Multi-Core Architecture for AUTOSAR based on Virtual Electronic Control Units

Moisés Urbina, Roman Obermaisser
University of Siegen, Germany

***Abstract*—Message-based Networks-on-a-Chip (NoC) provide significant advantages with respect to temporal predictability, fault isolation and energy efficiency. However, the AUTOSAR standard for multi-core operating systems focuses only on multi-core platforms with shared memories. In order to obtain the benefits of message-based interactions, this paper maps an AUTOSAR system to a multi-core platform with a time-triggered NoC. Cores serve as virtual Electronic Control Units (ECUs), each containing a lightweight AUTOSAR operating system and a Run-Time Environment (RTE). Virtual ECUs provide meaningful units of abstraction and ensure freedom of inference from other cores. Computationally expensive functionality of the basic software is delegated to system cores, which serve as hardware accelerators for the application cores.**

I. Introduction

For the development of embedded systems, such as those in the automotive industry, suitable programming models are needed that support a high degree of concurrency and sensitivity to the needs of embedded systems in terms of reliability, real-time capability, resource efficiency and support for heterogeneous computing cores. Today the paradigm of shared memory is predominant in multi-core architectures. However, a shared memory typically leads to temporal unpredictability, since the access of the cores is not planned and simultaneous memory accesses are resolved dynamically. In addition, memory hierarchies and cache coherence protocols contribute significantly to the temporal unpredictability [1]. In contrast, a message-based on-chip network has significant advantages [2] [3] for embedded real-time systems such as improved fault isolation, real-time support and energy efficiency. As explained in [4] a message-based NoC is superior to a shared memory in case of a high computation/communication ratio, which is typical of automotive electronics. Compared with shared-memory architectures, message-based NoCs eliminate the overhead and the hardware complexity of a protocol for cache coherency [5], offer a better temporal predictability, support the seamless integration of autonomous cores and exhibit higher reliability and energy efficiency.

Due to the increasing importance of Multi-Processors System-on-a-Chip (MPSoC), the automobile industry introduced support for multi-core processors [6] in the last version of the AUTOSAR standard (Version 4). An AUTOSAR multi-core operating system is introduced for managing an MPSoC with multiple cores, which are used for parallel execution of software components (SWC) that interact through a shared memory. However, the multi-core version of AUTOSAR lacks support for message-based MPSoC architectures, therefore the mentioned advantages of message-based NoCs over shared memory cannot be exploited. In particular, temporal predictability has been identified as a weak point in AUTOSAR [7].

In previous work MPSoC platforms based on time-triggered NoCs [8] were introduced that support stringent fault isolation and temporal predictability. A major scientific challenge is the instantiation of the AUTOSAR architecture on such an MPSoC platform. A requirement is the mapping of the Virtual Function Bus (VFB) to an NoC.

Interfaces that can also be bypassed for performance reasons within the AUTOSAR BSW and the possibility of direct access through cross-layer shortcuts represent a risk to the realization of the AUTOSAR Basic Software (BSW) functions on different cores. The limited local memory resources within the cores are another technical risk. Finally, the complete compatibility with existing interfaces and specifications (e.g. AUTOSAR RTE, AUTOSAR OS) is a challenge of central importance.

This paper presents the extension of time-triggered message-based on-chip architectures towards an AUTOSAR-platform with support for reliability and real-time requirements. Autonomous application cores serve as virtual Electronic Control Units (ECUs). The interaction occurs only using messages on the VFB, which are mapped to the time-triggered NoC. Each virtual ECU is a unit of abstraction, where the timely provision of message-based services can be analyzed and understood independently from the other virtual ECUs. Each virtual ECU has its own basic software and there are no hidden interactions between the BSW of different virtual ECUs. Specific system cores serve as hardware accelerators (e.g., off-chip networks, I/O gateway, etc). Appropriate system models are defined for extending the AUTOSAR BSW towards a hierarchical platform comprising:

- *Communication between SWCs on the same application core* [3].
- *Message-based on-chip communication.* Enabling communication between application cores on the same MPSoC [2].
- *Message-based off-chip communication.* Enabling inter-communication between different MPSoCs.

This proposed AUTOSAR time-triggered multi-core platform offers significant advantages in comparison with the existing AUTOSAR multi-core operating system version.

- *Better reliability.* The Time-Triggered Network-on-a-Chip (TTNoC) and the autonomous virtual ECUs provide fault isolation in the time and value domains. Fault isolation is a weak point in the AUTOSAR multicore operating system [9].
- *High performance with low overhead.* The parallel execution of SWCs on multiple cores, system cores with

978-1-4673-7929-8/15/$31.00 © 2015 European Union

hardware support for gateways, I/O and memory bus and simplified BSW increase the performance and reduce the resource requirements in the AUTOSAR layer model.

- *Temporal predictability.* The TTNoC and the decoupling of the cores lead to deterministic temporal behavior and suitability for real-time applications.
- *Smooth integration.* Autonomous virtual ECUs (application cores) lead to clear responsibility and support for seamless integration of independently developed SWCs.
- *Seamless communication with SWCs on other ECUs.* Gateway system cores support seamless communication between the chip-level and distributed systems, especially with the support of time-triggered off-chip networks such as FlexRay [10].

The remainder of this paper is organized as follows. Section II provides an overview of related work in the area of MPSoC. In Section III we present the multicore background on AUTOSAR. The model of the proposed AUTOSAR MPSoC platform is presented in Section IV. The paper finishes with a discussion and future work in Section V.

II. Related Work

Over the years, the construction of embedded systems in different application domains has resulted in developed domain-specific system architectures. For example, Automotive Open System Architecture (AUTOSAR) [11] , which is the predominant standard in the automotive domain, Integrated Modular Avionic (IMA) [12] for the aerospace domain and Network on Terminal Architecture (NoTa) in the mobile domain [13].

Furthermore, different MPSoC architectures have been developed for specific application domains (e.g. CellBE [14], AEthereal [15], Sonics [16], CoMPSoC [17], Nostrum [18], and Spidergon [19]). All of them implement a shared memory approach via transaction-based mater/salve protocol for the communication between cores.

Within the European embedded systems initiative ARTEMIS [20] common challenges were identified on specific-domain architectures (Composability, predictability, robustness, security). Driven by these domain-independent challenges, the European research project GENESYS [8] developed a MPSoC reference architecture blueprint that can be universally implemented on different application domains as automotive, avionic, industrial control, mobile and consumer electronic systems. The European ARTEMIS ACROSS [21] project developed a MPSoC architecture which was specifically designed for high safety-critical embedded system applications. These architectures implement a TTNoC for the communication between cores. Additionally, the benefit of the implementation of such an MPSoC architecture specifically on the automotive domain is addressed in the state-of-the-art [4].

III. Background on AUTOSAR

AUTOSAR [11] is an open standard architecture for the automotive domain that supports the integration of multiple application subsystems in a distributed hardware platform with the potential to reduce the number of ECUs, to reduce wiring and to introduce enhancements in the scalability, flexibility and reliability. AUTOSAR supports the reuse of SWCs and provides technology-independent platform services for several automotive platforms to original equipment manufacturers (OEMs) and suppliers.

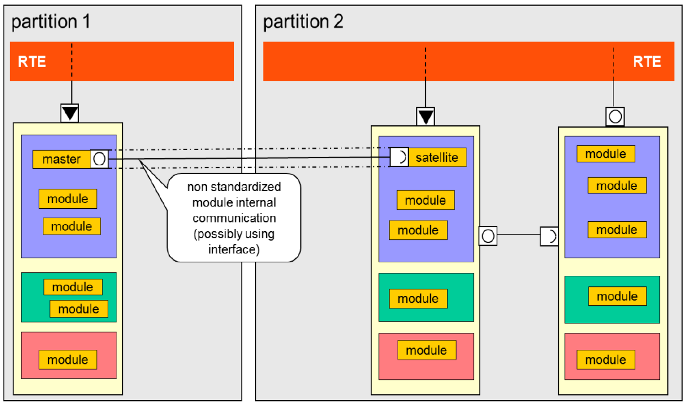

Fig. 1: Standardized functional clusters of the same type [6]

