Self-Test Mechanisms for Automotive Multi-Processor System-on-Chips

Doctoral Dissertation
Doctoral Program in Control and Computer Engineering (33rd Cycle)

Self-Test Mechanisms for Automotive Multi-Processor System-on-Chips

By

Andrea Floridia

******

Supervisor
Prof. Ernesto Sanchez, Supervisor

Politecnico di Torino
2021

While the technology for enabling fully autonomous self-driving cars is still ahead, today automobiles massively rely on electronics for a variety of functionalities. As these functionalities require more and more computational power, the embedded systems introduced in the automobiles had to evolve accordingly. As of today, Multi-processor System-on-Chips (MPSoCs) are commonly found for these applications. Such SoCs embeds two or more processor cores within the same silicon die, in conjunction with different peripherals and levels of memories (both static and non-volatile). It is known that due to the harsh environment in which they are deployed, physical malfunctions due hardware faults can manifest. Periodic in-field self-testing represents a common countermeasure against these threats. These mechanisms can be implemented both in hardware and software but, most of them were originally devised for simpler single-processor SoCs. While in recent years there has been a considerable effort in improving the hardware-based self-test mechanisms, the same is not true for the software-based ones. Software-based approaches mainly consist in the application of software self-test routines (or procedures) of a Software Test Library (STL). Over the years they have been shown to be valuable especially for the processor core, being the most critical portion of the system.

The first contribution of this thesis consist of a study of the applicability of STLs in a multi-processor context. Rather than the development of new Software-Based Self-test (SBST) methods, the main focus was the parallel execution of already developed STLs. This originates from the fact that automotive MPSoCs reuse the same IP processor cores that have been used (and extensively verified) for single-processor devices. In MPSoCs, it has been widely reported that the major hurdle of embedded software is its predictability in terms of execution time. In fact, when all the processor cores are active at the same time, the system bus activity considerably increases with respect to a single-core system. This higher activity (which induces the processor pipeline to stall) impacts the performances of each processor, since the accesses to the memory sub-system are delayed. Since the STL is an embedded software, it is exposed to the same issues. The most significant achievement under this perspective consisted in the development of a software scheduler. Such a scheduler differs from any other software scheduler for embedded systems since it is tailored for the needs of STLs in an automotive MPSoC. Through extensive experiments, it was proven to be a valid solution to fit the narrow test windows of industrial automotive MPSoCs (maximizing the overall system availability). At the same time, the proposed scheduler demonstrated a good execution time predictability. This is remarkable, since it was already mentioned that the embedded software execution time is hard to be predicted accurately. Furthermore, the research developed in this thesis demonstrates for the first time through real industrial case study that this unpredictability is harmful for self-test procedures. Indeed, it was observed that such unpredictability not only alters self-test procedure execution time (when executed in parallel). Additionally, some self-test procedures intermittently fail when in field and/or produce a fluctuating fault coverage. For both cases, it was proposed a mitigation technique based on the usage of the inner most level of private cache memories. When the self-test procedures are executed from such private memories, with precautions, it is

possible to isolate the self-test procedure execution form the rest of the system (thus achieving the required stability).

When dealing with self-test mechanisms, it exists a further category of mechanisms which are said to be hybrid. They indeed consist of both hardware and software cooperating for implementing an efficient self-test. This thesis contributes to this new emerging self-test approach with a novel hybrid technique for checking the integrity of the comparators used in the lockstep configuration (commonly found in automotive MPSoCs).

The third contribution of this thesis consists of optimizing the fault grading methodologies to meet the necessities of the functional fault simulation. Functional fault simulation is an emerging approach for performing fault simulation in safety-critical applications when a processor executing software is involved. Recently, the Electronic Design Automation (EDA) companies are providing tools able to support these methodologies. However, different aspects must be considered, not previously considered with the traditional fault simulation approaches. It is worth noting that the applicability of these researches is not limited to processors of an automotive MPSoCs. One of the main applications of functional fault grading addressed in this thesis is in the STL development. The main bottleneck of the STL development remains the fault simulation. Therefore, part of the research efforts was directed towards the formulation of functional fault grading methodologies intended for STLs. The key concept behind these methodologies is the fault dropping which allows to considerably reduce the effort for the fault simulation.

The fourth contribution still concerns the functional fault grading, but in a rather different scenario. Due to the ever-increasing complexity of the newer devices, designers are shifting to emulation of the ASICs in order to speed up the verification process. The same emulators can be used as well for quickly evaluating the effectiveness of self-test mechanisms or general dependability analyses that require functional fault grading. This can be achieved by instrumenting the original netlist, in order to enable the injection of different faults (most often either Single-Event Transient or stuck-at). In this context, this thesis introduces a fault emulation platform to support dependability analyses of safety-critical Application-Specific Integrated Circuits (ASICs). Differently than existing works, the focus was the fault detection mechanism that allows to mimic the detection mechanisms of a fault simulator (including fault dropping introduced above). The proposed platform can be integrated in the already-existing IEEE 1149.1 JTAG infrastructure of the target ASIC. Therefore, it can be easily accessed with standard tools and perfectly compatible with the modern industrial emulators based on Field-Programmable Gate Arrays (FPGAs).