Is Single-Scheme Trojan Prevention Sufficient?

Yier Jin* and Yiorgos Makris†
*Department of Electrical Engineering, Yale University
†Department of Electrical Engineering, The University of Texas at Dallas
{yier.jin@yale.edu, yiorgos.makris@utdallas.edu}

***Abstract*—We discuss a new type of a structural hardware Trojan, which does not attack the target circuit itself but tries to mute the internal hardening scheme instead. By implementing this type of hardware Trojan, we argue that most of the currently proposed hardware Trojan prevention methods are far from adequate, assuming that attackers are patient, smart and have basic knowledge of the hardening structure. As demonstrated through our work for the CSAW Embedded System Challenge hosted by NYU-Poly in 2010, attackers can easily construct test patterns to "reverse-engineer" the hardening scheme from the Register Transfer Level (RTL) description. A simple look-up table can then invalidate the hardening scheme, even if it is as sophisticated as the Ring Oscillator (RO)-based Trojan prevention method used in this competition. Hence, our conjecture is that any single-scheme Trojan prevention method is insufficient to keep hardware Trojans out of the door and only a combination of several methods is a plausible solution.**

I. Introduction

The recently publicized threat of hardware Trojans has attracted several researchers who have been investigating the problem and developing various countermeasures. Various Trojan detection methods and Trojan prevention schemes have been proposed [1], [2], [3], [4], [5], [6] and a comprehensive hardware Trojan taxonomy is presented in [7]. The basic principle of these methods is that they try to alleviate the shortcomings of traditional manufacturing testing, which fails to uncover hardware Trojans for following reasons:

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 pattern testing is impractical as chips become more complicated with a large number of primary inputs and inner gates.

A main emerging trend is to embed a Trojan prevention scheme inside the chip in order to increase the burden of attacking and finally lead to the detection of hardware Trojans [8]. While these methods have been proven successful in detecting inserted malicious circuits which may escape traditional functional and structural testing, their overall effectiveness depends on the skill, resources, and patience of the attacker. Although the implementation details of the hardening scheme are kept secret, the additional pins and extra internal logic may (partially) give away the structure of the prevention scheme. From the attackers' point of view, it is therefore prudent to first carefully scrutinize the entire circuit in order to separate the Trojan detection scheme from the actual logic, before maliciously modifying the circuit. As a consequence, it is rather shortsighted to assume that a Trojan prevention scheme used as a complementary enhancement to traditional manufacturing testing will ensure trustworthiness of a chip, especially when such Trojan prevention schemes lack some key characteristics including:

1) Low overhead: If the inserted protection scheme consumes too much power and/or area, the performance of the chip will be downgraded, making such methods unattractive.
2) High sensitivity: The Trojan detection scheme should be of high sensitivity in detecting malicious modifications.
3) Full knowledge: This is the most important part of a successful Trojan prevention scheme. Specifically, the designer should always assume that attacker has full knowledge of the Trojan protection mechanism.

Most of the existing hardware Trojan prevention schemes, however, have paid little attention to the third characteristic - full knowledge. In this paper, we demonstrate that the assumption that attackers have limited knowledge of the method itself can easily undermine and eventually incapacitate even the seemingly most sophisticated Trojan prevention schemes. The pitfall of this assumption is that, by carefully analyzing the hardened circuit, attackers can uncover the structure of the inserted hardening logic. And although it may be impossible to also recover the details of the trustworthiness test procedure from the revealed structure, attackers can selectively simulate a subset of all input patterns to record all possible responses and eventually mimic the behavior of a Trojan-free design. The demonstration system we will use in this work is the carry look-ahead adder (a.k.a Beta Design) provided by NYU-Poly as part of the CSAW Embedded System Challenge[1].

In the remainder of the paper, we first analyze the embedded Trojan prevention scheme and present the working mechanism of the protection scheme from the HDL codes. We note that, as part of the competition, we were given access to the HDL code of the hardened design but were not provided with details of the protection scheme. Hence, all conclusions we give are obtained by "reverse-engineering" the provided HDL code. Limitations of this scheme are also discussed in order to help us develop a muting technique to invalidate the Trojan prevention method. We argue that if our muting system works, we can insert as many functional Trojans as we want into the target circuit without worrying about being detected by the Trojan prevention scheme. Finally, we

[1]We do not discuss the other target circuit (a Tiny Encryption Algorithm (TEA) core (a.k.a Alpha Design)) in this paper because it involves a different protection scheme which is not the focus of this work.

