SAFEPOWER project: Architecture for Safe and Power-Efficient Mixed-Criticality Systems

Alina Lenz*, Mikel Azkarate-Askasua Blázquez[§], Javier Coronel[‡], Alfons Crespo[‡],
Simon Davidmann[¶], Juan Carlos Diaz Garcia[†], Nera González Romero[†], Kim Grüttner**, Roman Obermaisser*,
Johnny Öberg[‖], Jon Perez[§], Ingo Sander[‖] and Ingemar Söderquist[††]

*University of Siegen Email: alina.lenz@uni-siegen.de, Germany [†]CAF Signalling, Spain [‡]FentISS, Spain, [§]IK4-Ikerlan, Spain,
[¶]Imperas, United Kingdom [‖]KTH Royal Institute of Technology, Sweden **OFFIS e.V., Germany [††]Saab, Sweden

***Abstract*—With the ever increasing industrial demand for bigger, faster and more efficient systems, a growing number of cores is integrated on a single chip. Additionally, their performance is further maximized by simultaneously executing as many processes as possible not regarding their criticality. Even safety critical domains like railway and avionics apply these paradigms under strict certification regulations.**

As the number of cores is continuously expanding, the importance of cost-effectiveness grows. One way to increase the cost-efficiency of such System on Chip (SoC) is to enhance the way the SoC handles its power resources. By increasing the power efficiency, the reliability of the SoC is raised, because the lifetime of the battery lengthens. Secondly, by having less energy consumed, the emitted heat is reduced in the SoC which translates into fewer cooling devices. Though energy efficiency has been thoroughly researched, there is no application of those power saving methods in safety critical domains yet.

The EU project SAFEPOWER[1] targets this research gap and aims to introduce certifiable methods to improve the power efficiency of mixed-criticality real-time systems (MCRTES).

This paper will introduce the requirements that a power efficient SoC has to meet and the challenges such a SoC has to overcome.

I. Introduction

Mixed-criticality real-time systems support functions with different criticality on one shared system. Their importance is based on the relentless demand for increased dependability, security, more intelligence, connectivity, better performance, energy efficiency and cost-size-volume reduction in industrial systems [1] [2]. The most important advantages provided by mixed criticality are:

- Power, cost, size, and weight reduction: The physical integration of components with different criticality on a single shared platform significantly reduces the overall number of ECUs, connectors and cables [2].
- Reliability increase: Connector failures are a source of failures in many MCRTES. The reduction of the overall number of ECUs, connectors and cables can improve the reliability of the overall system.
- Scalability and competitiveness: The possibility to continuously include new value-added functionalities without jeopardizing dependability and reducing the impact on the overall cost-size-power consumption must be ensured [2].

Several platforms for MCRTES have been introduced in previous works at chip level, in distributed systems and at operating system level [3]. However, some important challenges remain, one of them being the power consumption management and optimization in dependable MCRTES. The available energy has to be shared by all running applications, regardless if they are critical or not. The maximum power consumption of a SoC is effectively limited by its waste heat discharge capabilities and expected lifetime. The absence of power management and optimization can lead to a reduction in the availability of the system and expected lifetime.

Even in scenarios where the critical systems are not powered by batteries, power is a resource (together with time and space) that has to be considered for several reasons:

- Reliability: Low power consumption is an important factor to increase the operational reliability and availability in many industrial systems. If power consumption and heat are reduced, the impact on reliability is doubled. First, the negative influence on the aging of hardware elements is lowered, and second, it may avoid the use of cooling systems and mobile parts (e.g., ventilators) in the hardware design. Cooling systems contribute significantly to the probabilities of failure or add additional maintenance intervals.
- Availability: A low power consumption allows extending the operation of a system in special situations such as blackouts and energy disruptions.
- Ecology: Power consumption reduction is also a desired feature towards near-zero emission in systems with tens/hundreds of ECUs.

While mixed-criticality is the focus over several research projects (e.g., DREAMS, PROXIMA, CONTREX, Multi-PARTES, EMC2) [4] and publications, power and energy constraints in mixed-criticality systems have gained some attention [5] [6]. Nevertheless, they are still an unsolved research problem.

[1] This project and the research leading to these results has received funding from the European Communitys H2020 program [H2020-ICT-2015] under grant agreement 687902

The power management is required at different levels: at the chip-level hardware (e.g., processor cores, network-on-a-chip), in the system software (e.g., hypervisors) and at the level of distributed systems (e.g., nodes, networks). In particular, a low-power architecture is needed to enable the development of low-power MCRTES combined with already available energy saving approaches such as DVFS, clock/core gating or power mode switching.

