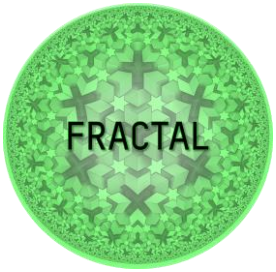

Deliverable

D3.1 Preliminary Fractal hardware node and support

Deliverable Id:	D3.1
Deliverable Name:	Preliminary FRACTAL hardware node and support
Status:	Completed
Dissemination Level:	Public
Due date of deliverable:	31. August 2021
Actual submission date:	31. August 2021
Work Package:	WP3
Organization name of lead contractor for this deliverable:	ETH Zurich
Author(s):	Frank K. Gürkaynak, ETH Zürich
Partner(s) contributing:	Jérôme Quévremont, Thales Alexander Flick, PLC2 Jaume Aubella, BSC Palacio Edurne, IKER Carles Hernandez, UPV Bekim Chilku, Siemens Iñaki Paz, LKS Juan Garcia Enamorado, Qualigon

Abstract:
This deliverable describes the different hardware nodes of the FRACTAL project that will be used to in the technical WPs to develop different services of FRACTAL, explain how they are being utilized for different Use Cases and present an up to date status.

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056

Co-funded by the Horizon 2020 Programme of the European Union under grant agreement No 877056.

Contents

History

Version	Date	Modification reason	Modified by
v0.1	08.June.2021	Initial skeleton	kgf
v0.5	20.August.2021	Version ready for review	kgf
V1.0	31.August.2021	Completed review	kgf

1 Summary

The main objective of the FRACTAL project is to "*create a cognitive edge node enabling a fractal Edge that can be qualified to work under different safety-related domains*". Furthermore, it is stated in the DoA that "*This computing node will be the basic building block of intelligent, scalable and non-ergodic IoT*". As such the hardware node is a central part of the FRACTAL project around which 28 partners collaborate, investigate and industrial partners develop their use cases.

This deliverable (D3.1) is the first of a series of deliverables that describe work done within the FRACTAL project on the hardware of the FRACTAL node. This is the first of three deliverables on HW node, and it will be updated throughout the project with D3.3 (M18), and D3.5 (M20). These three deliverables are also paired with the "software node and services" deliverables D3.2, D3.4 and D3.6.

The FRACTAL project brings together a large number of partners (28) both from the industry and academia, working on varied and challenging topics as well as eight industrial use cases. It was already a challenging task to provide a set of solutions for the hardware node in this context and combined with restrictions around COVID and worldwide supply disruptions for electronic components, partners in WP3 had to face additional challenges.

In the original plan two main options for the hardware node were foreseen:

- **Commercial node** based around the Xilinx VERSAL ACAP (Adaptable Compute Acceleration Platform)
- **Customizable mode** based around the open-source RISC-V based PULP platform

These two main nodes continue to form the backbone of the developments and implementations for the FRACTAL project, but to accommodate practical needs and requirements of project partners additional (related) platforms were also leveraged when necessary.

The organization of the deliverable is as follows. Section 2 provides a general introduction to the hardware nodes and following the discussions around D2.1 "Platform Specification (a)" clarifies the role of hardware platforms in FRACTAL for use cases. Section 3, then summarizes the current state of the hardware platforms, and Section 4 describes how the technical developments in work packages WP4/5/6 make use of the hardware nodes being described in this deliverable. Section 5 lists the plans for the use case providers and the hardware nodes described in their use cases and finally Section 6 provides conclusions for the first year of FRACTAL activities on Hardware Node development.

2 Introduction

FRACTAL is an ambitious project to design a cognitive edge node that is capable of learning how to improve its performance against the uncertainty of the environment. In the project proposal, we had identified four strategic objectives of FRACTAL to reach this goal:

- **Objective 1**: Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity. This part is mainly being addressed as part of WP3.
- **Objective 2**: Guarantee extra-functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems), which has determined the tasks of WP4
- **Objective 3**: Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviours, which is the topic of WP5
- **Objective 4**: To integrate fractal communication and remote management features into FRACTAL nodes, which will be covered by WP6.

While the development of both the hardware and software of the node architecture is the primary goal of WP3 that addresses (Objective 1) it can be seen that the node plays an essential part of the developments of the other objectives as the technical developments in WP4/5/6 are expected to be demonstrated around the hardware nodes described as part of this deliverable in several use cases.

From the inception of the project, a key aspect was to identify a hardware platform that could lead to commercialization within time and cost limitations of the project. It was important to offer a mature platform to end-users for the integration and assessment of their use cases already at the start of the project, as well as a relatively short path towards commercialization of the FRACTAL approach. Project partners had already identified Xilinx VERSAL platform supported through FRACTAL partner PLC2 as the most suitable SoC system as the **commercial hardware node**. As a commercial system with significant resources, Xilinx VERSAL is able to fulfill the performance requirements of even the most challenging use case and project considered within FRACTAL while offering a mature development environment based around an industry standard design flow.

While the commercial platform offers many advantages, especially in short-term commercialization efforts, such a platform presents some well-established solutions, and the customization options are limited within the processing system (known as PS within a Xilinx MPSoC system). In order to explore, practically without any additional constraints, a **customizable node** was added. Designed around the open RISC-V instruction set architecture (ISA) and based on the open-source PULP platform (maintained by partner ETHZ) the customizable node allows FRACTAL partners a powerful and flexible starting point for the development of the custom node and a viable path for longer-term product development without an early commitment to a

proprietary ISA and platform . The customizable node was also offered in an extremely flexible FPGA-based development platform where resources in the node, as well as their organization, can be adapted as needed to enable a larger range of tradeoffs.

The status of both of these platforms will be discussed in Section 3 with some detail. Moreover, at the beginning of the project, especially during the work done for D2.1 "Platform specification (a)", it became clear that especially in the initial stages of the project the official FRACTAL nodes had to be augmented by other complementary platforms as well. There are two main reasons for this:

- Supply chain issues partially as a result of global response to COVID related restrictions, has limited the availability of Xilinx VERSAL platforms, with few partners having access to the node within the first year. This is of course a temporary issue and all FRACTAL partners that need access to a Xilinx VERSAL platform are expected to get one sooner than later.
- While the customizable node offered an interesting alternative, especially as it supported a RISC-V based open-source solution, some of the use case requirements were geared towards higher end solutions that exceeded the capabilities of the IoT based platform made available. Some partners realized they could use other RISC-V based platforms and still remain compatible with the FRACTAL platforms in the longer term. We will describe these platforms and their rationale in Section 3.3.

A further discussion centered around how the FRACTAL hardware nodes would be demonstrated in use cases. This will be discussed in the following subsection 2.1 and the individual solutions currently in plan for use cases will be covered in Section 5.

2.1 Role of HW platforms in FRACTAL Use Cases

The comprehensive discussions with all FRACTAL partners during the preparation of D2.1 "Platform specification (a)" showed a number of issues with the initial approach regarding how FRACTAL hardware nodes will be demonstrated as part of the use cases.

As the use case is a concrete demonstration for the use case provider, it should not be surprising that the main goal of the use case provider is to make sure that the use case can run without issues within the FRACTAL project. As a result, some project partners expressed rather extensive requirements for their own use cases in order not to be limited by the hardware capabilities in the future and some others expressed interest in using systems that they are more familiar with. In practice this has led to several use case providers stating the need for a symmetric multi-core system running a standard Linux distribution.

In all cases, these requirements are perfectly understandable and most of them could be implemented using the commercial node of FRACTAL, the Xilinx VERSAL platform. As outlined in Section 3.2 the basic customizable node has been targeted towards

simpler IoT applications and lacks the 'horsepower' to fulfill several of these requirements. At first sight this creates an apparent imbalance of utilization between the commercial and the customizable node.

FRACTAL partners have discussed various approaches to provide a solution and have decided on a number of measures to make sure that ideas developed as part of FRACTAL are validated on common platforms that are available to all project partners and have made the following recommendations.

- There are several use cases (a detailed breakdown per use case is given in Section 5) that are content to use the FRACTAL hardware nodes as provided.
- FRACTAL has identified three tiers of FRACTAL hardware nodes: low (mist), medium (edge), high (cloud) versions that all share similar interfaces and interact with each other. Figure 1 shows such an organization where simpler nodes are acquiring data and delegating more complex tasks to nodes with higher complexity. The figure is meant as an example, and different allocation of tasks are currently under discussion within WP5/6. In this model, the commercial node covers the higher-end version, while the customizable node is seen as the lower-end version. As described in Section 3, partners have suggested several alternatives for the medium-end nodes.

