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

Project Acronym:

EMC^2

Grant agreement no: 621429

Deliverable no. and title	**D4.4 – Revised report on Cores, Memory, Virtualisation, Accelerators, Interconnecting Cores, Chips, and Peripheral devices in Mixed-Criticality Systems: Results from the First 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	24/03/2015	
Status	Final	
Dissemination level		

Authors

Partici-pant no.	Part. short name	Author name	Chapter(s)
01A	IFAG	Wolfgang Schneidt Jens Harnisch	3.2, 3.5,4.9
01R	NXPGE	Henning Moeller	4.8, 3.8
01V	TUBS	Rolf Mayer	3.3
01W	TUDO	Lilian Tadros	4.10
02A	AVL	Peter Priller	4.6
02D	TTT	Andres Eckel	4.5, 3.4
02E	TUW	Haris Isakovic	4.3
04D	UTIA	Zdenek Pohl	4.4
02C	IFAT	Alexander Lipautz Norbert Druml	3.2
11D	Selex	Massimo Traversone	3.10
15E	Tecnalia	Alonso Muñoz, Asier	4.11
15O	SevenS	José Luis Gutiérrez	4.7
16A	Chalmers	Ioannis Surdis	4.1, 3.1
17B	Imec-NL	Tobias Gemmeke	3.9
17C	NXPNL	Hamed Fatemi	3.11
18H	UoBr	Steve Kerrison	4.2
15F	TASE	Manuel Sanchez Renedo	3.6

Document History

Version	Date	Author name	Reason
v0.1	21/01/2016	Haris Isakovic	Draft
v0.2	10/02/2016	Haris Isakovic	Draft
v0.3	12/02/2016	Haris Isakovic	Draft
v0.6	20/02/2016	Haris Isakovic	Draft
v0.7	16/03/2016	Haris Isakovic	Final Draft
v0.8	24/03/2016	Haris Isakovic	Final Draft
v1.0	24/03/2016	Haris Isakovic	Final
	31/03/2016	Alfred Hoess	Final editing and formatting, deliverable submission

Contents

List of figures

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 steps and activities of partners performed in the second review period of the project. It considers design and implementation phases of the technology development process.

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 partitioned on individual activities of each partner. Next section provides the report from the task T4.3 divided in the same way as the previous section. Next section provides a short disclaimer on changes performed by partners in technical or administrative aspect of the project. Last section concludes the document with an overview of the topics represented in the document.

2. Publishable Executive Summary

This document is a revised report on the activities performed by the partners of the WP4, ARTEMIS project EMC2, involved in tasks T4.2 and T4.3. 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.

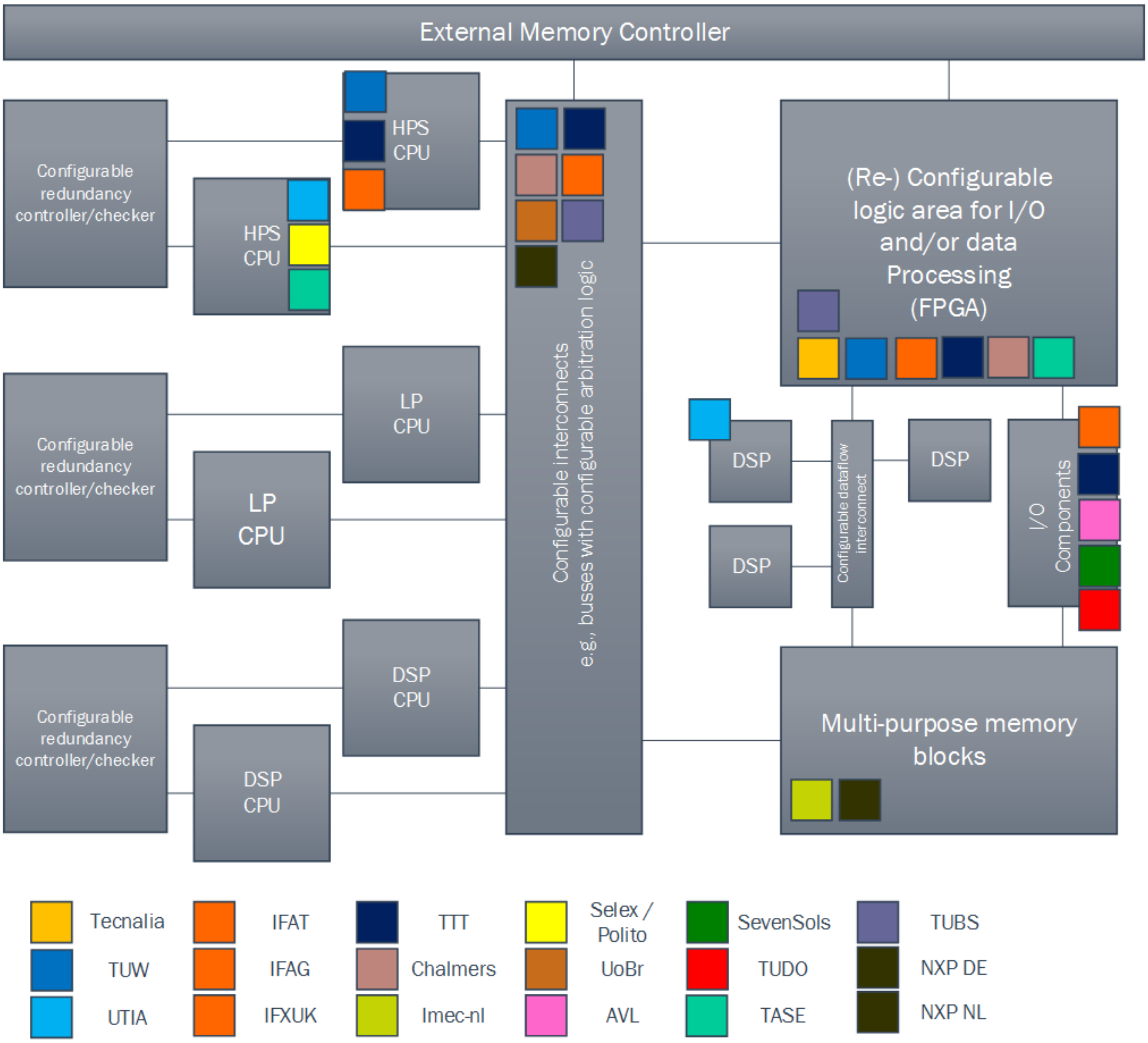

External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks
Tecnalia
TUW
UTIA
IFAT
IFAG
IFXUK
TTT
Chalmers
Imec-nl
Selex / Polito
UoBr
AVL
SevenSols
TUDO
TASE
TUBS
NXP DE
NXP NL

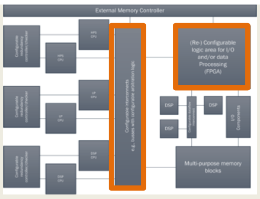
Figure 1: A generic hardware architecture and a partner activity by sector

3. T4.2 Advanced processor concepts and architecture definition

3.1 Chalmers (16A Chalmers)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
Configurable dataflow interconnect
DSP
I/O Components
Multi-purpose memory blocks

3.1.1 Reconfigurable RISC architecture for tolerating permanent faults

As discussed in D4.3, Chalmers in ST4.2.3 works 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.

We are on track with our plan set in period 1 as we have completed the design and implementation of the technique. In year 3, we will evaluate the performance, area, and power overheads of the implemented reconfigurable RISC processor (a probabilistic analysis for evaluating the fault tolerance of the approach is described in T4.4). More precisely in period 2 the following was performed. We worked on an in-order 32-bit RISC with five pipeline stages: IF, DE, EX, MEM, WB. It has local instruction- and data-memory (32Kbyte each) and a 16-entry register file (RF). We then modified its microarchitecture to address a number of challenging issues for allowing the processor parts to be interchangeable without affecting its functionality. In particular, the micro-architecture needs to allow a variable number of decoupled pipeline stages (its stage composes an substitutable processor part, which is replaced if damaged). Our processor architecture must adapt and operate correctly while remaining oblivious to the number of (extra) stages in its pipeline; that is, reconfigurability needs to be transparent to the application level. This design requirement is imperative for guaranteeing binary compatibility among different processor configurations, avoiding recompilation of the code. The above design requirements call for several micro-architectural changes in the data flow and control flow of the baseline core. Allowing a variable number of stages per processor and eliminating global control directly influences the hazard resolution in a typical five-stage RISC architecture such as the SiMS architecture.

1) *Data-hazard resolution:* As in between the original pipeline stages (in the fault-free configuration) new empty pipeline stages may be inserted when substituting faulty parts, the data-hazard resolution cannot be supported with traditional bypassing mechanisms. Instead, we store the produced, yet uncommitted, results in bypass buffers at the stages they are produced (EX and MEM) and may possibly be reused.
2) *Control-hazard resolution, Global stalls, pipeline flushing:* All three aspects are supported via pipeline flushing in a distributed way. Instructions carry an extra bit, read at the IF stage, indicating the flow they belong to. This bit is compared with a similar instruction-flow bit stored at the EX stage. A mismatch prevents the instruction from being executed. In order to flush the pipeline, the bit stored at the EX stage is inverted (e.g. after a branch mis-prediction identified at the EX stage) and all the subsequent instructions are flushed until the equivalent Instruction-flow bit stored at the IF stage is updated.

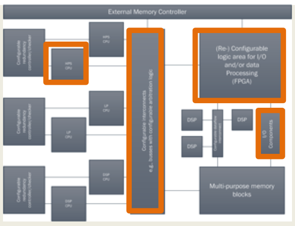
The above were designed and implemented in RTL and verified using various benchmarks, which can be demonstrated standalone.

The proposed approach offers a flexible solution for tolerating permanent faults. Compared to the state of the art, the clock frequency of the design and therefore performance is expected to have a lower overhead when faults occur. This is due to the pipelined interconnects which wide the latency of long wires used for sparing and therefore allow the frequency of our design to scale well despite the configuration used and the replacement of faulty parts.

3.2 Infineon Austria AG, Infineon Technologies AG (02C IFAT, 01A IFAG)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

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, the Time-of-Flight algorithms involved require considerable amounts of computation power. Therefore, novel hardware/software approaches are developed in order to cope with the todays and future challenges.

As described in D4.3, during the first year, requirements and concepts were developed based on various use-case scenarios (e.g., interior monitoring). During the second year, a first demonstrator employing the AURIX automotive controller was implemented. Furthermore, work started on an improved hardware/software co-design concept featuring the Xilinx Zynq platform. During year three, work on the Zynq-based architecture and the use-case scenarios will be intensified.

When developing systems that employ Time-of-Flight 3D imaging, the industrial and scientific challenges are given by the required processing resources for depth-data calculations and computer vision applications. Therefore, the major aim of year two and three is to develop a Zynq-based platform which will feature hardware-accelerated processing algorithms and applications for various use-case scenarios. Thus, a heterogeneous multiprocessor SoC architecture is given. Of particular interest is the automotive use-case scenario interior-monitoring (e.g., driver monitoring, human detection). With the help of the Time-of-Flight 3D imaging technology, which also provides a very robust infrared image during day and night, new and innovative system-approaches will be explored and exploited. As a result, demonstrators have been developed which will showcase the technological advances in Time-of-Flight based applications and systems, and which may be reused in LLs (e.g., WP12.2 video surveillance).

3.2.2 AMSPS-Systems for MCMC integrated Systems

A high level block diagram of the AMSPS-System is attached below. 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.

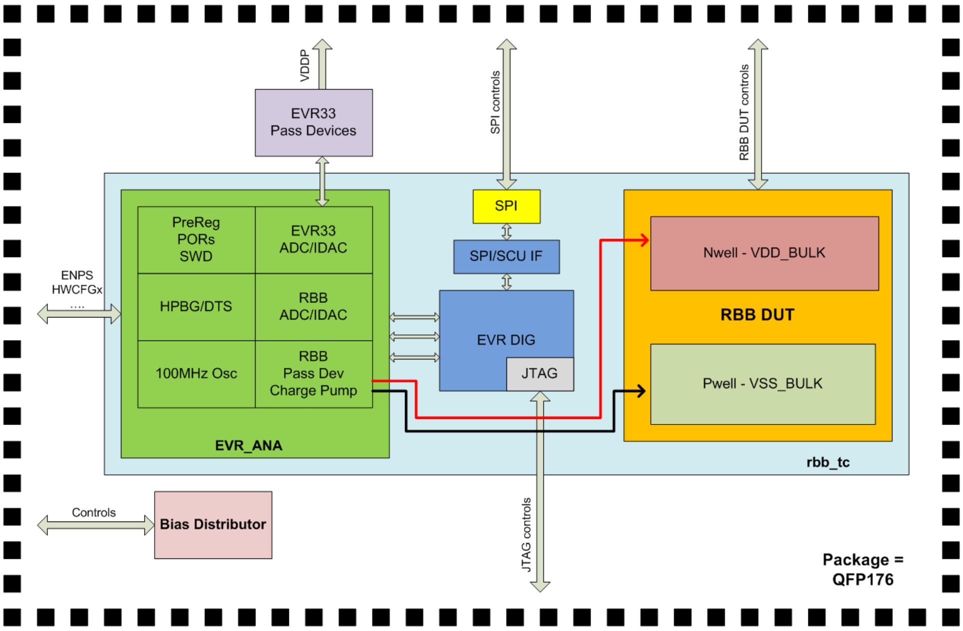


Figure 2: AMSPS-System block diagram

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

- Concept development (performance & functional safety), covered by D4.3 (status: finished)
- System-verification (pre-silicon) & -validation on test chip level (40nm CMOS Technology).
- Definition of application use-cases (status: still ongoing).

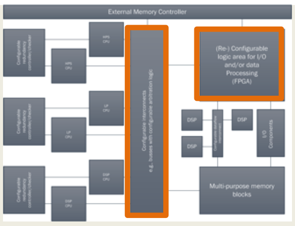
Generally spoken, the major challenge we are facing is to make the AMSPS sufficiently robust for industrial- & automotive applications (e.g. performance, efficiency, and life-time) under consideration of functional safety requirements (e.g. Technical Safety Requirements [TSR] and Safety Mechanism [SM]).