This paper analyses the state-of-the-art and requirements of such an architecture. We focus on different integration levels: the chip-level hardware (e.g., processor cores, network-on-a-chip), the system software (e.g., hypervisors) and the level of distributed systems (e.g., nodes, networks). In addition, the paper introduces an architectural concept for power-efficient MCRTES.

The paper is structured into five segments. Section II focuses on the general requirements an architecture has to meet to fulfill low power features. Section III introduces the challenges in power-efficient operating systems services for MCRTES, while section IV discusses the power-efficient on-chip communication. The final section V describes the SAFEPOWER project and its used methodology.

II. REQUIREMENTS OF AN ARCHITECTURE FOR A LOW POWER MIXED-CRITICALITY SYSTEM

This section discusses fundamental requirements of a low-power architecture for mixed-criticality systems.

A. Software Architecture fundamentals

Mixed criticality systems are integrated by several applications with different levels of criticality. Spatial and temporal isolation are basic properties to achieve the system partitioning. The concept of partitioned software architectures was developed to address security and safety issues [7]. In [8] a separation kernel to enforce a stronger isolation between processes or groups of processes was proposed. Each group of isolated processes was considered as a partition. Partitioning kernels can be obtained as extension of operating systems (OS) enforcing the process isolation or specific virtualization layer providing processor virtualization to partitions [9]. This virtualization layer is called hypervisor. Virtualization refers to the creation of a virtual machine or partition that acts like a real computer with OS, but executing the software applications separately from the underlying hardware resources. Among other objectives, virtualization is intended to support system partitioning and to protect the execution time and memory space of each application. Virtualization can provide a full or partial virtualization and can be implemented on top of the hardware (bare metal hypervisor) or the OS [7].

B. Energy and Power Effiency

The state-of-the-art encompasses a broad spectrum of low power techniques. Since the supply voltage is an important factor for both dynamic and static power, many techniques manipulate the supply voltage. One way is to use different supply voltages (Multi Voltage) for different components in combination with level shifters. Dynamic Voltage and Frequency Scaling (DVFS) is another technique where a power manager controls different power modes, consisting of a pre-defined set of supply voltage and clock frequency tuples. In some new technologies, this can be combined with Adaptive Body Biasing (ABB), to control the leakage power more effectively. Instead of scaling down the supply voltage, it can be switched off completely (power gating) if the switched-off parts are not used over a longer period of time. Points of consideration are the high energy costs and delays for shut down and start up, the need for isolation cells and state retention registers. Alternatively, clock gating disables the clock for complete system blocks or selectively suspends clocking. It requires less effort than power gating, but only controls the dynamic power consumption, while power gating also attacks the static leakage power, that may have a considerable impact on the overall power consumption.

Further low-power techniques include power modes (e.g., idle, sleep), effective cache usage, selectively clock gated caches, small architectures with less static power, DPM (Dynamic Power Management), application-driven and operating-system driven solutions. The scheduling of software tasks and the design of the software tasks themselves should also consider these techniques. For example, components should as long as possible remain in the switched off state or in low voltage modes.

In the context of mixed-critical systems, these techniques cannot be used at their full potential because they have a significant influence on the timing. For example, voltage/frequency scaling leads to different execution times and switching off components leads to different response times since they have to be switched on first. All these leads to a more complex timing behavior and potentially an unpredictable impact of less critical application components on critical ones. Furthermore, the benefits of power savings cannot be fully explored because, in general, they are not fully predictable or observable. Hence, todays safety critical systems cannot take advantage of these savings.

When considering low-power features in safety critical systems, several concerns may arise e.g., ensure that low-power techniques do not jeopardize the safe operation of the system. When it comes to mixed-criticality, independence is a crucial factor to allow mixed-criticality systems to be certified separately and thus considerably reduce certification costs, improve scalability and flexibility of the system. Much research effort has been devoted to achieve such a spatial and temporal independence, however, power consumption is also a crucial factor. As reported in [10], increased power consumption of one application may reduce the available energy for other applications or the reliability and lifetime of the complete chip. Therefore, the power consumption of one application can induce a negative impact on other applications of different safety criticality, violating the required independence on which mixed-criticality systems are based. In this vein, in [5] authors claim that energy is as important as time in mixed-criticality systems and they demonstrate how an incorrect handling of energy can violate mixed-criticality guarantees. With the purpose of overtaking this issue, in [10] a monitoring and control mechanism to isolate the power consumption of mixed-criticality applications on a many-core platform has been proposed. In [6] a fully predictable and composable many-core platform successfully employs DVFS to save power during slack periods under a global TDMA scheduling.

