Is Single Trojan Detection Scheme Enough?

Yier Jin, Nathan Kupp, Yiorgos Makris
Department of Electrical Engineering, Yale University, 10 Hillhouse Avenue, New Haven, CT 06520, USA

Abstract

In this report, we proposed a new type of circuit attacking method which is not targeting the circuit itself but trying to mute the internal hardening technique. By implementing this attacking scheme, we argue that most of the currently proposed hardware Trojan prevention methods are far from security if we assume that attackers are patient and smart. As part of our work in the CSAW Embedded System Challenge hosted by NYU-Poly this year, we demonstrated that attackers can easily construct testing patterns by "reverse engineering" the hardened RTL code. A simple look up table can invalidate the sophisticated RO-based Trojan prevention method. We believe that any single Trojan prevention scheme is not enough to keep hardware Trojan out of the door and only the combination of several methods is a possible solution.

I. Introduction

The threats of hardware Trojan have attracted more and more researchers to work in this field and various Trojan detection methods and Trojan prevention schemes have been proposed [1], [2], [3], [4], [5], [6]. The guideline of these proposed methods is that they all tried to cover the shortage of current testing procedure to counter the special characteristics of hardware Trojan which are list below:

1) Unanticipated behavior is not included in the fault list, i.e., structural pattern testing will likely not cover Trojan test vectors [2];
2) Additional functionality of genuine designs is hard to predict without knowledge of the Trojan inserted by attackers. Hence, routine functional testing is unlikely to reveal harmful extra functions;
3) Exhaustive input patterns testing is impractical as chips become more complicated with a large number of primary inputs and inner gates.

One main trend in the field is to embed Trojan prevention scheme inside the chip to increase the burden of attacking and finally lead to the detection of hardware Trojan [7]. Although these methods have been proved successful in detecting inserted malicious circuits which may escape traditional functional and structural testing, there is a huge black hole here that researchers pretend not to know. That is, attackers should be some one who are of both patience and intelligence. Before modifying the circuits, attackers will first carefully scrutinize the whole circuit and analyze both the original circuit and Trojan detection schemes, if any. As a consequence, to blindly trust our Trojan prevention scheme as a complementary to traditional testing method is surely not enough to protect the whole circuit if these Trojan prevention schemes are lacking of some key characteristics. Those characteristics include

1) Low overhead. If the inserted protection scheme consumes too much power and area, the performance of the chip will be down graded to make the method less attractive.
2) High sensitivity. The Trojan detection scheme should be of high sensitivity in detecting any malicious modification.
3) Full knowledge. This is the most important part of a successful Trojan prevention scheme. More clearly, the designer should always assume that attackers have full knowledge of the mechanism of the Trojan protection scheme.

However, most of currently proposed hardware Trojan prevention schemes considered little of the third item - full knowledge. In this report, we will to demonstrate that any Trojan prevention scheme based on the assumption that attackers have limited knowledge of the method itself will easily invalid the sophisticated Trojan prevention scheme. The target system is provided by NYU-Poly as part of the CSAW Embedded

System Challenge, a carry look-ahead adder (a.k.a Beta Design) [1]. In the rest of the report, we will first analyze the embedded Trojan prevention scheme and presents the working mechanism of those schemes (note that as a competitor, we have the HDL code of the hardened design but are not told about the details of the protection scheme so that all conclusions we will give are made from a "reverse-engineering" style by analyzing the code itself). Shortages of these schemes will also be list and are used as the guideline to design hardware Trojans. A muting system is developed to mute the protection schemes. We argues that if our muting system works, we can insert as many Trojan as we want to the target circuit without worrying about being detected by the Trojan prevention system. Finally, we discuss possible solutions to overcome the shortage by implementing single Trojan prevention scheme.