Version 4 of the AUTOSAR standard specifies an AUTOSAR multicore operating system [6], and supports multicore processors with shared memory. The AUTOSAR cores share the same configuration and code, while different data is processed. A distinction is made between a master core and slave cores (see Figure 1).

The complete AUTOSAR BSW is provided only on the master core, while the slave cores have a subset of the AUTOSAR BSW to support SWCs and complex device drivers. The communication between the operating system applications (Inter-OS Application Communication (IOC)) is used for the communication between cores. This type of communication is implemented by shared memory [6] and is used by the AUTOSAR RTE for the communication between SWCs on different cores. The sender/receiver communication is the only communication type supported, while the client/server communication is not provided. Also, "*1:N*" communication is not directly supported but implemented with multiple interactions. For synchronization between cores busy-waiting with spinlocks is used. Configuration tools need to check offline that no deadlocks occur when using the spinlocks.

The AUTOSAR multicore operating system lacks support for message-based MPSoC architectures, whereby the advantages of this architecture model for temporal predictability, fault isolation, low overhead, energy efficiency and scalability for AUTOSAR-based systems can't be realized. Furthermore, the AUTOSAR multicore operating system gives a limited autonomy to the cores, which are managed by a common operating system. Dependencies between cores, for example, by the possibility of activating tasks, task chaining or setting alarms/events on other cores, can lead to error propagation between cores in case of permanent and transient hardware errors (e.g. soft errors) and design errors. These dependencies require the analysis of the temporal behavior of each core under the influence of the other cores on the MPSoC. The starting and stopping of individual cores is not supported, and only the complete MPSoC can be put into "*sleep mode*". Finally, the master core represents a single point of failure and provides limited scalability.

IV. Message-Based Time-Triggered Multi-Core Platform for AUTOSAR

The mapping of the AUTOSAR architecture to a time-triggered multi-core platform is depicted in Figure 2. Two types of cores are distinguished: application cores and system cores. The time-triggered NoC is used for the communication between application cores and system cores.

A. Application Cores / Virtual ECUs

Application cores implement the required automotive functionality by hosting one or more AUTOSAR SWCs. The application cores represent virtual ECUs in the MPSoC platform.

Each application core is provided with a middleware implementing the AUTOSAR RTE. The purpose of the RTE as defined by AUTOSAR is to provide an interface to the SWCs that makes them independent from the underlying platform and from the mapping to a specific ECU hardware [22]. SWCs can access via the RTE the following AUTOSAR services [23]: memory services (nvram manager), system services, communication services (e.g., CAN, LIN, FlexRay), input/output hardware, and complex drivers.

In the proposed AUTOSAR MPSoC, the RTE in the application cores offers access to system services (i.e., AUTOSAR OS) and communication services. The communication services are mapped to the NoC transparently to the application software. This provided version of the AUTOSAR RTE serves as the only interface for the interaction between SWCs on different application cores. The application cores require only a reduced realization of the AUTOSAR BSW, wherein BSW modules for memory services, I/O hardware abstraction and complex drivers are replaced by stubs to dedicated system cores. The motivation behind this approach are the improved performance and lower overhead for AUTOSAR in the virtual ECUs since the system cores act as hardware accelerators.

Furthermore, BSW modules in the "*ECU Communication Abstraction Layer*" are added for supporting access to the TTNoC (see Figure 3). In contrast, BSW communication modules for network off-chip accesses are also allocated to dedicated system cores.

