A Gateway Core between On-chip and Off-chip Networks for an AUTOSAR Message-based Multi-core Platform

Moisés Urbina and Roman Obermaisser,
University of Siegen, Germany
{moises.urbina, roman.obermaisser}@uni-siegen.de

Abstract

The instantiation of the AUTOSAR architecture on a message-based Multi-Processor System-on-Chip (MPSoC) platform allows to realize the advantages of NoC architectures in an AUTOSAR system. The TIme-triggered MEssage-based MPSoC platform for AUTOSAR (TIMEA) presents application cores as autonomous virtual AUTOSAR ECUs that act as meaningful units of abstraction with freedom of inference from other cores. Additionally, the TIMEA platform provides the possibility to replace functionalities of the Basic Software (BSW) by stubs to dedicated system cores which serve as hardware accelerators to the virtual ECUs. This papers presents a system core supporting the communication between on-chip and off-chip networks for the TIMEA platform. The mentioned system core works as a gateway used by the virtual ECUs for the exchange of messages with virtual ECUs located on a different MPSoC. The gateway core is implemented in a simulation environment which supports the simulation of the AUTOSAR-based software, the natural environment and the on/off-chip network communication. A realistic test application is used for its evaluation.*

1 Introduction

In the last years several Multi-Processors System-on-a-Chip (MPSoC) architectures have been developed for specific application domains (e.g. CellBE [1], Sonics [2]). Additionally, the paradigm of message-based Network-on-Chips (NoCs) has been used for the development of predictable multi-core platforms (e.g. ACROSS [3], GENESYS MPSoC [4], CoMPSoC [5]). These platforms provide a NoC for inter-core communication and implement time-triggered communication to ensure fault isolation.

On the other hand, the automobile industry is increasing the use of MPSoCs for the development of automotive embedded systems. The AUTOSAR (Automotive Open System Architecture) standard proposes a multi-core version [6] of its architecture since the publishing of the version 4. However, the multi-core version of AUTOSAR lacks support for message-based MPSoC architectures, therefore the advantages (e.g., temporal predictability, fault isolation) of NoCs [7] [8] over shared memories cannot be exploited. In particular, temporal predictability has been identified as a weak point in AUTOSAR [9].

In previous work [10] we introduced the model of a TIme-triggered MEssage-based multi-core platform for AUTOSAR (TIMEA). This platform presents application cores playing the role of virtual AUTOSAR ECUs and system cores implementing parts of the AUTOSAR Basic Software (BSW). The virtual ECUs require only a reduced realization of the AUTOSAR BSW, whereas BSW modules for memory services, I/O hardware abstraction and complex drivers are replaced by stubs to dedicated system cores. Furthermore, the inter-core communication is performed through a message-based NoC and therefore the benefits of NoC architectures can be realized with AUTOSAR.

The contribution of this paper is a dedicated core for the TIMEA platform which provides gateway functionalities between the on-chip network and the off-chip automotive networks supported by AUTOSAR. While the message-based NoC is used by the Software Components (SWCs) running in different cores in the same MPSoC, the gateway core allows the communication between SWCs running on different MPSoCs. Thus, the Virtual Functional Bus (VFB) provides location transparency to the SWCs in different MPSoCs.

This gateway core provides a frame-based gateway functionality at the PDU level. While the virtual AUTOSAR ECUs have only the protocol stack for communication on the time-triggered NoC, the mentioned gateway system core has the protocol stack for communication with a specific off-chip network (e.g. FlexRay). Also, it allows rate conversion to compensate the significant differences in bandwidth between off-chip networks and the message-based NoC.

A simulation environment is used for the implementation and evaluation of the gateway core. An Anti-lock Braking System (ABS) consisting of two MPSoCs serves for the evaluation.

The remainder of this paper is structured as follows. An overview of the TIMEA platform is the topic of section 2. In section 3 the gateway core for off-chip communication is presented. Section 4 explains the implementation of the gateway and its evaluation using a simulation scenario. The paper finishes with a conclusion in section 5.

*This work has been supported in part by the EU project DREAMS (No. 610640). Furthermore, gratitude to the dSpace Company for their software support.

