Execution Trace Mining for SoC Validation for Safe and Secure IoT Edge Node Design

Md Rubel Ahmed, Yuting Cao, Hao Zheng; University of South Florida

2nd IFIP International IoT Conference
Tampa, Florida, USA
October 31st – November 1st, 2019

The SEES Lab

Introduction

SoC is the hardware foundation for IoT edge nodes. Ensuring security properties such as confidentiality and integrity is crucial for the trustworthiness of IoT devices. However, due to the high complexity of the global supply chain, ensuring the trustworthiness of diverse third party suppliers becomes very much challenging. Thorough validation of SoC foundation is critical to guarantee the safety and security of those IoT edge nodes. Comprehensive and well-defined specifications are necessary to perform rigorous and thorough validation of SoC designs. However, in reality, such specifications are hardly available, often incomplete and ambiguous[1]-[7]. In this work, we aim to address such a challenge by proposing a sequential pattern mining framework to automatically extract message flow specifications.

Background

An SoC is a combination of reactive components, called IPs that work together to complete a set of intended tasks.

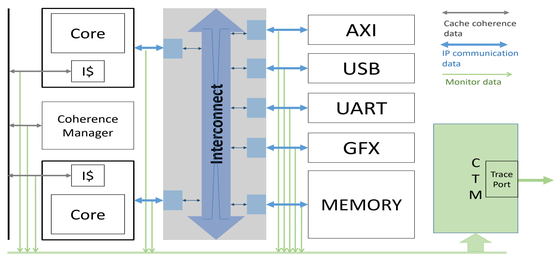

Fig. 1: An SoC prototype with different IPs

We can view a task as a message flow specification, for example, CPU downstream write.

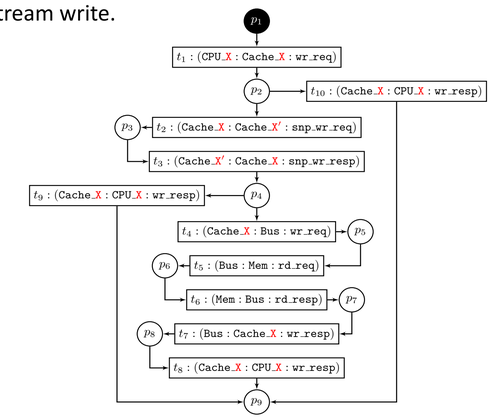

Fig. 2: LPN formulation of a CPU write flow

Problems Addressed

1. Post-silicon validation
2. Specification mining
3. False positive specification
4. Specification mining time

We characterize the patterns to be mined as:

- ❑ Set of events
- ❑ Strong temporal dependency
- ❑ In constant environment, each execution holds the rules

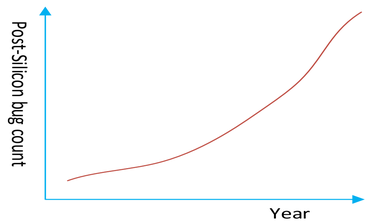

Fig. 3: Cost of Silicon validation getting worse (source: intel)

Proposed algorithm works on execution traces captured by monitoring the messages among the IPs of an SoC.

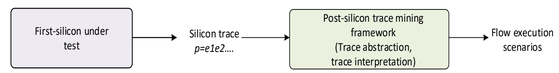

Fig. 4: Post-silicon trace mining

Mining Framework Flowchart

We utilize association rule mining technique to mine sequential patterns from the execution traces. We also apply domain specific heuristic to reduce the huge search space of association rules.

Mine Patterns: Using 100% confidence and recall to mine assertions that hold over all traces. The reason to mining assertion is to find out flows implemented by the IoT hardware core, which are invariant over different trace.

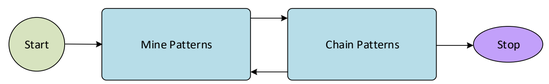

Fig. 5: Mining framework

Chain Patterns: We apply 3 inference rules two chain shorter patterns to form complex patterns. The iteration between mining and chaining keeps going until all the valid rules are found upto a user defined length *l*.

Challenges

- ❑ High number of false positive patterns
- ❑ Difficulties with branches
- ❑ Long execution time
- ❑ Need for perfect traces
- ❑ High concurrency yields poor correlation

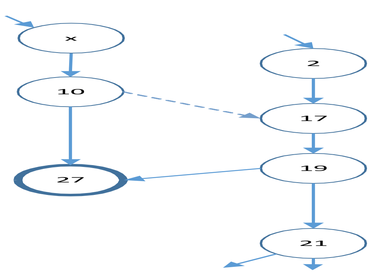

Fig. 5: Possible branch in flows

Mining sequential patterns of longer lengths has always been a challenging task, especially for concurrent systems that are also recurrent. Hence our proposed algorithm shows promising result reducing the number of mined patterns

Pattern Length	Total Search Space	Mined Patterns
2	1806	182
3	74046	115
4	2961840	290
5	115511760	495
6	4.38944E+9	969
7	1.62409E+11	1538
8	5.84674E+12	3341

Tab. 1: Result analysis from 57 distinct message of an SoC

Conclusion

We mine strict ordering relations among the events. The mined patterns will help to find violations for SoC internal communication protocols. IoTs are sensitive systems in terms of security and trustworthiness. Proposed framework will play an important role making the task of edge node verification easier. We are currently dealing with branching problems that may cause some valid execution flows to be missing due to high strictness in confidence and recall measure.

References

[1] Sandip Ray, Ian G. Harris, Goerschwin Fey, and Mathias Soeken. Multilevel design understanding: From specification to logic invited paper. In *Proceedings of the 35th International Conference on Computer-Aided Design*, ICCAD '16, pages 133:1–133:6, 2016.
[2] W Chen, S. Ray, J Bhadra, M Abadir, and Li-C Wang. *Challenges and trends in modern soc design verification*. IEEE Design Test, 34(5):7–22, Oct 2017.
[3] Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *Proceedings of the 21st International Conference on Software Engineering*, ICSE'99, pages 411–420, 1999.
[4] Glenn Ammons, Rastislav Bod ık, and James R. Larus. Mining specifications. In Proceedings of the 29th ACM *SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '02, pages 4–16,2002.
[5] Po-Hsien Chang and Li.-C Wang. Automatic assertion extraction via sequential data mining of simulation traces. In *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, ASPDAC'10,pages 607–612, 2010.
[6] Wenchao Li, Alessandro Forin, and Sanjit A. Seshia. Scalable specification mining for verification anddiagnosis. In *Proceedings of the 47th Design Automation Conference*, DAC'10, pages755–760, New York, NY, USA, 2010. ACM.
[7] Samuel Hertz, David Sheridan, and Shobha Vasudevan. Mining hardware assertions with guidance fromstatic analysis. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 32(6):952–965, June 2013.

Please scan for more information: