SYSGO
EMBEDDING INNOVATIONS

Use of Heterogeneous Computing Systems and Partitioned OS in Space Applications

Thierry Maudire, SYSGO
Oliver Ruiz Dorantes, SYSGO
Damir Bartakovic, SYSGO

ABSTRACT

Increased demand for higher level of integration, miniaturization in particular in the case of nano and picosatellite constellation requires the use of state-of-the-art embedded computing systems and operating systems. We describe how the combined use of MMU- and MPU-based partitioned real-time operating systems on COTS heterogeneous computing platforms contributes to increase integration, improve SWaP (Size, Weight and Power), and reduce costs.

1. INTRODUCTION

Increased demand for higher level of integration, miniaturization in particular in the case of nano and picosatellite constellation requires the use of state-of-the-art embedded computing systems and operating systems. We describe how the combined use of MMU- and MPU-based partitioned real-time operating systems on COTS heterogeneous computing platforms contributes to increase integration, improve SWaP (Size, Weight and Power), and reduce costs.

2. HETEROGENEOUS COMPUTING SYSTEMS

Space exploration has come a long way since the first successful satellite launch in 1957. With the increasing complexity of Space missions and the growing demand for reliable and efficient control systems, the use of advanced processing systems has become increasingly important. Heterogeneous computing systems, which use a combination of different processing elements, offer the potential for high performance, robustness, and flexibility for Space use cases.

A Heterogeneous Computing System (HCS) is a system that utilizes multiple processing elements, each optimized for specific tasks. These processing elements can be either General-Purpose Processors (GPPs) or specialized processors, such as Digital Signal Processor (DSP), Neural Processing Units (NPU), Field-Programmable Gate Arrays (FPGAs), and Graphics Processing Units (GPUs). Heterogeneous computing systems such as NanoXplore NG-Ultra, Xilinx Zynq™ UI-traScale+™ MPSoC/RFSoC, Xilinx Versal™ Adaptive SoC, amongst others are becoming increasingly popular in Aerospace applications [4]: They allow for flexible and efficient use of resources, as well as the ability to handle both demanding hard real-time tasks, and computing intensive tasks [2][3].

The use of Heterogeneous Computing Systems in Space use cases provides for the following advantages:

- **High Performance:** HCSs can provide high performance, as they allow tasks to be distributed among multiple processing elements. This can reduce processing time and improve overall system performance, as specialized units are more effective than GPPs for performing tasks such as image processing, sensor fusion, machine learning, etc.
- **Flexibility:** HCSs allow for the use of different processing elements for different tasks, which can improve the flexibility of the system and allow for future upgrades and modifications, for instance with the use of programmable logic arrays.
- **Robustness:** HCSs can provide robustness, as different processing elements can be used to perform critical tasks, providing redundancy in case of failures.
- **Energy Efficiency:** HCSs can be designed for energy efficiency, as different processing elements can be optimized for specific tasks and turned on or off as needed, even more if they can be integrated into a single System on Chip (SoC) or System on Module (SoM).

The implementation of HCSs for space use cases requires the selection of appropriate processing elements and the development of a system architecture that takes advantage of the strengths of each element. This can be done using a combination of hardware and software, including the use of embedded systems, Real-Time Operating Systems (RTOS), and Hypervisors.

3. OPERATING SYSTEMS FOR HETEROGENEOUS COMPUTING PLATFORMS

Partitioned Operating Systems (OS) are a type of operating system that allows multiple applications or tasks to run simultaneously on the same computing system, while also maintaining a degree of isolation between them. This type of OS is particularly well-suited to Space computing systems, where hardware resources are limited, and software reliability is critical.