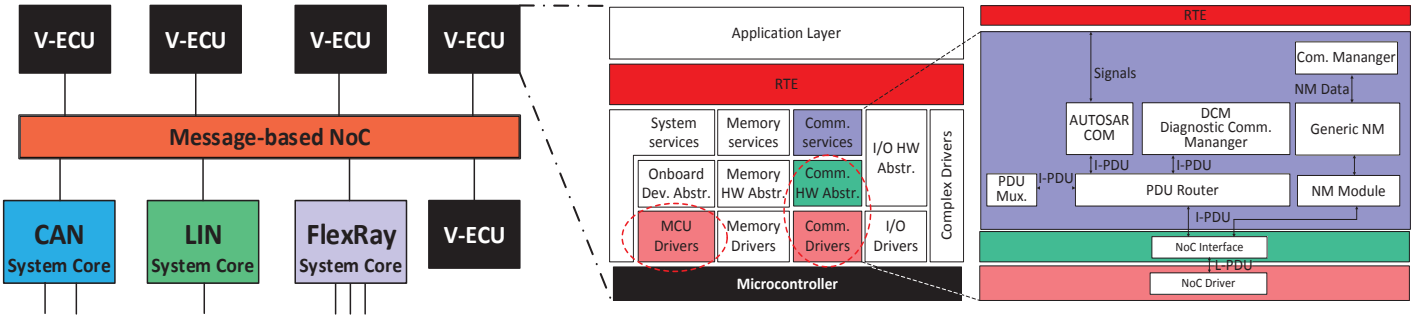

Figure 1 *TIMEA Platform*

2 TIMEA Platform Overview

As presented in [10], the TIMEA platform maps the AUTOSAR ECU architecture to a time-triggered on-chip platform. Application cores act as virtual AUTOSAR ECUs, each one of them provided with its own application layer (i.e., AUTOSAR SWCs), Run Time Environment (RTE) and a modified implementation of the AUTOSAR BSW. A message-based NoC is available for on-chip communication between the virtual ECUs. Additionally, the TIMEA platform suggests the introduction of system cores which act as hardware accelerators for the virtual ECUs.

The motivation behind this approach is an AUTOSAR multi-core architecture that exploits the benefits provided by NoC architectures (e.g. temporal predictability, fault isolation, energy efficiency). Figure 1 depicts the TIMEA platform including the internal structure of a virtual ECU.

2.1 Virtual AUTOSAR ECUs

In the TIMEA platform virtual ECUs are required to implement the automotive functionality. These virtual ECUs are configured based on the AUTOSAR ECU architecture with extended COM service modules for supporting message-based NoC communication. Additionally, a NoC interface hardware abstraction module allows the access from the virtual ECU to the NoC. As in a typical automotive distributed system, where ECUs use an off-chip network for the communication between each other, in the TIMEA platform virtual ECUs implement the message-based NoC for the interaction with other virtual ECUs in the same MPSoC.

As shown in figure 1, a new COM module and a PDU router are integrated into the BSW in the virtual ECUs. The COM module matches RTE SWC signals to Protocol Data Units (PDUs) and vice versa, while the PDU router is in charge of routing COM PDUs to the NoC interface and redirecting NoC PDUs to the COM module. Moreover, the NoC interface is in charge of the exchange of data between the message-based NoC and the BSW in the virtual ECU.

2.2 Introduction of a System Core for Off-Chip Communication

The TIMEA platform allows the introduction of system cores which are deployed to replace parts of the AUTOSAR BSW in the virtual ECUs. Based on the message-based NoC communication, system cores implement particular AUTOSAR functions such as gateways or input/output functions and use the NoC for the interaction with the virtual ECUs. The following system cores can be distinguished:

- *Off-chip network gateway core.*
- *Input/output core.*
- *Memory access core.*
- *Complex drivers core.*

The advantage of the implementation of such a system core is the ability to simplify the AUTOSAR OS in the virtual ECUs, since a computationally expensive functionality of the BSW is delegated to a dedicated core. Furthermore, there is potential of reuse for such system cores in different designs.

The focus of this paper is the development of an off-chip network gateway core which provides support for off-chip communication in the TIMEA platform.

