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

Maher Fakih[**], Alina Lenz[*], Mikel Azkarate-Askasua[§], Javier Coronel[‡], Alfons Crespo[‡], Simon Davidmann[¶], Juan Carlos Diaz Garcia[†], Nera González Romero[†], Kim Grüttner[**], Sören Schreiner[**], Razi Seyyedi[**], Roman Obermaisser[*], Adele Maleki [*], Johnny Öberg[‖], Mohamed Tagelsir Mohammadat[‖], Jon Pérez-Cerrolaza[§], Ingo Sander[‖] and Ingemar Söderquist[††]

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

***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 without 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 article 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 is 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-

[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

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.

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 Dynamic Voltage and Frequency Scaling (DVFS), clock/core gating or power mode switching.

This article analyses the state-of-the-art and requirements of such 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 article introduces an architectural concept for power-efficient MCRTES.

This article is structured into five segments. Section II gives a coarse overview of the key elements of energy minimization techniques and mixed criticality. Section III focuses on the general requirements an architecture has to meet to fulfill low power features. Section IV introduces the challenges in power-efficient system software for MCRTES, discusses power-efficient hardware for MCRTES and the impact of low-power capabilities on dependability. In section V we describe the SAFEPOWER project and its used methodology. Finally, we summarize this work in the Conclusion section.

II. BACKGROUND

In this section we will take a look at the most relevant concepts from literature concerning energy minimization techniques, mixed-critical and dependable systems.

A. Energy minimization techniques

1) Sources of Power consumption: Sources of power consumption are either of static or dynamic nature. The static power contribution is completely application and data independent. It mainly depends on parameters that are fixed at MPSoC design time (chip area, used technology and process variation/corner) and dynamic properties that can be externally controlled (supply voltage, ambient temperature). Static (or leakage) power are majorly influenced by the leakage current which flows even when the device is inactive and is represented by the following [7]:

$$P_{leak} = I_{leak} V, \tag{1}$$

where I_{leak} is the leakage current and V is the operating voltage.

The dynamic power contribution (i.e. switching activity) is completely application and data dependent [8] and is affected by many factors. E.g., the software functionality, mapping of software tasks to processors, software scheduling, communication between tasks and the resulting communication and computation resource utilization. These factors impact the charging and discharging the switch capacitance of the load represented as follows [7]:

$$P_{dyn} = \alpha C V^2 F \tag{2}$$

where α is switching activity, F the operating frequency, C is the load capacitance and V is the operating voltage.

2) Overview of power management techniques: Especially for battery-operated embedded devices, energy saving is of vital impact. In addition, applying power management techniques reduces heat dissipation which in turn increases the long-term availability and reduces cooling equipment costs. Furthermore, with the help of the low-power techniques, resource usages can be also optimized (for e.g. by shutting down resource when not used) leading to an overall cost reduction. Additionally, it has been reported in [9] that information and communication technology contributes with 3% to the overall carbon footprint, so power management could contribute to making computing green. The state-of-the-art encompasses a broad spectrum of low power techniques which we will briefly review in the following (for a detailed survey c.f. [7]). The supply voltage is an important factor for both dynamic (decreases quadratically by voltage decrease) and static power (decreases linearly). Due to that, several techniques exist which manipulate the supply voltage and threshold voltage and state encoding dynamically or statically to reduce the voltage swing of switching transistors and the total number of switching transistors in the design. For e.g. *different supply voltages (Multi Voltage)* can be used for different components in combination with level shifters. DVFS [10] is another technique where a power manager controls different power modes, consisting of a pre-defined set of supply voltage and clock frequency tuples. The idea here is to find the ideal combination of the clock frequency and supply voltage for achieving lower power consumption while still fulfilling real-time requirements. In some new technologies, DVFS 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.

Using *Dynamic Power Management (DPM)* (also called Power Mode Management: PMM) [11] based on low power modes (e.g., idle, sleep, stand-by) supported by the underlying hardware can bring much in terms of energy saving. In every mode, different energy budgets and response times are needed. The intelligent management of transitions between different modes is done at the runtime based on the current system state. Other low-power techniques include *microarchitectural*

techniques for specific components of the MPSoC. One proposal here is to use small architectures (e.g. with scratch-pad memory instead of caches) with less static power. Other techniques utilize run-time parameters (e.g. workload) to apply dynamic reconfiguration of specific components for saving energy. Examples are (c.f. [7]): selectively clock gated caches, effective cache reconfiguration, memory compression or usage of appropriate cores (GPUs, FPGA, ASICs, DSPs, etc.). The scheduling of software tasks and the design of the software tasks themselves should also consider above techniques. For example, components should as long as possible remain in the switched off state or in low voltage modes.

B. Mixed Criticality

Mixed-criticality is the concept of allowing applications at different levels of criticality to seamlessly interact and coexist on the same computing platform. Figure 1 highlights the challenge of integrating two applications of different criticalities on the same multicore platform. Safety critical tasks (`F1, F2, F3`) have hard deadlines (e.g. $d_{2->3}$ in Fig. 1). For the safety-critical tasks a static schedule is predefined beforehand based on a Best-Case Execution Time (BCET)/ Worst-Case Execution Time (WCET) analysis and freedom from interferences is guaranteed. Typically, these kinds of tasks have no power and temperature constraints. Mission critical tasks, on the other hand, possess soft deadlines (e.g. $d_{4->5}$ in Fig. 1) based on Quality of Service (QoS) metrics. Such tasks have dynamic schedules with no guaranteed freedom from interference. Typically, mission-critical applications have hard power and temperature constraints. Systems applying Mixed-criticality concept must meet strict requirements up to the highest criticality levels (e.g., DAL A in RTCA DO-178B [12], ASILD in ISO26262, SIL4 in EN ISO/IEC 61508 [13]). Integration of such a system is only possible with the use of mechanism for temporal and spatial partitioning [14], which ensure fault containment and diminish unintended side effects between components.

One example of this methods are partitions, they encapsulate resources temporally (e.g., latency, jitter, duration of availability during a scheduled access) and spatially (e.g., prevent components from altering code or private data in other partitions). Therefore, partitioning is a necessity for modular certification, where each application subsystem is certified to the corresponding level of criticality [15]. Partitioned software architectures were developed as a concept to address security and safety issues [16]. In [17] a separation kernel was proposed which enforces a stronger isolation between processes or groups of processes in which every group of isolated processes was interpreted as a partition. Partitioning kernels can be realized as an extension of operating systems (OS) to enforce the process isolation or specific virtualization layers, therefore, providing processor virtualization to partitions [17]. This additional virtualization layer is called hypervisor (see for e.g. Fig. 2). It is referred to as virtualization because it is a virtual machine or partition that acts like a real computer with the OS, but executes the software applications

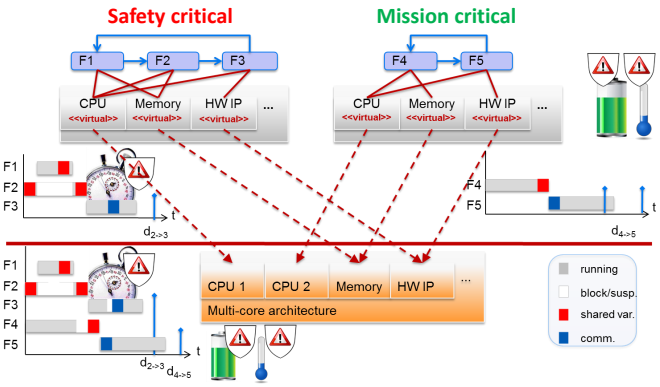

Fig. 1. Integration of mixed critical applications on single chip [18]

separately from the underlying hardware resources. Therefore, it manages the hardware access and creates timing and spatial partitioning between the applications running on the same core. This virtualization can be used to provide full or partial virtualization and it can be used directly on top of the hardware (bare metal hypervisor) or on top of an OS [16]. Multi-core chips are especially difficult to implement as mixed-criticality systems because various cores can easily cause interference for each other due to the shared use of resources like caches, bus arbitration policies and input/output. To avoid these interferences, the resource allocation must be predefined at design time in form of schedules. Also many safety-critical applications are hard real-time control applications, where the achievement of control stability and safety depends on the completion of activities (like reading of sensor values, performing computations, communication activities, actuator control) in bounded time. In such hard real-time systems, missed deadlines represent system failures with the potential of consequences as serious as in the case of providing incorrect results. Thus, hard real-time systems must ensure a guaranteed response even in the case of peak load and fault scenarios. A timing and resource analysis must assess the worst-case behaviour of the system in terms of communication delays, computational delays, jitter, end-to-end delays, and temporal interference between different activities. These requirements are ensured by a schedule which plans the execution of the application w.r.t. its deadlines.

III. 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. Energy and Power Efficiency

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 behaviour and potentially an unpredictable impact of less

TABLE I
OVERVIEW OF LOW-POWER TECHNIQUES

Low-Power Technique	*Targeted Power Source*	*Disadvantage(s)*
Voltage Scaling	Dynamic and Static	• Limited to manufacturing retention voltage, • Increasing application execution time.
DVFS	Dynamic and Static	• Limited to retention voltage/frequency • Finding lowest possible operating frequency and supply voltage is not easy • Increasing application execution time
Clock gating	Dynamic	• Ignores static leakage power
Power gating	Static	• High energy costs and delay for shut down and start up phases
Microarchitectural Optimizations	Static	• Ignores dynamic leakage power, • High costs

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 [19], "*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 [19] 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 Time-Division-Multiple-Access (TDMA) scheduling.

B. Predictability and Real-Time Response Requirements

Predictability is an important issue in real-time systems because real-time systems must guarantee response in specific time constraint. 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 [20]. An off-line timing and resource analysis assesses the worst-case behaviour 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 $2ms$ [21]. Taking the vehicle dynamics into account, a transient outage-time of the steering system must not exceed $50ms$ [21].

Delay jitter, which is defined as the difference between the maximum and minimum observed delays, would introduce additional uncertainty and could degrade the quality of control [22]. It can be regarded as the uncertainty about the observed timing of a real-time entity and can be stated as an additional error parameter in the value domain. State estimation techniques allow us to compensate a known delay between the time of observation and the time of use of a real-time image in case of low jitter or a global time-base with a good precision.