Partitioned operating systems provide for the following key advantages:

- **Fault Isolation and Containment:** One of the primary advantages of partitioned OS for space computing systems is fault isolation. In a partitioned OS, each application or task is isolated from the others, meaning that if one application fails, it does not affect the others. This is important in Space computing systems, where the reliability of software is critical, and failure can have serious consequences. By isolating applications from one another, partitioned OS can reduce the likelihood of failures spreading between applications and causing system-wide failure.
- **Resource Allocation:** Another advantage of partitioned OS for Space computing systems is the ability to allocate resources to specific applications or tasks. In a partitioned OS, resources such as CPU time, cores, memory, and input/output controllers can be allocated to specific partitions, ensuring that each application or task has the resources it needs to operate effectively. This is particularly important in Space computing systems, where hardware resources must be used efficiently.
- **Security:** Partitioned OS can also provide enhanced Security for Space computing systems. By isolating applications from one another, partitioned OS can prevent unauthorized access to sensitive data or resources. This is critical in Space computing systems, where the Security of data and systems is paramount. Partitioned OS can also provide enhanced Security by allowing for the implementation of Security policies specific to each partition.

- **Real-Time Performance:** Partitioned OS can also provide real-time performance for Space computing systems. Real-time performance refers to the ability of a system to respond to events in a timely and deterministic manner. In a partitioned RTOS, real-time applications can be given priority over non-real-time applications, ensuring that they receive the resources they need to operate effectively. This is important in Space computing systems, where real-time performance is critical for tasks such as navigation and control.
- **Ease of Maintenance:** Finally, partitioned RTOS can provide ease of maintenance for Space computing systems. In a partitioned RTOS, each partition can be maintained independently, meaning that updates or changes can be made to one partition without affecting the others. This can simplify maintenance and reduce downtime, which is important in Space computing systems where downtime can have serious consequences.

Most of the above characteristics are key to allow the consolidation of multiple applications having different level of criticality, either for the functional Safety or the Security of the onboard equipment.

As a consequence, Partitioned Real-Time Operating Systems offer numerous advantages for Space computing systems, including fault isolation, resource allocation, Security, real-time performance, ease of maintenance, and support for Multiple Independent Levels of Security/Safety architectures (MILS) [1]. By using partitioned OS, Space computing systems can ensure the reliability and Security of their software, while also maximizing the efficiency of their hardware resources.

Partitioned RTOS need hardware support to provide for some of the above services. In particular, Memory Management Unit (MMU), Memory Protection Unit (MPU), as well as IOMMU are typically relied on for the spatial protection mechanisms. The next section covers differences between MMU and MPU.

4. MMU VERSUS MPU

A. MMU

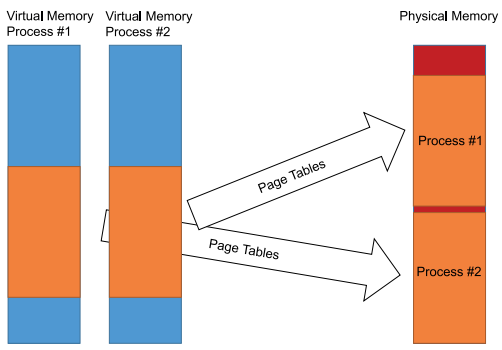

Figure 1: Process separation in an MMU-based system

As shown in Figure 1, a Memory Management Unit (MMU) is part of complex processor CPUs and controls the access of the operating system (kernel) and applications to the main memory. The term memory refers to all locations that can be addressed by the CPU and may comprise Read Only Memory (ROM), Random Access Memory (RAM) as well as I/O memory.

On multi-core processors, there is one MMU per core. The MMU provides virtual memory that gives the illusion of a large address space, dedicated to a single process. Those address spaces are usually unique to each process; however, their address ranges are typically overlapping. Each chunk of virtual memory (called page) can be mapped to a page in the linear physical memory. While two processes may have overlapping virtual memory areas, their mapping to the physical address space is usually distinct. This allows the implementation of complex operating systems with multiple independent processes, like for instance Linux based OSes. When combined with hardware virtualization assistance or extensions (for instance ARM VHE, Intel VT-x/VT-d, or AMD AMD-V), hypervisors can be used efficiently to host general purpose guest operating systems with only minimal overhead.