978-1-4577-1954-7/11/$26.00 ©2011 IEEE

discuss possible solutions to overcome the shortcomings of implementing single-scheme Trojan prevention.

II. Target Circuit

The target circuit consists of a 4-bit look-ahead adder and its Trojan prevention method of 3 different levels of difficulty: easy, medium, and hard, For reasons that will be explained after we analyze the hardening structure, we only consider the hard version of the target circuit in this paper. The HDL codes are written both with Verilog and VHDL. The gate-level structure of the 4-bit adder, which has 9 inputs and 5 outputs, is shown in Figure 1 (not including the additional inverter and multiplexor).

A. Hardening Scheme Analysis

A careful analysis of the HDL codes reveals that the hardening scheme is a ring oscillator-based Trojan prevention method. Similar methodologies and related analysis can also be found in [9]. However, different from previous approaches which insert entire ROs into the circuit, designers in this case only add inverters and Multiplexors (MUXs) to connect existing gates and construct internal loops.

Rather than detecting additional malicious circuitry in an indirect way by inserting ring oscillators into the design, these internal loops highly improve sensitivity in detecting hardware Trojans because any modification will directly change the internal architecture and result in significant frequency changes. In the worst case, insertion of malicious circuitry may mute the ring oscillators. The coverage rate, which is defined as the percentage of the on-chip area covered by the constructed ring oscillators, is adjustable in order to control the area overhead of this hardening method. For example, the easy, medium and hard hardening schemes contain 2, 4 and 6-ring oscillators, respectively. Among them, the low protection level (2 ring oscillators) using 2 MUXs and 2 inverters covers 62% area of the original design (16 out of 26 gates). The medium protection level (4 ring oscillators) using 4 MUXs and 4 inverters covers 85% area of the original design (22 out of 26 gates). The high protection level (6 ring oscillators) with 6 MUXs and 6 inverters covers 92% area of the original design (24 over 26 gates). Figure 1 shows one sample ring oscillator in the 4-bit carry look-ahead adder constructed by an additional inverter, an additional MUX and three gates from the original design (an XOR, an AND and an OR gate). When the ring oscillator control signal `RO` is '0', the circuit performs its normal functionality. When it is set to '1', however, the ring oscillator starts oscillating under certain input patterns. Testers can then measure the frequency of the constructed ring oscillators from primary outputs.

Any malicious modification made in the design is reflected by a frequency change in one or more ring oscillators. If this frequency change is greater than 6.6% [10] in an FPGA implementation, the tester will then claim that a Trojan is detected. Since the easy and medium version hardening schemes are simply subsets of the hard version, we only analyze and attack the hard version here. Essentially, we argue that if our hardening scheme muting technique is valid for the hard version, it will also be valid for both the easy and the medium versions.

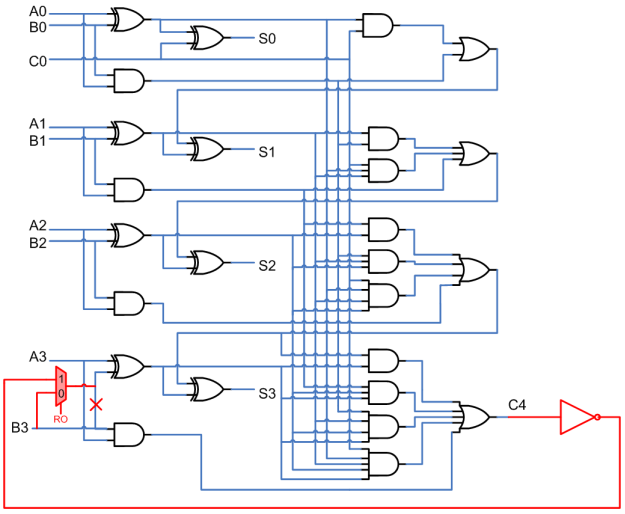

Fig. 1. The gate-level schematic of the 4-bit carry look-ahead adder and one constructed RO

The entire system is implemented on a Digilent Basys2 Board with a Xilinx Spartan-3 FPGA and other peripheral circuits [11]. The board also provides an easy-to-handle interface (three push button switches and four 7-segment displays) for testers to quickly get the precise frequency information for each constructed RO.