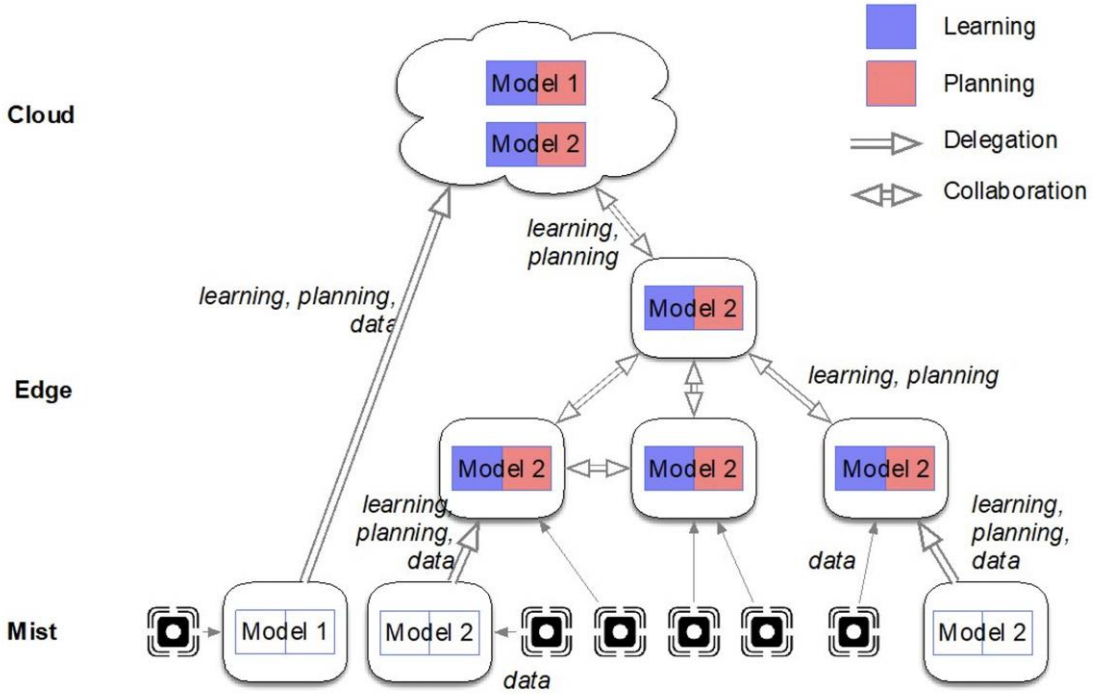

Figure 1. A schematic drawing of a possible FRACTAL system deployment using three different tiers of FRACTAL hardware nodes with different capabilities (drawing from WP5 technical meetings).

- Some partners are relying on their prior work and experience to implement some of their contributions. Most of these are based on hardware systems that are similar and/or compatible with FRACTAL nodes but have some

differences. These include implementations in earlier models of Xilinx MPSoC platforms rather than the VERSAL, as well as other openly available RISC-V systems. Out of practical considerations, FRACTAL partners have added these as additional platforms to the initially identified hardware nodes.

It was also recognized that official FRACTAL nodes could be instrumental for research aspects involving developments in WP4/5/6 and the experience from these explorative works could then be used to evaluate the potential of these developments in use cases that use more traditional solutions. As a concrete example, novel safety solutions with hardware support could be explored on a small scale in the customizable node as part of WP4. The results of this exploration could then be used to directly estimate the gains achievable by this approach in a use case that used an alternative hardware node. The work done throughout the first part of the project allowed partners to realize different possibilities and showed that the key point was that all developments from FRACTAL technical work packages should be accessible for all FRACTAL partners. While the initially identified Hardware Nodes cover a large range of the specification spectrum, partners could also make use of additional hardware nodes as long as this work could be used/verified/evaluated by all partners.

3 HW nodes provided for FRACTAL

The FRACTAL project relies on a set of hardware nodes to demonstrate the FRACTAL approach, especially the technical developments in WP4/5/6, and allows all partners to be able to experiment and use these developments in their own environment.

The two main hardware nodes from FRACTAL were already identified prior to the start of the project as:

- **Commercial node** based around the Xilinx VERSAL platform, a high-end FPGA platform with state of the art acceleration engines for machine learning applications.
- **Customizable node** based around the RISC-V based PULP platform, geared towards more IoT domain that allows the partners to experiment not only with the surrounding system, but all aspects of the platform, including instruction set extension.

To support these, PLC2 and ETHZ have joined the Fractal consortium and have been assisting partners to find solutions with the respective platforms. As the project progressed, to cope with the large variety of requirements and to leverage existing prior work, additional nodes that remain compatible with the FRACTAL project goals have been identified.

This deliverable (D3.1) and its subsequent updates (D3.3, D3.5) describe the state of the hardware nodes used within FRACTAL.

3.1 The commercial node (Xilinx VERSAL)

The Xilinx VERSAL ACAP is expected to be deployed as part of the VCK190 Evaluation Kit board, which provides support for several I/O interfaces and memory devices. The VERSAL architecture combines different engine types with a wealth of connectivity and communication capability and a network on chip (NoC) on a

configurable platform to enable seamless access to the full height and width of the device.

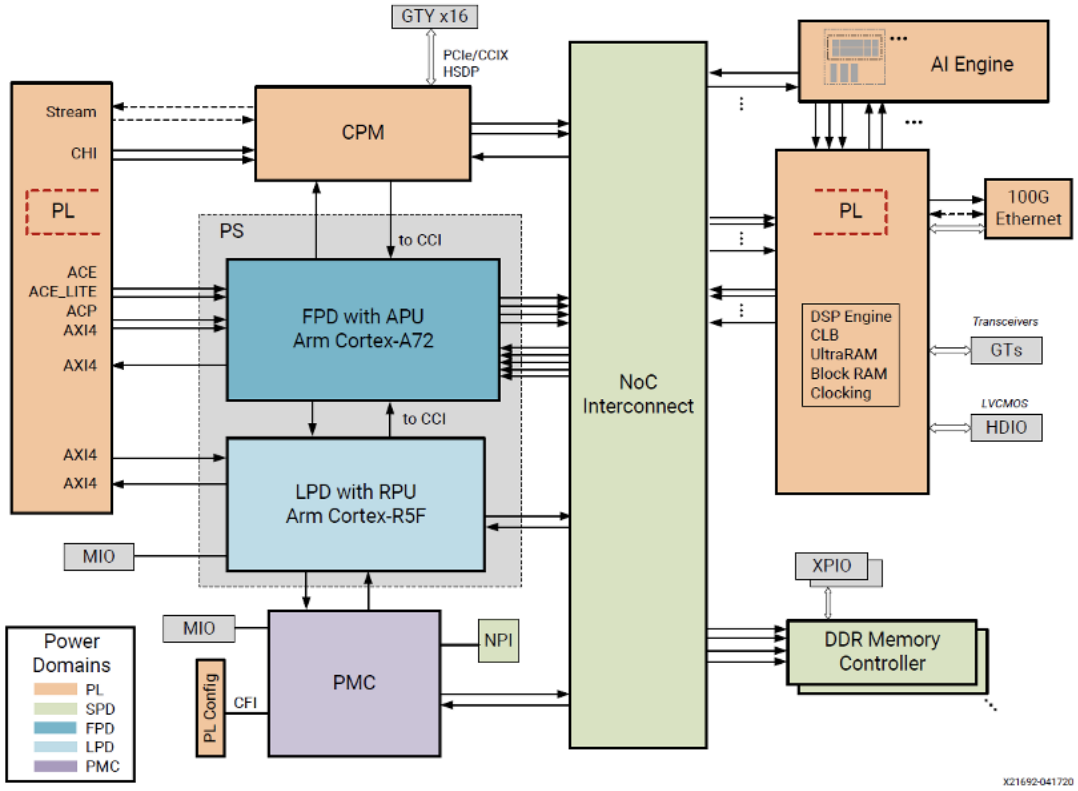

Figure 2. Top level schematic of Xilinx VERSAL Platform

As the latest generation Xilinx products, the VERSAL series provides impressive performance while retaining its programmability. The VC1902 ACAP Xilinx FPGA device that the VCK190 evaluation board[1] is centered around, boasts 400x AI engines, almost 2'000 DSP engines, close to 2 million system logic cells and 900'000 LUTs that are traditionally used to map custom logic as well as 191 Mb of internal memory. The FPGA has 4x 256-bit DDR memory controllers, as well as 4x PCIe4.0, 4x 100G Ethernet and one CCIX serial interface giving tremendous amount of bandwidth that should satisfy even the most demanding applications.