- Detailed plans for the Y3 of the project:
  - 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)
- Plans for 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),
  - Technology transfer to LLs (no transfer).

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***


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
LP CPU
LP CPU
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

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

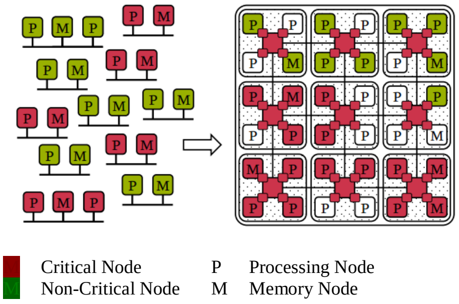


Figure 3: Mixed Critical Tasks Distribution in a NoC

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 [1], [2], [3].

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.

First 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.

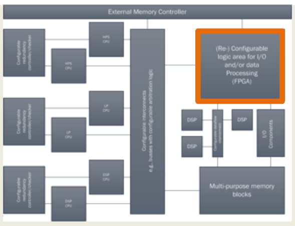
Expected outcome

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.

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*



External Memory Controller
Configurable redundancy controller/checker
HPS CPU
LP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

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 4). 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.

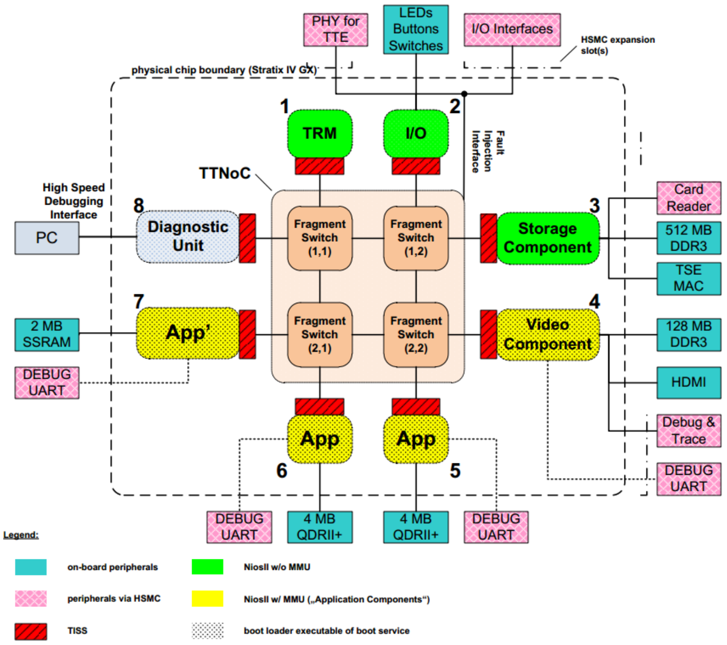


Figure 4: ACROSS architecture implemented on an ALTERA FPGA

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.

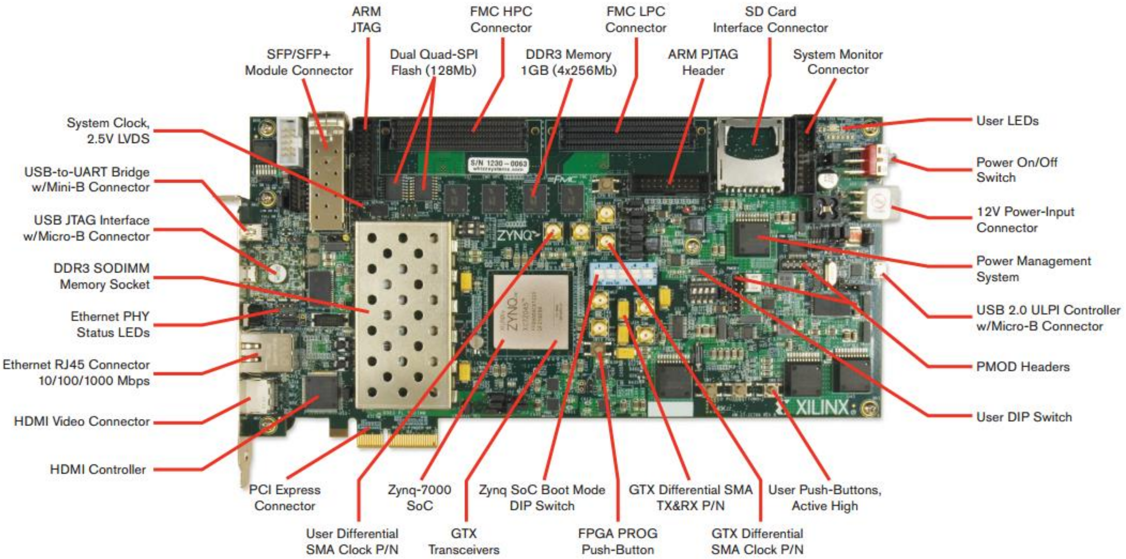

ARM JTAG
FMC HPC Connector
FMC LPC Connector
SD Card Interface Connector
SFP/SFP+ Module Connector
Dual Quad-SPI Flash (128Mb)
DDR3 Memory 1GB (4x256Mb)
ARM PJTAG Header
System Monitor Connector
System Clock, 2.5V LVDS
USB-to-UART Bridge w/Mini-B Connector
USB JTAG Interface w/Micro-B Connector
DDR3 SODIMM Memory Socket
Ethernet PHY Status LEDs
Ethernet RJ45 Connector 10/100/1000 Mbps
HDMI Video Connector
HDMI Controller
User LEDs
Power On/Off Switch
12V Power-Input Connector
Power Management System
USB 2.0 ULPI Controller w/Micro-B Connector
PMOD Headers
User DIP Switch
PCI Express Connector
Zynq-7000 SoC
Zynq SoC Boot Mode DIP Switch
GTX Differential SMA TX&RX P/N
User Push-Buttons, Active High
User Differential SMA Clock P/N
GTX Transceivers
FPGA PROG Push-Button
GTX Differential SMA Clock P/N

Figure 5: Xilinx ZYNQ ZC706 evaluation board with ZYNQ 7000 SoC [1]

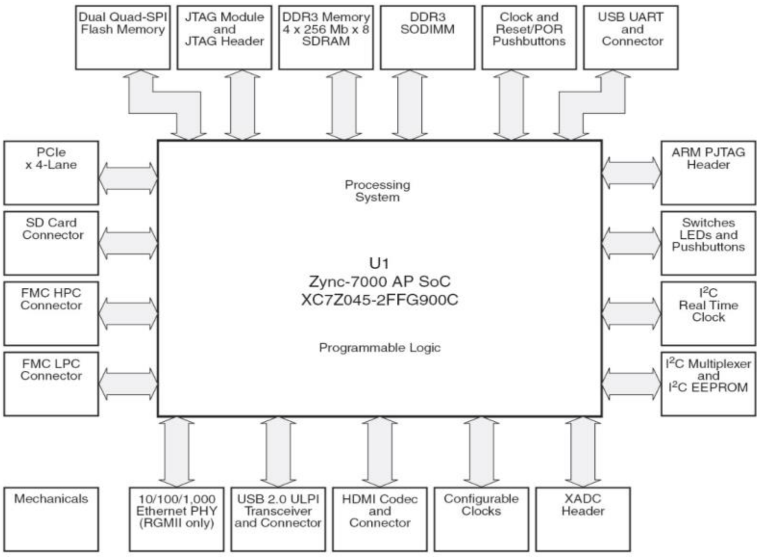


Figure 6: ZC706 Evaluation Board Block Diagram [1]

Information on the evaluation board architecture and components is provided in Figure 5, Figure 6, and Figure 7. 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.

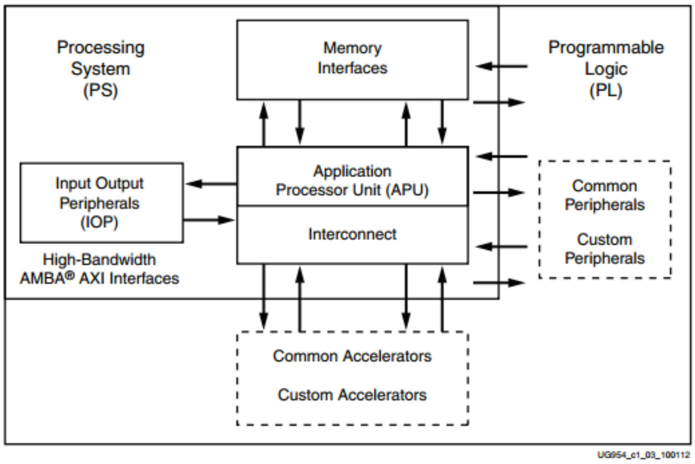


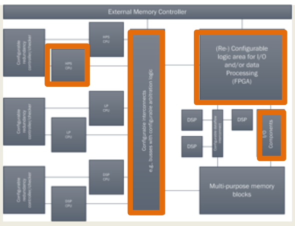
Figure 7: High level block diagram [1]

Concluding it needs to be stated that compared to the solution using the ALTERA ARRIA system as it is described in Section 4.3, 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.

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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
LP CPU
LP CPU
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

3.5.1 Tracing Mixed-Criticality

Often tracing mixed criticality systems refers to execution time monitoring (both the net and gross execution times). Based on this the topic of potential interference in space can be addressed. However, interference in space also needs to be addressed for mixed criticality systems. In this regard tracing the joint access to data, differentiated by criticality, is an interesting challenge. The ISO26262 introduces 6 different safety levels, listed here in the order of increasing criticality: NSR (not safety relevant), QM (quality monitoring), ASIL A, ASIL B, ASIL C, ASIL D. It is required to avoid that:

- A process from a core corrupts code or data from another safety level
- A process from a core executes code from a lower safety level

The figures Figure 9, Figure 8, Figure 10 depict, access to which memory safety area should be allowed (containing code or date which was qualified for a certain ASIL level), depending on which context the application is currently operating in (e.g. control of an air bag). The access is different for code execution, data read and data write respectively. The challenges are to define the access permissions correctly, keeping in mind that code and data may need to be shared between different criticality levels, and then to make sure the access definitions are properly implemented.

		Memory Area Safety Level					
		NSR	QM	ASIL A	ASIL B	ASIL C	ASIL D
Execution Context Safety Level	NSR	x					
	QM		x				
	ASIL A			x			
	ASIL B				x		
	ASIL C					x	
	ASIL D						x

Figure 9: Suggested access for code execution

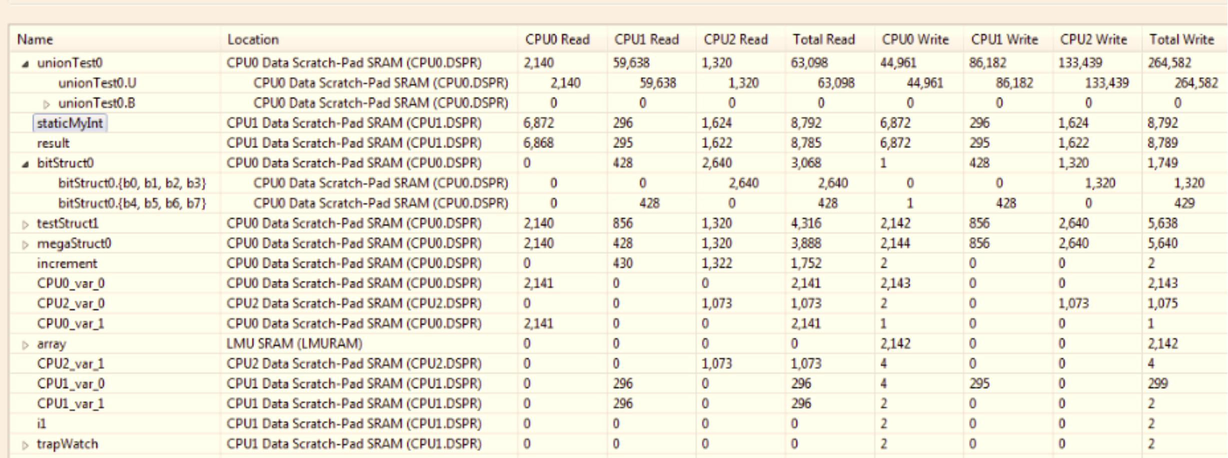
		Memory Area Safety Level					
		NSR	QM	ASIL A	ASIL B	ASIL C	ASIL D
Execution Context Safety Level	NSR	x					
	QM		x				
	ASIL A			x			
	ASIL B				x		
	ASIL C					x	
	ASIL D						x

Figure 8: Suggested access permissions for data read

		Memory Area Safety Level					
		NSR	QM	ASIL A	ASIL B	ASIL C	ASIL D
Execution Context Safety Level	NSR	x	x	x	x	x	x
	QM	x	x	x	x	x	x
	ASIL A	x	x	x	x	x	x
	ASIL B	x	x	x	x	x	x
	ASIL C	x	x	x	x	x	x
	ASIL D	x	x	x	x	x	x

Figure 10: Suggested access permissions for data write

For checking that the access permissions are properly implemented, tracing is very helpful. It needs to be traced in which execution context a certain core is operating (this is the first trace), and which code or data is being accessed, and in which manner (this is the second trace). Infineon developed within EMC^2 a software tool for tracing the access to different variables (the second trace as introduced above), actually as a specific Eclipse view for a tool with the internal project name ChipCoach. A snapshot from the data trace is shown below. The remaining tasks are now to generate the first trace (indicating the execution context safety level), to combine both traces, and to match the result with the matrices as defined above for code read, data read and data write.



