

Autonomic MPSoCs for Reliable Systems

Walter Stechele¹⁾, Oliver Bringmann²⁾, Rolf Ernst³⁾, Andreas Herkersdorf¹⁾, Katharina Hojenski⁴⁾, Peter Janacik⁴⁾, Franz Rammig⁴⁾, Jürgen Teich⁵⁾, Norbert Wehn⁶⁾, Johannes Zeppenfeld¹⁾, Daniel Ziener⁵⁾

¹⁾ TU München, ²⁾ FZI Karlsruhe, ³⁾ TU Braunschweig, ⁴⁾ Uni Paderborn, ⁵⁾ Uni Erlangen, ⁶⁾ TU Kaiserslautern

Abstract

Future MPSoCs have to cope with unreliable functionality of their nanometer-scale internal components, mainly due to increasing sensitivity for technology parameter variations and natural radiation. While error correction is widely known in memory design, protection for arithmetic logic units within CPUs is an issue for future research. For on-chip communication resources there are many error protection techniques available, but a trade-off has to be found between techniques on various levels. In this paper we present our concept for Autonomic MPSoCs with capabilities for runtime detection and correction of sporadic errors, and adaptation of performance, power, and dependability on changing environmental conditions. Functional elements of the MPSoC are continuously monitored and controlled by autonomic elements. Re-distribution of tasks is supported by run-time performance analysis and an autonomic operating system. Life time dependability of the MPSoC is introduced in the design optimization process.

1. Introduction

In many complex systems, Multi Processor Systems on Chip (MPSoC) have proven to be efficient platforms to implement application solutions. They are used to control complex systems, like avionic or automotive. Future MPSoCs will face increasing challenges arising from nanoelectronic technology parameter variations, soft errors through environmental radiation, and sporadic timing errors. To cope with such challenges, we propose to design MPSoCs with some degree of autonomic behavior, to give them capabilities to detect unwanted effects and to adjust themselves to maintain functionality and to optimize performance and power consumption. This requires introducing new self-organizing techniques in the hardware and software design process, on system level and component level.

To design reliable MPSoCs, with the capability to autonomously react on internal or external disturbances, requires a new paradigm in the design process. Not only performance, area, and power have to be in the focus of the designer, but also the online monitoring of the system's behavior and its reaction on disturbances. This means, that adequate sensors, evaluators, and actors inside the MPSoC have to register and analyze sporadic disturbances and trigger adequate reactions. In addition to error detection and correction on the hardware level, an autonomic operating system has to support self-organizing task migration between various computing elements.

2. Autonomic MPSoC Concept

Our proposed Autonomic MPSoCs will consist of three logical layers: A functional hardware layer, an autonomic hardware layer, and an autonomic operating system layer (fig. 1). The functional layer contains the usual IP components or Functional Elements (FEs). The autonomic hardware layer consists of Autonomic Elements (AEs) and an interconnect structure among various AEs. FEs are either general purpose CPUs, memories, on-chip busses, special purpose processing units (PUs) or system and network interfaces as in conventional, non-autonomous designs. AEs on the other hand contain any extensions necessary to improve the reliability of the FE and, thus, convert the FE-AE pairs into autonomous units. This FE/AE split represents only a logical structuring concept; in reality both FEs and AEs will be integrated on the same CMOS substrate. The autonomic operating system layer will enable the dynamic and self-optimizing distribution of operating system services among the MPSoC nodes.

Our Autonomic MPSoC architecture presents a smooth shift from present non-autonomous to autonomous designs. In fact, semiconductor companies can continue using existing IP-libraries by extending them with the proper autonomic elements. This will preserve the huge investments used to build those IP-libraries.

