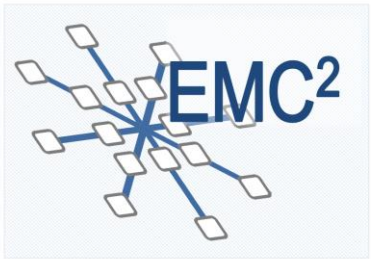


Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments

Project Acronym:

EMC²

Grant agreement no: 621429

Deliverable no. and title	**D4.5 – Final report on Cores, Memory, Virtualisation, Accelerators, Interconnecting Cores, Chips, and Peripheral devices in Mixed-Criticality Systems: Updates from the Second Innovation Cycle**	
Work package	WP4	SP_Multi-core Hardware Architectures and Concepts
Task / Use Case	T4.2, T4.3	Advanced processor concepts and architecture definition Core and chip interconnect, peripheral devices
Subtasks involved	ST4.2.1, ST4.2.2, ST4.2.3, ST4.2.4 ST4.3.1, ST4.3.2, ST4.3.3	
Lead contractor	Infineon Technologies AG Dr. Werner Weber, mailto:werner.weber@infineon.com	
Deliverable responsible	02E TUW Haris Isakovic, haris@vmars.tuwien.ac.at	
Version number	v1.0	
Date	01/04/2017	
Status	Final	
Dissemination level	PU	

Authors

Partici-pant no.	Part. short name	Author name	Chapter(s)
01A	IFAG	Wolfgang Schneidt Jens Harnisch	3.13,4.7 3.5
01R	NXPGE	Michael Philipp Kiran Shekhar	3.8 4.6
01V	TUBS	Rolf Mayer	3.3
01W	TUDO	Lilian Tadros	3.7, 4.8
02A	AVL	Peter Priller	4.4
02D	TTT	Andres Eckel	3.4,4.10
02E	TUW	Haris Isakovic	4.2
04D	UTIA	Jiri Kadlec Zdenek Pohl	4.3
02C	IFAT	Alexander Lipautz Norbert Druml	3.2.2 3.2.1
11D	Leonardo (Selex) Polito	Massimo Traversone Stefano Esposito	3.11
15E	Tecnalia	Alonso Muñoz, Asier	4.9
15O	SevenS	José Luis Gutiérrez	4.5
16A	Chalmers	Ioannis Surdis	3.10
17B	Imec-NL	Xin Fan	3.9
17C	NXPNL	Dominique Defossez	3.12
18H	UoBr	Steve Kerrison	3.1
15F	TASE	Manuel Sanchez Renedo	3.6

Document History

Version	Date	Author name	Reason
0.1	15.02.2017	Haris Isakovic	Draft, initial integration of contributions
0.2	01.03.2017	Haris Isakovic	Draft, additional contributions integrated
0.3	15.03.2017	Haris Isakovic	Draft, released for review
0.4	23.03.2017	Rolf Mayer Haris Isakovic	Review Comments Integration
0.6	24.03.2017	Haris Isakovic	Final revision call.
0.7	27.03.2017	Haris Isakovic	Final draft revision.
0.8	28.03.2017	Haris Isakovic	Final draft integration.
1.0	29.03.2017	Haris Isakovic	Final version
	01.04.2017	Alfred Hoess	Final editing and formatting, deliverable submission

Table of *Contents*

List of figures

List of tables

1. Introduction

1.1 Objective and scope of the document

The document provides an overview of the technical work performed by the partners in WP4, specifically tasks T4.2 and T4.3. It includes a short description of a motivation behind the activity of each partner, a preliminary specification or design, planned development phases, and expected results of each individual contribution.

1.2 Structure of the deliverable report

The document is organized according to the structure of the work package and the corresponding tasks. First section provides introduction in the document. In the second section a report from the task T4.2 grouped by subtasks. Next section provides the report from the task T4.3 grouped by subtasks.

2. Publishable Executive Summary

This document is the final report on the activities performed by the partners of the WP4, ARTEMIS project EMC2, involved in tasks T4.2 and T4.3 during the final reporting period. It includes technical topics of multi-core architectures, memory hierarchies, virtualization technologies, on- and off-chip interconnects, dynamic reconfiguration in hardware, peripheral devices, accelerators, chip design and energy consumption. It is organized by task and activity of each individual partner. It provides a progress of the partners in the final reporting period, and the outcome of individual contributions with regard to partner goals and overall project objectives.

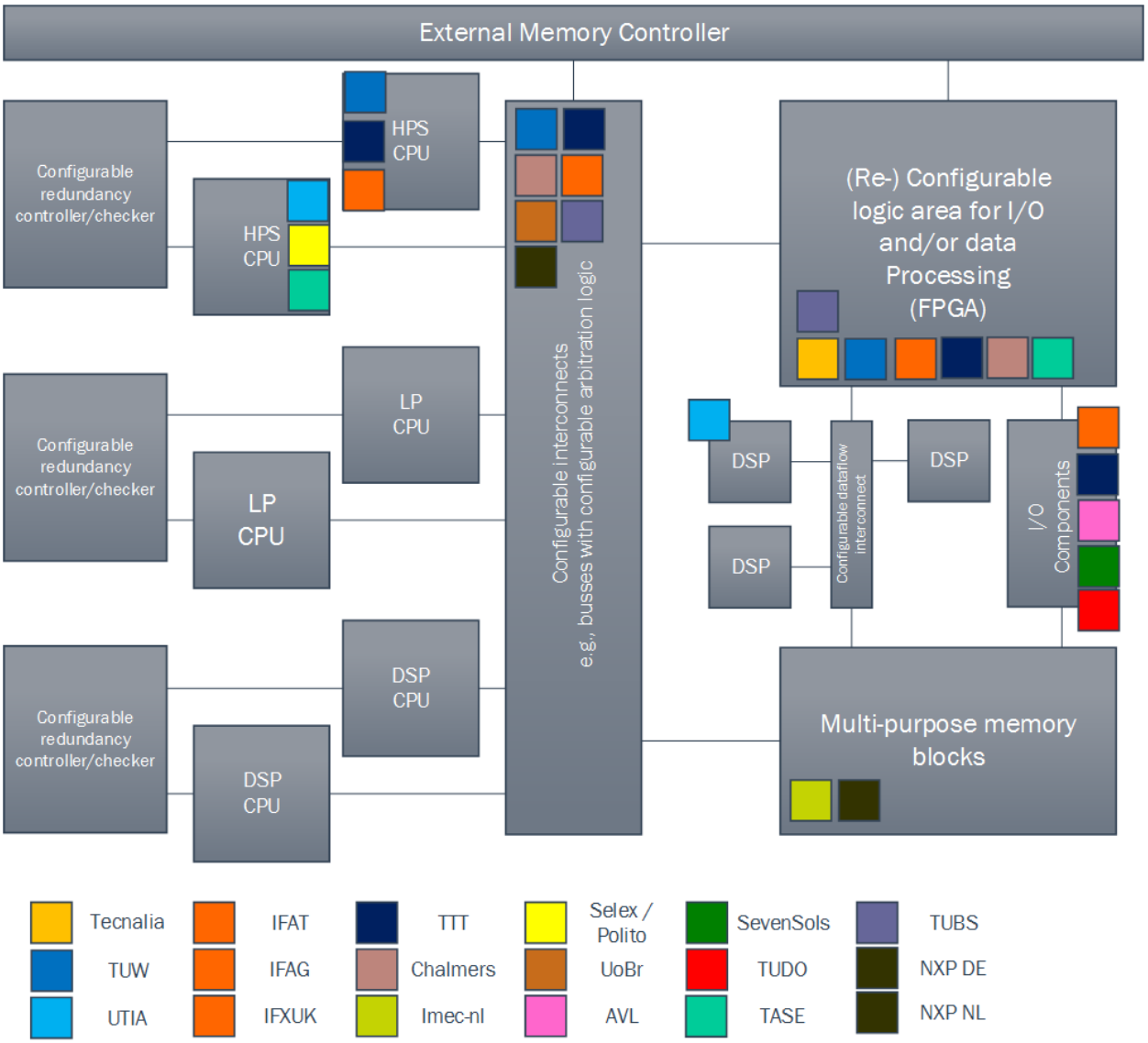


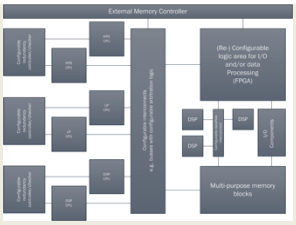
Figure 1 A generic hardware architecture and a partner activity by sector

3. T4.2 Advanced processor concepts and architecture definition

3.1 University of Bristol (18H UoBR)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



In this deliverable UoBR contributes further advances in MCENoC, its non-blocking NoC for EMC2 systems, as part of ST4.3.1. The main contributions are:

- Evaluation effort for benchmarking and comparison of MCENoC to other network approaches.
- Integration of a prototype debugging interface.

This progress is in addition to the results reported in D4.4 and D4.17. Following D4.17, a focus on evaluation and analysis was deemed the most effective way of motivating further use and exploration of the MCENoC.

3.1.1 MCENoC Simulation and benchmarking

In [1], the need to perform full system assessment was proposed as future work, including integration of MCENoC with CPUs, and then benchmarking. To de-couple this task, CPU integration, reported in D4.17, and benchmarking, reported here, were handled separately. This was enabled through the use of simulation tools based on Netrace [2], used on the PARSEC benchmark suite with 64 processor cores. The MCENoC is compared to a network with zero-delay and a simple mesh.

Initial results indicate that, depending on length of routing phases used for the MCENoC, it performs simliar to the mesh network. For example, in a "Blackscholes" benchmark, the mesh network is 0.73% slower than a zero-delay network, and the MCENoC's Benes structure is between 0.54% and 2.00% slower, depending on configuration. MCENoC's additional predictability adds technical value without significant negative performance costs.

A future opportunity for comparison exists by integrating the MCENoC with project partner KTH's NoC System Generator tool. Following discussion, an up-coming release of the NoC System Generator will create a path for adding the MCENoC as a custom network structure, then enabling further design space exploration.

Debug

In D4.17, integration with RISC V cores was presented. Debug support for RISC V varies between implementations, and development by the community is ongoing. In-line with this, UoBR sought to provide additional debug information from the MCENoC, collaborating with UltraSoC to create a prototype.

UltraSoC, an organisation external to the EMC2 project, provides IP to SoC designers that delivers richer information on the internal operation of hardware. It can do so in real-time without interfering with the device's operation. In the real-time sensitive context of EMC2, this is therefore a compelling integration.

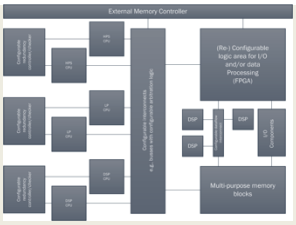
In the prototype, UoBR developed an interface to a series of performance counters in the MCENoC's network interfaces, which can then be read and streamed out of the system by UltraSoC's IP. The counters provide information such as number of connections formed, bytes transferred and the number of error conditions generated. Through this integration, the data can be observed without modifying applications

running on the system, thus not impacting their behaviour or performance. The prototype has been tested successfully in simulation.

3.2 Infineon Austria AG (02C IFAT),

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



3.2.1 Hardware/Software Multi-Core Architecture for ToF 3D Imaging

Time-of-Flight (ToF) is a 3D imaging technology that provides distance information by measuring the travel time of the emitted and reflected light. However, Time-of-Flight cameras provide only raw sensor data which needs to be processed in order to obtain distance information of the environment.

Given this initial situation, the aim of IFAT during the WP4 activities was to realize innovative hardware/software co-designs in order to achieve hardware-accelerated ToF processing. The achieved results of IFAT's WP4 activities (heterogeneous core implementations, system architecture, memory and area requirements, computation performance, demonstrators, etc.) are detailed in deliverable D4.22.

3.2.2 AMSPS-Systems for MCMC integrated Systems

Note: This activity was performed in cooperation with IFAG

The rapidly increasing computing power in many applications and the further trend of integration of many system functions leads to new and demanding challenges for micro-controller development of various application fields (e.g. for automotive or industrial subsystems). These subsystems (e.g. motor management of vehicles) combine sensor and actuator functions with high compute power and programmability.

Increasing compute power goes hand in hand with an increasing requirement set for accompanying power supply, measurement, and support functions for the future multi-core architectures.

Furthermore, the complexity of new integrated microcontroller systems with embedded multiple cores require new concepts of integrated power supply management architectures, new demanding concepts for measurement functions (e.g. ADCs,…), and other support functions (e.g. PLL, high speed interfaces,…).

Technical requirements of power supply architectures of integrated multi-core systems shall offer a variable input voltage range of different applications. Furthermore, multiple input supply concepts shall be possible to offer high application flexibility.

A Power Management System must support any possible start-up condition as long as maximum ratings in terms of input voltage and overload condition at the input pins over the ambient temperature range are not violated.

Power supply levels and voltage monitoring accuracy shall fulfill the requirements for steady state accuracy to ensure the safe function of multi-core systems. Additionally, dynamic functionality for safe function of multi-core systems should be offered accordingly.

Voltage monitoring accuracy is offered to ensure a safety compliant function of multi core systems.

The Power Management scheme allows activation of power down modes so that the system operates with the minimum required power for the corresponding application state.

High efficiency is required within the power management concepts for various MCMC system modes. Maximum output current of voltage regulator should support the need of multi core systems to ensure full functionality without performance reduction within the systems.

Investigations on functions to reduce the static leakage current and potentially new derived concepts are needed to keep the system performance high. Investigation on new solutions for reduced quiescent current of a multi - core system is needed (e.g. in Standby Mode).

The AMSPS architecture has been designed to fulfill all the above requirements. The figure below shows: a high level block diagram of the AMSPS-System. It shows the main sub blocks the power management is consisting of.

- EVR_ANA: analog subsystem (e.g. Bandgap, ADC's, oscillators…)
- EVR_DIG: digital control of the Voltage-Regulators and interface to peripherals
- RBB DUT: Mixed Signal IP targeting to reduce static current consumption (e.g. channel leakage) of a digital core area.

Covering both, performance and Functional Safety (ISO26262) requirements were in the Scope of the AMSPS project. Hence, the following subtasks were considered:

- Definition and refinement of functional requirements.
- Definition and refinement of Safety requirements.
- Definition of AMSPS Concepts for integrated MCMC systems.
- Definition of test chip Concepts.
- Implementation of test chips for AMSPS functionality.
- Definition of application use-cases.
- Verification and Validation of test chips.
- Final documentation.

Generally spoken, the major challenge is to make the AMSPS sufficiently robust for industrial- & automotive applications (e.g. performance, efficiency, and life-time) under consideration of functional safety requirements (e.g.Technical Safety Requirements [TSR] and Safety Mechanism [SM]).

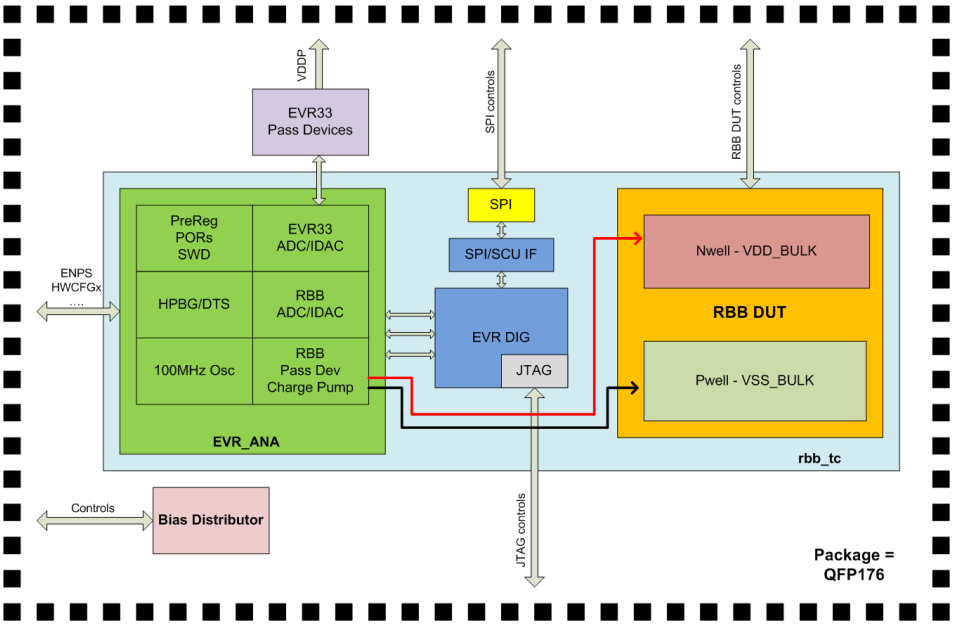


Figure 2 AMSPS-System block diagram

Broadly, the following tasks were planned and completed:

- Electrical parameter characterization of AMSPS (e.g. accuracy, power consumption…).
- AMSPS Robustness Validation (e.g. Current- & Noise-Injection tests, Supply Spikes…).
- Working out a methodology for functional safety pre-silicon verification & post-silicon lab-validation
- Use-case definition with partners in WP4 (e.g. Load profile, Power Sequencing, Power Modi & Mode transitions)
- The demonstration of the technology
    - WP Demonstrator: Test Chip of AMSPS in 40nm CMOS technology,
    - LL demonstrator: AVL demonstrator use cases,
    - Simulation: Mixed Signal Verification Setup (Regression),

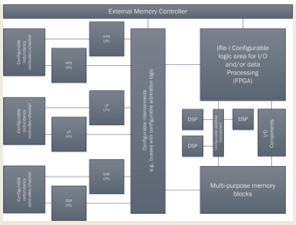
Final Achievement:

- AMSPS Concepts for integrated MCMC systems defined and verified on test chips.
- AMSPS Concepts for integrated MCMC systems with safety requirements implemented to secure ISO26262 compliance.
- AMSPS Concepts for integrated MCMC systems with safety requirements implemented to secure ISO26262 compliance defined and pre-verification of new challenges of the next generation multi core systems.

3.3 TU Braunschweig (01V TUBS)

Technology lanes covered:

- *Heterogeneous Multiprocessor SoC architectures*
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- ***Networking***
- ***Virtualization and verification technologies***



3.3.1 Enhanced Run-Time Monitoring for Mixed-Criticality

On a multi- or many-core platform that runs applications of different safety criticality (mixed-criticality), all applications have to be certified to the highest level of criticality, unless they are sufficiently isolated. Isolation enables individual certification of applications and cost-efficient re-certification of single applications after an update. We introduced a parameterizable and synthesizable many-core platform with a fast and scalable monitoring and control mechanism that supports safe sharing of resources.

With the IDAMC we introduced a parameterizable and synthesizable many-core platform that allows the evaluation of concepts and mechanisms for running multiple mixed-critical applications on a single platform. But for further development, we need the design on a higher abstraction to enable faster evaluation of new parameters and monitoring and control capabilities. So we re-implement the design in SystemC based on the SoCRocket platform and extend it with an interface for easy addition of such.

Development phases:

- Step 1: M12 (end of March 2015)
  - ✓ A concept defined
  - ✓ Initial SystemC NoC Model
  - ✓ The base architecture defined
- Step 2: M24 (end of March 2016)
  - ✓ Tested NoC Model
  - ✓ Transparent Address remapping and IRQ rerouting
  - ✓ Initial network interface with parameters and capabilities
  - ✓ Initial prototypical integration of EMC2 results demonstrated
- Step 3: M36 (end of March 2017)
  - ✓ Detailed performance evaluation exists
  - ✓ Final EMC2 demonstrator available and accessible to whole consortium
  - ✓ Demonstrates innovations and input from EMC2

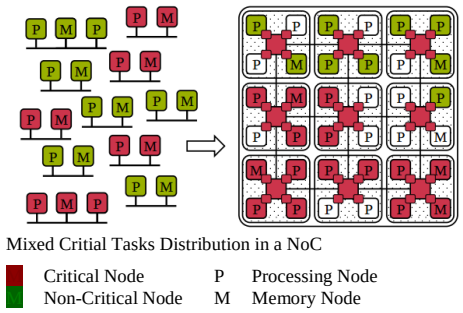


Figure 3 Mixed Critical Tasks Distribution in NoC

Contributing to high-level requirements as defined in Requirements_Document_D0401. Overview defined in D4.2 Specification of Demonstrator Platforms (chapter "Communication and verification of mixed-criticality MC control units"). Additional resources can be found in [3] [4] [5].

Progress

Until now the design and implementation of the models proceeds without delays. The progress is therefore on time as planed in D4.3. In addition, interrupt routing and transparent address remapping is implemented in the NoC and network interface to serve the extended needs of the component demonstration.

Results

Our tests and experiments show that manual routed on careful selected routes with alternate routes and nodes in mind can produce a stable mixed-critical system with the needed behavior. We think it might be possible to use the SystemC models to calculate these positions, routes and fallback routes while in simulation with some additional input.

One of our test configurations is a Symmetric 2D-Mesh with one packet generator/receiver per router which produces continuous all-to-all traffic and is logging of flit latencies. It is implemented in vative RTL and TLM for speed measurement. The goal of this configuration is to create a high network load to maximize simulation effort and high contention based latency to identify deviation in behavior and timing.

The fixed 8x8 network configuration shows a >160x speedup between RTL and the contention aware TL-Model and 1400x - 2000x speedup between RTL and the contention unaware TL-Model by a simulation uncertainty of less than 10%. The variable size network configuration shows the TL-Models allow feasible simulations of significantly larger networks and single threaded nature of the SystemC simulation kernel becomes a bottleneck which larger networks (50x50 contention aware routers require 2500 threads).

