Computer Science Ph.D Proposal (CIFRE)

Improving Diagnosis for a Formal Verification Tool for Electrical Circuits at Transistor Level

LIP Laboratory (Lyon – France) & VERIMAG Laboratory (Grenoble – France)
Aniah company (Grenoble – France)
Supervisors: Matthieu Moy (Associate Professor, UCBL/LIP)
Bruno Ferres (Associate Professor, UGA/VERIMAG)
Mehdi Khosravian Ghadikolaei (Ph.D, R&D Engineer at Aniah)
Physical location: Lyon and/or Grenoble, France

Keywords: Formal Methods; SMT Solving; Logical Formulas; Integrated Circuits; User Requirements; Debugging; `OCaml`

Context

Aniah is a Start-up that offers tools for analyzing integrated circuits at an industrial scale[1]. Aniah has introduced algorithms that significantly pushes the boundaries of the size of analyzable circuits, from a few hundred thousand elements to several trillion. Aniah is working in collaboration with the Laboratoire de l'Informatique du Parallélisme (LIP) and the Verimag laboratory bringing expertise on state-of-the-art formal methods. We already co-supervised one post-doc and have an ongoing CIFRE[2] Ph.D on the topic. We already have novel results on the use of a generic solver for logical formula (Z3, a Satisfiability Modulo Theory solver) to electric verification. While the use of Z3 for formal verification is well-known, this is to the best of our knowledge the first application to electrical verification at transistor level. Our approach is published [6, 10], implemented in the commercial tool, and a patent is pending on more elaborated techniques.

Model-checking [13] consists in exploring all the reachable states of a system, typically to check the unreachability of a set of error states. It is a well-established set of techniques, and has successfully been applied both to software [1, 8, 7] and hardware [2, 4]. It is usually applied to check properties on the *behavior* of a system. For example, hardware model-checking usually considers boolean values (0 and 1, possibly extended with X and Z to model short-circuits and disconnected signals), but abstracts away the physical details (typically, voltage values are not modeled). Model-checking can be either enumerative (reachable states are explored one by one), or symbolic. Symbolic model-checking consists in representing a possibly very large set of states using a symbolic formula, that can be exponentially more efficient in terms of memory footprint and execution time. Common tools for symbolic model-checking are Binary Decision Diagrams (BDD) [5] and SAT-solvers [3] that allow manipulating boolean logical formulas. Satisfiability Modulo Theory (SMT) solvers extend SAT-solvers with non-boolean variables (e.g. rational numbers, integers, or other data structures).

Aniah proposed a graph based algorithm to detect electrical errors in a hierarchical design circuit. In this regard, the algorithm first assigns a finite set of values to the input variables of the circuit. Then, by analyzing the behavior of each net within the circuit, the algorithm detects electrical errors. One of the main issues in this analysis is the time and space complexity that is exponential with respect to the size of input variables. While the existing algorithm is usually fast enough in practice thanks to the good properties of the circuit topology, we are working on using symbolic model checking tools (BDD, SAT- and SMT-solvers) to speed up verification even more, as has been done in previous works [12, 11].

[1] https://www.aniah.fr/

[2] **C**onventions **I**ndustrielles de **F**ormation par la **RE**cherche, a joint Ph.D between a private company and public laboratories

Going for another approach, we currently have a prototype tool [6, 10] that compiles a circuit description into a logical formula comprising both numerical variables (representing voltage values) and booleans, that we solve using the Z3 [9] SMT solver. Z3 either proves the unsatisfiability of the formula (which mean that no electrical state can lead to an error), or provides a model for the formula. The model represents an electrical configuration leading to an error, and can be displayed to the user as a diagnosis. Unfortunately, this provides only a single electrical configuration, while a user may want to see all configurations leading to the same error to properly fix the root cause of the issue. We already experimented a *blocking* strategy to query Z3 repeatedly, adding the negation of the model as a clause to avoid getting the same model multiple times. However, this leads to an enumeration of model that can be very large, or even infinite, which is counter-productive as a debugging tool.

Also, typical errors in circuits usually cause a cascade of other errors. The user should not be bothered with these secondary errors before the root cause is fixed.

Objectives of the Ph.D

The objective of the thesis is to provide tools to help the precise diagnosis and debugging of errors. Among the topics to study:

- Analyze the existing tool and blocking strategy
- Implement a new strategy to list electrical configurations corresponding to each error. A first strategy, like "enumerate electrical configurations leading to different transistor states for all transistors on the path from the error to any supply" can be implemented, but further development should be done after discussions with Aniah's engineers to understand better the need of electrical engineers.
- Complement the model enumeration with further analysis, to understand the cause-effect relationships between errors. For example, a short circuit or a floating net can be the cause of other errors in other parts of the circuit. These other errors should not be investigated before their cause, because fixing the cause should also fix them. An analysis of the circuit giving the graph of cause-effect relationship between errors would be a key tool to help debugging.

As part of a CIFRE Ph.D, the candidate is expected to work closely with Aniah's engineers, to understand their needs and make the link with the academic side.

Context of the Collaboration and Physical Location

The Ph.D is proposed as part of the collaboration between LIP laboratory (Lyon), Verimag laboratory (Grenoble), and Aniah company (Grenoble). A CIFRE Ph.D (Oussama Oulkaid) student is already working on a related subject, along with one of the supervisor (Bruno Ferres), who developed the original prototype during his post-doc [6, 10]. The student recruited for this Ph.D will interact closely with them.

