



Article

Creation and Detection of Hardware Trojans Using Non-Invasive Off-The-Shelf Technologies

Catherine Rooney ¹, Amar Seeam ²  and Xavier Bellekens ^{1,*} ¹ Division of Cyber Security, Abertay University, Dundee DD1 1HG, UK; c.rooney@abertay.ac.uk² Department of Computer Science, Middlesex University, Mauritius Campus, Flic en Flac, Mauritius; a.seeam@mdx.ac.mu

* Correspondence: x.bellekens@abertay.ac.uk; Tel.: +44-(0)-1382-30-8482

Received: 6 June 2018; Accepted: 20 July 2018; Published: 22 July 2018



Abstract: As a result of the globalisation of the semiconductor design and fabrication processes, integrated circuits are becoming increasingly vulnerable to malicious attacks. The most concerning threats are hardware trojans. A hardware trojan is a malicious inclusion or alteration to the existing design of an integrated circuit, with the possible effects ranging from leakage of sensitive information to the complete destruction of the integrated circuit itself. While the majority of existing detection schemes focus on test-time, they all require expensive methodologies to detect hardware trojans. Off-the-shelf approaches have often been overlooked due to limited hardware resources and detection accuracy. With the advances in technologies and the democratisation of open-source hardware, however, these tools enable the detection of hardware trojans at reduced costs during or after production. In this manuscript, a hardware trojan is created and emulated on a consumer FPGA board. The experiments to detect the trojan in a dormant and active state are made using off-the-shelf technologies taking advantage of different techniques such as Power Analysis Reports, Side Channel Analysis and Thermal Measurements. Furthermore, multiple attempts to detect the trojan are demonstrated and benchmarked. Our simulations result in a state-of-the-art methodology to accurately detect the trojan in both dormant and active states using off-the-shelf hardware.

Keywords: hardware trojan taxonomy; thermal imaging; side channel analysis; infrared; FPGA

1. Introduction

A hardware trojan can be described as a malicious alteration or inclusion to an integrated circuit (IC) that will either alter its intended function or cause it to perform an additional malicious function. These malicious inclusions or alterations are generally programmed to activate only under a specific set of circumstances created by an attacker and are extremely hard to detect when in their dormant state [1]. As technology advances, so does the demand for IC boards leaving many technology companies without the resources to produce secure enough ICs to meet current demands. This has pushed companies into the ‘fabless’ trend prevalent in today’s semi-conductor industry, where companies are no longer attempting to produce the goods in their own factories, but instead are outsourcing the process to cheaper factories abroad [2,3].

This growth brings with it a significant rise in the level of threat posed by hardware trojans, a threat that directly affects all companies concerned with products that utilise ICs. This encompasses many different industries, including the military and telecommunications companies, and can potentially affect billions of devices from mobile phones and computers to military grade aviation and detection devices, particularly at a time when wireless devices are being introduced as links in critical infrastructure, compounding trust and security issues even further [4,5]. It is also a direct threat to the

already vulnerable Internet of Things, meaning that wireless-enabled household devices also become potential targets [6].

The problem is such that even previously ‘reputable’ factories are vulnerable to attacks, since all that is required is one employee to alter the existing code to include a trojan [7,8]. As most IC designs are extremely large and contain a huge amount of hardware description, these inclusions are difficult to detect and the sheer size of the code can require many people having access to the code at production level.

Regarding military grade products utilising ICs, the problem of hardware trojans is critical with the threat level of the trojan being such that it could potentially be catastrophic. Malicious inclusions of code could cause life saving equipment to fail, missiles to lose control, and cryptography keys to be leaked. While incidents of hardware trojans, are not openly discussed there have been a few noted. In 2007, it was assumed that a backdoor built into a Syrian radar system was responsible for the system’s failure [9]. There are also reports of trojans being used by the USSR to intercept American communications during the cold war [3].

The problem is aggravated further still when considered in relation to the growth in production of counterfeit goods. Such goods may be produced in less than reputable factories, so the inclusion of malicious code in the production process is far from unrealistic [10]. As counterfeit goods are not generally sold through trustworthy channels, it is impossible to recall products found to be unsafe or indeed to produce updated firmware to deal with emerging threats. This can expose consumers to a plethora of malicious attacks by hackers. For example, a trojan leaking cryptography keys in counterfeit IoT devices could potentially give hackers access to a network of devices that can be utilised in ‘Mirai’ like attacks and cannot be recalled or patched [11].