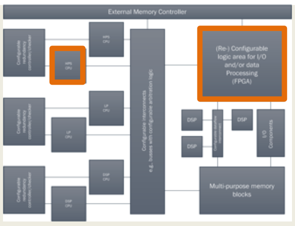
Name	Location	CPU0 Read	CPU1 Read	CPU2 Read	Total Read	CPU0 Write	CPU1 Write	CPU2 Write	Total Write
◢ unionTest0	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,140	59,638	1,320	63,098	44,961	86,182	133,439	264,582
unionTest0.U	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,140	59,638	1,320	63,098	44,961	86,182	133,439	264,582
▷ unionTest0.B	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	0	0	0	0	0	0	0	0
staticMyInt	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	6,872	296	1,624	8,792	6,872	296	1,624	8,792
result	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	6,868	295	1,622	8,785	6,872	295	1,622	8,789
◢ bitStruct0	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	0	428	2,640	3,068	1	428	1,320	1,749
bitStruct0.{b0, b1, b2, b3}	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	0	0	2,640	2,640	0	0	1,320	1,320
bitStruct0.{b4, b5, b6, b7}	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	0	428	0	428	1	428	0	429
▷ testStruct1	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,140	856	1,320	4,316	2,142	856	2,640	5,638
▷ megaStruct0	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,140	428	1,320	3,888	2,144	856	2,640	5,640
increment	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	0	430	1,322	1,752	2	0	0	2
CPU0_var_0	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,141	0	0	2,141	2,143	0	0	2,143
CPU2_var_0	CPU2 Data Scratch-Pad SRAM (CPU2.DSPR)	0	0	1,073	1,073	2	0	1,073	1,075
CPU0_var_1	CPU0 Data Scratch-Pad SRAM (CPU0.DSPR)	2,141	0	0	2,141	1	0	0	1
▷ array	LMU SRAM (LMURAM)	0	0	0	0	2,142	0	0	2,142
CPU2_var_1	CPU2 Data Scratch-Pad SRAM (CPU2.DSPR)	0	0	1,073	1,073	4	0	0	4
CPU1_var_0	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	0	296	0	296	4	295	0	299
CPU1_var_1	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	0	296	0	296	2	0	0	2
i1	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	0	0	0	0	2	0	0	2
▷ trapWatch	CPU1 Data Scratch-Pad SRAM (CPU1.DSPR)	0	0	0	0	2	0	0	2

Figure 11: Data access trace, implemented within the tool ChipCoach

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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

3.6.1 Fault tolerant Platforms based on SoC FPGAs

In the following sections, we provide a short overview of the current developments for space domain based on SoC COTS components. We describe a demonstrator based on SoC FPGA and Cortex-R5F as a supervisor for fault-tolerant applications.

3.6.1.1 SoC for Space missions: State-of-the-art

OPS-SAT is a nanosatellite (3U form factor) designed by ESA to provide a platform for in-orbit validation of these new concepts developed by ESOC (European Space Operations Center). OPS-SAT is devoted to demonstrating drastically improved mission control capabilities that will arise when satellites can fly more powerful on-board computers. It consists of a satellite that is only 30cm high but that contains an experimental computer that is ten times more powerful than any current ESA spacecraft. The processing platform running Linux as operating system consists of a flexible and reconfigurable framework featuring sophisticated processing capabilities, interfaces, memory integrity and reconfigurable logic based on Cyclone V SoC.

The CSP (CHREC Space Processor) was developed at CHREC, in collaboration with the NASA SpaceCube team. The CSP offers a low-cost solution for space processing requirements in a 1U Cubesat form factor. With a hybrid of COTS (Zynq SoC) and Radiation Hardened Components, the CSP offers high performance computing abilities protected by dependable fault tolerant architecture. The CSP includes the lightweight embedded OS Wumbo GNU/Linux customized for this system.

3.6.1.2 Fault-tolerant platform based on Zynq SoC and Cortex-R5F supervisor

A Zynq SoC FPGA has been selected to implement space solutions. Several strategies will be implemented to mitigate the radiation effects:

- External Watchdog controller
    - If a problem is detected with Zynq SoC, watchdog can reset it.
- Scrubbing
    - ARM cores in Zynq can do blind scrubbing
- Internal Watchdogs and built in ECC
    - Several internal watchdogs are employed

We plan to use a Cortex-R5F for safety-critical applications instead of using a conventional external Watchdog controller. The main advantage of this change is the possibility to interact with Zynq to report the status of several Zynq subsystems.

The communication between both kits will be based on high-speed SPI interfaces with PS/PL areas and dedicated lines to reset the Zynq. A high-level block diagram can be seen in next figure.

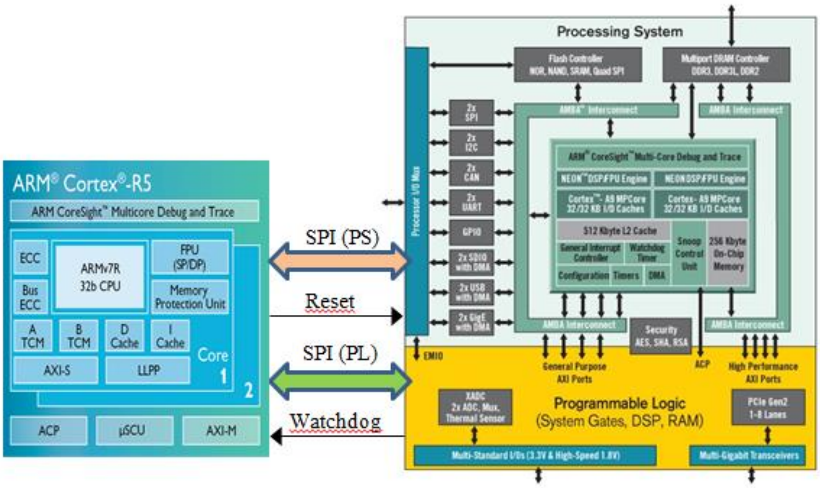


Figure 12: Demonstrator architecture

Two evaluation kits will be used to implement the Zynq SoC FPGA (Zedboard kit) and ARM Cortex-R5F (TI Launchpad TM570).

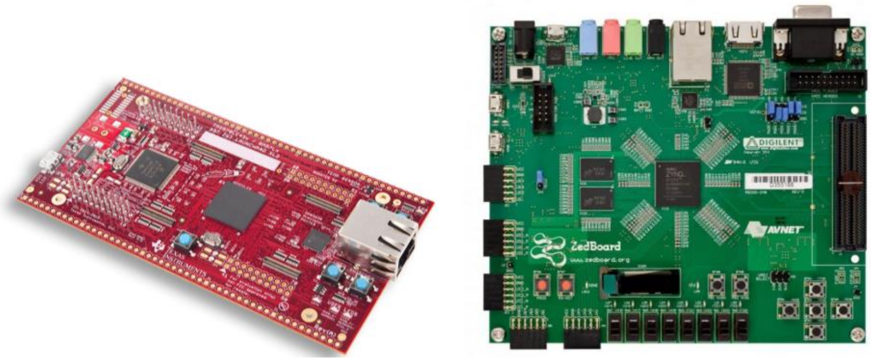
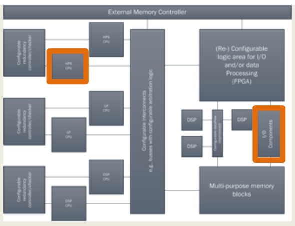


Figure 13: Evaluation kits for Dual-Core ARM Cortex-R5F and Zynq SoC FPGA

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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

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

SoCRocket is a SystemC TLM-model of the LEON3-based GRLIB multi-core platform developed by the Technical University of Braunschweig. Throughout Y2, we have been working on an in-house SystemC AT/LT-TLM-model of the ARM CortexTM-A9, to be integrated in the SoCRocket framework. Our approach seeks to combine the real-time advantages of the LEON3 with the performance of the Cortex. We are currently analyzing different scenarios for deploying the cores, including options for hybrid AMP/SMP core-clustering. Intra- and inter-cluster synchronization can both minimize concurrency over shared resources and guarantee time-predictability. This will furthermore elucidate the effects of heterogeneity on data integrity and analyzability, as investigated in the following section.

Current Status: Completed implementation of the CortexTM-A9 SystemC model featuring the full ARMv7 instruction set.

Upcoming in Y3: LEON3/Cortex deployment scenarios.

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

The power of multi-cores is best harnessed with fine-grained core utilization, which calls for low-overhead inter-core communication, best achieved in shared-memory systems. With a growing number of cores, cache hierarchies prove indispensable for avoiding latencies and upping performance. Conventional cache coherence techniques deriving from the classic MSI protocol show largely indeterministic timing behavior due to complex cache interactions, rendering them inapplicable to hard real-time systems. We have successfully developed a time-predictable L1-cache coherence protocol specifically tailored to mixed-critical systems. Our protocol guarantees a-priori analyzable invalidations, while maintaining efficient caching of both private and shared data. Furthermore, we have augmented the

SoCRocket LEON3 cache with some of the classic coherence techniques for a qualitative performance comparison of the different coherence protocols.

Current Status:

- Implemented and integrated the time-predictable coherent cache in the SoCRocket platform.
- Implemented several coherence protocols for comparison.

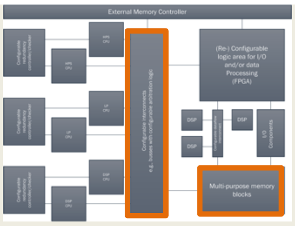
Upcoming in Y3:

- Run suitable benchmarks for testing coherence protocols.
- Deploy a WCET-analysis tool.

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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
LP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

3.8.1 Memory compression techniques

In highly embedded systems, memory is a significant cost adder. Algorithms that reduce intermediate data in the decoding flow can significantly contribute to reduction of memory requirements and, thus, chip area of the product. By applying smart compression schemes, the same or a similar algorithmic performance can be achieved despite the memory reduction.

As a promising candidate, a digital radio system’s interleaver memory was chosen. Interleaver memories typically have to store reception data over a relatively long time, in order to achieve effective time interleaving. Furthermore, this data should be stored at high bit resolution. Interleaver memories therefore are typically several 100 kByte big, which makes them an attractive target for compression techniques.

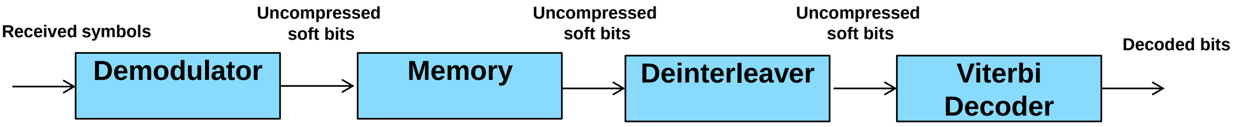


Figure 14: Typical data flow from demodulator via interleaver to Viterbi decoder

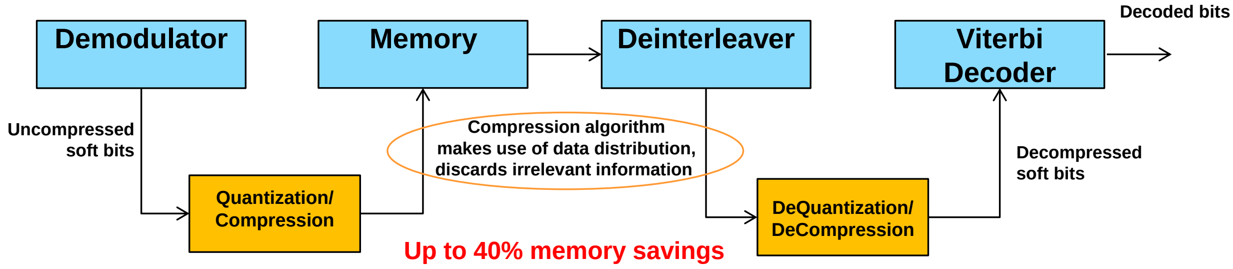

Demodulator
Memory
Deinterleaver
Viterbi Decoder
Decoded bits
Uncompressed soft bits
Quantization/ Compression
Compression algorithm makes use of data distribution, discards irrelevant information
DeQuantization/ DeCompression
Decompressed soft bits
Up to 40% memory savings

Figure 15: Memory optimized data flow from demodulator via interleaver to Viterbi decoder

Development phases:

NXP developed a technique to decrease the interleaver memory consumption based on data quantization and smart data compression. The performance of the system was compared for uncompressed data, data reduced by quantization, and a system with a smart compression algorithm.

As interleaver data must be stored or read in a non-linear fashion, standard compression techiques like LZW cannot easily be applied: The nature of such an algorithm makes it hard to determine the position of a certain input bit in the compressed memory, thus adding a high administrative effort for non-linear access to the stored input bits. NXP has developed a special compression algorithm that makes it easiy to determine the position of certain bits, and yet provides an effective compression of the data.

Outcome:

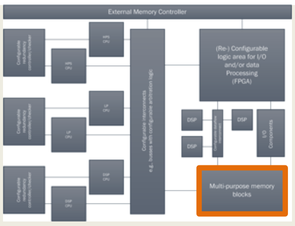
Stronger quantization of data results in an effective memory reduction, but also affects the performance in a noticeable way. As a reduction of reception performance of half a dB and more is not seen as acceptable for higher end systems, this simple approach is applicable for such systems.

The developed smart compression algorithm allows a memory reduction of 20% without measurable effects on performance, and even allows a 40% memory reduction within the given margin for higher end systems.

3.9 Imec-nl (17B imec-nl)

Technology lanes covered:

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


External Memory Controller
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Multi-purpose memory blocks

3.9.1 Modeling of Reliability of Memories

Introduction and Work Progress

On-chip memory consumes a large portion of power budget of a multiprocessor SoC. Voltage scaling is known as an effective approach to saving the power dissipation of memory. Lowering the supply voltage however sacrifices memory reliability. Modeling the failure behavior of memory cells at the low voltage is therefore important for the development of reliable and power-efficient multiprocessor SoCs.

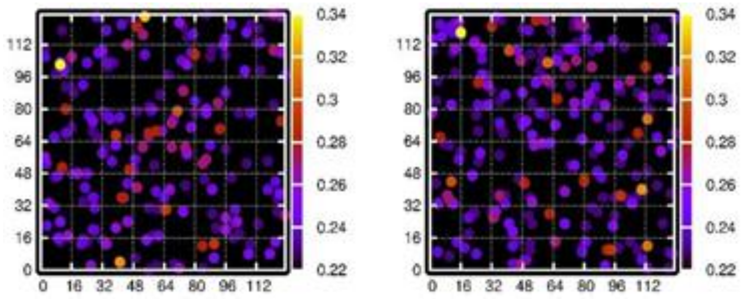


Figure 17: DRV vs. memory location: commercial memory IP (left) and standard-cell-based memory (right)

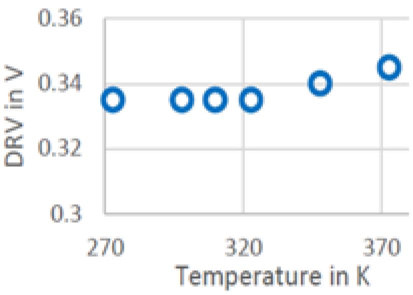


Figure 16: DRV as function of temperature

For the reliability characterization, we measured two cases of memories, namely a commercial off-the-shell (COTS) 6T-cell SRAM and a 18T standard-cell-based (SC) SRAM. Both instances (1k x 32b in a 40-nm process) are integrated on a test chip and was measured on silicon. The SC-SRAM is designed fully based on combinational standard cells, in our case a pair of cross-coupled AND-OR-INVERT gates for each bit cell. The data retention voltage (DRV: the minimal voltage that is required to preserve the logic states) is captured for both memories across variant supply voltage and temperature Figure 16 presents the individual bit failure as a function of DRV for one die with the word address on the x-axis and the bit address on the y-axis. The color scheme highlights the DRV from 340mV down to 220mV, which is randomly distributed over the die. Figure 16 shows the DRV as a function of temperature. The DRV features a weak dependency of 10mV on temperature from 50°C to 100°C.

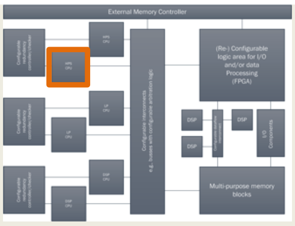
Work Plan for 3rd Year

Data will be measured systematically on silicon in different aspects. The measurement results will be used to model the voltage and temperature dependent effects, including DRV, leakage current, as well as active switching energy and failure rate, of both COTS and SC-SRAM.

3.10 SELEX/POLITO (11D SELEX, 11K POLITO)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
LP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

3.10.1 Mixed Critical Avionic Application

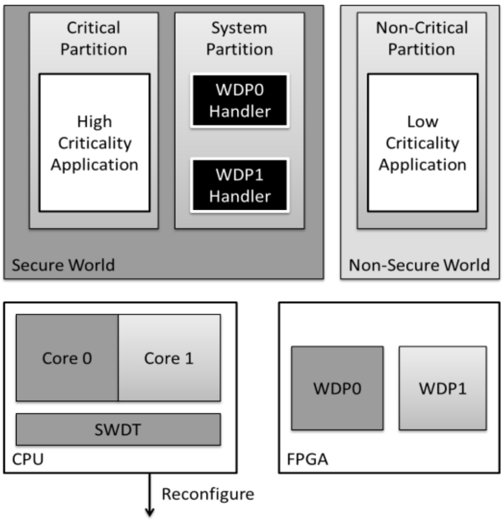


Figure 18: Hardware-based solution Dark: Secure World. Light: Non-Secure World.

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 a Xilinx Zynq System-on-Programmable-Chip (SoPC), to consolidate two avionic applications part of a Fligh Control System (FCS) for an Unmanned Aerial Vehicle (UAV).

The architecture uses a partitioning mechanism to separate the two applications and a specifically designed watchdog processor (WDP) IP deployed in the Programmable Logic (PL) of the target SoPC. 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 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.

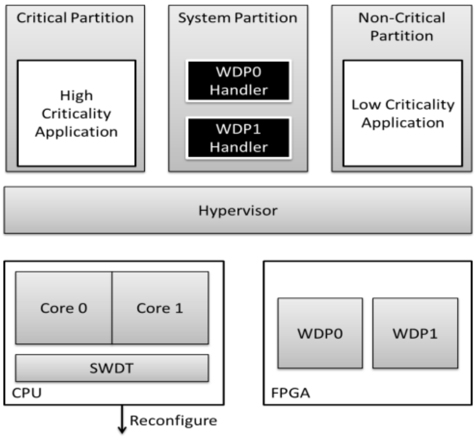


Figure 19: Hypervisor-based solution

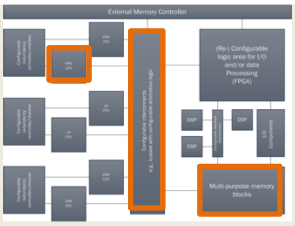
Two partitioning mechanisms can be used. The first is based on a type-1 hypervisor. Each application is deployed in its dedicated partition and mapped on a specific core. The hypervisor is in charge of isolation. A third partition is used to manage the WDPs interrupts. The second partitioning mechanism is a hardware-based mechanism known as the TrustZone, allowing the definition of two partitions, one is the Secure World, the other is the Non-Secure World. The non-critical application is mapped to the non-secure world and can only access resources and memory areas explicitly assigned to it. The critical partition was mapped to the secure world and it is in charge of managing the WDPs interrupts. At boot, the secure world is in charge of configuring the non-secure world and launching the non-critical application.

Both solutions were evaluated through fault injection experiments, for both hardware and software faults effects. Results show that both solutions are viable, although the hardware partitioning scheme is slightly more robust with respect to hardware faults, although it is less portable.

3.11 NXP Semiconductor (17C NXPNL)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

3.11.1 Parallelization in multi-core architecture

Following the architectural trend in desktop computing, new devices for deeply embedded applications are increasingly adopting multicore architectures to address ever-increasing performance requirements while being constrained by power dissipation limitations. This trend is reflected in various NXP automotive microcontrollers ranging from "simple" loosely-coupled dual core architectures to more advanced next-generation architectures that contain Single Instruction Multiple Data Signal Processors next to SISD processors.

This common platform architecture approach allows implementing several radio products for multiple markets with an optimum balance of cost and performance. The benefits offered by SDR include low cost of ownership for tier-1s and OEMs, and reduced time-to-market for new standards. It also reduces logistical support and operating expenditures when adding new features to existing infrastructures. In addition, SDR ensures easy adaptation to regional variations via software download, and over-the-air (OTA) upgrades for quick and cost effective updates while the radio is still in service.

NXP works to get a truly optimized solution that has the necessary DSP performance to tackle multi-standard next-generation digital radio applications in a small and power-efficient core. The parallelization needs to be addressed by smart algorithm making use of the SIMD architecture of the DSP and a tightly coupling to the control CPU to run efficiently the control code. A block diagram of the concept is shown below (see Figure 20).

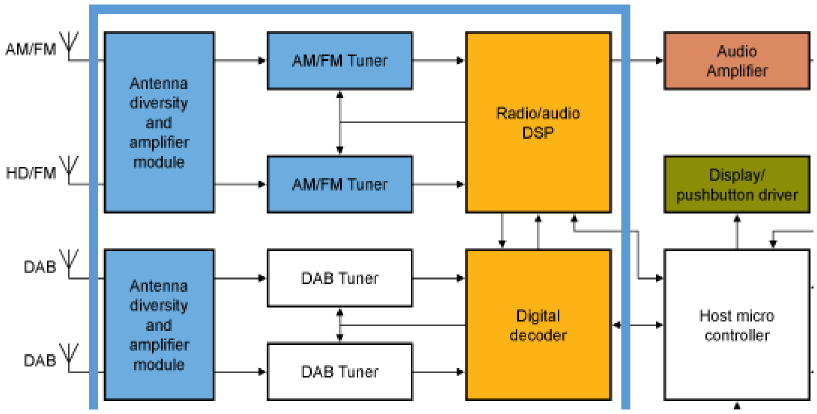
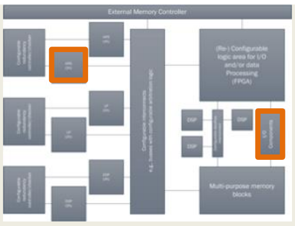


Figure 20: Block diagram of parallelization concept in a multi-core architecture

3.12 Infineon Great Britain GB (18B IFXUK)

Technology lanes covered:



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

3.12.1 Scalable multi-core architecture for usage of the effective ADAS sensors

D4.3: An FPGA Virtual platform which was develop for a feasibility task during the first year to capture Mixed Criticality requirements and concepts thru using modelled Radar Signal.

D 4.4: During the second year, a first demonstrator employing the AURIX automotive controller was implemented and a Radar application was emulated using single FPGA platform and LVDS interface scalable per lane.

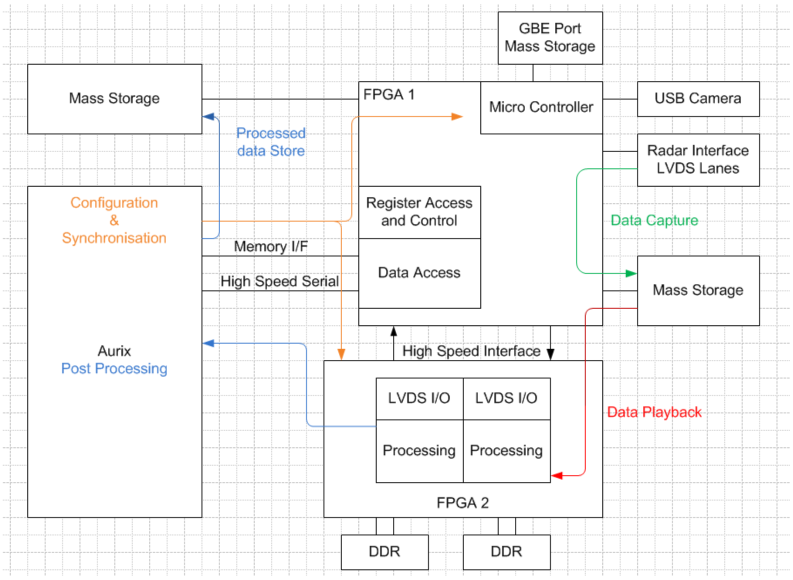


Figure 21: ADAS Sensor Architecture

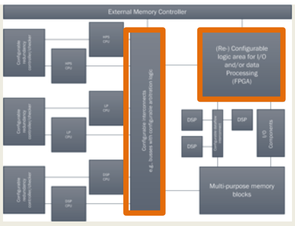
Again during the second year an improved architecture employing second Xilinx FPGA was suggested for a complete Radar MCU with HW Accelerator using one of Aurix core, and peripherals for real time critical data post processing, and another FPGA for only Front End data ordering, classification, number crunching and mass storage infrastructure controlling on a same platform - the other cores (Aurix family / scalable) are to deal monitoring of other passive and active non ADAS Safety mechanisms. During the year three this improved virtual multiple FPGA platform will be developed to validate several challenging use cases combined with Vision applications with real Radar Signal captured using an Antenna board (under design). Several mixed critical use cases for active Safety Monitoring are blind spot detection, rear cross alert, long range RADAR for Adaptive Cruise control, Pedestrian detection, and Radar Imaging.

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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.1.1 Resilient Service-oriented NoC

As discussed in D4.3, Chalmers in ST4.3.1 works on the design 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) or by using NoC resources of other services (inter-service redirection, Service merge). 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.

We are on track with our plan set in period 1. In period 2, we have completed the design and implementation of the technique on a service oriented NoC. We considered the QNoC design[1] as our baseline. That is a packet switched 2D mesh NoC with four channels and four separate datapaths for each service that uses wormhole XY routing. Our Service detour mechanism was implemented in the Network interface by adding a detour table per service to indicate an intermediate network node for packets that would normally use faulty NoC resources towards their destination. Our Service Merge mechanism was implemented by modifying the router architecture to allow packets of one service to be redirected through the router resources/datapath of another service and then switch back to their own resources (when available) in the next hop. This required (i) changes in the credit handling according to the router defect-map, (ii) implementing a service merging policy (that decides which services can be merged with each other at different cases of faults) that satisfies the traffic classes requirements, (iii) modifications to the output of the router and to the router datapath for keeping track of the packet service-id and allow the

[1] E Bolotin, I. Cidon, R. Ginosar, A. Kolodny. QNoC: QoS architecture and design process for network on chip. J. Syst. Archit. 50, 2-3, pp.105-128, 2004.

packets of a merged service to use again their original resources in the next hop, (iv) each neighboring router needs to be aware of merging decisions, too. Finally, it is worth mentioning that links are designed to be more robust using spare wires and a shifting mechanism to allow bypassing faulty wires.

The above have been implemented in RTL and can be demonstrated standalone.

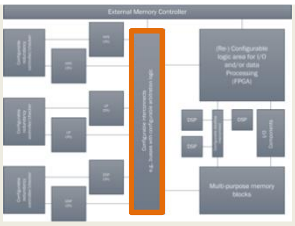
In period 3, we will evaluate the performance, area, power and fault tolerance of the implemented resilient service-oriented NoC.

The proposed approach offers a flexible solution for tolerating permanent faults in a service-oriented NoC exploiting resources redundancy and trading QoS with fault tolerance. Currently, very little has been done to protect service-oriented NoCs from permanent faults and even less in describing a holistic approach that covers all NoC parts.

4.2 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*


External Memory Controller
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
I/O Components
Multi-purpose memory blocks

In D4.3, UoBR focused on core-based support for mixed-criticality systems, through the Swallow system as part of T4.2. Work during the period covered by this deliverable, D4.4, changes focus towards the interconnect via T4.3. This is motivated by structural changes within UoBR's DOW, explained in Section Figure 22 as well as observations of the grid-like interconnect of Swallow and other similar systems, that can be restrictive in a multi-core mixed-criticality environment. Swallow remains as a research platform, and is now open-sourced to broaden access to it[2].

4.2.1 A predictable non-blocking NoC for multi-core mixed criticality systems

As part of ST4.3.1, UoBR addresses the issue of providing a highly predictable network for interconnecting processors, memories and peripherals. Such networks are beneficial in a mixed-criticality context, as well as in addressing performance imbalances resulting from scaling issues as core-counts increase in contemporary computing systems. It is therefore a compelling and timely research challenge to investigate.

