Towards a Framework and a Design Methodology for Autonomic Integrated Systems

* Andreas Herkersdorf, + Wolfgang Rosenstiel

* Technical University of Munich, Institute for Integrated Systems
A.Herkersdorf@ei.tum.de
+ University of Tuebingen, Wilhelm-Schickard Institute for Informatics
Rosenstiel@informatik.uni-tuebingen.de

Abstract: The transition from microelectronics to nanoelectronics reaches physical limits and results in a paradigm shift in the design and fabrication of electronic circuits. The conservative worst-case-approach is no longer feasible and has to be replaced by new design methods. These new design methods and tools have to guarantee reliable and robust systems in spite of unsafe and faulty functions on the lowest process levels.

This paper proposes autonomic or organic computing principles to be applied to hardware design methods for future SoC solutions. Incorporating self-calibration, fault tolerance or even self-healing concepts into IC systems represents a major conceptual shift which requires new design processes and tools. In the future, guarantee of functional correctness at the chip level will include self-configuration of adaptable components and flexible interfaces supporting a flexible component composition within complex SoC systems. A high quality design process leading to more reliable systems is instrumental to secure a leading position in integrated system design among international competition. Of special interest are typical European application areas like automotive electronics, mobile systems, medical technology, smartcards, etc.

1 Integrated Systems (R)Evolution

The ITRS Roadmap [It03] projects micro- and nanoelectronic integrated CMOS circuits to witness a continued capacity growth rate corresponding to doubling transistor counts every two to three years ("Moore's Law"). By 2012, one single chip will host several billion transistors. In comparison, a 1992 Intel Pentium-I processor had a complexity of approximately three million transistors [Mo03]. These capacities enable systems of ever increasing functional complexity and heterogeneity, so called SoC (System on Chip). Systems which few years ago did consist of multiple components on multiple boards, now can be assembled from intellectual property macros and integrated on a single die.

The capacity problem, which dominated semiconductor scaling for decades, transformed into a complexity problem and changed key integrated circuit design challenges. Today and in the future the primary objective will be to develop complex systems with affordable cost and within reasonable time frames. Development and validation time are dominating the system design cycle and cost.

This paper suggests that such a significant shift in IC design challenge demands a new conceptual approach in the IC design method to overcome the productivity bottleneck. We propose autonomic system properties – self-configuring, self-administrating, self-healing and self-protection – to be incorporated into future IC designs and be supported by corresponding tools. Section 2 briefly discusses related work in this field. Section 3 sketches properties and scenarios of autonomic ICs. It also introduces a high level overview on the autonomic integrated systems (AIS) framework we have in mind. Section 4 summarizes important research directions to support AIS.

2 Related Work

In 2001, IBM declared "Autonomic Computing" to be the most important challenge for information technology in the future [Ho01, KC03]. The German Information Technology and Informatics Society (ITG/GI) identified "Organic Computing" as a key technology for computer and system architecture of 2010 [VIG02] and illustrates various application scenarios. In [Mi03] requirements for new design methods and tools are described to guarantee reliable and robust systems in spite of unsafe and faulty functions on the lowest process levels. Self calibrating, fault-tolerant or even self healing systems require totally new design processes and new design tools.

While the primary focus of these activities are systems at chip or box level, the CARUSO project [BBU04] proposed autonomic self-x functions to be provisioned by the middleware layer of a multi-threaded CPU system. Our contribution is complementary to CARUSO because we focus on autonomic principles which are entirely embedded in the hardware layer. As integrated systems form the (hardware) basis of information technology we want to give a perspective of what autonomic or organic computing means at the chip level, and what type of “hardware hooks” higher layer software concepts can base their decision making on.

3 Autonomic IC Design – Conquer Complexity with Complexity

Our approach to master the complexity and reliability demands of future IC systems foresees a partial rededication of chip capacity. Next to the pure application related functions of an ASIC, additional macros shall be dedicated to autonomic surveillance and control functions to ease system diagnosis, debugging and ensure an overall stable and performing system behaviour. Following the analogy to the autonomic nervous system in living organism, we call such ICs “autonomic integrated systems” (AIS).

