Formal Methods for Test and Reliability - Abstract

In the rapidly evolving landscape of nanotechnology, where innovations promise groundbreaking advancements in various industries, ensuring the reliability and safety of digital circuits becomes paramount. While applicable to the broad category of commercial off-the-shelf products, this becomes significantly more evident when examining domains or industry sectors, such as automotive, aviation, railways, and the biomedical sector, that fall into the category of safety-critical applications. In these cases, the probability that a fault may activate an error and propagate to a failure that would endanger human lives or cause environmental damage should be carefully evaluated and kept under predefined thresholds. To achieve this result, the manufacturers must comply with strict safety standards and procedures that mandate rigorous coverage thresholds and comprehensive testing protocols. From end-of-manufacturing up until the in-field phase, each integrated circuit (IC) is subjected to several testing procedures to ensure that it meets stringent quality standards, functions reliably within specified parameters, and remains resilient to various environmental conditions throughout its operational lifespan. Design-for-testability (DfT) techniques are incorporated during the design phases of electronic circuits to enhance the testing process.

However, despite the presence of powerful electronic design automation (EDA) utilities, such as automatic test pattern generation (ATPG) tools, intended for use alongside DfT-compliant designs during testing, the relentless evolution of technology brings about faster, smaller, and denser circuits. This evolution renders certain utilities inadequate as the complexity of the test procedure significantly increases, as seen with Burn-In (BI) test. Burn-In, an omnipresent step in the test chain for products intended for use in safety-critical domains, was, until recently, conducted in its traditional static format. Notwithstanding its effectiveness, static BI test became less effective for newer, denser technologies as in its static form it was found not to fully exercise all internal parts of the ICs. Hence, it evolved into new dynamic forms, where stress stimuli are applied in an internal manner on top of the external temperature and voltage increase. However, the generation of appropriate stress-inducing stimuli is a costly and arduous task for the test engineers, due to the lack of automation to aid the generation process.

Another test domain that could substantially benefit from automation is the in-field test. Continuous in-field testing enables the detection of faults or anomalies that may occur over time. Early detection of potential issues allows for proactive maintenance or corrective measures, reducing the risk of system failures in critical situations. However, the task of developing appropriate software test libraries (STLs) for such scenarios is typically a task that requires a lot of manual effort from the perspective of the test engineer. In fact, not only the test must consider parameters such as application time and memory footprint but it must also avoid targeting untestable faults of the design. This means that the test should only focus on those faults that are able to produce a failure in the operating scenario, ignoring those that can not produce any (critical) failure.

This PhD thesis proposes solutions, based on Formal Methods (FMs), addressing the aforementioned test topics. The manuscript is organized in three main parts.

The initial part provides an introduction and overview of the imperative need for testing and reliability in the modern digital era. It delves into the distinct testing areas that form the focus of this thesis.

The second part includes the three main contributions of the thesis. A section proposing FM-based solutions for dynamic BI test stress stimuli generation is first presented. These methods consider various switching activity metrics, and their effectiveness is showcased by applying them on scalar pipelined processors. The second contribution regards FM-based solutions targeting the identification of functionally untestable faults under the stuck-at and the cell-aware fault models. Lastly, the final contribution regards methodologies that aid the generation of STLs for microprocessors and GPUs.

The last part provides the conclusions of the overall work.