C. *Predictability and Real-Time Response Requirements*

Achievement of control stability in real-time applications depends on the completion of activities (like reading of sensor values, performing computations, communication activities, actuator control) in bounded time. Hard real-time systems ensure guaranteed response even in the case of peak load and fault scenarios. Guaranteed response involves assurance of temporal correctness of the design without reference to probabilistic arguments. Guaranteed response requires extensive analysis during the design phase such as an off-line timing and resource analysis [11]. An off-line timing and resource analysis assesses the worst-case behavior of the system in terms of communication delays, computational delays, jitter, end-to-end delays, and temporal interference between different activities.

In hard real-time systems, missed deadlines represent system failures with the potential of consequences as serious as in the case of providing incorrect results. For example, in drive-by-wire applications, the dynamics for steered wheels in closed control loops enforce computer delays of less than 2ms [12]. Taking the vehicle dynamics into account, a transient outage-time of the steering system must not exceed 50 ms [12].

While control algorithms can be designed to compensate a known delay, delay jitter (i.e. the difference between the maximum and minimum value of delay) brings an additional uncertainty into a control loop that has an adverse effect on the quality of control [13]. Delay jitter represents an uncertainty about the instant a real-time entity was observed and can be expressed as an additional error in the value domain. In case of low jitter or a global time-base with a good precision, state estimation techniques allow to compensate a known delay between the time of observation and the time of use of a real-time image.

D. *Fault isolation*

Especially in mixed criticality systems fault isolation is important, since low and high critical processes share the same devices and memory space. Having a fault emerging in a low priority process may not in any way impact the performance of a high criticality process. The faults must be contained to prevent correlated failures of safety-critical and non safety-critical processes. Therefore a scheme for strict temporal and spatial partitioning is essential for the generic architecture. Additionally faults can be masked by fault tolerance mechanisms before they can cause any damage to the running task or application. Using redundancy can improve the systems reliability in case the replicas do not fail in a correlated manner.

E. *Safety certification*

Safety-critical applications have made very limited use of energy and power management features. Non-safety related embedded applications (e.g., consumer electronics) can shut- or slow-down hardware features only affecting the user experience, but safety-critical applications must also carefully consider the impact of those actions on the overall system safety. In the latter, those low-power features must comply with safety standard requirements (e.g., IEC 61508) in both: (1) the product life-cycle or functional safety management (to avoid systematic design faults) and (2) techniques and measures to control failures during operation (to control physical random faults).

For instance, according to the product life-cycle, dynamic reconfiguration is not recommended for SIL 2-4 integrity levels and this suggests that the adaptation to changing scenarios (e.g., a low power mode) must be addressed with precompiled and verified schemes, like in [14] at operating system level or at network level. Gating actions, such as for peripheral clock or core, must perform safe shutdown and startup actions. In [15], for instance, safe startup and shutdown scenarios are considered for an IEC 61508 compliant hypervisor partitions, but not primarily for power management proposes.

Additionally, a mixed-criticality approach can also benefit from modular certification. This feature is considered in several domain safety standards with different name: in IEC 61508 each module is named compliant item, in ISO 26262 it is called safety element out of context (SEooC) and in EN51019 generic product. The modular approach reduces the impact of changes to a subset of the safety case, enabling reusability of its parts [16]. Low power services must comply with the safety argumentation behind such an approach.

III. POWER-EFFICIENT OPERATING SYSTEM SERVICES FOR MIXED-CRITICALITY SYSTEMS

A. *State of the Art*

In the presence of power restriction and real-time constraints, real time operating systems (RTOS) should implement power management strategies encompassing the entire system. Several research issues on power restrictions at operating system level can be found in the literature. In recent years, several techniques have been proposed to address this issue [17]. The proposed techniques can impact at different levels such as I/O, memory, processor management and network. Most active research topics are focused on the processor, which contributed to the development of the two most popular power aware scheduling techniques: dynamic voltage and frequency scaling (DVFS) [18] which exploits the convex and normally quadratic relationship between CPU energy consumption and processor speed, and dynamic power management (DPM) [19] which is based on the use of low power energy states (like sleep or stand-by) every time the processor is idle.

