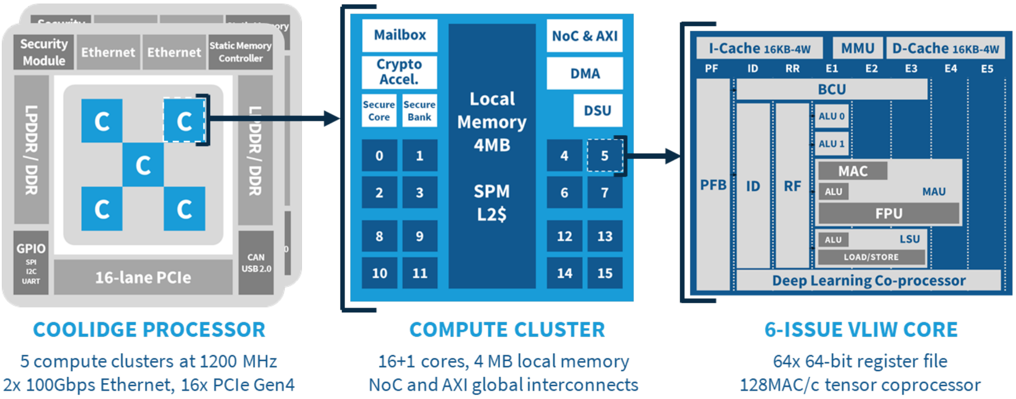
Consolidating High-Integrity, High-Performance, and Cyber-Security Functions on a Manycore Processor

Benoît Dupont de Dinechin
Kalray S.A.
benoit.dinechin@kalray.eu



COOLIDGE PROCESSOR
5 compute clusters at 1200 MHz
2x 100Gbps Ethernet, 16x PCIe Gen4

COMPUTE CLUSTER
16+1 cores, 4 MB local memory
NoC and AXI global interconnects

6-ISSUE VLIW CORE
64x 64-bit register file
128MAC/c tensor coprocessor

Figure 1: Overview of the MPPA3 processor.

ABSTRACT

The requirement of high performance computing at low power can be met by the parallel execution of an application on a possibly large number of programmable cores. However, the lack of accurate timing properties may prevent parallel execution from being applicable to time-critical applications. This problem has been addressed by suitably designing the architecture, implementation, and programming models, of the Kalray MPPA (Multi-Purpose Processor Array) family of single-chip many-core processors. We introduce the third-generation MPPA processor, whose key features are motivated by the high-performance and high-integrity functions of automated vehicles. High-performance computing functions, represented by deep learning inference and by computer vision, need to execute under soft real-time constraints. High-integrity functions are developed under model-based design, and must meet hard real-time constraints. Finally, the third-generation MPPA processor integrates a hardware root of trust, and its security architecture is able to support a security kernel for implementing the trusted execution environment functions required by applications.

CCS CONCEPTS

• **Computer systems organization** → **Multicore architectures**; **Heterogeneous (hybrid) systems**; **System on a chip**; *Real-time languages*.

KEYWORDS

manycore processor, cyber-physical system, dependable computing

ACM Reference Format:
Benoît Dupont de Dinechin. 2019. Consolidating High-Integrity, High-Performance, and Cyber-Security Functions on a Manycore Processor. In *The 56th Annual Design Automation Conference 2019 (DAC '19), June 2–6, 2019, Las Vegas, NV, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3316781.3323473

1 INTRODUCTION

Cyber-physical systems are characterized by software that interacts with the physical world, often with timing-sensitive safety-critical physical sensing and actuation [10]. Emerging applications such as aircraft pilot support or automated driving systems require more than what classic cyber-physical systems (CPSs) can provide. In particular, application functionality relies on pervasive application of machine learning techniques, while the cyber-security requirements have become significantly more stringent. We refer to the CPSs enhanced with advanced machine learning capabilities and strong cyber-security support as "intelligent systems".

Given the state of CMOS computing technology [8], providing the processing performances required by intelligent systems while meeting the Size, Weight, and Power (SWaP) constraints of embedded systems can only be achieved by parallel computing and by the specialization of processing elements. For instance, automated driving systems around year 2022 are estimated to require over 100

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06...$15.00
https://doi.org/10.1145/3316781.3323473

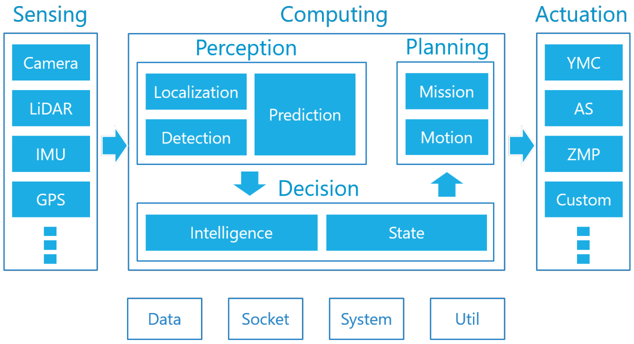


Figure 2: Autoware automated driving system functions.

