

A Survey of Hardware Technologies for Mixed-Critical Integration Explored in the Project *EMC*²

Haris Isakovic¹(✉), Radu Grosu¹, Denise Ratasich¹, Jiri Kadlec²,
Zdenek Pohl², Steve Kerrison³, Kyriakos Georgiou³, Kerstin Eder³,
Norbert Druml⁴, Lillian Tadros⁵, Flemming Christensen⁶, Emilie Wheatley⁶,
Bastian Farkas⁷, Rolf Meyer⁷, and Mladen Berekovic⁷

¹ Institute of Computer Engineering, Vienna University of Technology,
Vienna, Austria

{haris.isakovic,radu.grosu,denise.ratasich}@tuwien.ac.at

² Institute of Information Theory and Automation, Prag, Czech Republic
{kadlec,xpohl}@utia.cas.cz

³ Department of Computer Science, University of Bristol, Bristol, UK
{steve.kerrison,kyriakos.georgiou,kerstin.eder}@bristol.ac.uk

⁴ Infineon Technologies Austria AG, Graz, Austria
norbert.druml@infineon.com

⁵ Technische Universität Dortmund, Dortmund, Germany
lillian.tadros@tu-dortmund.de

⁶ Sundance Multiprocessor Technology, Chesham, UK
{flemming.c,emilie.w}@sundance.com

⁷ Technische Universität Braunschweig, Braunschweig, Germany
{farkas,meyer,berekovic}@c3e.cs.tu-bs.de

Abstract. In the sandbox world of cyber-physical systems and internet-of-things a number of applications is only eclipsed by a number of products that provide solutions for specific problem or set of problems. Initiatives like the European project *EMC*² serve as cross-disciplinary incubators for novel technologies and fuse them together with state-of-the-art industrial applications. This paper reflects on challenges in scope of hardware architectures and related technologies. It also provides a short overview of several technologies explored in the project that provide bridging solutions for these problems.

1 Introduction

Cyber-physical systems (CPS) integrate computation with the physical environment [9]. A vast number of systems can be classified as CPS in various domains of implementation (e.g., consumer electronics, automotive, space, avionics) including multiple disciplines (e.g., computer science, electrical engineering, mechanical engineering, biology, chemistry). The concept of CPS was introduced to unite diverging disciplines and establish a fundamental set of rules and methodologies for the design and development of these systems. The project *EMC*² (Embedded

Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments) explores different aspects of the design and development of CPS in an interdisciplinary and cross-domain approach. The goal is to unify the design process from requirements and specification phase to the validation and verification phase, such that individual processes of different disciplines are merged into a single process.

Establishing a chain of multi-disciplinary links is a highly complex task. Commonly, each discipline would provide their respective part of the system which is further integrated with the rest of the system. The goal of the CPS paradigm is to establish common guideline for multiple disciplines to co-exist and co-develop a system together. The project EMC² represents an incubator of different technologies which strive towards common goal of bridging gaps in design and implementation process CPSs. A common example is the mixed-criticality integration issue addressed in Sect. 2.

The paper provides a short overview of the major challenges related to hardware technologies and mixed-critical applications (see Sect. 2) and some of the techniques explored in the project EMC² to resolve them. Section 3 describes seven different technologies explored in the scope of the project. Section 4 provides a reflection on the effect of the technologies on the presented challenges and industrial applications. Section 5 concludes the paper.

2 Background

2.1 Mixed-Criticality Integration

Two basic conditions for the mixed-criticality integration are spatial and temporal isolation [18]. These emerging properties depend on a series of interlocked architectural properties. Respectively they represent abilities to separate applications in a system both in space and time. A spatial isolation represents distribution of system resources (e.g., memory, IO) among applications without interference between individual applications. A temporal isolation allows deterministic execution and interaction off applications without overlapping and interference. The architectural structure of the system dictates whether these properties can be achieved in hardware or in software. On conventional single-core and multi-core architectures individual applications share resources and the distribution of resources is administered by system software. However, this task is extremely simpler on a single-core processor than on the architectures with multiple processors that share multiple resources.

2.2 Performance

Main driver behind advance of general purpose computing was performance. This progression was accurately predicted by Moore's Law [14] is reaching its limits. The performance depends more and more on core and thread multiplication, rather than increase of frequency. Industrial applications require must

conform to various standards and must be certified as such. A switch from single-core processors to multi-core processors for safety critical or real-time applications is an uphill battle. The COTS multi-core hardware architectures are non-deterministic and can be certified for safety-critical and real-time applications only in single-core operation mode.