This work utilizes network structures based on Clos and Benes styles of layered network, historically used in telephony for non-blocking routing of messages, but in this case scaled down to the NoC level. The objective of this work is to create a highly scalable network of switches with precise behavioral guarantees. Resources can then be allocated statically, where routing and access periods are pre-determined based on requirements including

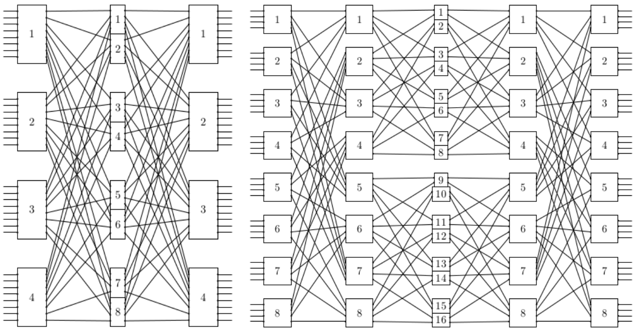


Figure 22: Two example 32-way networks configurable from the UoBR switch design.

[2] http://swallow-project.github.io/

desired bandwidth, latency and criticality/error sensitivity.

The design and implementation of this switch was started towards the end of this reporting period and will continue into the next period. Currently, an initial prototype switch design is implemented and is undergoing verification for node counts of 16, 64 and beyond. The verification techniques are contributed within task T4.5 and are explained in D4.12. The network allows latencies to scale logarithmically to the number of endpoints in the system, and the structure guarantees that all routes have the same latency. Per-layer (individual switching unit) latency is targeted at a single-cycle per symbol on a channel. Completing our work in T4.2 we have identified candidates for integration with the switch design, including the Xcore XS1-L processor used in Swallow, the newer XS2 based Xcore-200, and synthesized Microblaze cores.

The main challenges of this work include constructing a switch that is sufficiently simple to both implement, synthesise and verify, that can then be replicated and connected in a Clos/Benes structure. Guaranteed timing behavior of the network structure is particularly important, such that the switch fabric will provide a precise latency. In addition, there are software challenges, with respect to allocation of resources, pre-calculation of routes and handling of errors without negatively impacting higher-criticality tasks.

During the next reporting period, UoBR will be improving the switch and network design based on initial validation and verification runs. We will then begin working on workload examples and routing strategies, as well as connecting compute, memory and if feasible peripheral resources to the network for a more complete demonstration of the technology that allows its scalability and other practical aspects such as performance to be explored. This will be assisted by synthesis to FPGA, which will take place after functional verification is satisfied.

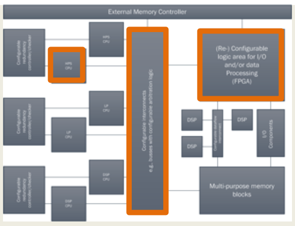
There are two target hardware platforms that can potentially be explored. First, the UoBR internally developed MAny Compute Cores System (MACCS) board that can use Kintex-7 and Zynq FPGA modules, with several multi-gigabit transceiver links between boards for large scale designs. Second is the EMC^2-DP hardware produced by project partner 18D, Sundance. Integration with candidate processors will be investigated on FPGA and other hardware as well as in simulation, to de-risk potential hardware issues that would block further research. Thus, the ARM cores of Zynq are an option for processor integration, but not the sole route to demonstration.

The ground-up nature of this work restricts the scope of technology transfer, such that the designs and techniques will transfer into the living labs, but not be implemented directly into a demonstrator. However, automotive application contexts will be used to guide the validation and verification criteria for the network.

4.3 TU Wien (02E TUW)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

In the D4.3 TU Wien proposed an extension of the hardware architecture based on time-triggered NoC (TTNoC). Also, a basic motivation behind the augmented architecture was introduced, with the overviews of the activities performed in the first year of the project. In this section, we provide overview of the progress achieved in the second year of the project.

4.3.1 Heterogeneous TTNoC many-core architecture

As mentioned in the deliverable D4.3 a platform for the implementation of the architecture is a hybrid SoC platform. We chose the Altera ArriaV SoC platform for this project. The reason is high compatibility with former implementation, which also used Altera platform (Stratix IV).

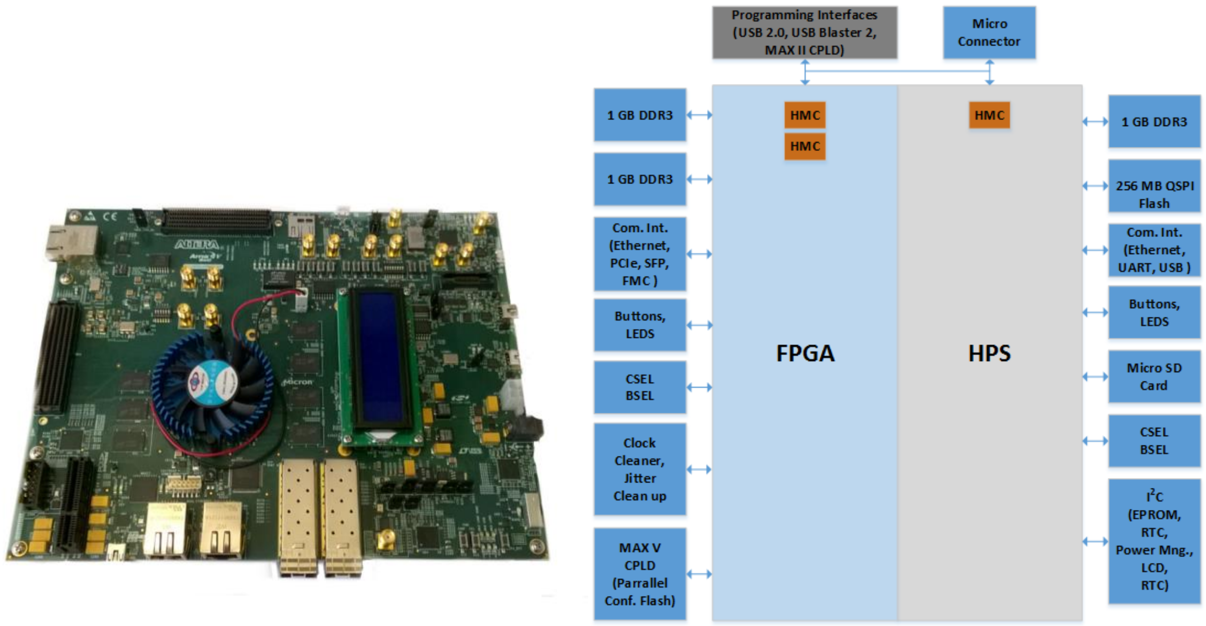

Programming Interfaces (USB 2.0, USB Blaster 2, MAX II CPLD)
Micro Connector
1 GB DDR3
1 GB DDR3
Com. Int. (Ethernet, PCIe, SFP, FMC)
Buttons, LEDS
CSEL BSEL
Clock Cleaner, Jitter Clean up
MAX V CPLD (Parrallel Conf. Flash)
HMC
HMC
HMC
FPGA
HPS
1 GB DDR3
256 MB QSPI Flash
Com. Int. (Ethernet, UART, USB)
Buttons, LEDS
Micro SD Card
CSEL BSEL
I²C (EPROM, RTC, Power Mng., LCD, RTC)

Figure 23: Arria V SoC Development Board and Block Diagram of the Architecture

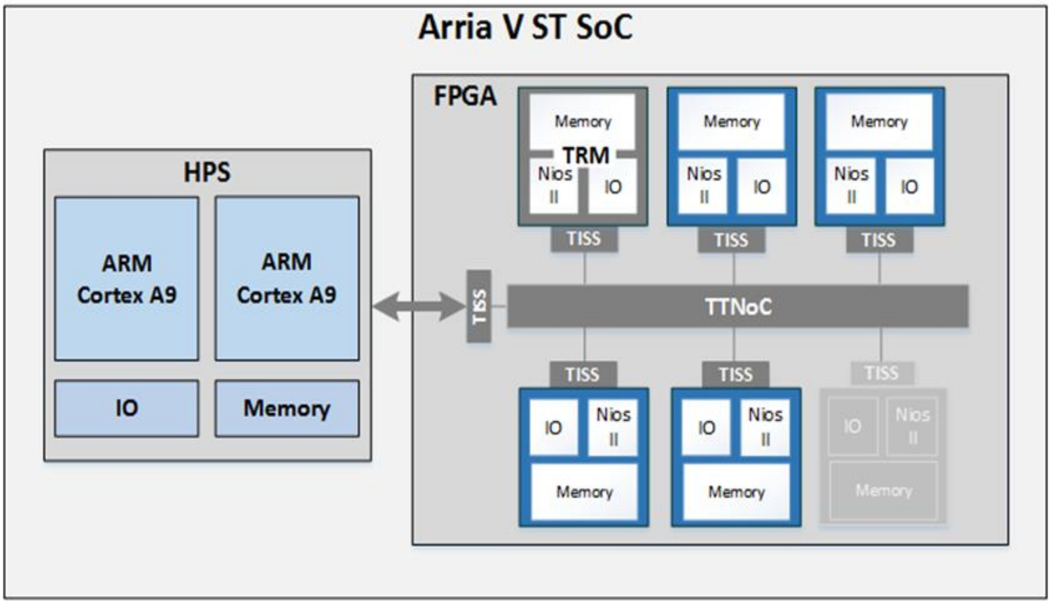


Figure 24 Heteroegenous TT MPSoC architecture

The Figure 23 shows the platform with the corresponding block diagram of the system. It contains a mid-range FPGA device Arria V and a hard-processor system (HPS) based on dual-core ARM Cortex A9 processor. The hybrid SoC devices provide extensive flexibility, where each component can use the advantages of the other and thus increase its limits set by architectural design.

The architecture presented in the document utilizes both devices of the SoC. It integrates five components implemented on the FPGA device and a single component that uses HPS. The components are also called µComponents. Each µComponent is a fully functional independent computer system, with local memory and IO connections. The FPGA components are implemented on NIOS2 processors and local memory. All components are isolated from each other and the rest of the system in both time and space. This allows mixed-criticality, and even cross-domain, integration on this architecture. The communication backbone of the architecture is the TTNoC.

It is a modular and flexible massage based time-triggered network. All µComponents interact with the TTNOC trough a dedicated trusted interface called trusted interface subsystem (TISS). It holds configuration files and communication schedule. TISS is connected to a processor of a µComponents via dual-ported memory, and interrupt signals. For the soft-coded components implemented in an FPGA this concept has been implemented in an earlier project. The integration of the HPS component required mirroring this principle to interface HPS as an additional component. Basic block diagram of the heterogeneous architecture is shown in Figure 24.

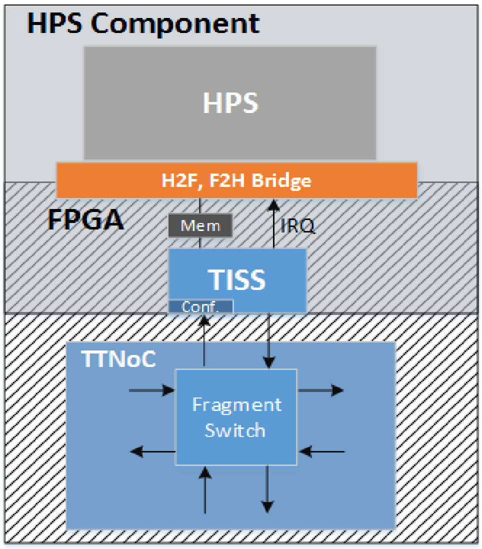


Figure 25: HPS µComponent

The HPS µComponent contains parts implemented in HPS and FPGA. Basic parts of the HPS component are depicted in Figure 25. The selected platform provides infrastructure for the integration of two devices in the architecture. The original properties of the system (e.g., spatial and temporal isolation) of the components have been preserved. The HPS components extend the capability of the system with increased processing power, which is required for applications with high computation load requirements. One of the main advantages in using hybrid SoC is a flexibility of a hardware design. It allows hardware designer to setup custom bindings of different HW modules on the platform. Following this principle, the proposed architecture is organized in a modular fashion. It is a basis for a system modeling tools that would be able to mold the architecture to a specific application.

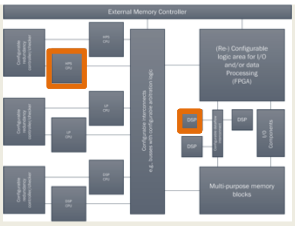
In the second reporting period, the presented architecture advanced from a design and specification phase in the implementation phase. First version of the hardware design has been completed and deployed on the platform. Basic tests have been performed and they showed successful hardware integration. Implementation of system software, operating systems and drivers is in progress.

In the upcoming period it is planned to finish the implementation phase and testing, and work on the demonstration application. It is planned to use the architecture in a space related demonstrator, as a part of technology exchange between technical work packages and living labs. We are consulting experts from the space sector in WP9 on a most optimal application for this architecture. The documents related to these contributions are D4.3, D4.15, D4.16, D9.1, D9.3.

4.4 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*


External Memory Controller
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
Multi-purpose memory blocks

4.4.1 Asymmetric Multiprocessing on ZYNQ

The asymmetric multiprocessing of ARM Cortex A9 and MicroBlaze soft core with floating point vector accelerator EdkDSP has been described in the deliverable D4.16. This section is describing the progress of generation of accelerators defined as C functions and by Xilinx high level synthesis (HLS) into IP cores for the programmable logic of ZYNQ device. This can be done in the Xilinx SDSoc environment 2015.4. The environment is capable to complement the HLS generated IP cores with data movers serving for DMA transport of data from DDR3 to programmable logic.

Initial Vivado 2015.4 design has to be prepared in advance for the concrete HW platform. See Figure 26 for concrete implementation for the Sundance EMC2 DP HW platform with ZYNQ 7Z015-1I SoM.