This gateway core supports the data exchange between the message-based NoC and the off-chip automotive networks. Thus, the communication between the SWCs in different MPSoCs is possible. Figure 1 depicts the realization of the off-chip gateway functionality in distinct system cores for each communication protocol instead of an off-chip communication middleware in each virtual ECU.

3 Off-Chip Network Gateway Core

The off-chip gateway system core provides support for off-chip communication to the virtual ECUs in the MPSoC. In figure 2 the software architecture of the presented off-chip network gateway core is depicted. Three layers compose the gateway core architecture: a communication service layer, a hardware-abstraction layer and the Micro Controller-Abstraction Layer (MCAL) for the communication drivers.

As shown in the software architecture of the virtual ECUs (see figure 1), the gateway core requires an implementation of the NoC interface at the hardware-abstraction level in order to be able to access the message-based NoC for on-chip communication. Besides the NoC interface, communication hardware-abstraction modules for supporting off-chip communication are also integrated. Depending on the off-chip communication network to be supported by the gate-

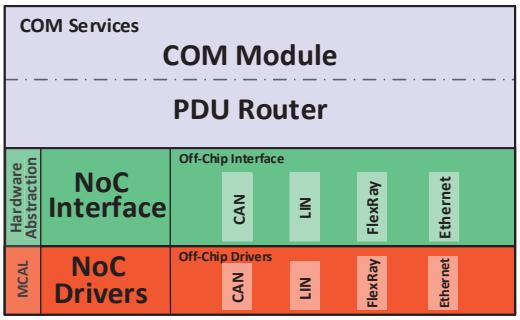

Figure 2 *Architecture of the Off-Chip Network Gateway Core*

way, specific AUTOSAR BSW hardware-abstraction modules of the specific communication bus are selected for the implementation.

Based on the integration of the off-chip gateway core into the TIMEA platform, the AUTOSAR VFB is kept available to the SWCs for transparent communication between each other defining a communication hierarchy that is structured as follows.

- *Inner-core communication:* The communication between SWCs in the same virtual ECU is performed through the RTE contained in the virtual ECU.
- *Inter-core communication:* The extension of the BSW for NoC communication allows the communication between SWCs allocated to different virtual ECUs in the same MPSoC using the message-based NoC.
- *Communication between different MPSoCs:* Communication between SWCs allocated to different virtual ECUs in different MPSoCs is possible due the integrated gateway core supporting off-chip network communication.

  In the rest of this section the modules of the communication service layer are explained.

3.1 COM module

In the AUTOSAR architecture the COM module [11] works as an interface between the RTE and the PDU router. Its main function is the mapping of the RTE signals to PDUs and vice versa. A PDU entity is a data structure that contains the properties of a message as well as its content. Moreover, the implementation of the COM module is independent of the communication protocol used by the ECU for off-chip communication.

For the off-chip gateway core, the COM module is in charge of receiving PDUs from the the PDU router and to make use of a look up table to assign incoming NoC PDUs with off-chip PDUs and vice versa. The format of the PDUs sent through the off-chip network depends of the implemented communication protocol as explained in [11], while for NoC PDUs a format description is found in [12].

Additionally, a message-based NoC supports bandwidths of several *Gbps* and supports communication plans with precise phase positions of messages in the range of a few *ns*. Since the bit rate of off-chip networks is typically lower by several orders of magnitude in comparison to a message-based NoC, the COM module is also used to change the sending time and cycle time for providing rate conversion support to the gateway. This is done by using the transfer property of the PDUs. This property defines whether the PDU should be transmitted whenever its message content changes or if the PDU is sent periodically with the most recent message stored into it.

3.2 PDU Router

A typical PDU router implements routing tables and a router engine for routing and transferring PDUs from the COM module to the communication hardware-abstraction, which transforms them into an actual message that is then forwarded to the MCAL in order to be sent over the communication hardware provided by the ECU. As defined in [13], the purpose of the PDU router is to statically route PDUs based on their PDU identifier and to avoid any dynamic routing at run-time. Just as the COM module, the PDU router has the same design independent of the off-chip communication protocol.