2.3 Power Consumption

Optimizing for power and energy consumption has mainly been done through hardware innovation. Currently, there are no appropriate tools that can provide feedback to the software developer on how their programming choices affect the energy consumption of a system. Such feedback could help programmers, toolchains and runtime systems to utilize the available energy-saving hardware capabilities and meet strict energy budgets. Hardware energy measurements are difficult to use within the software development process and are usually insufficient for providing fine-grained energy consumption attribution to various software components. New techniques are needed that can estimate the energy consumption of software without any hardware energy measurements. These techniques must be easily integrated into existing toolchains to lift energy consumption information from the hardware, through the different software abstraction layers, and up to the programmer.

2.4 Verification

To ensure correct behaviour, systems must be verified. The complexity of EMC²-type systems poses verification challenges, where test-based verification can easily miss bugs and formal verification can require an infeasible amount of resources. The MCENoC, discussed in Sect. 3.4, addresses this complexity by building a predictable network from simple, repeated logic components, where certain behaviours are already mathematically proven. Non-functional properties, such as total energy consumption, may also need verification. This is particularly challenging in a software context, where better tools for estimating the energy consumption of code are needed.

3 Survey of Hardware Technologies in EMC²

3.1 Asymmetric Multiprocessing with Video Processing for EMC²-DP-V2 Platform

Video processing and very fast digital I/O requires processing based on combination of standard processors and HW acceleration blocks in programmable logic. The Xilinx 28 nm Zynq devices contain two 32bit ARM Cortex A9 processors and programmable logic on a single chip. We describe accelerator designs for a standalone EMC²-DP-V2 platform developed in the EMC² project in the Xilinx SDSoC 2015.4 environment [22].

Development Environment and the Board Support Package for the EMC2-DP-V2. UTIA developed board support package for the SDSoC compiler with support for Full HD I/Os and the asymmetric multiprocessing. The video processing algorithms have been modelled and debugged on ARM A9 processor in C. Some user-defined SW functions have been compiled by the SDSoC compiler to HW accelerator blocks together with the corresponding DMA data movers. Figure 1 presents example of video processing HW generated in the SDSoC.

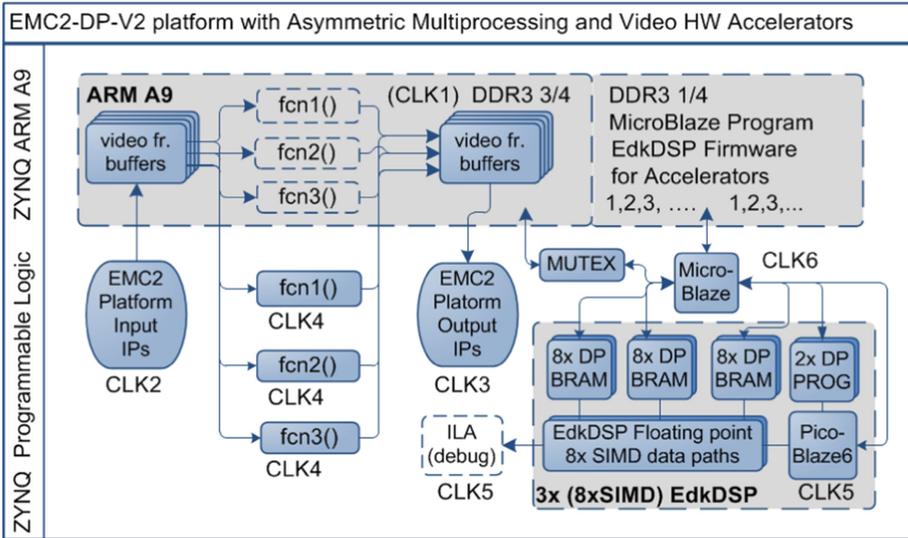


Fig. 1. AMP system with $3 \times (8xSIMD)$ EdkDSP and Full HD HDMII-HDMI for EMC2-DP-V2

The developed board support package supports the asymmetric multiprocessing of the ARM Cortex A9 processor with the MicroBlaze processor and three runtime reprogrammable floating point accelerators. Each EdkDSP accelerator consists of a vector floating point unit and an reprogrammable sequencer. The accelerators perform sequences of vector operations in 8xSIMD floating point data paths. The run-time reconfiguration is performed by reprogramming of each sequencer.