C. Fault isolation

A fundamental issue in mixed criticality systems is fault isolation. In mixed criticality systems, where low and high criticality processes share the same device and memory space, faults need to be detected and isolated in order to avoid their propagation. Severity of a fault can be categorized according to the priority of the embedding process. Emergence of a fault in a low priority process should not impact the performance of a high criticality process. Faults need to be contained in order 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, other mechanisms based on fault tolerance mechanisms, such as masking faults, may be utilized to prevent any damage to the running tasks. Redundancy can be used to improve the reliability of systems in cases where the failures of replicas are not correlated.

D. 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),
2) techniques and measures to control failures during operation (to control physical random faults).

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 re-usability of its parts [23]. Low power services must comply with the safety argumentation behind such an approach.

Certifiability measures are identified by authorities depending on the deployment domain. In the following, we give a clear identification of the certification objectives for the avionics domain enlightening the challenges faced within such a certification process. All newly developed aircraft systems must obtain a type certificate from the responsible aviation regulatory authority (e.g. EASA or FAA) before they enter into operation. This certificate, which always is on aircraft level, testifies that the type of aircraft including all used technology meets the safety requirements. One important overall certification requirement defined in EASA reference (see [24]) for equipment, systems and installations, is the development under the guidance of current aircraft and system development assurance process (ARP) [25]. ARP 4754 identifies relationship to often-referred guidelines RTCA/DO 297 for integrated Modular Avionics (IMA) development [26], RTCA/DO 178C for Software development [27], and RTCA/DO 254 for Airborne Electronic Hardware (AEH) development [28]. Safety Critical means that the system is hosting software that supports aircraft functionality assigned to *Development Assurance Level (DAL)* A, B or C during the Safety Assessment Process [29]. Mixed-criticality is the situation when the same equipment or system is hosting software from different aircraft functions, and they have been assigned to different DAL (A/B/C). In Avionics (the blending of the words aviation and electronics) there is a shift from federated avionics, where each aircraft function is implemented on its own processor to integrated modular avionics (IMA) where several aircraft functions share the same processor. Key characteristics for IMA platform is to provide robust partitioning of shared resources to guarantee that under no circumstances lower criticality software can affect higher criticality software in an adverse manner [26]. In parallel, avionics has followed Moore law and therefore, evolved considerably during last decades especially in the field of highly integrated SoC and multicore processors. Consequently, authority continuously updates means of compliance and corresponding guideline material. First, CAST-32 position paper [30] and recently MCP CRI Issue 3.0 [31] by EASA fully harmonized with the FAA equivalent issue paper. Due to their potential processing performance capabilities, SoC platforms and multicore processors (MCP) are expected to be used in future IMA platforms, however there is currently a considerable uncertainty about determinism compared to the traditional single-core processors and power handling for MCP referred to as *Dark Silicon* [32]. Consequently, a major concern of applications running on different cores with simultaneous access shared resources for power efficiency may leads to interference which violates the robust partitioning which is a key characteristic of IMA.

To summarize ARP 4754 defines the overall rules and guidelines that MCRTES has to be meet at system and equipment level. Power efficiency techniques, algorithms and related hardware support in the SoC platform (with shared resources) can be identified in [31] and the corresponding objectives must be addressed.

IV. STATE OF THE ART LOW-POWER TECHNOLOGIES

Low power design in multiprocessor system on chip can be achieved at different design levels such as the system software, application, system architecture including processing elements, memory and interconnection architectures, circuit implementation and design, operating voltage and frequency levels, standard-cell and family logic design level, and the process technology level. In the following we will take a look at the state of the art (SoA) research concerning these issues identifying the shortcomings and the challenges to overcome these. A summary of the main challenges, identified in this section, is depicted in Tab. II.

TABLE II
SUMMARY OF THE SHORTCOMINGS IDENTIFIED IN THE SOA RESEARCH

	Shortcomings	*Benefits when addressed*	*Challenges*	ID
System Software	Power management techniques in combination with hypervisors and the resulting restrictions/impact not yet addressed	Optimizing the usage of low-power techniques for MCRETS (partitioned systems managed with hypervisor)	Impact of transition time between different power consumption modes in terms of time and energy should be assessed.	SS1
	Applying global energy management in a (at design) statically scheduled MCRETS system is difficult	Taking use of the local (tile) and SoC global status for optimizing the power consumption on the global (SoC) level.	Global scheduler decisions must be known and analysed beforehand.	SS2
Chip-Level Hardware	System-level design methodology for MCRTES with NoC based communication have not been yet addressed	Supporting an automatic MPSoC generation with application-tailored parameters towards low-power predictable NoCs	Developing generic approach for optimizing MPSoC generation for classes of applications is difficult.	HW1
	Current on-chip power monitors are of low resolution	Enabling analysis of short phases of SW applications	• Benefits of power savings cannot be fully explored because, in general, they are not fully predictable or observable. • Complex designs and high costs	HW2
Dependability	No complete safety process in the overall design flows of related projects considering MCRETS complemented with Techniques considering extra-functional properties (e.g. power and temperature)	Earlier and more efficient certifiability analysis with significantly lower risks of mixed-critical systems	• Ensure that low-power techniques do not jeopardize the safe operation of the system. • Dynamic reconfiguration is not recommended • Faults need to be detected and isolated in order to avoid their Propagation. • Redundancy has to be used to improve the reliability of systems	D1
	Few approaches exist to assess the interoperability of secure measures and power consumption in MCRETS	More secure systems with lower power consumption	More secure versions of the same cryptographic algorithm are also more power hungry	D2

A. Power-efficient System Software for Mixed criticality Systems

1) State of the Art:

a) Execution Environment: Execution environments running applications with different criticality levels shall implement power management strategies encompassing energy and time budgets in the entire system. These execution environments are the underlying software that manage and control the hardware devices and offers services for application programs. These environments can be hypervisors, operating system or simple runtime systems. 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 [7]. However, although in the literature there is not any reference to hypervisors due to its recent appearance, some of the traditional real-time techniques used by the operating systems can be moved from the OS layer to the hypervisor layer in a partitioned system. This presents some new opportunities for new techniques based on the collaboration of both layers (hypervisor and partition OS). The proposed power-aware techniques can impact at different levels such as I/O, memory, processor management and network.

From the operating system point of view, the memory management can impact in two main issues: allocation of applications and management of memory types to reduce the energy consumption. In [33], it is focused on the amount of memory and the need of saving memory by compressing pages of memory. It requires from the OS the virtual memory management unit (MMU) to store and load compressed pages. In [34], hardware mechanisms for compression of data between cache and RAM are proposed. These approaches are intended to be performed via hardware and only in specific points of the execution. Dynamic memory allocation or dynamic storage allocation (DSA) has been a relevant part of the OSs for allocating memory to applications. The allocator algorithm is

crucial for memory allocation and two main problems arise: temporal cost of the allocation and space usage. [35] presents a survey of these techniques. The TLSF allocator, which performs the allocation and deallocation in constant time and achieve bounded fragmentation in the system, is proposed in [36].

The processor management at OS level can have an impact on the energy consumption via scaling CPU voltage and frequency, and setting the processor in a low power state or switching-off during the time intervals of no activity. DVFS, clock and power gating techniques can be used for this purpose and can be extended on the management of devices. In the recent years, many techniques, which decide an on-line or off-line schedule to guarantee the task deadlines while minimizing the CPU energy consumption, have proliferated in journals and conference papers ([7], [11], [37]). Considering the transition time between different power consumption modes is critical. For e.g. 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) Power-efficient Scheduling: Several scheduling techniques have been proposed to save energy while guaranteeing task deadlines. Some of these techniques compute the *slack* available and adjust the frequency or/and voltage to reduce the slack. In general, high priority tasks are executed at higher frequencies to generate more slack and adjust lower priorities task to reduce the frequency. The unused processing time in a system is called slack and can be categorized in two types: dynamic and static. The latter exists due to spare capacity since the system is loaded less than what can be guaranteed by the schedulability tests. Differences between the worst-case assumptions and the actual behaviour result in dynamic slack [38]. Lin et al. [39] presented four principles for effective slack management and developed four slack scheduling algorithms for Earliest Deadline First (EDF)-based systems that support mixed criticality. The principles are:

1) to allocate slack as early as possible with the priority of the donating task,
2) to allocate slack to the task with the highest (earliest original) priority,
3) to allow tasks to borrow against future resource reservation (with priority of the job from which the resources are borrowed) to complete their current job,
4) to retroactively allocate slack to tasks that have borrowed from their current budget to complete a previous job.

Using these principles can reduce the average deadline miss ratio by up to 100%. The authors of [38] improved these algorithms to reduce power consumption in combination with DVFS such that the system can consume available slack in idle mode. Another techniques use non-linear optimization [40] to find the optimal frequency for every task. This technique, however, has large complexity and hence is only suitable for the off-line use. In [41] the authors describe three types of scheduling techniques. The first type controls DVFS and DPM to dynamically throttle voltage and frequency of the CPU or temporarily suspend its operation. The second one performs thermal management. It primarily relies on the placement of threads in cores to avoid thermal hotspots and temperature gradients. At last, asymmetric systems are depicted. These systems are built with low-power and high-power cores on the same chip executing the same binary. The goal is to assign threads to cores according to their requirements on resources. All the algorithms discussed need dynamical monitoring properties of the workload to make decisions that consider the characteristics of the interplay between hardware and workload, and controlling the configuration and allocation of CPU cores to make for a best trade-off between performance and power consumption.

c) Global Energy Management: Different approaches were studied taking use of the local (tile) power management techniques for optimizing the power consumption on the global (SoC) level. [42] focuses on optimizing each tile for itself to lower the systems overall energy consumption. Their approach is to optimize the scheduling procedure by using an on-line scheduler which at run-time applies heuristics to use less pessimistic WCETs. This approach is not applicable to safety-critical applications because on-line scheduling cannot guarantee that real-time constraints are always met. [43] proposed an inter-tile slack propagation to optimize the effect that slack has on the system. This way, the slack can be used by all cores to perform DVFS to execute their tasks at a lower frequency. This approach is yet missing temporal predictability and a certifiable technique to be applicable to safety critical applications. A different approach is the super scheduler proposed by [44], [45] which injects new high criticality messages into the system to deal with critical events while maintaining the deadlines. The critical events get downgraded and interrupted by the new events. This super scheduler allows the system to react dynamically to environmental changes, by monitoring a pre-scheduled execution by an on-line scheduler. The super scheduler disturbs the current high critical applications and can lead to a failure of the system stability with a rate of 0.3.