Like the earlier Xilinx Zynq MPSoC products VERSAL ACAP devices still offer the two main components.

- **Processing System** (PS) consists of a dual high-performance ARM Cortex A72 cores that can run Linux or other operating systems. This system is augmented by a dual-core ASIL-C certified real-time processing subsystem based on ARM Cortex R5F cores. Together these systems address the needs of most modern computing needs using a traditional programming interface.

1 https://www.xilinx.com/products/boards-and-kits/vck190.html

- **Programmable Logic** (PL) allows this system to be augmented by hardware accelerators customized to a particular compute function, which makes them best at latency-critical real-time applications (e.g., automotive driver assist) and irregular data structures (e.g., genomic sequencing).

What sets VERSAL ACAP devices apart from conventional FPGA approaches is the hardened IP platform to provide highly configurable connectivity and infrastructure to drive all computation features. On this platform each of these computation engines are deployed to best serve specific computation models to support the full application.

The compute cluster of ARM cores in the PS part constitutes the **Scalar Engines** to help with complex sequential algorithms. A further type of these engines are the **Adaptable Engines** (AE), which are made up of traditional FPGA programmable logic (see PL above) providing local memory in connection with the next generation of the industry's fastest programmable logic.

The **Intelligent Engines** are available for high computation workloads and are available with the PL side digital signal processing blocks (DSP engines) or the AI Engines (AIE) as a specific IP block within specific device family members. The AI Engines are set up as an array of innovative very long instruction word (VLIW) and single instruction, multiple data (SIMD) processing engines and memories. These permit 5X–10X performance improvement for machine learning and DSP applications. The AI Engine processors deliver more compute capacity per silicon area versus PL implementation of compute-intensive applications. AI Engines also reduce compute-intensive power consumption by 50% versus the same functions implemented in programmable logic and also provide deterministic, high-performance, real-time DSP capabilities. Because the AI Engine kernels can be written in C/C++, this approach also delivers greater designer productivity. Signal processing and compute-intensive algorithms are well suited to run on the AI Engines.

The Versal devices provide the hardened NoC that connects these engines together providing an aggregate bandwidth of 1Tb/s+. In addition to the NoC, the massive memory bandwidth enabled by programmable logic enables programmable memory hierarchies optimized for individual computing tasks.

The memory is devised in form of a hierarchy to improve timing performance and energy consumption by exploiting temporal reusability of Convolutional Neural Network’s (CNN’s) parameters. This is achieved through small and fast buffer memories located near the Processing Elements (PEs). While one buffer supplies the PEs with data the other one prefetches the anticipated data from DRAM and vice versa. PEs also have a local memory in the form of registers to keep the current input and output data. Each operation on PE requires at least two memory read and one memory write. If all these accesses are performed directly on the off-chip DRAM memory the generated latency and the amount of consumed energy would make the accelerator not adequate to deal with high computation workload of CNN.

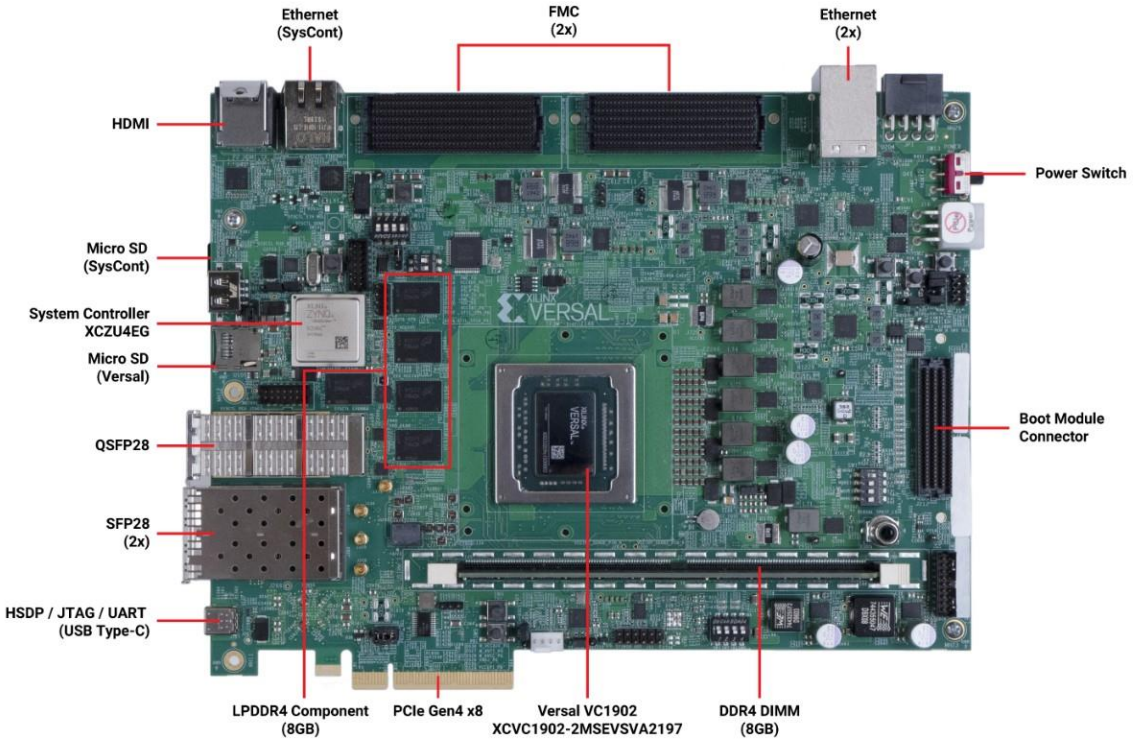

Figure 3. The Versal AI Core Series VCK190 Evaluation Kit

In general, the choice of acceleration hardware, whether PL or AI Engines, depends on the type of algorithm and data ingress and egress paths. Scalar Engines provide complex software support. Adaptable Engines provide flexible custom compute and data movement.

As VERSAL provides many different implementation possibilities, in the beginning of the project the main approach was not only to analyze, understand and determine the requirements coming from every UC, but also the proposed roadmap in order to achieve use case objectives. These inputs will potentially determine the required hardware development needed for providing cognitive awareness to FRACTAL node based on VERSAL platform. Reference architecture of a cognitive edge computing node with FRACTAL properties will be defined and a common repository of generic qualified components will be set up. Particular attention will be paid on providing flexible computing nodes, that are reusable by others and support efficiently the software by providing acceleration for the learning part.

Several acceleration approaches (e.g., approximate computing on general-purpose CPUs, GPUs, custom AI/Machine Learning (ML)-oriented accelerators on Field-Programmable Gate Array (FPGA), Component Off The Shelf (COTS) AI/ML-oriented accelerators, etc.) will be considered, evaluated and compared in order to identify the best ones for the different FRACTAL nodes also with respect to extra-functional properties (e.g., timing performance, power consumption, etc.).

The VERSAL platform provides support for integration of heterogenous compute elements with emulation and co-simulation in the Xilinx Vitis development environment. To use these features, the proper and shareable addition of FRACTAL elements like acceleration kernels in PL or the AI Engines requires packaging for this tooling. Adding such elements in the context of the tools further sets requirements for OS layer in an edge device. In the VERSAL ecosystem, services of the Xilinx runtime (XRT) are commonly used to set up and operate these accelerator components. These services restrict and define the form of the accelerators and need to be followed in the design. Also, this must be supported with insight into the application partitioning between the heterogenous compute elements in the VERSAL devices, i.e., the type of kernel and topology, e.g., for memory resources.

The complete Xilinx Versal ACAP solution extends the platform notion beyond the device capabilities. To efficiently drive applications on these devices, the platform approach also extends into the design tools that support the common project integration across these heterogenous cores. The Versal ACAP hardware and software are targeted for programming and optimization by data scientists, software and hardware developers by providing a host of tools, frameworks to start designs at any granularity. As contribution, PLC2 will provide support in this domain.

3.2 The customizable node (PULP Platform)

FRACTAL applications that need a more mature technology and SW support and need higher performance would target the Xilinx VERSAL platform. For use cases that have lower performance requirements (closer to IoT applications), the RISC-V based open-source PULP (Parallel Ultra Low Power)[2] platform provides a second and flexible architecture that can be tailored to applications. As part of the PULP platform, there are several different single, multi-core and multi-cluster systems.