Parameters of EMC2-DP-V2 Platform. The board can be fitted with two supported system-on-modules. Clocks $CLK1 \dots CLK6$ are specified in Table 1 and Fig. 1. Measured performance for both supported modules is summarized in Table 2.

In case of acceleration of a motion detection algorithm, the total energy of standalone EMC2-DP-V2 board needed for processing of each Full HD frame has

Table 1. Clock frequencies

Device	CLK1	CLK2	CLK3	CLK4	CLK5	CLK6
	MHz	MHz	MHz	MHz	MHz	MHz
7z030-1I	666.7	148.5	148.5	150.0	150	125
7z030-3E	1000	148.5	148.5	200.0	200.0	142.8

Table 2. Performance

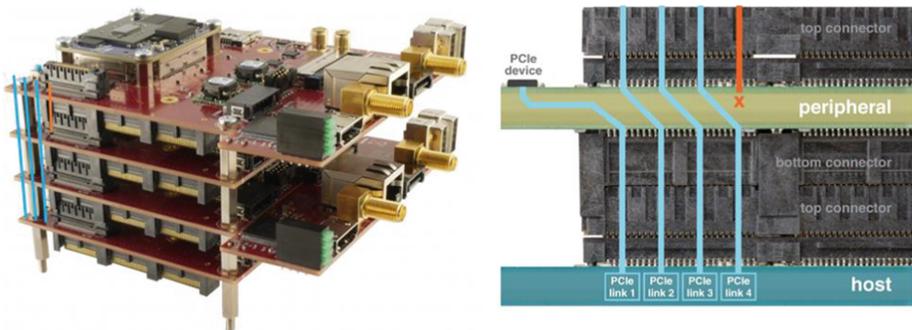
Video algorithms in Full HD	7z030-1I	7z030-3E
1× Sobel edge detection (FPS)	49.7	57.7
1× Motion detection (FPS)	42.8	51.8
Motion detection HW/SW acceleration	36×	30×
LMS adaptive filter (GFLOP/s)	0.914	1.189
FIR filter (GFLOP/s)	1.419	1.798
Peak: 3× (8xSIMD) EdkDSP (GFLOP/s)	7.2	9.6

been reduced from 6065 mJ/FRAME (ARM SW) to 177 mJ/FRAME (this is 34× for the slower module) and from 4113 mJ/FRAME to 149 mJ/FRAME (28× for the faster module). Application notes and evaluation packages describing these designs are publicly accessible from [21].

3.2 Multicore Stack Using the $EMC^2 - DP$

The EMC^2 -DP, a PCIe/104 FMC carrier developed by Sundance, can be used as a stand-alone module (like in Fig. 1), but it was really intended for large-scale, stacked multiprocessing ARM/FPGA systems (see Fig. 2).

The EMC^2 -DP integrates an on-board PCI Express switch allowing an infinite number of EMC^2 -DP to be stacked and therefore providing large I/O solutions.

**Fig. 2.** EMC^2 -DP stack and PCI Express switch

The PCI Express switch also provides high-speed communications between each *EMC*²-DP. Moreover the *EMC*²-DP can be expanded with a VITA57.1 FMC-LPC compatible Daughter Cards for I/O expansion from the FPGA fabric. The *EMC*²-DP is a versatile board that can be used for various commercial, medical, industrial and military applications.

The host communicates with the FPGA modules with the PCI Express driver on Windows 7 64-bit. Each board appears as a separate device in Windows, and has its own PCI express hardware link (see Fig. 2). It is thus possible for several transfers between the host and the boards to be in progress simultaneously.

The PCIe interface software is split between the Windows device driver primarily responsible for managing communication and hardware-specific drivers implemented as embedded functions in a microblaze soft-processor in the ADC/-DAC controller FPGA. The Windows driver was written in such way that it is possible for the host application to overlap transfer operations with host-side processing hence improving the system performance. The PCI Express driver integrated DMA engine for transfers between board and host memory under control of the soft-processor controller. The board and the host share 1 GB of external DDR3 memory, as well as 128 KB of on-chip Block RAM. The DDR memory is reserved for data storage, while the Block RAM is used for coordination between the board controller and its host. The PCI Express firmware was developed with Xilinx Vivado 2015.2.

3.3 Time-of-Flight 3D Imaging on Zynq

Time-of-Flight (ToF) is a technology providing distance information by measuring the travel time of emitted infrared light with the help of photonic mixing devices (PMD), cf. [3]. However, the provided raw data of a ToF sensor requires processing in order to obtain depth data, which is typically done in software. In the following, a novel Xilinx Zynq solution is presented, which closes the gap in the field of flexible but fast hardware-accelerated ToF processing.