2) Shortcomings: 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 off-line.

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 hierarchical scheduling at hypervisor level (static and based on cyclic scheduling) and OS (based on static or dynamic priorities) [46]. 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. As far as we know, no other related work examined the involvement of hypervisors and above approaches in the context of MCRTES.

Applying global energy management in a (at design) statically scheduled system is difficult as the schedule does not allow the system to vary greatly at runtime. As far as we know, the current approaches try to avoid this limitation by either injecting additional unplanned messages or using an on-line scheduler. These approaches are not feasible in safety critical systems where all system states must be known beforehand. Accordingly, the main challenge in this area is to introduce a pretend dynamic behaviour, where the system is able to reach a number of different predefined schedules according to its state at runtime.

B. Power-efficient Hardware for Mixed criticality Systems

Several techniques have been developed over the last two decades leveraging established principles of reducing the voltage swing and switching of transistors per unit of time and overall used transistors [47].

1) State of the Art:

a) Low-Power Embedded Processors: To increase performance and provide scalability, nowadays multicore systems are widely used. Multicore systems are more complex than single cores, e.g. high level caches must be coherent, so some of the low-power approaches only consider single-core systems. In future work, these techniques must be applied to multicore systems. There has been much of work applying different technologies to standard processors families, commonly used in the industry, to improve their low-power features. Some of these low-power processors to mention: ARM Cortex-A (Application), Cortex-R (Real-time) Cortex-M [48], TI *MSP430* [49], Renesas RL78 [50], Atom D525 [51] and Quark SoC X1000 series [52]. All above embedded processors are designed for low power. Some using RISC architectures with fewer transistors (e.g. ARM), others offering different power modes and power services. For instance, one interesting low-power feature of the ARM family is the *big.LITTLE* technology. The *big.LITTLE* processing is a power-optimization technology where high-performance ARM CPU cores are combined with the low-power and more efficient ARM CPU cores to deliver peak-performance capacity, sustained performance, and parallel processing performance at lower average power.

b) Cache and Scratchpad memories: Scratchpad memory has been used as a partial or entire replacement for cache memory due to its better energy efficiency and predictability. Scratch-Pad Memory (SPM) is intended to avoid the main drawbacks of caches. They consist of small, fast memory areas (SRAM...), very much like caches, but are directly and explicitly managed at the software level, either by the developer or by the compiler. Hence, no dedicated circuit is required for SPM management. This would mean that there is even a deterministic behaviour which is not provided by typical cache implementations. Deterministic behaviour is a major benefit for safety related applications. In [53], a comparison of several SPM with their advantages is presented. One of these advantages is the important reduction of the energy (up to 40% less energy than caches). In [54], a survey of techniques for SPM management is detailed.

c) Power-efficient On-Chip Communication: Energy-efficiency for interconnection networks can be achieved at design time by the appropriate choice of interconnection architecture. Several interconnection architectures exist including: point-to-point, bus and Network-on-chip (NoC). The optimum energy-efficient architecture may vary depending on the number of communicating processing elements, bandwidth throughput and latency requirements, the pattern of communication and the target hardware platform. Lee et al. quantitatively evaluated energy, and latency and bandwidth trade-offs for the interconnection architectures using MPEG2 encoder as a case study on FPGA device target [55]. According to this work, power and energy consumption was minimum for the case of NoC when the degree of communication parallelism is eight. Similar results and the promising scalability of NoCs [56] motivated research on low power interconnect design based on NoCs as described in the work of Silvano et al. in [57]. Nonetheless, there are further opportunities to reduce the power consumed by NoC (which can be up to 39% of overall power consumption [58]) to yield considerable system wide energy savings. Consequently, specific low-power design techniques have been exploited. These low power techniques can be categorized in *application mapping*, *topology selection* (architecture), *network interface*, *routing* algorithm and router design (transport layer), *flow control* (data link layer) and *links design* (physical layer).

At the *physical layer*, Lee et al. showed that the employment of low-swing signaling links, mux-tree based round-robin scheduler within the router, crossbar partial activation and low-energy coding for serial links can achieve significant energy

saving for instance up to 38% in their work [59]. At the *device level*, promising results for further energy savings can be achieved for the choice of proper electron device such as the VeSFET for 3-D NoC based multicore architecture [60].

At the *network layer*, several design decisions with regards to the routing algorithm and router design can be made [61], [62]. For example, Hesham et al. suggested that packet switching is more energy-efficient for streaming applications compared to circuit-switching routing scheme whereas for discrete packets circuit-switching performs better [63], [64]. Other researchers suggested that an optimum energy-efficient architecture could be a heterogeneous one, i.e. incorporating a mixture of router designs and routing schemes, depending on the application communication pattern [65], [66].

At the *architectural level*, researchers carried out evaluation with regards to the network topology and reported that the latency and energy of packets can change depending on the topology [67]–[69]. For instance, considering *FFT*, *Romberg integration*, *object recognition* and *poisson traffic* applications, *mesh* topology is shown to achieve the lowest energy dissipation compared to *torus*, *star* and *fat-tree* topologies. Custom topologies for specific applications have been shown to achieve better energy-efficiency such as for the case of the *ZMesh* architecture [70].

Due to the dependence of energy consumption on the mapping of the application's processes onto the NoC, several algorithms have been developed for static mapping of processes such as [71]–[74] and for dynamic mapping such as [75], [76] including that with specific consideration for mixed-critical systems [74].

There has been a lot of research done in adapting and improving DVFS power management technique for NoCs. Some of them are described and compared in [77]. These solutions take advantage of e.g. communication idle states during computation and memory accesses, to either globally reduce energy consumption via applying adaptive design techniques on local NoC units, or to apply core-wise DVFS. Also additional hardware is required like a Power Management Unit (PMU) that controls the generation of the supply voltage and clock. One disadvantage of DVFS is that due to increased execution time also leakage energy rises [38]. An optimum DVFS strategy should therefore, consider the introduced energy losses due to extended static power consumption. Control Earliest Deadline First schedule [78]. DVFS is often combined with other techniques. For example, in most cases memory limits the reduction of frequency and voltage in the whole system. Using voltage islands is lucrative since communication and memory can run at different voltages such that both are safe and meet their throughput requirements [79].

System-level exploration of run-time power clusterization, as presented in [80], increases energy efficiency of on-chip communication using an adaptive system architecture for power management called *Dynamically-Clustered DVFS* (dynamic voltage and frequency scaling), for short DCDVFS. At runtime, overburdened or idle network regions are identified and reconfigured with new power schemes. This method improves *Voltage Island partitioning* (V/F partitioning), where spatial locality of communication traffic on a parallel platform is exploited. The benefit of DCDVFS is that clusters are configured at runtime while in V/F partitioning the islands are defined at design time. On that score, spatial variations of communication traffics are also considered. Simulations on an 8×8 mesh Network-on-Chip (NoC) and a $65nm$ technology model extracted from Orion 2.0 show that the approach achieves much lower energy for traffic with spatial variations compared to existing approaches (9% to 42%). Besides, the approach incurs a moderate and predictive latency and minimal area overhead.

d) Power measurement and monitoring support: In general, power measurement and monitoring support is indispensable to circumvent inaccuracy when deploying analytical power analysis tools since estimates obtained by these methods can deviate from the actual power consumption of the working MPSoC. Another advantage of power measurements is their usage by power managers at run-time to monitor the power consumption the system (or certain subcomponents) in order to optimize power consumption.

Yet, power estimation of an MPSoC is not an easy task due to the lack of observability. In many cases, power measurement can only be performed at the MPSoCs power rail inputs and no direct relationship between the running software and measured power consumption can be established. The measurement of the electrical current consumption of a given system can be performed either via the usage of *shunt resistors* or using a *clamp meter* [81]. In the first approach, shunt resistors, having very low resistance (for e.g. 1mΩ) and a high accuracy $\pm0.1\%$ are connected in series with the positive supply lines [81]. They are deliberately chosen with low resistance in order not to influence the load supply, offering at the same time an interface to measure the current supply. By knowing the current and the supply voltage the power consumption can then be easily calculated. In the *clamp meter* based approach, a clamp meter device measures (based on the hall-effect phenomena) the induced magnetic field variations around the supply wire and uses the measured values to obtain the electric current [81]. While the clamp-meter based measuring is less intrusive, the shunt-based measurements are more accurate and less eligible to noisiness. Both techniques require either ADCs or digital multi-meter to sample the analog signals measured.

Various approaches can be found in the literature utilizing measurement-based methods for the direct evaluation of power consumption of MPSoCs or for the validation of power models (estimations). According to [81] a combination of infrared imaging, and electric current measurement techniques can yield high-resolution spatial power maps of individual parts for a given circuit. A mathematical foundation taking the thermal map and the current measurements and outputting the corresponding power maps is given in [81]. The infrared imaging uses infrared technology to obtain a thermal profile of the individual circuit components. By knowing the thermal behaviour of a certain circuit and its heat diffusion to the ambient temperature, the power consumption can be obtained

(e.g. using least-square estimation see [82]). This method is considered the most flexible since it is non-invasive and does not require extra design setup. In [83], a laboratory power supply with built-in measuring device (Keithley SourceMeter 2400) is used. In [84]–[87], oscilloscopes connected to shunt resistors are used to measure power consumption. In [88], a cycle-by-cycle energy measurement in FPGAs based on switched capacitor, is presented. This approach, achieves a high resolution of the measured values (every 20 ns). Also another measurement approach presented in [89] achieves a high resolution and is capable of measuring SW applications with detailed granularities when running on FPGAs. Using oscilloscopes or specialized power measurement device can obtain accurate results with high resolutions but has the disadvantage of high costs. In [96], a low-cost measurement methodology for measuring the power consumption of an FPGA-based MPSoCs which enables measuring the execution time and power consumption of multiple SDF applications at different granularity levels running on an FPGA-based MPSoC. Above approaches (depending on external measuring devices) are useful to compare the actual power consumption with the estimated one. Yet, unlike on-chip power monitors, these do not enable an on-line monitoring of the power consumption of an application running on an MPSoC for optimization purposes.