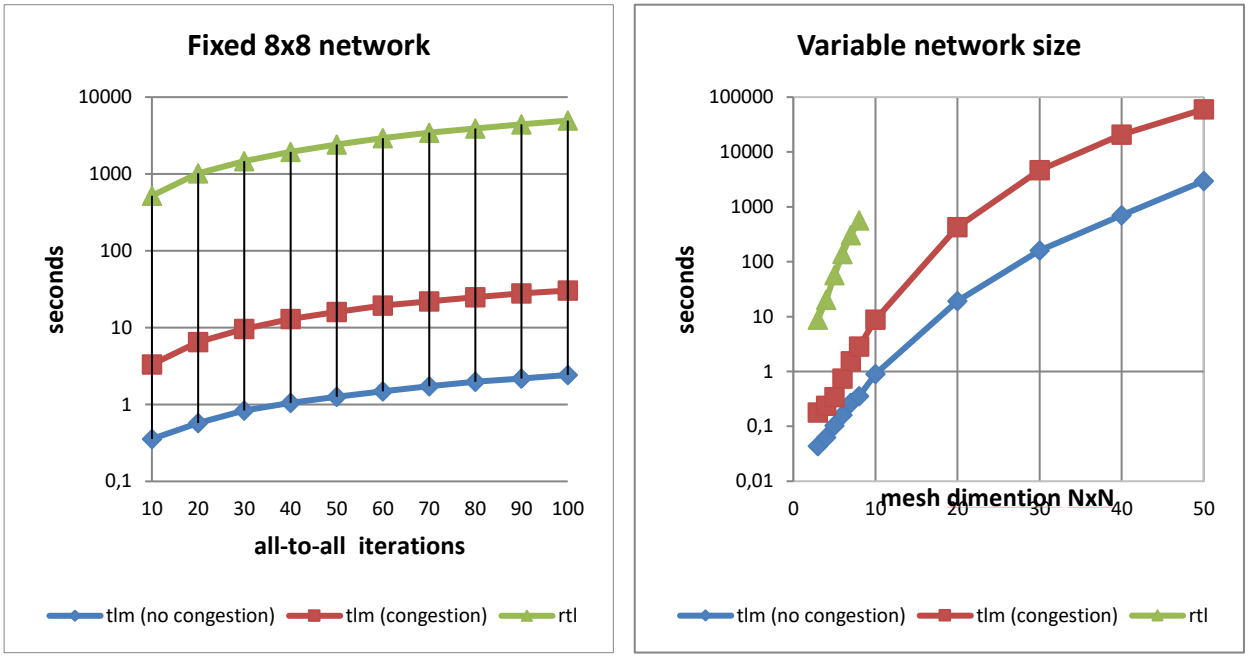


Figure 4 Results in fixed and variable network size

Outcome

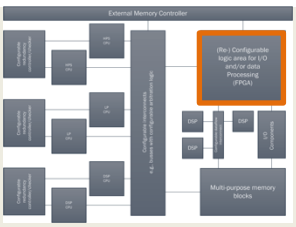
We show the effects of sharing resources by applications of different criticality on the response time of a highly critical task. With our monitoring and control mechanism, we develop a fast and scalable solution to isolate the timing of mixed-critical applications to reduce their cost for certification and re-certification.

We demonstrate the effectiveness of the mechanism to isolate a highly critical task against a (faulty) low criticality task. The NoC and NI is fully integrated in the emc2 SoCRocket Platform delivered in D4.17.

3.4 TTTech Computertechnik AG (02D TTT)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architecture***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- ***Networking***
- *Virtualization and verification technologies*



3.4.1 Portability of the ACROSS MPSoC architecture to a Xilinx Zynq architecture

The ACROSS architecture was implemented on an ALTERA Stratix IV FPGA development kit. It implemented eight ALTERA NIOS cores allocated around a configuration called “fragment switch”. Each switch had four ports whereof two were used to connect to the appropriate neighboring switch in the fragment switch configuration. The remaining two ports were used to connect to one NIOS core each. Between the physical connection of the fragment switch and the NIOS core a so-called “trusted interface sub-system (TISS) was inserted. The TISS assured strict partitioning between the individual cores (Figure 5). Peripherals such as interfaces, gateways or memory were connected to each of the cores. The cores were used for management tasks and service tasks like the Trusted Resource Manager (TRM), the I/O module,

the storage unit or the diagnostic unit. Four cores were reserved for applications. The idea was that no shared resources would be available that are not strictly controlled by a TISS. All peripherals are connected to one core only. In case a neighboring core would require to use a peripheral from another core, the only way to access it would be via the core "owning" the peripheral. Thus the peripherals are under control of the "owning" core and due to the protection of the TISS and the TRM, problems of collisions or untended use at the same time or writing/reading from a memory at the same position or at the same time are expected to be "locked out" due to the strict partitioning of the ACROSS MPSoC architecture.

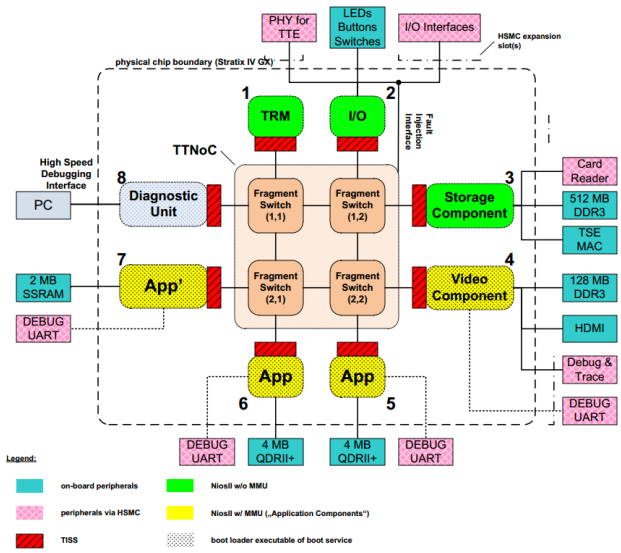


Figure 5 ACROSS architecture implemented on an ALTERA FPGA

A ZYNQ platform in principle allows composing a very similar architecture but using much more powerful cores since the ZYNQ platform uses two ARM Cortex 9 cores. The ZYNQ platform also provides some FPGA space for individual designs to be connected to the core. However, the programming space on the ZYNQ platform is not large enough to host another seven ARM cores plus the TISSes and the NoC.

We base the considerations on the Xilinx Zynq-7000 AP SoC ZC706 Evaluation Kit with Zynq-7000 XC7Z045 FFG900 – 2.

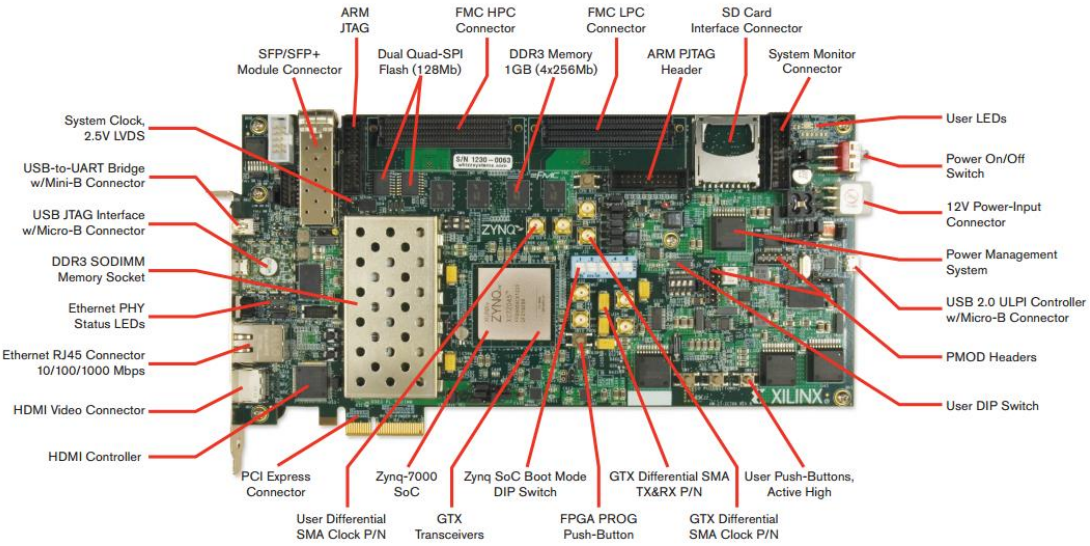


Figure 6 Xilinx ZYNQ ZC706 evaluation board with ZYNQ 7000 SoC [6]

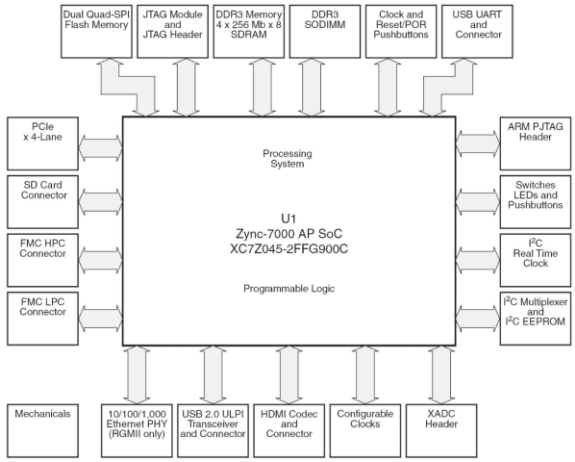


Figure 7 ZC706 Evaluation Board Block Diagram [6]

Information on the evaluation board architecture and components is provided in Figure 6, Figure 7, and Figure 8. The programmable logic on the evaluation board is connected to the ARM core via an AMBER® interconnect.

The idea was to see one of the Zynq-7000 XC7Z045 AP SoCs (Zynq SoC) as one quarter of an ACROSS architecture. This would contain one switch of the fragment switch module on the Zynq SoC programmable logic (PL). In addition the TISSes for the two on chip ARM cores would have to be implemented on the PL.

The challenge now is how to interconnect the switches since there is no Ethernet connection provided directly and potentially the AMBER interconnect would have to be used instead. Thus it is not clear if the same functionality as it has been implemented in the ACROSS MPSoC would be possible with such approach. In addition, it will be difficult to use four of the Zynq SoCs in order to have an identical structure implemented by the use of the Zynq SoC as a central design brick.

Concluding it needs to be stated that compared to the solution using the ALTERA ARRIA system as it is described in Section 4.2, the approach using the Zynq platform will raise a number of problems and challenges to overcome. One is the difficulty that the connection between the fragment switch elements is different to the TTEthernet compliant connection used in the ACROSS approach. The other is price, one development kit of the Zynq platform is traded at approax. €2.500,-. To obtain eight cores one would need four of these boards in order to implement a demonstrator using a Zynq SoC based approach. Even if we would use less cores due to their significantly higher performance, this might cause problems in isolating the applications if they would have to be on one and the same core. We anticipate that a solution integrating a hypervisor that allows virtualization would improve the approach.

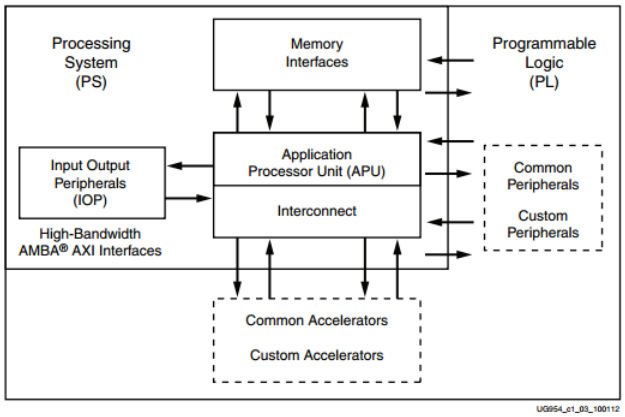
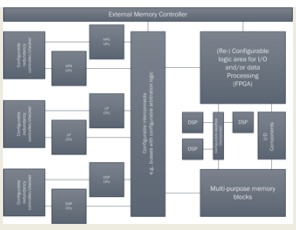


Figure 8 High level block diagram [6]

3.5 Infineon Technologies AG (01A IFAG)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



3.5.1 Tracing Mixed-Criticality

Infineon successfully continued with its activities developing dedicated approaches for tracing on multi-core microcontrollers. The internal development platform, called ChipCoach, was enhanced by several

features, and is meanwhile used by several field application engineers, to ensure the correctness of customer applications, also for mixed criticality applications.

In the previous reporting period Infineon had focused on interference in space, meaning to analyze which core and function does access which memory, what is the execution context (which ASIL level is expected), and which ASIL level the corresponding software qualified for.

In the current reporting period Infineon focused onto laying the foundations for tracing of interrupt behavior in mixed criticality applications. Time driven approaches, for example checking a sensor value in regular time intervals, facilitate easier software design and seem more and more accepted. However, event driven approaches, eventually often implemented with the help of interrupts, are still often applied, in particular for applications operating close to the performance limit of the microcontroller, such as engine control. Consequentially a systematic approach for analyzing the interrupt behavior, in particular for mixed criticality applications, is important.

Mixed criticality in event driven software is usually expressed by assigning different priority levels to the interrupts which need to be handled by a service provider, meaning a core or a DMA controller. A higher level interrupt can pre-empt a lower level interrupt. Still, this pre-emption takes time, impacts the behavior of the lower level interrupt, and if multiple interrupts arrive at the same time, arbitration is needed, again impacting the timing behavior. Hence it is important to differentiate at least between the two following timings:

- Arriving time: An interrupt arrives at a certain point in time from an interrupt generator (e.g. a communication peripheral) at the interrupt system. After the arriving time the arbitration can start and forwarding to the interrupt service provider.
- Response time: At a certain point in time the interrupt service routine is started at the interrupt service provider.

Infineon implemented tracing of both times in its internally developed tool ChipCoach, for multiple interrupt sources and service providers in parallel. Furthermore statistics, e.g. the average time needed to start processing an interrupt, is shown to the user. Next step is to point out automatically potential anomalies with the help of the ChipCoach tool.

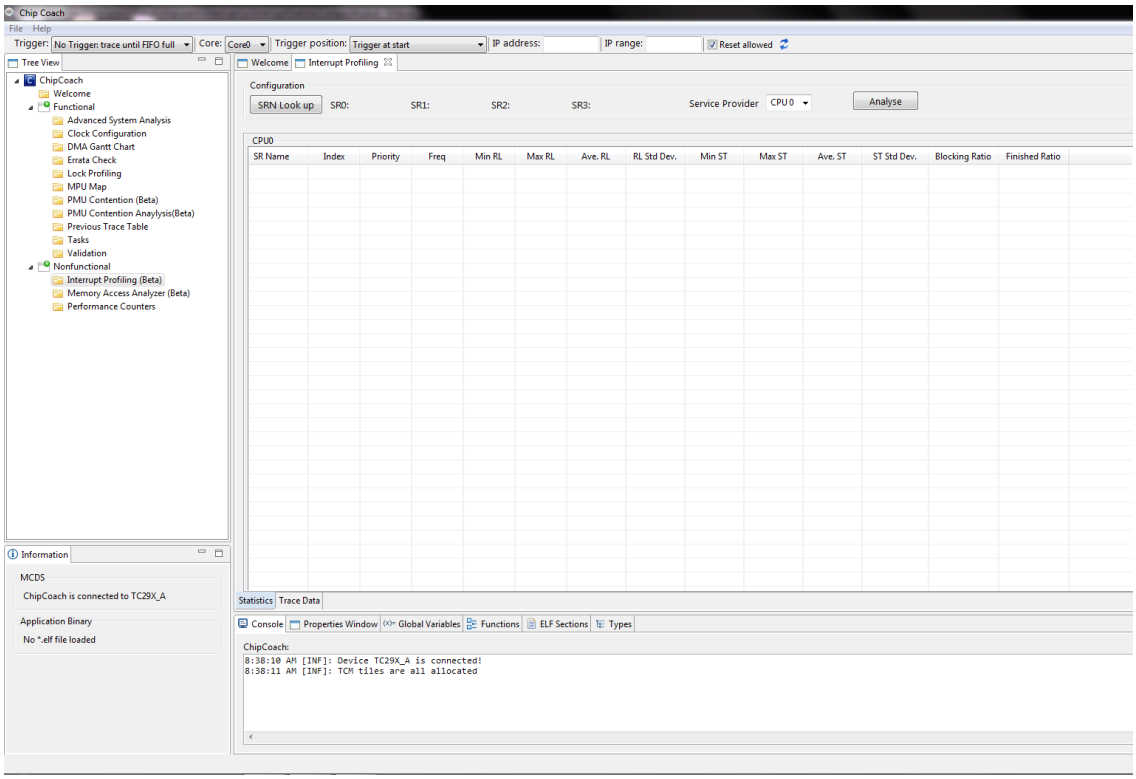


Figure 9 Template for statistical overview of interrupt behavior (work ongoing)

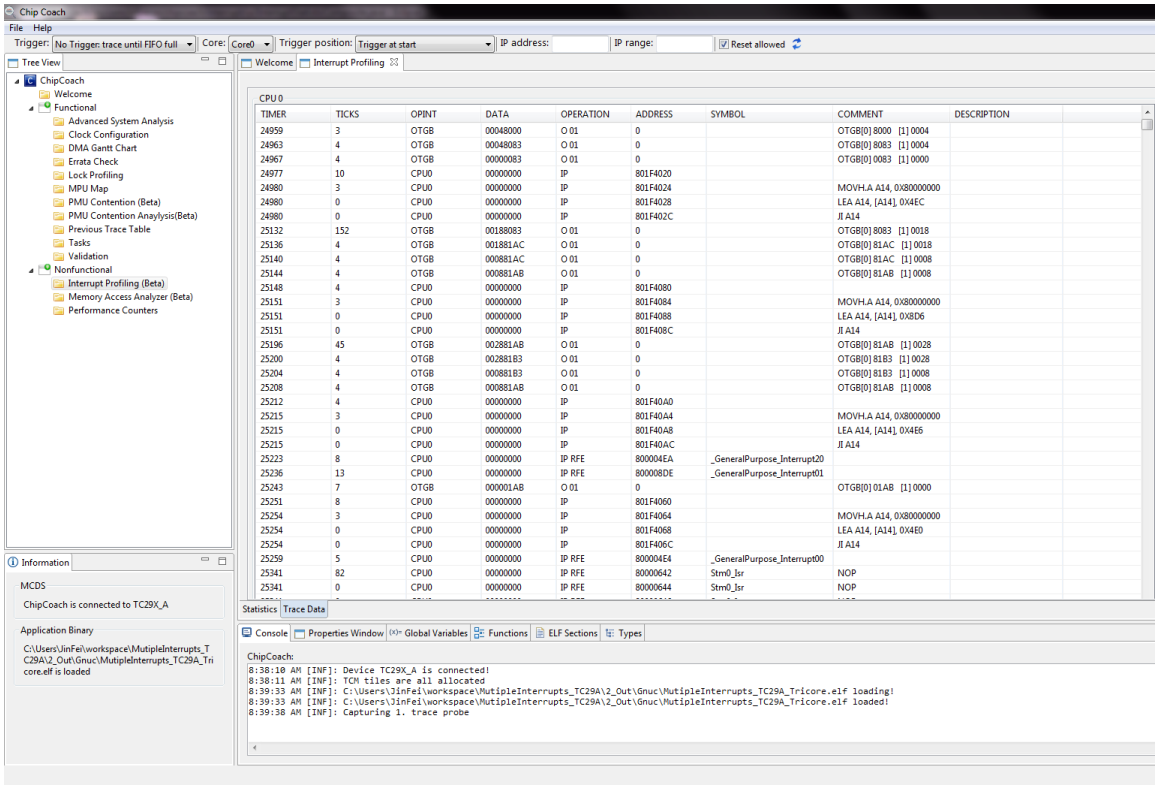


Figure 10 Several interrupts arriving almost at the same time, which may cause timing impact for mixed criticality software (work ongoing)

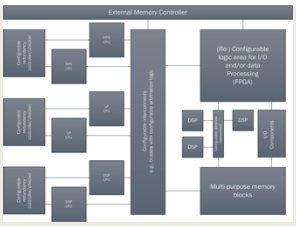
Achievements:

- Tracing of mixed criticality software (interrupt related share) implemented, very comprehensive analysis options, for example pointing out interference in time due to frequent arbitration
- Generation of statistics implemented
- Automatic finding of hot spots implemented

3.6 Thales Alenia Space Spain (15F TASE)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- *Networking*
- *Virtualization and verification technologies*



3.6.1 MPSoC hardware architecture for fault tolerance systems

LEON2-FT processor and Watchdog Timer module

The hardware system developed for the RTAX device has the required features of fault-tolerance to warranty reliability and self-healing mechanisms. Basically, it is formed by a LEON2-FT processor,

hardened with fault-tolerance characteristics which provide greater robustness, and a Watchdog Timer (WDT), an autonomous module that monitors continuously the status of scrubbing processes in Virtex-5, acting accordingly.

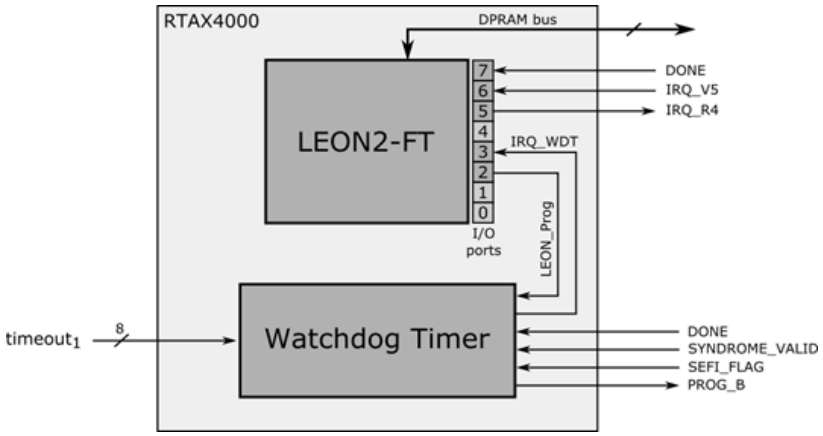


Figure 11 Modules implemented in RTAX4000 device

As a result of the scrubbing performed in Virtex-5 some signals are asserted: SYNDROME_VALID from FRAME_ECC_VIRTEX5 primitive, which reports integrity of configuration frames during Readback, and SEFI_FLAG from MicroBlaze that informs when SEFIs are detected. A brief description of signals used in this system is shown in Table 1.

