Demonstration of Security Threats from Malicious FPGA Tools and Corresponding Countermeasures

Zhiming Zhang, Jaya Dofe, and Qiaoyan Yu
Dept. of Electrical and Computer Engineering
University of New Hampshire
Durham, NH 03824, USA
zz1017@wildcats.unh.edu

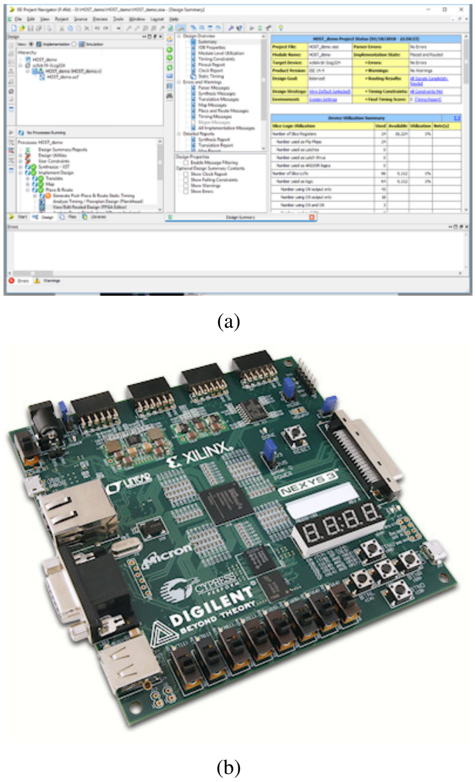
I. Description of the Research

Field Programmable Gate Arrays (FPGAs) enter a rapid growth era due to their attractive flexibility and CMOS-compatible fabrication process. Because of the high demand on the FPGA usage in data processing, industrial, automotive, consumer electronics, telecom, military and aerospace, FPGA market achieves a compound annual growth rate of 8.4% [1]. The increasing popularity of FPGAs also attracts attacker's attention because high improper benefits may be obtained once the FPGA-based system is manipulated. To protect FPGAs from being attacked, a great amount of works on FPGA security have been done [2]. Existing works primarily focus on reverse engineering the downloaded FPGA configuration, retrieving the authentication code or crypto key stored on the FPGA memory, and countermeasures for the security threats above. However, there are limited works addressing the security threats from malicious FPGA design software, which could harm the integrity of a design running on SRAM FPGAs [3]. In this demo, we introduce the potential security vulnerabilities of computer-aided design (CAD) tool. This group of attacks are implemented on the CAD tool which is used to generate bitstream so that the FPGA behavior can be modified without even touching the top level Verilog design as well as the hardware on FPGA board [4].

Proposed countermeasures can give attackers multiple levels of unpredictability when inserting hardware Trojans such that different levels of attacks, which are based on how well the design is known, can be thwarted. Blindly inserting a Trojan may not have impact on the design at all [5].

II. Features of Research Targeted in Demo

Xilinx ISE 14.1 FPGA design suite will be used as the software environment in which designs are created and mapped into FPGA board. Xilinx Spartan-6 FPGA is the hardware which the targeted attacks and the proposed countermeasures play effect to. To complete all the experiments included in the demo, a normal monitor which supports VGA signal input and a computer with Xilinx ISE 14.1 installed are also needed. Bitstream generated from the FPGA design suite will be downloaded into the Spartan-6 FPGA through USB cable which links the computer and the FPGA. The FPGA board will also be connected to the monitor through VGA cable. The whole process of the attack can be watched through the computer display and the FPGA on-board LEDs. The example attack on VGA signal generation can be seen from the monitor. The experimental setup on software and hardware is shown in Fig. 1.