II. Beta Design

A. Hardening Technique Analysis

The Beta Design for this CSAW competition consists of an implementation of a 4-bit adder. There are 3 different levels of difficulty: easy, medium, and hard. The HDL code is written in both Verilog and VHDL. The user will use interface with the FPGA using 8 switches and 3 push button switches. After the user enters a test vector, the frequency of the ring oscillator is displayed onto the 7-segment display.

The design is hardened by embedding a ring oscillator. Any changes made in the design is reflected in the change in the frequency of the ring oscillator. If the change in the frequency is greater than 6.6% [8] (due to process variation) in the FPGA, the trojan is detected. The difference between easy, medium and hard version of Beta design lies on the difference of the number of ring oscillators (ROs) that are embedded. Since the easy and medium version hardening techniques are no more than subsets of the hard version, we only analyze and attack the hard version here. Again, we argue that if our hardening technique muting method is valid for hard version, it will also be valid for both easy and medium versions.

In the hard version, a total of 6 MUXes are inserted into original carry look-ahead circuit (see Figure 1 to generate 6 ROs with appropriate control signals.

Although we have full access to the HDL code, we don't know what testing vectors will be used in the testing stage. That is, judgers hold the testing patterns as a surprise to attackers and hope that any blindly inserted Trojan will violate original timing of the internal loops. But this assumption is too weak because the attackers can easily go step by step to construct testing patterns themselves from the hardened circuit.

B. Muting Hardening Technique

In this competition, we first write a Verilog-based test bench to generate us all input patterns which will trigger one or more ROs to oscillate. Table I shows part of the generated testing pattern list as well as the consequential oscillating ROs and their measured frequency. By analyzing the testing pattern table, we have several findings:

1) Under same input pattern, the oscillating output signal will flip at similar frequency. This phenomena can be easily explained that under certain input pattern, only one loop is constructed so that any signals connect to that loop should flip at the same frequency (in reality, due to noise and measurement error, the measure frequencies are not stable but the variation is insignificant compared to process variation).
2) Under different input patterns, as far as the loop control signal is the same (e.g., with the same `CKTM` signal in Table I), the oscillating output signal will flip at similar frequency. This is because with same loop control signal, the consequential internal loops share the same path.

As there are only 6 ROs embedded in the adder, we only need to prepare 6 frequency values mapping to different loop controlling signal `CKTM` [2].

[1] we did not discuss the other target circuit, a Tiny Encryption Algorithm (TEA) core (a.k.a Alpha Design) in this report because the protection scheme is much simpler compared to that in Beta design.

[2] In the competition, we actually use 7 different frequency values because the number of oscillating outputs will affect the frequency value for some input patterns.

