IP options of the embedded GPU domain that are potential candidates, most of them European, but each of them has different trade-offs that must be evaluated in the next project stages, in order to proceed with the final platform choice that will be experimentally evaluated. However, our research suggests that soft GPU IP solutions in FPGA, as well as high-level synthesis, are not suitable for further investigation in this project (HLS) or at all (soft GPUs).

Our preliminary experimental results under the same 10W power budget indicate that there is not a clear performance advantage of one embedded GPU over the others in our candidate list. In particular, the results depend on the input size of the algorithm, the algorithm and its implementation, so a complete analysis is required with the rest of the benchmarks we have developed in our benchmark suite. The results of this task are expected to be included in a future publication which is currently under peer review, together with detailed information about our benchmark suite, which is going to be released under an open source license. However preliminary results with a space case study ported to an embedded GPU and compared with existing space processors, show clearly that space algorithms can be a good fit for GPUs and that embedded GPUs can provide the desired required performance.

Finally on the reliability side we have explored the use of proposals from the automotive domain both at software level, in case COTS GPUs are used, as well as at hardware level when a GPU IP solution is adopted.

ACKNOWLEDGMENTS

This work has received funding from the the European Space Agency (ESA) under the GPU4S (GPU for Space) Project, answer to the ESA ITT AO/1-9010/17/NL/AF tender with title "Low Power GPU Solutions For High Performance On-Board Data Processing" and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 772773). This work has also been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grant TIN2015-65316-P and the HiPEAC Network of Excellence. MINECO partially supported Leonidas Kosmidis under Juan de la Cierva Formación postdoctoral fellowship (FJCI-2017-34095) and Jaume Abella under Ramon y Cajal postdoctoral fellowship (RYC-2013-14717).

REFERENCES

[1] D. Gonzalez-Arjona and G. Furano, "High-Performance Avionics Solution for Advanced and Complex GNC Systems for ADR (HIPNOS)," *TEC-ED & TEC-SW Final Presentation Days at ESA*, December 2017.

[2] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, "High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation," *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 178–192, February 2018.

[3] "Bringing Console Quality Lighting to Mobile (Presented by Imagination Technologies), Game Developer Conference 2014," http://www.gdcvault.com/play/1020691/Bringing-Console-Quality-Lighting-to, Last accessed: 24-06-2019.

[4] N. Rajovic, P. M. Carpenter, I. Gelado, N. Puzovic, A. Ramirez, and M. Valero, "Supercomputing with commodity CPUs: Are mobile SoCs ready for HPC?" in *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Nov 2013, pp. 1–12.

[5] J. A. Ross, D. A. Richie, S. J. Park, D. R. Shires, and L. L. Pollock, "A case study of OpenCL on an Android mobile GPU," in *2014 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2014, pp. 1–6.

[6] M. M. Trompouki and L. Kosmidis, "Optimisation Opportunities and Evaluation for GPGPU applications on Low-End Mobile GPUs," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017.

[7] ESA, "The Gaia Mission," https://sci.esa.int/gaia, Last accessed: 30-01-2020.

[8] ESA, "The Hipparcos Space Astrometry Mission," https://www.cosmos.esa.int/web/hipparcos, Last accessed: 30-01-2020.

[9] A. Behcet, "Alpha Magnetic Spectrometer (AMS02) experiment on the International Space Station (ISS)," *Nuclear Science and Techniques*, vol. 14, pp. 182–194, August 2003.

[10] NASA, "Alpha Magnetic Spectrometer - 02," https://ams.nasa.gov, Last accessed: 24-06-2019.

[11] S. Kawamoto, Y. Ohkawa, H. Okamoto, K. Iki, T. Okumura, Y. Katayama, M. Hayashi, Y. Horikawa, H. Kato, N. Murakami, T. Yamamoto, K. Inoue, and M. Ohnishi, "Current Status of Research and Development on Active Debris Removal at JAXA," *7th European Conference on Space Debris (SDC7)*, 2017.

[12] O. Notebaert, J. Franklin, V. Lefftz, J. Moreno, M. Patte, M. Syed, and A. Wagner, "Way Forward for High Performance Payload Processing Development," in *Data Systems in Aerospace (DASIA)*, 2012.

[13] M. Patte and O. Notebaert, "Enabling Technologies for Efficient High Performance Processing in Space Applications," in *European Conference for Aeronautics and Space Sciences (EUCASS)*, 2015.

[14] B. Glass and R. Jansen, Eds., *International Workshop on Analogue and Mixed-Signal Integrated Circuits for Space Applications (AMICSA) 2016, Gothenburg, Sweden*. ESA, 2016.

[15] M. M. Trompouki and L. Kosmidis, "Towards General Purpose Computations on Low-end Mobile GPUs," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016.

[16] M. M. Trompouki and L. Kosmidis, "Brook Auto: High-Level Certification-Friendly Programming for GPU-powered Automotive Systems," in *Proceedings of the 55th Annual Design Automation Conference (DAC)*, 2018.

[17] Khronos, "OpenGL ES Overview," 2018, http://www.khronos.org/opengles, Last accessed: 24-06-2019.

[18] Google Developers, "OpenGL ES Version," 2018, https://developer.android.com/about/dashboards/index.html, Last accessed: 24-06-2019.

[19] ARM, "Mali-400," 2018, https://developer.arm.com/products/graphics-and-multimedia/mali-gpus/mali-400-gpu, Last accessed: 24-06-2019.

[20] Xilinx, "Device Reliability Report- Xilinx UG116 (v10.5.2)."

[21] R. Balasubramanian, V. Gangadhar, Z. Guo, C.-H. Ho, C. Joseph, J. Menon, M. P. Drumond, R. Paul, S. Prasad, P. Valathol, and K. Sankaralingam, "Enabling GPGPU Low-Level Hardware Explorations with MIAOW: An Open-Source RTL Implementation of a GPGPU," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 2, Jun. 2015.

[22] K. Andryc, M. Merchant, and R. Tessier, "FlexGrip: A soft GPGPU for FPGAs," in *International Conference on Field-Programmable Technology (FPT)*, 2013.

[23] M. Al Kadi, B. Janssen, and M. Huebner, "FGPU: An SIMT-Architecture for FPGAs," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2016.

[24] I. Troxel, J. Schaefer, M. Gruber, and P. Gauvin, "Summary of Radiation Test Data for Several GPUs," in *2019 Radiation Hardened Electronics Technology (RHET)*, 2014.

[25] ESA, "The Euclid Mission," https://sci.esa.int/web/euclid, Last accessed: 30-01-2020.

[26] I. Rodriguez, L. Kosmidis, O. Notebaert, F. Cazorla, and D. Steenari, "An On-board Algorithm Implementation on an Embedded GPU: A Space Case Study," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2020.

[27] S. Alcaide, L. Kosmidis, C. Hernández, and J. Abella, "Software-only Diverse Redundancy on GPUs for Autonomous Driving Platforms," in *25th IEEE International Symposium on On-Line Testing and Robust System Design, IOLTS 2019, Rhodes, Greece, July 1-3, 2019*, 2019, pp. 90–96.

[28] S. Alcaide, L. Kosmidis, C. Hernández, and J. Abella, "High-integrity GPU designs for critical real-time automotive systems," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*, 2019, pp. 824–829.