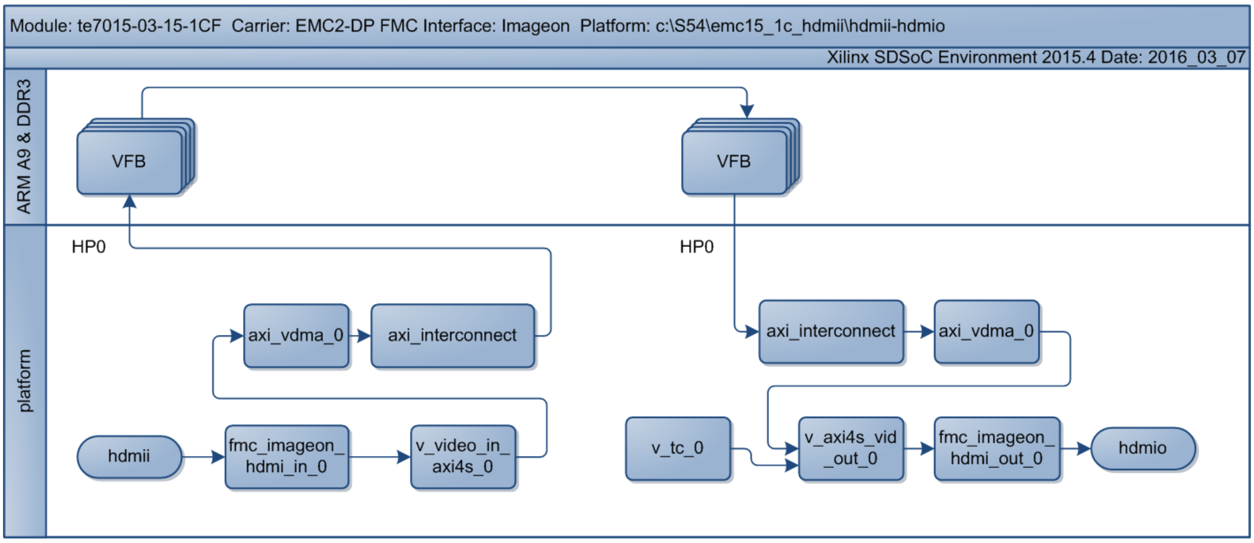


Figure 26: EMC2 hdmii-hdmio HW Platform for SDSoC environment and Sundance EMC2-DP with ZYNQ 7Z015-1I System on Module

The Xilinx SDSoC environment is capable to compile C and C++ program for ARM Cortex A9. This is used for initial debug of algorithms, encapsulated in C and C++ functions. Multiple functions can be selected in the Xilinx SDSoC for compilation into HW accelerator with data-movers for the programmable logic part of the ZYNQ device. We have declared two parallel computing chains working in 120MHz on 2 parts of the video frame in parallel. Figure 27 is highlighting the auto-generated data movers and the automatically generated direct connectivity of accelerators. The corresponding C++ code has been transformed to control the auto-generated four DMA data movers. The HDMII-HDMIO platform is present and performs the VDMA at 150 MHz as described in Figure 26. The HDMII input and HDMIO output are both 148,5 MHz Full HD 1920x1080p60.

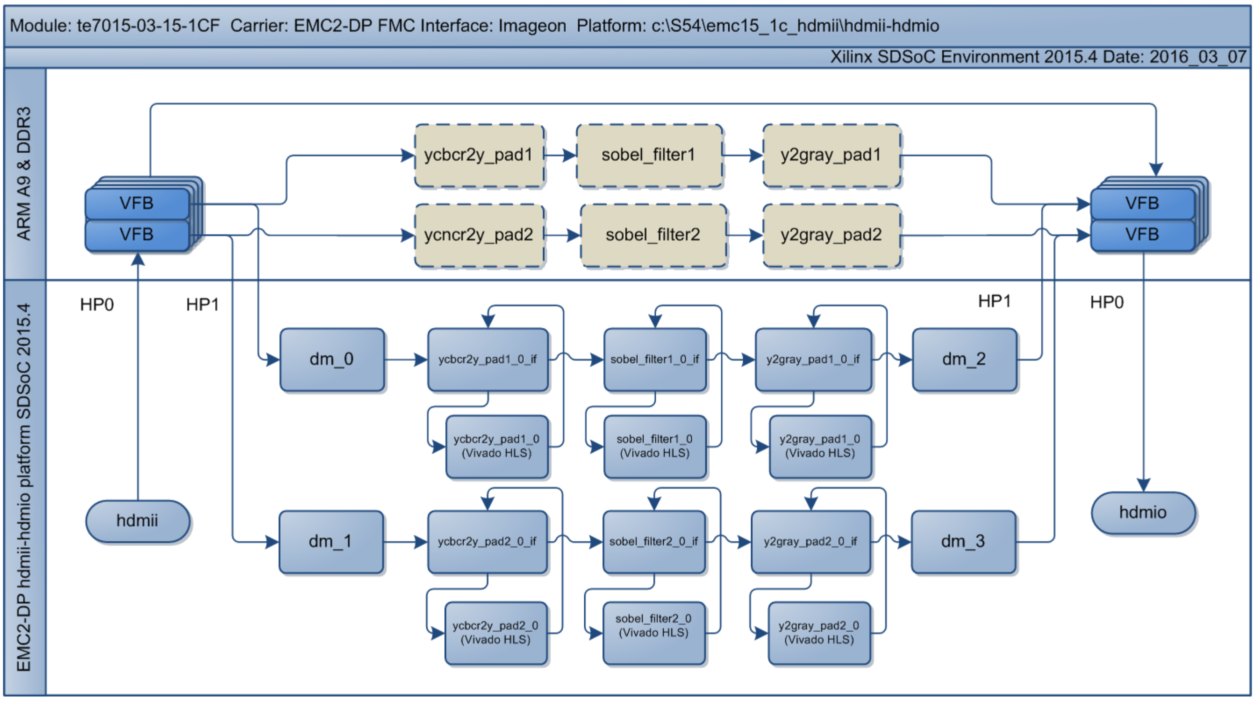

Module: te7015-03-15-1CF Carrier: EMC2-DP FMC Interface: Imageon Platform: c:\S54\emc15_1c_hdmii\hdmii-hdmio
Xilinx SDSoC Environment 2015.4 Date: 2016_03_07
ARM A9 & DDR3
EMC2-DP hdmii-hdmio platform SDSoC 2015.4
VFB
VFB
ycbcr2y_pad1
sobel_filter1
y2gray_pad1
yencr2y_pad2
sobel_filter2
y2gray_pad2
VFB
VFB
HP0
HP1
HP1
HP0
dm_0
ycbcr2y_pad1_0_if
sobel_filter1_0_if
y2gray_pad1_0_if
dm_2
ycbcr2y_pad1_0 (Vivado HLS)
sobel_filter1_0 (Vivado HLS)
y2gray_pad1_0 (Vivado HLS)
hdmii
dm_1
ycbcr2y_pad2_0_if
sobel_filter2_0_if
y2gray_pad2_0_if
dm_3
hdmio
ycbcr2y_pad2_0 (Vivado HLS)
sobel_filter2_0 (Vivado HLS)
y2gray_pad2_0 (Vivado HLS)

Figure 27: Edge-detection algorithm developed in ARM C code and accelerated by Xilinx SDSoC 2015.4 into HW programmable logic for Sundance EMC2-DP with ZYNQ 7Z015-1I System on Module

Figure 28 is comparing the accelerated edge-detection in full HD HDMII-HDMIO expressed in frames per second, percentage of slices and percentage of block RAM used in the programmable logic part of the ZYNQ 7Z015-1I System on Module. The 120MHz Edge detection accelerator framework (with two parallel data paths) has accelerated from 5,33 FPS to 56,95 FPS.

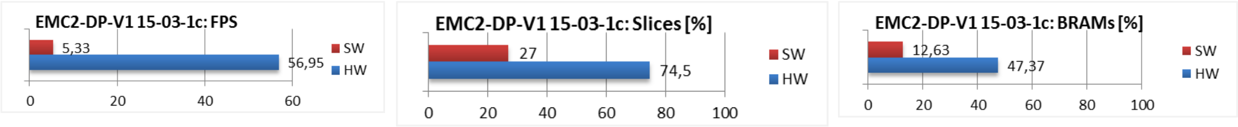


Figure 28: Performance of Full HD Edge-detection. EMC2-DP with ZYNQ 7Z015-1I System on Module

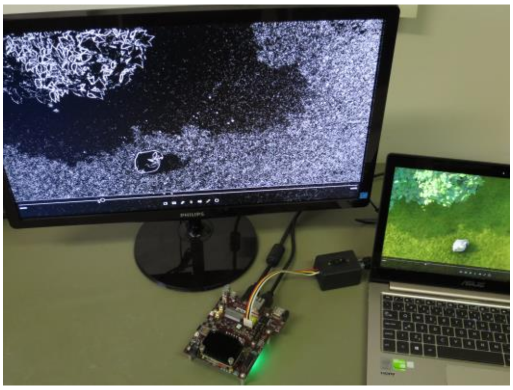


Figure 29 is documenting the edge-detection algorithm develop accelerated by Xilinx SDSoC 2015.4 running on Sundance EMC2-DP with ZYNQ 7Z015-1I System on Module

Average compile time in SDSoC:

C/C++ to ARM:	1 min
C/C++ to Vivado IP Integrator project:	10 min
C/C++ to HW (SD card BOOT.BIN):	60 min
C to ARM (SDK, HW in static lib.a):	1 min

Figure 29: Edge-detection running on Sundance EMC2-DP with ZYNQ 7Z015-1I System on Module

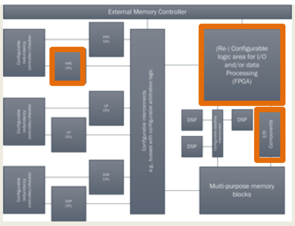
Contact: UTIA, Prague, Czech Republic, See: http://sp.utia.cz/index.php?ids=projects/emc2

Jiri Kadlec kadlec@utia.cas.cz Zdenek Pohl xpohl@utia.cas.cz Lukas Kohout kohoutl@utia.cas.cz

4.5 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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
Configurable dataflow interconnect
DSP
I/O Components
Multi-purpose memory blocks

4.5.1 TTEthernet WireLess (WL)

The following contribution uses results and explanations published in [2]. 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 interfering 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 herein. Also, by allowing more than one receiver, the future implementation of steps to increase reliability is left open.

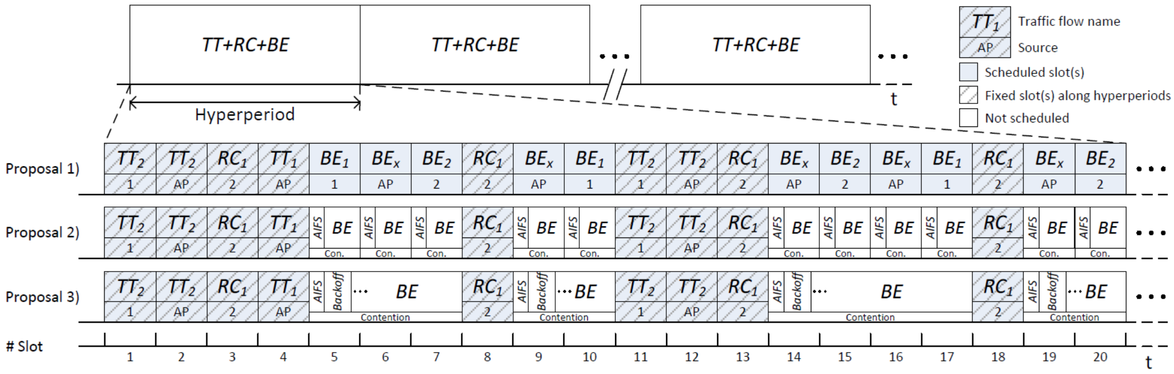


Figure 30: Example of slot allocation for the tree MAC approaches

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 most fair way of sharing these resources is using a round-robin schedule, giving the same amount of timeslots to each node in a circular order.

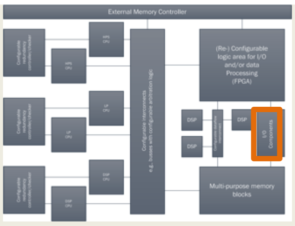
Ad 2 Pre-scheduled time-slots and contention-based timeslots: This approach is similar to approach 1, but the remaining time-slots not used by TT and RC can be used for BE traffic flows following a contention process. Further, one single node in each slot is given a higher priority, using a higher priority 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 a 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 [3]. 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.

4.6 AVL (02A AVL)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.6.1 Networking for verification and validation of mixed-criticality applications on multicore automotive control units: motivation

In order to achieve the challenging goals to decrease energy consumption and environmental impact, together with increased expectations for customer safety convenience, automotive industry has to turn to modern multi-core architectures for next generation of electronic control units (ECU). Multiple tasks of various criticality and real-time requirements have to be executed in parallel in a multi-core system, and thus are hosted in a single ECU. However, verification and validation of such systems requires performant and deterministic communication via in-vehicle networks (CAN, in future automotive Ethernet) between such ECU's and test systems. AVL researches on methods how to support and manage such multiple streams of communication on top of today's and next generation automotive networks.

The main idea behind networking for V&V is to create a component, so called "Connectivity Manager", which takes care of managing communication of mixed-critical data of applications.

Since multiple applications may operate simultaneously while differ in criticality, a classification followed by prioritization of application data needs to be achieved. Moreover, the applications may run on different cores, which need to share one single hardware port for communication. Therefore, the "Connectivity Interface" instance needs to be available on each core.

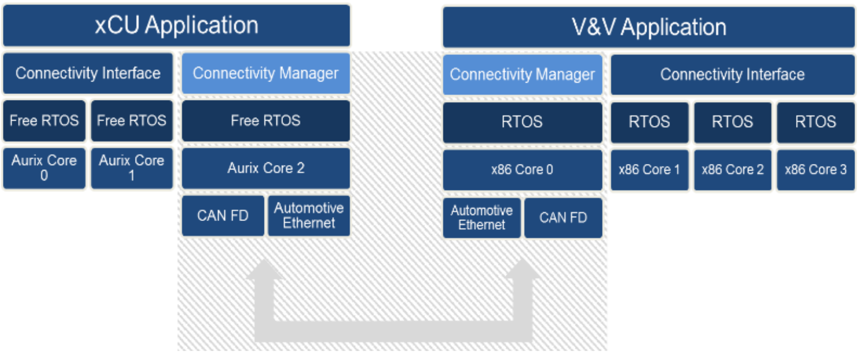


Figure 31: Managing communication between multiple mixed-criticality applications on a multi-core

Concept

As can be seen in Figure 31, the architecture consists of two platforms with both of them hosting multiple cores.

The left side represents the embedded system with its electronic control unit (ECU) and its applications. The right side illustrates the automotive test system (also kwon as V&V system) which runs validation and verification applications.