In the off-chip gateway core the PDU router is extended to support the routing of PDUs coming from the message-based NoC. With this purpose, it contains extended routing tables to forward not just PDUs between the COM module and the off-chip interface in the gateway but also for PDUs between the COM module and the NoC interface module.

4 Implementation and Evaluation

The simulation environment presented in [12] was extended for the implementation of the off-chip gateway core. This co-simulation environment consists of an AUTOSAR-based system simulator for the simulation of the AUTOSAR-based software, the physical environment and the off-chip communication level, and an on-chip system simulator for the simulation of the on-chip communication level. The architecture of the co-simulation and its components is depicted in figure 3.

The gateway core is developed using the AUTOSAR architecture tool SystemDesk and is integrated into the AUTOSAR simulator.

4.1 Overview of the Simulation Environment

The simulation environment of the TIMEA platform is based on the automotive simulation tool VEOS and a simulation model [14] of the message-based NoC. The VEOS simulator is used for the simulation of the virtual AUTOSAR ECUs and their physical environment while the NoC simulation is running in an independent SystemC application. The communication between the simulation tools is performed through TCP/IP. Additionally, local coordinators are responsible for the data exchange and the synchronization of the simulation steps between both simulation systems.

The NoC local coordinator controls the NoC simulation based on the exchange of synchronization messages with the VEOS local coordinator. Furthermore, it is in charge

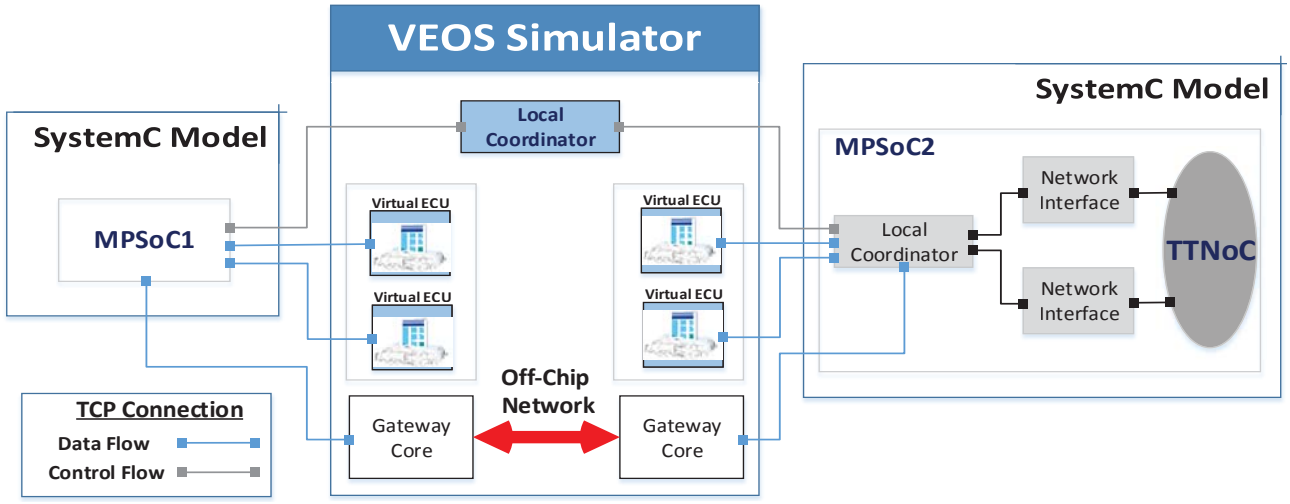

Figure 3 *Co-simulation Environment for TIMEA*

of the exchange of data between the virtual ECUs and their corresponding network interfaces in the NoC model. Likewise, the VEOS local coordinator is in charge of the synchronization of the AUTOSAR simulation with the NoC simulation. Once a simulation step is performed in the VEOS simulation, the local coordinator sends a synchronization message to the SystemC simulation and waits for its response. Thus, the simulation of the virtual ECUs and environment models is synchronized with the simulation of the NoC.