Schreiner et al. [90] used an FPGA board with an integrated measuring electronics for this purpose. The on-board sensor of the used FPGA samples the power with a very low sampling rate of 6.25 Hz. Other work in [91]–[93] utilized the on-board power monitors of a modern Xilinx FPGA ZC702 board [94] for measuring the power consumption and evaluating the efficiency of their approach. The ZC702 samples the power rails with the help of the integrated power controller Texas Instruments UCD9248 [95] every 200 μs (i.e., at a frequency of 5 kHz). Typically, such on-board power monitors have a low sampling rate to perform detailed analysis of short phases of software execution on the MPSoC.

2) Shortcomings: There have 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 behaviour of the NoC itself (see Sect. IV-B1c). Taking those various low-power design techniques into account, a systematic design methodology can be adopted to achieve low power performance at design time such as in the work reported in [97]. In such design methodologies, power and performance models such as in [98] play crucial role. Pessimistic performance assumptions could result in higher than expected run-time slacks and dynamic power management could thus be used [99]. In order to achieve further power reduction, power consumption can be further managed [100] with the aid of run-time monitoring of performance and energy of the network-on-chip [101]. Depending on the state of the NoC, the power manager can change the voltage and frequency of system components and packet routing in a node-centric or network centric manner to save energy. Examining such a system-level design methodology for MCRTES with NoC based communication have not been yet addressed. In addition, supporting an automatic MPSoC generation with application-tailored parameters towards low-power predictable NoCs have not been addressed by typical MPSoC generators (c.f. [102]–[105]). This is of high interest, not only because it raises the technology readiness level (TRL) considerably, but also allows the exploration of the predictability and thereby the safety of the final system.

In addition, on-board power monitors of higher resolution, than the currently existent (c.f. Sect. IV-B1d), could enhance the capability to assess current system status and open the way towards various low-power optimizations.

C. Impact of Low-power Capabilities on Dependability

In the following an investigation on safety and low-power is presented and, additionally, a brief study on security and low power is also presented.

1) Safety, low-power and mixed criticality: Due to the explosion of autonomous systems thanks to the big improvements on the energy storage technologies (e.g., batteries) or purely motivated by energy budget requirements, power efficiency and power management are also very interesting cost competitive features for safety critical systems. In fact, the power management (and temperature) of embedded electronic components is also closely coupled with its lifetime. A proper (and low) power demand of a specific hardware component could prolong its lifetime (w.r.t. permanent faults) and, directly, the intrinsic reliability of the system. In fact, temperature monitoring is suggested as a major diagnostic element when using an on-chip redundancy for safety proposes (e.g., IEC-61508-2 Annex E). One can address those requirements with external chip external monitoring components or with a more efficient way using ring oscillator if the target device is an FPGA [106]. Directly linked with temperature and power consumption, the embedded system cooler could also have a significant impact on the embedded system reliability. If the embedded system power consumption (and proportional heating) requires a non-passive refrigerator system with mobile parts (like a ventilator) the overall embedded system reliability could be undermine due to the fault-prone nature of such coolers. Finally, even if power scaling (e.g., usage of very low-voltages) could be beneficial to decrease the probability of permanent faults (e.g., aging faults), this low power sensitive logic is more vulnerable to transient faults, such as, soft-errors (but, this sensitivity could be compensated with new micro-electronic architectures).

Multicore technology is also a way of overcoming Pollack's rule and reduces scaling limits of single-core processors. When moving towards mixed-criticality embedded systems, there are fundamentals on multicore and safety that has been extensively researched in several European Projects. For example, the safety-concept approach within MULTIPARTES, PROXIMA and DREAMS [4] EU projects proposed an argumentation for the use of multicores for mixed-criticality system considering spatial and temporal isolation among partitions mapped to different cores, but the impact of temperature or power was

not explicitly analysed. In fact, this safety-concept approach is an effective way to establish a formal dialogue with a certification authority and move away from the academic safety-certification analysis with a rigorous safety argumentation. This early contact with certification authorities identifies possible conflicts w.r.t. to certification standards and paves the way to the future industrialization of the technology. In the case of the CONTREX EU project [18], current activities in the area of predictable computing platforms and segregation mechanisms were complemented with techniques considering extra-functional properties such as real-time, power, and temperature for safety/certification in mixed-critical systems. In contrary to the SAFEPOWER proposal, while some safety measures were partly considered, no complete safety process was integrated to the overall design flow of the CONTREX project.

The SAFEPOWER approach aims to perform a detailed analysis of low-power mechanisms on specific COTS platforms (e.g., Xilinx Zynq) and identify the conflicts (e.g., through dedicated FMEAs) of that those mechanism raise on specific processor elements and, subsequently, to the application safety.

Above safe power management cannot be done, according to the product life-cycle, taking on-line decisions. 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 [107] 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 [108], for instance, safe startup and shutdown scenarios are considered for an IEC 61508 compliant hypervisor partitions, but not primarily for power management proposes.

2) Security, mixed-criticality and low-power: Classically, safety-critical systems have been considered close or semi-close systems with very limited and controlled interactions with their environment. Current embedded systems and, particularly, mixed-criticality with their non-safety related part are more connected to open networks (e.g., local networks, wireless networks, the cloud). In fact, even the safety standards have started considering the inclusion of security aspects on their life cycle. In this mixed-criticality area, there are several hardware and software mechanisms to protect critical parts from the non-secure ones. For instance, in software, the same spatial and temporal separation mechanisms used on hypervisors to isolate partitions from design faults could prevent attacker to access safe (now also secure) partitions from the non-safe (or non-secure) partition. The US Government has a protection profile for separation kernels in environments requiring high robustness [109] which is commonly known as the Separation Kernel Protection Profile (SKPP). Separation kernel is defined by SKPP as "*hardware and/or firmware and/or software mechanisms whose primary function is to establish, isolate and separate multiple partitions and control information flow between the subjects and exported resources allocated to those partitions*" [109]. It has to be proved that there is not any unexpected channel for information between domains. This protection profile specifies the security evaluation criteria so that a given system, in case is compliant with, can be certificated under the Common Criteria (also called IEC-15408) standard. It has to be mentioned that the Common Criteria certification does not assure security, albeit it guarantees that the declaration and specification about the system given is true or not [110]. One of the commercial real-time operative systems which is compliant with this protection profile is INTEGRITY-178B by Green Hills Software Inc. [111] This system was used as baseline to partly implement a software crypto demonstrator in the separation kernel by J. Frid [112]. In addition, a state of art concerning separation kernels from a historical and technical perspective is provided. Similarly, although in hardware, the ARM Trustzone technology [48] is able to separate the execution environment between two different worlds: secure and normal (non-secure). This security feature is achieved by dividing all the hardware and software resources of the system on chip so that they exist in those two worlds. The system is designed in such way that it ensures that no secure world resources can be accessed by the normal world elements. However, secure world resources have access to the non-secure ones. Thus, employing this technology, a single physical core is able to securely and efficiently execute code from both secure and normal worlds, which removes the necessity of another dedicated processor core.

Security and low-power are coupled in the sense that more secure versions of the same cryptographic algorithm are also more power hungry. In [113] [114] [115], one can see the different comparisons of several cipher algorithms and their performance depending on the power consumptions. The power consumption itself could be also a trace for attacker to get information on the encryption algorithm and a way to hack secret key. In [115], dynamic voltage and frequency scaling (switching) is used to distort the power consumption trace and further protects the secret key integrity. A kind of attack could also consider to hack the system so e.g., requesting to perform a task that increases the consumption and makes the system out of battery, but few bibliography have been found on this track.

V. SAFEPOWER REFERENCE ARCHITECTURE

In the following, we will first discuss how the identified challenges of the state-of-the-art will be addressed in context of SAFEPOWER. Then we will take a look at the reference architecture of the SAFEPOWER and elaborate on the related low-power orchestration. In addition, the planned use-cases with first sketches and benefits will be presented.

A. Challenges addressed in SAFEPOWER

The goal of SAFEPOWER is to decrease the power consumption of MCRTES up to 50%, while maintaining the necessary operation requirements identified in Sect. III. 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. 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. 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.

In the following, we will describe how SAFEPOWER will address the shortcomings identified in Tab. II. To deploy low-power safety-critical approach at embedded system level there is a main research gaps and challenges that have to be addressed. No research (to writers knowledge) has considered the impact of low-power mechanisms into COTS embedded processors targeting safety-certification. This includes:

1) the impact of the different low-power techniques into different COTS embedded processor elements (see SS1 in Tab. II),
2) the implications of those impact into system safety and mixed-criticality (see D1 and D2 in Tab. II).

By addressing the certification issues (identified in Sect. III-D) early at low Technology readiness levels (TRL), the introduction of SAFEPOWER research results into industrial products (avionics, railways) is expected to enable an earlier and more efficient certifiability analysis with significantly lower risks.

Using hypervisors can offer a simpler and flexible solution in energy-constrained mixed-criticality systems (see SS1 in Tab. II) which can tackle the shortcomings identified in Sect. IV-A. The hypervisors such as [46] provides virtual machines where the resources allocation are predefined and restricted based on earliest analysis. Nowadays, this resources allocation is focused only in the CPU time, memory, interrupts, Input/Output devices, partitions communication and fault processor management. This resources assignment can be extended to energy features, where for instance, operational processor frequencies can be restricted by virtual execution environment, and it can be extended to the device management where low-power techniques such as clock-gating can be applied. Low-power services would be provided and controlled by the hypervisor, which is who has the global view of the system (see SS2 in Tab. II). In this way, the energy management would be provided to two levels: at hypervisor level, focused mainly in assuring the availability and execution of critical applications and in energy saving; at partition level, providing flexibility and increasing the energy saving based on local actions supervised by the hypervisor.