The autonomic nervous system of the human body controls complex and life critical tasks without our conscious awareness. It ensures that we can dedicate our attention to "functional" activities like skiing, playing tennis, eating, or working. Translated into the world of IC systems, the future may look as follows: With rising workloads, the clock frequency and supply voltage of processor cores are increased to elevate processing performance. Simultaneously, critical transactions on on-chip buses, for example between processors and the memory subsystem, are prioritized and the bus bandwidth of less critical transactions is being reduced. Redundant building blocks, deactivated under regular operating conditions, are activated on demand to increase system performance.

Secondary, low priority functions are deactivated to reduce power consumption. System monitoring and self-test units analyse diagnosis traces of functional blocks or system buses showing suspicious behaviour. Such analysis can lead to the de- or reactivation of entire IC portions.

There do exist already a number of techniques which can be applied and further developed in the context of AIS. For example:

- Dynamic Voltage and Frequency Scaling (DVFS) [CSP04] does control already processor clock and operation voltage.
- Dynamic reconfigurable processing units [SV03] adapt their behaviours during operation to external conditions.
- Built-in-self-test concepts [BCN01] validate circuitry at system start up and partially also during system operation.
- On-chip debugging aids like ChipScope from Xilinx, RISCWatch from IBM, and bus monitors [SLA04] grant partial access to signals and system state information which aren't otherwise observable through regular chip I/Os.
- In [MHS04] a new self-repairing architecture based on dual FPGAs with embedded soft microcontrollers is utilized in design self-healing systems.
- There are mechanisms and strategies to support fault tolerant behaviours of complex systems [Mi03].

However, these concepts are either not yet widely used or, if applied, then only in isolation. In order to let the vision of self-x enabled integrated systems come true they at least have to be made syntactically compatible and semantically coupled. Furthermore, there is demand for an integrating design framework and associated method with tools support.

Autonomic integrated systems (AIS) framework

Figure 1 shows the proposed AIS framework. It basically splits the SoC into two logical layers: The functional layer contains the systems macros and processing units (PU) as in a conventional, non-autonomic design. The autonomic layer consists of Autonomic Control Elements (ACE) and an interconnect structure among the ACEs. This interconnect may but need not be identical to the on-chip interconnect at the functional layer. Likewise, there may but need not be an ACE per functional macro. Each ACE contains a monitor section, which senses state information from the associated functional macro, an evaluator, which merges the locally obtained information with state from other ACEs, and an actuator, which executes a possibly necessary action on the local macro.

ACEs and functional macros form closed control loops which can autonomously alter the behaviour or availability of resources on the functional layer. Control over clock and supply voltage of redundant macros can provision additional processing performance or replace on-the-fly a faulty macro. Likewise, temporarily switching off external chip interfaces when not needed, or narrowing down wide on-chip communication busses under low load conditions, contributes to system power savings.

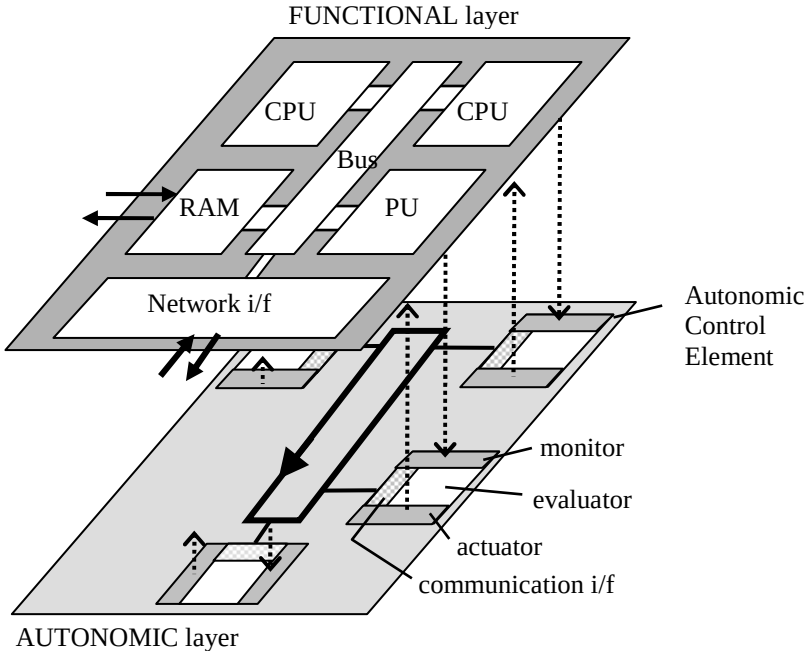

Figure 1: Two-layer autonomic SoC framework

All ACE functions are configurable hardware entities (no software in the loop) to ensure control loop reaction times of few system clock cycles. Autonomic software control loops, e.g. as described in [BBU04, Ho01], can use information gathered by ACEs. ACEs operate autonomously in a decentralized fashion but may be initiated and dynamically configured from a central system control point.

4 Autonomic IC Research Challenges

Academic and industrial research institutions have just started to investigate what autonomic behaviour means at different levels of system abstraction. In the particular context of SoC design, we see a strong demand for research in the following areas:

- New methods and tools which are capable to deal with graceful degradation and redundancy in distributed integrated systems. This includes methods and tools which are able to model and optimize structural changes in the event of failure or suboptimal performance.
- Validation techniques for interdependent functional macros which could only partially be verified. Since functional test coverage of multi-hundred million transistors macros cannot be exhaustive, we need to search for efficient ACE evaluators to inspect the dynamic behaviour of these macros in real-time.
- Methods for dealing with redundant NoC (Network on chip) interconnect structures and redundant functional units within SoCs.
- Concepts for dynamic and coupled power-performance management in SoCs.
- The possibility for flexible and dynamic hardware-software (HW-SW) repartitioning. Defective HW is replaced by an equivalent SW process, or dynamically loaded HW configurations replace a low performing SW process.
- Mechanisms for autonomic detection and adaptation to changing external system environments.

5 Conclusions and Outlook

Secure design of reliable systems is the new challenge to master nanoelectronics when it comes closer to physical limits. We presented the AIS framework and discussed key challenges for future SoC design methods and tools. For the time being, our target is an AIS framework where the spectrum of autonomic behaviour is considered by the designer during system architecture development. In a next step, we can see the autonomic layering concept to be recursively applied on itself. This will result in optimized ACE operations and inter-ACE communication structures developing towards truly emergent system behaviour.

References

[BBU04] Brinkschulte, U ; Becker, J ; Ungerer, T: “CARUSO – an Approach Towards a Network of Low Power Autonomic Systems on Chip for Embedded Real-Time Applications”, WPDRTS04 Workshop, Santa Fee, NM, April 2004.

[BCN01] Benso, A. , et al.: “Online and Offline BIST in IP-Core Design”, IEEE Design & Test of Computers, September-Oktober 2001, pp. 92 – 99.

[CSP04] Choi, K.; Soma R.; Pedram M.: Fine-Grained Dynamic Voltage and Frequency Scaling for Precise Energy and Performance Trade-off based on the Ratio of Off-chip Access to On-chip Computation Times, DATE 04, pp. 4-9, Paris, February 16-20, 2004

[Ho01] Horn, P: “Autonomic Computing: IBM’s Perspective on the State of Information Technology”, IBM Corporation, Oct. 2001, http://www.research.ibm.com/ autonomic/manifesto/

[It03] “International Technology Roadmap on Semiconductors 2003”

[KC03] Kephart, J.; Chess, D.: „The Vision of Autonomic Computing”, IEEE Computer Magazine, January 2003, pp. 41 – 50.

[MHS04] Mitra, S.; Huang, W.-J.; Saxena, N.R.; Yu, S.-Y.; Mc Cluskey, E.J.: “Reconfigurable Architecture for Autonomous Self-Repair”, IEEE Design and Test of Computers, May-June 2004, pp. 228-240

[Mi03] De Micheli, G.: „Designing Robust Systems with Uncertain Information“, Asia and South Pacific Design Automation Conference (ASPDAC 03), January, 2003

[Mo03] Moore, G.: „No Exponential is Forever … but We Can Delay Forever“, International Solid State Circuits Conference (ISSCC), February 10, 2003.

[SV03] Schaumont, P.; Verbauwhede, I.; Keutzer, K.; Sarrafzadeh, M.: „A quick safari through the reconfiguration jungle“, Design Automation Conference (DAC-01), pp. 172–177, New York, June 18–22 2001. ACM Press.

[SLA04] Stollen, N. , Leatherman, A, Ableidinger, B , Edgar, E: “Multi-Core Embedded Debug for Structured ASIC Systems”, DesignCon2004, February 2-5, 2004, Santa Clara, USA.

[VIG02] VDE/ITG/GI – Positionspapier: „Organic Computing: Computer- und Systemarchitektur im Jahr 2010“, Arbeitsgruppe Organic Computing