TOPS of deep learning inference in the vehicle perception functions, while the motion planning functions would require more than 50 FP32 TFLOPS (Figure 2).

In order to address the challenges of high-performance embedded computing with time-predictability, Kalray has been refining a manycore architecture called MPPA (Massively Parallel Processor Array) across three generations. The first-generation MPPA processor was primarily targeting accelerated computing [2], but implemented the first key architectural features for time-critical computing [4]. Kalray further improved the second-generation MPPA processor for better time-predictability [12], providing an excellent target for model-based code generation [11] and enabling accurate analysis of the network-on-chip (NoC) service guarantees through a Deterministic Network Calculus formulation [3].

In this report, we introduce the third-generation MPPA processor manufactured in 16FFC CMOS technology, whose manycore architecture is significantly improved over the previous ones in the areas of performance, programmability, functional safety, and cyber-security. These features are motivated by application cases in defense, avionics and automotive where the high-performance, high-integrity, and cyber-security functions must be consolidated onto a single or dual processor configuration. In Section 2, we introduce the target applications and discuss how existing manycore computing platforms appear unsatisfactory. In Section 3, we present the main features of the MPPA processor in relation with the target application requirements.

2 MOTIVATIONS

2.1 Target Applications

The third-generation MPPA (MPPA3) processor is designed for building embedded intelligent systems in the areas of defense, avionics, and automotive. Applications in these areas rely on some form of dependable computing, that is, the ability to achieve prescribed levels of reliability, availability, functional safety, and cyber-security. They also require that the different application components operate at different levels of functional safety and cyber-security.

In case of automated driving applications (Figure 2), the perception and the decision functions require high performances that can only be met with parallel computing techniques. These techniques entails significant execution resource sharing, which negatively impacts time predictability [13]. As a result, the functional safety of these perception and decision functions targets ISO 26262 ASIL-B. Conversely, vehicle control algorithms, as well as sensor & actuator management, must be delegated to micro-controller processors specifically designed to host ASIL-D functions.

	Defense	**Avionics**	**Automotive**
Hardware root of trust	✓	✓	✓
Physical attack protection	✓		✓
Encrypted boot firmware	✓	✓	
Application code decryption	✓	✓	✓
Event data record encryption		✓	✓

Table 1: Security requirements by application area.

Similarly, a targeted unmanned aerial vehicle application is composed of two functional domains, one being safety-critical and the other non safety-critical. The two domains are segregated by physical isolation mechanisms, which ensures no execution resources can be shared between them. The safety-critical domain hosts the trajectory control partition (DO-178C DAL-A/B), the detection and avoidance partition (DAL-C), and payload applications such as video streaming (DAL-C). The non-critical domain hosts a data management partition (DAL-E) and a secured communication partition (ED-202 SAL-3). Of interest is the fact that the secured partition is located in the non safety-critical domain, as the availability requirements of functional safety appear incompatible with the integrity requirements of cyber-security.

Finally, embedded applications in the areas of defense, avionics, and automotive have common requirements in the area of cyber-security (Table 1). The main one is the availability of a hardware root of trust (RoT), that is, a secured component that can be inherently trusted. Such RoT can be provided as an external hardware security module (HSM), or integrated as a central security module (CSM). In both cases, this security module maintains the critical security parameters (CPS) such as public authentication keys, device identity and master encryption keys in non-volatile secured area. The security module embeds a TRNG, hashing, symmetric and public-key cryptographic accelerators in order to extend the chain of trust, starting with boot firmware signature verification.

2.2 Manycore Processors

A multicore processor refers to a computing device that contains multiple software programmable processing units (cores with caches). Multicore processors deployed in desktop computers or datacenters feature homogeneous cores, and a memory hierarchy composed of coherent caches. Conversely, a manycore processor can be characterized by the architecturally visible grouping of cores inside compute units: cache coherence may not extend beyond the compute unit, or the compute unit may provide scratch-pad memory. A multicore processor scales by replicating its cores, while a manycore processor scales by replicating its compute units. A manycore architecture may thus be scaled to 100s if not 1000s of cores.