While we have full access to the HDL codes, we do not know what vectors will be used in the testing stage. In other words, designers/testers hold the testing patterns as the element of surprise to defend from attackers, hoping that any inserted hardware Trojan will violate the valid frequency ranges of the internal loops. Together with the previously mentioned advantages, this RO-based Trojan prevention scheme seems to be a very effective method for protecting digital circuits. However, the argument of its effectiveness is valid only if attackers try to modify the circuit disregarding the existence of the Trojan prevention scheme. Indeed, for attackers who would first scrutinize the Trojan prevention structure, this method may actually turn out to be quite vulnerable, as we show in the next section.

III. Muting Technique

Different from functional hardware Trojans, the Trojans we propose in this paper are structural, i.e. they do not attack the original circuit directly but try to mute the Trojan prevention scheme instead. In order to differentiate between structural and functional Trojans, we use the term "muting technique" rather than structural hardware Trojan hereafter. After muting the protection scheme, attackers can insert any kind of hardware Trojan to alter the functionality of the original circuit. Interested readers can refer to [12] for more information on how to design functional hardware Trojans.

A. Simulating RO Frequencies

Using the hardened RTL code as input, we first write a Verilog-based testbench to generate all input patterns which trigger oscillation of one or more ROs. These simulation patterns are then grouped into six categories corresponding to triggers for each of the six internal ROs.

Table I shows part of the generated testing pattern list as well as the pertinent oscillating ROs and their measured frequency. `TE` and the corresponding `CKTM` are the selection signals to activate a certain oscillating RO. The frequency of the selected RO is then shown on the LED display and can be recorded by testers/attackers.

B. Analyzing RO Frequencies

As we mentioned earlier, the testing patterns used to assess trustworthiness of a chip are kept secret by the designers. Thus, in order to invalidate the protection scheme, the attacker needs to "guess" the testing patterns. Table I provides an excellent starting point to duplicate the procedure through which circuit designers assess trustworthiness. A thorough analysis of the patterns and frequency responses from the table reveals that:

1) Under the same input pattern, different oscillating output signals will flip at similar frequency. This can be easily explained since under certain input patterns only one loop is constructed, so any output signals (i.e., `S0, S1, S2, S3` and `C4`) connected to that loop flip at the same frequency (in reality, due to noise and measurement error, the measured frequencies are not identical but the variation is insignificant compared to process variation). For example, we can find in Table I that when the first RO is chosen (`TE0='1'`) and input pattern is `0000,0000,1`, `S0` and `S1` will flip at frequencies close to each other (`4C5E` vs. `4C51`[2].).
2) Under different input patterns, as long as the loop control signal is the same (i.e., with the same `CKTM` signal in Table I), the oscillating output signal will flip at similar frequency. This is because with the same loop control signal, the internal loops share the same path. For example, when the second RO is chosen (`TE1='1'`), input pattern `0000,1110,0` and `0000,1110,1` will produce almost the same frequency at `S3` output signal (`5028` vs. `5030` from Table I). One exception exists in the case `TE4='1'` where two different frequency ranges can be measured when two outputs (`S2, S3`) or three outputs (`S2, S3, C4`) are toggling under different input patterns.
3) Different ROs are of significantly different oscillating frequencies even under the same input pattern.

These three findings indicate that, if circuit designers wish to choose testing patterns with the highest coverage rate over the entire circuit, they will need to pick at least 7 patterns, each mapping to one internal RO (with one exception in the case `TE4='1'` where two patterns are required). We also conclude that more testing patterns, other than the selected seven, will not help in further improving detection ability. This finding convinces us that scalability will not be a problem when the target circuit becomes larger because the stored frequency values are only proportional to the internal RO counts.

[2]Both 4C5E and 4C51 are relative frequencies to the clock signal. An explanation on how to measure the RO relative frequency is discussed in [9].

C. Muting Hardening Scheme

Knowing that the designers will choose testing patterns from a table similar to Table I and that in order to maximize coverage they will prepare testing patterns covering all six ROs, we propose a muting technique by inserting a pre-defined look-up table (LUT) into the hardened design. The LUT only contains 7 values which represent the different frequencies of the six ROs. During the testing stage, patterns which are supposed to control an internal loop are re-directed to the inputs of the look-up table. A fake frequency value is then read from the table and provided to the output. A comparison of this frequency value to the one obtained from the golden model does not reveal any abnormality. One limitation of this muting method is that the result may appear to be too good to be true, since measurement noise is expected to slightly affect the loop frequency in each measurement, even with the same input; yet the value read from the table will always be exactly the same, possibly raising suspicions. In order to address this problem without increasing the size of LUT, we insert a light-weight random number generator inside the chip as a complementary part of the muting system, to simulate measurement noise. With each test, a small random number is generated and the final result is the sum of this small random number and the value read from the LUT. Figure 2 shows the structural modification of the target circuit with the proposed hardening scheme muting technique.