In this work, we extend the local coordinator in VEOS for the synchronization of multiple NoC simulation models. That means, virtual AUTOSAR ECUs can be mapped to different MPSoCs using different instances of the NoC simulation. Thus, after one simulation step, the VEOS local coordinator sends a synchronization message to each NoC simulation and waits for the responses of all of them before performing the next simulation step. In figure 3 an example of a co-simulation of VEOS with two NoC systemC models is presented.

4.2 Implementation of the Gateway Core

SystemDesk allows the definition of ECU instances where each of them can have a communication controller and a connector to access a physical channel of a communication network of an automotive distributed system. Simulated virtual ECUs are developed as ECU instances configured based on the AUTOSAR architecture with extended COM modules for supporting message-based NoC communication.

For the simulation of the off-chip network gateway core, depending on the specific automotive distributed system, a number of ECU instances is defined in SystemDesk and configured as depicted in figure 4 without application layer, RTE implementation or AUTOSAR OS. Each gateway core is a system core of an independent message-based AUTOSAR MPSoC.

The configuration of each off-chip gateway core is performed as follows.

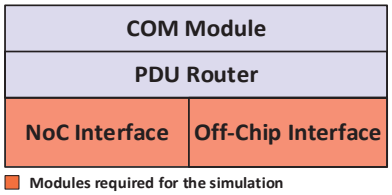

Figure 4 *Configuration of the simulated Gateway Core*

- *Network Description.* A network communication description file is imported into the SystemDesk development environment. This description file contains elements such as system signals, I-Signals/I-PDUs and frames which will be used by the gateway core for the exchange of messages with other MPSoCs. Typically, the network description is based on one of the AUTOSAR description files (e.g. DBC, FIBEX) depending of the automotive off-chip network implemented for off-chip communication.

- *ECU Configuration.* A single empty ECU configuration is selected for each gateway core ECU instance. An off-chip interface module, a COM module and a PDU router are added to the ECU configuration. The off-chip interface module serves as the communication hardware abstraction to connect the gateway core with a simulated off-chip network running in VEOS while the COM module and the PDU router represent the communication service layer.

- *Off-chip Interface.* Depending on the automotive off-chip network the off-chip interface module is required to implement a specific network interface. For CAN and LIN communication protocols SystemDesk provides the specific CAN interface module and LIN interface module respectively. Theses modules allows to connect the gateway core with the simulation of a CAN bus or a LIN bus running in VEOS. For Ethernet or FlexRay, SystemDesk provides the DsIdBusIf module, which offers idealized bus simulation with VEOS (i.e., not all the PDU properties are taken into account in a FlexRay simulation). Independently of the imple-

mented off-chip network, the off-chip module is configured automatically based on the imported network description file.

- *COM module.* The implementation (C code) of the COM module is manually developed. The COM module matches NoC PDUs with off-chip PDUs and vice versa. Using SystemDesk counters are generated to handle message cycles and sending times according to the time-triggered schedule of the message-based NoC and the off-chip network.
- *PDU Router.* The implementation of the PDU router is automatically generated by SystemDesk based on the imported network description file. As mentioned in section 3.2, the PDU router is in charge of routing the PDUs from the message-based NoC to the off-chip network and vice versa. For this, new routing tables are added to the generated PDU router for handling incoming and outcoming NoC PDUs.

Before building the ECU configurations and the generation of the off-chip gateway cores a developed NoC interface module is integrated in each ECU configuration. For the simulation of the gateway core the NoC interface uses the TCP/IP protocol to connect the gateway with its corresponding network interface of the NoC simulation. PDUs coming from the PDU router are sent by the NoC interface to the NoC local coordinator. Additionally, the NoC interface accepts PDUs coming from the NoC simulation and makes them available to the PDU router.
After this, the ECU configurations can be built, generating the gateway cores which are integrated into the simulation system of virtual ECUs for being run in VEOS.