DVFS approach is to dynamically adjust the frequency of the processor in such a way that the tasks finish execution before their deadline. The idea is to find the ideal frequency for lower power consumption while also ensuring that the tasks are finishing their execution before deadline. DPM is a mechanism that dynamically reconfigures a system to provide the requested services and tasks at the same performance level but with a minimum number of active components or a minimum load. DPM considers the transition time between different power consumption modes. Switching from the active mode to the sleep mode and then back to the active mode has a penalty in time and energy overhead, therefore, it requires to check the impact from a point of view of scheduling and energy consumption.

B. Challenges and Research Gap

Traditionally in real-time and embedded systems the timeliness has been the dominant criterion and energy has played only a subordinate role, i.e., the main goal has only been to guarantee timely completion of tasks. However, in mixed-criticality systems some tasks are more important than others and it is allowed to guarantee their completion even at the expense of others. In these systems the role of the energy budget could surpass or have the same relevance as the temporal dimension. In fact, in some scenarios the only way to avoid violations of the mixed-criticality guarantees is to consider energy and time with the same importance [5].

In this sense, the partitioned systems based on hypervisors present additional opportunities and limitations to the use of power-aware scheduling techniques. In partitioned systems, the hypervisor is in charge of the memory and processor management. IO management is delegated to the OS allocated in the partition. From this perspective, the hypervisor is in charge of the efficient memory management and scheduling of the partitions. The scheduling of internal tasks to a partition is the responsibility of the partition OS. Additionally, all the resources at hypervisor level are statically allocated. It means that the decision about the operating frequency of the processor for executing a partition should be taken offline.

The limitations imposed by the coexistence of the hypervisor and OS to manage the execution of the tasks in a partition are impacted by the scheduling at hypervisor level (static and based on cyclic scheduling) and OS (based on static or dynamic priorities). However, this approach presents some new opportunities for new techniques based on the collaboration of both layers (hypervisor and partition OS). Consumption models at hypervisor, partition and tasks level could be considered to optimize the global energy consumption. From this point of view, the hypervisor can fix the operating frequency or frequencies available to be used for partition execution, the hypervisor can take decisions to save energy when the partition has finished its activity or no partition is ready to be executed. The OS can request for changed frequencies during the task execution. New hypervisor services for power management to assist the partition execution will be required.

IV. POWER-EFFICIENT ON-CHIP NETWORKS FOR MIXED-CRITICALITY SYSTEMS

A. State of the Art

There has been a multitude of studies on how to make power-efficient on and off-chip networks, but they mainly focus on implementation details and the power behavior of the NoC itself, which is of limited interest in this context. Then there are those that aim to generate the NoC itself. These pure NoC generators typically produce VHDL and/or Verilog code for a certain type of NoC, with a bunch of parameters to modify its settings, together with a Network Interface so it can be operated from a testbench. However, in general they do not provide support for integration into a working multiprocessor system-on-chip (MPSoC) system. The third category of providers is those that generate an entire MPSoC system, but are less flexible when it comes to generate different types of NoCs. Instead, they focus on reducing the efforts of integrating the NoC into the system, and how to generate a working HW/SW MPSoC system. This is of high interest to the SAFEPOWER project, not only because it raises the technology readiness level (TRL) considerably, but also lets us explore the predictability and thereby the safety of the final system. The fourth category of providers is the commercial ones, with ready-made NoC chip solutions, ready to be integrated as an add-on to some FPGA boards. However, since these systems come with a fixed notion of NoC and memory hierarchy, they are of less use since they were not designed with predictability and programmability using models of computations (MoCs) in mind. In the coming subsections, we will go through SoA research that is relevant to this study.

1) Low-Power Aspects: Reducing power when it comes to implementing NoC structures is pretty straightforward. There are only two parameters to play around with: area and frequency. Since power is proportional to the switched capacitance, which in turn is proportional to the area, the rationale behind minimizing area is: the smaller the design, the lower the power consumption will be. This opts for implementing the NoC structures in a bit-serial manner. However, for constant throughput, the frequency must then be increased with the same factor as area was reduced, gaining very little in the end. You only gain if the channel is silent for long periods of time. The other option is to reduce the switching frequency, either the operating/clock frequency of the switch/routers or the frequency of the data traffic frequency in the network. Since it is hard to reduce clock frequency in the switch network, asynchronous communication has been suggested between the switch/router nodes [20]. The rationale behind this is that the network should only switch and consume power when it has something to switch. However, asynchronous NoCs are difficult to make predictable, and how to start sending once the reset signal is globally released is a really interesting problem. Thus, most designs continue to be synchronous implementations. For a comprehensive overview of methods to reduce power in NoCs, we refer the reader to [21].

2) Open Source NoC Generators: Many NoC structures have been suggested over the years since the first paper with the word NoC in the title was published in the year 2000 [22]. A few of those has even been released as open source, but then mainly for use as a help for doing research. The more interesting of these are the ones that come with a generator, i.e., with a method or program that allows the user to configure the NoC according to his or her needs. For instance, the CONNECT tool from Carnegie-Mellon University [23] [24] can generate Verilog code for various NoC implementations, but the code is released under copyright and cannot be reused by anyone else to create a commercial product. The Atlas framework, developed by the GAPH group at PUCRS in Brazil [25] [26], can produce different NoC topologies and generate synthesizeable VHDL files. Another example is Netmaker from the University of Cambridge. It is a library of fully-synthesizable parameterized Network-on-Chip (NoC) implementations, released under GPL license, so it cannot be used for commercial purposes either. Others generate only parts of the NoC, for instance [28], from Stanford, which presents a parameterized RTL implementation of a state-of-the-art VC router, or HNOCs, which is targeted for simulation of Heterogeneous NoCs.

3) NoC MPSoC System Generators: The most interesting NoC Generators are those that come with a complete design flow, which let designers to compose entire MPSoC systems, including SW stack and Device Drivers. These generators have in common that they use an XML description to specify the MPSoC system. The XML is then used as input to a generator program that then produces the actual implementation. They are typically limited to a few NoC type and topologies, but focus instead on ease of use and producing a working design that is correct-by-construction. The two most prominent ones are the NoC System Generator from KTH in Sweden [30] [31] [29] [32], the CompSoC platform from TU Eindhoven and TU Delft in the Netherlands [28]. The KTH NoC System Generator is a tool suite using the Nostrum NoC from KTH. It has a GUI as frontend for entering the MPSoC system, and uses MoCs inspired by the ForSyDe methodology [33]. The tool suite is also FPGA vendor agnostic. The tool generates an image of its internal representation of the system in the target FPGA tool vendors own frontend language, i.e., it generates sopc, or qsys files for Altera, and mhs and vivado tcl scripts for Xilinx implementations. The CompSoC platform is centered around the Aethereal NoC, and targets Xilinx technology. It also has hooks for importing designs generated by the ForSyDe methodology.

B. Challenges and Research Gap

In order to utilize the techniques presented in this paper, it is of key importance that the properties and services of the architecture can be used at higher levels of abstractions in the design flow. Designing a power-efficient mixed-criticality system, where several applications share the same platform is extremely challenging. Thus the design process should start at a high level of abstraction and needs to be supported by tools for design space exploration and synthesis.

Given a set of application models with individual design constraints, a set of global constraints, and a platform model, the objective of the design space exploration (DSE) activity is to find an efficient implementation of all applications on the shared platform that satisfies all individual and global design constraints. In the context of mixed-criticality systems it is of utmost importance that the DSE process can give guarantees that all constraints in terms of timing and power will be met by the proposed implementation. The techniques discussed in this paper and the research within the SAFEPOWER project are key prerequisites for a DSE-tool because of the QoS-guarantees that can be provided by the platform. The DSE-tool presented in [36] formulates the DSE-problem as constraint satisfaction problem and captures applications as synchronous data flow (SDF) graphs [35] and uses a predictable MPSoC platform with TDM-bus. A solution does not only give a mapping of SDF-actors to processing elements, but also generates the schedule for the set of actors on each processing element and schedules the communication on the TDM-bus. However, the tool focuses only on timing guarantees and does not take power into account. The analytical DSE-tool [36] has been combined with a simulation-based DSE-tool in [37] into a joint analytical and simulation-based DSE tool to analyze typical scenarios in addition to the worst case through simulation. The simulation tool can also be equipped with power-models to give the power consumption for a typical scenario. However, there is still a lack of good power models for higher levels of abstraction, while it is already possible to give good timing models for predictable platforms. This makes it so far difficult for DSE-tools to give absolute power guarantees.

The NoC system generator [29] currently uses a heart-beat model [30] to support applications to support applications modeled with a synchronous models of computation, and can generate implementations from Simulink models [31]. Inside the SAFEPOWER project, the NoC system generator will be extended to support techniques for low power predictable NoCs. Furthermore, an integration of a DSE-tool into the NoC-system generator would facilitate system design, because then the designer can focus on the design of the applications, while the DSE-tool calculates an efficient implementation and the NoC system generator generates the full FPGA implementation.

V. RESEARCH PROJECT SAFEPOWER

The SAFEPOWER project aims to enable the development of cross-domain mixed-criticality systems with low power and safety requirements by a reference architecture orchestrating different local power-management techniques based on safe and securitized built-in low-power services.