The Zynq SoC, depicted in Fig. 3, is designed as a supporting co-processor, thus it is controlled and operated from an external processing system. Specific commands can be sent to the SoC through peripheral interfaces (I2C, Ethernet, etc.). The use-case application, which runs on one of the ARM cores, implements the actual use-case (e.g., gesture recognition). Its task is also to configure all the other HW/SW components of the SoC and to configure and control the ToF camera. Finally, it evaluates the calculated depth data, which is saved in the *Depth Data RAM*, according to the use-case implementation and transmits results, events, commands, etc. to an external CPU. Every ToF camera based on Infineon's REAL3TM sensor provides calibration data which is used by the processing algorithms to compensate lens distortions, ToF systematic errors, etc. The calibration data is typically saved within the camera's flash memory and is loaded by the SoC into its dedicated calibration RAM area. When the ToF camera is started, ToF raw data is received through the FPGA's parallel interface. The *Video In* unit generates a video stream and forwards it to the *Video DMA* which pushes the data into the *Raw ToF Data RAM* and notifies the

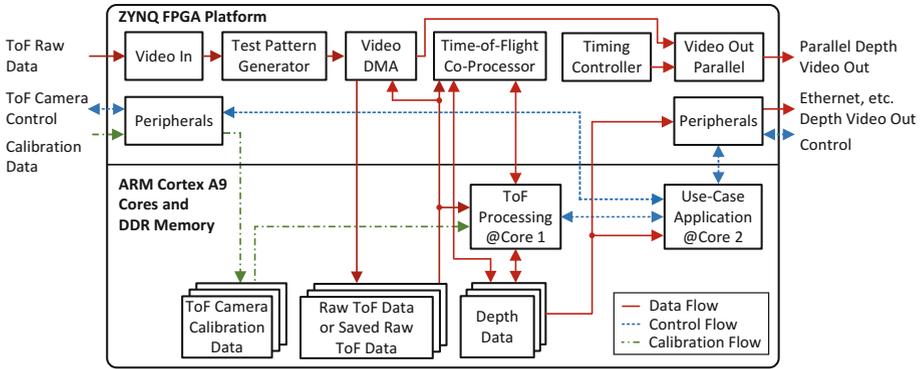


Fig. 3. PMD-based ToF processing on Zynq platform.

ToF Processing software. This software runs on a separate ARM core and utilizes the *ToF Co-Processor* for hardware-acceleration. The co-processor’s control logic block interprets incoming instructions, configures the data buffers and the processing engine, and starts the hardware-accelerated operations. More than a dozen of hardware-integrated operations (such as arcus tangent or square root of two images) are supported, which are typically used by ToF algorithms. After an instruction was executed, the ToF processing software is notified through interrupts. Thanks to its efficient and fine-grained implementation, high-speed and yet flexible ToF solutions can be realized. Finally, the resulting depth data is saved in the *Depth Data RAM* and is employed by the use-case application (Table 3).

A 4-phases ToF measurement represents a typical gesture recognition use-case scenario. In this work, the implementation of the reference processing includes the following operations: depth, amplitude, 3D point cloud calculations, and the compensation of common ToF systematic errors. Compared to the high-precision reference implementation in software (using floating point operations), an average depth error (caused by the inexact hardware calculations) of only 0.08 mm is introduced. Overall, this framework sets a new benchmark for hardware-accelerated ToF processing.

Table 3. Timing results for gesture recognition use-case

Time-of-Flight algorithm	t [ms]	FPS	#HW Instr.
Reference in software	248.15	4.0	-
Reference in Zynq HW/SW	28.38	35.2	20
Optimized HW accel. processing	10.11	98.9	14

3.4 A Predictable, Formally Verifiable NoC

EMC2 type systems demand both safety and performance, where predictability is essential in providing both simultaneously. For example, there must be guarantees that a high-bandwidth video process activity cannot adversely affect the response time of a safety-critical control circuit. The majority of NoC implementations do not provide suitable latency and behavioural guarantees, requiring conservative utilisation or over-provisioning of resources.

In response to this, the MCENoC [7] provides a non-blocking topology built from simple switching elements based on Clos [2] and Beneš [1] type networks. Such a network arrangement allows N concurrent connections between N nodes, with the number of switches, S , scaling logarithmically, where $S = 2 \log_2(N) - 1$.