Table 1: Signals interfaces in RTAX4000 device

Signal	Interface	Description
DPRAM bus	RTAX-V5	Control and data signals to manage DPRAM memory: clk, address[11:0], data[15:0], chip select (cs), write enable (we), output enable (oe)
IRQ_V5	RTAX-V5	Interrupt sent from Virtex-5. It notifies that LEON processor can access to DPRAM memory to process new data
IRQ_V4	RTAX-V5	Interrupt sent from LEON. It notifies that MicroBlaze can access to DPRAM memory to process new data
DONE	RTAX-V5	Configuration status of Virtex-5
PROG_B	RTAX-V5	Programming signal for Virtex-5
SYNDROME_VALID	RTAX-V5	Active one cycle for each uncorrupted frame during a readback operation (signal associated with FRAME_ECC_VIRTEX5 primitive)
SEFI_FLAG	RTAX-V5	SEFI indicator sent by MicroBlaze
IRQ_WDT	LEON-WDT	Interrupt sent from WDT to LEON warning that a reconfiguration of Virtex-5 has been done
LEON_Prog	LEON_WDT	Direct order from LEON processor to reprogram Virtex-5

LEON2-FT processor

Systems of each device are communicated through a Dual-Port RAM (DPRAM) memory, implemented in the Virtex-5, acting as Mailbox between them. Each one has a reserved area to avoid conflicts during memory accesses. Interrupts are used to manage a synchronize access from both sides, notifying the other device that data are ready to be processed after finishing the write process.

Additionally, there is a communication interface between LEON and Watchdog Timer which allows the processor to have direct access to reprogram Virtex-5. Some capabilities are possible through I/O ports: assert PROG_B signal for Virtex-5 device, manage interrupt sent from WDT and analyze Virtex-5 DONE signal.

Watchdog Timer

This custom module designed by TASE will check SYNDROME_VALID, SEFI_FLAG and DONE signals coming from Virtex-5 to verify the integrity of its configuration memory. In case of failure, it is in charge of total reconfiguration of Virtex-5 asserting PROG_B signal.

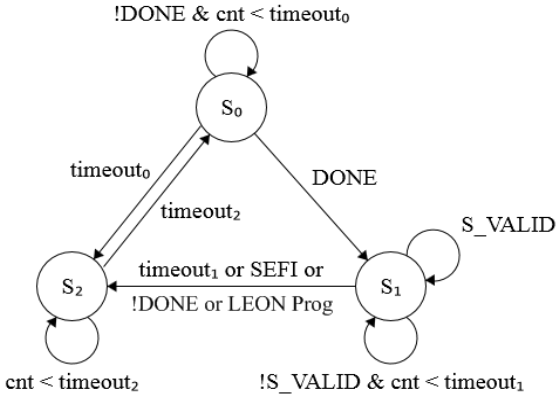


Figure 12 FSM diagram of Watchdog Timer

The implementation of the Watchdog module is based on a 3-states FSM (Figure 12) and a configurable timer in each one. After each programming of Virtex-5 (S_0), WDT waits for assertion of DONE signal with a predetermined $timeout_0$ (See Table 2). If configuration is successful, WDT passes to a normal working state (S_1) waiting for the corresponding $timeout_1$ of the periodic SYNDROME_VALID signal. If counter reaches timeouts, or SEFI_FLAG signal is asserted, or DONE signal is deasserted, or a programming order comes from LEON, WDT enters in a programming state (S_2) during a period fixed by $timeout_2$. After this time, WDT comes back to S_0 waiting for Virtex-5 gets programmed.

Table 2: Timeout's values used in each FSM state

Timeout	Time	Comments
Timeout₀	160 ms	Maximum Configuration Time for XC5VFX130T (48 MHz): 119.1 ms
Timeout₁	10 ms – 2,56 s	Externally configurable using R4_SPEC_IO[7:0] pins. For real-time systems it will be managed by LEON processor.
Timeout₂	280 ns	Minimum Program Pulse Width: 250 ns

LEON is informed by an interrupt every time WDT leaves S_0 state. Hence the processor always know when a full reconfiguration has been performed and must check the status of DONE signal to find out if the result of that process has been successful or not.

Validation design

A test design have been developed to validate previous functionalities. This design (Figure 13), which has been implemented in Virtex-5 FPGA, contains a FSM that controls a DPRAM memory and simulates the behavior of the MicroBlaze, and a Frame ECC emulator of Virtex-5.

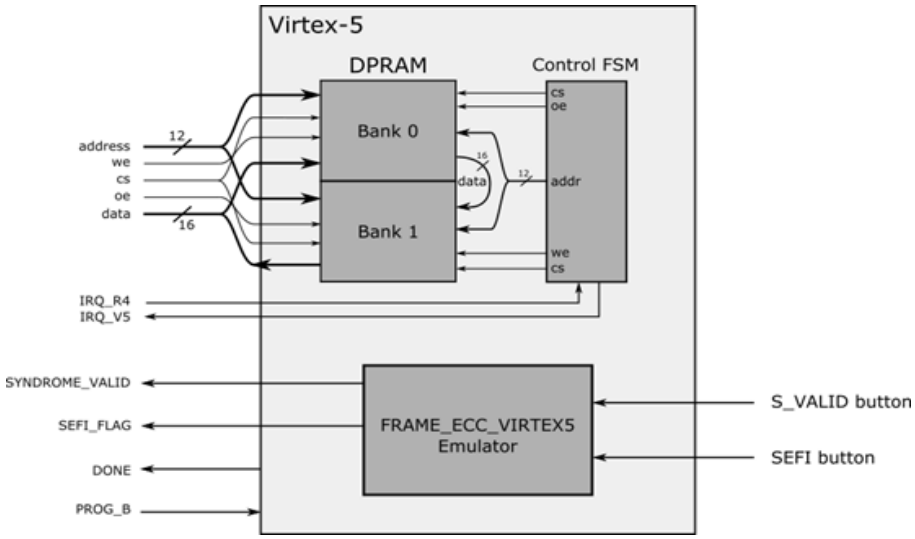


Figure 13 Modules implemented in Virtex-5 to validate the design

The goal of the first test is to validate correct access to Dual-Port RAM from RTAX device. For this purpose, two DPRAM are implemented with a control FSM. This state machine acts the same as the MicroBlaze: it attends the interrupt when LEON stops writing, then it processes the information and finally it notices LEON when have finished. In this case, to copy the content from one memory to the other one is the data processing.

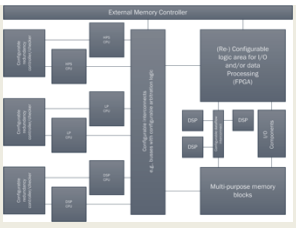
Externally, LEON will write and read from the same addresses, however the aforementioned procedure takes place inside the Virtex-5. To validate this test, LEON must read the same data it wrote previously, after sending the corresponding interrupt.

The purpose of the second test is to validate the self-healing capabilities that provides the Watchdog module. To emulate SYNDROME_VALID signal like FRAME_ECC_VIRTEX5 logic would do is the methodology used. A periodic signal with pulse width of 0.48 s. and non-active during 1.55 s. (time for *full device scrubbing*, the worst case) has been chosen to make easier its visualization. Furthermore, it is possible to handle SYNDROME_VALID and SEFI_FLAG signals using the push buttons of the platform. SYNDROME_VALID is non-active while pressing PUSH2 button, and PUSH3 emulates a SEFI in the system.

3.7 TU Dortmund (01W TUDO)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



3.7.1 **Deterministic and Efficient Memory Hierarchies for Mixed-criticality Real-Time Systems**

SoCRocket is a SystemC/TLM model of the Leon3-based GRLIB multi-core platform provided by the Technical University of Braunschweig (TUBS). Throughout Y1 and Y2, we have been working on augmenting SoCRocket with support for mixed-critical use-cases. In collaboration with TUBS and OFFIS (WP2), the initial platform has been revamped to model a quadcopter (Figure 14).The dual ARM Cortex A9 cores provide application processing, while the Leon and Microblaze work in lock-step on the safety-critical flight control. A voter/watchdog combination provides for error-correcting means by possibly recalculating inconsistent results on the ARM. A simpler error-detection scenario is also possible, where

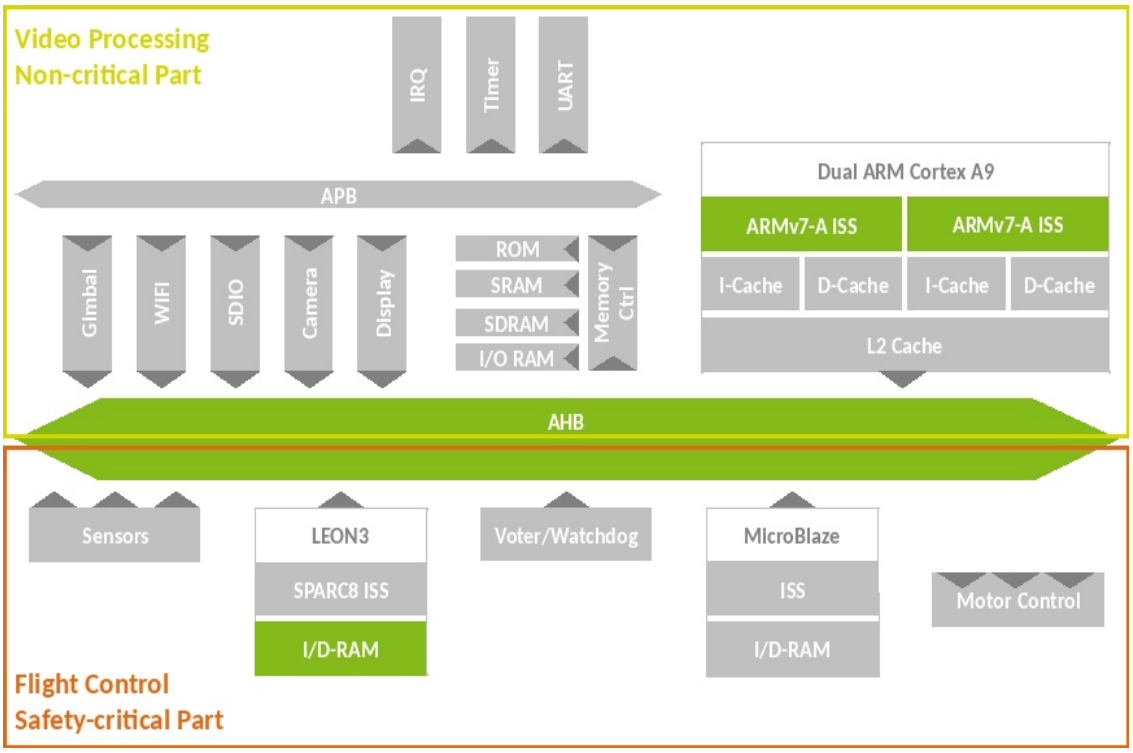


Figure 14 Mixed-Critical Quadcopter Platform

the ARM could trigger graceful failure. Some of the implemented algorithms are highlighted and explained in sections 19, 20 and 35.

3.7.2 **Core-based Support for Mixed-Criticality Applications (ST4.2.1)**

In Y2, we have developed an in-house SystemC multi-level (LT/AT/CT) model of the ARM Cortex A9, which has now been successfully integrated in the SoCRocket/Quadcopter platform. Generating a decoder for the rich ARMv7 instruction set has been a challenging task and has resulted in an algorithm for the automatic generation of non-orthogonal instruction sets (to be published). Furthermore, a tool has been developed that generates abstraction-agnostic SystemC processor models [5]. The ARM model can therefore be parametrized from a loosely-timed/functional down to a cycle-accurate/pipelined model. This enables varying the level of detail against simulation speed as required.

Besides the quadcopter use-case, we are analyzing a core clustering scenario, both to minimize concurrency over shared resources and to guarantee time-predictability. This will furthermore elucidate the effects of heterogeneity on data integrity and analyzability, as investigated in the following section.

3.7.3 Deterministic and Efficient Memory Hierarchies for Mixed-criticality Real-Time Systems (ST4.2.2)

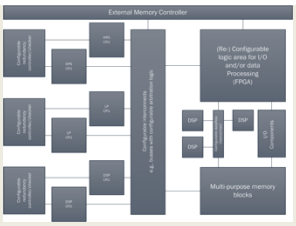
Tightly-coupled multi-core systems with shared memory and central, fine-grained task scheduling can achieve the highest core utilization, provided that low-overhead inter-core communication and data sharing is guaranteed. As the number of cores grows, however, the memory bottleneck dominates and caches become indispensable for upping performance. Caches, in turn, require mechanisms for keeping shared data coherent. Conventional coherence techniques deriving from the classic MSI protocol show largely nondeterministic timing behavior due to complex cache interactions, rendering them inapplicable to hard real-time systems. A time-predictable L1-cache coherence mechanism specifically tailored to real-time systems, has been developed in [6]. The goal is to enable fast access to shared data while maintaining a tight worst-case execution time (WCET) estimate, necessary for realistic timing analysis. The key idea is to hold shared data only as long as necessary, after which it is dumped to memory and reloaded before the next access. Only one core is granted access to a shared region at a single point in time. For this strategy to provide satisfactory performance, instructions are grouped into sequences, denoted by either shared or private. The cache does not attempt to maintain coherence as long as memory accesses are marked as private.

As soon as a shared block is entered, the cache switches to the afore-mentioned on-demand coherence mode. Thus, the granularity of shared/private blocks has to be carefully chosen: Smaller blocks enable finer interleaving and balancing between cores at the price of higher overhead for flushing and reloading. The performance of the caching strategy has been analyzed in [7]. The algorithm has been integrated into the LEON3 caches of SoCRocket, where we are currently fine-tuning our strategy in the context of mixed-critical applications.

3.8 NXP Germany (01R NXPGE)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



3.8.1 Memory compression techniques

On-chip memory in SOC systems is typically one of the largest area and therefore the main cost driver for embedded systems. One way to reduce memory is an external memory interface but this as well adds area to the IC and increases the BOM cost. Therefor it's typically not an option for competitive high performance architectures.

Status after the previous reporting period

As was shown in the previous reports, smart algorithms can help to reduce memory size at the cost of performance. In the first report year, we presented a compression scheme for audio PCM data in the NXP hybrid broadcast embedded multicore receiver.

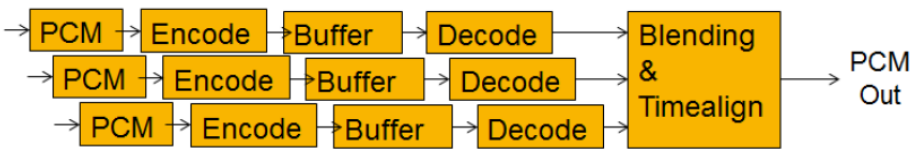


Figure 15 Memory compression of multiple audio sources

With this approach a memory reduction by a factor 5 could be achieved at the cost of audio quality. The picture below depicts an example receiver with three audio sources and the audio compression algorithm.

Intensive evaluation with different audio sources showed that the performance loss caused by the audio compression was greater than expected for some corner cases. The solution had three drawbacks:

1. For short delays the buffer was not fully used and a lower compression level would have been sufficient. The result was a worse performance than would have been necessary.
2. Very long delays could not have been handled because of the fixed buffer size and the fixed compression rate.
3. Additional features could not be introduced because of the fixed buffer memory and MIPS consumption of the compression algorithm.

Improved algorithm

To overcome these disadvantages, the algorithm was extended and was made more intelligent and adaptive with the goal to always get the best possible audio quality depending on the available resources. The requirements of the new algorithm were the following:

- Flexible memory size for the audio sample buffer with a guaranteed minimum size
- Adaptive compression level depending on:
  - the required audio delay
  - available memory
  - available processing resources of the multi core system
- Real time processing of incoming data
- Support of longer audio delays

The new algorithm is depicted in the following picture for an example of two audio sources:

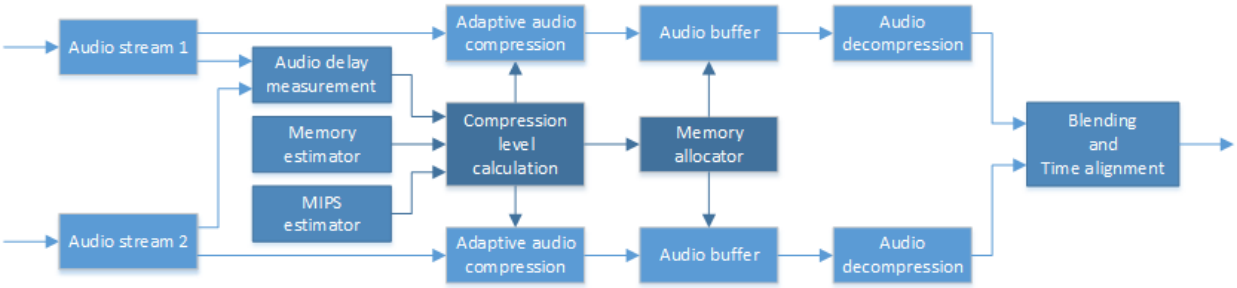


Figure 16 Advanced audio compression algorithm

The adaptive audio compression algorithm is based on the previous implementation and suitable methods of state of the art audio compression techniques. This not only allows flexible compression rates but also a higher compression level and therefor better memory usage.

Outcome

With this new approach a graceful degradation of the audio quality is possible, depending on the available resources in MIPS and memory. In any circumstances the best audio quality can be achieved, memory usage is optimized and the range of possible delays between audio sources is extended.

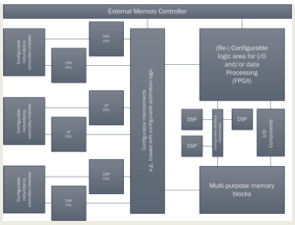
The minimum required memory could be reduced to half of the previous fixed memory, allowing the addition of other features in the future. The audio compression rate which was a fixed value of 5 is now flexible from 0 to 8.

This approach exploits the capabilities of a multi-core architecture and gives significantly better performance compared to a conventional design with no extra area for memory.

3.9 Imec-nl (17B imec-nl)

Technology lanes covered:

- **Heterogeneous Multiprocessor SoC architecture**
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- Virtualization and verification technologies



3.9.1 Modeling of Reliability of Memories

A test chip was fabricated in a 40nm ULP CMOS technology. Two SRAM macros of 90kb, an IMEC 18T-bitcell SC-SRAM that is implemented fully based on standard logic cells and a COTS 6T-bitcell SRAM, are integrated on each die. The silicon-measurement results on both SRAMs are outlined below. Here, we highlight a comparison, between the IMEC SC- and COTS 6T-SRAMs, particularly in reliability features when operating at a near- and sub-threshold voltage for energy-efficient computing.

Across 50 dies the DRV (data retention voltage) of each bitcell on both SRAMs has been measured. Figure 17 shows the statistic distributions of DRV on a die (left – IMEC SC-SRAM; right – 6T-SRAM). Log-normal distributions are manifested on both SRAM macros. The SC-SRAM however, features mean (μ) as well as standard deviation (σ) of DRV remarkably lower – 12% for μ and 17% for σ – in comparison with the 6T counterpart. This allows the SC-SRAM macros to operate at a more aggressive downscaling of voltage. As plotted in Figure 17 DRV distribution on a SC-SRAM (left) and a 6T-SRAM (right), on 46 of 50 dies the SC-SRAM macros work at a DRV of 300 mV. In contrast, the COTS 6T-SRAM macros give rise to a DRV of 313 mV on average and up to 410 mV across the dies. The SC-SRAM thereby improves the energy efficiency of SRAM designs significantly over the conventional 6T-SRAMs. A minimum read energy of 30 fJ/access/bit has been measured on the 18T SC-SRAM at a supply voltage of 700 mV – over 6X lower than the COTS 6T-SRAM Figure 18.