In this paper, a hardware trojan is created and emulated on a consumer FPGA board. The experiments to detect the trojan in a dormant and active state are made using off-the-shelf technologies which rely on thermal imaging, power monitoring, and side-channel analysis. Furthermore, three attempts to detect the trojan are demonstrated and benchmarked. Finally, a state-of-the-art methodology is presented which allows accurate detection of the trojan in both dormant and active states.

Other researchers have attempted to detect trojans using similar techniques to the ones presented in this paper, using thermal imaging [12–14] and side channel attack and power analysis [15,16] However, to the best knowledge of the authors this is the first manuscript successfully proposing a practical approach using off-the-shelf hardware for the detection of hardware trojans.

The rest of this paper is organised as follows, Section 2 provides an overview of the different hardware threats, the industries affected and a detailed hardware taxonomy, while Section 3 provides an overview on the testing methodology, the design of the trojan, its functions and details of its implementation. In Section 4 an overview of the different techniques used to detect the hardware trojan are provided. Section 5 describes the results obtained using the different off-the-shelf devices. The results and techniques are further discussed in Section 6 and the paper finishes in Section 7 with the conclusion.

2. Background

Whilst the ongoing war between hackers and software developers has been raging since the 1980s, the underlying hardware being utilised was always considered to be secure [17]. Over the last decade, however, the technological progress has been such that the demand for IC boards has grown to unprecedented levels. To cope with such demand, many technology companies have evolved into the so-called ‘fabless’ companies, meaning that their designs are outsourced to cheaper, usually foreign, foundries for production [18]. As the number of companies involved in the production chain has grown, so have concerns, and the problem of hardware trojans has escalated.

As the full potential of the threat posed by the hardware trojan has been realised and acknowledged by the electronics industries many different conspiracy theories have emerged.

Unfortunately, those theories are not in nature groundless or out-with the realms of the possible. In fact several of them have already been instantiated. One such rumour of ‘kill switches’ being hidden in commercial processors was confirmed by an anonymous U.S. defence contractor who indicated the culprit to be a ‘European Chip Maker’ [17]. The potential consequence of the existence of such a switch could be catastrophic. Indeed, as previously highlighted, this particular hardware trojan was blamed for the failure of a Syrian radar to detect an incoming air strike [19].

2.1. *The Threat of the Hardware Trojan*

2.1.1. At Design Level

The complexity and cost of the design of ICs has grown exponentially over the last decade as the semiconductor industry has scaled to sub-micron levels. A typical IC board will go through a rigorous process consisting of several stages.

Firstly, the specifications must be translated into a behavioural description, usually in a hardware description language such as Verilog or VHDL. Once this has been completed, the next phase is to perform synthesis to transform the behavioural description into a design implementation using logic gates such as a netlist. Once the synthesis has been completed, the netlist is implemented as a layout design and the digital files are passed to the foundry for fabrication [17].

As well as outsourcing the production of ICs, many companies are also purchasing third party intellectual property (IP) cores, and utilising third party Electronic Design Automation (EDA) tools. Each use of a third-party software presents a new opportunity for attacks such as hardware trojan insertion, IP piracy, IC tampering, and IC cloning. Although these attacks are all of importance, the hardware trojan is by far the most dangerous attack, and, as such, has garnered much attention [17].

2.1.2. At Foundry Level

As semiconductor technology has advanced, the cost of owning foundry has increased dramatically. In 2015, the cost was estimated to be in the region of 5 billion USD [20]. As a direct result of this, many companies can no longer afford to fund the production process from start to finish, and are outsourcing their production to cheaper foreign foundries [17].

Whilst undesirable modifications to ICs should ideally be detectable by pre-silicon verification and simulation, this would require a specific model of the entire IC design and this is not always readily available particularly where third party IP cores or EDA tools have been used. In addition, large multi module designs are rarely compliant with exhaustive verification [21].