Switches are arranged into stages as depicted in Fig. 4, such that all possible routes traverse the same number of switches. This tightly bounds the latency of all network communication to a fixed value for a given size of network.

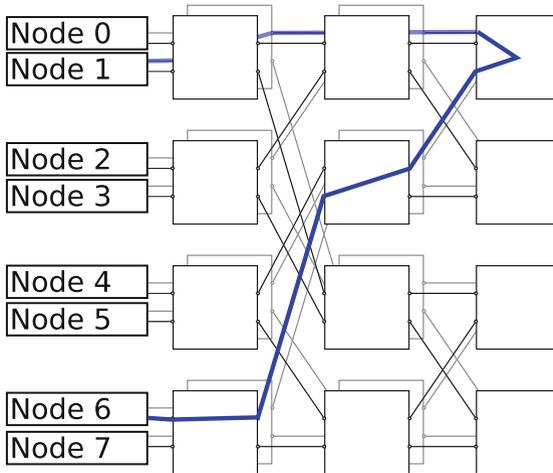


Fig. 4. Example of an eight node network using five switching stages totalling 20 switches. One possible route between node 0 and node 6 is depicted.

Formal Verification (FV) techniques are used to ensure that the design specification is robust and unambiguous, and that the implementation of switches, the total network structure, and edge interfaces, are correct with respect to the specification. System Verilog Assertion [11] language (SVA) is used in combination with the Jasper Gold FV tool to prove safety-critical properties, such as guaranteeing the routing behaviour and error responses in all possible input conditions [7]. The simple nature of the switching elements permits this, and using FV in place of test-based verification provides stronger behavioural guarantees, provided that the specification and SVA properties are adequately defined.

Implementations of the MCENoC have been targeted at the Kintex-7 FPGA, using both in-house custom hardware and the EMC2-DP. Implementation alongside 16–32 RISC-V processors is possible within a single FPGA at 100 MHz, using a configuration that provides a timing-predictable, cache-less array of processors. The MCENoC then provides a predictable network that is appropriate for use in combination with these predictable processors.

3.5 Enabling Software-Driven Energy Consumption Optimization

A novel target-agnostic mapping technique, introduced in [4], can be used to lift existing ISA resource models to higher levels of abstraction, such the Intermediate Code representation of the LLVM compiler infrastructure (LLVM IR) [10]. Mapping an ISA energy model to the compiler’s IR level has significant benefits over static LLVM IR energy models. Firstly, the mapping-based approach benefits from the accuracy that ISA models can provide, because the ISA is closer to the hardware than LLVM IR. Secondly, the dynamic nature of the mapping technique can account for specific architecture and compiler behavior, such as code transformations.

The mapping technique was used together with a new target-agnostic profiling technique to retrieve energy estimations at the LLVM IR level. This profiling technique was designed to ensure that the instrumentation code required for profiling does not lead to energy overheads. The experimental evaluation on a comprehensive set of single- and multi-threaded deeply embedded programs, demonstrated that the achieved estimations had an average absolute error of only 3% compared to hardware measurements. Furthermore, the technique was able to attribute energy consumption to the various software components at the LLVM IR level, such as basic blocks and functions, and then correlate this information with the source code.

The profiling-based estimation proved to be significantly more efficient than existing instruction set simulators. The high accuracy and performance of the profiling can enable feedback-directed optimization for energy consumption. Further research is needed to improve energy-transparency techniques for energy-aware software development.

3.6 A Heterogeneous Time-Triggered Architecture on a Hybrid System-on-a-Chip Platform

As described in Sect. 2 ensuring performance for future safety critical applications and implementing mixed-critical applications on COTS hardware presents a major challenges. The proposed architecture provides an alternative approach that ensures these basic properties and adds additional functionality extremely beneficial for industrial applications.

The presented architecture [6] utilizes underlying hybrid SoC technology and time-triggered communication principles. Former allows designers to engage in design of custom hardware in an FPGA fabric, while being able to use advantages of the hard-coded processor.