4.3 Use Case and Evaluation

An ABS use case consisting of 5 virtual ECUs was developed. The virtual ECUs are distributed on two TIMEA multi-cores, one of them hosting 3 virtual ECUs and the other one hosting 2 virtual ECUs. Both MPSoCs are provided with an implementation of the off-chip gateway core supporting TT-CAN bus communication for the interaction between each other.
The presented ABS system is simulated using the simulation environment described in section 4.1. A Simulink model was integrated into the AUTOSAR simulation which represents the physical environment of the MPSoCs (driver behavior, wheel, street characteristics, etc). We test the developed ABS system in a simulation scenario where the driver performs a hard braking having an initial angular speed of the wheel of $70.4 rad/sec$. The scheduling of the inter-core communication for the two NoCs is summarized in table 1. In both MPSoCs core 0 represents the core acting as gateway. Additionally, table 2 presents the scheduling of the TT-CAN bus communication between the MPSoCs.
To validate the correct operation of the off-chip gateway core we compare the angular speed of the wheel and the braking distance having enabled and disabled the gateway.

NoC Configuration 1			
Period	**Phase**	**Sender Core**	**Receiver Core**
10us	2us	0	2
10us	4us	1	2
10us	5us	2	0
NoC Configuration 2			
Period	**Phase**	**Sender Core**	**Receiver Core**
10us	2us	1	0
10us	4us	0	2
10us	5us	2	3

Table 1 *NoC configurations*

TT-CAN Configuration			
Period	**Phase**	**Sender MPSoC**	**Receiver MPSoC**
10ms	1ms	1	2
10ms	2ms	2	1

Table 2 *Off-Chip Communication configuration*

In the second scenario the ABS functionality should not work since there is no exchange of messages between the MPSoCs. Figure 5 compares the wheel speed while figure 6 shows a braking distance reduction of $24.6m$ when the the gateway is activated.
In order to quantify the influence of the gateway core on the AUTOSAR multi-core system figure 7 compares the overall of task invocations performed by the OS of each virtual ECU in the two MPSoCs. For this, the use case was also executed without an implementation of the gateway core, which means, keeping the off-chip network functionality in the own BSW of the virtual ECUs.
Figure 7 reflects a reduction of the OS overhead by 28.56%, 66.64% and 28.56% in the virtual ECUs of MPSoC 1 when the gateway is used, while virtual ECUs in the MPSoC 2 present a reduction by 18.17% and 49.96% respectively. These results demonstrate that the developed off-chip gateway core improves the efficiency of the MPSoC platform acting as a hardware accelerator for the AUTOSAR application running in the virtual ECUs.

5 Conclusion

The introduction of an off-chip network gateway core in a message-based multi-core platform for AUTOSAR sim-