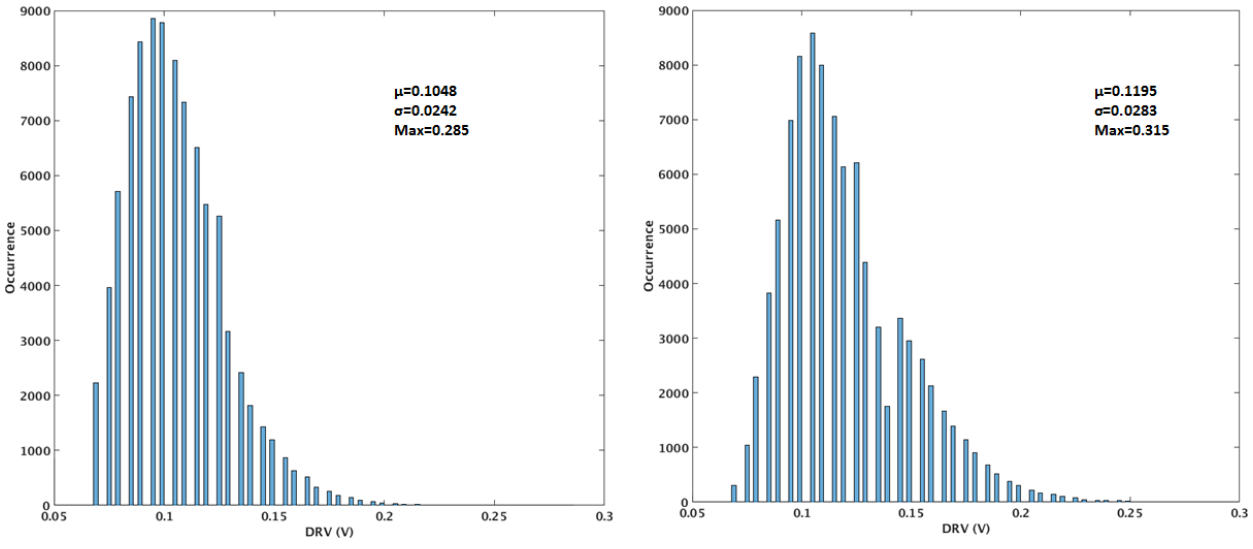


Figure 17 DRV distribution on a SC-SRAM (left) and a 6T-SRAM (right)

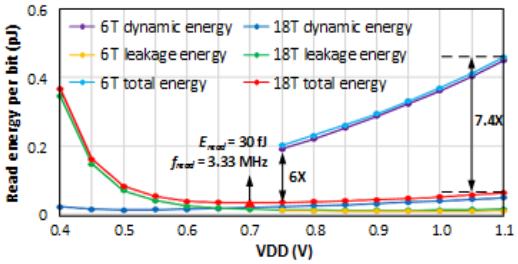


Figure 18 Read access energy of SC- and 6T-

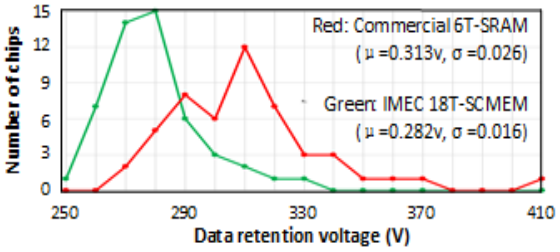
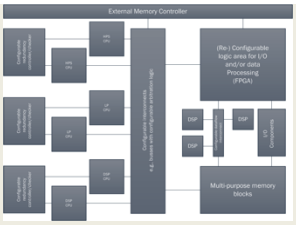


Figure 19 DRV measured on 50 dies

3.10 Chalmers (16A Chalmers)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- *Networking*
- *Virtualization and verification technologies*



3.10.1 Reconfigurable RISC architecture for tolerating permanent faults

As discussed in D4.4, Chalmers in T4.2 and T4.4 worked on a reconfigurable RISC multiprocessor array which is able to tolerate permanent faults by replacing faulty processor parts either with spare parts of neighboring processors (coarse-grain reconfigurability) or by instantiating them in a block of fine-grain reconfigurable logic (fine-grain reconfigurability). Thereby, a multiprocessor array can be more robust and tolerant to permanent faults. This relates to multiprocessor SoC architectures technology lane, as it considers multiple cores, as well as to the dynamic reconfiguration technology lane, due to its dynamic reconfiguability to bypass and replace damaged processor parts, repairing a faulty core.

In period 3, we evaluated the performance, area, and power overheads of the reconfigurable RISC processor, which was implemented in period 2 (a probabilistic analysis for evaluating the fault tolerance of the approach is described in D4.6). The results of this work were published in the following paper:

In summary we evaluated the performance, power and area costs of our adaptive processor using EEMBC CoreMark and Telebench 1.1 embedded benchmarks, and subsequently performed a design space exploration of the mixed-grain reconfigurability (exploring various granularities).

The coarse-grain and fine-grain parts of our design (henceforth denoted as CG and FG, respectively) are implemented in 65 nm technology considering STM-65nm and a Xilinx Virtex-5 65nm FPGA substrate, respectively.

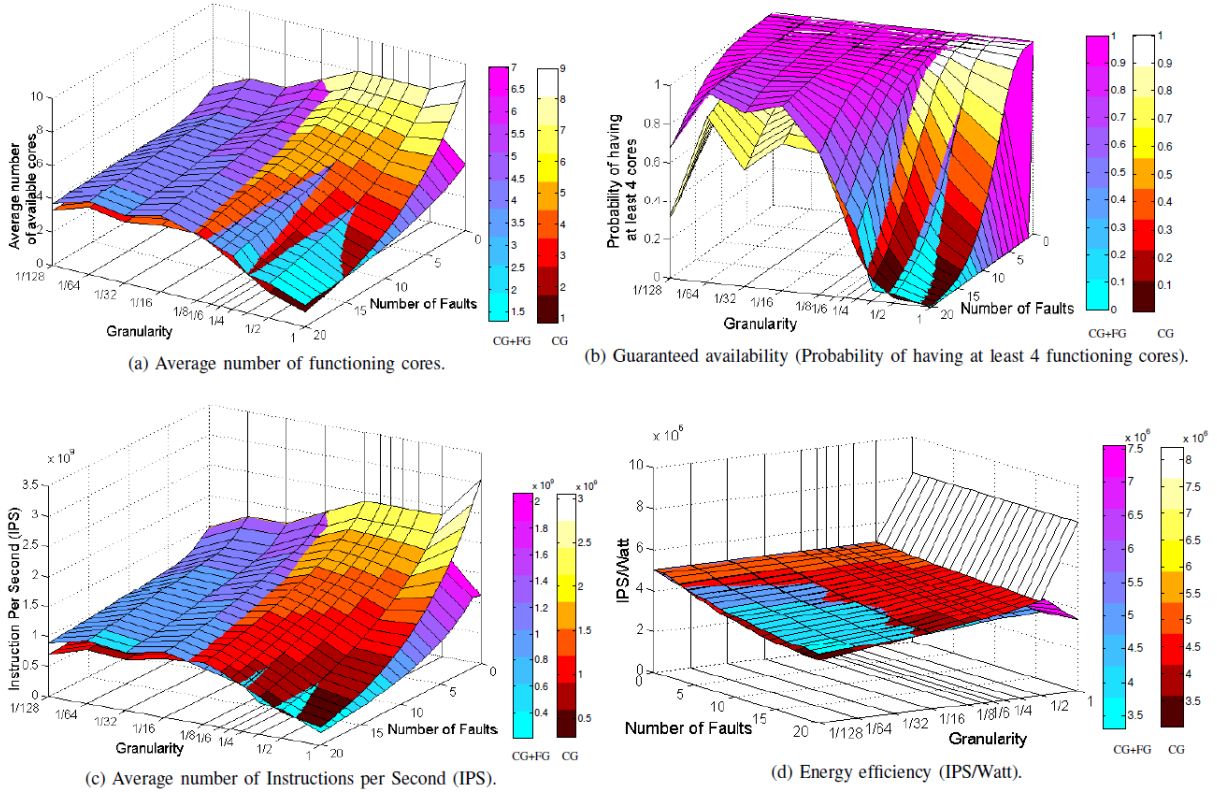
Our adaptive multicore array, when using only CG parts maintains 90% of the baseline RISC processor frequency, dropping from 500MHz to 450MHz. On the other hand, pipelines using fine-grain logic (denoted as CG+FG) operate at 40% of the baseline speed (200MHz), limited by the slowest fine-grain implementation of a substitutable block (SB).

The reconfigurable wires and the micro-architectural changes evenly contribute to the area overheads of the adaptive processor (only the CG part), which occupies 25% more area than the baseline. The area cost of the FG part is roughly 6 times the (ASIC) size of the largest SB it instantiates.

In general, when FG or distant CG spare blocks are used to replace faulty components, the performance of a single core drops to 1/2 or even to 1/4, its power consumption increases up to 2x, and its energy efficiency

reduces to 1/5. However, more efficient configurations maintain over 4/5 of their performance, have 0.8-1.4x of the baseline power consumption, and retain 60% of the baseline energy efficiency. The above constitute a significant cost for providing increased tolerance to permanent faults, however this is a fair price to pay for online self-repairing in safety-critical embedded systems with low accessibility.

Considering the above area, power, and performance overheads for a specific design point of our adaptive multiprocessors, we explored various granularity mixes for our mixed-grain Reconfigurable RISC architecture. As shown in, we evaluate the fault-tolerance, performance and energy efficiency of different CG and CG+FG Chip Multi- Processor designs at different fault-densities and granularities, all occupying the same area1. The number of reconfigurable wires needed per block is estimated using Rent's rule with Rent's exponent β= 0.5. We consider a fault density that causes up to 20 faults in the total area[2]. Even for our small RISC microprocessor, this translates roughly to 1-20 out of 10^7 transistors faulty, which is close to the current estimate for designing safety-critical systems in technologies below 18nm.



(a) Average number of functioning cores.

(b) Guaranteed availability (Probability of having at least 4 functioning cores).

(c) Average number of Instructions per Second (IPS).

(d) Energy efficiency (IPS/Watt).

Figure 20 Design-space exploration of coarse-grain and mixed-grain reconfigurable CMPs with various granularities, evaluating availability, performance and energy-efficiency at different fault densities. All designs use area smaller or equal to 9 baseline cores.

We quantify fault tolerance through the average number of available cores by analytically calculating the probability of constructing correctly functioning pipelines for the given options at each design point. Figure 20a shows that for low number of faults (≤ 2), component redundancy provides a higher number of fault-free components, as the area overheads of reconfigurability are not capitalized yet. For higher number of faults (≤ 8), coarse-grain reconfigurability of granularity 1/8 (CG(1/8)) maximizes availability of components. Adding fine-grain logic at smaller SB sizes (CG+FG(1/16)) is up to 13.5% better for fault densities beyond that point. Even at high number of faults, CG+FG(1/16) provides almost five available

[1] The silicon area constraint in our design-space exploration is equal to that of 9 baseline cores.

[2] We consider that faults are uniformly distributed in the silicon, including both the CG and FG part. For the FG part we consider that the faults can be tolerated in the above fault densities via reconfiguration, even without using any detection (testing) mechanism, simply by relocating the bit stream as proposed in [10].

components, tolerating over 3x and 1.5x more faults than component-redundancy (CG(1)) and other CG points, respectively.

Another measure of reliability, important for safety-critical systems that require guaranteed rather than best-effort performance, is the probability to deliver at least 4 working components (roughly half of the baseline components that fit in the area). Then, assuming that a system requires at least 4 processors to operate and that below that threshold it would fail, the above probability is equivalent with the probability of system failure (due to permanent faults in the processors). As shown in Figure 20b, the probability for the CG+FG(1/16) design does not drop below 99% for up to 20 faults. On the contrary, component redundancy (CG(1)) is below 90% and 50% beyond 6 and 10 faults, respectively. Finally, the best CG granularity crosses the threshold of 90% for more than 17 faults.

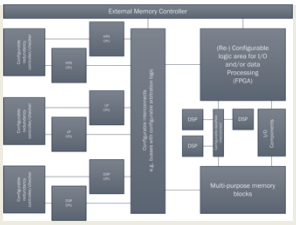
We measure performance and power by analytically calculating the probability of each particular processor configuration to use or not the FG block or to have a particular number of extra stages. Each of these configurations then has a specific IPS and power consumption[3] per core, extracted from our previous evaluation. Performance, shown in Figure 20c, follows the same trend as the average number of available cores, however it drops faster as the fault density increases. This is because the repaired cores have lower performance due to more stages or due to using the FG block more often. CG(1) scores better performance up to 4 faults. For higher fault- densities CG(1/8) is better, while CG+FG(1/16) provides up to 7% higher performance above 13 faults. Besides, as illustrated in Figure 20d, CG(1) is up to 1.7x more energy efficient than any other design alternative although for high fault densities the number of working cores reduces dramatically. For medium and high fault densities CG is up to 12% more energy efficient than CG+FG.

In summary, adding fine-grain logic in a CMP design to bypass hard errors can increase availability of the cores tolerating 3x more faults than core-level redundancy. Different fault densities require different granularities in the substitutable core-parts to maximize availability and minimize performance and energy overheads. Core-level redundancy is the most cost-effective solution for low fault densities. As the number of faults increases, coarse-grain and mixed-grain reconfigurability (of granularity 1/8 and 1/16, respectively) provide the best availability-cost tradeoff.

3.11 SELEX (11D SELEX)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architecture***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- ***Virtualization and verification technologies***



3.11.1 Mixed Critical Avionic Application

Multicore systems offer the possibility of consolidating multiple applications on a single computer, thus allowing a great saving in Space, Weight and Power (SWaP). We propose a proof-of-concept architecture on an i.mx6Quad (SoC), to consolidate four avionic applications: two are part of a Fligh Control System (FCS), the other two are part of an Engine Indicating and Crew Alerting System (EICAS). Three of these applications are considered of a lower criticality than the fourth one.

The architecture uses a partitioning mechanism to separate the applications and a specifically designed watchdog processor (WDP) IP deployed in a Field Programmable Gate Array (FPGA) used as a companion chip. Each application has its dedicated watchdog and is in charge of sending signatures to the WDP at predefined points. The WDP detects an error if no signature is received within a timeout or if an unexpected

[3] Unused processor parts are considered switched off and thus not consuming power.

signature is received. When an error is detected, the WDP asserts an interrupt request triggering a proper recovery action. A system watchdog timer (SWDT) is used to detect errors causing a system hang. If the SWDT is not refreshed within a given timeout, it sends a signal to an external interface and triggers a system reset. All hardware exceptions can trigger recovery actions, too.

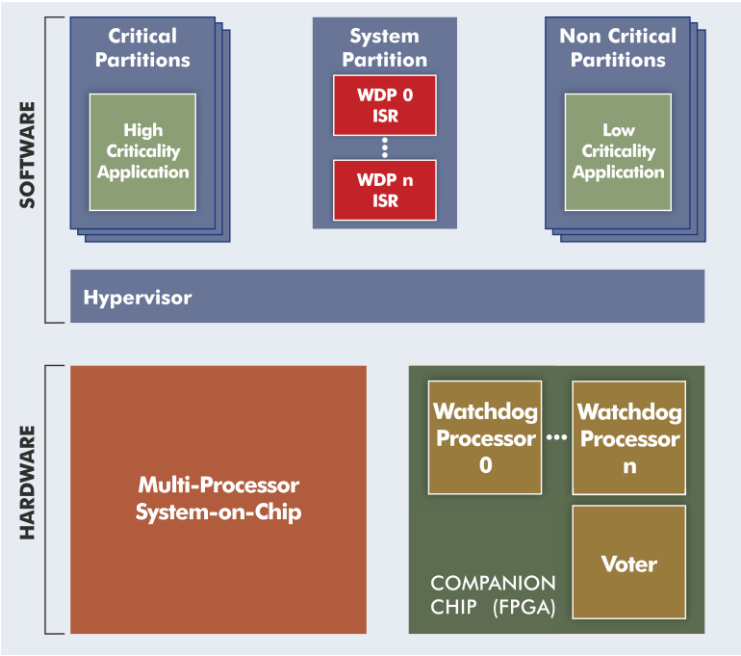


Figure 21 Hypervisor-based solution

The partitioning mechanism is based on a type-1 hypervisor. Each application is deployed in its dedicated partition and mapped on a specific core. The hypervisor is in charge of isolation. A third partition is used to manage the WDPs interrupts.

Through partitioning it is possible to guarantee the correct execution from a functional point of view, i.e. it is possible to always achieve correct results, by forbidding that a misbehavior in one of the low criticality applications can interfere with the data used by the critical partition. However, to be able to consolidate mixed-criticality applications on a multicore system, one other problem should be solved: temporal interference. By temporal interference, it is here intended any misbehavior that, if unchecked, can cause a longer-than-expected execution time. To detect and react to temporal interferences, we used a bounded-interference analysis in order to implement an online safety-net. The analysis is performed offline and it is based on data gathered during application design process, such as profiling data or results of worst case execution time (WCET) analysis. One of the metrics measured during profiling or WCET analysis is used online as an interference metric, i.e. a measurement that enables detection of a temporal interference. In our experiments, we used the data cache dependent stall cycles (DCSC) as an interference metric. DCSC is the number of clock cycles that a given core is stalled waiting for data from the data cache. It can be used as an interference metric because an abnormal value of DCSC is a symptom of an abnormally saturated interconnect, which is in turn an indicator of a misbehavior by one of the other applications in the system. DCSC can be measured using the performance counters available in the target SoC as well as in most other SoCs commercially available. Performance counters can be set and reset by means of special-purpose instructions and can trigger an interrupt on threshold crossing or can be read in polling.

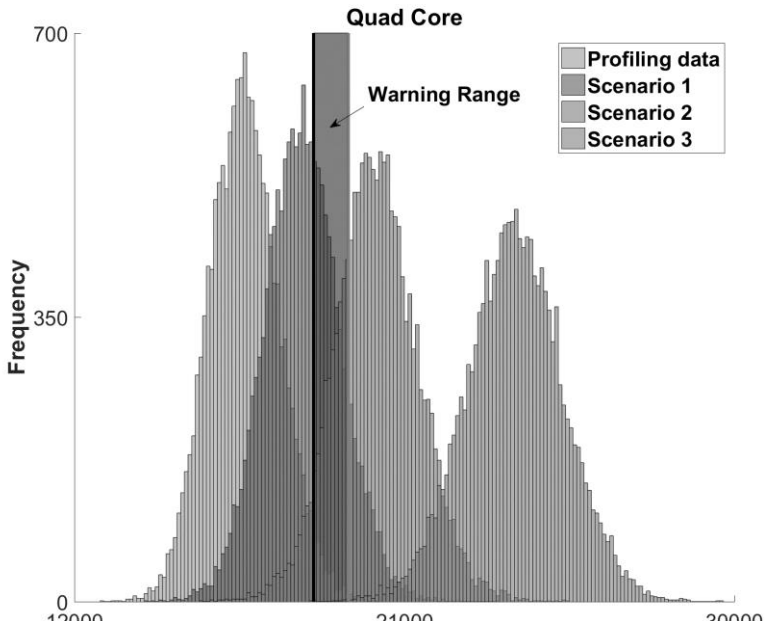


Figure 22 Experimental results on the quad core NXP i.MX6Quad. The graph shows the profiling data (nominal scenario) together with 15,000 measurements taken on the same monitored task while other tasks were affected by a bug as described in the text. The warning

The method is based on three thresholds: a warning threshold (T_W), a detection threshold (T_D), and an alarm threshold (α). If the interference metric is above T_D, an interference is occurring and a recovery mechanism should be actuated. If the interference metric is above T_W, attention must be paid to the situation but it does not mean necessarily that an interference is occurring. The system counts the number of consecutive measurements by which the interference metric is above T_W, and actuates a recovery mechanism if this counter crosses the α threshold. All three thresholds are defined by statistical profiling of the WCET analysis data.

The experimental evaluation was performed by running the critical task in a nominal case and in three interference scenarios. In the nominal case, the system was running its expected workload, with four tasks running in parallel, each on its own core using the partitioning mechanism described above. In the interference scenario 1, one of the low criticality applications was affected by a bug causing continuous accesses to memory. In interference scenario 2, there were two low criticality applications affected by a bug causing a similar misbehavior. In interference scenario 3, there were three buggy low criticality applications. In each scenario, 15000 measurements were performed.

Results are reported Figure 22 and in Table 3.

Table 3 Detection results on the quad-core NXP i.MX6Quad architecture. Scenarios are specified in the text.

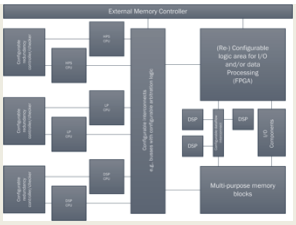
	Alarm rule detections	Warning rule detections	Tolerated
Scenario 1	1668 (11.12%)	1114 (~7.43%)	12218 (~81.45%)
Scenario 2	11348 (~75.65%)	528 (3.52%)	3124 (~20.83%)
Scenario 3	14990 (~99.93%)	0	10 (~0.07%)

Results show that the method is able to identify significant interference and to tolerate those interferences which are not able to change the execution time of the monitored task enough to require a recovery action. Results gathered on both dual and quad core architectures show the viability of the method for what concerns interference detection and recovery. The thresholds defined in the proposed method can be fine-tuned to achieve an acceptable accuracy-performance trade-off, as needed by the designer. Such thresholds also allow to classify interferences as critical or non-critical, and implement proper and differentiated recovery actions in each case.

3.12 NXP Semiconductor (17C NXPNL)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- ***Networking***
- *Virtualization and verification technologies*



3.12.1 Parallelization in multi-core architecture

The Software Defined Radio (SDR) application has been deployed using the multicore architectures to address ever-increasing performance requirements while being constrained by power dissipation limitations

In addition, the Heterogeneous Multiprocessor methodology has been implemented also in a second architecture derivative. This product is to be used in a different application domain (Vehicle to Vehicle/Infrastructure communication, V2X). In this case, a microcontroller is combined with a SIMD architecture (Single Instruction Multiple Data Signal) DSP and software configurable hardware accelerators, with tight integration especially between the DSP and the accelerators. The relevant blocks are highlighted in below figure.

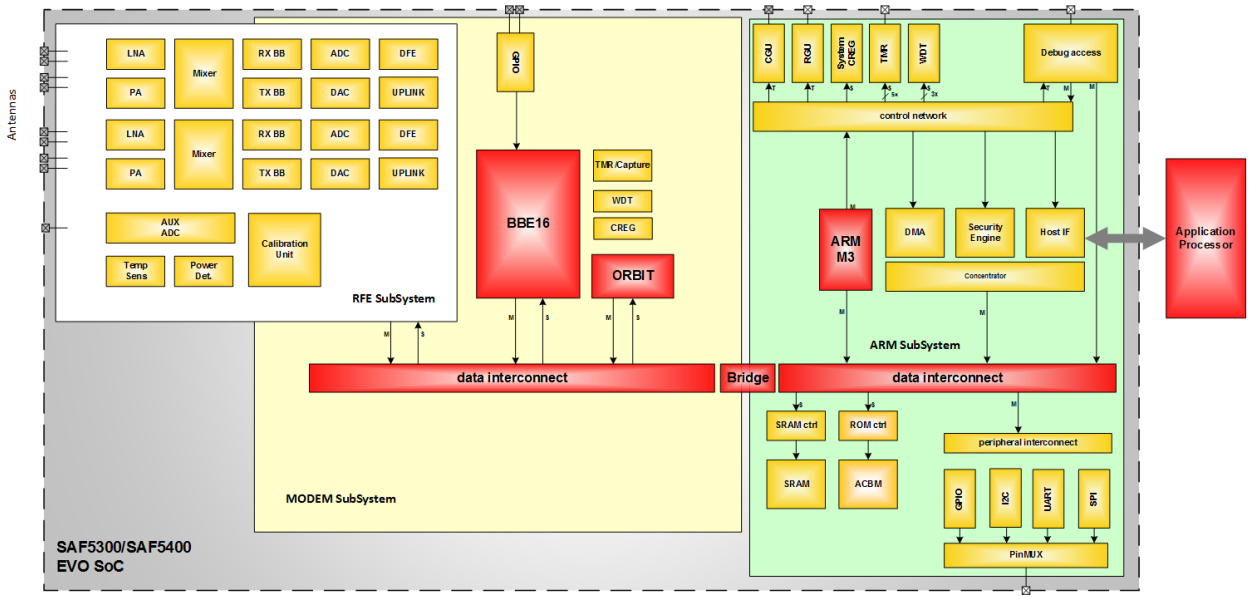


Figure 23 Top-level block diagram of a V2X product

The integrated microcontroller is here the ARM Cortex M3 as depicted in the green 'ARM subsystem' component. It is connected via an on-chip AHB bus network to 'MODEM subsystem'. This subsystem contains the SIMD DSP 'BBE16' and configurable accelerator 'ORBIT'. The DSP is tightly coupled to the accelerator, by means of instruction set extensions which directly control the ORBIT at low and controlled latency. Data flow from the ORBIT does not have to go via the DSP, and can directly use the on-chip AHB bus network. A third subsystem (RFE Subsystem) contains the analog radio parts and analog-digital domain crossings.