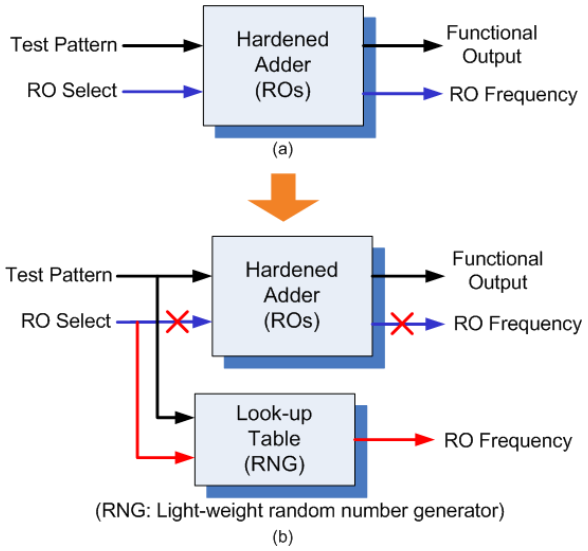

Fig. 2. The working procedure of (a) the original hardened adder, and (b) the Trojan-infested adder with a muted hardening scheme

Through the above method, we incapacitate the hardening technique by using preset frequency values to replace the "true" values as the testing outputs. With this hardening technique muting method implemented in the circuit, we

TABLE I
TESTING PATTERNS AND FREQUENCY RESPONSES FOR HARDENED 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
...	...	...
TE1 (SEL = 000010, CKTM=001)	0000,1110,0	S2(503E), OUTM = 010 S3(5028), OUTM = 001 C4(502D), OUTM = 000
TE1 (SEL = 000010, CKTM=001)	0000,1110,1	S2(502E), OUTM = 010 S3(5030), OUTM = 001 C4(5034), OUTM = 000
...	...	...
TE2 (SEL = 000100, CKTM=010)	0010,0101,1	S1(5F63), OUTM = 011 S2(5F4E), OUTM = 010 S3(5F68), OUTM = 001
TE2 (SEL = 000100, CKTM=010)	0000,1110,1	S1(5F3E), OUTM = 011 S2(5F4D), OUTM = 010 S3(505D), OUTM = 001 C4(5F2C), OUTM = 000
...	...	...

can now do any desired functional modification to the chip without worrying about being detected. In fact, no active Trojan prevention scheme exists any longer. Our claim is also supported by the result of the CSAW competition where the testing vectors, the secret weapon of the designer, cannot detect any malicious Trojans inserted into the circuit if the hardening technique is muted [8].

IV. CONCLUSION

We demonstrated that existing single-methods for hardware Trojan prevention are insufficient to thwart attacks by skilled and resourceful attackers. Even sophisticated Trojan prevention method, such as those based on internal Ring Oscillators, can be easily rendered inoperable through a simple look-up table mimicking their Trojan-free behavior. However, such hardware Trojan detection/prevention schemes can prove very effective as part of a constellation of multiple methods, which can cumulatively enhance the trustworthiness of the target circuits. Indeed, using our hardening scheme muting technique as an example, while it evades the RO-based Trojan prevention scheme it may still be detected by power-based detection methods, due to the look-up table and random number generator added to the design. The hardening scheme muting technique may also be detected under varied power supply schemes because the stored frequencies will not change when VDD changes. Overall, the conventional wisdom of setting up as many defenses as possible is the way to go in order to raise the barrier to entry for hardware Trojans.

ACKNOWLEDGEMENTS

We would like to thank Xilinx and the CSAW hosts at NYU-Poly for providing the FPGA platform and source code. This work was supported by the National Science Foundation (NSF-1017719) and the National Science Foundation travel grant (CNS-0958510).

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] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware Trojans," *Computer*, vol. 43, no. 10, pp. 39–46, 2010.

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

[9] Y. Jin, E. Love, and Y. Makris, *Book Chapter in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang editors*, chapter Design for Hardware Trust. Springer, 2011.

[10] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2010, pp. 94–99.

[11] *http://www.digilentinc.com*.

[12] Y. Jin, N. Kupp, and Y. Makris, "Experiences in hardware Trojan design and implementation," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 50–57.