Maximizing Power Consumption by Exploiting Genetic Algorithms for Automatic System-Level Test Program Generation

Denis Schwachhofer[1,4], Francesco Angione[2], Steffen Becker[1], Stefan Wagner[1],
Matthias Sauer[3], Paolo Bernardi[2], Ilia Polian[4]

[1]Institute of Software Engineering, University of Stuttgart, Stuttgart, Germany
[2]Dip. di Automatica e Informatica, Politecnico di Torino, Turin, Italy
[3]Advantest Europe, Boeblingen, Germany
[4]Institute of Computer Engineering and Computer Architecture, University of Stuttgart, Stuttgart, Germany

***Abstract*—In the last decade, the manufacturing test flow for safety-critical scenarios has been enhanced with an additional test step: *System-Level Test (SLT)*. In safety-critical scenarios, SLT has emerged as an essential test phase for increasing quality requirements for electrical/electronic (E/E) systems due to technological limits, further verifying the design, and quickly adding fault coverage for escapes.**

SLT mainly consists of running functional applications that mimic the in-field behavior, workload, and environment. A weak point of SLT, or in general, of functional test programs, is how to automatically generate such applications without investing too much effort and skills from test engineers. Its grading is prohibitive in terms of computational time. Therefore, researchers are looking into different directions, for example, instruction-based measurements for verifying the data flow of an SLT application or indirect measurements on the power consumption.

This paper presents a method to generate SLT programs based on Genetic Algorithms. For this purpose, we use MicroGP, a Genetic Programming framework, and let it generate RISC-V assembly snippets. The feedback for MicroGP consists of current measurements of an out-of-order processor running on an FPGA and the instruction per cycle generated from an Architectural simulation of the given out-of-order processor. We will also investigate if instruction per cycle correlates with power consumption and vice versa.

***Index Terms*—System-Level Test, test generation, genetic algorithm, RISC-V**

I. Introduction

In the last decade, the manufacturing test flow for safety-critical scenarios has been enhanced with an additional test step, the so-called *System-Level Test (SLT)*. The concept of SLT is not new: it comes from non-safety critical manufacturing test flow where it is sufficient to verify only the black-box functionalities of manufactured System-on-Chips (SoCs). However, in safety-critical scenarios, SLT has emerged as an essential test phase [1] for increasing quality requirements for electrical/electronic (E/E) systems due to technological limits, further verifying the design, and quickly adding fault coverage for escapes. SLT mainly consists of running functional applications that mimic the in-field behavior, workload, and environment.

A weak point of SLT, or in general, of functional test programs, is how to automatically generate such applications without investing too much effort and skills from test engineers. Moreover, SLT fault simulation may become prohibitive regarding computing time due to the rising complexity of modern SoCs and SLT application [2]. A possible solution could be partitioning the SoC following the module hierarchies and fault-simulating an SLT application on a small subset of gates. However, it does not scale when moving to a system level. Therefore, researchers are looking into different directions, for example, instruction-based measurements for verifying the data flow of an SLT application [3] or indirect measurements on the power consumption [4].

This paper presents a method to generate SLT programs based on Genetic Algorithms. For this purpose, we use MicroGP [5], a Genetic Programming framework, and let it generate RISC-V assembly snippets. The feedback for MicroGP consists of current measurements of an out-of-order processor running on an FPGA and the instruction per cycle generated from an Architectural simulation of the given model of the out-of-order processor. We will also investigate if instruction per cycle correlates with power consumption and vice versa.

II. Background

As the transistors scale, the density of transistors per area increases. This important achievement led to an exponential rise in the complexity of integrated circuits. As the complexity increases, the testing scenario becomes more complex [6].

The test flow is divided into stages, each targeting various defects and with a specific goal to discard manufactured faulty devices [2], [7]. Fig. 1 shows the test flow. Missing are Burn-In and package test.

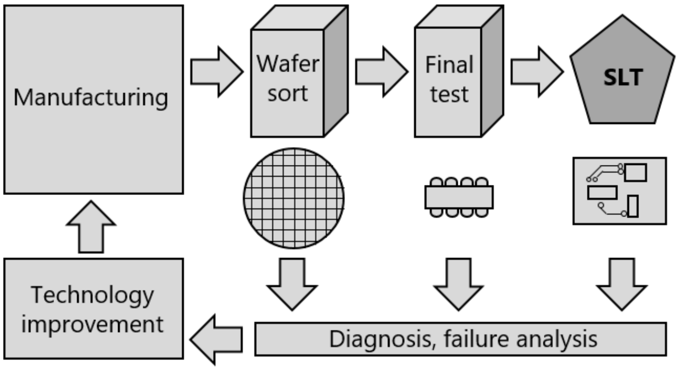

Figure 1: The test flow with SLT.

Structural tests cover a vast but incomplete spectrum of all possible defects without considering test escapes from previous test steps. Therefore *System-Level Test* has been introduced to increase the growing quality requirements dictated by safety standards and quickly add coverage to escapes.

III. Proposed approach

An essential aspect in developing SLT programs, or general test programs, is the challenge of automating this process due to the different nature of CPUs and submodules used. Moreover, SLT lacks a commonly defined metric to assess the quality of such programs. Fault simulations are prohibitive due to the rising complexity of devices and programs. Therefore, methods ranging from instruction-based analysis [3] or indirect measurements [4] are currently being investigated.

More in detail, the scope of this paper is to investigate the effectiveness of SLT programs generated by genetic algorithms with indirect measurements based on the device's power consumption during the execution of such programs.

Fig. 2 shows the workflow of the work. SLT programs are generated by the genetic algorithm used in the μGP toolkit [5]. There are two evaluators for the genetic algorithm, an on-chip evaluator for the power measurements and an off-chip evaluator based on the architectural simulation of the given generated individual (SLT program).