The GPGPU architecture introduced by NVidia with Fermi is a mainstream manycore architecture, whose compute units are called Streaming Multiprocessors (SMs). Each SM comprises 32 Streaming Cores (SCs) that share a local memory, caches and a global memory system. Threads are scheduled and executed atomically by 'warps', where SCs execute the same instruction or are inactive at any given time. Hardware multithreading enables warp execution switching on each cycle, helping cover the external memory access latencies. The GPGPU architecture has further evolved with the NVidia Volta by integrating 8 ’tensor cores’ per SM, in order to accelerate machine learning workloads.

Although the embedded GPGPU processors provide adequate performance and energy efficiency for accelerated computing in intelligent systems, their architecture carry inherent limitation with regards to time-predictability. The first is related to memory coalescing, that is, the automatic hardware grouping of the memory accesses issued by a warp into a reduced number cache blocks. Success of memory coalescing is dependent on run-time addresses and has a significant impact on performances. Second, the ’thread blocks’ allocation to SMs is performed at run-time, while the warps are executed out-of-order inside a SM. Finally, GPGPUs must be programmed in a restricted if not proprietary software environment, whose run-time is essentially not time-predictable.

3 MPPA3 PROCESSOR

3.1 Architecture Overview

The design objectives of the MPPA processors are to combine the performance scalability of GPGPUs, the timing-predictability of DSP cores, and the I/O capabilities of FPGA devices. Software development tools and run-time environments must conform to CPU standards, specifically the availability of POSIX operating systems, RTOSes, and of C/C++/OpenMP programming environments. As these standards are restricted to a multicore shared memory architecture, there is a need to provide higher-level application code generators that automate code and data distribution across the multiple compute units and their local memories. This is applicable in particular to computer vision thanks to the OpenVX environment [7], to deep learning inference when starting from standard description of trained networks, and to model-based software development using synchronous-reactive languages [5].

The MPPA3 processor architecture (Figure 1) applies the defining principle of manycore architectures: processing elements are regrouped with a multi-banked local memory and a slice of the memory hierarchy into compute units, which share a global interconnect and access to external memory. The key differentiation of the MPPA manycore architecture is the integration of fully software-programmable cores for the processing elements, and the provision of a RDMA engine in each compute unit.

The cores implement a 64-bit, 6-issue VLIW architecture, which is an effective way to design instruction-level parallel cores targeting numerical, signal and image processing applications. The implementation of this VLIW core and its caches ensure that the resulting processing element is fully timing compositional, a critical property with regards to computing accurate bounds on the worst-case response time (WCRT) [9]. Each VLIW core is also paired with a tightly-coupled coprocessor for the mixed-precision tensor operations of deep learning inference.

3.2 Global Architecture

The structuring of the MPPA3 architecture into a collection of compute units, each comparable to an embedded multicore processor, is the main feature that enables the consolidation of application partitions operating at different levels of functional safety and cyber-security on a single processor. This feature requires that global interconnects with support for partition isolation be provided. From experience with previous MPPA processors, it became apparent that interconnects implemented as ’network-on-chip’ (NoC) may be specialized for different purposes: generalization of busses, or integration of macro-networks (Table 2).

Generalized busses	Integrated macro-network
Connectionless	Connexion-oriented
Address-based transaction	Stream-based transactions
Flit-level flow-control	[End-to-end flow control]
Implicit packet routing	Explicit packet routing
Inside coherent address space	Across address spaces (RDMA)
Coherency protocol messages	Message multicasting
Reliable communication	[Packet loss or reordering]
QoS by priority and ageing	QoS by traffic shaping
Coodination with DDR memory controller scheduling	Termination of macro-networks (Ethernet)

Table 2: Types of network-on-chip interconnects.

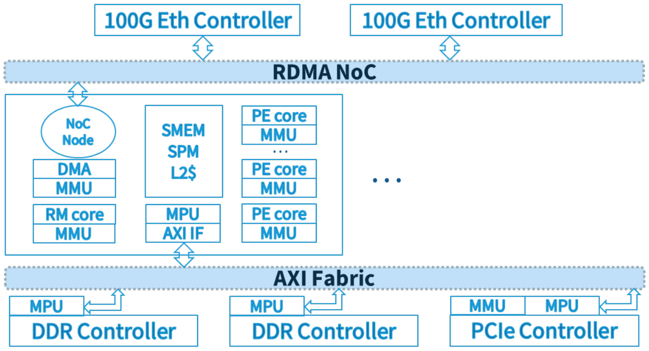


Figure 3: Global interconnects of the MPPA3 processor.