State-of-the-art mixed-criticality architectures provide partitioning and real-time guarantees, but they are limited to predefined execution schedules (see SS2 in Tab. II). An important research problem is the development of specific adaptability mechanisms for mixed-criticality applications providing semi-dynamic behaviour for safety-critical applications allowing them to react to environmental or intrinsic events. In addition, safety-critical application subsystems could benefit from adaptability to use it for certifiable fault recovery strategies. One idea to achieve this is to mimic a dynamic behaviour where a meta-scheduler switches between different predefined schedules according to a runtime system state.

We plan in the context of SAFEPOWER to use the NoC system generator [116] to support applications modelled with synchronous models of computation, and generate implementations from Simulink models [117]. Inside the SAFEPOWER project, the NoC system generator will be extended (see HW1 in Tab. II) to support techniques for low-power predictable NoCs. This enables the usage of analytical DSE-tools to explore the design space of such designs also targeting power optimis ations. Furthermore, an integration of a DSE-tool like DeSyDe [118] into the NoC-system generator would move system design to a higher level of abstraction, 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.

In addition, power management monitors will be developed to assess the power consumption of the MCRETS (see HW2 in Tab. II). Moreover, health monitors will be implemented to achieve fault isolation and insure safe operation (see HW2 in Tab. II).

B. Reference Architecture

SAFEPOWER mixed-critical architecture is depicted in Fig. 2 consisting of a number of tiles connected to each other via Network-on-Chip (NoC). Our architecture is chosen to enable strict temporal and spatial separation and to ensure that no low-critical application can impact or delay a higher critical one or access its dedicated resources. To insure this, each tile (see $Tile_x$ in Fig. 2) is managed by a bare-metal hypervisor (XtratuM) running on the tiles hardware and allowing for arbitrary amounts of partitions. The applications running in these partitions can be of varying criticality. Yet, in order to realize the spatial separation, only applications with the same criticality can be executed in a single partition. The hypervisor assigns hardware resources of its tile to its managed partitions, ensuring the timing and spatial separation for the mixed-critical system. It also manages the partitions by scheduling their execution by the corresponding deadlines to provide the real-time functionality. Since the partitions do not really access the hardware on their own, the messages the partitions send are also relayed by the hypervisor. This prevents message flooding to the NoC and the applications of other tiles are secured from malicious behaviour. Each tile is connected via a network interface (NI) to a time-triggered NoC. The time-triggered behaviour is executed based on an a priori computed communication schedule which commands the message injection times. This schedule ensures that no packets collide during the transmission and guarantees timing bound for the message traversal. For this, it does not only provide the message injection but also the path the message is routed on.

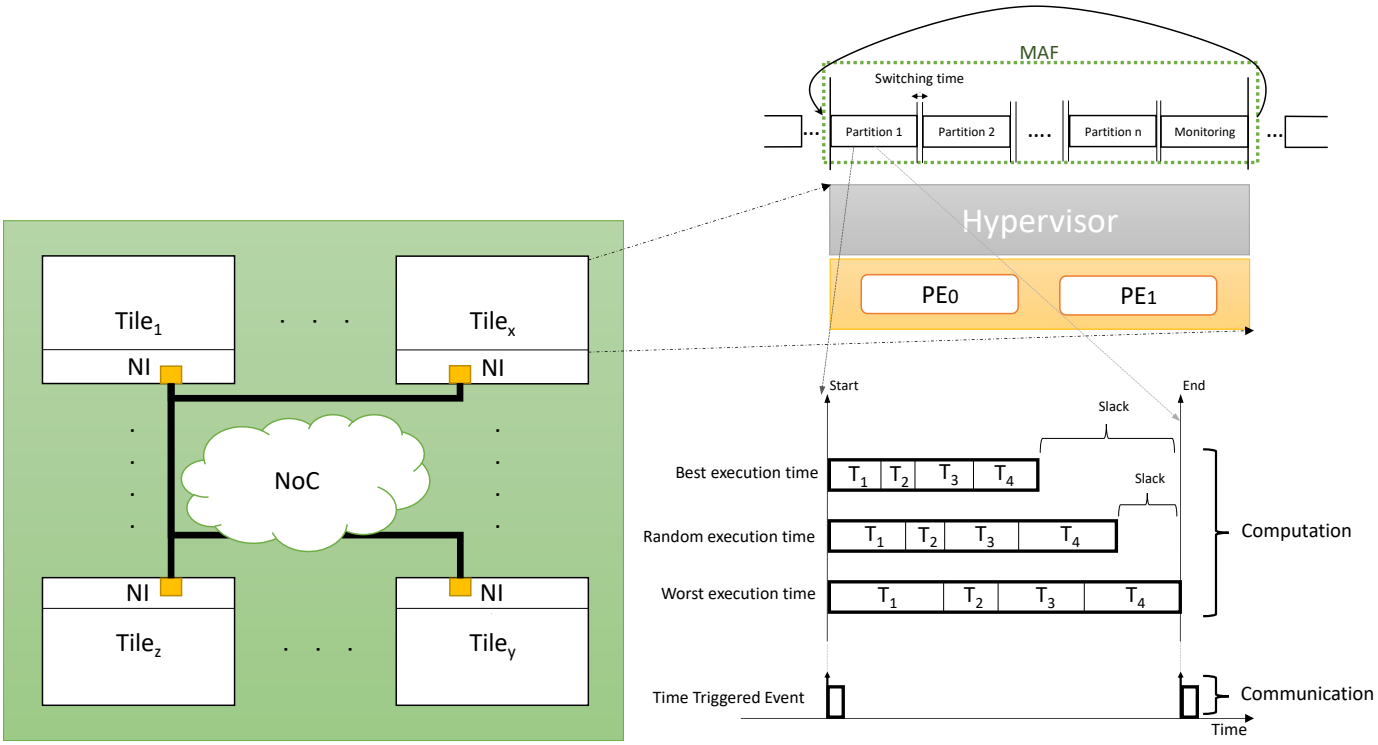

Fig. 2. SAFEPOWER time-triggered architecture: *Left:* tile-based architecture connected via NoC. *Right-top:* example of a tile consisting of a dual-core platform managed by an hypervisor. *Right-bottom:* example of statically scheduled group of tasks, running within a partition, with variable slack times in each scenario of execution.

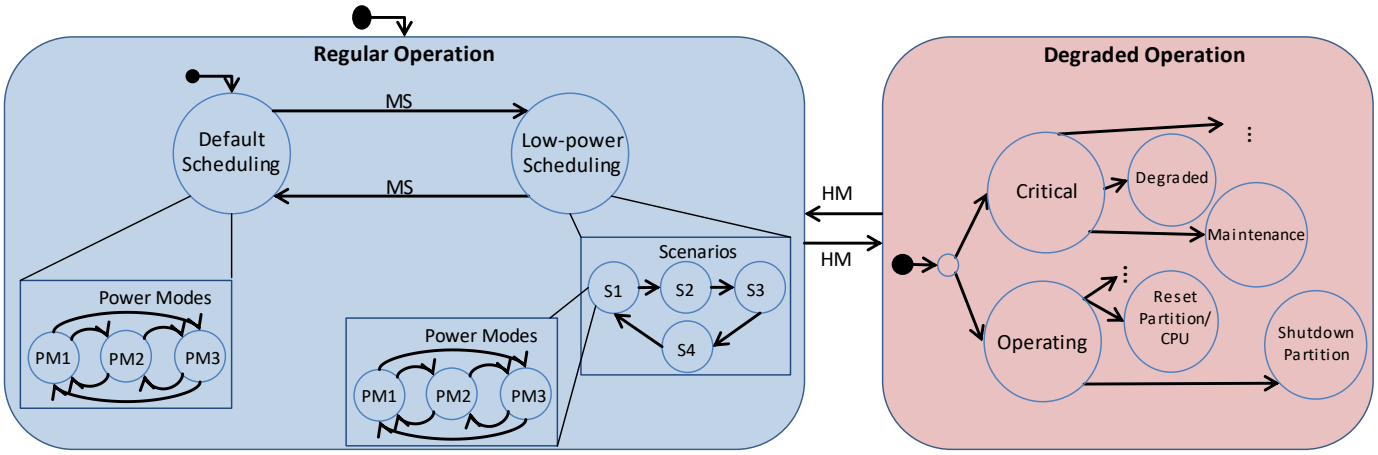

Fig. 3. Hierarchical State Machine (HSM) representation of the low-power orchestration within SAFEPOWER architecture. The HSM consists of two super states: *Regular* and *Degraded* operation where the system operates as default in the *Regular* state. Within the *Regular* operation state, a default schedule (obtained off-line) is initially run. Within this schedule, different *Power Modes (PM)* with different low-power techniques (e.g. DVFS, Run-Fast-Then-Stop (RTFS)) can be chosen depending on the current slack (see Right-bottom of Fig. 2). In addition, a meta scheduler (MS) can issue events, depending on the current system status, to switch between different schedules targeting a low-power schedule (for e.g. a schedule with suspended partitions). The MS events can be launched locally (at the scope of a tile) or globally (via a dedicated tile at the SoC level). If emergent *Health Monitors (HM)* events are detected (at the hypervisor level), the HSM changes to the *Degraded* operation. In the *Degraded* operation state, depending on the criticality of error, the system changes to one of two states. In case of minor error, it continues with a partial functionality of its nominal operation (*Partial Operating* state) taking measures to isolate error sources (e.g. via procedures such as completely shutting down partitions). In case of a major error, the system is driven into a *Critical* state (for e.g. maintenance: shutting down parts of the system and insure safe landing in case of a copter use-case with critical battery capacity).

In safety critical systems, consequently also in SAFEPOWER, it is guaranteed that all tasks are finished before the end of the partition. As shown in Fig. 2, in best-case scenarios, when a batch of tasks finishes early enough, the partition will have a reasonable amount of slack. However, in the worst-case scenario tasks end slightly before the deadline (i.e. Zero Slack). Since the tasks scheduling is done statically (at design time), the minimal slack[2] available is known to the designer. This slack time is very helpful in low-power system design. A designer can exploit this slack and applies a suitable low-power technique accordingly (e.g. power off) represented in different power modes (PM) in Fig. 3. The Hierarchical State Machine (HSM) orchestrating the different low-power operation states, as planned in SAFEPOWER, is depicted and described in Fig. 3. Beside all partitions running over the hypervisor, there is always a monitoring partition appended at the end each execution frame which finishes the major active frame (MAF). In the monitoring partition, health monitors (HM) are checked if anything went wrong (for e.g. errors in partitions).