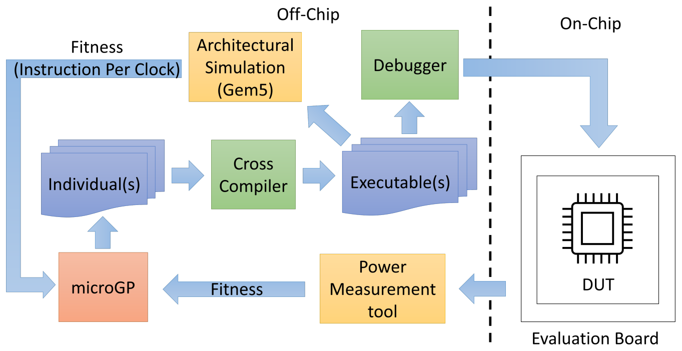

Figure 2: Proposed approach workflow.

The approach is to generate individuals (programs) maximizing *Instruction per Cycle* (IPC), based on the assumption that a higher IPC could lead to a higher power consumption [8]. Afterward, the power consumption is evaluated for each of the best individuals from the architectural simulation, and the generation can continue.

IV. Experimental results

The experimental results are obtained from the Design-Under-Test (DUT), which is an out-of-order, super-scalar CPU called *BOOM* [9] based on the *RISC-V* Instruction set architecture. The DUT is synthesized for an FPGA-based evaluation board from which the power measurements are extracted and used as fitness value for μGP, as shown in Fig. 2. On the other hand, Architectural simulations are executed off-chip to speed up the evaluation process. The experimental results obtained on a population of 100 generated individuals, i.e., SLT test programs, are shown in Fig. 3

As it can be seen, in a given range of *Instruction per Cycle* (IPC), orange dots in Fig. 3, the power consumption falls back in a wide range. The wide range of power consumption is mainly due to the nature of patterns used in the registers and dependencies in the generated SLT programs. Therefore, high power consumption can be achieved even with a lower IPC but with the correct activation patterns.

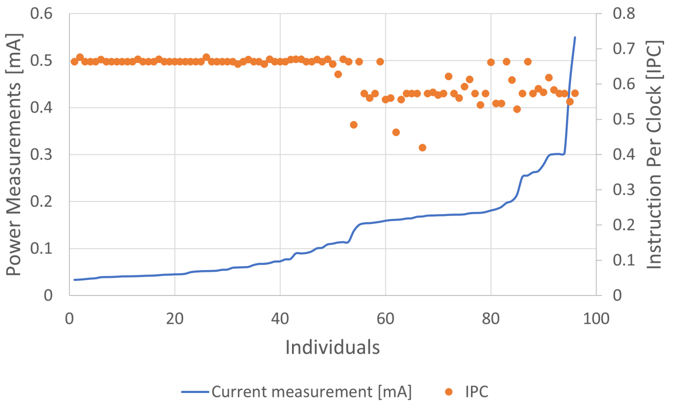

Figure 3: Power measurements and Instruction Per Clock for each individual.

What one can observe as well that there is a slight downwards trend in the IPC the higher the power consumption becomes. We assume this is due to the fact, that the more power-hungry snippets cause backpressure on the different stages of the BOOM core. This in turn causes a higher activity increasing power consumption.

V. Conclusions and Future Works

The use of a genetic algorithm for SLT program generations is promising. Combining an Architectural Simulator and a Power measurement tool allows fine-tailor programs that generate a high power consumption. An essential aspect of the experimental results is that the nature of patterns affects the power consumption more than a slight change in the Instruction per Cycle. Therefore, the correlation between power consumption, Instruction per Cycle, and patterns has to be deeply investigated.

Acknowledgements

This work was supported by Advantest as part of the Graduate School "Intelligent Methods for Test and Reliability" (GS-IMTR) at the University of Stuttgart.

References

[1] D. Appello et al., "System-level test: State of the art and challenges," in *IEEE IOLTS*, 2021.

[2] I. Polian, J. Anders, S. Becker, *et al.*, "Exploring the mysteries of system-level test," in *2020 IEEE 29th Asian Test Symposium (ATS)*, 2020, pp. 1–6.

[3] F. Angione, D. Appello, P. Bernardi, *et al.*, "An innovative strategy to quickly grade functional test program," 2022.

[4] H. Amrouch *et al.*, "Intelligent methods for test and reliability," in *IEEE DATE*, Mar. 2022, pp. 969–974.

[5] E. Sánchez, M. Schillaci, M. S. Reorda, G. Squillero, L. Sterpone, and M. Violante, "New evolutionary techniques for test-program generation for complex microprocessor cores," in *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, ser. GECCO '05, New York, NY, USA: Association for Computing Machinery, Jun. 2005, pp. 2193–2194.

[6] M. Campbell, "Plenary presentations: Keynote: The product complexity and test — how product complexity impacts test industry," in *2010 15th IEEE European Test Symposium*, 2010, pp. 9–9.

[7] C. He and Y. Yu, "Wafer level stress: Enabling zero defect quality for automotive microcontrollers without package burn-in," *2020 IEEE International Test Conference (ITC)*, pp. 1–10, 2020.

[8] N. Chang *et al.*, "Cycle-accurate energy consumption measurement and analysis: Case study of arm7tdmi," in *ISLPED*, Jul. 2000, pp. 185–190.

[9] J. Zhao, B. Korpan, A. Gonzalez, and K. Asanovic, "SonicBOOM: The 3rd Generation Berkeley Out-of-Order Machine," en, *Fourth Workshop on Computer Architecture Research with RISC-V*, p. 7, May 2020.