This approach allows for a flexible allocation of processing tasks to those elements optimally suited for the type of operations required:

- The DSP BBE16 performs the compute-intensive algorithmic operations as channel equalization and FFT transforms, optimally exploiting the SIMD architecture
- The ORBIT accelerator is used for the so-called 'outer receiver' tasks, which include interleaving, viterbi decoding and forward error correction: typical more bit-oriented and data permutation type of operations, not efficiently done on classical signal processor architectures
- The ARM processor takes care of control and administrational tasks, as message parsing and host processor protocol implementations.

These operations can be paralleled in a pipeline strategy, optimally using the available resources.

The system makes use of a data flow approach, making sure data blocks (like received or to be transmitted messages) are efficiently routed through the system using DMA mechanisms. For example, the ARM is not required to fetch the payload of messages, as these can be streamed directly from buffer memory to the host application processor (outside the SoC system). Instead, the ARM can be focused on administrative tasks, allowing reduced operating frequency and thereby also power consumption.

The main elements are programmable/configurable, thereby implementing a Software-Defined Radio system allowing in-the-field upgrades to evolving or new standards, for example using Over-the-Air (OtA) update mechanisms. A first actual tape-out of an instantiation based on this architecture is expected mid-2017.

3.13 Infineon Technologies AG (01A IFAG)

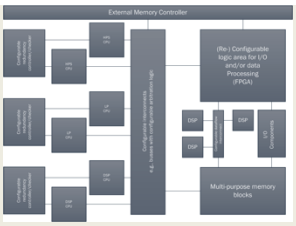
The work reported by the Infineon Austria AG reported in section 3.2.2 has been performed in cooperation with IFAG.

4. T4.3 Core and chip interconnect, peripheral devices

4.1 Chalmers (16A Chalmers)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- ***Networking***
- *Virtualization and verification technologies*



4.1.1 Resilient Service-oriented NoC

As discussed in D4.4, Chalmers in ST4.3.1 worked on the design and implementation of Service-oriented NoCs, which are resilient to permanent faults. These are NoCs that provide separate NoC resources to different traffic-classes/services, each service having different QoS requirements. Our design provides resilience to permanent faults by compromising this isolation among services when a fault appears. This is achieved by mechanisms that allow service traffic to be redirected when some of its NoC resources are damaged, either using other resources of its own service (intra-service redirection, Service detour - SDetour) or by using NoC resources of other services (inter-service redirection, Service merge - SMerge). This relates to multiprocessor SoC architectures technology lane, as it interconnects multiple cores or SoC components in a multiprocessor architecture, as well as to the Networking technology lane, naturally due to the topic of on-chip interconnection networks.

In period 2, we have completed the design and implementation of the techniques on a service oriented NoC called RQNoC. In period 3 we evaluated the fault tolerance of our approach and its performance, power and area overheads. Our research finding were published in the following paper:

- "RQNoC: a Resilient QoS NoC with Service Redirection", A. Malek, I. Sourdis, S. Tzilis, Y. He and G. Rauwerda, ACM Transactions on Embedded Computing Systems (TECS), Vol. 15, No. 2, Article 28, February 2016.

Below we summarize the evaluation results of three variations of the proposed resilient NoC (RQNoC) in terms of fault tolerance and network performance. Moreover, we retrieve post-synthesis results for operating frequency, power consumption and area cost of the resilient NoC designs and compare them to the baseline NoC.

Area, Power, Frequency overheads

We have implemented in RTL three 4x4 resilient NoC 2D mesh designs that provide (i) service detour (SDetour), (ii) service merge (SMerge), or (iii) both (SMerge+SDetour). We further implemented a baseline 4x4 NoC 2D mesh and a NoC providing global detour[4] (GDetour). All designs, except the baseline, tolerate permanent faults at the links and router-control as described in D4.4.

The designs are implemented using ST 65nm low-power library in Synopsys Design compiler. Power consumption is measured by simulating the designs netlists for 1ms. Table 4 summarizes the overheads in operating frequency, power consumption and silicon resources of the three alternative Resilient NoCs versus the baseline NoC. The NoC with GDetour has the same area and frequency with the NoC with SDetour and is therefore omitted from the table. Employing Service detour increases the area by 24.3%; 2/5 of it is due to the fault tolerant links, 2/5 due to the triplication of the common router control parts, and

[4] All services are redirected the same way.

the remaining 1/5 due to the changes in flit handling for the additional detour header. SDetour power consumption is increased by 6.8% and its operating frequency is equal to the baseline. SMerge introduces a higher area overhead of 50.8% primarily due to the changes in the control needed for merging services and to a smaller extend due to a few datapath additions for tagging packets of different services. SMerge power consumption increases compared to the baseline by 26.6% and its clock rate decreases by 9.1% due to the complexity of the router control. Finally, a resilient NoC integrating both SDetour and SMerge requires 55.8% additional resources mainly and consumes 31.6 more power, having 9.1% slower clock.

Table 4: Implementation results of the resilient and baseline designs.

Design	Power (Watt)	Area(μm2)	Frequnecy(Mhz)
Baseline NoC	1.63	946k	1100
Resilient NoC Smerge	2.06 (+26.6%)	1428k (+50.8%)	1000 (-9.1%)
Resilient NoC SDetour (also GDetour)	1.74(+6.8%)	1117k (+24.3%)	1100
Resilient NoC SMerge + Sdetour	2.14 (+31.6%)	1476k (+55.8%)	1000 (-9.1%)

Performance results and Fault Tolerance

We evaluated the above 4x4 resilient NoCs and the baseline NoC in terms of performance, measuring packet latency and throughput, and in terms of fault tolerance, measuring the percentage of transmitted packets that is successfully received. Our experiments have been performed simulating the RTL of the network designs for 1.5 million cycles each, having a warm-up phase of 20 thousand cycles. Synthetic uniform random traffic is injected with up to one flit/cycle/node injection rate distributed among the four SVCs as described in Table 4.

We performed experiments injecting to the networks 0, 1, 4, 8, 16 and 32 faults, the location of which is randomly selected considering the silicon area of each network part. It is worth noting that, all networks have the same performance when fault free.

In summary, Resilient NoC SMerge provides the highest fault tolerance for up to 16 faults, but introduces significant performance overheads increasing packet latency up to 2× and reducing throughput to 50% in the worst case compared to a fault-free network. Resilient NoC SDetour performs better in terms of latency and has similar throughput reduction with SMerge, but is not able to tolerate more than a handful of faults in most cases. Resilient NoC that combines SMerge with SDetour, increases fault tolerance, successfully delivering 83% of the packets on average even for a network with 32 faults. Compared to SMerge, its latency slightly increases for high priority traffic and reduces for low priority, but throughput is overall lower.

The Figure below depicts a thorough fault tolerance evaluation of the proposed NoCs and compares them with a conventional Noc that employes standard Detour (Global Detour - GDetour). We measure for different fault densities (number of faults), the mean network connectivity as well as the range of values that are up to 3 standard deviations ($\pm 3\sigma$) away from the mean. SMerge+SDetour provides the highest network connectivity, which is above 99% for up to 8 network faults, 92% for 32 faults, 80% for 50 faults and 37% 100 faults. SMerge alone can support a similar connectivity preserving 99.3%, 89%, 75% and 30% of the network paths for 8, 32, 50 and 100 faults, respectively. SDetour follows with 82%, 41%, 20% and 3.6% connectivity for the same fault densities. Finally, conventional NoCs with Global Detour of traffic (GDetour) shows poor fault tolerance maintaining only 44% of the network connectivity at 8 network faults, 3.5% for 32 faults and below 1% for more than 50 faults.

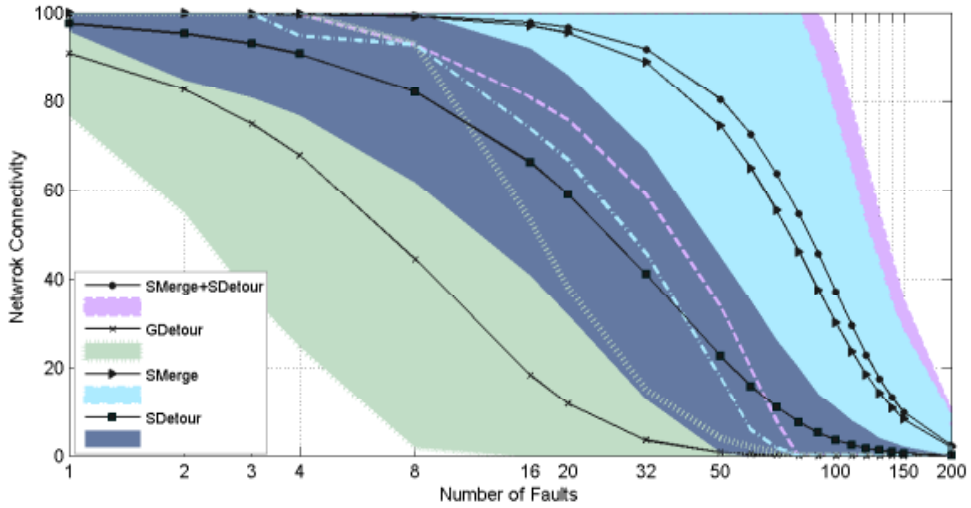


Figure 24 Mean and ±3σ range of connectivity values for the RQNoC designs at different fault densities.

The Figure below illustrates the probability of a network to offer 100%, 75%, 50% and 25% connectivity at a particular fault density. This is important for networks that are expected to guarantee a particular quality of service. SMerge and SMerge+SDetour offer similar fault-tolerance having above 80% probability to offer a fully connected network even with 20 permanent faults. The probability of full connectivity is significantly reduced for SDetour and GDetour as the turn model used for the routing cannot always connect all node pairs. However, as the connectivity requirements reduce to 75%, 50% and 25%, SDetour performs better than GDetour. Even so, SMerge and SMerge+SDetour can deliver at fault densities 2-3x higher the same probability as SDetour for a particular network connectivity.

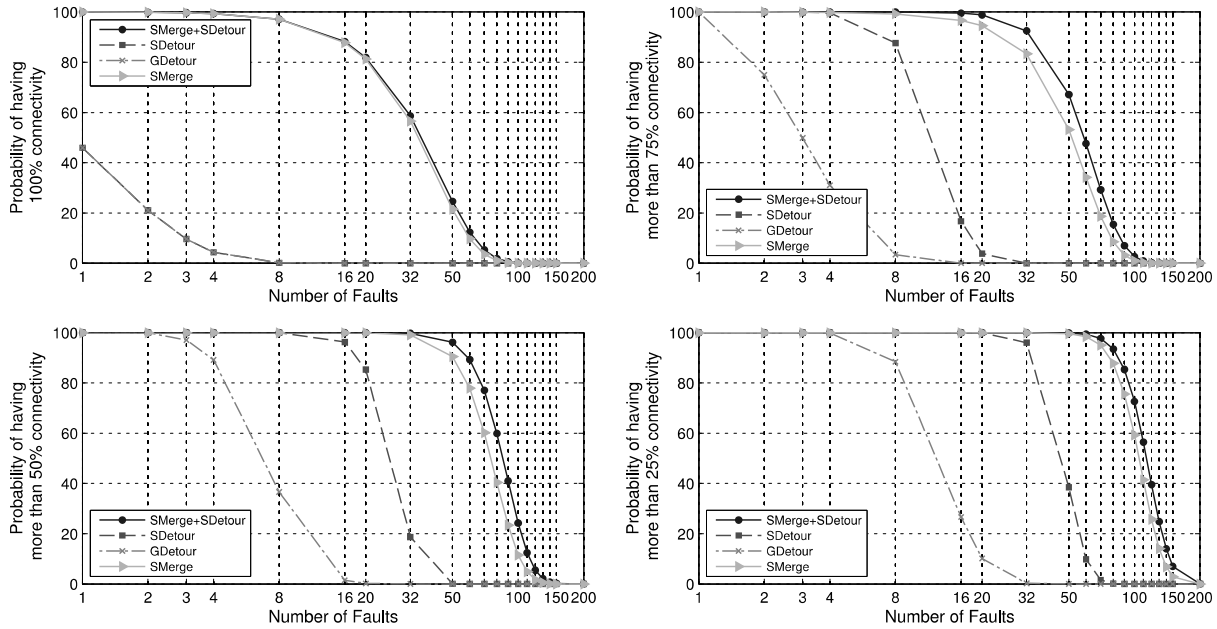
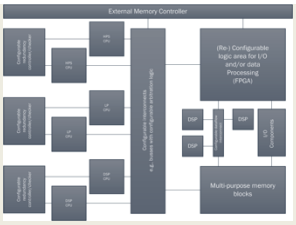


Figure 25 Probability of different RQNoC networks to deliver a particular network connectivity under different fault densities.

4.2 TU Wien (02E TUW)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- *Virtualization and verification technologies*



4.2.1 Heterogeneous TTNoC Many-Core Architecture

As described in D4.5 TUW is working on a heterogeneous many-core architecture implemented on a hybrid SoC platform Altera Arria V. The focus of the activity in the first two reporting periods was to acquire relevant requirements, build a specification and develop a design for parting of legacy hardware on a new platform and augmenting it with the additional hardware to incorporate hard-coded heterogeneous component.

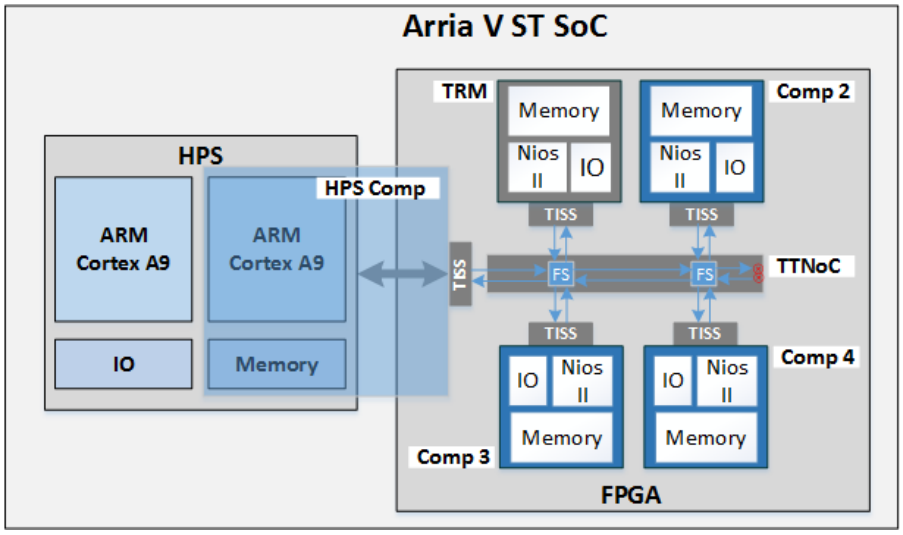


Figure 26 Heterogeneous TTNoC Many-Core Architecture

In the final reporting period TUW experimented with the architecture presented in Figure 26 on several topics:

- Platform capabilities
  - Capacity
  - Configuration and reconfiguration
- Integration capabilities
  - System and application integration
  - Software integration,
  - Tools and deployment integration.
- Fault tolerance capabilities
  - Monitoring
  - Reconfiguration

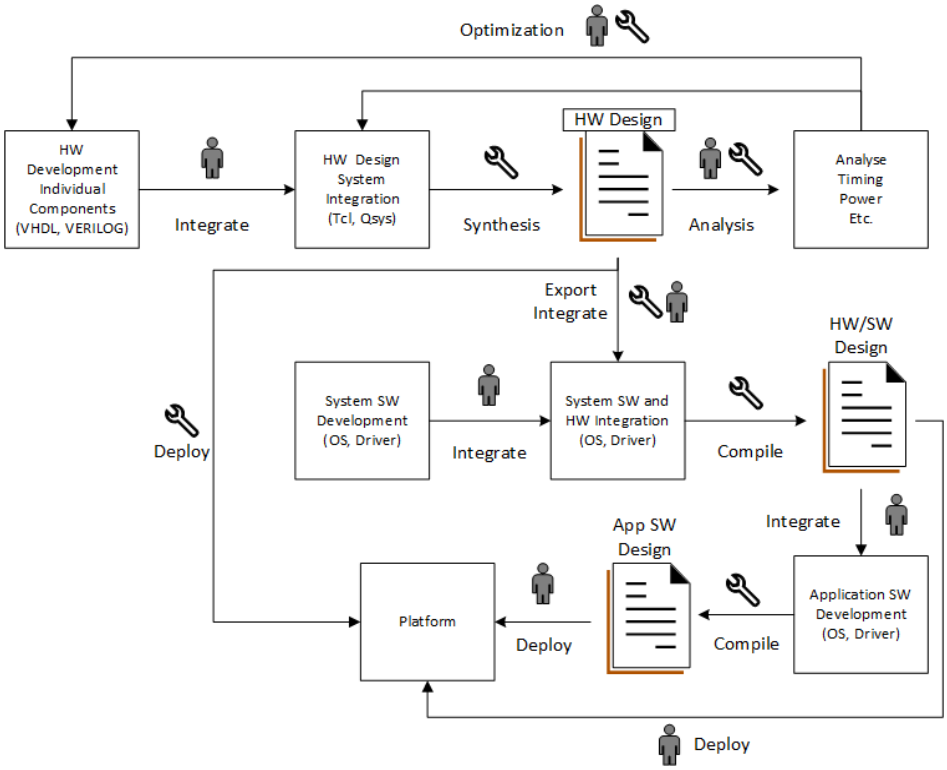


Figure 27 Simplified design and deployment model

A huge factor in the selection hardware platform is cost and the ability to fuse it with the application system. One of the goals is to explore usability of the presented architecture on a larger scale, this means to provide modular implementation model where the architecture can be molded according to application needs. The implementation presented in Figure 26 uses about 12% of the FPGA logic elements, 22% of block memory bits and 12% of pins. This platform is capable of hosting a much larger architecture. Also, the modular approach allows us to implement this platform even on the low scale platform, making it more accessible.

A part of the research on the presented architecture is the conformity to the selected platform and its portability to other hybrid SoC platforms. This is partly connected with the activity presented in Section 153.4. As a major topic related to mixed-critical application is integration on different abstraction levels and different phases of development. To achieve the usability levels of COTS hardware similar platforms must provide tools for seamless integration such that end user must invest minimal effort in building a given application. Figure 27 provides simplified model for the development on the presented platform. It highlights the manual and tool/automated actions. One can see that the complexity of this process even on this scale would present itself as a challenge for the average application developer. As a conclusion we determined that this process needs to be simplified by adding and unifying toolchains, and we identified potential activities in WP2 and WP3 that represent a step in the right direction (e.g., art2kitect, PikeOS).

The future goal is to build a tool which combines current toolchain with the architecture and application specific configuration. This would reduce percentage of manual actions required to implement an application on the presented architecture. This tool would integrate all actions from hardware architecture configuration, TTNoC configuration (routing, schedules etc.,), system software configuration and application mapping. This approach would simplify the process depicted in Figure 27, provide end-to-end integration and increased

On the level of core, we have explored the use of single-path code. Having a code with single execution trace allows the TTNoC node to achieve full time-predictability and with that also to be deterministic considering the task respond time. Furthermore, due to the drawback of such a code to consume long execution time, we have also designed a new time-predictable memory that would improve performance of the node that runs task consisted of single-path code. From performed measurement, we show that the new memory hierarchy can achieve up to 17% improvements of the execution time.

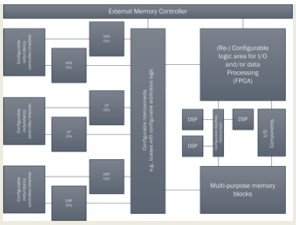
The hybrid SoC allows cross configuration of individual devices, which can be utilized as a fault tolerance mechanism. The initial fault hypothesis of the architecture defines single component as a fault containment unit and SoC in general as an error containment unit. The monitoring and re-configuration capabilities provide additional layer in the fault hypothesis structure.

The work performed in the project is described in couple of publications namely the heterogeneous TTNoC architecture in [7], while to publications are still pending.

4.3 The Institute of Information Theory and Automation (04D UTIA)

Technology lanes covered:

- ***Heterogeneous Multiprocessor SoC architectures***
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- *Networking*
- *Virtualization and verification technologies*



4.3.1 Asymmetric Multiprocessing on ZYNQ

Video processing and very fast digital I/O requires processing based on combination of standard processors and HW acceleration blocks in programmable logic. Xilinx Zynq devices contain two 32bit ARM Cortex A9 processors and the programmable logic on a single (28nm) chip.

We describe set of UTIA HW accelerator designs for a stand-alone EMC2-DP-V2 platform developed by Sundance. Board provides PCI-Express connectivity, serving for construction of larger (stacked) parallel systems. Each EMC2-DP-V2 board can work with the standard industrial grade Zynq System on Module (SOM) TE0715-03-30-1I or with faster TE0715-03-30-3E module.