As shown in Figure 1, SAFEPOWER builds a comprehensive suite of analysis, simulation and verification tools for low-power mixed-criticality systems, including hardware and software reference platforms assisting the implementation, observation and test of such applications. SAFEPOWER will

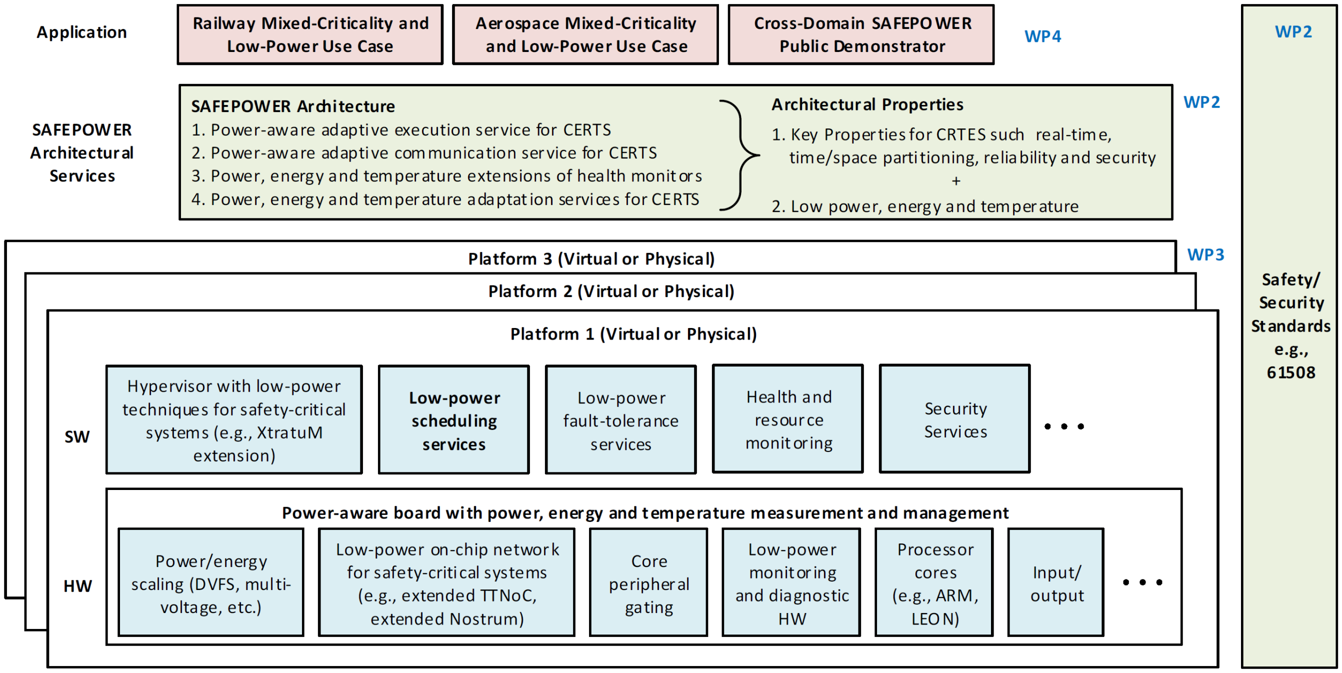
Application
Railway Mixed-Criticality and Low-Power Use Case
Aerospace Mixed-Criticality and Low-Power Use Case
Cross-Domain SAFEPOWER Public Demonstrator
WP4
WP2
SAFEPOWER Architectural Services
SAFEPOWER Architecture
1. Power-aware adaptive execution service for CERTS
2. Power-aware adaptive communication service for CERTS
3. Power, energy and temperature extensions of health monitors
4. Power, energy and temperature adaptation services for CERTS
Architectural Properties
1. Key Properties for CRTES such real-time, time/space partitioning, reliability and security
+
2. Low power, energy and temperature
WP2
Platform 3 (Virtual or Physical)
WP3
Platform 2 (Virtual or Physical)
Platform 1 (Virtual or Physical)
SW
Hypervisor with low-power techniques for safety-critical systems (e.g., XtratuM extension)
Low-power scheduling services
Low-power fault-tolerance services
Health and resource monitoring
Security Services
• • •
HW
Power-aware board with power, energy and temperature measurement and management
Power/energy scaling (DVFS, multi-voltage, etc.)
Low-power on-chip network for safety-critical systems (e.g., extended TTNoC, extended Nostrum)
Core peripheral gating
Low-power monitoring and diagnostic HW
Processor cores (e.g., ARM, LEON)
Input/ output
• • •
Safety/ Security Standards e.g., 61508

Fig. 1. SAFEPOWER Project Architecture

demonstrate and support effective low-power execution and adequate certification cognizance of mixed-criticality power-aware systems through EU industry representative use-cases and illustrative safety concepts assessed by an external certification authority. Additionally, SAFEPOWER will also maximize the impact of the project promoting a cross-domain public demonstrator, available for other different industrial domains.