B. MPU

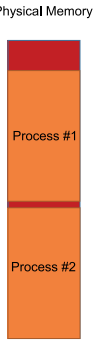

Figure 2: Process separation in an MPU-based system

Figure 2 shows a Memory Protection Unit (MPU), which is often found in less complex CPUs such as microcontrollers. This significantly simplifies the chip design; however complex operating systems (such as Linux) generally cannot be executed. In this case there is exactly one linear physical address space visible to the operating systems as well as all applications and as a consequence, all processes share the same address space. However, read/write access can be explicitly granted to processes by means of MPU regions.

The amount of memory regions is limited. One region can cover a contiguous memory range, depending on the architecture the start address and the size of a region might have different alignment restrictions, for example on ARMv8-R it must be aligned to 64bytes, for ARMv7-R it must be naturally aligned to power of two. Memory regions typically have a larger size than 4 KB and are typically stored in few (e.g. 10s) registers.

The advantages of MPU-based cores when compared to MMU-based ones are typically:

- Less complex hardware
- More deterministic
- Less sensitive to single event upset (SEU) failures
- Faster boot time
- More common support for lock-step mode on MPU-based processors

5. SYSGO PIKEOS AND XILINX SOCS

The Xilinx Versal® AI Edge Adaptive Compute Acceleration Platforms (ACAP) are a series of devices providing various heterogeneous elements in a single package [8]:

- Scalar Engines:
  - ARM Dual-Core Cortex-A72 (MMU-based)
  - ARM Dual-Core Cortex-R5F (MPU-based)
- Adaptable Engines:
  - Programmable Logic Array
- Intelligent Engines:
  - AI engines
  - DSP units

The AI Edge ACAP is a device optimized for real-time, high-performance applications in the most demanding environments, such as Space, where its reduced Size, Weight and Power (SWaP) is key to design embedded computing platforms with increased miniaturization. Its heterogenous architecture offers a wide range of specialized units (ARM Cortex-A cores, ARM Cortex-R cores, FPGA, AI engine, DSP, Crypto accelerators, etc.) that the various tasks satellite must perform will benefit from.

Of course, the satellite on-board software, and in particular the operating systems must leverage the flexibility and adaptability capabilities of such heterogenous system to make the most of it.

SYSGO's PikeOS combines real-time operating system (RTOS), virtualization platform and Eclipse-based Integrated Development Environment (IDE) for embedded systems. The PikeOS real-time operating system has been developed for Safety- and Security-critical applications with certification needs in the fields of Aerospace & Defense, Automotive & Transportation, Industrial Automation, Medical, Network Infrastructures and Consumer Electronics.

One of the key features of PikeOS is the capability to safely execute applications with different Safety and Security levels concurrently on the same platform. This is achieved by the strict spatial and temporal segregation of these applications by means of software partitions. A software partition can be seen as a container with pre-allocated privileges that can have access to memory, CPU time, I/O, but also a pre-defined list of PikeOS services. With PikeOS, the term application refers to an executable linked against the PikeOS API library and running as a process inside a partition. Due to the nature of the PikeOS API, applications can range from simple control loops up to complete para-virtualized guest operating systems like Linux or hardware virtualized guests. Software partitions are also called Virtual Machines (VMs), because it is possible to implement a complete guest operating system inside a partition which executes independently from other partitions. PikeOS can be seen as a Type 1 Hypervisor in addition of being an RTOS.

Security-critical applications for real-time embedded systems benefit from the PikeOS Multiple Independent Levels of Safety/Security (MILS) architecture. This MILS architecture offers a separation microkernel allowing the combination of trusted and untrusted code on a single hardware platform. PikeOS complies with the MILS architecture concept has been evaluated against the Common Criteria (CC) standard to an assurance level EAL5+ [5] [6], and is also qualified to ECSS Category A level.

