Designing a Secured Hardware for Reconfigurable Systems on Chip by a Proof-Carrying Code Approach

SREE RANJANI R[1], A. NITHISH[2], S. SYED MUSTHAFA[3], M. SUDHARSUN[4]
[1, 2, 3, 4] *Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology*

Abstract: Hardware security is an emerging topic in integrated-circuit (IC) industries. Research in the domain of the hardware security is at a full swing and many schemes to enhance the security are being explored. The hardware Trojan (HT) design and its various detection techniques to ensure the trust in design are the most sought for schemes. The analysis of the reported techniques explores the major threat in the IC industries known as hardware Trojans and their countermeasures.

I. INTRODUCTION

Secured hardware is necessary to upgrade the performance, reliability and efficiency of any system. Globalization of the IC design flow is the main reason for hardware vulnerabilities. The fabless industries have to depend on the untrustworthy fabrication units where the attacker can easily access the implementation of IC at any stage in the original IC design. Some untrusted IC fabrication company may illegally overbuild ICs and sell them in the market or an attacker in a fabrication unit may add a malicious circuit (hardware Trojan) to the original design [1]. It is reported that a hardware attack causes a loss of $4 billion annually to the semiconductor industry [2]. These hardware-related security issues directly spoil the efficiency of the architectures where hardware plays a major role in implementation such as cryptographic applications [3]. Various threats and hardware Trojans are proposed and their deterring methods are analyzed in [4, 8 &9]. Hardware security includes detection and diagnosis of the hardware Trojans and design for secured hardware.

Active partial reconfiguration or dynamically reconfigurable hardware is the computer architecture combining some of the flexibility of software with the high performance of hardware. Also permits to change the part of the device while the rest of an FPGA is still running. The dynamic reconfiguration provides the network system to download new hardware and software. The downtimes are often unacceptable, so that we had to install the new hardware in specific way. For security and safety critical systems we use reconfigurable hardware where internal construction flaws can cause some consequences, that may be loss of human life, financial damages, and national security treats and so on.

Novel contribution on this paper is designing a secured hardware for reconfigurable system on chip. [5]Lee and Necula's Proof carrying code is the key concept which combines both the formal proof and software module. The untrusted external source produced the proof carrying hardware and delivered in unsecured way. The fraction effort is taken to verify the proof by reconfigurable platform. Without any previous guarantee the consumer trusted the module. Potential and feasibility of PCH shown experimentally in initial research.

II. PROOF-CARRYING CODE

PCC, Proof Carrying Code is the software mechanism which can able to execute the untrusted code in safest manner. In 1996,[5] PCC has been proposed by Necula and lee. The code and its proof is delivered to the code consumer, before execution code consumer validate the code from untrusted source. Amount of effort should be done in code producer for establishing and formally proving the safety policy of the untrusted code. This can be particularly useful in ensuring memory safety. The formal verification can be helpful in proving the correctness of the systems with the source code and the internal memory. It also guarantees the safety policy which specified in security properties.

III. PROOF-CARRYING HARDWARE

We proposed Proof carrying hardware (PCH) is similar as Proof carrying code in this reconfiguration platform. Also to give complete security for embedded system. This paper distinguished as follows.

Ease of PCH: In spite of absolute guarantee of certain security features, reconfigurable hardware system provides only limited computational resources and increasing security and safety critical. In formal proof, elaborate computation can be done in some platforms which might not have the computational power. The few difference between the software and hardware are notable. Minimizing the reconfigurable time so that we can assess the new module as quick as possible which can be trusted. Proof carrying hardware offered the safety and security for this platform with applied limitation. Target platform only has to use little computational resources

Completeness of PCH: In this paper, we approach the understanding of system security techniques. These techniques focus on the one of the aspect of reconfigurable systems. Proof carrying hardware bases the safety policy verification but do not have one aspect of security. Reconfigurable host and incorporate any formal verification are the safety policies established by the consumer. The proof carrying hardware is arranged in spatial manner. Well-studied about the safety policy and security on the systems (i.e.) processor-based. The challenges are why the flexibility of the proof carrying hardware involved in any of the formal verification and what defined the software and hardware difference in the security regarding.

A. Project Description:

In FPGA there are two types of IC mainly used in VLSI one is FPGA and another is ASIC. In ASIC they are permanently circuitry which is can be changed or reconfigurable for it operating life time. But in FPGA they can be reconfigurable. While the area of the chip is still working it can be reconfigure a part of chip. So that if any Trojan is affected it can able to bypass it and operate the circuit as normal.

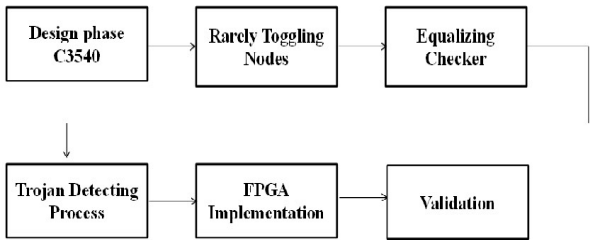

Fig.1 Proposed Methodology

In the FPGA kit they used the million numbers of transistors, gates etc. So we choosed the ALU of the C3540 it consists of the 1668 gates by using the program we should choose the number of rarely toggling nodes. The program can be written in Verilog or Python to find the rarely toggling nodes, the each number of gates should have the some of the toggling nodes are given below,

1. Zero toggling nodes,
2. Minimal togging nodes,
3. Rarely toggling nodes

The nodes are have some of zero toggling nodes like it should always 0's or 1's the Trojan cannot be enter into the nodes so they are neglect. The second node are minimal toggling node it can be toggle at minimum time in this state also cannot be enter the Trojan. The final node is the rarely toggling node in this state it can be easily added the Trojan at any nodes. In the ALU it consists of 1668 gates by using the Verilog or Python we can easily find the rarely toggling nodes.