The VFB defined by the AUTOSAR standard is available for the AUTOSAR SWCs through a hierarchical platform. The communication hierarchy is performed as follows:

- *Inner-core communication:* Message passing between SWCs on the same core is performed via the BSW within the virtual ECU.
- *Communication with SWCs on different application cores on the same MPSoC:* For this purpose, the AUTOSAR COM is extended to route messages through the NoC.
- *Inter-communication between SWCs on different MPSoCs:* For this purpose a message is sent to the off-chip gateway, which is a system core.

The advantages of this approach are the reduced requirements on OS functionality within a single application core as well as the inherent encapsulation of the virtual ECUs using the TTNoC.

B. System Cores

The system cores are deployed to implement parts of the AUTOSAR basic software. System cores implement particular AUTOSAR functions such as gateways or input/output functions. The system cores perform visualization of specific resources, e.g., ensure absence of interference on I/O or NVRAM. The access to the system cores is mapped to message-based on-chip communication. At the system cores a modified implementation of the system services layer is required to process those requests from the BSW on the application cores, integrating an interface module to interact with the deterministic on-chip-network. For all lower layers of the AUTOSAR Layered Software Architecture, the specified AUTOSAR interfaces stay equal.

The introduction of these system cores in the multi-core AUTOSAR platform is expected to radically simplify the AUTOSAR BSW in the virtual ECUs. The advantage of these dedicated system cores are simpler application cores, since they are tailored to the needs of the actual application only, and the potential of reuse for such system cores in different designs.

Identified system cores are described as follows.

- *Off-chip network gateway core.* This gateway core supports the data exchange between the TTNoC and the off-chip automotive networks. The gateway core has, in addition to the communication hardware abstraction for the TTNoC, also software modules and interfaces for accessing off-chip networks such as FlexRay, LIN or CAN. Thus, the communication between the SWCs in different MPSoCs is possible. Figure 2 depicts the realization of the off-chip gateway functionality in distinct system cores for each communication protocol instead of a communication middleware in each application core.
- *Input/output core.* AUTOSAR adopts a layer model to provide a uniform access to input/output devices, where at the microcontroller abstraction level, basic drivers for analog and digital input/output (e.g., pulse width modulation, capturing of input signals and analog to digital conversion) are provided. An input/output core provides hardware support for the realization of the input/output hardware abstraction services.
- *Memory access core.* The NVRAM management functionality is delegated to a dedicated system core. This system core is connected to a chip-external memory component (e.g., flash memory) on the ECU. The memory access core implements the functionality of the memory hardware abstraction of the ECU abstraction layer. It provides an abstraction from the type and location of the specific memory. A flash or EEPROM memory is accessible via the same interface.
- *Complex drivers core.* AUTOSAR defines the purpose of complex drivers as an interface for complex sensors or actuators, which have stringent requirements on timing and are implemented for specific designs. A complex driver (e.g., for a fast ADC) implemented as a system core highly increases the efficiency of the specific driver functionality.