A standalone EMC2-DP-V2 board is extended with an FMC I/O card serving for HDMI input of Full HD video data and for HDMI output in Full HD. See Figure 28.

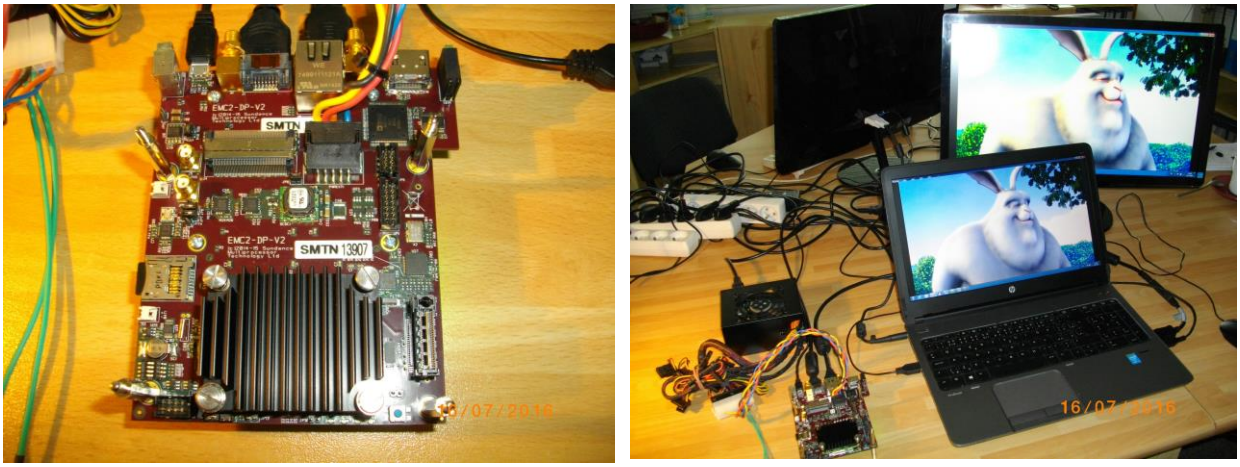


Figure 28 Standalone EMC2-DP-V2 platform with 3x (8xSIMD) EdkDSP (detailed view in left) and Full HD motion detection algorithm accelerated in HW (right).

Figure 29 presents the developed asymmetric multiprocessing system (AMP) with 3x (8xSIMD) EdkDSP runtime reconfigurable floating point accelerators. Each accelerator consists of an 8xSIMD DSP floating

point unit for vector data processing and the PicoBlaze6 sequencer. The EdkDSP algorithms are sequences of elementary operations supported by the floating point 8xSIMD DSP unit.

Figure 29 demonstrates the three parallel Full HD video processing HW accelerator data paths generated in the Xilinx SDSoC 2015.4 environment. The generated parallel video processing data paths correspond to three parallel C/C++ function call instances (in a setup similar to synchronization of parallel SW threads with SW barrier. Clock frequencies CLK1 … CLK6 (indicated in Figure 29) are specified in Table 5. The evaluation designs support an ILA capable to sample 32k words with CLK5 resolution. Performance for both SoM modules is summarized in Table 6.

Table 5 Clock frequencies

Carrier (Sundance)	System on Module (Trenz electronic)	CLK1 (MHz)	CLK2 (MHz)	CLK3 (MHz)	CLK4 (MHz)	CLK5 (MHz)	CLK6 (MHz)
EMC2-DP-V2	TE0715-03-30-1I	666.7	148.5	148.5	200.0	150.0	125.0
EMC2-DP-V2	TE0715-03-30-3E	1000.0	148.5	148.5	200.0	200.0	142.8

Table 6 Performance of the EMC2-DP-V2 platform

Video algorithms in Full HD	**EMC2-DP-V2 TE0715-03-30-1I**	**EMC2-DP-V2 TE0715-03-30-3E**
sh01 (1x sobel edge detection)	SW: 7.3 FPS HW: 49.7 FPS	SW: 11.0 FPS HW: 57.7 FPS
md01 (1x motion detection)	SW: 1.2 FPS HW: 42.8 FPS	SW: 1.7 FPS HW: 51.8 FPS
md01 (1x motion detection)	HW/SW acceleration: **36 x**	HW/SW acceleration: **30 x**
Single (8xSIMD) EdkDSP	In parallel to the video processing	In parallel to the video processing
LMS adaptive filter (2000 c.)	0.914 GFLOP/s	1.189 GFLOP/s
FIR filter (2000 coefficients)	1.419 GFLOP/s	1.798 GFLOP/s
Peak: 3x (8xSIMD) EdkDSP	7.200 GFLOP/s	9.600 GFLOP/s

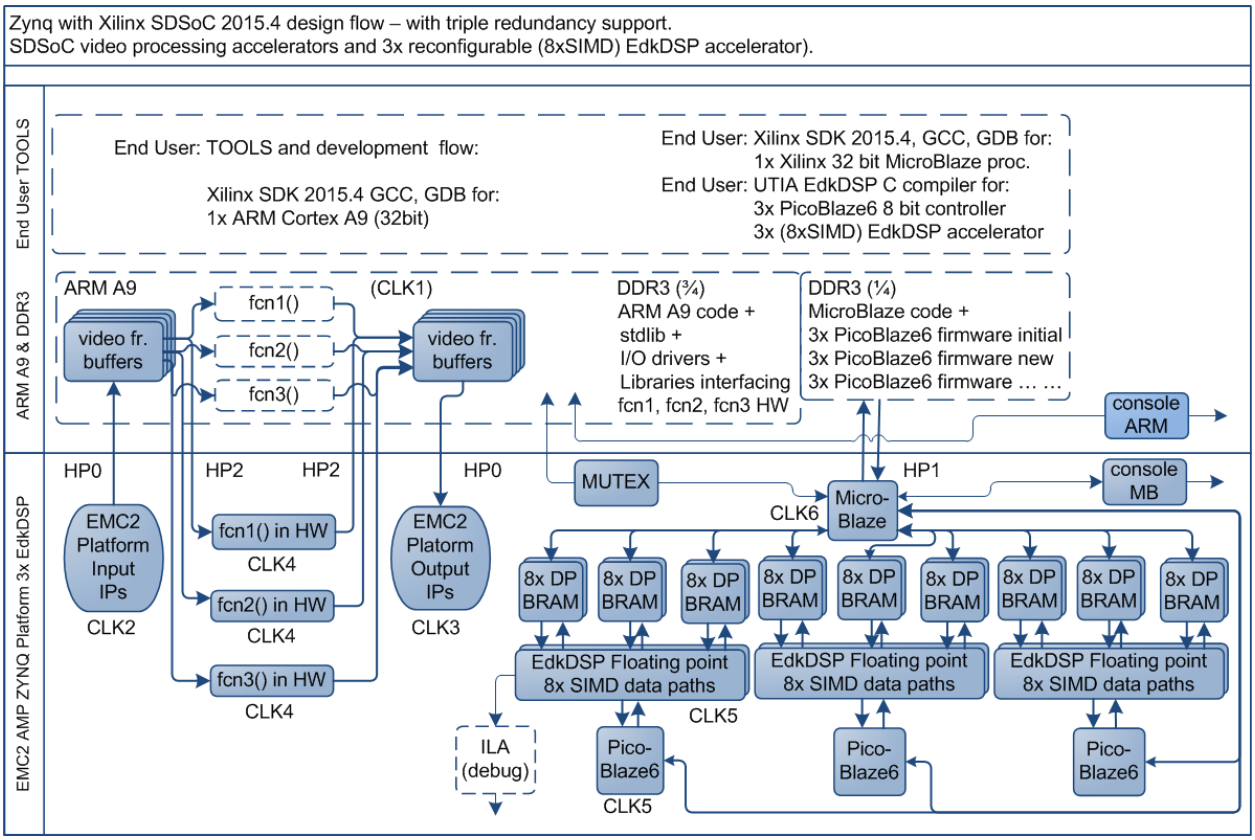


Figure 29 AMP system with 3x (8xSIMD) EdkDSP and Full HD HDMII-HDMIO for EMC2-DP-V2

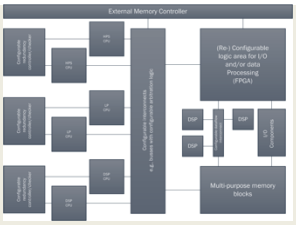
The detailed application note [8] (65 pages) and the evaluation package for the EMC2-DP-V2 can be downloaded from the UTIA public server: http://sp.utia.cz/index.php?ids=results&id=s30i1hm4

Contact: Jiri Kadlec; UTIA AV CR v.v.i. Prague, CZ; kadlec@utia.cas.cz tel. +420 2 6605 2216

4.4 AVL (02A AVL)

Technology lanes covered:

- *Heterogeneous Multiprocessor SoC architectures*
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- ***Networking***
- *Virtualization and verification technologies*



4.4.1 Networking for V&V of mixed-criticality applications on multicore automotive control units

The main idea of this work was in providing means of communication between multiple tasks of a multi-core automotive ECU system on one hand, and a multi-core test (V&V) system on the other hand, while using available communication networks (specifically: CAN, later CAN-FD, automotive Ethernet). Having multiple, independent tasks operate on both sides means that existing communication links (e.g. a single CAN connection) need to be shared. In order to allow independent composition of the tasks, this "multiplexing" should be transparent to the application, e.g. tasks shall remain independent and do not need to consider other tasks. Thus, the management of the communication resource needs to come from a central middleware, which we proposed and implemented as ConnectivityManager and ConnectivityInterface.

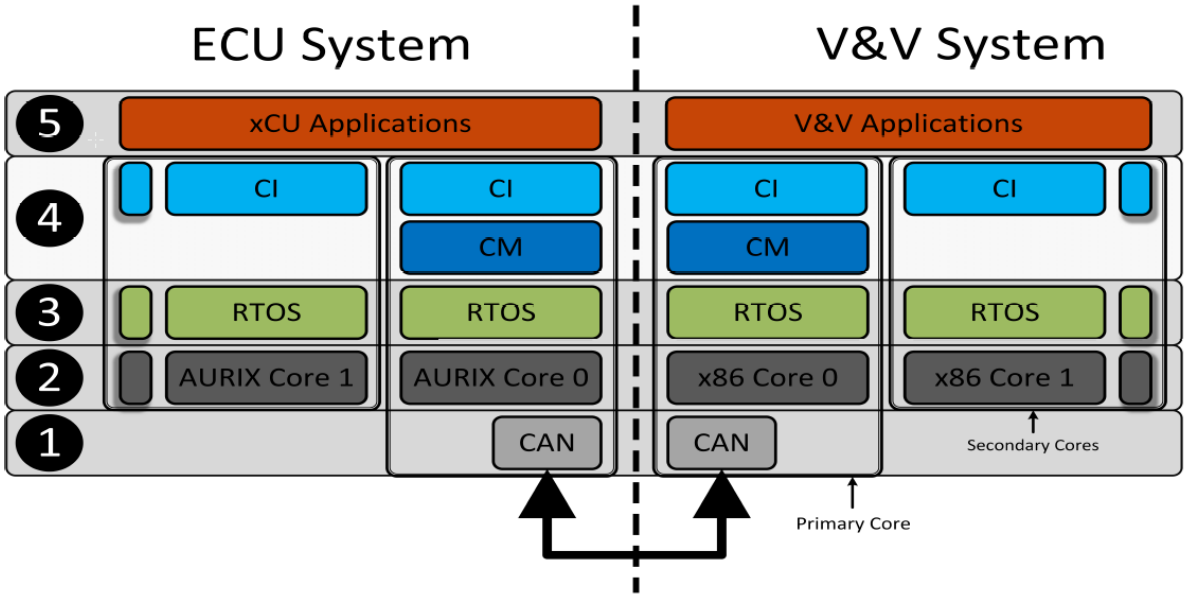


Figure 30 System Overview

Besides real-time constraints this new architecture also complies to criticality aspects. If resources / bandwidth of the communication channel is sufficient (should be the "normal" state), all deadlines are met. If the bandwidth temporarily decreases (e.g. due to the need to re-transmit many CAN messages, or because a task has a temporary, unforeseen burst of communication), the ConnectivityManager re-assigns bandwidth based on criticality. This bandwidth scheduling mechanism is to the best of our knowledge new, and will be filed for patent. In addition, a publication at a suitable conference is planned.

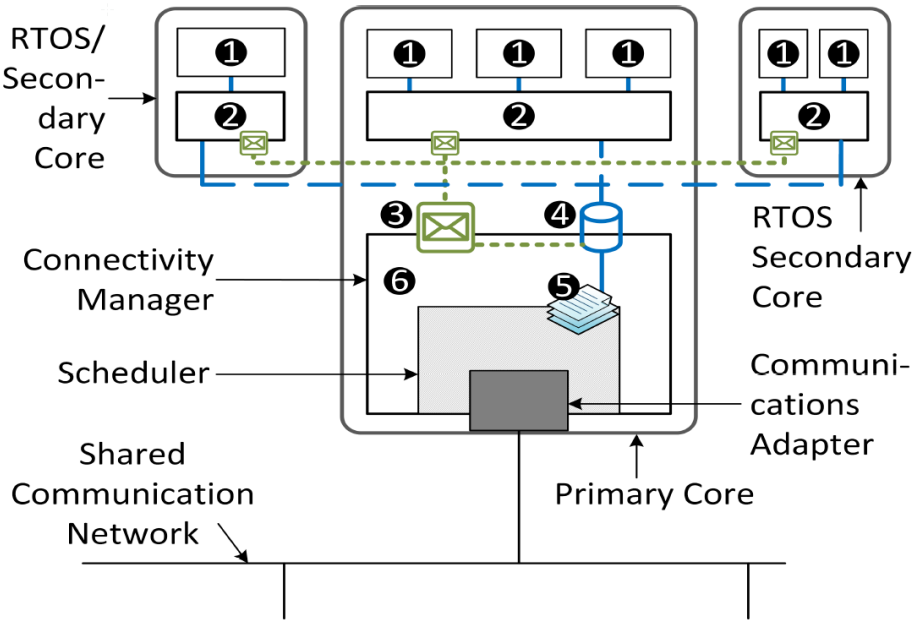


Figure 31 Inter-core communication scheme.

The implementation of the prototype was done in C++ and runs as an independent .rta process on the real-time operating system INtime in the version 4.20. As already stated, the prototype consist of two separate components, which is necessary to achieve an appropriate inter-core communication.

Detailed measurements achieved the following results:

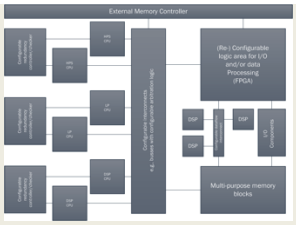
- Prioritization of information streams across multiple cores worked
- Bandwidth utilization up to 98 % on CAN
- Stable system in long term runs (88 hours test)
- Handling multiple information streams simultaneously worked
- just minor increase of RT Kernel CPU usage (6-8 %)

As a summary, the Test (V&V) system supports now multi-core architecture with real parallel communicating tasks, while respecting mixed-criticality of applications. Due to the increased complexity of the system, a flexible scheduling approach was designed. The demonstrator shows a novel dynamic priority communication scheduling approach that is capable of adapting to bandwidth changes on the shared communication network. The validation of the concept was done in both simulation and measurements on real CAN networks and proves the functionality of the solution in close to real world scenarios. The behavior of this solution in overload situations was tested and worked remarkably robust, allowing to maximize bandwidth utilization of the underlying CAN bus.

4.5 SevenSols (15O SevenS)

Technology lanes covered:

- *Heterogeneous Multiprocessor SoC architectures*
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- ***Networking***
- *Virtualization and verification technologies*



4.5.1 Extending QoS and Redundancy for High-Accurate Distributed Control Systems

Distributed control systems have strict requirements for control commands transfer (critical packets). This data should reach all network nodes in spite of network congestion (QoS) or even when any of the links between nodes experience a failure.

In our application scenario, timing data is considered critical, whilst the rest will be considered as non-critical. White-Rabbit, known for being the most accurate time distribution solution, will be used to synchronize all network nodes. Thus, the extension of QoS and the development of redundancy protocols such as HSR (High-availability Seamless Redundancy) will be focused on the propagation of White-Rabbit Ethernet frames over the network. This section resumes the work done by Seven Solutions on this framework during EMC2 Y3 period related to WP4.

Development phases:

- Step 1: M12 (end of March 2015)
  - Concept and requirements definition
  - Redundant topologies requirements definition (SW/HW/Gateware)
- Step 2: M24 (end of March 2016)
  - Evaluation of QoS effects in terms of system jitter and wander
  - Time accuracy tests of time-triggering using redundant network topology architectures (based on HSR)
  - First prototype of the HSR event-trigger application for distributed mixed-critical systems using White Rabbit and evaluation of QoS and VLAN support.
- Step 3: M36 (end of March 2017)
  - Final version of the HSR even-trigger distributed application.
  - Time accuracy tests of time-triggering using redundant network topologies (HSR)
  - Final EMC2 demonstrator available to whole consortium.

During Y3 SevenS has been working on the new design of the WR-ZEN Time Provider to improve clocks and thus, clocking stability in terms of jitter and phase noise.

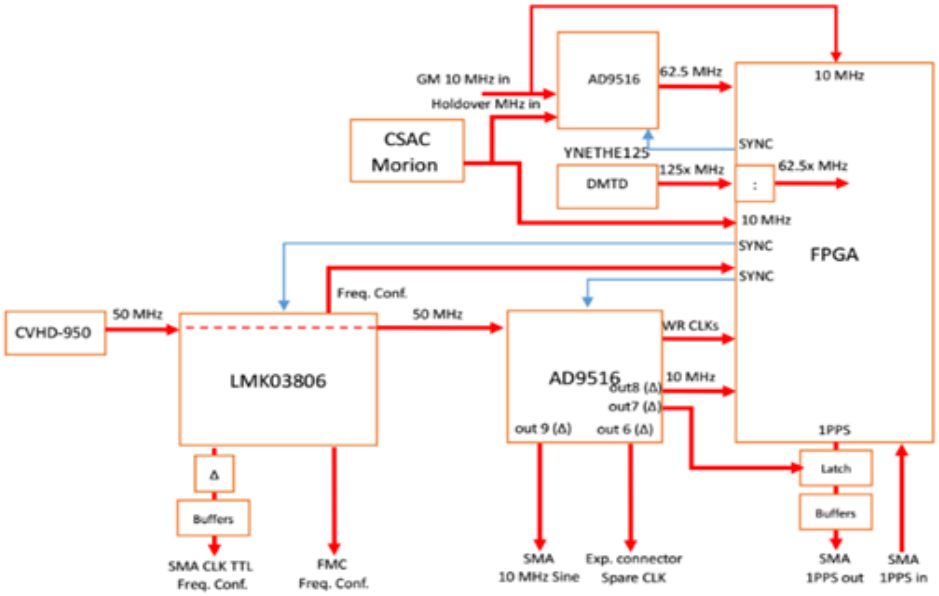


Figure 32 WR-ZEN (Time Provider) new clocking design to improve clock stability.

This figure presents the changes that have been introduced within the new WR-ZEN design. We have implemented a new electric circuit input that isolates the clock references from the design inside the FPGA. A new PLL chip is now outside the FPGA that makes able to switch between two time references in approximately zero-time using hardware. The FPGA will remain synchronized to the clock reference provided by this PLL chip. Next figure shows the comparison between the Grand Master mode for the time provider WR-ZEN (blue) and the WR-Switch (purple) in terms of phase noise for two different ranges [1-30Hz] and [30-100kHz]. The former presents an improvement thanks to the control loop connected to the most stable external reference. The latter shows how the quality of the internal oscillator improves considerably the phase noise results.

Thanks to the utilization of this new design, and the improvements of the internal clocks, we are able avoid single point of failure from a hardware perspective, offering better results that our previous switchover mechanism which was already implemented in software during the previous EMC2 period. The WR-ZEN new design will be also able to provide a backup timing reference for several hours maintaining the accuracy of less than 1 nano-second (holdover mechanism).

In addition, during this Y3 period we have been finishing the development of the High-availability Redundancy Protocol (HSR) for our timing solution White Rabbit (WP1 T.1.6). We have also being working on the implementation of transparent and hybrid clocks for the WR-LEN, which is also part of the WP11 UC5.5 demonstrator: *Synchronized low-latency deterministic networks*. Moreover, implementing transparent clocks eases using White Rabbit in the industrial domain, which is one of the key points for WP11 UC5.5 demonstrator.

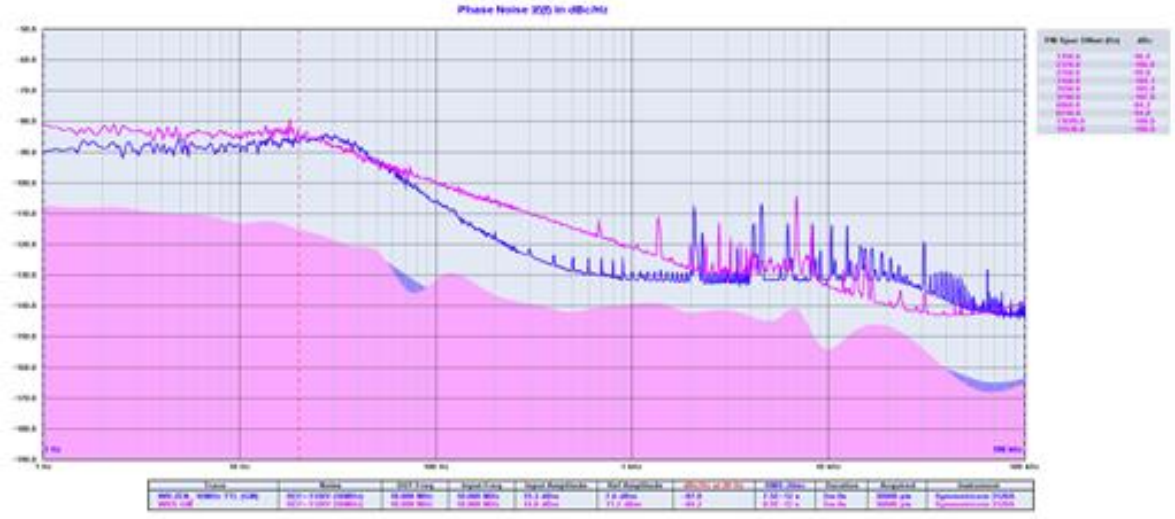


Figure 33 WR-ZEN Phase Noise comparison between WR-ZEN and the WR-Switch clocking system.