The two platforms are connected via a single physical line, which can be CAN, CAN-FD or Ethernet. Both platforms take use of a Connectivity Manger multiplexing the bidirectional communication. In addition each core of the two platforms own a Connectivity Interface for communication reason.

Implementation

The implementation strides to provide an efficient communication between Connectivity Manager (CM) and Connectivity Interface (CI). A Connectivity Manager can be responsible for multiple Connectivity Interfaces and a Connectivity Interface can be responsible for multiple Applications (Tasks). So all the data which should go through the physical line is managed and funneled at the Connectivity Manager, and is then prioritized according to the criticality of the application.

Features to implement:

- **Session management**

  Sessions can be seen as longer lasting connections between applications which are monitored by the Connectivity Manager for bookkeeping reasons.

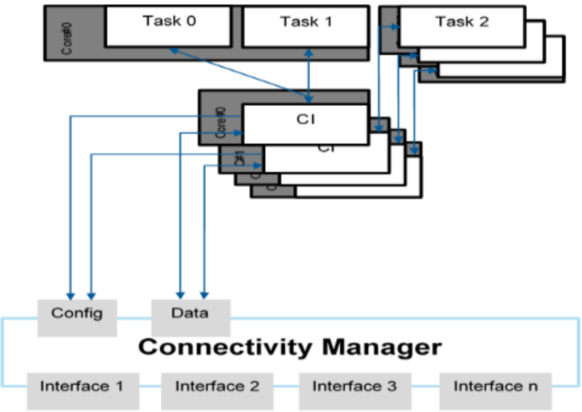


Figure 32: Architecture overview: a system global Connectivity Manager (CM) manages all core-specific Connectivity Interfaces (CI)

- **Traffic prioritization**

  The given link is utilized with respect to the criticality of the applications.

- **Profiling**

  A fine grained profiling of the data streams as well as an analysis of the core interactions is necessary to gain enough information for a load balancing concept.

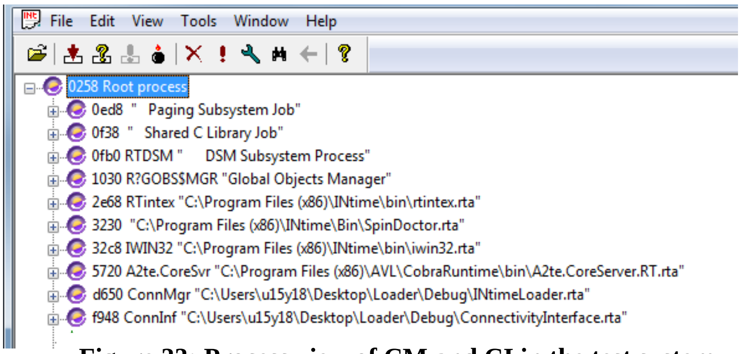
- **Load balancing**

  A concept for load balancing, a separate instance of the Connectivity Manager for each core, based on the profiling information will be elaborated.

Status

We are currently finishing the implementation of a first version on the V&V System, which is based on a x86 industrial-PC hardware. We use INtime™ from Tenasys® as real-time operating system. The prototype consists of

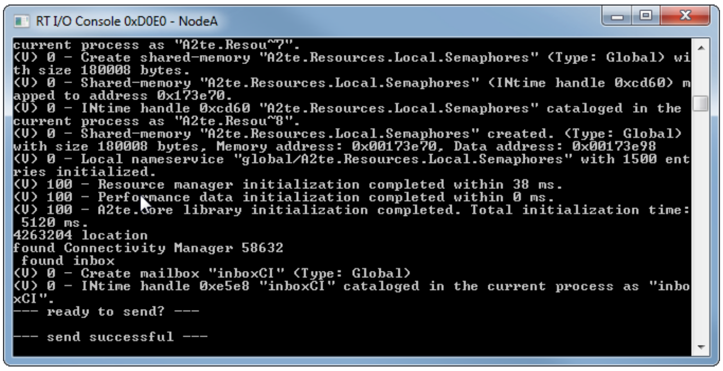
- Independent INtime processes
  - “ConnMgr”
  - “ConnInf”
- Creation of mailboxes for data exchange
- Allocation of shared memory



```
0258 Root process
  0ed8 "   Paging Subsystem Job"
  0f38 "   Shared C Library Job"
  0fb0 RTDSM "      DSM Subsystem Process"
  1030 R?GOBS$MGR "Global Objects Manager"
  2e68 RTintex "C:\Program Files (x86)\INtime\bin\rtintex.rta"
  3230  "C:\Program Files (x86)\INtime\Bin\SpinDoctor.rta"
  32c8 IWIN32 "C:\Program Files (x86)\INtime\bin\iwin32.rta"
  5720 A2te.CoreSvr "C:\Program Files (x86)\AVL\CobraRuntime\bin\A2te.CoreServer.RT.rta"
  d650 ConnMgr "C:\Users\u15y18\Desktop\Loader\Debug\INtimeLoader.rta"
  f948 ConnInf "C:\Users\u15y18\Desktop\Loader\Debug\ConnectivityInterface.rta"
```

Figure 33: Process view of CM and CI in the test system

- CM connects across cores (from an application running on core “A” to the actual networking hardware mapped to core “B”)
- Notifications implemented via mailboxes
    - CI can contact CM
- Data exchange
    - listening for incoming data

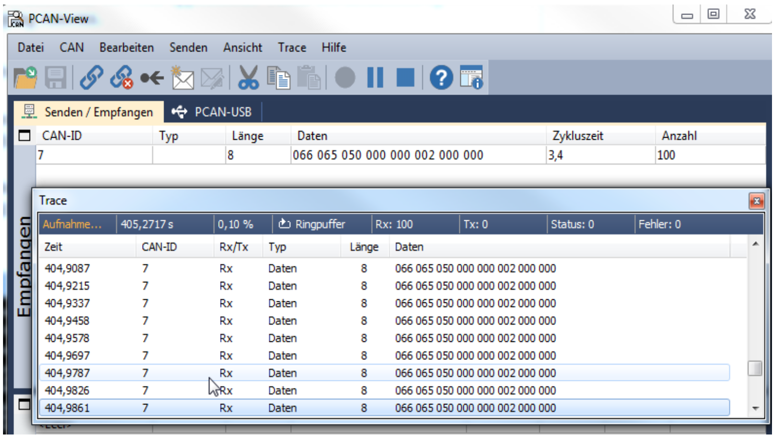


```
current process as "A2te.Resou~7".
<V> 0 - Create shared-memory "A2te.Resources.Local.Semaphores" <Type: Global> wi
th size 180008 bytes.
<V> 0 - Shared-memory "A2te.Resources.Local.Semaphores" <INtime handle 0xcd60> m
apped to address 0x173e70.
<V> 0 - INtime handle 0xcd60 "A2te.Resources.Local.Semaphores" cataloged in the
current process as "A2te.Resou~8".
<V> 0 - Shared-memory "A2te.Resources.Local.Semaphores" created. <Type: Global>
with size 180008 bytes, Memory address: 0x00173e70, Data address: 0x00173e98
<V> 0 - Local nameservice "global/A2te.Resources.Local.Semaphores" with 1500 ent
ries initialized.
<V> 100 - Resource manager initialization completed within 38 ms.
<V> 100 - Performance data initialization completed within 0 ms.
<V> 100 - A2te.Core library initialization completed. Total initialization time:
 5120 ms.
4263204 location
found Connectivity Manager 58632
 found inbox
<V> 0 - Create mailbox "inboxCI" <Type: Global>
<V> 0 - INtime handle 0xe5e8 "inboxCI" cataloged in the current process as "inbo
xCI".
--- ready to send? ---

--- send successful ---
```

Figure 34: Screen shot of a console-based test app

- Data segmentation to CAN messages
- Utilization of a priority queue
    - for messages to be sent
- Ring buffer for incoming messages
- Sending messages via CAN hardware
- Reading messages from CAN hardware
    - piece up relevant messages according to application’s read pointer



CAN-ID	Typ	Länge	Daten	Zykluszeit	Anzahl
7		8	066 065 050 000 000 002 000 000	3,4	100

Aufnahme...	405,2717 s	0,10 %	Ringpuffer	Rx: 100	Tx: 0	Status: 0	Fehler: 0

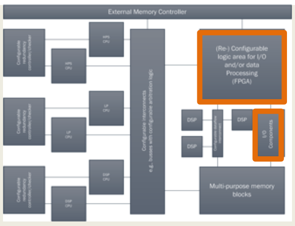
Zeit	CAN-ID	Rx/Tx	Typ	Länge	Daten
404,9087	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9215	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9337	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9458	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9578	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9697	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9787	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9826	7	Rx	Daten	8	066 065 050 000 000 002 000 000
404,9861	7	Rx	Daten	8	066 065 050 000 000 002 000 000

Figure 35: Combined multi-core data traffic via CAN-bus (Bus Analyzer view)

4.7 SevenSols (15O SevenS)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.7.1 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 Y2 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 Y2 SevenS has evaluated the QoS and scalability of the White Rabbit technology and its effects on the 1-PPS signal in terms of jitter (WP4 T4.3). For this study, we have used 14 7S-LEN nodes forming a daisy-chain row and we have performed 3 tests with different components: 12 nodes, 14 nodes, and 12 nodes with a 5 km fiber link (Fig 1, left). As a result, SevenS has achieved very good stability results for the synchronization presenting a standard deviation for the 1-PPS signal jitter under 250ps in all the nodes and maintaining the sub-nanosecond White Rabbit accuracy up to the ~12th level of the chain. Moreover, the VLAN support developed for White-Rabbit devices has been successfully tested within this scenario.

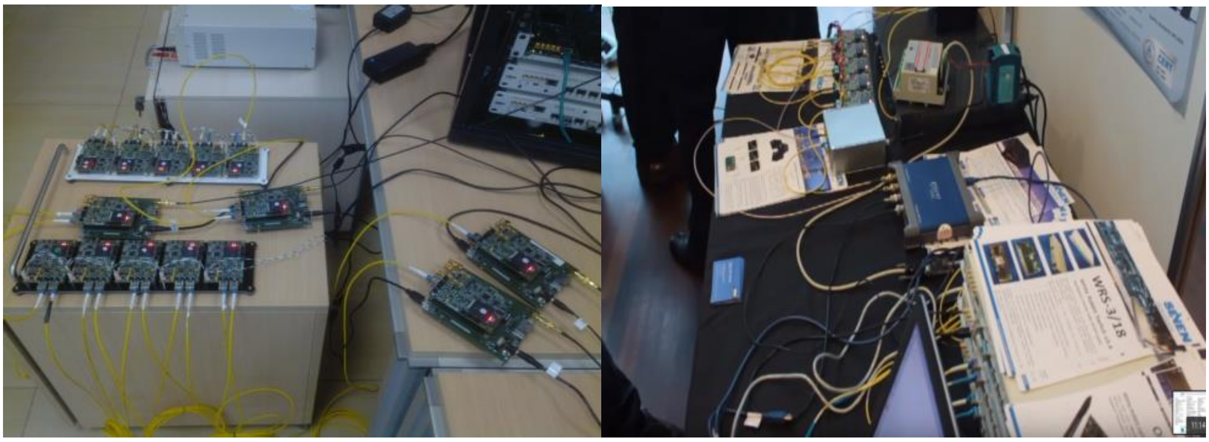


Figure 36: Daisy chain jitter and QoS evaluation using 7S-LEN devices (left). UC5.5 Demo including HSR implementation of the White-Rabbit technology (right)

In addition, a first prototype of a redundant-fault-tolerant timing network using the HSR implementation (WP1 T1.6) compatible with VLANs was presented in last EMC2 workshop in Vienna, where efforts from tasks T.1.1, T1.6, T4.1 and T.4.3 took shape as the WP11 UC5.5 demonstrator: *Synchronized low-latency deterministic networks*.

Finally, during Y3 SevenS plans to finish the development of the HSR protocol for White-Rabbit devices and perform the time accuracy tests of the event-trigger application where QoS and 1-PPS signal jitter will be definitely tested within the UC5.5 demonstrator scenario to ensure synchronization accuracy and performance.

Expected 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.

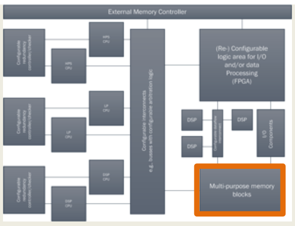
Novel technology in comparison to state-of-the-art:

- 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.
- Based on open specifications with basic open software and open hardware reference designs.

4.8 NXP Germany (01R NXPGE)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
LP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.8.1 **Multi Secure Elements Managing Unit**

The multi-secure-element managing unit is referred to as Multi-secure-element handler (MSEH) internally. NXPGE is responsible for development of multi-core root of trust that provides security services to a network. An MSEH is composed of multiple security elements or cryptographic co-processors that are capable of providing following security services for any network.

- Confidentiality using message encryption/decryption with symmetric (AES) or asymmetric (RSA, ECC) cryptography protocols (i.e.no eavesdropping on the channel)
- Message authentication to prove the identity of sender to receiver (Digital certificates)
- Message integrity which ensures that message is not tampered on the channel (MAC or message authentication codes)
- Non-repudiation or denial of sending a message by client on the network (Digital signature)

A functional block diagram of a network with MSEH as root of trust is shown in Figure 37.