Post silicone approaches to design verification include destructive de-packaging and reverse engineering of the IC. However, current techniques do not allow destructive verification of ICs to be scalable [22]. It is also possible for an attacker to infect only a portion of the produced ICs, making these tests futile [23].

Most post silicone logical testing techniques are also unsuitable for detecting hardware trojans. This is attributed to the stealthy nature of the hardware trojan and to the large numbers of differing taxonomies that can be employed by the attackers. Most hardware trojans are programmed to activate under a specific set of conditions, and a skilled attacker would ensure that these conditions were undetectable by the testing routine. This is particularly true of trojans targeting sequential finite state machines [24].

2.2. *Industries Affected*

2.2.1. Military

Hardware trojans are a huge threat to many industries. However, security conscious industries, such as the military, are in a particularly high risk bracket and defence departments are very aware of this. The U.S. Department of Defense (DoD) has created a “Trusted Foundry Program” to ensure its

military equipment remains free of hardware trojans by using only accredited foundries. This means that only American foundries which are located on the American soil and which underwent the strictest vetting process are allowed to work on the chips for the U.S. DoD. In addition to vetting the foundries, close attention is being paid to the other links in the design and supply chain [9].

While this approach may seem effective, it has its limitations. The majority of western foundries are woefully behind their foreign counterparts when it comes to the level of technology they can provide. This seriously limits access to more advanced chips which are required for modern avionics and weapons systems [9].

2.2.2. Financial Infrastructures

In 2008, an experiment was carried out by the University of Illinois in which researchers designed and inserted a small backdoor circuit that gave access to privileged areas of the chip's memory [9]. This trojan could then be used to change process identifiers allowing attackers to access all data contained on the chip's memory. It is easy to see why this could be catastrophic in settings such as critical infrastructures. Trojans such as the one described are usually small and are nigh impossible to detect.

2.2.3. Consumer Industries

Security industries are not the only potential targets of a hardware trojan attack. There exists the possibility of utilising a hardware trojan by rival firms in industrial sabotage. The potential damage that could be caused by such an attack could be enough to disable even global corporations, particularly in industries such as telecommunications.

Aside from industrial sabotage, the potential threat to consumer privacy is also of major importance. Devices such as smartphones and tablets could be targeted by trojans designed to leak private encryption keys or private information [9].

2.3. Hardware Trojan Taxonomy

Although the terms and definitions used to classify different hardware trojans can vary between different authors, the general taxonomy is universally agreed to consist of the physical representation, the behavioural phase or trigger, and the action phase in which the trojan will execute its payload.

2.3.1. Physical Representation

When designing malicious circuitry, there are several characteristics that must be considered. The first of these characteristics is the 'type' of the hardware trojan, which can be defined as functional or parametric.

A trojan is categorised as functional when the attacker adds components such as logic gates to the original design. Accordingly, the deletion of components to cause a malicious function would also place it in this category [25].

If the attacker creates the trojan through the modification of the existing code, then it will be classified as a parametric. Typically, this can be achieved by thinning wires or weakening transistors and flip flops. This type of trojan is notoriously hard to detect as the alteration can be minuscule.

The next physical characteristic the attacker would have to consider would be the size of the hardware trojan. In this context, the size refers to the physical extension of the hardware trojan or the number of components it consists of. In case of a large trojan consisting of many components, an attacker can distribute these across the IC, placing components where they are necessary to execute their payload in accordance with the functions of the hardware trojan. This is known as loose distribution [25].

In contrast, a smaller hardware trojan consisting of only a few components allows for the components to be placed together as they will occupy only a small part of the layout of the IC. This is known as tight distribution.

On rare occasions, a determined attacker could regenerate the layout to encompass the hardware trojan, moving the components of the IC to accommodate the components of the hardware trojan. This is referred to as a structural alteration [25].

2.3.2. Activation Characteristics

Typically, a hardware trojan will be condition-based, meaning that its activation will be dependent on a trigger defined by the attacker. The trigger itself will generally consist of either a predefined input pattern, or specific internal logic state, or counter value, and can be triggered both internally and externally.

