ETS'18 Introduction and previous work

December 3, 2017

1 Introduction

Verifying functional correctness of gate level arithmetic circuits is a challenge due to increasing complexity and requires manual intervention to localize a bug and add correction, since it is still a resource intensive process. Traditional automated debugging techniques based on simulation and decision diagrams such as BDDs and BMDs, suffer from exponential blow-up, while theorem provers require extensive manual intervention and expertise. Computer algebra methods are believed to be the best techniques for solving such arithmetic verification problems and a lot of effort is spent on modeling automated formal verification methods to prove the correctness of such designs.

Given a circuit implementation C and a golden specification f, we do verification. If the verification fails, we deem the circuit as buggy and go on to find the faulty gate in order to rectify it. Identifying the buggy gate is a much harder problem to solve and is in the future scope of work. The current challenge is to realize the correct implementation for this buggy component. Once a particular gate has been identified as suspicious, we label the gate as an unknown component and the next step would be to find the correct functionality implemented by this unknown component such that it conforms to the given golden specification. The reference golden model can either be a specification polynomial f or a structurally different circuit C_1 implementing the same function. Both the notions will be addressed by analyzing the circuit polynomials using concepts from computer algebra[1][2] such as *Gröbner* basis based reduction, ideal membership testing, and weak *Nullstellensatz*.

The most recent and relevant approach [3][4] resolves an unknown component using an incremental *Boolean Satisfiability*(SAT) formulation with the help of LUTs. The work in [5] poses the unknown component circuit as a camouflaged model and tries to de-obfuscate several types of camouflaging techniques using incremental SAT solving. The approach used in [6] inserts logic corrector MUXs on the unknown sub-circuits and relies on SAT solvers to realize the functionality. Despite using the stat-of-the-art SAT solvers, all these approaches fail to verify large and complex arithmetic circuits. Techniques from Farimah et. al [7][8] deals with automatic debugging and correction using computer algebra concepts. The coefficient computation concept in gate correction section borrowed from[9] relies heavily on the half adder textbook structure and doesn't

talk about the ambiguities in weight calculations when the gate structure differs from the given topology. The approach also fails to arrive at a conclusive solution when the circuit is tweaked with some redundancy and hence lacks completeness.

In this paper, we utilize the concepts from computer algebra techniques to realize the unknown component, prove the completeness of the approach, and go on to show on how this approach can be extended to any arbitrary random logic circuit. For simplicity, we shall take a single gate replacement error model as our target design i.e.,only one gate in the design has been incorrectly replaced, for example an AND gate replaced with an XOR/OR gate. For the given specification polynomial f, we do a polynomial reduction until the unknown component gate and then use a *Gröbner* basis based guided ideal membership testing to arrive at the function implemented by the component. For the case where the specification is given in terms of a different implementation C_1, we use *Gröbner* basis based reduction on a miter setup and apply *Nullstellensatz* principles to arrive at the function implemented by the unknown component. This paper seeks to outline the verification challenges and presents an approach for completeness with some preliminary but encouraging experimental results.

References

[1] W. W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*. American Mathematical Society, 1994.

[2] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.

[3] M. Fujita, "Toward Unification of Synthesis and Verification in Topologically Constrained Logic Design," *Proceedings of the IEEE*, 2015.

[4] S. Jo, T. Matsumoto, and M. Fujita, "SAT-Based Automatic Rectification and Debugging of Combinational Circuits with LUT Insertions," in *IEEE 21st Asian Test Symposium*, 2012.

[5] C. Yu, X. Zhang, D. Liu, M. Ciesielski, and D. Holcomb, "Incremental SAT-Based Reverse Engineering of Camouflaged Logic Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[6] A. Smith, A. Veneris, M. F. Ali, and A. Viglas, "Fault Diagnosis and Logic Debugging using Boolean Satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005.

[7] F. Farahmandi and P. Mishra, "Automated Debugging of Arithmetic Circuits Using Incremental Gröbner Basis Reduction," in *IEEE International Conference on Computer Design (ICCD)*, 2017.

[8] F. Farahmandi and P. Mishra, "Automated Test Generation for Debugging Arithmetic Circuits," in *Design, Automation Test in Europe Conference Exhibition(DATE)*, 2016.

[9] S. Ghandali, C. Yu, D. Liu, W. Brown, and M. Ciesielski, "Logic Debugging of Arithmetic Circuits," in *IEEE Computer Society Annual Symposium on VLSI*, 2015.