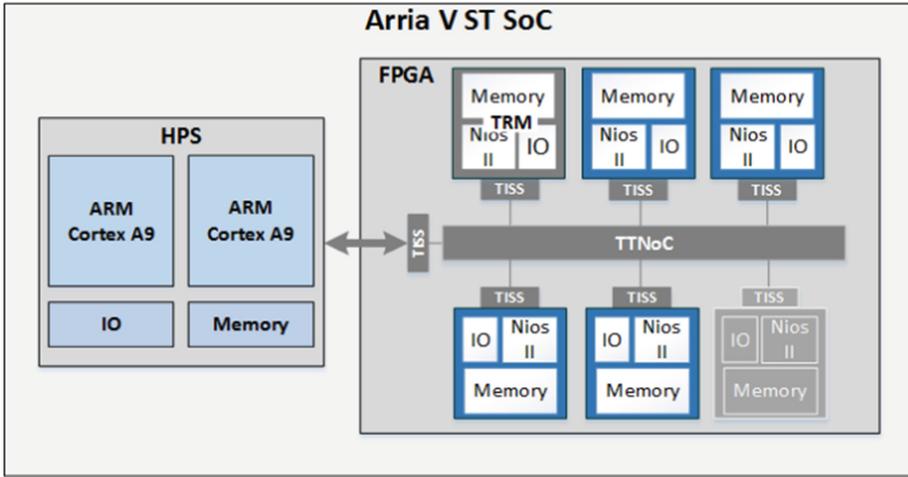


Fig. 5. Block diagram of the deterministic MPSoC architecture on hybrid SoC.

The architecture is built around a communication backbone called time-triggered network-on-a-chip (TTNoC) introduced in [18]. It is message based communication medium interfaced with an arbitrary number of computational components using a trusted interface subsystem (TISS) (see Fig. 5). The interface ensures temporal and spatial isolation of each component and provides them ability to operate in a synchronized fashion. TTNoC provides distribution of chip global time ensuring timeliness of all components.

The original architecture presented in [18] implements a homogeneous set of components implemented fully in an FPGA fabric. The architecture described in [6] and presented in this paper uses underlying hybrid SoC platform to implement a heterogeneous solution that combines hard-coded processor with FPGA based set of components, interconnected with the TTNoC.

Both architecture executions establish spatial and temporal isolation as vital properties. They enable integration of safety critical applications and non-critical applications on a single chip in a high-performance deterministic structure. The concept enables increase in performance while maintaining essential safety properties.

3.7 A Deterministic Coherent L1 Cache

Tightly-coupled multi-core systems with shared memory and central, fine-grained task scheduling can achieve the highest core utilization, provided that low-overhead inter-core communication and data sharing is guaranteed. As the number of cores grows, however, the memory bottleneck dominates and caches become indispensable for upping performance. Caches, in turn, require mechanisms for keeping shared data coherent. Conventional coherence techniques

deriving from the classic MSI protocol show largely indeterministic timing behavior due to complex cache interactions, rendering them inapplicable to hard real-time systems.

A time-predictable L1-cache coherence mechanism specifically tailored to real-time systems, has been developed in [16]. The goal is to enable fast access to shared data while maintaining a tight worst-case execution time (WCET) estimate, necessary for realistic timing analysis. The key idea is to hold shared data only as long as necessary, after which it is dumped to memory and reloaded before the next access. Only one core is granted access to a shared region at a single point in time. For this strategy to provide satisfactory performance, instructions are grouped into sequences, denoted by either *shared* or *private*. The cache does not attempt to maintain coherence as long as memory accesses are marked as private. As soon as a shared block is entered, the cache switches to the afore-mentioned on-demand coherence mode. Thus, the granularity of shared/private blocks has to be carefully chosen: Smaller blocks enable finer interleaving and balancing between cores at the price of higher overhead for flushing and reloading.

The performance of the caching strategy has been analyzed in [15, 17]. The algorithm has been integrated into the LEON3 caches of the SoCRocket SystemC platform [20], where we are currently testing our strategy in the context of mixed-critical applications.

3.8 Platform NoC Simulation with EMC2 SoCRocket

EMC2 SoCRocket is a virtual platform which enables early prototyping of Hardware/Software systems without the need of real hardware [19]. It eases the debugging and evaluation efforts, particularly focussing on full-system simulation. Resulting in a higher development speed for software and faster hardware exploration. The approach is tested and benchmarked with a real-world full-system example, demonstrating the overall benefits [13].

With SocRocket we assembled a platform to simulate a crosssection of the EMC2-DP hardware for special heterogeneous use cases. Said platform consists of the core components of the GRLib library with the LEON3 Processor extended by an ARM Cortex-A9 [20], MicroBlaze and for interconnection a NoC simulation executing tasks of different criticality levels [5]. The Zynq inside the EMC2-DP hardware uses an AMBA interconnect, this is replicated inside the SoCRocket simulation platform, enabling engineers to evaluate accelerator algorithms within a realistic design environment.