A suitable instantiation of PULP that can support the functionality of the FRACTAL framework will be used as a base platform for the customizable FRACTAL node that can be enhanced by additional cores and accelerators according to the requirements of specific use cases. The implementation of the customizable node will be on a suitable FPGA prototyping board, allowing prototypes to be rapidly deployed (Figure 4).

As a basic platform, FRACTAL will use the single core PULPissimo system, but UC owners will be free to use any other implementation that fits their requirements. PULPissimo and other PULP based systems have already been implemented on a variety of FPGA based platforms, any of which can be used by the UCs.

2 https://pulp-platform.org/

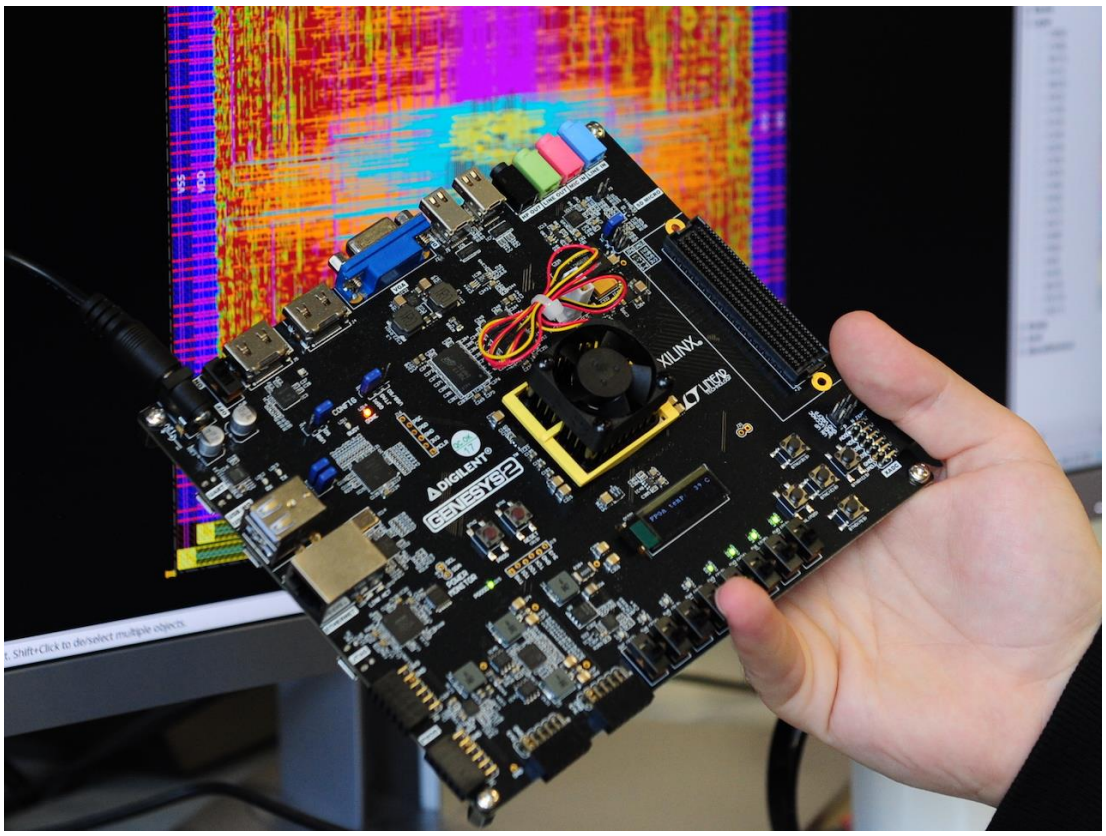

Figure 4. The Digilent Genesys 2 Xilinx FPGA board that has PULPissimo images ready to be used. The same board is also targeted by the CVA6/Ariane platform described under Section 3.3.2

The block diagram of PULPissimo is given in Figure 5. The heart of the system is a RISC-V core developed by ETH Zurich. Initially named RI5CY, this core has been adopted by the OpenHW group and has been rebranded as CV32E40P (CORE-V, 32bit, Embedded class, 4 pipeline stages with PULP extensions). The core supports the RV32IMCF extensions as well as DSP centric extensions that were developed by ETH Zurich[3]. It can use a regular RISC-V development environment for standard RISC-V instructions, and a modified compiler toolchain is needed to take advantage of the customized instructions.

3 M. Gautschi et al., "Near-Threshold RISC-V core with DSP extensions for scalable IoT endpoint devices", In Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25 (10): 2700-2713, New York, NY: IEEE, 2017. DOI: 10.1109/TVLSI.2017.2654506

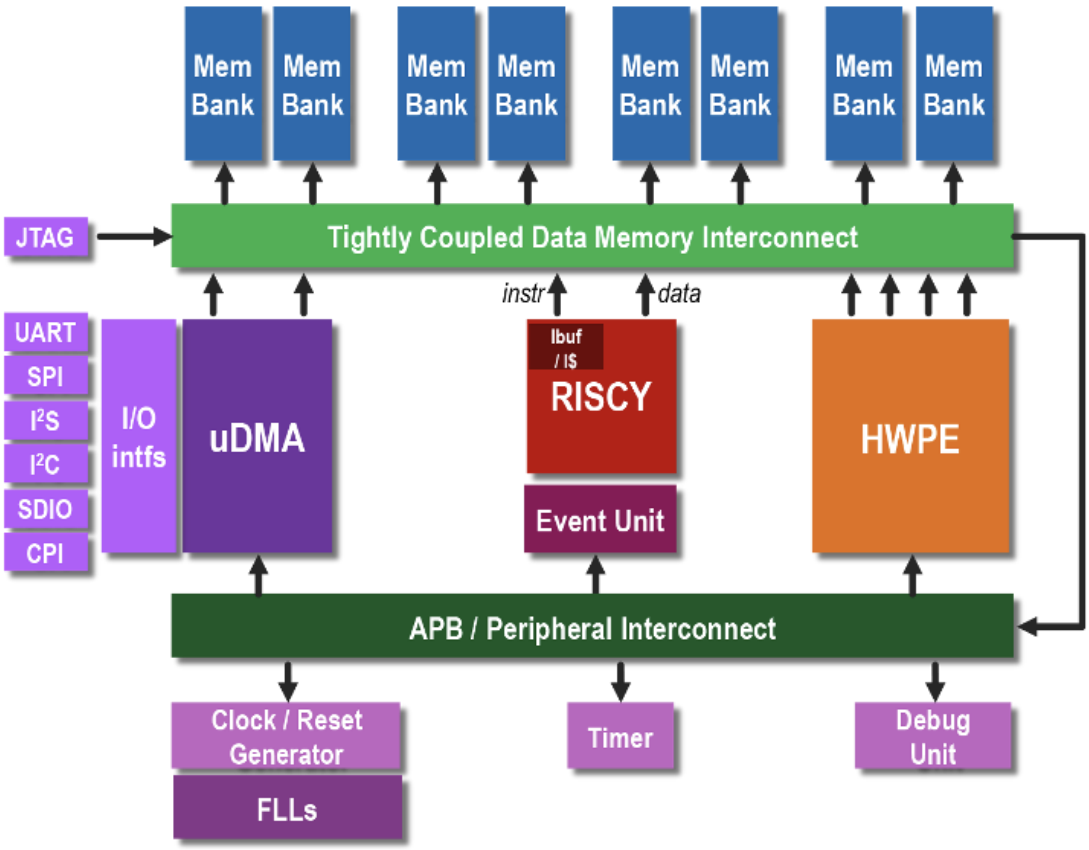

Figure 5. Block diagram of the PULPissimo system using a single 32bit RISC-V core (RI5CY/CV32E40P) and can easily be extended with accelerators, APB/AXI peripherals as well as instruction set extensions.

However, the processor core is only one part of the hardware node. There is a rich set of permissively licensed open-source peripheral components, connected over an Advanced eXtensible Interface (AXI) interface and a Direct Memory Access (DMA) that can transfer data independently between memory and peripherals.

Another important aspect of the architecture is the tightly coupled data memory interface between the processor core and the memory subsystem. The interconnect allows additional accelerators to be added (drawn in orange in Figure 5) that can access the memory the same way the processor does. This reduces the overhead of passing data between processor core and an accelerator and has proven to be extremely successful in multiple applications[4].