By using the Xilinx software we can execute the program and find the result. The rarely toggling nodes are maximum 15 are chosen because the area threshold of 5% should be design overhead, the literature only of the 5% are allowed so they are choose the 15 nodes of the rarely toggling nodes.

B. Use of Simulation software:

For simulation we use Xilinx XC3S500E FT256 Spartan 3E FPGA ISE8.2i software written in Verilog code. This kit provides a easy way to develop and evaluate the platform for Spartan-3E FPGA design.

Xilinx XC3500E which consists of 500k gates and 10,467 logic cells. Totally 16nos of digital inputs using slide switches and 16nos of digital output using discrete LEDs, one reset switch.

190 I/O pins : 80 pins used for integrating peripheral like LED, Switches etc. Balance 110pins available for user. On board programmable oscillator from 3MHZ to 200MHz.

IV. RESULT and DISCUSSION

In the proof carrying code approach if the error in the nodes it can be identified and reconfigurable then the error values are bypassed and run the original values and implemented in the FPGA kit. For the proof code approach we can use the simple design using the AND and EX OR gates

Table 1. Design overhead

Circuit	Area Threshold (%)	Area Overhead (%)	No. Of Equivalent Checker Inserted	Power Overhead (%)	Timing Overhead (%)
C3540	-	10.17	15	8.6	0.00
C3540 + PCH	10	15	15	10	0.00

Table 2: Detection Coverage

CIRCUIT	NO. OF HARDWARE TROJAN INSERTED	NO. OF TEST VECTORS	COVERAGE WITHOUT EQUIVALENT CHECKER (%)	COVERAGE WITH EQUIVALENT CHECKER (%)	DETECTION COVERAGE (%)
C3540	5	20000	1%	96%	90%

After checking the process the error free functions are implement in kit and check the output and we can run the program safely and also avoiding the kit damages and also saving the times.

Conventional techniques of C3540 with and without trojan

V. IMPROVEMENT AS PER REVIEWER COMMENTS

We describe main concept of proof carrying hardware and details of reconfigurable hardware security approaches and workload was successfully shifted. Therefore we used to achieve results with the following goals.

Proof carrying hardware extended as our concept. Formal verification is done to meet the security challenge of active partial reconfiguration platforms. The definitions of this challenge may include verifiable hardware properties. The concept is nothing but only achieving its formats, properties formats for proving the security.

The implementation of the conceptual tool flows was the developed methods to apply in this concept. To achieve real time verification we designed the tool flows. The reconfigurable platform will be efficient and feasible. For open source and self-made tools there will be mapping of algorithms. The specified formats will be implemented to make the framework applicable to real verification tasks. Resulting of this uses open file format. Then we planned to apply the isolation primitives of modulation on one chip [6].

On the principle of PCH, the performance will be evaluated. The performance also evaluated for selected verification problems then it will be readily applicable. We also define FPGA architecture as reference of VPR tool [7].

VI. CONCLUSION

A secured hardware design with a proof that is hardware modules for reconfiguration are being embedded by a designer. In essence, the consumer is enabled to only run verified hardware modules without having to trust the producer or rely on a

secured transmission process or having to compute a formal proof of security features.

FPGA architecture of the proof carrying bitstream it includes a reconfigurable systems chip. In all the verification are checks by the equivalence checker with the help of the miter circuit after completing the process and if any Trojan are detected by using the proof carrying code approach it can be bypass and run the original outputs in the FPGA kit.

REFERENCES

[1] M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” IEEE Design Test of Computers, vol. 27, no. 1, pp. 10–25, 2010.

[2] Ranjani, R. S., & Devi, M. N. (2017). Malicious Hardware Detection and Design for Trust: an Analysis. Elektrotehniski Vestnik, 84(1/2), 7.

[3] Chakraborty, R. S., Pagliarini, S., Mathew, J., Ranjani, R. S., & Devi, M. N. (2017). A Flexible Online Checking Technique to Enhance Hardware Trojan Horse Detectability by Reliability Analysis. IEEE Transactions on Emerging Topics in Computing.

[4] Mal-Sarkar, Sanchita, et al. "Hardware trojan attacks in fpga devices: threat analysis and effective counter measures." Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI. ACM, 2014.

[5] G. Necula and P. Lee, “Proof-carrying code,” School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, Tech. Rep. CMU-CS-96-165, November 1996.

[6] T. Huffmire, B. Brotherton, G. Wang, T. Sherwood, R. Kastner, T. Levin, T. Nguyen, and C. Irvine, “Moats and Drawbridges: An Isolation Prim-itive for Reconfigurable Hardware Based Systems,” in Symposium on Security and Privacy. Oakland, CA: IEEE, May 2007, pp. 281–295.

[7] V. Betz and J. Rose, “VPR: A new packing, placement and routing tool for FPGA research,” in International Conference on Field Programmable Logic and Applications (FPL), vol. 1304. Londong, UK: Springer, 1997, pp. 213–222.

[8] Sree Ranjani R., and Dr. Nirmala Devi M., “Golden-chip free power metric based hardware trojan detection and diagnosis”, Far East Journal of Electronics and Communications, vol. 17, pp. 517-530, 2017.

[9] Sree Ranjani R., and Nirmala Devi M. “Enhanced Logical Locking for a Secured Hardware IP against Key-guessing Attacks,” 22nd International symposium on VLSI Design and Test (VDAT), June 28th -June 30th 2018.