In the course of the project SoCRocket was extended by several features for mixed criticality development. One such feature is a standards-compliant powerful and flexible method of deriving, logging, and filtering detailed status information in different execution contexts. Another notable feature enhancement has been described in the previous section. By leveraging the coherency enhanced caches within the simulation framework we can better predict the real-time behaviour during simulation.

At the core of the simulation is a flexible scripting interface which may change all simulation parameters during run-time, thus not requiring recompilation of the to-be-simulated models [12]. The simulation with SoCRocket shows a speedup up to $160\times$ between RTL and the approximately timed TL-Model and $1400\times$ – $2000\times$ speedup between RTL and the loosely timed TL-Model by a simulation uncertainty of less than 10%.

4 Discussion and Future Work

The works presented in the paper provide insight in the hardware techniques for mixed-criticality integration. The heterogeneous TTNoc architecture presented in Sect. 3.6 provides can implement applications with different levels of safety and security without performance loss, while maintaining full spatial and temporal isolation. The future challenges include implementation of tools for configuration and deployment that would connect the whole development process from hardware to the application. Also, for MCENoc future work includes software and toolchain improvements, where communication needs such as bandwidth and periodicity must be known in advance. However, the fixed latency of this network simplifies resource allocation. Where dynamic network traffic is needed that cannot be known in advance, portions of network time could be dedicated TDM phases, controlled by a central unit, which has previously been demonstrated successfully on mesh networks [8]. To improve verification of energy consumption requirements, further research is needed to develop more energy-transparency techniques that can enable energy-aware software development. The EMC2 SoCRocket virtual platform can simulate real-time systems with mixed criticality tasks in a much faster way than RTL while still maintaining good enough accuracy. It can be further enhanced by speeding up the evaluation of energy requirements early in the design stage together with the software development could greatly enhance design efficiency. It provides a rapid prototyping platform for mixed-criticality applications. The asymmetric multiprocessing architecture on EMC2-DP demonstrates feasibility of the hybrid SoC platforms to carry high performance applications. The future work on this field considers full tool integration and further performance optimization. The ToF hardware/software framework enables flexible hardware-accelerated ToF processing for various types of use-case applications. It provides high-quality 3D point cloud data with nearly 100 frames per seconds while introducing an average calculation error of only 0.08 mm. The work on deterministic coherent cache memory provides ability to access data in deterministic fashion thus maintaining WCET bounds. This approach has a enormous advantage for mixed-criticality and real-time applications.

5 Conclusion

Technologies described in the paper provide hardware solution from architectural level up to the peripheral and application specific hardware. Moreover

paper presents extendable multiprocessing hardware platform based on Zynq hybrid SoC, an asymmetric multiprocessing in video processing architecture, Time-of-Flight sensor and image processing architecture, predictable and verifiable Network-on-Chip (NoC), heterogeneous time-triggered NoC architecture, virtual hardware platform, software-driven energy consumption optimization techniques, and time-predictable L1 cache memory. The application of hybrid SoC platforms opposed to COTS multi-core architecture provide multiple benefits and can be seen as a viable bridging solution in the gap between single- and multi-core architectures.

Acknowledgment. This research has received funding from the ARTEMIS Joint Undertaking (JU) in European project *EMC²* under grant agreement no. 621429.

