

Chapter 12

Fault Simulation and Fault Injection Technology Based on SystemC

Silvio Misera

Kjellberg Finsterwalde, Germany

Roberto Urban

Brandenburg University of Technology Cottbus, Germany

ABSTRACT

Electronic systems are essential parts in the everyday life, and these systems have grown continuously in their performance and in their complexity. Therefore, the design process for such a system has also become more and more complex and has become a critical challenge. A popular tool for solving the design process of complex systems is SystemC. SystemC is a C++ language class library, which allows the modeling of hardware behavior in a programming language for software. Advantages by the usage of SystemC are e.g. an integrative concept of the design description in a continuous design flow, combined with a fast execution of compiled code during the simulation process. This chapter describes the simulation of faults in electronic systems by the usage of SystemC. It treats especially faults in hardware, which can occur after a successful validation of the design. The simulation of faults is well known in two operation areas. One field is the fault simulation of test pattern for the detection of fabrication faults. The other field is the simulated fault injection that the effects of faults emerging during the operation of the device. Such procedures are important for analysis of electronic system designs for safety critical applications with respect to their dependability under fault conditions. At first, the chapter explains some issues in the context of fault injection and fault simulation. Furthermore, it relates something about SystemC and the possibility to simulate the design. State of the art applications are also presented for this purpose. A prerequisite for a simulation is an appropriate modeling. The modeling of faults is presented in the

DOI: 10.4018/978-1-60960-212-3.ch012

context of their design level. It is also shown how a simulation with fault models can be implemented by several injection techniques. In addition, the advantages and disadvantages of the injection techniques are explained. Another challenge for a simulation task is the execution in an acceptable amount of time. Approaches are presented, which help to speed up simulations. Some practical simulation environments are shown at the end of the chapter.

INTRODUCTION

Beyond applications such as video game consoles or mobile phones, electronic computer-based systems are heavily used in time- and safety critical applications like in automotive applications, financial services or nuclear power stations. Unfortunately, neither hardware nor software can be assumed as “fault free”. Therefore, it is an essential problem of today’s system design technology to cope with faults of different nature. Structures in deep sub micron technologies lead to smaller components and facilitate the creations of complex systems. Therefore, we can find complete systems on a silicon chip as a System on a Chip (SoC) or in a package as System in a Package (SiP) with a heterogeneous structure. But their ever-shrinking feature size and the associated growth in complexity have drawbacks. The probability of the occurrence of faults increases with the amount of elements. In addition, smaller structures and lower voltage levels boost the arising of transient faults. These transition faults, e.g. induced by particle radiation, can be the main source of errors, accounting for more than 90% in modern circuits. Such problems have to be solved by additional measures in order to guarantee a correct operation even under fault conditions. Regrettably, these measures increase the system complexity, too. The behavior of a system has to be validated in the design process to comply with the confidence. For this purpose, the simulation of faults and fault effects is an essential tool. The simulation of faults has two important areas of application. On the one hand, the simulation of faults is well known for validation of test patterns, which are used for detection of faults introduced by the integrated

circuit manufacturing process. This simulation of typical permanent faults for manufacture tests is often denoted as “fault simulation”, and it has been well known for decades. On the other hand, after the production and implementation of a system from known-good devices, most of faults occurring in the field of application have a transient nature, they arise during operation. The simulation of the design with faults that arise during operation is denoted as simulated fault injection (SFI) or simulation based fault injection. Such simulators have a different focus compared with the traditional fault simulation, because also the diagnosis of a fault is important. But the validation of a correct operation is not only important. Additionally, it can be interesting to examine the severity of the error, and whether a fault has a fatal consequence.

Today, SystemC has been established as a de-facto standard for the design of complex electronic systems since its arose in the last decade. SystemC allows the modeling at several levels of abstraction. Furthermore, it allows the composition of different parts like software, processors, busses, logic, etc. On the one hand, SystemC can not only be used for the modeling. It is also possible to simulate the design, whereby the simulation of faults is important. Therefore, it is useful to think about a simulation of faults in SystemC. Another benefit is the simulation - also of faults - at different levels of abstraction. With recent extensions of SystemC models also in the domain of analog and mixed-system designs (SystemC-AMS), the possibilities of simulation and validation become even more comprehensive. This chapter gives an introduction of the simulated fault injection by the usage of SystemC. The chapter is further structured as follows. The next sections introduce

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/fault-simulation-fault-injection-technology/51405

Related Content

Impact Assessment of Policies and Practices for Agile Software Process Improvement: An Approach Using Dynamic Simulation Systems and Six Sigma

George Leal Jamiland Rodrigo Almeida de Oliveira (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming* (pp. 1616-1641).

www.irma-international.org/chapter/impact-assessment-of-policies-and-practices-for-agile-software-process-improvement/261093

The Role of Compliance and Conformance in Software Engineering

José C. Delgado (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 103-133).

www.irma-international.org/chapter/the-role-of-compliance-and-conformance-in-software-engineering/192874

Rural Scenery Narrative and Field Experiences: From an Aspect of Kansei

Tadashi Hasebe, Michiaki Ohmuraand Hisashi Bannai (2011). *Kansei Engineering and Soft Computing: Theory and Practice* (pp. 255-265).

www.irma-international.org/chapter/rural-scenery-narrative-field-experiences/46402

Understanding the Role of Knowledge Management in Software Development: A Case Study in Very Small Companies

Rory V. O'Connorand Shuib Basri (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 485-500).

www.irma-international.org/chapter/understanding-the-role-of-knowledge-management-in-software-development/192889

Various Extensions for the Ambient OSGi Framework

Stéphane Frénot, Frédéric Le Mouël, Julien Pongeanand Guillaume Salagnac (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 1799-1810).

www.irma-international.org/chapter/various-extensions-ambient-osgi-framework/62545