Outcome

The integration of HSR developed in WP1 with the notion of critical packets and QoS (VLANs), together with the precise White Rabbit timing features and its results in terms of scalability and 1-PPS jitter stability, are expected to form a base for next-gen time-based systems in different domains, such as Smart Grid and Automotive.

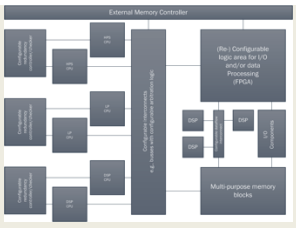
Novel technology in comparison to state-of-the-art:

- It includes White-Rabbit, the most accurate PTP (IEEE-1588) solution for time and frequency distribution.
- It extends standard protocols such as PTP and SyncE for time distribution for high-accurate time-triggered systems.
- Zero-time recovery in case of failure using a HSR implementation (software/firmware).
- New hardware design to improve switchover and holdover using hardware.
- Based on open specifications with basic open software and open hardware reference designs.

4.6 NXP Germany (01R NXPGE)

Technology lanes covered:

- **Heterogeneous Multiprocessor SoC architectures**
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- ***Networking***
- Virtualization and verification technologies



4.6.1 Multi Secure Elements Managing Unit

NXP-GE is responsible for development of a flexible multi-core root of trust solution *(also referred to as Multi Secure Element Managing Unit)* for embedded multi-processor/core, mixed criticality applications in dynamic and changeable real-time environments. The task of such a module is to act as a trust anchor that

identifies & authenticates every communicating node in order to establish a chain of trust in any given network. For this purpose, such root of trust module needs a crypto coprocessor (also called a secure element) that can handle all cryptographic operations/protocols related to identification and authentication process. A secure element is a tamper resistant, certified (like Common Criteria, EMVCo standardization institutes) microcontroller that provides secure/isolated processing environment and secure storage features which finds application in banking cards, passports, transit, insurance cards etc. A single secure element can perform one task and only one type of crypto operation at a time as per the configuration. However, a multi-secure-element-core root of trust (MSECROT) module can process multiple tasks in parallel and support multiple type of crypto protocols simultaneously. Thus, in a network, a MSECROT has faster response time for diverse types security requests and better throughput for higher load scenarios. This is the motivation for developing a MSECROT module.

A typical network with MSECROT can be envisaged as shown in Figure 34

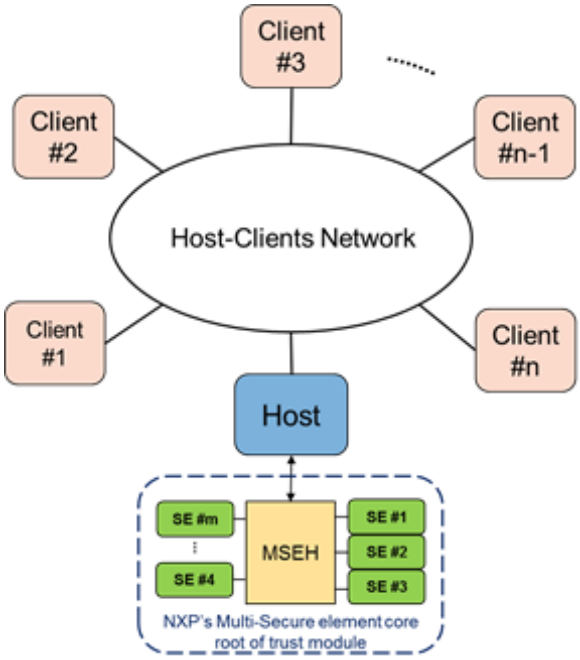


Figure 34 An example Host-clients network with MSECROT

Functionality

It provides security as service to the connected network following the principles of service oriented architecture(SoA). It implements a scale down version of public key infrastructure performing unilateral / mutual authentication of client nodes to a host node or vice versa. Typically, in any network, clients authenticate to a host using some challenge-response schemes (like TLS handshake mechanism) with asymmetric key crypto protocols (like RSA/ECC) which is the same principle employed in MSECROT. It can also negotiate session keys for encrypted message transactions between host and clients (using symmetric key cryptoprotocols like AES). The list of all currently supported functionalities (also called as security specifications) of MSECROT is given in Table 7.

Table 7 Security Specifications or APIs of MSECROT

Crypto protocols	Security APIs	Description
True Random Number Generation	RND_GetRandomNum()	Needed for challenge-response scheme. A random number acts as a challenge sent by a verifier which is signed by private key of prover. The resulting signature is verified by verifier using prover's public key
RSA asymmetric key protocol (Supports 1024/2048 key sizes)	RSA_Sign()	Used for signing a challenge
	RSA_Verify()	For verifying a signature against challenge
	RSA_Encrypt()	Message encryption
	RSA_Decrypt()	Message decryption
ECC (elliptical curve cryptography) asymmetric key protocol (Supported Curves are NIST192, NIST224, NIST256, BrainPoolP192r1, BrainPoolP224r1, BrainPoolP256r1)	ECC_Sign()	Used for signing a challenge
	ECC_Verify()	For verifying a signature against challenge
Elliptical curve Diffie Hellmann	ECDH_GenerateEphemeralKeyPair()	Prior to session key creation between client and host, an ephemeral public/private key pair is generated by both and respective public key is exchanged insecurely on the network
	ECDH_GenerateSharedSecret()	After public key exchange, a common shared secret for symmetric key protocol is generated
AES (Supported modes are ECB, GCM, CBC, GMAC with key sizes 128,192 and 256. Key wrapping supported)	AES_Encrypt()	Message encryption
	AES_Decrypt()	Message decryption
General purpose secure storage	SetParam()	Save data (certificates, master keys)
	GetParam()	Retrieve data
	EraseParam()	Delete data

Hardware Architecture

The multi-core root of trust module is composed of central ARM controller (NXP ARM Cortex M3 LPC1769) interfaced with multiple secure elements (NXP A7001) on board using I2C as shown in Figure 35. The ARM controller is called multi-core secure element handler (MSEH) as it is responsible for initialization & configuration of secure elements, scheduling, arbitration & load distribution of security tasks assigned by host and provide multiple communication interfaces to host (like USB, Ethernet, SPI, UART etc).

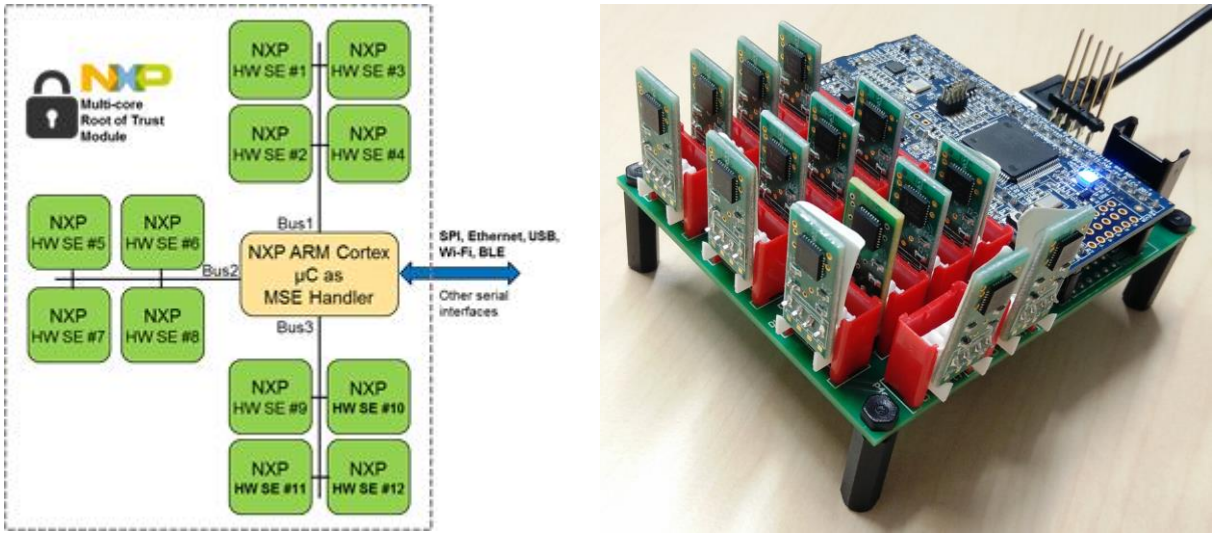


Figure 35 NXP Multi-secure-element core root of trust module

Assessment of MSECROT with respect to EMC2 requirements in WP10

The requirements for such a module as defined by EMC2 consortium is as described in Table 8

Table 8 Requirements of a multi-core root of trust

Requirement ID	Category	Short description
01R_NXP_001	Networked security	To adapt the security system to the UC identification and authentication in a multicore environment it needs to be connected to a non-linear decentralized bus system; the interface will be defined with a layered approach to make the hardware layer interchangeable
01R_NXP_002	Security counter parts	each multi-core node needs to be equipped at least with a secure element as trust anchor
01R_NXP_003	needed security levels	the needed level of security of the whole system as well as necessary differentiations in terms of required security need to be defined
01R_NXP_004	Parallel Processing	Scalability of the security system must not rely on an increase of the performance of single processing elements but on adding additional elements
01R_NXP_005	Runtime certification of	provide methods for the runtime evaluation of trust certificates and initiation of an asymmetrically encrypted communication;

	cyber-physical systems	
01R_NXP_006	Identification	Each part of the system, e.g. a service or communication channel, should be identified by a unique name; for low cost security needs symmetric encryption via device pairing shall be considered as an optional aspect;

EMC2 objective

It fulfills OBJ63 (i.e. Security as a service) that states that "Hybrid solutions as combinations multiple secure elements and managing processors to provide flexible roots of trust". Although the KPIs associated with OBJ63 (i.e. KPI48- KPI53) aren't directly relevant to MSECROT, it still accomplishes some of it in an indirect fashion. It has better latency (KPI49), efficiency (KPI52) and bandwidth utilization (KPI53) in comparison to single secure core solution. The KPIs of OBJ63 are however more relevant to industrial manufacturing and logistics which is the main use case for the living lab work packages.

Other WPs

MSECROT employs concepts and technologies developed in other work packages of EMC2. Although there is no direct technology transfer from other WPs to MSECROT, it based on similar principles. A table (see Table 9) describing the cross linkage of MSECROT features to the other WP technologies illustrates the connection in detail.

Table 9 MSECROT features linkage to other WPs of EMC2

Technology	**Respective Work package name**	**Description of linkage in MSECROT**
System and service level security (T1.5)	WP1	- It can provide different kinds of authentication and identification services to its network based on agreed service levels & desired Quality of service - Unilateral /bilateral authentication services are possible - Multiple configurable cryptographic routines for handshake (like ECC, RSA) and encryption/decryption (AES in ECB, CBC, GMAC etc.) services possible
Communication services (T3.2) Networking (T4.3)	WP3 WP4	It supports multiple communication protocols for interfacing to the network. It could be Ethernet, USB, SPI, I2C, UART etc. This gives the system architect the flexibility to select any one of the above the communication mode for interfacing to the rest of the system
Virtualization and isolation (T3.3)	WP3	- It is composed of multiple secure elements working either in parallel or tandem based on configuration to provide security as a service to the network. It can act as single virtual trust anchor of the system that distributes its security load amongst the underlying cores based on priority or service levels or configurations. - Each core can provide an isolated execution environment for a request and thus service as many requests in parallel as the number of cores

Security mechanisms and services (T3.4)	WP3	The individual secure elements have inbuilt security mechanisms to counter physical, side channel and other types of attacks. They have tamper resistant memory units to safeguard the critical data of the system and provide a secure execution environment. Through cryptographic handshake mechanisms they can protect the communication channel between them and client requesting their services.
Heterogeneous Multiprocessor SoC architectures (T4.1)	WP4	Each core can provide a different set of security services compared to its neighbor. Thus, forming a heterogeneous multi-processing architecture. As mentioned before each core can be configured to run different types of cryptographic routines and can also be reconfigured at run time dynamically based on the load and desired quality of service
Verification and Validation Techniques (T4.4)	WP4	Individual secure element cores are common criteria certified (EAL6+), approved by EMVCo, BSI and other international certification authorities. The multi-core root of trust solution is composed of these certified secure elements as individual cores and the central handler simply distributes the incoming security traffic to these cores and play no role in security activities such as handling public/private keys, performing crypto operations etc. Hence the overall system can be considered as certified interconnect of security cores. The validation & verification process of individual secure cores can be extended to multicore root of trust solutions too

Final Demo of MSECROT

A final demo is planned together with WP10.3 which deals with heterogeneous tracking module (Multiple location / position sensors are connected to a central Linux gateway module) from Ambar Telecommunications. The use case to be demonstrated in the final demo is unilateral authentication of tracking sensor nodes to central Linux gateway. The gateway or the host forwards all security requests from tracking nodes to MSECROT and sends its response back to sensor nodes. The host is connected to MSECROT via SPI. A block diagram of the complete system is shown in Figure 36.

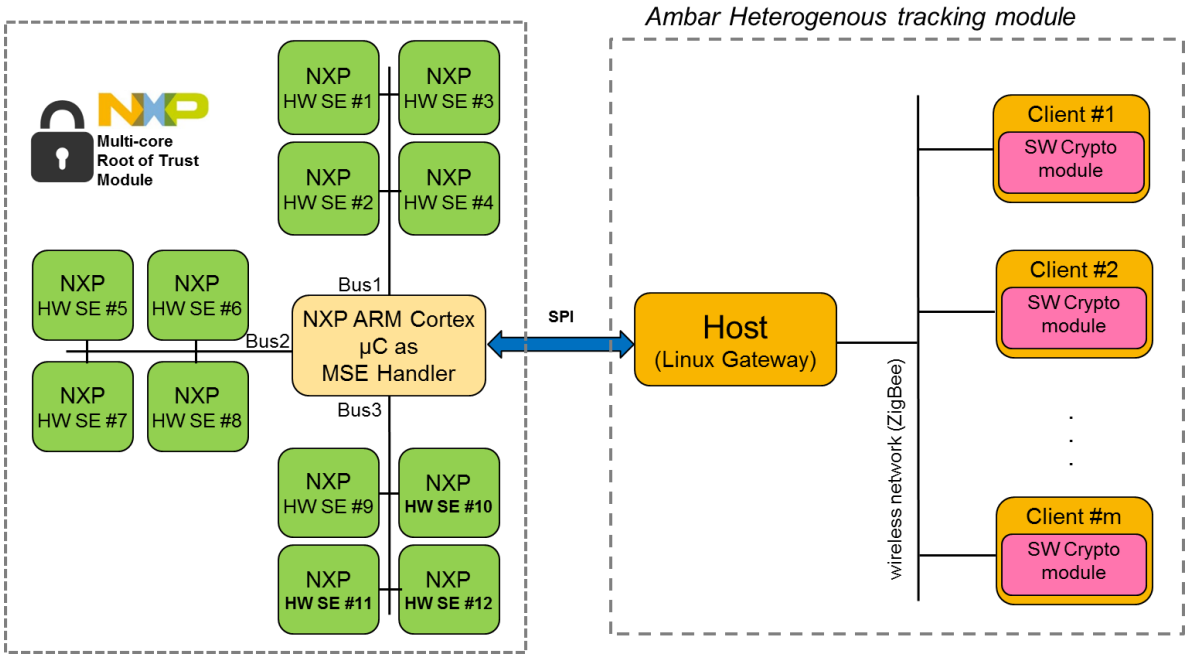


Figure 36 Block diagram of MSECROT with AMBAR Tracking module

The unilateral authentication of sensor nodes with host is done using RSA 1024 crypto protocol. A representation of authentication process is shown in Figure 37.

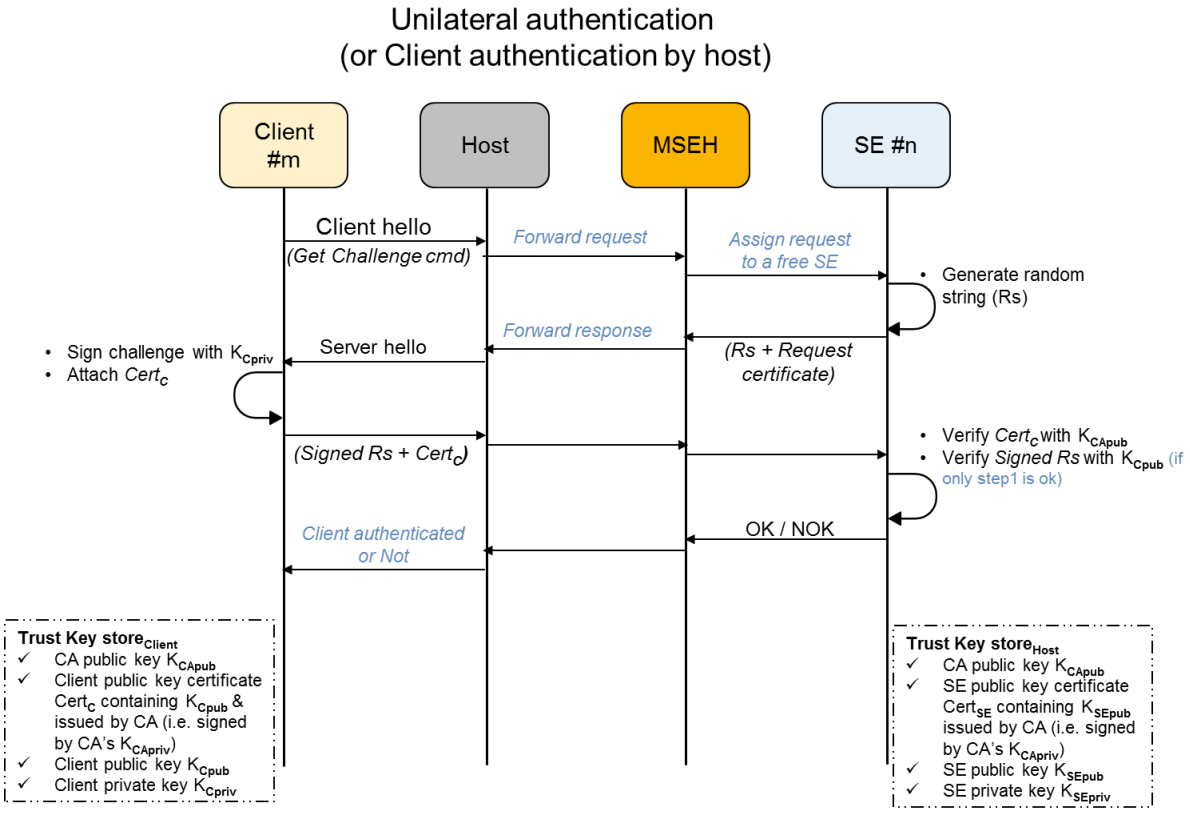


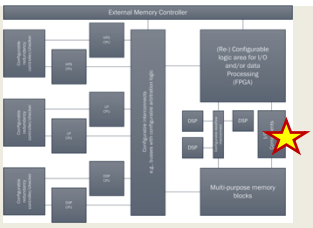
Figure 37 Unilateral authentication process

Since sensor nodes are remotely located and battery powered (most of the times), it often lacks computation power and resources to do complex PKI authentication processes. Hence a simplified authentication protocol without a central certificate authority is implemented in the demo.

4.7 Infineon Technologies AG (01A IFAG)

Technology lanes covered:

- Heterogeneous Multiprocessor SoC architectures
- Dynamic reconfiguration on HW accelerators and reconfigurable logic
- **Networking**
- Virtualization and verification technologies



4.7.1 High speed inter-chip communication

IFAG Munich develops a high speed serial link peripheral for inter chip communication.

It provides point-to-point communication of both single data values and of large data blocks, called streams. The communicating devices can be complex microcontrollers, or a microcontroller and a device with only basic execution capabilities. There are four channels to transfer single values to/from target. They support

direct writing of 8/16/32 bit data from the initiator into a target's register, as well as reading a value from a target, performed by a modules internal master on the target side. For transferring large data blocks there is a channel containing FIFOs. The module implements Transport Layer tasks and hands over the data to another module which provides Data Link Layer and Physical Layer services, data serialization and transmission. In multi slave use-cases, one master can communicate with up to three slaves. All transfers are protected by safety features like CRC and timeout.

Development phases:

- M1-12: Specification
- M12-24: VHDL Implementation and pre-silicon verification
- M24-36: Physical implementation, production and post-silicon verification

Outcome

The outcome of this project will be a peripheral IP module implemented on a microcontroller chip with proven functionality in a multi-slave environment. A validation and measurement report will be provided as a final delivery.

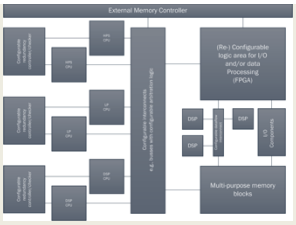
Y3 achievements

- The peripheral IP module has been implemented in a first design step of a new microcontroller family
- Silicon production has been completed and samples are available
- Post silicon verification and measurements are almost completed and confirmed the correct functionality according to the specification
- Single and multi-slave communication is working as intended
- Preparation of validation and measurement report has been started

4.8 TU Dortmund (01W TUDO)

Technology lanes covered:

- Heterogeneous Multiprocessor SoC architectures
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- *Networking*
- Virtualization and verification technologies



4.8.1 Peripheral Virtualization Manager

Both timing and spatial separation of peripherals is necessary for integrating applications of mixed-criticaliy on a single platform. To this end, we have extended the SoCRocket platform with a virtualization manager that assigns resources based on criticality domains.