The thesis' goal is to use theoretical tools for a very practical concern, that is to provide the best possible tool for debugging. Depending on the student's motivation, the Ph.D can focus more on theory, implementation, or requirement analysis with some discussions with the actual users.

The Ph.D is proposed by LIP, Verimag and Aniah, as part of a CIFRE (industrial) collaboration. The physical location of the thesis is to be discussed with applicants. The student will visit other sites and meetings with all co-supervisors will be organised frequently.

- Laboratoire de l'Informatique du Parallélisme (LIP) – École Normale Supérieure de Lyon.
- Laboratoire Verimag, Grenoble.
- Aniah, Grenoble.

Required profile

The candidate should be familiar with algorithm design, understand the basics of Boole's algebra and logic as well as SAT/SMT solving. Good programming skills are required for the experimental validation of the approach. Since the software prototype is implemented in OCaml, prior knowledge of OCaml is appreciated, but the student can learn OCaml's basics during the thesis. While the application domain is electronics, no knowledge of electronics is required.

How to apply

Send an email to matthieu.moy@univ-lyon1.fr and bruno.ferres@univ-grenoble-alpes.fr with your CV, a short text describing your motivation, and any document that can support your application.

Advisors

- Matthieu Moy (`https://matthieu-moy.fr/`) will be the Ph.D supervisor (50% of the academic supervision).

  He is associate professor HDR at UCBL / LIP laboratory, Lyon. He has a long experience in program verification by model-checking abstract interpretation, applied to either general programs or models of Systems-on-a-Chip. A large part of his research deals with close-to-hardware computer science. He leads the CASH (`http://www.ens-lyon.fr/LIP/CASH/`) research team.
- Bruno Ferres (`https://ferres.me/`) will co-supervise the Ph.D (50% of the academic supervision).

  He is associate professor at UGA / Verimag laboratory, Grenoble. He has been involved in the collaboration with Aniah since the beginning of the project, and is an expert of the prototype that has been developed for ERC analysis. His research includes digital hardware design and verification as well as secure compilation of programs.

The Ph.D is proposed in partnership with Aniah, our industrial contact is Mehdi Khosravian Ghadikolaei (`https://www.linkedin.com/in/mehdikhosravian/`). He defended his Ph.D in algorithmic graph theory in 2019, and joined Aniah as Algorithm Engineer and Hierarchical Graph Analyst in 2020. He is currently working on a prototype of algorithm for a future version of Aniah's tool.
The Ph.D candidate will have access to some confidential data from Aniah (test cases or source code of the tool) if relevant.

Matthieu Moy and Bruno Ferres are from different laboratories (LIP and Verimag), but already co-supervised a Ph.D together, along with Mehdi Khosravian Ghadikolaei [6, 10].

References

[1] Thomas Ball, Vladimir Levin, and Sriram K Rajamani. A Decade of Software Model Checking with SLAM. *Communications of the ACM*, 54(7):68–76, 2011.

[2] Ilan Beer, Shoham Ben-David, Cindy Eisner, and Avner Landver. RuleBase: An Industry-Oriented Formal Verification Tool. In *33rd Design Automation Conference Proceedings, 1996*, pages 655–660. IEEE, 1996.

[3] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. Symbolic Model Checking without BDDs. In *International conference on tools and algorithms for the construction and analysis of systems*, pages 193–207. Springer, 1999.

[4] Aaron R Bradley. SAT-based Model Checking without Unrolling. In *International Workshop on Verification, Model Checking, and Abstract Interpretation*, pages 70–87. Springer, 2011.

[5] Jerry R Burch, Edmund M Clarke, Kenneth L McMillan, David L Dill, and Lain-Jinn Hwang. Symbolic Model Checking: 1020 States and Beyond. *Information and computation*, 98(2):142–170, 1992.

[6] Bruno Ferres, Oussama Oulkaid, Ludovic Henrio, Matthieu Moy, Gabriel Radanne, Pascal Raymond, and Mehdi Khosravian. Electrical Rule Checking of Integrated Circuits using Satisfiability Modulo Theory. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–2. IEEE, 2023.

[7] Patrice Godefroid. Software Model Checking: The VeriSoft Approach. *Formal Methods in System Design*, 26(2):77–101, 2005.

[8] Daniel Kroening and Michael Tautschnig. CBMC–C Bounded Model Checker. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 389–391. Springer, 2014.

[9] Leonardo de Moura and Nikolaj Bjørner. Z3: An Efficient SMT Solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

[10] Oussama Oulkaid, Bruno Ferres, Matthieu Moy, Pascal Raymond, Mehdi Khosravian, Ludovic Henrio, and Gabriel Radanne. A Transistor Level Relational Semantics for Electrical Rule Checking by SMT Solving. In *Design, Automation and Test in Europe Conference*, Valencia, Spain, 2024. Available at https://hal.science/hal-04527225/file/date2024.pdf.

[11] S Rodriguez-Chavez, AA Palma-Rodriguez, E Tlelo-Cuautle, and SX-D Tan. Graph-based Symbolic and Symbolic Sensitivity Analysis of Analog Integrated Circuits. In *Analog/RF and Mixed-Signal Circuit Systematic Design*, pages 101–122. Springer, 2013.

[12] Guoyong Shi. A Survey on Binary Decision Diagram Approaches to Symbolic Analysis of Analog Integrated Circuits. *Analog Integrated Circuits and Signal Processing*, 74(2):331–343, 2013.

[13] Wikipedia contributors. Model checking — Wikipedia, the free encyclopedia, 2021. [Online; accessed 21-September-2021].