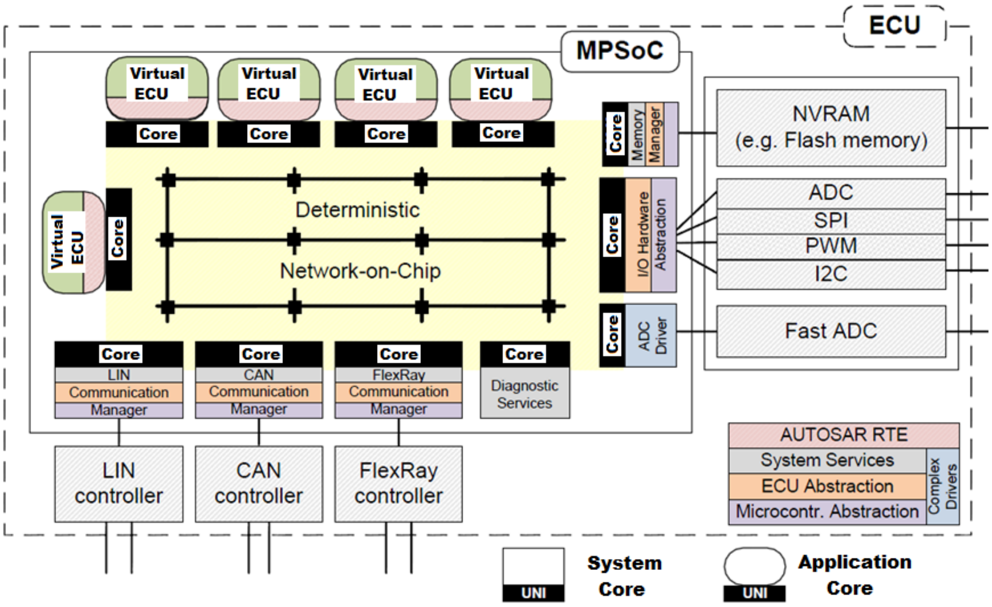

Fig. 2: Mapping of AUTOSAR Architecture to a Time-Triggered Multi-Core Platform

C. Time-Triggered Network-on-a-Chip

The TTNoC provides configurable topologies with a global time base for the temporal coordination of the message transmissions between the cores. The TTNoC contains several *time-triggered channels* and *network interfaces*, one for each application core. For the execution of the TTNoC, the configuration parameters of the required topology have to be defined at design time. This priori knowledge is used later to setup the TTNoC and define its time-triggered channels, network interfaces and the time-triggered schedule as will be described later.

The TTNoC consists of multiple time-triggered channels to transfer the messages based on their predefined source-based routes. Time-triggered channels allocate the resources for the transmitted messages to guarantee temporal and spatial partitioning. Partitioning is done through the use of the a priori knowledge of the permitted communication behavior.

In the TTNoC, the a priori knowledge defines also the communication schedule of the messages on the top of the TTNoC. It is the responsibility of the network interfaces to transmit the messages using the correct time-triggered channel according to the communication schedule. The schedule also defines the source application core and its output port towards the corresponding channel that leads to the destination application cores.

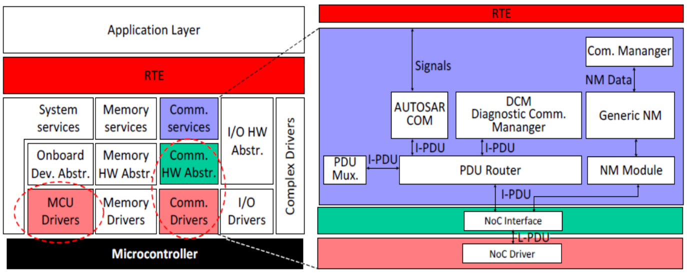

Fig. 3: BSW Modules for TTNoC Communication

Network interfaces have a well-defined temporal behavior and offer a standard interface to the application cores with incoming and outgoing message ports. They process incoming messages from the TTNoC and autonomously send messages at the scheduled points in time to the TTNoC.

Application cores have pre-assigned time slots based on a global time base to send their time-triggered messages. *Time-triggered messages* have the following configuration information: {period, phase, sender node ID, and receiver nodes IDs}. These configuration parameters represent the a priori knowledge that the network interface exploits to isolate faulty applications and to prevent interference between the application cores. Application cores provide data for time-triggered messages, but they do not influence the scheduled transmission times.

V. Discussion / Conclusion

The presented AUTOSAR multi-core platform offers important benefits over the multi-core version provided by the AUTOSAR standard, which can be identified as follows.