ISE Project Navigator (P.49d) - D:\HOST_demo\HOST_demo\HOST_demo.xise - [Design Summary]

File Edit View Project Source Process Tools Window Layout Help

Hierarchy
- HOST_demo
- xc6slx16-3csg324
- HOST_demo (HOST_demo.v)
- HOST_demo.ucf

No Processes Running

Processes: HOST_demo
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Translate
- Map
- Place & Route
- Generate Post-Place & Route Static Timing
- Analyze Timing / Floorplan Design (PlanAhead)
- View/Edit Routed Design (FPGA Editor)

Design Overview
- Summary
- IOB Properties
- Module Level Utilization
- Timing Constraints
- Pinout Report
- Clock Report
- Static Timing

Errors and Warnings
- Parser Messages
- Synthesis Messages
- Translation Messages
- Map Messages
- Place and Route Messages
- Timing Messages
- Bitgen Messages
- All Implementation Messages

Detailed Reports
- Synthesis Report
- Translation Report

Design Properties
- ☐ Enable Message Filtering

Optional Design Summary Contents
- ☐ Show Clock Report
- ☐ Show Failing Constraints
- ☐ Show Warnings
- ☐ Show Errors

HOST_demo Project Status			
Project File:	HOST_demo.xise	Parser Errors:	No Errors
Module Name:	HOST_demo	Implementation State:	Placed and Routed
Target Device:	xc6slx16-3csg324	• Errors:	No Errors
Product Version:	ISE 14.4	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constraints Met
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	24	18,224	1%	
Number used as Flip Flops	24			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	[illegible]	9,112	1%	
Number used as logic	[illegible]	9,112	1%	
Number using O6 output only	[illegible]			
Number using O5 output only	[illegible]			
Number using O5 and O6	[illegible]			

(a)

(b)

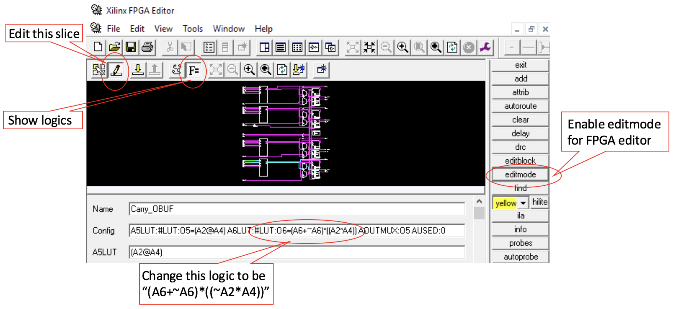
Fig. 1. Experimental setup on (a) software and (b) hardware.

III. What Will the Audience See?

In this hardware demonstration, we will show the impact of the targeted attacks on the system final output and how the proposed countermeasure mitigates the Trojan insertion attacks.

A. Attack Procedure

The targeted CAD tool based attacks will be implemented manually through the Xilinx FPGA design suite built-in tool



Name: Carry_OBUF

Config: A5LUT:#LUT:O5=(A2@A4) A6LUT:#LUT:O6=(A6+~A6)*((A2*A4)) AOUTMUX:O5 AUSED:0

A5LUT: (A2@A4)

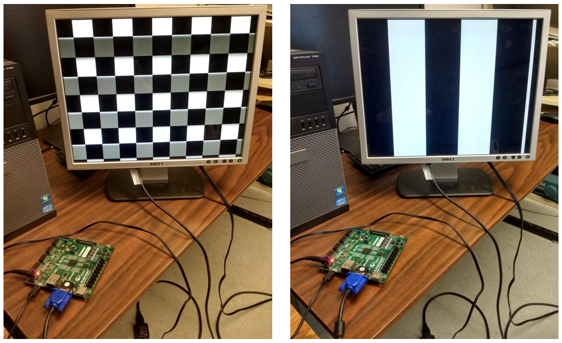
Edit this slice

Show logics

Enable editmode for FPGA editor

Change this logic to be "(A6+~A6)*((~A2*A4))"

Fig. 2. Attack performed through the FPGA editor tool available in the Xilinx ISE 14.1 design suite.



(a) (b)

Fig. 3. VGA signal display (a) before and (b) after modification.