In the SAFEPOWER architecture, computation and communication are separate from each other. A task starts when the input data is ready and when it finishes, output is ready to be sent. There is a periodic I/O event associated to each partition that defines the actual slack time. In analytical scheduling, tasks are guaranteed to be done before the next I/O event. At the end of the partition, the output data is always ready to be sent before the next periodic I/O event is triggered.

C. Use-cases

SAFEPOWER will 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.

1) Railway use-case: This use-case describes the development of an autonomous object controller on the railway signalling network. In a typical installation the electronic interlocking is responsible of controlling the trackside field elements directly connected to it. Large installations need to decentralize the interlocking operation using object controllers located near the field elements. In this case, although the safety logic is centralized in the interlocking, the object controller shall be able to supervise the field elements controlled by itself autonomously. The main field elements (see Fig 4) controlled by the object controller are signal lamps, powered points, rack circuits, axle counters, treadles, level crossings, hand-operated points and derailers, indicator lights, LECPs, tramway route request loops and push button boxes, etc. The main interfaces to control this field elements are inputs/outputs and Ethernet.

[2]For task t with deadline d_t and worst-case execution time $wcet_t$, the slack s_t can be calculated as follows: $s_t = d_t - wcet_t$

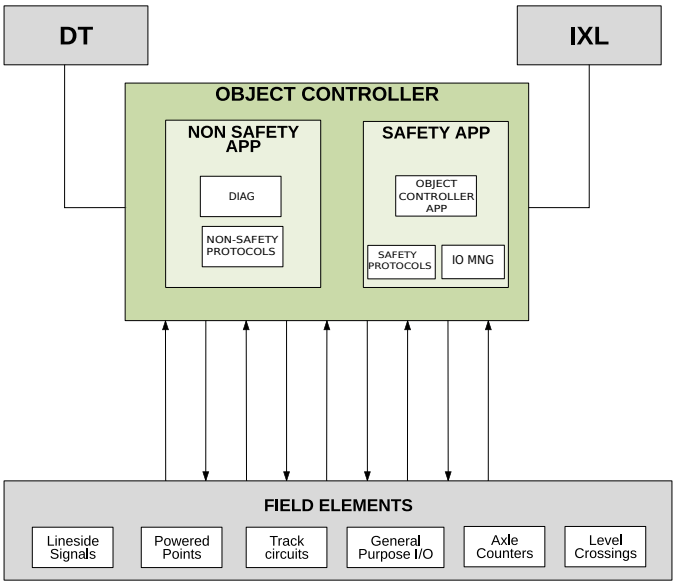

Fig. 4. Railway mixed-critical use-case: wireless object controller interacts with an interlocking (IXL) and diagnostic and maintenance (DT) systems.

The flexibility of combining into a variety of configurations is a key characteristic of an object controller.

As seen in Fig 4, the object controller exchanges information with the interlocking (IXL) and with the diagnostic and maintenance system (DT). Low-power consumption technologies for the electronics together with the use of an autonomous power supply and efficient and secure wireless communication would allow the object controllers to work with no need of any wiring between them and the interlocking. Therefore, the SAFEPOWER power-management will be crucial to provide enhanced availability at the desired safety-criticality levels.

The object controller will perform functions with different levels of criticality regarding the parameters of safety, security, reliability, availability and maintainability. For this reason, the implementation of this use-case will be based on the mixed critically architecture proposed by SAFEPOWER. This application will mix functionalities of different criticality levels that will be integrated on the SAFEPOWER architecture using the hypervisor and implemented core services. Reliability will be achieved by executing redundant applications and using redundant links with the interlocking. In this use case, we will take advantage of the connectivity, security and performance functions provide by SAFEPOWER platform. These techniques will be used to implement safety and non-safety related protocols.

The potential benefits of the application of an autonomous object controllers in the deployment of electronic Interlocking are very significant and are summarized below:

- Reduction of the deployment costs due to less wiring,
- Reduction of corrective maintenance costs and improvement of the service (i.e. due to theft),
- Reduction of system evolution and scalability costs due to better system modularity and scalability.

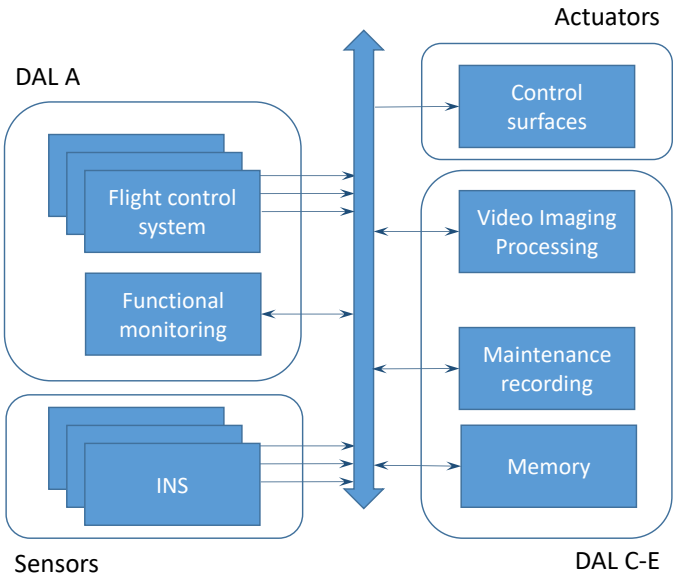

Fig. 5. Avionics use-case: Applications with falling DAL classification used for the evaluation of the SAFEPOWER techniques.

2) Avionics use-case:

Nowadays airplanes comprise a number of computer systems and the total power consumption can be substantial which requires a lot of cooling of the electronics. Cooling, in turn, requires also a lot of power. This extra demand of power requires resources that needs to be brought on the plane, which may influence on its performance in the end. It would therefore, be beneficial if the total power consumption could be lowered. The power management proposed by SAFEPOWER will provide a lot of information on how the power consumption can be lowered for applications with different safety criticality. The Avionics use case describes a set of applications with falling safety criticality, according to the Design Assurance Level (DAL) methodology, which is necessary to properly evaluate the use of SAFEPOWER techniques on a safety critical system for the avionic domain. In Fig. 5 the applications and their interconnection is briefly outlined. The system with highest DAL is the flight control system (FCS) and the functional monitoring (FM). The FCS provides all flight logic and laws for the flight and the integration of the requests from the pilot, sensor data and the control surfaces and the FM embodies the health monitoring of the system. For redundancy, the FCS is tripled. Three other applications has been added for their lower DAL classification. For the evaluation, it is relevant to achieve a spatial and temporal seggregation of the applications of the SAFEPOWER mixed critical avionic system. In addition, it is also relvant not to compromise the reliability and safety of the software during power saving operations on the hardware.

The potential benefits of avionics use-case are very significant and can be summarized as follows:

- Achieving a better understanding in how power saving techniques can be applied to multicores with NoCs without having the safety compromised for safety criticality systems.
- Executing safety-, mission- and non-critical applications on the same multicore platform with low-power services would result in increased payload fraction.

Fig. 6. Public demo multi-rotor system use-case

3) Public demonstrator use-case: The public demo use-case is represented by a multi-rotor system with four propellers (see Figure 6). This demonstrator is indeed a safety-critical system, since its carbon reinforced propellers are turning very fast and it can reach up to $60\,\mathrm{km/h}$ by a weight of $2.5\,\mathrm{kg}$. Its avionics is self-developed and based on a Xilinx Zynq 7020 MPSoC [94]. This chip combines an ARM Cortex-A9 dual-core and a programmable logic part for further needed hardware parts.

Next to the safety-critical flight algorithms, the Zynq 7020 executes a mission-critical video processing task on the video stream that is provided by the on-board camera system. For e.g. a pink football can be recognized and the camera gimbal is triggered to keep the football in center of the field of view. With reference to Section II-B, the multi-rotor system is a mixed-criticality battery-powered system. In that way it must meet the requirements identified in Section III. Especially the mission-critical video processing application is not allowed to influence the safety-critical flight control one . To ensure the segregation of the two applications and their error-free processing, the proposed architecture shown in Fig. 7 will be implemented in the Zynq 7020. Both, the ARM dual-core as well as the programmable logic will be used. All communication in the avionics will be handled by an NoC that is implemented in the programmable logic. The dashed arrows in the figure represent the main communication routes between the system components. The ARM dual-core will be managed by the XtratuM Hypervisor. In that way, flight algorithms and the video processing application will each use a dedicated ARM core. The video processing application gets and returns all data over an ethernet interface. Over ethernet, a camera, a gimbal and a wireless LAN connection will be accessible. As a further safety feature, a triple modular redundancy will be implemented for the flight algorithms. Next to the mentioned ARM core also two Xilinx MicroBlazes in the programmable logic will process the flight algorithms. All sensor and remote control data is collected by a dedicated MicroBlaze (*Dataminer*), that is only responsible for the data mining. In that way, the three processing elements will work

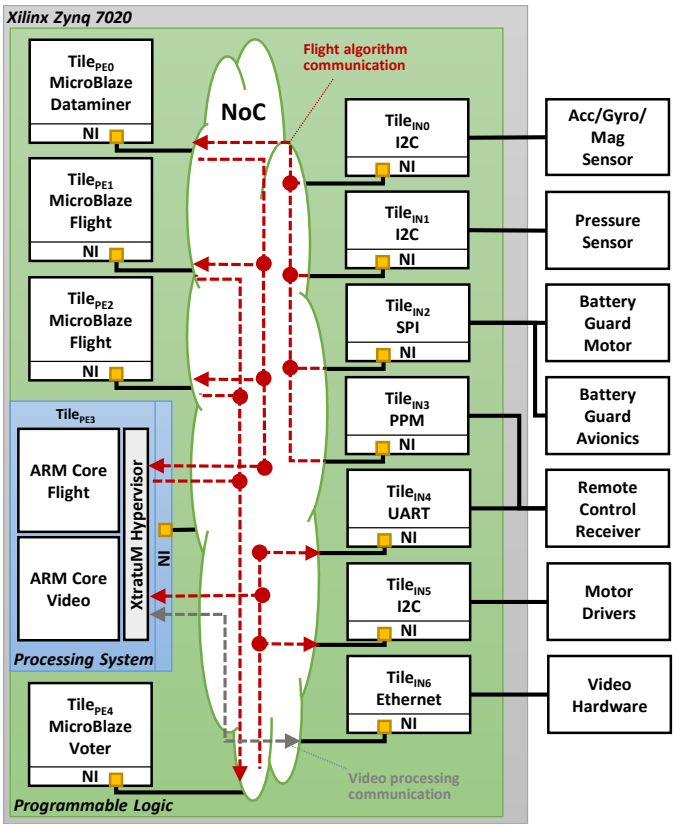

Fig. 7. Proposed architecture for public demo use-case

on the same inputs. Their results are processed by another MicroBlaze that is used as a *voter*, to recognize errors in at least one processing chain. The voter transfers the final setpoints to the motor drivers.