Since our model assumes memory-mapped peripherals, all peripheral accesses (whether programmable I/O or DMA) are chanelled through an AMBA AHB bus. Timing separation is achieved by a bus priority arbitration algorithm that grants access on a task-level. This offers greater flexibility than core-level arbitration, which statically assigns critical tasks to specific cores. Within a given criticality, the algorithm falls back to round-robin. Currently, we are working on fine-tuning the granularity of the assigned time slot for reducing the arbitration overhead and optimizing results.

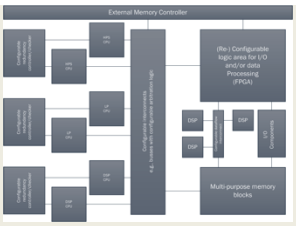
Spatial separation is achieved by banking the peripheral control registers. The consequence is twofold: First, the access rights to a peripheral can be assigned according to the security requirements of each accessing task. Second, this guarantees freedom of interference between all tasks independently of the task

access rights. Currently, we have defined the interface and implemented the functionality in one test peripheral. In the upcoming months, we aim to extend all SoCRocket peripherals to support the new interface to enable more accurate measurements.

4.9 Tecnalia (15E Tecnalia)

Technology lanes covered:

- Heterogeneous Multiprocessor SoC architectures
- ***Dynamic reconfiguration on HW accelerators and reconfigurable logic***
- *Networking*
- Virtualization and verification technologies



4.9.1 Reliable and Shelf-Healing Dynamic Reconfiguration Manager

FPGA devices are present in today's embedded systems being used for a wide variety of purposes from implementation of peripheral components, processors or co-processors to full multi-processor systems with several processors, peripherals, hardware accelerators and the interconnection network inside one device. In this context, if FPGA technology provides the flexibility of modifying a design by re-programming without going through re-fabrication, Partial Reconfiguration (PR) takes this flexibility one step further. PR is the ability to dynamically modify blocks of logic while the remaining logic continues to operate without interruption. It allows designers to change functionality on the fly, eliminating the need to fully reconfigure and re-establish links. It makes possible, for example, the implementation of dynamically reconfigurable peripherals.

When FPGA partial reconfiguration is used to dynamically change, add or remove peripherals in mixed-criticality systems, a reliable methodology has the following requirements:

- Include a Dynamic Reconfiguration Manager (DRM) in the FPGA static partition. This DRM will control the programming of the reconfigurable partitions where the dynamically reconfigurable peripherals will be implemented.
- Add fault-tolerance elements and design techniques to make the DRM reliable and shelf-healing.
- Include an external agent to control and monitor the DRM.

The Reliable and Self-Healing Dynamic Reconfiguration Manager (DRM) is the solution proposed by TASE and TECNALIA in order to perform partial reconfiguration of a Xilinx Virtex-5QV FPGA in a safe (reliable) way in the space environment. This solution has been designed in the context of WP4, while its robustness is evaluated in WP9.

This DRM consists in a MicroBlaze embedded system implemented in the static (that is, not reconfigurable) area of the Virtex-5 device with capability to manage the reprogramming process of several reconfigurable partitions. The DRM is not the main system processor, but it can be considered as a co-processor or a specific peripheral of the full system. One of the main elements of this sub-system is the controller to access the FPGA configuration memory at run time (HW ICAP). This capability to access the FPGA configuration memory is used for a double purpose: first, to perform dynamic reconfiguration of the reconfigurable partitions; second, to perform configuration memory scrubbing in order to prevent SEU accumulation in the FPGA configuration memory.

By the end of the second project year, a basic DRM, that is, a DRM without fault-tolerance elements, was completed and the outcome was shown in a demonstrator within the Space Living Lab (WP9). The demonstrator platform is the LADAP (L3SoC and Data Processing) board developed by TASE. The two main elements of this hardware platform are: the already mentioned Xilinx Virtex-5 device, and a Microsemi ProASIC FPGA that includes the main processor which is a LEON core. This LEON processor behaves as the external agent that controls and monitors the DRM. Figure 38 shows the architecture block diagram of this basic DRM integrated in the LADAP board. At this point, the communication interface

between the DRM and the external controller was implemented at physical level and partially implemented at protocol level.

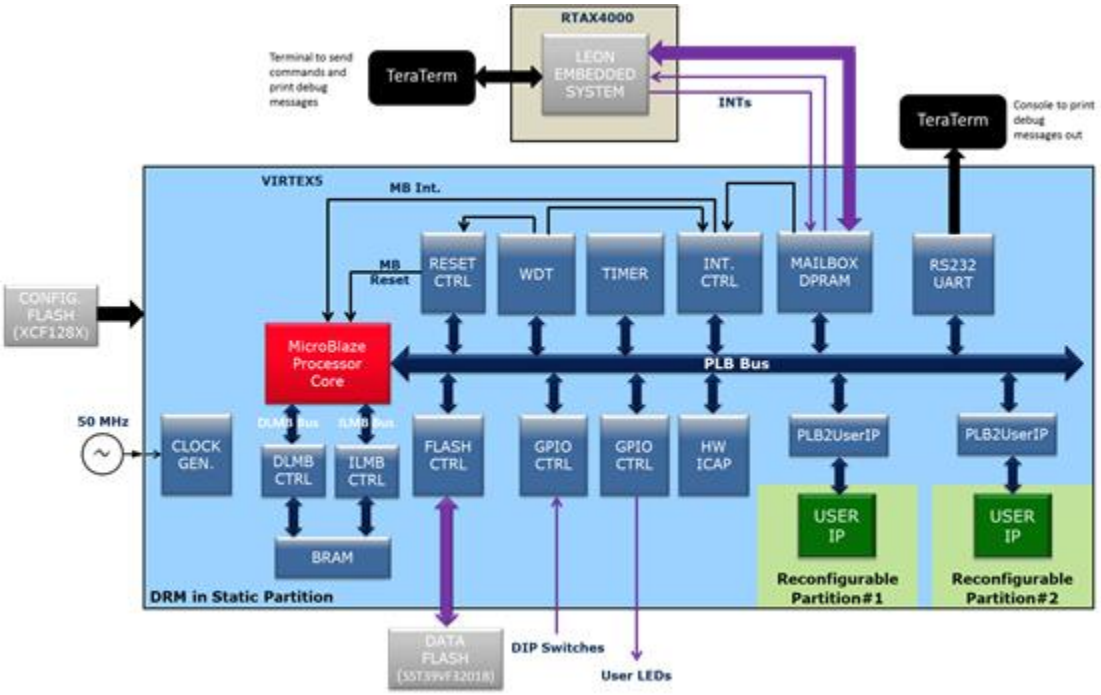


Figure 38 System architecture diagram with the basic DRM

In the third project year, the already specified fault-tolerant features have been implemented in order to make the DRM reliable and shelf-healing. These fault-tolerance elements and design techniques, which were described in detail in Section 3.4 of D4.8, are:

- FPGA configuration memory management
- SEFI detection and mitigation
- MicroBlaze softcore protection
- Partial reconfiguration data integrity check
- EDAC (Error Detection And Correction) circuit of the FPGA internal block RAMs and SET filters of the Virtex-5QV CLB flip-flops enabled

The final architecture block diagram of the Reliable and Self-Healing DRM is shown in Figure 39. A detailed description of the final DRM hardware design was included in section 3.1 of D4.9. The final DRM software and the final DRM-LEON communication protocol were described in detailed in section 2.1 of D9.5. The validation tests are currently being completed. The final results and their evaluation will be included in the final deliverables D4.10 and D9.6. In order to test the DRM shelf-healing capability, a function to insert errors in the FPGA configuration memory has been developed. In this way we simulate the effect of SEUs (Single Event Upsets) due to radiation.

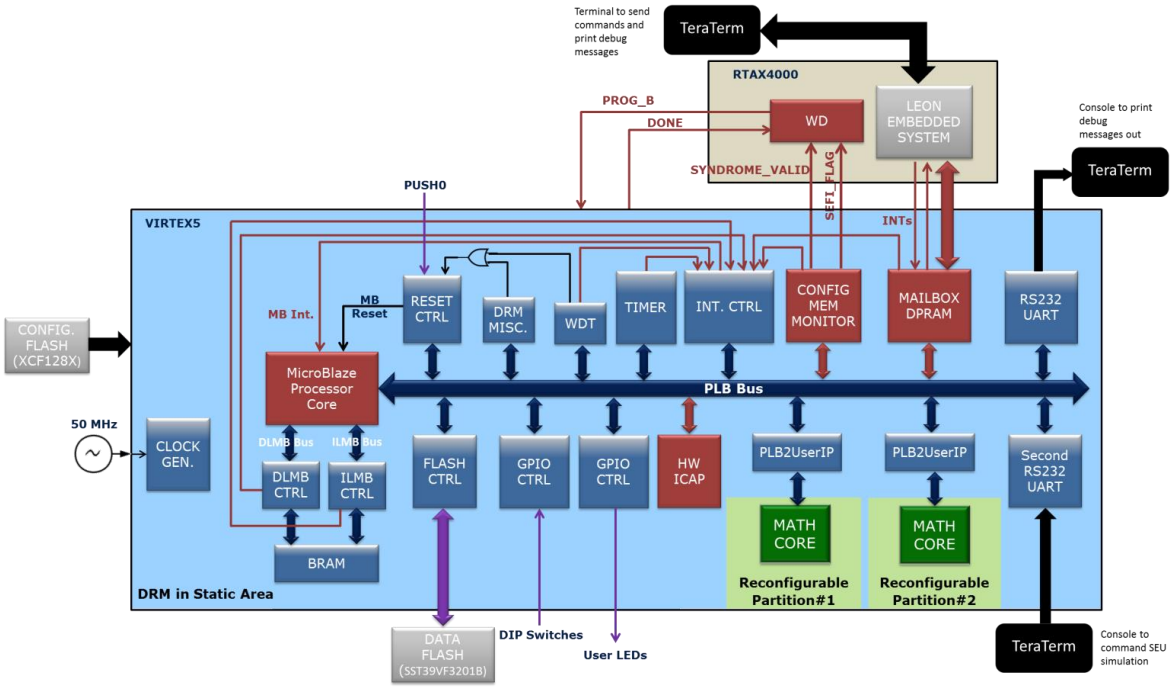
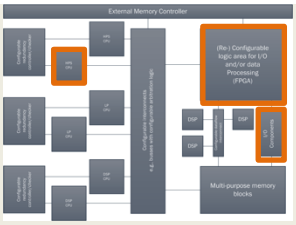


Figure 39 System architecture diagram of the Reliable and Self-Healing DRM

4.10 TTTech Computertechnik AG (02D TTT)

Technology lanes covered:

- *Heterogeneous Multiprocessor SoC architectures*
- *Dynamic reconfiguration on HW accelerators and reconfigurable logic*
- ***Networking***
- *Virtualization and verification technologies*



4.10.1 TTEthernet WireLess (WL)

The following contribution uses results and explanations published in [7]. The publication is based on the research results TTTech conducted in a cooperation with the Mälardalen University Västerås, Sweden.

TTEthernet is standardized according to a SAE standard SAE 6802. Generally, TTEthernet was initially intended for backbone networks, supporting three different traffic classes: Time Triggered Ethernet traffic, rate constraint traffic (like ARINC 664 part 7) or standard Ethernet traffic like anybody knows from PCs or laptop. This allows to integrate several traffic flows used by applications with mixed critically communication requirements. Traffic prioritization is imposed between the flows such that critical delay-sensitive traffic is completely deterministic, while low priority traffic can suffer from starvation. The highest priority traffic flow is the time-triggered (TT) traffic class, intended for time-critical communication, with predefined instants in time when a message must be sent and received. The next priority level is given to the rate-constrained (RC) traffic class, which supports traffic flows that are characterized by an average bandwidth requirement, usually employed for data streaming. Finally, the best-effort (BE) traffic class is for asynchronous traffic (legacy Ethernet traffic) and is sent without any delivery guarantee whenever no TT or RC packets are transmitted. The support of flows with different requirements under the same infrastructure is of great value for industry, since emerging industrial control applications need to combine periodic traffic for monitoring and control with event-driven traffic from autonomous or mobile nodes, if possible using standard-based technologies.

Due to the increasing need to connect even safety-relevant applications like i.e. automated driving to internet based service providers and networks, the demand for a wireless connectivity rose significantly. This effect became even more essential, when such services started to require to interfere with the safety-relevant functions and applications and even start to become essential in safety-relevant control loops.

A key aspect of any wireless solution with industrial requirements is the media access control (MAC) protocol. In essence, it is responsible for sharing the medium among users. To give support to traffic flows from time-critical applications, deterministic medium access is required. This means that any device that tries to access the medium will have a guarantee in terms of an upper-bounded channel access delay. Furthermore, the jitter, i.e., the difference between the minimum and maximum access delay is also a requirement for periodic traffic.

We propose three different MAC methods with deterministic channel access delay with support for the three different traffic classes of TTEthernet namely TT, RC and BE. We aim for a solution that is implementable on top of standardized protocols such as IEEE 802.11 and IEEE 802.15.4 without requiring any changes to the hardware.

Due to the requirement of deterministic access to the wireless medium, we will use a centralized network topology with an access point (AP). All messages must go through the AP which can also connect to a potential wired segment. In addition, the selected infrastructure network topology is thereby similar to e.g., the wired TTEthernet switch topology. We can guarantee deterministic access to the wireless medium if we propose a MAC mechanism that has an upper-bounded channel access delay. Based on the outlined evaluation criteria, we have opted for a TDMA-based approach. The resource that the wireless scheduler will manage is then timeslots. Dividing periods of time into slots fits easily with the time-triggered paradigm and the requirement of deterministic access to the medium. The actual allocation of the three different TTEthernet traffic flows to those time-slots opens a range of possibilities with different implications, from which three of them are selected. In all of them, the TT traffic class will be scheduled offline, guaranteeing its deterministic access to the wireless medium. In the worst case, RC flows are periodic, so they can be modelled as TT flows and therefore added to the scheduler offline. Based on the individual periods of the TT and RC traffic, a hyper-period, being the least common denominator of the individual periods, can be defined. In the case of BE, the packets are just placed in the internal queues of the nodes by the user applications at runtime, without any predefined traffic pattern. Their characteristics are not known at the moment of creating the schedule, so it is not possible to assign time-slots to specific frames. As a consequence, its access will not be deterministic, but still this is consistent with the definition of BE traffic in TTEthernet. The schedule will be repeated continuously during the system run-time. It is important to notice that the receiver in a slot does not need to be specified for the wireless schedule due to its broadcast nature. The only time it could be good to schedule a receiver is to preserve energy, but this is out of the scope of the considerations contained here-in. Also, by allowing more than one receiver, the future implementation of steps to increase reliability is left open.

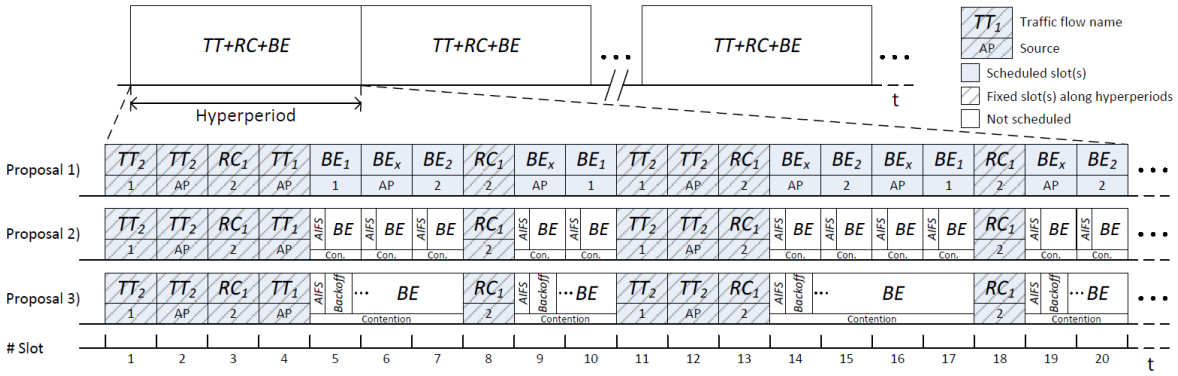


Figure 40 Example of slot allocation for the tree MAC approaches

The three approaches are:

1) Pre scheduled time slots only
2) Pre-scheduled time-slots and contention-based timeslots
3) Pre-scheduled time-slots and contention-based phase

Ad 1: Pre-scheduled time-slots only: This approach does an offline allocation of time-slots for all three types of traffic and all transmitters (i.e., in each slot, there is one unique transmitter and one unique traffic class allowed). The specific allocation of slots for TT and RC will be given by the scheduler. Next, BE traffic will be placed on the remaining slots after the allocation of TT and RC traffic. As BE traffic characteristics are not known in advance, the remaining slots cannot be directly assigned, so the fairest way of sharing these resources is using a round-robin schedule, giving the same amount of timeslots to each node in a circular order.

Ad 2 Pre-scheduled time-slots and contention-based timeslots: This approach is similar to approach 1, but the remaining time-slots not used by TT and RC can be used for BE traffic flows following a contention process. Further, one single node in each slot is given a higher priority, using a higher priority class in IEEE 802.11e, yielding a shorter T_{AIFS}. Thereby prioritized access is given to nodes in a round-robin fashion. However, using CSMA-based contention, all nodes waiting exactly for an AIFS in the beginning of a slot is not recommended, as when more than one node wants to transmit in a slot, they will collide when starting to transmit. Therefore, we will add a random time in addition to the AIFS to help to avoid collisions. However, the high priority node will not add any back-off.

Ad 3 Pre-scheduled time-slots and contention-based phase: Here the proposed mechanism establishes a contention-based continuous phase for BE traffic, and a scheduled phase for TT and RC traffic. The ratio of the contention-based phase and the scheduled phase can be determined using schedulability analysis [9]. The two extreme cases are that all TT and RC traffic is grouped together in one long scheduled phase, leaving the reminder of the hyper-period to the contention based phase, or alternatively, that each TT and RC instance is evenly spread in the hyper-period, leaving the space in between for BE traffic. During a contention phase, access is granted after waiting a AIFS plus a random back-off value. The benefit of having the scheduled phases evenly spread over the hyper-frame, is that the best-case channel access delay is reduced for BE. The drawback is that, as the contention phase size is multiple of a scheduled slot size (to facilitate scheduling), the spaces between scheduled slots may not be big enough to accommodate the time it takes for contention and transmission. This MAC approach is similar to IEEE 802.15.4 GTS, but with the advantage that slots are guaranteed to be booked as they are not reserved during a contention phase.

5. Conclusions

The above presented document and its individual parts provide extensive insight in to activities performed by partners in WP4, tasks T4.2 and T4.3. The technology reediness levels vary in each individual case however they full spectrum from theoretical concepts to full functional prototypes. It is notable to mention that although highly individual the technologies in the document share number of typical characteristics making them potentially compatible for future applications or combined hardware platforms. They also cover different aspects from safety and security, to performance and interoperability. The above described technologies are used or examined by other work packages in the project. It is another confirmation that the work presented can used in a real life applications.

6. References

[1] S. Kerrison, D. May and K. Eder, "A Benes Based NoC Switching Architecture for Mixed Criticality Embedded Systems," in *IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC 2016)*, Lyon, France, 2016.

[2] B. G. a. S. W. W. J. Hestness, "Netrace: Dependency-Driven, Trace-Based Network-on-Chip Simulation," in *3rd International Workshop on Network on Chip Architectures (NoCArc)*, 2010.

[3] e. a. Boris Moturk, "Power Monitoring for Mixed-Criticality on a Many-Core Platform".

[4] e. a. Boris Motruk, "Safe Virtual Interrupts Leveraging Distributed Shared Resources and Core-to-Core Communication on Many-Core Platforms".

[5] e. a. Boris Moturk, "IDAMC: A Many-Core Platform with Run-Time Monitoring for Mixed-Criticality".

[6] "Xilinx, ZC706 Evaluation Board fro the Zynq-7000 XC7Z045 All Programmable SoC - User Guide," [Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf.

[7] R. G. Haris Isakovic, "A heterogenous time-triggered on a hybrid system-on-a-chip platform," in *ISIE 2016*, Santa Clara, CA, USA, 2016.

[8] UTIA, "EdkDSP," 2017. [Online]. Available: http://sp.utia.cz/index.php?ids=results&id=s30i1hm4, http://sp.utia.cz/results/s30i1hm4/s30i1hm4_2015_4.pdf.

[9] M. a. E. U. A.Bohm, Performance comparison of a platooning application using IEEE 802.11p MAC on the control channel and a centralized MAC on a service channel, Lyon, France: in Poc. IEEE WiMob, 2013, pp 545-552.

[10] D. A. K. A. M. S. T. G. S. C. S. Ioannis Sourdis, "Resilient CMPs with Mixed-grained Reconfigurability," in *IEEE Micro magazine , Volume 36, Issue 1, special series on "Harsh Chips"*, 2016.

7. Abbreviations

Table 10: Abbreviations

Abbreviation	Meaning
µC	Micro-Controller
AMP	Asymmetric Multi-processing
AMSPS	Analog-Mixed-Signal Power System
AVFS	Adaptive Voltage Frequency Scaling
EMC^2	Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments
HSR	High-availability Seamless Redundancy
IDAMC	Integrated Dependable Architecture for Many Cores
MPSoC	Multiple-processor System-on-chip
PRP	Parallel Redundancy Protocol
PTP	Precision Time Protocol
ToF	Time of Flight
TTNoC	Time Triggered Network-on-Chip
xCU	Electronic control unit in a car
DPR	Dynamic Partial Reconfiguration
DAL	Development Assurance Level
MCMC	Multi-core systems for mixed criticality applications
MCP	Multi-core Processor