- *Autonomous application cores in the role of virtual ECUs.* Each application core uses a local AUTOSAR operating system, which is solely responsible for the management of the resources of the application core. The elimination of hidden dependencies between cores avoids error propagation and facilitates the integration of independently developed and validated cores. Therefore, the AUTOSAR BSW on the application cores is adapted to support the communication through the TTNoC.
- *Interaction between cores using a TTNoC.* The paradigm of time-triggered control leads to significant advantages (e.g. temporal predictability, fault isolation, higher reliability) [2], [3]. These advantages are also achieved by time-triggered off-chip communication systems in the

automotive domain such as FlexRay [10]. The TTNoC of the proposed AUTOSAR MPSoC platform realizes these benefits at the chip level and provides high temporal predictability, inherent fault isolation and a global time base. Thus, an interface module for the TTNoC is defined for connecting higher layers of the AUTOSAR BSW (e.g. PDU router) with the TTNoC.

- *AUTOSAR-specific system cores.* The AUTOSAR MPSoC platform delegates costly functions of the AUTOSAR BSW to system cores to improve the performance of the execution environment by acceleration through dedicated hardware and reduction of the operating system overhead on the application cores. Specific specialized system cores such as gateway cores, input/output cores and memory cores are defined.
- *Simplified AUTOSAR BSW.* A reduced AUTOSAR BSW is used on the application cores that uses the functions of the system cores. In contrast, the AUTOSAR multi-core operating system leads to higher complexity of the AUTOSAR BSW, since more cores are managed and additional synchronization and communication mechanisms are implemented via shared memory.
- *Efficient implementation of drivers*: For I/O hardware, sensors, actuators and in particular for realizing complex drivers an efficient implementation with system cores is possible. For instance, for dedicated peripherals that have stringent time constraints a hardware acceleration of a device driver is possible. The choice whether a driver would be accelerated in hardware is transparent to the RTE and the AUTOSAR applications.

A. Future Work

Future work will continue to focus on the experimental evaluation of the proposed AUTOSAR MPSoC platform. A simulation environment for the AUTOSAR time-triggered MPSoC platform will serve as a starting point for the evaluation of the advantages described in section I in comparison with the existing AUTOSAR OS.

Based on a realistic test application, the performance, real-time capability and the reliability of the AUTOSAR MPSoC platform will be examined. It is planned to analyze the bandwidth, the delay time and the temporal variation (”jitter”) of the communication between the SWCs. The examination will take place both between cores on the same MPSoC, and cores on different MPSoCs using the off-chip network gateway core. In order to demonstrate the higher reliability, fault injection experiments will be undertaken to verify the fault isolation in design or hardware failures of individual cores.

Acknowledgments

This work has been supported by the EU project DREAMS (No. 610640)

References

[1] Y. Li, V. Suhendra, Y. Liang, T. Mitra, and A. Roychoudhury, “Timing Analysis of Concurrent Programs Running on Shared Cache Multi-Cores,” in *Real-Time Systems Symposium 30th IEEE*, Dec 2009.

[2] H. K. R. Obermaisser and C. Paukovits, “A Cross-Domain Multi-Processor System-on-a-Chip for Embedded Real-Time Systems,” *IEEE Transactions on Industrial Informatics*, 2010.

[3] F. Poletti, A. Poggiali, D. Bertozzi, L. Benini, P. Marchal, M. Loghi, and M. Poncino, “Energy-Efficient Multiprocessor Systems-on-Chip for Embedded Computing: Exploring Programming Models and Their Architectural Support,” *Computers, IEEE Transactions on*, 2007.

[4] B. Huber and R. Obermaisser, “An ARTEMIS Cross-Domain Embedded System Architecture and Its Instantiation for Real-Time Automotive Applications,” in *Proceedings of the 30th IFAC Workshopon Real-Time Programming and 4th International Workshop on Real-Time Software*, Poland, 2009.

[5] J. Leverich, H. Arakida, A. Solomatnikov, A. Firoozshahian, M. Horowitz, and C. Kozyrakis, “Comparing Memory Systems for Chip Multiprocessors,” in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA ’07. New York, NY, USA: ACM, 2007, pp. 358–368. [Online]. Available: http://doi.acm.org/10.1145/1250662.1250707

[6] *AUTOSAR Guide to Multi-Core Systems, AUTOSAR Release 4.1*, AUTOSAR, 2014.