The goal of the project to decrease the power consumption of MCRTES up to 50%, based on mixed criticality systems while maintaining the necessary operation requirements. SAFEPOWER will draw upon pre-existing results from the FP7-DREAMS project architecture, the FP7-MULTIPARTES and FP7-PROXIMA safety concept approach and the estimation and analysis of SoC power and temperature of the FP7-CONTREX project.

VI. Conclusion

In this paper, we presented the european project SAFEPOWER addressing the research problem of power management in mixed-criticality systems. The project's aim is to achieve space, weight and power reduction as well as enhancing reliability and availability.

We examined the requirements of an architecture needed for a low-power mixed-criticality system. The integrated power management must not affect the real-time and predictability features, the safety arguments and the fault tolerance. Addressing these challenges is one of the project's main goals. Furthermore, we discussed how the hardware and software of an MPSoC can be enhanced with power saving schemes. Based on a start-of-the-art analysis we depicted the challenges of including low power techniques in the system software and in the on-chip communication network.

Finally, we discussed the project itself, including its goals and the workplan. We also showed the projects architecture providing an overview of the components and low power services planned at hardware and software level. Using the proposed power management, the MPSoCs energy consumption shall be decreased by up to 50%.

References

[1] A. Burns and R. Davids. *Mixed Criticality Systems A Review*, 2016.

[2] C. El Salloum, M. Elshuber, O. Hftberger, H. Isakovic and A. Wasicek. *The ACROSS MPSoC A New Generation of Multi-Core Processors designed for Safety-Critical Embedded Systems*, 2012.

[3] Roman Obermaisser, Zaher Owda, Mohammed Abuteir, Hamidreza Ahmadian and Donatus Weber. *End-to-end real-time communication in mixed-criticality systems based on networked multicore chips*, Digital System Design (DSD), 2014 17th Euromicro Conference on, Verona, 2014, pp. 293-302.

[4] S. Trujillo, R. Obermaisser, K. Grüttner, F. J. Cazorla, and J. Perez. *European Project Cluster on Mixed-Criticality Systems*. In 3PMCES Workshop (Performance, Power and Predictability of Many-Core Embedded Systems) at DATE'14.

[5] Marcus Völp, Marcus Hähnel, and Adam Lackorzynski. *Has energy surpassed timeliness? Scheduling energy-constrained mixed-criticality systems*, In: 20th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2014.

[6] Andrew Nelson, Anca Mariana Molnos, and Kees Goossens. *Composable power management with energy and power budgets per application*, In: 2011 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, SAMOS XI, Samos, Greece, 2011.

[7] G. Heiser. *The role of virtualization in embedded systems*. First workshop on Isolation and Integration in Embedded Systems, 2008.

[8] John Rushby. *Security requirements specifications: How and what? In Symposium on Requirements Engineering for Information Security (SREIS)*, Indianapolis, 2001.

[9] R.P. Goldberg. *Survey of virtual machine research*. IEEE Computer Magazine, 7(6):3445, 1974.

[10] Boris Motruk, Jonas Diemer, Rainer Buchty and Mladen Berekovic. *Power Monitoring for Mixed-Criticality on a Many-Core Platform*. In Proceedings of the 26th international conference on Architecture of Computing Systems (ARCS), 2013.

[11] N. C. Audsley, I. J. Bate and A. Grigg, *The role of timing analysis in the certification of IMA systems*, Certification of Ground/Air Systems Seminar (Ref. No. 1998/255), IEE, London, 1998, pp. 6/1-6/6.

[12] J.S. Hoogheimestra and M.J.G Teunisse. *The use of simulation in the planning of the dutch railway services*, In Proceedings of the Winter Simulation Conference, 1998.

[13] H. Kopetz. *Real-Time SystemsDesign Principles for Distributed Embedded Applications*, KluwerAcademic, 1997.

[14] S. Baruah and G. Fohler. *Certification-cognizant time-triggered scheduling of mixed-criticality systems*, 3rd ed. (2011) Proceedings - Real-Time Systems Symposium, art. no. 6121421, pp. 3-12.

[15] A. Larrucea, J. Perez, I. Agirre, V. Brocal, and R. Obermaisser. *A Modular Safety Case for an IEC 61508 compliant Generic Hypervisor*, presented at the Digital System Design (DSD), Euromicro Conference on, Madeira, Portugal, 2015.

[16] A. Larrucea, J. Zwirchmayr, R. Obermaisser, J. Perez and I. Agirre. *A Modular Safety Case for an IEC 61508 compliant Generic Mixed-Criticality Network*, DREAMS EU Project Internal Report 2016.

[17] S. Mittal. *A survey of techniques for improving energy efficiency in embedded computing systems*. IJCAET 6(4): 440-459 (2014).

[18] J.O. Coronel and J.E. Sim. *High performance dynamic voltage/frequency scaling algorithm for real-time dynamic load management*. Journal System Software, p. 906-919. ISSN 0164-1212, 2012.

[19] C. S. Stangaciu, M. V. Micea, and V. I. Cretu. *Energy efficiency in real-time systems: A brief overview*. IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI): 275-280 (2013).

[20] E. Kasapaki, M. Schoeberl, R. B. Srensen, C. Mller, K. Goossens and J. Spars. *Argo: A Real-Time Network-on-Chip Architecture With an Efficient GALS Implementation*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems (Volume:24, Issue: 2), 479 492, 2015.

[21] Cristina Silvano, Marcello Lajolo and Gianluca Palermo. *Low Power Networks-on-Chip*, Springer, 1st edition, 2010.

[22] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Öberg, M. Millberg and D. Lindqvist. *Network on chip: An architecture for billion transistor era*, In Proceedings of the IEEE NorChip Conference, 2000.

[23] Michael K. Papamichael and James C. Hoe. *CONNECT: Re-Examining Conventional Wisdom for Designing NoCs in the Context of FPGAs*, In FPGA-2012, In Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays, Pages 37-46, 2012.

[24] Michael K. Papamichael. *CONNECT - CONfigurable NEtwork Creation Tool*, (http://www.cs.cmu.edu/ mpapamic/projects/connect.html).

[25] Aline Mello, Alexandre Amory, Ney Calazans and Fernando Moraes. *ATLAS - A NoC Generation and Evaluation Framework*, In University booth of DATE 2011.

[26] PUCRS, Brazil, *Atlas: Network-on-Chip Generation and Evaluation Framework*, https://corfu.pucrs.br/redmine/projects/atlas.

[27] University of Cambridge. *Netmaker* http://www-dyn.cl.cam.ac.uk/~rdm34/wiki/index.php?title=Main_Pag.

[28] Stanford. *Open Source Network-on-Chip Router RTL*, (http://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/Router)

[29] Johnny Öberg and Francesco Robino. *A NoC system generator for the Sea-of-Cores era* In Proceedings of the 8th FPGAWorld Conference (FPGAWorld '11). 2011.

[30] F. Robino and J. Öberg. *The HeartBeat model: A platform abstraction enabling fast prototyping of real-time applications on NoC-based MPSoC on FPGA*, Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2013 8th International Workshop on, Darmstadt, 2013, pp. 1-8.

[31] F. Robino and J. Öberg. *From Simulink to NoC-based MPSoC on FPGA*, Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, Dresden, 2014.

[32] *NoC System Generator*, KTH Royal Institute of Technology, Sweden, https://forsyde.ict.kth.se/noc_generator/.

[33] I. Sander and A. Jantsch. *System modeling and transformational design refinement in ForSyDe*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 23(1):17-32, January 2004.

[34] Y. Ben-Itzhak, E. Zahavi, I. Cidon and A. Kolodny. *HNOCS: Modular open-source simulator for Heterogeneous NoCs*, In Proceedings of the International Conference on Embedded Computer Systems (SAMOS), 2012.

[35] E. A. Lee and D. G. Messerschmitt. *Synchronous data flow*, in Proceedings of the IEEE, vol. 75, no. 9, pp. 1235-1245, Sept. 1987.

[36] K. Rosvall and I. Sander. *A constraint-based design space exploration framework for real-time applications on MPSoCs*, In Design Automation and Test in Europe (DATE '14), Dresden, Germany, Mar. 2014.

[37] F. Herrera, K. Rosvall, I. Sander, E. Paone, and G. Palermo. *An efficient joint analytical and simulation-based design space exploration flow for predictable multi-core systems*, In Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO), Amsterdam, The Netherlands, Jan. 2015.