While this system was mainly designed to be used in an Application-Specific Integrated Circuit (ASIC) setting (and more than 10 ASICs have been manufactured and tested with a PULPissimo system), the important advantage for FRACTAL is that

4 F. Conti et al. "An IoT Endpoint System-on-Chip for Secure and Energy-Efficient Near-Sensor Analytics", In proc. IEEE Transactions on Circuits and Systems I, Regular Papers, 64 (9): 2481-2494, New York, NY: IEEE, 2017. DOI: 10.1109/TCSI.2017.269801

it has been mapped to a smaller scale Xilinx FPGA board Genesys II allowing rapid development (Figure 4). The system is also supported by a virtual platform that allows both HW and SW development to be performed using either the virtual platform, behavioral RTL simulation and FPGA emulation.

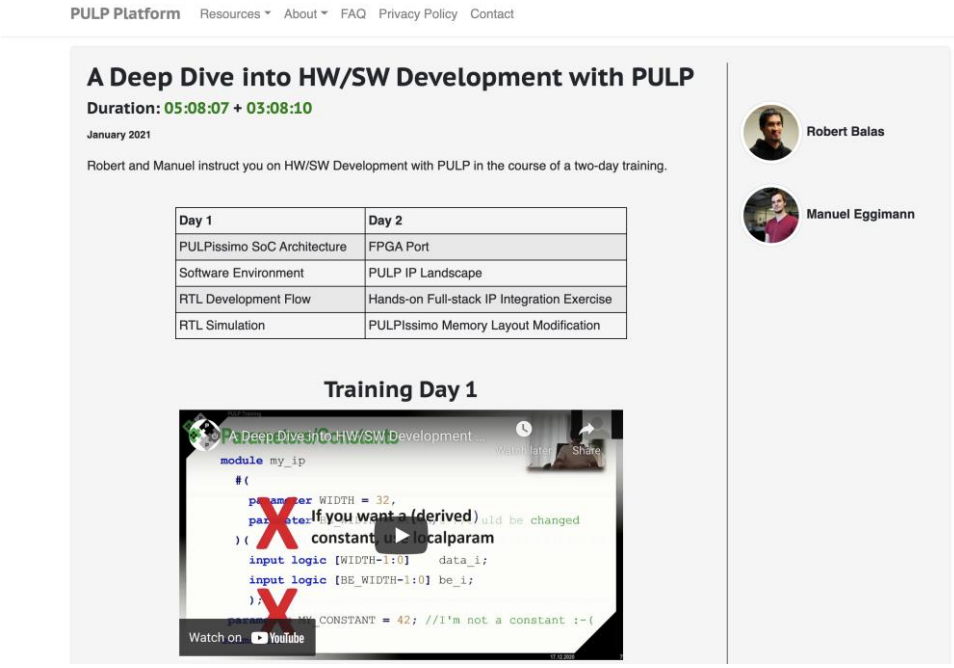

Figure 6. An excerpt from the PULP training page (accessible under https://pulp-platform.org/pulp_training.html) that shows the free and accessible tutorials on the PULPissimo platform that will be used as the customizable node within FRACTAL. This particular tutorial covers more than 8 hours of training

Since the customizable node will be prototyped on an FPGA, the node will benefit from great flexibility to choose what FPGA or set of FPGAs to use to explore an arbitrarily large node with an arbitrary set of physical resources. The customizable node will be based on a highly scalable tile-based architecture in which cores and accelerators can be interconnected in a flexible manner. Furthermore, in FRACTAL this platform will include RISC-V processor cores that are made available using a permissible open-source license removing roadblocks towards commercial exploitation as well as avoiding premature lock-in and enabling unconstrained use by the different partners in the consortium, as well as by third parties willing to use FRACTAL technology. The customizable platform will allow many different approaches to implement accelerators in the project including re-using Xilinx and other existing IPs or the use of high-level synthesis.

When compared to the commercial node, the customizable node will offer complimentary opportunities. By selecting across the FPGAs in the market, one can prototype from tiny to very large nodes. In fact, multiple FPGAs can be connected to form very powerful nodes if needed. Hence, the range of tradeoffs that can be explored is, by construction, much larger than that of the commercial node, which builds upon specific hardware resources. Additionally, high-performance features included in the commercial platform are not suitable for critical tasks which may limit

the utilization of such platforms in the context of fail-operations autonomous systems such as the autonomous car. With the customizable platform, the aim is to extend safety properties beyond relatively simple single-core nodes by incorporating the appropriate hardware support to this node.

At the same time, while the PULP platform offers a good base to start, a FRACTAL node will still have to be developed building on technology already owned by the partners. This bears certain risks in the development of such an open and flexible platform due to the uncertainties on how the final node will look like. Moreover, time-to-market is also higher since the maturity of this platform is behind that of the commercial node.

FRACTAL involves several UCs, each with different computing requirements. In the first part of the project, the goal was to understand and determine the requirements from every UC that potentially could use the experimental FRACTAL node based on RISC-V cores, as was captured in D2.1. A research system like PULPissimo differs from conventional microcontroller systems, and FRACTAL partners have requested and were given introduction talks to the PULP architecture and open-source hardware. In addition, ETHZ made more than 12 hours of video tutorials available on the PULPissimo system under:

https://pulp-platform.org/pulp_training.html

With topics covering:

- PULPissimo SoC Architecture
- PULP IP Landscape
- Hands-on Full-stack IP Integration Exercise
- RTL Development Flow
- RTL Simulation
- FPGA Port
- PULPIssimo Memory Layout Modification
- PULP SDK / GCC Compilation Toolchain

3.3 Other customizable nodes

During this study, several use cases stated the need for more traditional RISC-V based systems (capable of running single-core or SMP Linux) which resulted in some additional hardware nodes being added. These will be covered in this section.

3.3.1 NOEL-V

The NOEL-V based RISC-V platform is a multicore SoC based on the open source platform from the H2020 SELENE project[5]. The schematic of the computing part of the SoC is shown in Figure 7. As shown, the SoC includes 4 NOEL-V 64-bit cores from

[5] C. Hernàndez et al., "SELENE: Self-Monitored Dependable Platform for High-Performance Safety-Critical Systems," 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020, pp. 370-377, doi: 10.1109/DSD51259.2020.00066.

Cobham Gaisler implementing the RISC-V ISA, an AMBA AHB bus connecting the cores with the I/O, and an AXI crossbar based on the AXI PULP interconnect (https://github.com/pulp-platform/axi) to allow cores and accelerators share a DDR3 controller. An AMBA APB peripheral bus is also included for I/O device controllers like DUART, I2C, etc.

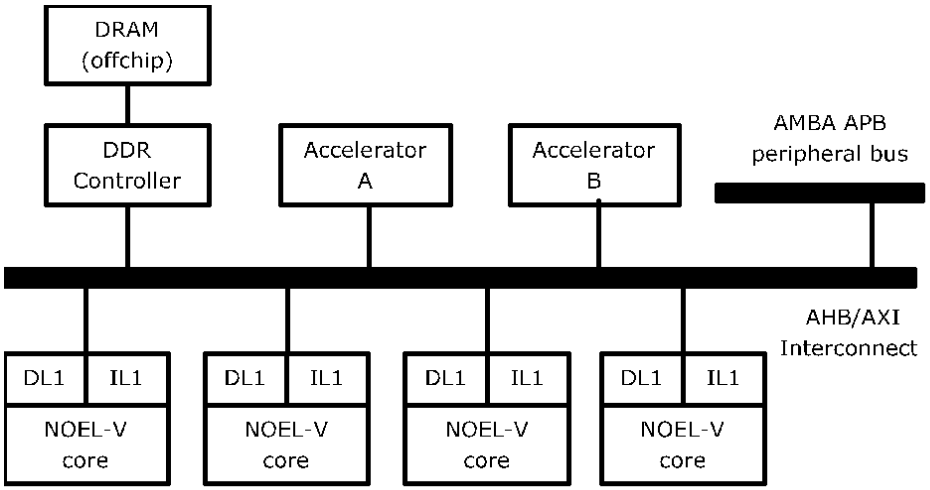

Figure 7: Schematic of the computing part of the NOEL-V based SoC

Gaisler's NOEL-V cores provide moderate performance figures despite being in-order cores building on their dual-issue 7-stages pipelined architecture with branch prediction, return address stack, write buffers, and data (DL1) and instruction (IL1) 16KB set-associative cache memories. The cores also implement integer and floating-point pipelines.