The public demonstrator will consist of the SAFEPOWER reference software and service stack integrated into the XtratuM hypervisor which can either be executed on the SAFEPOWER virtual platform offering a subset of the processors models, on-chip communication network, peripherals and I/O components or on the SAFEPOWER development and evaluation board, providing an interface for power measurement. The demo application will demonstrate the main feature of the platform, e.g.: running of different independent partitions (temporal and spatial isolated), managed by the XtratuM hypervisor with different power and thermal management services, advanced health monitoring and error reporting. It is also intended to enable easy and interactive access to this public demonstrator for the scientific community, all relevant software (virtual platform, XtratuM, SAFEPOWER services, cross-compiler and demo applications) will be provided as a virtual machine image.

VI. CONCLUSION

In this article, 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 planned use-cases. 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] S. Mittal. *A survey of techniques for improving energy efficiency in embedded computing systems*. IJCAET 6(4): 440-459 (2014).

[8] H. G. Lee, K. Lee, Y. Choi, and N. Chang *Cycle-accurate energy measurement and characterization of FPGAs*, Analog Integrated Circuits and Signal Processing, 42(3):239251, 2005

[9] L. Smarr, *Project greenlight: Optimizing cyber-infrastructure for a carbon-constrained world*, Computer, vol. 43, no. 1, pp. 2227, 2010

[10] 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.

[11] 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).

[12] *DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, Radio Technical Commission for Aeronautics, Washington, DC, Dec. 1992.

[13] *IEC 61508 Edition 2.0: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, IEC: Int. Electrotechnical Commission, 2010.

[14] J. Rushby, *Partitioning for avionics architectures: Requirements, mechanisms, and assurance*, NASA Langley Research Center, NASA Contractor Report CR-1999-209347, Jun. 1999.

[15] J. Rushby, *Modular certification*, Computer Science Laboratory SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA, Tech. Rep., Sep. 2001.

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

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

[18] R. Grgen et al., *CONTREX: Design of Embedded Mixed-Criticality CONTRol Systems under Consideration of EXtra-Functional Properties,* 2016 Euromicro Conference on Digital System Design (DSD), Limassol, 2016, pp. 286-293. doi: 10.1109/DSD.2016.95

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] EASA CS25, EASA, *Certification Specification. Acceptable Means of Compliance for Large Aeroplanes CS-25*, European Aviation Safety Agency (EASA), 2003.

[25] ARP4754A, S. A. E. *Guidelines for development of civil aircraft and systems.*, December 2010.

[26] RTCA/DO 297, *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*. RTCA, 2005.

[27] RTCA/DO 178C, *Software Considerations in Airborne Systems and Equipment Certification*. RTCA, 2011.

[28] RTCA/DO-254, *Design Assurance Guidance for Airborne Electronic Hardware*, RTCA, 2000.

[29] ARP4761, S. A. E. *Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment.* SAE International (1996): 1-331.

[30] *Position Paper CAST-32 Multi-core Processors*, Certification Authorities Software Team (CAST), May 2014.

[31] MCP CRI Issue 3.0, *The Use of Multi-Core Processors in Safety-Critical Applications, European Aviation Safety Agency (EASA)*, March 2016.

[32] Esmaeilzadeh, Hadi, et al. *Dark silicon and the end of multicore scaling.* Computer Architecture (ISCA), 2011 38th Annual International Symposium on. IEEE, 2011.

[33] Lei Yang, Robert P. Dick, Haris Lekatsas, Srimat T. Chakradhar, *Online memory compression for embedded systems,* ACM Trans. Embedded Comput. Syst. 9(3) (2010)

[34] Benini, L., Bruni, D., Macii, A., and Macii, E., *Hardware-assisted data compression for energy minimization in systems with embedded processors,* In Proc. Design, Automation & Test in Europe Conf, 2002.

[35] P.R.Wilson,M.S.Johnstone,M.Neely,andD.Boles. *Dynamic Storage Allocation: A Survey and Critical Review,* In H.G. Baker, editor, Proc. of the Int. Workshop on Memory Management, Kinross, Scotland, UK, Lecture Notes in Computer Science. Springer- Verlag, Berlin, Germany, 1995. Vol:986, pp:1116.

[36] Miguel Masmano, Ismael Ripoll, Alfons Crespo, Jorge Real, *TLSF: A New Dynamic Memory Allocator for Real-Time Systems,* ECRTS 2004: 79-86

[37] Pillai, P., Shin, K., 2001. *Real-time dynamic voltage scaling for low-power embedded operating systems,* In: ACM symposium on operating systems principles, Banff, Canada

[38] M. A. Awan and S. M. Petters, *Enhanced Race-To-Halt: A Leakage-Aware Energy Management Approach for Dynamic Priority Systems,* in 2011 23rd Euromicro Conference on Real-Time Systems, 2011, pp. 92101.

[39] C. Lin and S. A. Brandt, *Improving soft real-time performance through better slack reclaiming,* in 26th IEEE International Real-Time Systems Symposium (RTSS05), 2005, p. 12 pp.-pp.421

[40] Harada, F., Ushio, T. and Nakamoto, Y. (2007), *Power-aware optimization of CPU and frequency allocation based on fairness of QoS.* Syst. Comp. Jpn., 38: 3745. doi:10.1002/scj.20819

[41] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, *Survey of Energy- Cognizant Scheduling Techniques,* IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 7, pp. 14471464, Jul. 2013.

[42] Paterna, Francesco, Andrea Acquaviva, and Luca Benini. *"Aging-aware energy-efficient workload allocation for mobile multimedia platforms."* IEEE Transactions on Parallel and Distributed Systems 24.8 (2013): 1489-1499.

[43] P. Zaykov, G. Kuzmanov, A. Molnos, K. Goossens, *"Run-time Slack distribution for real-time data-flow applications on embedded MP-SoC"*. in Proc. of DSD, 2009, pp. 409-418.

[44] S. Ogrenci Memik, E. Bozorgzadeh, R. Kastner and M. Sarrafzadeh, *A super-scheduler for embedded reconfigurable systems* in IEEE/ACM International Conference on Computer Aided Design 2001. ICCAD 2001, pp. 391-394.

[45] Persya, A. Christy, and TR Gopalakrishnan Nair. *"Model based design of super schedulers managing catastrophic scenario in hard real time systems."* Information Communication and Embedded Systems (ICICES), 2013 International Conference on. IEEE, 2013.

[46] E. Carrascosa, J. Coronel, M. Masmano, P. Balbastre, and A. Crespo. 2014. *XtratuM hypervisor redesign for LEON4 multicore processor,* SIGBED Rev. 11, 2 (September 2014), 27-31.

[47] V. G. Oklobdzija, *Design for Low Power*, Wiley-IEEE Press, 1999, pp. 169259.

[48] ARM, ARM. *Security Technology Building a Secure System Using TrustZone Technology (white paper),* ARM Limited (2009).

[49] Texas Instruments, *MSP430TM Ultra-Low-Power Microcontrollers,* ed, 2014, p. 27.

[50] Renesas, MPUs & MCUs RL78 Family. Available: http://www.digikey.com/web%20export/supplier%20content/Renesas_559/pdf/renesas-rl78-brochure.pdf

[51] Intel, Atom Processor D525 (1M Cache, 1.8 GHz). Available: http://ark.intel.com/products/49490/Intel-Atom-Processor-D525-1M-Cache-1_80-GHz

[52] Intel Corporation, *Product Brief: Intel Quark SoC X1000 Series,* ed, 2014, p. 2.

[53] Rajeshwari Banakar, Stefan Steinke, Bo-Sik Lee, M. Balakrishnan, and Peter Marwedel. *Scratchpad memory: design alternative for cache on-chip memory in embedded systems.* In 10th international symposium on Hardware/software codesign (CODES02),pages 7378, New York, NY, USA, 2002. ACM Press.

[54] Maha Idrissi Aouad, Olivier Zendra. *A Survey of Scratch-Pad Memory Management Techniques for low-power and -energy.* 2nd ECOOP Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS2007), Jul 2007, Berlin, Germany. pp.31-38, 2007.

[55] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, *On-chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-point, Bus, and Network-on-chip Approaches*, ACM Trans. Des. Autom. Electron. Syst., vol. 12, no. 3, pp. 23:123:20, May 2008.

[56] 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.

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

[58] C. Li and P. Ampadu, *Energy-efficient NoC with variable channel width,* in 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS), 2015, pp. 1 4.

[59] K. Lee, S.-J. Lee, and H.-J. Yoo, *Low-Power Network-on-Chip for High-Performance SoC Design*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 2, pp. 148160, Feb 2006.

[60] V. S. Nandakumar and M. Marek-Sadowska, *A Low Energy Networkon-Chip Fabric for 3-D Multi-Core Architectures*, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 2, no. 2, pp. 266277, June 2012.

[61] A. Jantsch and H. Tenhunen, Eds., Networks-on-Chip. Kluwer Academic Publishers, 2003.

[62] W. J. Dally and B. Towles, *Principles and practices of interconnection networks. Morgan Kaufmann publications*, 2004.