FPGA Editor. Figure 2 provides the main screenshot for the attack procedure. From this demonstration, audience will see how the logic function of a full adder design is changed without disrupting the top level Verilog design file. We will use the FPGA LEDs to indicate the success of Trojan attacks on the full adder.

B. A Practical Example of the Attack

An attack on VGA signal generation will be shown next as a practical example. In this section, audience will see the original signal of ”chessboard” displayed in the monitor being manipulated into ”color bars” after running the attack to FPGA Editor.

C. Proposed Countermeasures

In the final section of the demo, we will show the proposed countermeasures. Our method includes three defense lines:

- *D-1*: We specify the slice position on the FPGA die for the selected LUTs instead of using default settings. In this way, the design placement on FPGA can be changed significantly so that attacker will have hard time to decide where to place hardware Trojans.
- *D-2*: We duplicate the design by n copies and only one of the replicas will be active at a time. The replica selection is shown in Figrue 4. Without knowing which copy is active, blindly inserting a Trojan may not impact the design at all.
- *D-3*: We further divide the design into m submodules and each submodule is duplicated by n times to realize a hot-swappable submodule assembling technique, which is shown in Figure 5. The total number of design configurations is m^n such that it is more difficult for attacker to identify which copy to attack.

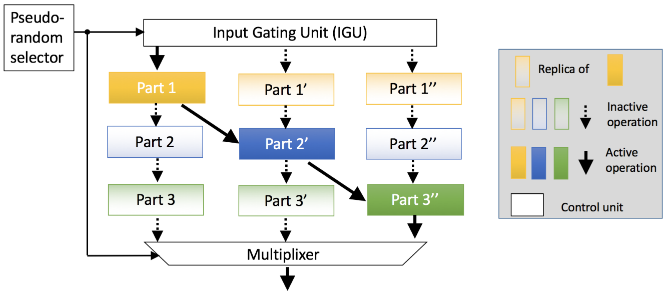


Fig. 4. Replica selection provided by the proposed method.

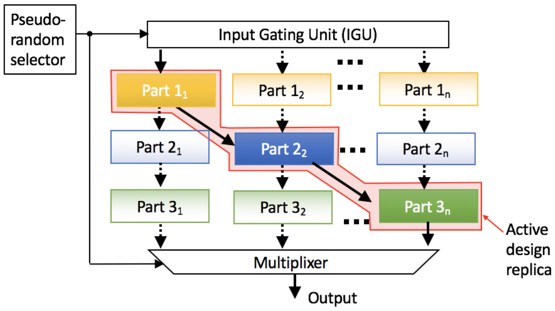


Fig. 5. Hot-swappable assembling technique.

In this section, designs with the protection of our countermeasures will be run instead of baselines. Audience will see the same attacks failing modifying the normal operation because of the proposed protection.

IV. Conclusion

The imbalance relationship between FPGA hardware/software providers and FPGA users challenges the assurance of secure design on FPGAs. Different than the existing literature primarily focusing on reverse engineering the downloaded FPGA configuration, retrieving the authentication code or crypto key stored on the embedded memory in FPGAs, this demo shows the practical FPGA attacks due to the untrusted FPGA software. We also demonstrate the possible countermeasures against the FPGA attacks.

References

[1] “FPGA Market size set to exceed USD 9.98 Billion by 2022, with over 8.4from 2015 to 2022: Global Market Insights Inc.” https://goo.gl/uEmByo.

[2] S. Trimberger and J. Moore, “FPGA security: From features to capabilities to trusted systems,” *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1-4, June 2014.

[3] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, “Hardware Trojan Insertion by Direct Modification of FPGA Configuration Bitstream,” *IEEE Design Test*, Vol. 30, No. 2, pp. 45-54, April 2013.

[4] Z. Zhang, *et al.*, “Securing FPGA-based Obsolete Component Replacement for Legacy Systems,” to appear in Proc. ISQED’18.

[5] Z. Zhang, *et al.*, “FPGA-Oriented Moving Target Defense against Security Threats from Malicious FPGA Tools,” to appear in Proc. HOST’18.