Error Detection Techniques Applicable in an Architecture Framework and Design Methodology for Autonomic SoCs*

Abdelmajid Bouajila[1], Andreas Bernauer[2], Andreas Herkersdorf[1], Wolfgang Rosenstiel[2,3], Oliver Bringmann[3], and Walter Stechele[1]

[1] Technical University of Munich, Institute for Integrated Systems, Germany
[2] University of Tuebingen, Department of Computer Engineering, Germany
[3] FZI, Microelectronic System Design, Karlsruhe, Germany

Abstract. This work-in-progress paper surveys error detection techniques for transient, timing, permanent and logical errors in system-on-chip (SoC) design and discusses their applicability in the design of monitors for our Autonomic SoC architecture framework. These monitors will be needed to deliver necessary signals to achieve fault-tolerance, self-healing and self-calibration in our Autonomic SoC architecture. The framework combines the monitors with a well-tailored design methodology that explores how the Autonomic SoC (ASoC) can cope with malfunctioning subcomponents.

1 Introduction

CMOS technology evolution leads to ever complex integrated circuits with nanometer scale transistor devices and ever lower supply voltages. These devices operate on ever smaller charges. Therefore, future integrated circuits will become more sensitive to statistical manufacturing/environmental variations and external radiation causing so-called soft-errors. Overall, these trends result in a severe reliability challenge for future ICs that must be tackled in addition to the already well-known complexity challenges. The conservative worst case design and test approach will no longer be feasible and should be replaced by new design methods. Avizienis [1] suggested integrating biology-inspired concepts into the IC design process as a promising alternative to today's design flow with the objective to obtain higher reliability while still meeting area/performance/power requirements. Section 2 of the paper presents an Autonomic SoC (ASoC) architecture framework and design method which addresses and optimizes all of the above mentioned requirements. Section 3 surveys existing error detection techniques that may be used in our Autonomic SoC. Section 4 discusses implications on the ASoC design method and tools before section 5 closes with some conclusions.
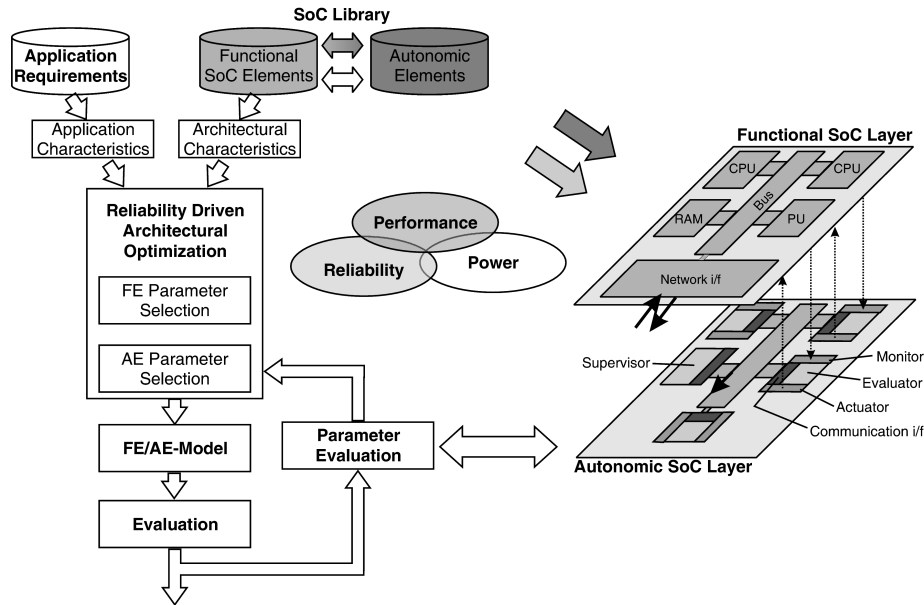

Fig. 1. Autonomic SoC design method and architecture [2]

2 Autonomic SoC Architecture and design method

Figure 1 [2] shows the proposed ASoC architecture platform. The ASoC is split into two logical layers: The functional layer contains the intellectual property (IP) components or Functional Elements (FEs), e.g. general purpose CPUs and memories, as in a conventional, non-autonomic design. The autonomic layer consists of interconnected Autonomic Elements (AEs), which in analogy to the IP library of the functional layer shall eventually represent an autonomic IP library (AE_lib). At this point in time, it is not known yet whether there will be an AE for each FE, or whether there will be one AE supporting a class of FEs.

Each AE contains a monitor or observer section, an evaluator and an actuator. The monitor senses signals or states from the associated FE. The evaluator merges and processes the locally obtained information originating from other AEs and/or memorized knowledge. The actuator executes a possibly necessary action on the local FE. The combined evaluator and actuator can also be considered as a controller. Hence, our two-layer Autonomic SoC architecture platform can be viewed as a distributed (decentralized) observer-controller architecture. AEs and FEs form closed control loops which can autonomously alter the behavior 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 with a "cool" stand-by alternative.

Although organic enabling of next generation standard IC and ASIC devices represents a major conceptual shift in IC design, the proposed ASoC platform represents