The NOEL-V based SoC can be easily extended with different accelerators and other components. Therefore, it eases the integration of features such as the AI-based accelerator from UPV, as well as BSC's statistics unit, both intended to be connected to the interconnect.

The NOEL-V SoC supports memory management units, and implements Translation Lookaside Buffers (TLBs), both for data and instructions, locally in each core. The SoC also provides support for cache coherence. Those features allow booting SMP Linux and RTEMS operating systems among others and allow sharing data across cores.

The SELENE SoC has been synthesized in a Xilinx Virtex UltraScale VCU118 FPGA and the original NOEL-V SoC is also available for the KCU115. While its primary target is the space domain, it has been retargeted to enable its use for avionics, railway and automotive applications.

3.3.2 Ariane/CVA6

As noted in Section 3.2, the open-source PULP Platform provides many different RISC-V based solutions, but the 32bit PULPissimo system was chosen as the default

configuration for the customizable node due to its simplicity and wide range of options to adapt it, including adding peripherals, adding hardware accelerators as well as adding instruction set extensions. At the same time, the PULPissimo system is too simple to run a modern Linux operating system.

To support partners that wish to use a RISC-V system with Linux support, the PULP-based Ariane system has also been considered to be used as part of the FRACTAL project. Similar to the RI5CY core used in PULPissimo, Ariane was the code name of the 64bit Linux capable core (RV64GC) developed at ETH Zurich[6]. This core has been taken over by the OpenHW group and has been renamed as CV64A6 (CORE-V, 64-bit, Application class, 6 stage pipeline).

In co-operation with OpenHW Group, Thales is developing a 32-bit version of Ariane codenamed CV32A6, also supporting Linux, and offering a reduced footprint (compared to CV64A6) and upcoming PPA optimizations for FPGA targets. CV64A6 and CV32A6 share a common RTL base and are therefore jointly referred as CVA6.

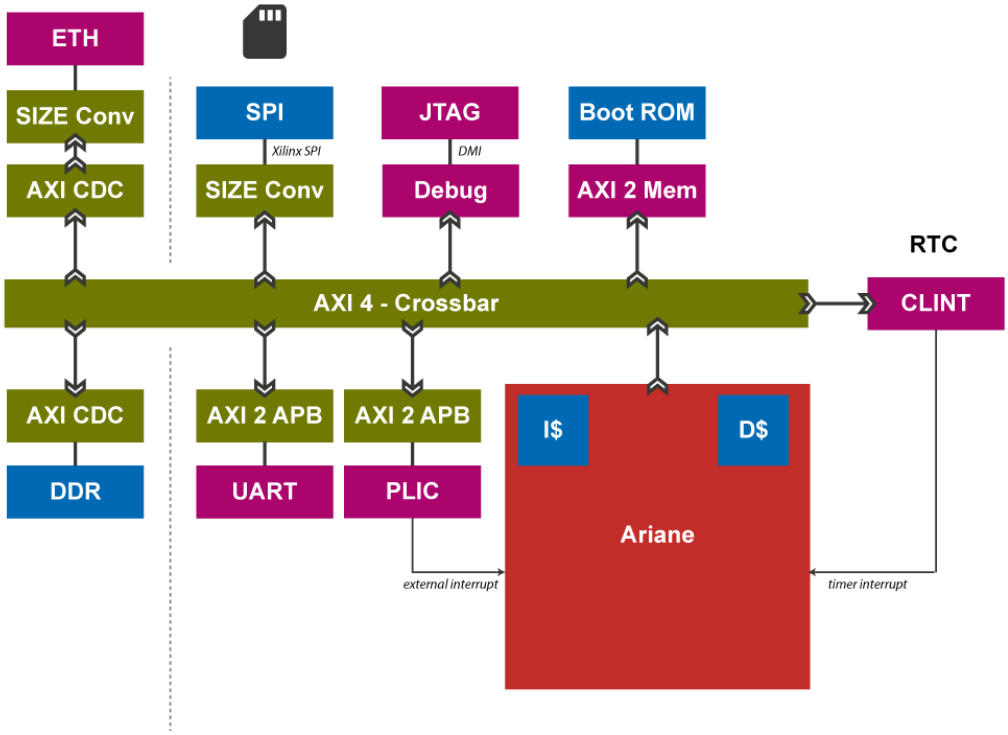

Figure 8: Schematic of the Ariane/CVA core mapped to the Genesys II board.

6 Florian Zaruba, Luca Benini, "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-ready 1.7GHz 64bit RISC-V Core in 22nm FDSOI Technology", In Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol: 27, Issue: 11, Page(s): 2629 – 2640, Nov. 2019, DOI: 10.1109/TVLSI.2019.2926114

Ariane/CVA6 comes with a mapping to the same Xilinx Genesys II board that also supports PULPissimo (Figure 8). The FPGA SoC currently contains the following peripherals:

- DDR3 memory controller
- SPI controller to connect to an SD Card
- Ethernet controller
- JTAG port with support for OpenOCD
- Bootrom containing zero stage bootloader and device tree.

And more additions are expected to come through independent contributions by both the OpenHW group and ETH Zurich developments which will be made available to FRACTAL partners as well.

4 Supporting FRACTAL developments on safety, security, low-power and cognitive awareness

4.1 Supporting FRACTAL developments on low-power

The PULP Platform is the result of the research work of FRACTAL partner ETHZ on low power architectures and as such it has been designed around principles to increase energy efficiency and reduce unnecessary power consumption. This is an ideal starting point for architectural modifications envisioned as part of WP4 to reduce power consumption of FRACTAL nodes.

The VERSAL platform relies on a centralized Platform Management Controller (PMC) for device management control functions. Power efficient designs require usage of complex system architectures with several hardware options to reduce power consumption and usage of a specialized CPU to handle all power management requests coming from multiple masters to power on, power off resources, and handle power state transitions. In addition, there are other resources like clock, reset, and pins that need to be similarly managed.

The platform management in VERSAL is available to support a flexible management control through the PMC. This platform management handles several scenarios and allows the user to execute power management decisions through its framework (equivalent to what it is done in Linux, which provides basic power management capabilities like CPU frequency scaling).

4.2 Supporting FRACTAL developments on safety

A multicore interference-aware Performance Monitor Unit (PMU) will provide safety support for verification, validation and deploying safety measures during operation. Additionally, it will allow secure access to PMU information. The PMU is an advanced statistical unit including controllability and observability channels that will be used to deal with timing interference concerns in safety-critical real-time applications on top of the PULP-related SoCs. Currently this effort is being integrated into NOEL-V platform, however the PMU is Advanced Microcontroller Bus Architecture compliant (AMBA-compliant), all registers can be accessed to be read or written through Advanced eXtensible Interface (AXI) or Advanced High-performance Bus (AHB) interfaces. The PMU is fully customizable and can be tailored to a wide variety of multicore architectures, including those based on RISC-V architecture. The PMU's control and parametrization will be done by software with an appropriate library. Some ongoing actions for the PMU are as follows:

Since the baseline platform setup is not fully complete (it will be by end of August or early September), how multicore interference manifests, and hence, how to tune some monitors of the PMU is not yet determined. This work will occur later in the year.

Additional statistics are being added providing additional information relevant for optimization of the application consolidation on top of the multicore platform. Those statistics include number of transactions and data transferred per master and slave, broken down across data sent and received.

Validation of the integration is necessarily partially complete only, and will only be fully complete once the previous steps are also complete. Software developments on the platform in WP4 can, however, proceed in parallel with validation work.

The AXI-pulp interconnect (https://github.com/pulp-platform/axi) integrated in the NOEL-V SoC will be extended to mimic the behaviour of Sifive's Worlguard specification. Worlguard restricts the access to shared resources to the masters that belong to a specific world preventing non-CPU masters, to access specific memory regions or resources extending the properties of the memory management unit to the I/O regions. To that end, every SoC request is labeled with a specific ID that defines the world that request belongs to. Only requests with a specific ID can be granted access to a specific resource.

Although, Worldguard was conceived for security purposes our goal in FRACTAL is to exploit its properties for safety as well. To support that, we plan to label SoC requests with an ID to allow improving arbitration decisions in shared resources.

- Additional safety features are actively being investigated on the Ariane/CVA6 core including:
- Ability to invalidate/flush the cache and Translation Lookaside Buffer (TLB) to return to a known state (predictability)
- Ability to define non-cacheable address chunks for peripherals (predictability)
- Ability to disable caches (higher predictability at the expense of performance)
- Ability to disable predictions, such as branch predictions (higher predictability at the expense of performance)
- Availability of local memory, such as scratchpad or locked lines/set in the L1 cache (strong real-time and higher predictability for critical processes)
- Addition of performance counters

The activities carried out by UNIMORE Within Task 3.1 are mainly focused on safety-related aspects. These activities are related to the mobility of the UC6 (intelligent totem), that represents a firm real-time UC.