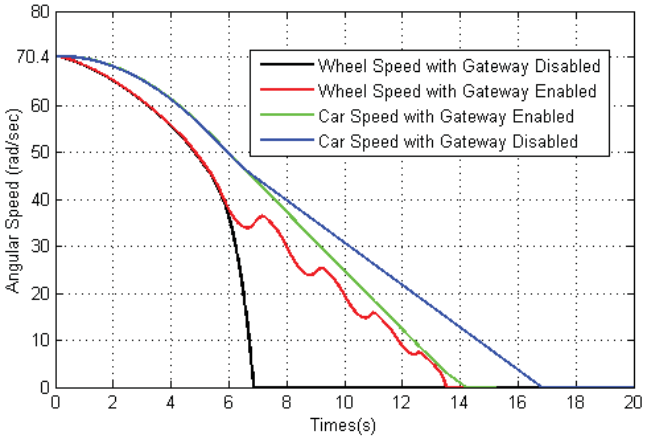

Figure 5 *Wheel and Car Angular Speed*

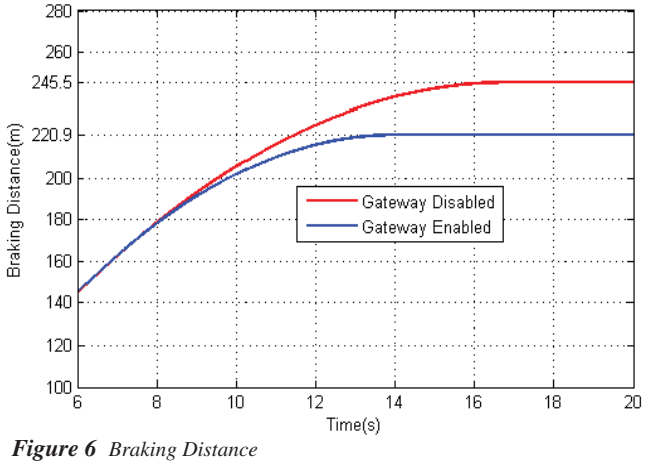

Figure 6 *Braking Distance*

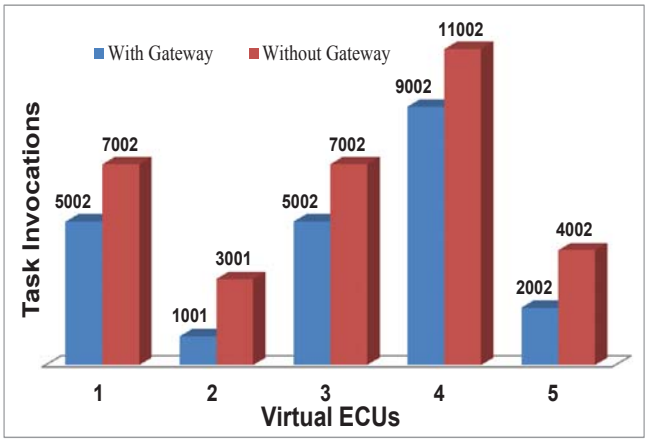

Figure 7 *Overhead Comparison*

plifies the AUTOSAR BSW in the virtual ECUs since the off-chip communication functionality is delegated to a dedicated core. This gateway core allows the communication between SWCs in different MPSoCs which is performed through the off-chip automotive network implemented by the gateway.

In this paper we presented the architecture of the gateway core as well as its integration into the TIMEA platform. Due to the use of the message-based NoC for inter-core communication, SWCs in different virtual ECUs are able to use the gateway for exchange of data with other MPSoCs. Additionally, we introduced a simulation model of the gateway and its development process using the dSpace AUTOSAR solution tools.

A co-simulation environment for message-based AUTOSAR MPSoCs was used for the implementation and evaluation of the gateway. The simulation scenario served to compare the overhead of the OS in the virtual ECUs with and without a gateway implementation. The obtained results reflect a simplified implementation of the BSW in the virtual ECUs as well as a significant increase in the performance of the AUTOSAR OS since it is not any more required to handle expensive off-chip communication functionalities, which are replaced by the integrated off-chip gateway core.

6 Literature

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

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

[3] C. Salloum, M. Elshuber *et al.*, "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.

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

[5] A. Hansson, K. Goossens *et al.*, "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

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

[7] R. Obermaisser, H. Kopetz *et al.*, "A Cross-Domain Multi-Processor System-on-a-Chip for Embedded Real-Time Systems," *IEEE Transactions on Industrial Informatics*, 2010.

[8] F. Poletti, A. Poggiali *et al.*, "Energy-Efficient Multiprocessor Systems-on-Chip for Embedded Computing: Exploring Programming Models and Their Architectural Support," *Computers, IEEE Transactions on*, 2007.

[9] M. Rudorfer, T. Ochs *et al.*, "Realtime system design utilizing AUTOSAR methodology," in *elektroniknet*, 2009.

[10] M. Urbina and R. Obermaisser, "Multi-Core Architecture for AUTOSAR based on Virtual Electronic Control Units," in *Emerging Technologies and Factory Automation 2015. IEEE Conference on*.

[11] *AUTOSAR Specification of communication, AUTOSAR Release 4.1*, AUTOSAR, 2014.

[12] M. Urbina, Z. Owda *et al.*, "Simulation Environment based on SystemC and VEOS for Multi-Core Processors with Virtual AUTOSAR ECUs," in *Dependable, Autonomic and Secure Computing 2015. IEEE Conference on*.

[13] *AUTOSAR Specification of PDU Router, AUTOSAR Release 4.1*, AUTOSAR, 2014.

[14] Z. Owda and R. Obermaisser, "Trace-based simulation framework combining message-based and shared-memory interactions in a time-triggered platform," in *IEEE First International Conference on Event-Based Control, Communication, and Signal Processing*, ser. EBCCSP '15. Krakow: IEEE, June 2015.