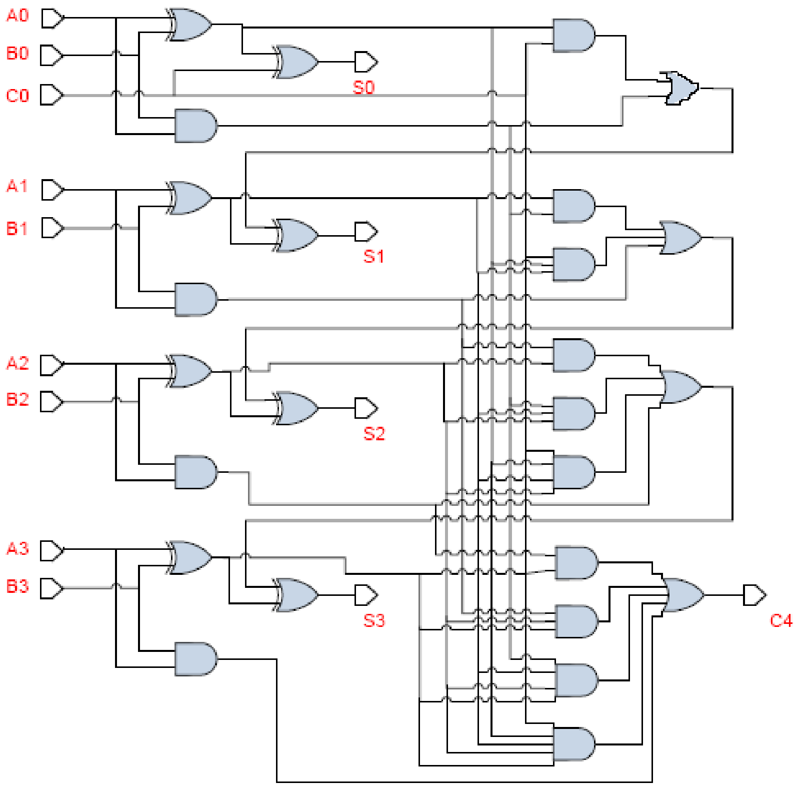

Fig. 1. The gate level schematic of 4-bit carry look-ahead adder

TABLE I

TESTING PATTERNS FOR CARRY LOOP-AHEAD ADDER

TE (MUX Selection)	Inputs (A[3:0], B[3:0], C0)	Toggling Outputs (Freq)
TE0 (SEL = 000001, CKTM=000)	0000,0000,1	S0(4C5E), OUTM = 100 S1(4C51), OUTM = 011
TE0 (SEL = 000001, CKTM=000)	0000,0001,1	S0(4C56), OUTM = 100 S1(4C65), OUTM = 011
TE0 (SEL = 000001, CKTM=000)	0000,0100,1	S0(4C50), OUTM = 100 S1(4C52), OUTM = 011
...	...	...

In order to mute the hardening technique, we construct a simple look up table inside the chip. In the testing stage, a testing pattern which is supposed to control internal loop and calculate the loop frequency will be re-directed to the inputs of the loop-up table. A fake frequency value is read from the table. If we merely compare this frequency value with what we get from the golden model, we cannot detect any abnormality. One problem of the hardening technique muting method is that the result is too good to be true because the measurement noise will change the loop frequency for each testing even with the same input, but the value read from the table won't change. In order to cover this problem, we insert a random number generator inside the chip as a complementary part of the muting system. Within each test, a random number will be generated and the final result is the addition of the random number and the value from the table.

Thus far, we hacked the whole hardening technique with preset frequency values to replace the “true” values as the testing outputs. With the hardening technique muting method being implemented in the chip, we can do any modifications we want to the chip without worrying about being detected by the Trojan prevention scheme. Actually, there is no Trojan prevention scheme any more.

III. Discussion and Conclusion

From the above discussion we demonstrated that currently proposed hardware Trojan prevention methods are far from security if we assume that attackers are patient and smart (which is always true, unfortunately). An simple look up table can invalidate the sophisticated RO-based Trojan prevention method. However, we should not simply conclude that currently proposed hardware Trojan detection/prevention methods are useless. One solution here is to combine several Trojan prevention schemes together to construct a more robust system. Taking our hardening technique muting method for example, although it can invalid the RO-based Trojan prevention scheme, it may be detected by power-based detection method as the extra look up table added in the chip.

Acknowledgements

We would also like to thank the competition sponsors, NYU-Polytechnic, for their provision of the hardware and source code.

References

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *IEEE Symposium on Security and Privacy*, 2007, pp. 296–310.

[2] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards Trojan-free trusted ICs: Problem analysis and detection scheme," in *IEEE Design Automation and Test in Europe*, 2008, pp. 1362–1365.

[3] H. Salmani, M. Tehranipoor, and J. Plusquellic, "New design strategy for improving hardware Trojan detection and reducing Trojan activation time," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 66–73.

[4] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 51–57.

[5] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware Trojans," in *IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 632–639.

[6] R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity analysis to hardware Trojans using power supply transient signals," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 3–7.

[7] *http://www.poly.edu/csaw-embedded*.

[8] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of ro-puf," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, 2010, pp. 94–99.