[63] A. Banerjee, P. T. Wolkotte, R. D. Mullins, S. W. Moore, and G. J. M. Smit, *An Energy and Performance Exploration of Network-on-Chip Architectures,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 3, pp. 319329, March 2009.

[64] S. Hesham, D. Göhringer, and M. A. E. Ghany, *ARTNoCs: An Evaluation Framework for Hardware Architectures of Real-Time NoCs, year=2016,* in IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), May, pp. 259264.

[65] M. Kreutz, C. Marcon, L. Carro, N. Calazans, and A. A. Susin, *Energy and latency evaluation of NoC topologies,* in IEEE International Symposium on Circuits and Systems, May 2005, pp. 58665869 Vol. 6.

[66] G. Jiang, Z. Li, F. Wang, and S. Wei, *A Low-Latency and Low-Power Hybrid Scheme for On-Chip Networks,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 4, pp. 664677, April 2015.

[67] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, *Performance Evaluation and Design Trade-offs for Network-on-Chip Interconnect Architectures,* IEEE Transactions on Computers, vol. 54, no. 8, pp. 10251040, Aug 2005.

[68] M. Kreutz, C. A. Marcon, L. Carro, F. Wagner, and A. A. Susin, *Design space exploration comparing homogeneous and heterogeneous network-on-chip architectures,* in 18th Symposium on Integrated Circuits and Systems Design, Sept 2005, pp. 190195.

[69] D. DiTomaso, R. Morris, A. K. Kodi, A. Sarathy, and A. Louri, *Extending the Energy Efficiency and Performance With Channel Buffers, Crossbars, and Topology Analysis for Network-on-Chips,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 11, pp. 21412154, Nov 2013.

[70] N. Prasad, S. Chattopadhyay, and I. Chakrabarti, *ZMesh: An Energy-Efficient Network-on-Chip Topology for Constant-Geometry Algorithms,* in IEEE International Symposium on Nanoelectronic and Information Systems, Dec 2015, pp. 146151.

[71] J. Hu and R. Marculescu, *Energy- and Performance-Aware Mapping for Regular NoC Architectures,* IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems, vol. 24, no. 4, pp. 551562, April 2005.

[72] C. L. Chou, U. Y. Ogras, and R. Marculescu, *Energy- and Performance-Aware Incremental Mapping for Networks on Chip With Multiple Voltage Levels,* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 10, pp. 1866 1879, Oct 2008.

[73] J. Zhan, N. Stoimenov, J. Ouyang, L. Thiele, V. Narayanan, and Y. Xie, *Designing Energy-Efficient NoC for Real-Time Embedded Systems through Slack Optimization,* in 50th ACM/EDAC/IEEE Design Automation Conference (DAC), May 2013, pp. 16.

[74] L. Yang, W. Liu, W. Jiang, M. Li, J. Yi, and E. H. M. Sha, *Application Mapping and Scheduling for Network-on-Chip-Based Multiprocessor System-on-Chip With Fine-Grain Communication Optimization,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 10, pp. 30273040, Oct 2016.

[75] E. L. d. S. Carvalho, N. L. V. Calazans, and F. G. Moraes, *Dynamic Task Mapping for MPSoCs,* IEEE Design Test of Computers, vol. 27, no. 5, pp. 2635, Sept 2010.

[76] C. L. Chou and R. Marculescu, *Run-Time Task Allocation Considering User behaviour in Embedded Multiprocessor Networks-on- Chip,* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 1, pp. 7891, Jan 2010.

[77] S. Usman, S. U. Khan, and S. Khan, *A comparative study of voltage/frequency scaling in NoC,* in Electro/Information Technology (EIT), 2013 IEEE International Conference on, 2013, pp. 15.

[78] Y.-H. Lee, K. P. Reddy, and C. M. Krishna, *Scheduling techniques for reducing leakage power in hard real-time systems,* in Real-Time Systems, 2003. Proceedings. 15th Euromicro Conference on, 2003, pp. 105112.

[79] A. K. Djahromi, A. M. Eltawil, and F. J. Kurdahi, *Fault Tolerant Approaches Targeting Ultra Low Power Communications System Design,* in 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring, 2007, pp. 26002604.

[80] L. Guang, E. Nigussie, and H. Tenhunen, *System-level Exploration of Run-time Clusterization for Energy-efficient On-chip Communication,* in Proceedings of the 2Nd International Workshop on Network on Chip Architectures, New York, NY, USA, 2009, pp. 6368.

[81] S. Reda and A. Nowroz. Power Modeling and Characterization of Computing Devices: A Survey. *Foundations and Trends in Electronic Design Automation*, 6(2):121–216, 2012.

[82] R. Cochran, A. N. Nowroz, and S. Reda. *Post-silicon power characterization using thermal infrared emissions*, In Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design, pages 331336. ACM, 2010.

[83] J. Ou. Rapid Energy Estimation for Hardware-Software Codesign Using FPGAs. *EURASIP Journal on Embedded Systems*, 2006:1–11, 2006.

[84] P. Albicocco, D. Papini, and A. Nannarelli. Direct Measurement of Power Dissipated by Monte Carlo Simulations on CPU and FPGA Platforms. Report 2012-18, Technical University of Denmark, 2012.

[85] R. Jevtic and C. Carreras. Power measurement methodology for fpga devices. *IEEE Transactions on Instrumentation and Measurement*, 60(1):237–247, Jan 2011.

[86] J. P. Oliver and E. Boemo. Power estimations vs. power measurements in Cyclone III devices. In *Programmable Logic (SPL), 2011 VII Southern Conference on*, pages 87–90. IEEE, 2011.

[87] E. Sotiriou-Xanthopoulos, G. S. P. Delicia, P. Figuli, K. Siozios, G. Economakos, and J. Becker. *A power estimation technique for cycle-accurate higher-abstraction systemc-based cpu models.* In Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), 2015 International Conference on, pages 70–77, July 2015.

[88] H. G. Lee, K. Lee, Y. Choi, and N. Chang. Cycle-accurate energy measurement and characterization of FPGAs. *Analog Integrated Circuits and Signal Processing*, 42(3):239–251, 2005.

[89] M. Weiland and N. Johnson. Benchmarking for power consumption monitoring. *Computer Science - Research and Development*, 30(2):155–163, 2015.

[90] S. Schreiner, K. Grüttner, S. Rosinger, and W. Nebel. Ein verfahren zur bestimmung eines powermodells von xilinx microblaze mpsocs zur verwendung in virtuellen plattformen. In *18. Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV 2015)*, 03 2015.

[91] M. Hosseinabady and J. L. Nunez-Yanez. Run-time power gating in hybrid ARM-FPGA devices. In *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–6. IEEE, 2014.

[92] A. F. Beldachi and J. L. Nunez-Yanez. Accurate Power control and monitoring in ZYNQ boards. In *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–4. IEEE, 2014.

[93] M. Hosseinabady and J. L. Nunez-Yanez. Energy optimization of FPGA-based stream-oriented computing with power gating. In *2015 25th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–6. IEEE, 2015.

[94] X. Inc. Zynq-7000 AP SoC Low Power Techniques part 2 - Measuring ZC702 Power using TI Fusion Power Designer Tech Tip. http://www.wiki.xilinx.com/Zynq-7000+AP+SoC+Low+Power+Techniques+part+2+-+Measuring+ZC702+Power+using+TI+Fusion+Power+Designer+Tech+Ti(30.04.2015), Januar 2014. Version 0.1.

[95] Datasheet: Ucd9248 - digital pwm system controller. Technical Report SLVSA33A, Texas Instruments, Jan 2010.

[96] Christof Schlaak, Maher Fakih and Ralf Stemmer *Power and Execution Time Measurement Methodology for SDF Applications on FPGA-based MPSoCs.* CoRR, abs/1701.03709, 2017.

[97] G. Reehal and M. Ismail, *A Systematic Design Methodology for Low-Power NoCs,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 12, pp. 25852595, Dec 2014.

[98] N. Banerjee, P. Vellanki, and K. S. Chatha, *A power and performance model for network-on-chip architectures,* in Design, Automation and Test in Europe Conference and Exhibition, vol. 2, Feb 2004, pp. 12501255 Vol.2.

[99] L. Benini, A. Bogliolo, and G. D. Micheli, *A Survey of Design Techniques for System-Level Dynamic Power Management,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 3, pp. 299316, June 2000.

[100] T. Simunic, S. P. Boyd, and P. Glynn, *Managing Power Consumption in Networks on Chips,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 1, pp. 96107, Jan 2004.

[101] X. Wang, B. Zhao, T. Mak, M. Yang, Y. Jiang, and M. Daneshtalab, *On Fine-Grained Runtime Power Budgeting for Networks-on-Chip Systems,* IEEE Transactions on Computers, vol. 65, no. 9, pp. 2780 2793, Sept 2016.

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

[103] 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.

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

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

[106] E. Boemo, S. Lpez-Buedo, *Thermal Monitoring on FPGAs Using Ring-Oscillators,* , FPL 1997.

[107] 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.

[108] 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.

[109] Directorate, I. A. *Protection profile for separation kernels in environments requiring high robustness,*Technical report, US Government, 2007.

[110] Skentzos, Paul. *Software safety and security best practices: A case study from aerospace,* NDIA ground vehicle systems engineering and technology symposium, Michigan, August 2014.

[111] Green Hills Software, Inc, (2008) *INTEGRITY-178B separation kenel security target version 4.2,* ed https://www.commoncriteriaportal.org/files/epfiles/st_vid10362-st.pdf

[112] J. Frid, *Security Critical Systems in Software,* ed, 2010, p. 57.

[113] J.-P. Kaps, *Cryptography for ultra-low power devices,* Worcester Polytechnic Institute, 2006.

[114] S. Kerckhof, F. Durvaux, C. Hocquet, D. Bol, and F.-X. Standaert, *Towards Green Cryptography: A Comparison of Lightweight Ciphers from the Energy Viewpoint,* in Cryptographic Hardware and Embedded Systems CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings, E. Prouff and P. Schaumont, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 390-407.

[115] K. Baddam and M. Zwolinski, *Evaluation of dynamic voltage and frequency scaling as a differential power analysis countermeasure,,* in VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on, 2007, pp. 854-862.

[116] 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.

[117] 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.

[118] 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.