Our preliminary study regards the impact on memory interference, to which the host cores (APU and RPU) of the two platforms considered (Zynq UltraScale+ and Versal ACAP) are subject, in presence of concurrent execution of memory-bound tasks on FPGA accelerators. When focusing on a FPGA-based HeSoC, a typical way of generating configurable synthetic memory traffic is that of deploying some form of traffic generators.

More in general, full-custom acceleration logic is typically designed as shown in left part of the following Figure 9, where the core acceleration logic (datapath) is coupled to some sort of data mover or DMA engine and a local memory. The datamover

leverages a Finite State Machine (FSM) or DMA controller to supervise the flow of data in and out the local memory as the datapath executes.

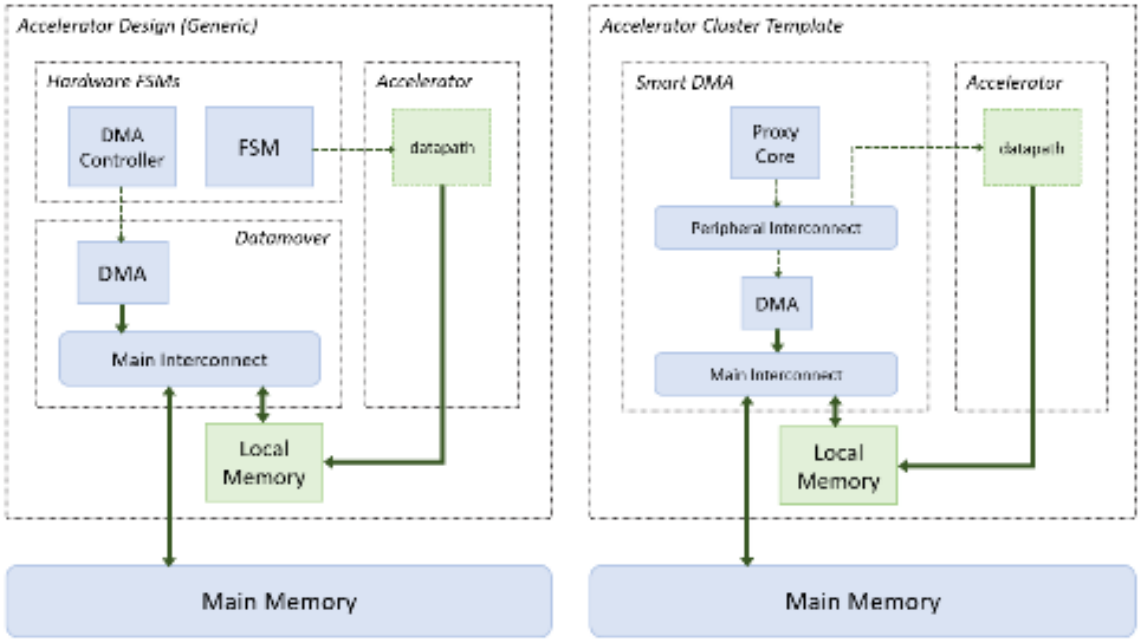

Figure 9 Accelerator Design Options

"Accelerator Cluster Template" is shown on the right of Figure 9. Here, we consider the DMA and local memory as immutable parts of an IP, the Smart DMA, that can be interfaced to different datapaths. To flexibly support different types of control logic for the DMA (and the datapath itself) we rely on a programmable core – the proxy core– rather than on custom FSM logic. This moves the control on the software side, allowing for improved flexibility and reconfigurability for different traffic patterns. The Smart DMA component allows DMA-based memory accesses in a controlled manner, which can be used to ensure that the bandwidth request generated from the FPGA accelerators does not impact the performance of other active cores/tasks in the system beyond what can be tolerated.

In this case we made use of the Smart DMA component to generate a variable amount of interference on the main memory, in order to study its behavior on the two reference platforms, Zynq UltraScale+ and Versal ACAP.

4.3 Supporting FRACTAL developments on security

As an open platform PULP based systems have a long track record in supporting research in security for evaluating changes and adaptations needed. PULPissimo based systems have already been designed to support accelerators for commonly used cryptographic primitives like AES and SHA3 and such solutions could also be

adapted to the needs of FRACTAL partners. In addition, the Ariane/CVA6 core has been used in recent work to suppress timing channels[7].

As explained in previous section, the AXI PULP interconnect will be extended to mimic the behaviour of the Worlguard specification.

The VERSAL platform supports several functionalities that could be used for device-level security such as boot image encryption and authentication. Additional functionality can be added through the Programmable Logic (PL) resources of the VERSAL platform as well.

4.4 Supporting FRACTAL developments on cognitive awareness

As an open platform, PULPissimo offers a wide range of possibilities to accelerate machine learning applications that can be used for cognitive awareness and support the software infrastructure. These include:

- Hardware accelerators connected over standard interfaces (APB/AXI).
- Hardware accelerators with shared memory access.
- Clusters of cores working as an accelerator to a main core.
- Instruction set extensions to RISC-V ISA.

These options give a lot of flexibility to FRACTAL partners that are working on solutions in this field.

One of the key features of the VERSAL platform are the AI engines that can be used to efficiently map machine learning applications. We expect that these will play a major role in implementing hardware assisted algorithms that support cognitive awareness.

As an example, an accelerator for age and gender recognition is under development to be available in the FRACTAL nodes. Both the gender and age part are based on a CNN of type VGG16. To train the model, an Adam optimizer and a binary cross-entropy loss function have been used. At current status, both CNNs have been trained using a machine equipped with a i7-9700K CPU Intel processor with 8 cores working at 3.60 GHz, a NVIDIA Titan Xp, and 32 GB or RAM. On that machine, the CNN response time has an average of 143 ms for the inference time of the gender model, and 68 ms for the age model.

7 N. Wistoff, M. Schneider, F. K. Gürkaynak, L. Benini and G. Heiser, "Microarchitectural Timing Channels and their Prevention on an Open-Source 64-bit RISC-V Core," 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021, pp. 627-632, doi: 10.23919/DATE51398.2021.9474214.

5 Interaction of UCs with FRACTAL nodes

Table 1: Use cases review (inputs and hardware architecture proposal)

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
	Drones construction	**Automotive air control**	**Smart meters**	**Object detection & recognition (box)**	**Autonomous trains**	**Smart totem**	**Autonomous robots (SPIDER)**	**Autonomous warehouses**
Inputs	Images	Time-Series	Images	Images (video streams)	Images + Videos	Video + Audio	Sensor data	Images
Architecture Proposal	VERSAL + ¿PULPissimo?	VERSAL	PULPissimo	VERSAL + ARIANE	VERSAL	VERSAL	NOEL-V	VERSAL + ¿ARIANE? ¿PULPissimo?

As part of WP2, one of the first actions of FRACTAL was to determine the use case requirements, which were detailed in D2.1. The results were then discussed internally at regular FRACTAL meetings and the following observations were made:

- **UC1** *Edge computing technologies applied for engineering and maintenance works* (PROI):

  Two separate use cases were identified, one for an autonomous drone application (UAV supervision of critical structures), and the second one involving workspace safety (Wireless Sensor Network for safety at construction sites).

  At the moment PROI is considering using both VERSAL and PULPissimo as part of its evaluation.

  UC1 (Demonstrator 1): UAV supervision of critical structures

  This demonstrator is focused on the supervision of critical facilities, where images of the structural status will be collected by the use of UAVs. The obtained images are analyzed with a visual segmentation system, so the families of cracks can be detected and categorized.

  The computational requirements of the crack detection system severely limit the hardware options to be used. Two options have been considered. First, it has been considered the use of a combination of PULPissimo as a low-end node and VERSAL as a high-end node (cloud). The PULPissimo node will perform less demanding tasks, such as image collection or detection of crack

families. The more complex tasks, such as the execution of the image segmentation system, will be delegated to the VERSAL node.

In addition, it has also been considered the usage of a medium-end more traditional node that is capable of handling the computational demand of the crack detection system, without the need of a high-performance hardware such as the one offered by VERSAL. However, this second option is still under study.