[7] M. Rudorfer, T. Ochs, P. Hoser, M. Thiede, M. Missmer, O. Scheickl, and H. Heinecke, “Realtime system design utilizing AUTOSAR methodology,” in *elektroniknet*, 2009.

[8] R. Obermaisser and H. Kopetz, *GENESYS: A Candidate for an ARTEMIS Cross-Domain Reference Architecture for Embedded Systems*. Sdwestdeutscher Verlag fr Hochschulschriften, 2009.

[9] S. F. D. Bertrand and Y. Trinquet, “An analysis of the AUTOSAR OS timing protection mechanism in Emerging Technologies and Factory Automation,” *ETFA 2009. IEEE Conference on*, 2009.

[10] *FlexRay Communications System Protocol Specification Version 2.1*, FlexRay Consortium. BMW AG, DaimlerChrysler AG, General Motors Corporation, Freescale GmbH, Philips GmbH, Robert Bosch GmbH, and Volkswagen AG., May 2005.

[11] *AUTOSAR Architecture Overview, AUTOSAR Release 4.1*, AUTOSAR Consortium, 2014.

[12] ARINC, *Specification 651: Design Guide for Integrated Modular Avionics*, Aeronautical Radio, Inc., 2551 Riva Road, Annapolis, Maryland 21401, 1991.

[13] K. Kronlof, S. Kontinen, I. Oliver, and T. Eriksson, “A method for mobile terminal platform architecture development,” *Advances in Design and Specification Languages for Embedded Systems*, pp. 285–300, 2007.

[14] IBM, Sony, and Toshiba, *Cell broadband engine architecture*, Tech, 2006.

[15] K. Goossens, J. Dielissen, and A. Radulescu, “Aethereal network on chip: concepts, architectures, and implementations,” *Design Test of Computers, IEEE*, vol. 22, no. 5, pp. 414–421, Sept 2005.

[16] Sonics, *Sonics u network technical overview*, 2002.

[17] A. Hansson, K. Goossens, M. Bekooij, and J. Huisken, “CoMPSoC: A Template for Composable and Predictable Multi-processor System on Chips,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 14, no. 1, pp. 2:1–2:24, Jan. 2009. [Online]. Available: http://doi.acm.org/10.1145/1455229.1455231

[18] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, “The Nostrum backbone-a communication protocol stack for Networks on Chip,” in *VLSI Design, 2004. Proceedings. 17th International Conference on*, 2004, pp. 693–696.

[19] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra, “Spidergon: a novel on-chip communication network,” in *System-on-Chip, 2004. Proceedings. 2004 International Symposium on*, Nov 2004, p. 15.

[20] *ARTEMIS Final Report on Reference Designs and Architectures Constraints and Requirements*, ARTEMIS (Advanced Research and Technology for Embedded Intelligence and Systems), 2006.

[21] C. Salloum, M. Elshuber, O. Hoftberger, H. Isakovic, and A. Wasicek, “The ACROSS MPSoC – A New Generation of Multi-core Processors Designed for Safety-Critical Embedded Systems,” in *Digital System Design (DSD), 2012 15th Euromicro Conference on*, Sept 2012, pp. 105–113.

[22] H. Heinecke, J. Bielefeld, K. Schnelle, M. Maldener, H. Fennel, O. Weis, T. Weber, L. Ruh, J. Lundh, T. Sandn, P. Heitkmper, R. Rimkus, J. Leflour, A. Gilberg, U. Virnich, S. Voget, K. Nishikawa, K. Kajio, T. Scharnhorst, and B. Kunkel, “AUTOSAR Current results and preparations for exploitation,” in *EUROFORUM ”Software in the vehicle”*, 2006.

[23] *AUTOSAR Layered Software Architecture, AUTOSAR Release 4.1*, AUTOSAR, 2014.