PikeOS does require a MMU and supports amongst other processor architectures ARM Cortex-A cores. The following diagram give an overview of PikeOS architecture.

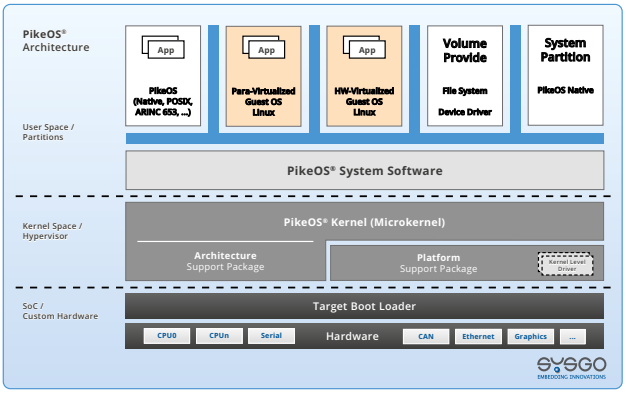

Figure 3: PikeOS software architecture

PikeOS for MPU re-uses most of PikeOS code base, and provides for the same API available for application development, offering then a high level of interoperability between applications developed for PikeOS and those developed for PikeOS for MPU. As its name indicates, PikeOS for MPU is targeted to MPU-based CPU cores, and therefore supports ARM Cortex-R cores.

Both partitioned real-time operating systems are based around the same separation kernel architecture, and use the same graphical development environment (CODEO). Apart from the MMU versus MPU difference, these two RTOS also differ in their kernel processing mode: PikeOS is architected around a SMP (Symmetric Multi-Processing) kernel, while PikeOS for MPU is based on an AMP (Asymmetric Multi-Processing) architecture.

The combination of PikeOS and PikeOS for MPU allows the integrator of the system to take advantage of the benefits

of the AI Edge ACAP device, while still respecting functional Safety and Security constraints.

Both PikeOS and PikeOS for MPU have already achieved or are in the process of obtaining functional Safety and Cybersecurity certifications for different standards addressing various markets, like to name just a few:

- ECSS for Space (functional Safety)
- DO-178C for Avionics (functional Safety)
- Common Criteria (Cybersecurity) [5] [6]

The large amount of code shared between PikeOS and PikeOS for MPU also means that each RTOS can benefit from the certification artefacts generated for the other, accelerating the availability of certification packages.

The following diagram describes a possible use of PikeOS and PikeOS for MPU on the Xilinx Versal® AI Edge ACAP device. PikeOS executes on the dual Cortex A72 core cluster. Its separation and virtualization capabilities allow for hosting very different functions such as Safety-relevant functions with hard real-time requirements (ECSS qualified), Machine Learning (ML) functionalities within a hardware virtualized Linux guest operating system partition using for instance TensorFlow libraries or some other ML engine. Finally, some navigation application relying on APEX.OS environment executed within a PikeOS POSIX partition.

6. Conclusion

The use of heterogeneous computing platforms, in particular those that leverage SoC or SoM that offer the highest level of integration and miniaturization to achieve low SWaP footprints, is one of the most promising trends for nano and picosatellites On-Board Computers (OBC) and mission payload computers architecture designs.

To take advantages of such platforms, and avoid increasing the complexity of the software development, validation and maintenance, care must be taken in choosing the most appropriate combination of Real-Time Operating Systems (RTOS), Hypervisors, General-Purpose Operating Systems (GPOS) and Middleware.

The level of specialization offered by such heterogenous hardware architecture must not be offset by the use of inadequate software platforms. SYSGO's PikeOS and PikeOS for MPU combination on such platform like Xilinx Versal® AI Edge ACAP device allows the integrator to distribute the satellite software functions to the specialized units while still ensuring the appropriate level of functional Safety and Cybersecurity assurance, and fulfill the strict hard real-time requirements typically required by satellite on-board software.