UC1 (Demonstrator 2): Wireless Sensor Network (WSN) for safety at construction sites

This demonstrator is focused on monitoring of both workforce and machinery within a construction area through the deployment of a WSN that will provide information about the status and location of the workers, the workforce in real time. This information will be managed through an IoT platform, registering possible dangers and alarms, apart from establishing a protocol in case of emergency.

The demonstrator requires wearable sensors to be sending data from the workers and machinery location, real-time processing of this data and then Machine Learning treatment for alert notifications and potential risk predictions. Although these tasks may not require really high computational capabilities, there exists a necessity of a stable OS (Linux distro preferably) which is a constraint that must be supplied by the platform. For this reason, the PULP platform may be not sufficiently adequate to fulfil the OS requirement. On the other hand, VERSAL supplies too high capabilities for the demonstrator, and not so much computational power is required, so an intermediate platform with a medium-end node is presumably going to be used. These options are still under study so that the best fitting platform is chosen for the Use Case.

- **UC2** *Automotive air path control* (AVL):
  This use case will be using VERSAL.

- **UC3** *Smart meters for everyone* (ACP):
  As a pure IoT application, PULPissimo will be used as part of this use case.

- **UC4** *Low-latency Object Detection as a generic building block for perception in the edge for Industry 4.0 applications* (SIEM):
  The use case is a vision-based object detection and recognition system. The detection, localization, and recognition of the objects is based on AI algorithm that runs on FRACTAL edge node. The part of the node responsible for the execution of AI is a special designed hardware accelerator with high computation capacity and low power consumption.
  The use case will explore and compare different acceleration options and has plans to evaluate it both on VERSAL (if it's available) and the Ariane platform.

- **UC5** *Increasing the safety of an autonomous train through AI techniques* (CAF):
  The processing requirements for this system calls for the VERSAL platform.

- **UC6** *Elaborate data collected using heterogeneous technologies - Intelligent Totem* (Aitek, Univaq, Modis, Rulex, Unimore, Unige, RoTechnology):

  The UC6 is focused on creating an infrastructure for a sentient space in a commercial mall, where fractal nodes are distributed and have the goal of supporting different types of users (e.g., buyers, vendors, commercial mall administrators) in different scenarios (...).

  Different sensors are used to collect images and sounds. The obtained data could be analyzed, for example, with people detection, idiom recognition, and age and gender recognition tasks, so that the data can be used to decide how to interact with the user supporting in solving the issue.

  These tasks are executed at the edge (on the node), both in standalone mode (all the tasks executed in the node) and fractal mode (the computational load is shared with nodes nearby).

  The computational loads conduced on considering as hardware option the usage of Versal in the nodes, associated with accelerators for the above mentioned more demanding tasks; these accelerators are going to be implemented either on the FPGA side or dedicated ASIC (off-chip connected). The nodes are able to interact to share the computational load.

- **UC7** *Autonomous robot for implementing safe movements* (VIF):
  With UC7 two driving functions of the mobile hardware-in-the-loop platform SPIDER are implemented on a Noel-V.

  The collision avoidance function and the path tracking function are both processing sensor data on the edge while exchanging data with cloud nodes. Both functions are safety relevant, thus usage of FRACTAL safety services, especially a software light-lockstep function will be integrated.

  The collision avoidance function further uses an AI function for obstacle avoidance based on a neuronal network. An AI accelerator from FRACTAL will be used to ensure the necessary latency for the computationally intensive functions.

- **UC8** *Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse* (BEE):

  The designated use case platform is VERSAL, but there are considerations in using a RISC-V based system such as Ariane and/or PULPissimo.

The list will be updated in the upcoming deliverables D3.3 and D3.5 with more up to date information.

6 Conclusions

The two main hardware nodes (commercial and customizable) are being made available to all FRACTAL partners. Following discussions regarding the design requirements, at least in the first phase of the project, it was seen that partners would benefit from additional hardware nodes that fall in between the two default options. WP3 partners are discussing in providing such solutions in agreement with other partners from technical WPs 4/5/6 as well as the UCs.

6.1 Next steps

In the first year of the project, the goal was to create an alignment between partners and allow the node supporting partners to understand the requirements as well as the use case and technical partners to see both the capabilities and limitations of the available systems. As the work in technical WP's intensifies, the interaction between the partners will increase due to the design, integration and implementation of the results of the WPs in the HW nodes presented in this deliverable.

6.2 Risks and Mitigation plans

Before the start of the project, the following hardware platform related risks had already been identified:

- FPGA based PULP platform implementations will not have the necessary performance (speed/cost/power) profile needed for use cases (Medium likelihood/Low Impact).
  For some use cases, this has been indeed the case. However, the issue was not the FPGA implementation of the platform, but the desire to have systems that are more like traditional computing systems complete with running large software ecosystems running on full fledged operating systems (i.e. Pytorch) which is beyond the practical capabilities of the system. In part this has been remedied by providing additional platforms (NOEL-V and Ariane). As the projects mature and partners become more experienced with different FRACTAL platforms, it is highly probable that more partners will make use of the experimental and customizable aspects of the provided systems.
- Diversity/maturity of tools/development environments for RISC-V systems is lower than expected (Low likelihood/Medium impact).
  As mentioned in the project proposal, the development rate of the RISC-V ecosystem is quite high, and so far this has not been a major issue.
- Unable to properly and timely integrate multiple services developed. Due to incompatibilites or services not provided by selected hardware platforms (Low likelihood/impact).
  The integration of services are still at an early stage, so it is too early to tell about the possible impact. However, the effort in WP2 has allowed partners to anticipate some of the issues, and there is good momentum in the project that leads us to believe these issues could be mastered.

- Difficulties to integrate hardware developments from different partners (Medium likelihood/impact).
  This is one of the main challenges that faces the developments around the HW platform at the moment. However, all partners are aware and are looking for solutions. In some cases, one solution will be to demonstrate technical solutions running in parts, and not in concert for a given platform, use case combination. I.e. a security service maybe demostrated on a smaller scale, allowing the use case partner to be able to judge and evaluate its impact, but the overall use case could still use a more traditional approach.

  Part of the mitigation efforts also led partners to add two additional hardware nodes, as not to spend initial efforts on porting previous work from architectures they were familiar with to the two nodes presently available.

In addition the following issues have been detected, and efforts have been put in place to mitigate the effects:

- Delivery difficulties with the VERSAL board. There is a global shortage on electronic components, and unfortunately the delivery of the development boards have been hit with longer delays than anticipated. For some of the projects that do not rely on exclusive VERSAL properties, suggestions were made to use more previous generation Xilinx MPSoC boards until the VERSAL shipments can be organized.
- Node computation demands are too high. Some partners on technical workpackages are working on solutions that require significant resources from the hardware nodes. While the VERSAL board can satisfy these requirements, both its price and its power envelope is higher than what could be expected for IoT applications.

  The addition of a lower-end (mist) node will help address this issue. The FRACTAL system will have nodes with less capabilities that defer to more capable nodes for higher complexity operations. This will allow hardware nodes within mW power envelope to be part of the FRACTAL system.

7 List of Abbreviations

ACAP	Adaptable Compute Acceleration Platform
AE	Adaptable Engines
AHB	Advanced High Performance
AIE	Artifical Intelligence Engines
AMBA	Advanced Microcontroller Bus Architecture
ASIC	Application Specific Integrated Circuit
AXI	Advanced Extensible Interface
CCIX	Cache Coherent Interface for Accelerators
CNN	Convolutional Neural Networks
COTS	Components of the Shelf
DDR	Double Data Rate
DMA	Direct Memory Access
DRAM	Dynamic Random-Access Memory
DSP	Digital Signal Processing
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GPU	Graphical Processing Unit
HeSoC	Heterogenous System on Chip
HW	Hardware
I/O	Input / Output
IoT	Internet of Things
ISA	Instruction Set Architecture
ML	Machine Learning
MPSoC	Multi-Processor, System on Chip
NoC	Network on Chip
OS	Operating System
PCIe	Peripheral Component Interconnect Express
PE	Processing Element
PL	Programmable Logic
PMU	Performance Monitoring Unit
PPA	Power Performance Area
PS	Processing System
PULP	Parallel Ultra Low Power
RTL	Register Transfer Level
SDK	Software Development Kit
SIMD	Single Instruction Multiple Data
SW	Software
TLB	Translation Lookaside Buffer
UC	Use Case
VLIW	Very Large Instruction Word
WP	Work Package
XRT	Xilinx Runtime

8 List of figures

9 List of tables