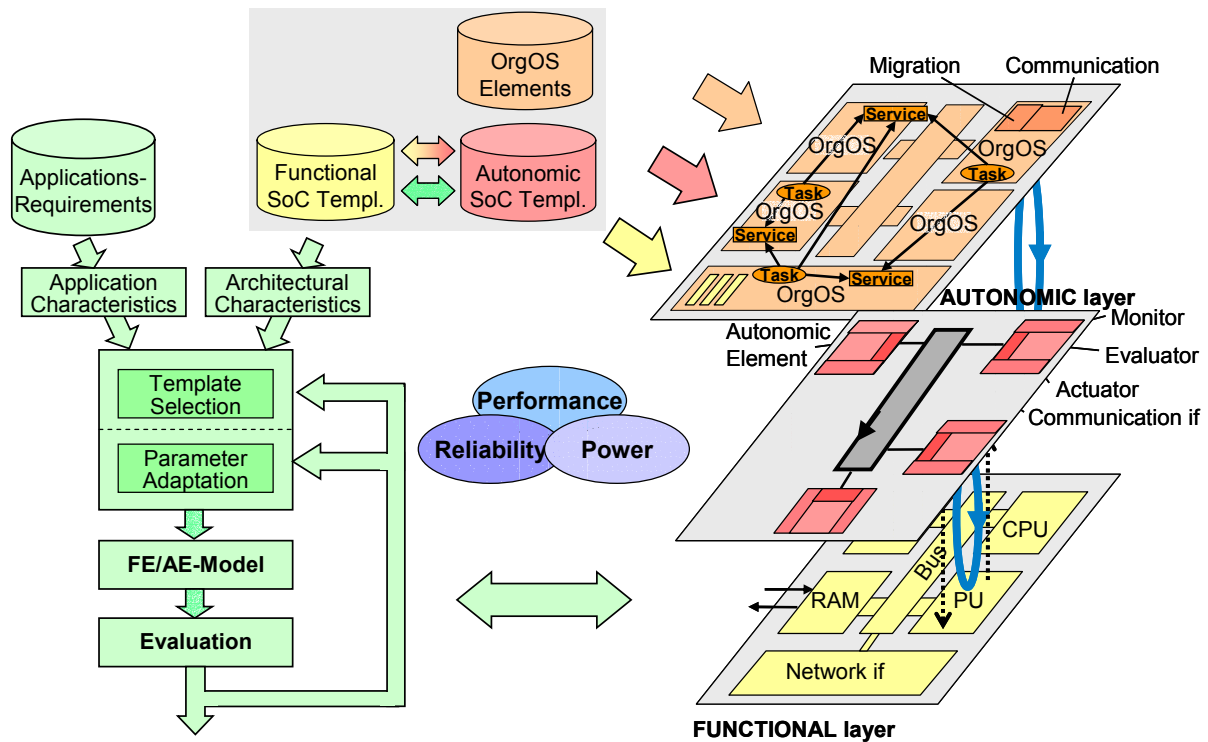


Figure 1: Autonomous MPSoC (right) with functional hardware layer (right bottom), autonomic hardware layer (right middle), autonomic operating system layer (right top), and related design flow (left).

For Autonomous MPSoCs we will investigate many techniques from classical fault tolerance [Avi97] [Avi04] [Avr01], e.g. hardware redundancy, information redundancy, time redundancy, and mixed redundancy techniques. In contrast to classical fault tolerance, we will pursue different approaches as well: (1) We will investigate fault tolerance techniques on chip level with substantially lower overhead than classical hardware redundancy techniques. (2) We will investigate online learning capabilities of the MPSoC to achieve emergent behavior that was not completely determined at design time. (3) We will focus on reuse of existing IP-libraries. (4) We will target not only fault tolerant systems, but also contribute to improve yield for general MPSoCs by giving them the capability to “learn to live with defects”.

3. Conclusions

Future nanoelectronic systems will be increasingly sensible to technology parameter variations, natural radiation, and changing environmental conditions. The classical worst-case design paradigm is not feasible for these challenges. We propose to investigate

Autonomous MPSoCs with the capability to detect and correct sporadic errors and to adapt themselves to changing conditions, in order to achieve reliable system behavior with unreliable components.

Acknowledgment

The authors would like to thank edacentrum in Hannover for coordinating “Autonomic Integrated Circuits” as a cluster research proposal to the German Ministry of Research (BMBF).

References

- [Avi97] A. Avizienis: “Toward Systematic Design of Fault-Tolerant Systems”, IEEE Computer, April 1997
- [Avi04] A. Avizienis et al.: “Basic Concepts and Taxonomy of Dependable and Secure Computing”, IEEE Trans. on Dependable and Secure Computing, 1(1) (2004)
- [Avr01] D. Avreski et al.: “Fault-Tolerant Embedded Systems”, IEEE Micro, Sept.-Oct. 2001