Accordingly, the MPPA3 processor is fitted with two global interconnects, respectively identified as ’RDMA NoC’ and ’AXI Fabric’ (Figure 3). The ’RDMA NoC’ is a wormhole switching network-on-chip designed to terminate two 100Gbps Ethernet controllers, and to carry the remote DMA operations found in supercomputer interconnects or communication libraries such as SHMEM [6]. The ’AXI Fabric’ is a crossbar of busses with round-robin arbiters that connects the compute clusters, the external DDR memory controllers, the PCIe controllers, and other I/O controllers.

Based on this global architecture, the consolidation of application functions operating at different levels of functional safety and cyber-security is supported by two mechanisms.

- Cores and other bus initiators have their address translated from virtual to machine addresses by memory management units (MMUs). These MMUs actually implement a double translation: from virtual to physical, as directed by the operating system or the execution environment; from physical to machine, under the control of a partition monitor operating at hypervisor privilege level. This first mechanism supports

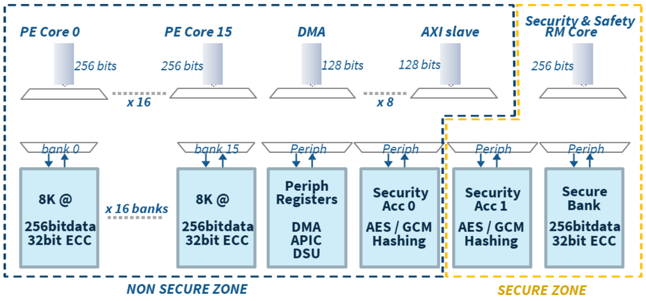

PE Core 0
PE Core 15
DMA
AXI slave
Security & Safety RM Core
256 bits
256 bits
128 bits
128 bits
256 bits
x 16
x 8
bank 0
bank 15
Periph
Periph
Periph
Periph
8K @ 256bitdata 32bit ECC
x 16 banks
8K @ 256bitdata 32bit ECC
Periph Registers DMA APIC DSU
Security Acc 0 AES / GCM Hashing
Security Acc 1 AES / GCM Hashing
Secure Bank 256bitdata 32bit ECC
NON SECURE ZONE
SECURE ZONE

Figure 4: Local interconnects of the MPPA3 processor.

the requirements of isolating safety-critical application partitions in multicore processors [1].

- Memory protection units (MPUs) are provided on the AXI Fabric targets to filter transactions based on their machine addresses. Similarly, selected NoC router links can be disabled. This second mechanism has its parameters set at boot time, and then cannot be overridden without resetting the processor. This is used to partition the processor and its peripherals into physically isolated domains, as in the unmanned aerial vehicle applications discussed in Section 2.

3.3 Compute Cluster

The compute unit of the MPPA3 processor, called compute cluster, is structured around a local interconnect (Figure 4) and comprises a secure zone and a non-secure zone.

The secure zone contains a security & safety management core (RM), a 256KB secure memory bank, and a dedicated cryptographic accelerator. The RM core of each compute cluster is also connected to the processor central security module. The purpose of the secure zone is to host a trusted execution environment (TEE), and a runtime system that performs on-demand code decryption for the applications that require it (Table 1).

The non-secure zone contains 16 application cores (PE), 16 memory banks of 32-byte words totalling 4MB of local memory, a DMA engine, a cryptographic accelerator, and cluster-local peripheral control registers. The non-secure zone of the MPPA3 compute cluster supports two types of execution environments:

- A symmetric multi-processing (SMP) environment, exposed through the standard POSIX multi-threading (supporting OpenMP in C/C++ compilers) and file system APIs. In this environment targeting high-performance computing under soft real-time constraints, all the core L1 data caches are kept coherent, and the 4MB local memory is partitioned between a shared L2 cache and scratch-pad memory (SPM).
- An asymmetric multi-processing (AMP) environment, seen as a collection of 16 cores where each is executing under a RTOS and is associated with one particular bank (256KB) of the local memory. In this environment targeting high-integrity computing under hard real-time constraints, L1 cache coherence is disabled, and the local memory is configured as scratch-pad memory only.