References

1. Beneš, V.E.: On rearrangeable three-stage connecting networks. *Bell Syst. Tech. J.* **41**(5), 1481–1492 (1962). doi:[10.1002/j.1538-7305.1962.tb03990.x](https://doi.org/10.1002/j.1538-7305.1962.tb03990.x). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6769814>
2. Clos, C.: A study of non-blocking switching networks. *Bell Syst. Tech. J.* **32**, 406–424 (1952). doi:[10.1002/j.1538-7305.1953.tb01433.x](https://doi.org/10.1002/j.1538-7305.1953.tb01433.x)
3. Druml, N., Fleischmann, G., Heidenreich, C., Leitner, A., Martin, H., Herndl, T., Holweg, G.: Time-of-flight 3D imaging for mixed-critical systems. In: 13th International Conference on Industrial Informatics (INDIN), pp. 1432–1437 (2015)
4. Georgiou, K., Kerrison, S., Chamski, Z., Eder, K.: Energy transparency for deeply embedded programs. *ACM Trans. Archit. Code Optim. (TACO)* **14**(1), 8 (2017)
5. Horsinka, S.A., Meyer, R., Wagner, J., Buchty, R., Berekovic, M.: On RTL to TLM abstraction to benefit simulation performance and modeling productivity in noc design exploration. In: NoCArc 2014: Proceedings of the 2014 International Workshop on Network on Chip Architectures. ACM (2014). <http://doi.acm.org/10.1145/2685342.2685349>
6. Isakovic, H., Grosu, R.: A heterogeneous time-triggered architecture on a hybrid system-on-a-chip platform. In: 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), pp. 244–253 (2016). doi:[10.1109/ISIE.2016.7744897](https://doi.org/10.1109/ISIE.2016.7744897)
7. Kerrison, S., May, D., Eder, K.: A benes based NoC switching architecture for mixed criticality embedded systems. In: 2016 IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSOC), pp. 125–132 (2016). doi:[10.1109/MCSOC.2016.50](https://doi.org/10.1109/MCSOC.2016.50)
8. Kostrzewa, A., Saidi, S., Ernst, R.: Slack-based resource arbitration for real-time. In: Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1012–1017 (2016)
9. Lee, E., Seshia, S.: Introduction to Embedded Systems: A Cyber-Physical Systems Approach. Electrical Engineering and Computer Sciences, Lulu.com (2011). <https://books.google.at/books?id=MgXvLFE7HIgC>
10. LLVMorg: The LLVM Compiler Infrastructure (2014). <http://www.llvm.org/>
11. Mehta, A.B.: System verilog assertions. In: Mehta, A.B. (ed.) *SystemVerilog Assertions and Functional Coverage*. Springer, New York (2014). doi:[10.1007/978-1-4614-7324-4_2](https://doi.org/10.1007/978-1-4614-7324-4_2)

12. Meyer, R., Wagner, J., Buchty, R., Berekovic, M.: Universal scripting interface for systemc. In: DVCon Europe Conference Proceedings 2015 (2015). https://dvcon-europe.org/sites/dvcon-europe.org/files/archive/2015/proceedings/DVCon_Europe_2015_TA3_1_Paper.pdf
13. Meyer, R., Wagner, J., Farkas, B., Horsinka, S., Siegl, P., Buchty, R., Berekovic, M.: A scriptable standard-compliant reporting and logging framework for systemC. *ACM Trans. Embed. Comput. Syst.* **16**(1), 6 (2016). doi:[10.1145/2983623](https://doi.org/10.1145/2983623)
14. Moore, G.E., et al.: Cramming more components onto integrated circuits. *Proc. IEEE* **86**(1), 82–85 (1998)
15. Pyka, A., Rohde, M., Uhrig, S.: Performance evaluation of the time analysable on-demand coherent cache. In: 4th IEEE International Workshop on Multicore and Multithreaded Architectures and Algorithms, Melbourne, Australia (2013)
16. Pyka, A., Rohde, M., Uhrig, S.: A real-time capable coherent data cache for multi-cores. *Concur. Comput.: Pract. Exp.* **26**(6), 1342–1354 (2014)
17. Pyka, A., Tadros, L., Uhrig, S.: WCET analysis of parallel Benchmarks using on-demand coherent cache. In: 3rd Workshop on High-Performance and Real-time Embedded Systems (HiRES 2015) (2015)
18. Salloum, C., Elshuber, M., Höftberger, O., Isakovic, H., Wasicek, A.: The ACROSS MPSoC - a new generation of multi-core processors designed for safety-critical embedded systems. In: 2012 15th Euromicro Conference on Digital System Design (DSD), pp. 105–113 (2012). doi:[10.1109/DSD.2012.126](https://doi.org/10.1109/DSD.2012.126)
19. Schuster, T., Meyer, R., Buchty, R., Fossati, L., Berekovic, M.: SoCRocket - a virtual platform for the european space agency's SoC development. In: 2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), pp. 1–7 (2014). doi:[10.1109/ReCoSoC.2014.6860690](https://doi.org/10.1109/ReCoSoC.2014.6860690)
20. Braunschweig, T.U.: Transaction-Level Modeling Framework for Space Applications. <https://github.com/socrocket>
21. UTIA: UTIA public www server dedicated to the EMC2 project (2016). <http://sp.utia.cz/index.php?ids=projects/emc2>
22. Xilinx Inc.: Xilinx Inc. (2016). <http://www.xilinx.com>