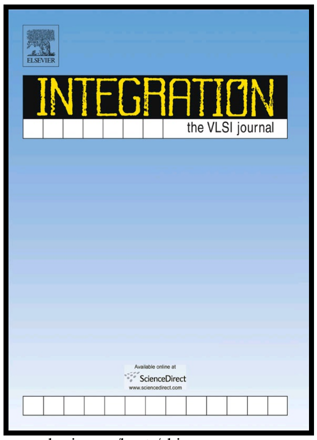
Author's Accepted Manuscript



www.elsevier.com/locate/vlsi

Ensuring Safety and Efficiency in Networks-On-Chip

Adam Kostrzewa, Selma Saidi, Leonardo Ecco, Rolf Ernst

PII: S0167-9260(16)30091-8
DOI: http://dx.doi.org/10.1016/j.vlsi.2016.10.015
Reference: VLSI1261

To appear in: *Integration, the VLSI Journal*

Received date: 29 April 2016
Revised date: 12 September 2016
Accepted date: 13 October 2016

Cite this article as: Adam Kostrzewa, Selma Saidi, Leonardo Ecco and Rol Ernst, Ensuring Safety and Efficiency in Networks-On-Chip, *Integration, th VLSI Journal,* http://dx.doi.org/10.1016/j.vlsi.2016.10.015

Ensuring Safety and Efficiency in Networks-On-Chip

Adam Kostrzewa[+], Selma Saidi[*], Leonardo Ecco[+], and Rolf Ernst[+]

[+]Technische Universität Braunschweig, Germany
[*]Technische Universität Hamburg, Germany

Abstract— Networks-on-Chip (NoCs) for real-time systems require solutions for safe and predictable sharing of network resources between transmissions with different quality-of service requirements. In this work, we present a mechanism for a global and dynamic admission control in NoCs dedicated to real-time systems. It introduces an overlay network to synchronize transmissions using arbitration units called Resource Managers (RMs), which allows a global and work-conserving scheduling. We present a formal worst-case timing analysis for the proposed mechanism and demonstrate that this solution not only exposes higher performance in simulation but, even more importantly, consistently reaches smaller formally guaranteed worst-case latencies than TDM for realistic levels of system's utilization. Our mechanism does not require the modification of routers and therefore can be used together with any architecture utilizing non-blocking routers.

Keywords: Networks-on-chip, Real-time systems, Safety-critical multicores, Dynamic admission control, Shared resources.

I. Introduction

Ensuring safety in Multiprocessor Systems-on-Chip (MPSoCs) typically implies worst-case dimensioning to provide guarantees for the response time of running applications. However, providing worst-case service guarantees in such setups is difficult. Indeed, even with a static task-to-processor mapping, the execution of applications is usually not independent due to accesses to the shared resources where tasks running on different processor interfere. Moreover in modern MPSoCs, to conduct a single memory access, a task must acquire several resources, e.g. interconnect and memory controller, with independent arbiters and often provided by different vendors. The designer must therefore assure that the effects resulting from coupling these different arbiters will not lead to pessimistic formal guarantees or decreased utilization. Consequently, such systems require mechanisms which allow a seamless integration of critical workloads by assuring a composable, efficient and safe coordination of accesses to the shared resources.

Networks-on-Chip (NoCs) are frequently considered in multi- and manycore architectures as the interconnect solution for future real-time systems due to their modular design allowing superior efficiency and scalability. However, in safety critical domains, such as automotive and avionics, the predominant requirement is to provide temporal guarantees i.e. prove that the worst-case system's behavior is predictable and adheres to the application's timing constraints resulting from real-time requirements e.g. worst-case network latency. This requires spatial and temporal separation of concurrent transmissions e.g. "sufficient independence" requested by the avionics safety standard DO-178B or following the same principle of "freedom from interference" requested by the automotive standard ISO26262.

Commonly used wormhole-switched NoCs with multi-stage arbitration are usually not designed to meet these requirements but rather to deliver high average performance e.g. [1], [2]. Therefore, in order to progress, ongoing transmissions must acquire buffers and links separately in each router along their path i.e. packets are switched as soon as they arrive, and all traffic receive equal treatment. Moreover, some interference cannot be resolved locally by the router's arbiter and requires inputs from adjacent neighbours e.g. a joint allocation of the crossbar switch and the router output due to a possible lack of buffers at the output (input-buffered router). This results in a complex spectrum of direct and indirect interferences between data streams which may endanger the system safety since the service depends on the runtime behavior of other streams. Consequently, such networks are considered to be hardly analyzable and generally not applicable to safety critical systems. Nevertheless, standard NoC architectures still remain appealing for use since they are affordable, fast and flexible [3].

Safety, in the context of NoCs, was already addressed by many custom mechanisms and architectures providing a safe and predictable sharing of the interconnect resources (e.g. links and buffers) in order to bound or avoid the interference between concurrent transmissions. There exists two established solutions to tackle this problem: non-blocking routers with rate control [4] and Time-Division Multiplexing (TDM) [5].

The first mechanism is based on a *local arbitration* performed *independently* in routers and conducting *dynamic* scheduling between transmissions competing for the same output port. Although this approach is capable of providing worst-case guarantees, in particular when using virtual channels for isolation, it comes at a high hardware cost and does not scale well with the number of isolated streams.

TDM offers a different solution where each transmission receives, in a cyclic order, a dedicated time slot to have an *exclusive* access to the NoC. TDM allows an easy implementation and provides timing guarantees, but also results in average latencies which are very close to the worst case even when the system is not highly loaded [6]. This is mainly due to the traffic from general-purpose applications that hardly ever follows a constant and predictable pattern assumed by TDM schemes. Hence, an efficient execution is only possible for a single selected use-case with a known and static behavior for which the TDM scheme is fully optimized.

The contribution of this work is an alternative mechanism for providing *efficient and safe sharing* of resources in NoCs