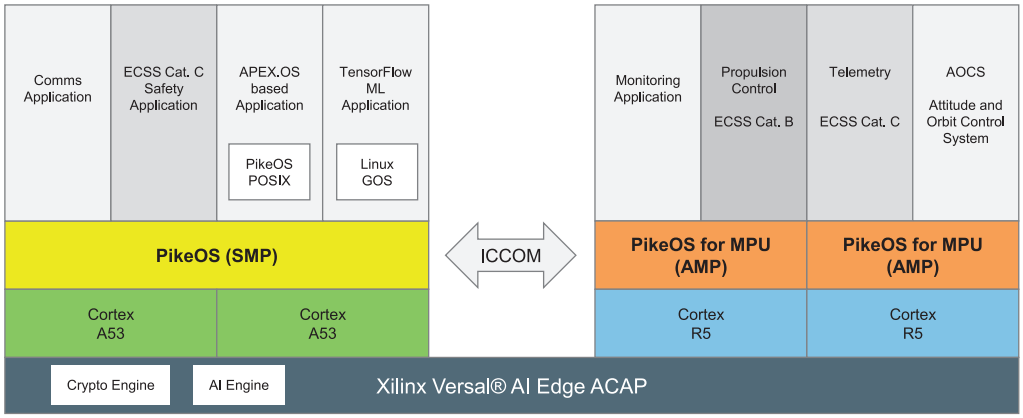

Figure 4: Satellite use case

On its side, PikeOS for MPU can be used to host the satellite functions with the strictest hard real-time requirements such as for instance propulsion control, attitude control or telemetry functionalities.

In addition, communication between applications running on PikeOS and applications running on PikeOS for MPU is possible via queuing/sampling ports or shared memory which offer communication mechanisms completely transparent whether the applications involved are local or hosted on the other RTOS. The communication mechanisms between both RTOS (ICCOM) rely on HW support provided by the SoC.

7. REFERENCES

[1] D. Bartakovic, A. S.-F. v. Blomberg, O. Kühlert, T. Jukić, G. Fumaroli, H.-J. Herpel, P. L. Cueva, J. M. Carranza and M. Guy, “Use of Multiple Independent Levels of Security and Safety (MILS) architecture for space on memory protection unit (MPU)-based systems,” in DASIA 2022, 2022.

[2] N. Tsog, "SPACE COMPUTING USING COTS HETEROGENEOUS PLATFORMS, INTELLIGENT ON-BOARD DATA PROCESSING IN SPACE SYSTEMS," 2021. [Online]. Available: https://www.es.mdh.se/pdf_publications/6325.pdf.

[3] A. González, "Trends in Processor Architecture," 2018. [Online]. Available: https://arxiv.org/vc/arxiv/papers/1801/1801.05215v1.pdf.

[4] M. F. C. Ames Research Center, “State-of-the-Art Small Spacecraft Technology,” jan 2022. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/2022_soa_full.pdf.

[5] S. GmbH, “Security Target PikeOS Separation Kernel v5.1.3,” 2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/Reporte1100/1146b_pdf.pdf.

[6] S. GmbH, "Security Certification Report for PikeOS Separation Kernel Version 5.1.3," 2022. [Online]. Available: https://www.allianz-fuercybersicherheit.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Report/Reporte1100/1146a_pdf.pdf.

[7] G. Richardson, D. K. Schmitt, M. Covert and C. Rogers, "Small Satellite Trends 2009-2013," 2015. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3212&context=smallsat.

[8] Xilinx, "ACAP at the Edge with the Versal AI Edge Series," 2021. [Online]. Available: https://docs.xilinx.com/v/u/en-US/wp518-ai-edgeintro

SYSGO Headquarters
+49 6136 9948 500

SYSGO France
+33 1 30 09 12 70

SYSGO Czech Republic
+420 222 138 111

sales@sysgo.com

www.sysgo.com