4 CONCLUSIONS

We introduced the MPPA3 processor, which implements a manycore architecture targeting 'intelligent systems' defined as cyber-physical systems enhanced with advanced machine learning capabilities and strong cyber-security support. Like the GPGPU architecture, the MPPA3 architecture is composed of a number of multicore compute units sharing the processor external memory and I/O through on-chip global interconnects. However, the MPPA architecture is able to host standard software, offers excellent time predictability, and provides strong partitioning capabilities. This enables to consolidate the high-integrity functions developed through model-based design, the high-performance functions implied by vehicle perception, and the cyber-security functions of secured communications, on units or tandems of MPPA3 processors.

REFERENCES

[1] Certification Authorities Software Team (CAST). 2016. *Multi-core Processors.* Technical Report CAST-32A. FAA.

[2] Benoît Dupont de Dinechin, Renaud Ayrignac, Pierre-Edouard Beaucamps, Patrice Couvert, Benoit Ganne, Pierre Guironnet de Massas, François Jacquet, Samuel Jones, Nicolas Morey Chaisemartin, Frédéric Riss, and Thierry Strudel. 2013. A Clustered Manycore Processor Architecture for Embedded and Accelerated Applications. In *IEEE High Performance Extreme Computing Conference, HPEC 2013, Waltham, MA, USA, September 10-12, 2013.* 1–6.

[3] Benoît Dupont de Dinechin and Amaury Graillat. 2017. Feed-Forward Routing for the Wormhole Switching Network-on-Chip of the Kalray MPPA2 Processor. In *Proceedings of the 10th International Workshop on Network on Chip Architectures, NoCArc/MICRO 2017, Cambridge, MA, USA, October 14-18, 2017.* 10:1–10:6.

[4] Benoît Dupont de Dinechin, Duco van Amstel, Marc Poulhiès, and Guillaume Lager. 2014. Time-Critical Computing on a Single-Chip Massively Parallel Processor. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014.* 1–6.

[5] Amaury Graillat, Matthieu Moy, Pascal Raymond, and Benoît Dupont de Dinechin. 2018. Parallel Code Generation of Synchronous Programs for a Many-Core Architecture. In *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018.* 1139–1142.

[6] Julien Hascoët, Benoît Dupont de Dinechin, Pierre Guironnet de Massas, and Minh Quan Ho. 2017. Asynchronous One-Sided Communications and Synchronizations for a Clustered Manycore Processor. In *Proceedings of the 15th IEEE/ACM Symposium on Embedded Systems for Real-Time Multimedia, ESTImedia 2017, Seoul, Republic of Korea, October 15 - 20, 2017.* 51–60.

[7] Julien Hascoet, Benoît Dupont de Dinechin, Karol Desnos, and Jean-François Nezan. 2018. A Distributed Framework for Low-Latency OpenVX over the RDMA NoC of a Clustered Manycore. In *2018 IEEE High Performance Extreme Computing Conference, HPEC 2018, Waltham, MA, USA, September 25-27, 2018.* 1–7.

[8] Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Ahmed Hemani, Axel Jantsch, and Hannu Tenhunen. 2017. *A Perspective on Dark Silicon.* Springer International Publishing, 3–20.

[9] Daniel Kästner, Markus Pister, Gernot Gebhard, Marc Schlickling, and Christian Ferdinand. 2013. Confidence in Timing. In *SAFECOMP 2013 - Workshop SASSUR (Next Generation of System Assurance Approaches for Safety-Critical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 2013.*

[10] Edward A. Lee, Jan Reineke, and Michael Zimmer. 2017. Abstract PRET Machines. In *2017 IEEE Real-Time Systems Symposium, RTSS 2017, Paris, France, December 5-8, 2017.* 1–11.

[11] Quentin Perret, Pascal Maurère, Eric Noulard, Claire Pagetti, Pascal Sainrat, and Benoit Triquet. 2016. Temporal Isolation of Hard Real-Time Applications on Many-Core Processors. In *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Vienna, Austria, April 11-14, 2016.* 37–47.

[12] Selma Saidi, Rolf Ernst, Sascha Uhrig, Henrik Theiling, and Benoît Dupont de Dinechin. 2015. The Shift to Multicores in Real-Time and Safety-Critical Systems. In *2015 International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2015, Amsterdam, Netherlands, October 4-9, 2015.* 220–229.

[13] Reinhard Wilhelm and Jan Reineke. 2012. Embedded systems: Many cores - Many problems. In *7th IEEE International Symposium on Industrial Embedded Systems, SIES 2012, Karlsruhe, Germany, June 20-22, 2012.* 176–180.