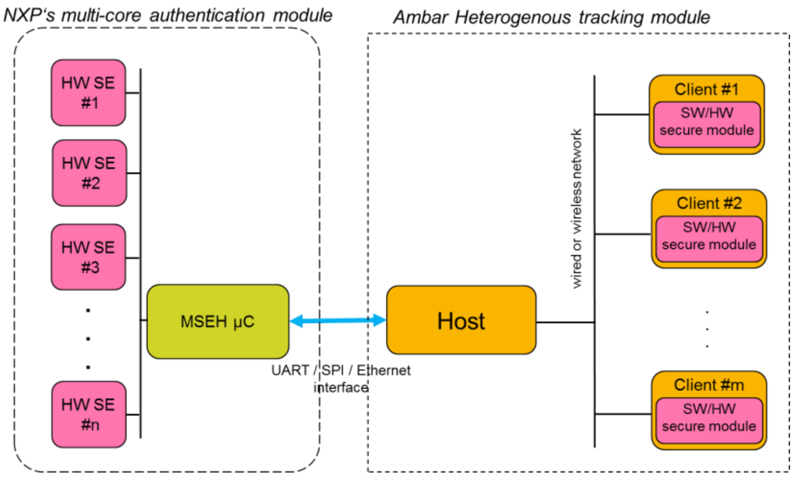

NXP's multi-core authentication module
Ambar Heterogenous tracking module
HW SE #1
HW SE #2
HW SE #3
HW SE #n
MSEH µC
UART / SPI / Ethernet interface
Host
wired or wireless network
Client #1
SW/HW secure module
Client #2
SW/HW secure module
Client #m
SW/HW secure module

Figure 37: Block Diagram of Multi-Secure-Element Root of Trust connected to Network (WP10-LL4 use case 10.2)

The service oriented architecture (SoA) of MSEH abstracts the multiple interconnected SEs to the outside network as a virtual (& single) security device that is interfaced to host using standard communication protocols. It plays a role of networked security module with plug & play functionality. It handles multiple security requests of different kinds from the clients on the network. Individual SEs can be dynamically reconfigured to provide different security services at runtime based on the type & number of requests. Thus increasing the utilization factor of individual SE as per the load. The MSEH increases the security service throughput of the system with multiple SEs servicing request in parallel. The hardware abstraction and redundancy by reconfiguration eases the replacement of individual SEs in case of malfunction without compromising on the system load (MSEH simply reassigns the request of malfunctioning SE to other free & functional SE instead). Based on the number of clients on the host, type of security requirements and desired quality of service, the number of secure elements on handler can be scaled appropriately. Its system architecture is flexible enough to accommodate more secure elements in a hierarchical topology to cater to higher network traffic as shown in Figure 38.

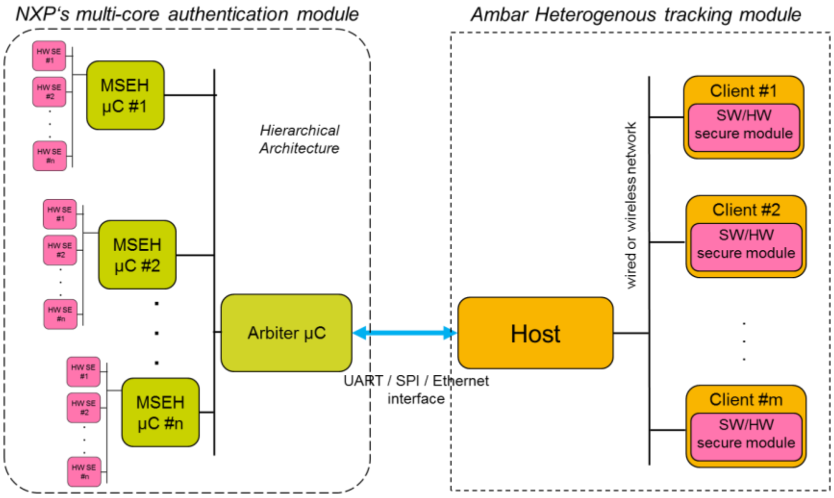

NXP's multi-core authentication module
Ambar Heterogenous tracking module
HW SE #1
HW SE #2
HW SE #n
MSEH µC #1
MSEH µC #2
MSEH µC #n
Hierarchical Architecture
Arbiter µC
UART / SPI / Ethernet interface
Host
wired or wireless network
Client #1
SW/HW secure module
Client #2
SW/HW secure module
Client #m
SW/HW secure module

Figure 38: Block diagram of hierarchical multi secure element root of trust interfaced to host network

The architecture of MSEH fulfills all the strategic targets of Artemis/EMC^2 **[FPP_EMC2_AIPP5]** as described below

- **Reduce system design cost:** The MSEH abstracts group of secure elements as a single module that offers security services to the complete network. Security request can be scheduled, prioritized and arbitrated on demand, enabling re-use of underlying secure elements and thus saving the cost of having individual secure elements for every node on the network. Parallel processing increases throughput on system level without increasing component costs. Usage of off-the-shelf components reduces the design/development cost of having single system on chip (SoC) solution.
- **Reduce efforts & time for re-validation and re-certification of systems:** Individual secure elements have been common criteria certified which reduces the multiple root of trust validation process
- **Manage a complexity with effort reduction:** Flexible design architecture described in Figure 37 and Figure 38 can accommodate variable number of clients on the host network (I.e. scaled proportionately). The plug & play capability of SEs provides configurability to the system by scaling the number of SEs based on the client requests.
- **Achieve cross-sectorial reusability of embedded system devices & architecture platforms:** Any network that needs to authenticate clients on it and safeguard the communication between them, would indispensably need a root of trust like MSEH.

Thus MSEH behaves as a (re)configurable/interoperable, scalable and flexible root of trust to secure a network. Please refer to WP 10 or LL4 use case 10.2 deliverables for detailed description of the multi-core root of trust system. It is currently planned to demonstrate client authentication functionality on any host in the network.

Progress

Del. no.	Deliverable name	Delivery date (proj. month)	Progress
D10.1	Requirements for industrial applications.	M8	done
D10.2	Evaluation of the requirements against the current solutions and the gap analysis.	M12	done
D10.3	First prototype application and the description of the conceptual architecture.	M20	done
D10.4	Evaluation of the first prototype against the requirements.	M24	In progress
D10.5	Completed and final demonstration	M33	-
D10.6	Final report	M36	-

The first prototype is designed as a proof of concept using off the shelf components to demonstrate the functionality of multi-secure element handler. A block diagram representation of the prototype is given in Figure 39.

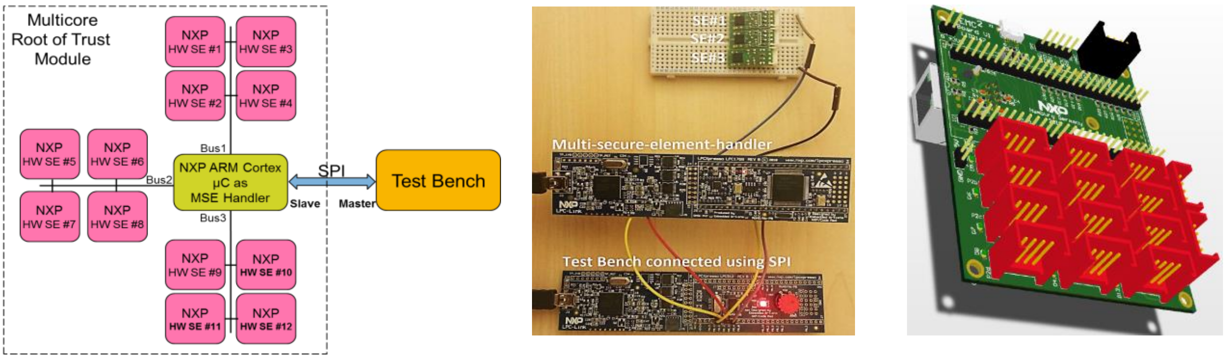
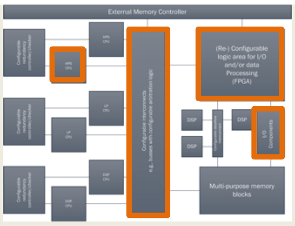


Figure 39: First prototype representation, Off-the-shelf implementation, and development board

4.9 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*


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
LP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.9.1 High speed inter-chip communication

Description

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

Expected 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.

Y2 achievements

- VHDL implementation and pre-silicon verification completed successfully
- Physical implementation and post silicon verification preparation started

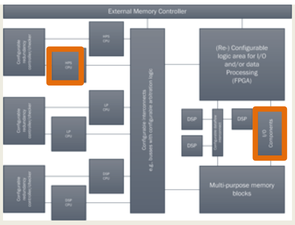
Y3 planning

- Complete physical implementation (Q1)
- Silicon production (Q2)
- Post silicon verification and validation (Q3)
- Validation and measurement report (Q4)

4.10 TU Dortmund (01W TUDO)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g., busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
Configurable dataflow interconnect
DSP
I/O Components
Multi-purpose memory blocks

4.10.1 Peripheral Virtualization Manager (ST4.2.5 and ST4.3.5)

Our virtualization approach assigns applications to distinct criticality domains. The criticality domains are granted access to the peripherals in spatial (register sets) and temporal (time slots) separation via the virtualization interface. A prioritized TDMA arbitration scheme will help guarantee a predictable communication path for Programmable-I/O and DMA. Address translation is transparently performed by the MMU/IOMMU to ease the integration of heterogeneous cores with their corresponding memory maps. The virtualization manager will be implemented in SystemC within the SoCRocket simulation platform.

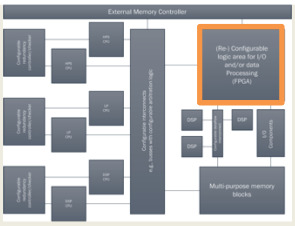
Current Status: Definition of the peripheral virtualization manager interface.

Upcoming in Y3: Analysis of state-of-the-art standard virtualization interfaces for fine-tuning our approach.

4.11 Tecnalia (15E Tecnalia)

Technology lanes covered:

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


External Memory Controller
Configurable redundancy controller/checker
HPS CPU
HPS CPU
Configurable redundancy controller/checker
LP CPU
LP CPU
Configurable redundancy controller/checker
DSP CPU
DSP CPU
Configurable interconnects e.g. busses with configurable arbitration logic
(Re-) Configurable logic area for I/O and/or data Processing (FPGA)
DSP
DSP
DSP
Configurable dataflow interconnect
I/O Components
Multi-purpose memory blocks

4.11.1 Reliable Dynamic Reconfiguration Manager

Dynamic Partial Reconfiguration (DPR) for FPGA devices has many advantages not only for the Space domain but also for many other industrial domains. The use of DPR requires including a module (peripheral) to control the process in a static partition of the FPGA (area that cannot be dynamically reconfigured). Therefore, a reliable reconfiguration manager peripheral specifically protected against radiation induced errors is highly desirable in this environment.

The peripheral will be a Microblaze based embedded system, with most of the functions related to DPR and error mitigation performed by software applications running in the Microblaze processor. The peripheral will also be able to communicate with external elements using a custom defined protocol. Three Fault-tolerant techniques will be included for error mitigation:

- MicroBlaze fault-tolerant features
- Periodical MicroBlaze memory scrubbing done by software
- Partial bitstream data integrity check function

In the reporting period, a low-level definition of the architecture and the communications protocol have been designed. Additionally, a basic HW/SW implementation of the peripheral which includes dynamic reconfiguration capabilities has been completed and partially validated. Work in the period has included the following tasks:

- Design of the communications protocol
- Design of the low level architecture
- Development of the embedded system reliable hardware *(not completed)*
- Development of the communications protocol software implementation *(not completed)*
- Development of the dynamic partial reconfiguration management software

Remaining work for year 3 of the project includes:

- Development of the embedded system reliable hardware *(started in Y2)*
- Development of the communications protocol software implementation *(started in Y2)*
- Development of the memory scrubbing control software
- Validation and tests

Results from this task will be shown in a demonstrator within the Space Living Lab (WP9). The mentioned demonstrator will be implemented using TASE's LADAP hardware platform and will consist on an implementation of the DPR peripheral in Virtex 5 FPGA device of the platform and a control software (developed by TASE) running on the RTAX device. For the demonstration, reconfigurable functionality will include just simple arithmetic operations. The software in the demonstration will control the dynamic reconfiguration features of the peripheral and configure some of the fault-tolerant features. Additionally, the software will have access to fault statistics. Finally, as it is not possible to provoke radiation-induced-errors, errors in the FPGA configuration memory will be “manually” introduced through the software and the peripheral. This last feature needs to be still further elaborated.

5. Conclusions

This document is a revised report on hardware elements in the scope of the EMC2 project. It contains an overview of the technical development of individual contributions presented in the first report marked as the deliverable D4.3: First report on Cores, Memory, Virtualization, Accelerators, Interconnecting Cores, Chips, and Peripheral devices in Mixed-Criticality Systems: Requirements, Concepts and Preliminary Designs.

There is an evident progress in development and high applicability of the respective technologies in diverse demonstrators. Most technologies are well represented in living labs, what confirms the development of practical hardware solutions for problems of mixed-criticality, dynamic reconfiguration, determinism and integration of multi-core architectures in existing systems. The status of the work presented in this report show that all partners are on schedule with their activities, with minor changes and adaptations as described in Section 5.

The core of technological development in WP4 is grouped around hybrid SoC architectures, an architecture that combines classical multi-core CPU with an FPGA device. The flexibility of such platforms is proven to be highly useful and applicable in various industrial applications. Further, strong focus is set on communication methodologies, QoS and security for on- and off-chip exchange of information. Noticeable advancement in the development of reconfigurable architectures and virtualization of peripheral devices. One of the aspects with a large potential is system modeling, of either individual architecture elements (e.g., memory hierarchies) or whole systems as a virtual architecture.

Although most of the work presented in this document is individual, it can be considered as part of a generic architecture molded on requirements form this project and legacy projects. Unification of these different aspects and analysis of their compatibility is a potential topic for further research. In the document, partners define what aspect of the generic architecture they are working on. This has been presented in the heading of each chapter and on Figure 1.

6. References

[1] "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.

[2] P. G. Péron, E. Uhlemann, W. Steiner and M. Björkmann, "A Wireless MAC Method with Support for Heterogeneous Data Traffic," in *41st Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Yokohama, Japan, November 2015.

[3] 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.

[4] UTIA, "EdkDSP," 2015. [Online]. Available: http://sp.utia.cz/index.php?ids=results&id=Utia_EdkDSP_Vivado_2013_4_EMC2..

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

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

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

[8] X. Inc., "Zynq-7000 Silicon Devices," 2014. [Online]. Available: http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/silicon-devices.html. [Accessed 2014].

7. Abbreviations

Table 1: 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
RTAX	High-reliability, radiation-tolerant, antifuse-based FPGAs
DPR	Dynamic Partial Reconfiguration
TDMA	Time division multipole access
DMA	Direct Memory Access