An externally triggered hardware trojan will usually consist of malicious logic within the IC that utilises an external sensor such as a radio antenna. The attacker will then communicate via the compromised component enabling them to trigger the antenna. It is easy to see why this can be extremely dangerous when it comes to security conscious industries such as the military. It is not out-with the realms of the believable to postulate that an attacker could feasibly re-route or switch off a missile via a radio signal as suggested in [22]. Conversely, an internally triggered hardware trojan will look within the circuitry for the set of conditions that will cause it to activate. A typical example of this would be a countdown logic.

In contrast to the condition-based trojan that will only activate when its trigger conditions are met, the “always-on” trojan is active from the moment of insertion, and relies on internal signals. This type of hardware trojan is generally split into two categories; combinational and sequential. A combinational trojan will activate upon detection of a specific set of circumstances within the internal signals of the IC. Sequential trojans will also monitor the internal signals of the IC. However, instead of looking for a specific condition, they activate when a specific sequence of events occurs [26].

2.3.3. Action Characteristics

The action characteristics of a hardware trojan refer to the effect the trojan will have on the execution of its payload. Hardware trojans will typically fall into one of two categories: implicit or explicit [27]. Implicit trojans will not change the board’s circuitry of the IC; instead, they will perform their malicious function in tandem with the intended function of the board. This makes these trojans easier to detect as they tend to cause small path delays on activation and consume more power whilst active.

In contrast, an explicit trojan will change the function of the board’s circuitry on activation. This can come in the form of a signal alteration or even leaking of information via predefined board pins. These trojans tend to cause distinct path delays as well as large changes in circuit’s capacity [27].

2.4. Hardware Trojan Detection

Detecting a hardware trojan requires overcoming numerous challenges. Namely:

1. Handling large architectures.
2. Being non-destructive to the IC.
3. Being cost effective.
4. Ability to Detect trojans of all sizes.
5. Authenticating chips in as small a time frame as possible.
6. Dealing with variations in manufacturing processes.
7. Detecting all trojan classifications.
8. Detecting trojans in a reasonable time frame.

To the best of the authors’ knowledge, there is no single method capable of detecting all types of hardware trojans, nor overcoming all the challenges described here-above. Over the

years, several methods have been developed to detect different types of trojans. These methods are described here-after.

2.4.1. Physical Inspection

One of the most obvious method of detection is physical inspection of the board itself. This method is sometimes classified as a failure analysis based technique. Those techniques usually comprise two steps: (1) cutting and lifting the molding coat to expose the circuitry; and (2) performing various scans [25,28].

2.4.2. Functional Testing

Often referred to as Automatic Test Pattern Generation (ATPG) this technique is more commonly used to locate manufacturing faults; it has been shown to be effective in detecting hardware trojans. ATPG involves inputs of ports are stimulated and then the output ports are monitored for variations that may indicate a hardware trojan has been activated. Functional testing techniques can also be useful when attempting to determine the trigger patterns of conditional trojans [25,27].

2.4.3. Built-In-Self-Test Techniques

Built-In-Self-Test (BIST) techniques are commonly used to detect manufacturing faults and are present in many chips. If unknown or malicious logic is detected during these tests a bad checksum result is given, although designed to detect manufacturing faults on some occasions these tests can detect hardware trojans [25].

2.4.4. Side Channel Analysis

Side channel analysis techniques are some of the most commonly used procedures in hardware trojan detection. These techniques generally measure signals such as power and path delay, looking for fluctuations potentially caused by trojans. Side channel analysis can have a high success rate as even in a dormant state the trojans trigger signal will cause some current leakage [25].

3. Methodology

In order to carry out the investigation our trojan was designed and loaded onto an Basys 3 Artix 7 FPGA board. Three different detection techniques are demonstrated, the first utilises power analysis techniques as well as side channel analysis, allowing security investigators to measure both the power variance, traces and current leakage, followed by a concentrated heat measurements using an infrared thermometer, and finally a thermal camera test is carried out. The three experiments are carried out using off-the-shelf hardware and are applied to both the trojan-free and trojan-inserted designs. Attempts are then made to detect the trojan in its dormant form. While in in their dormant form trojans do not perform any malicious actions, however, wait to be activated, through an activation signal, this can be done through the push of a button, or through a specific set of instructions. It is however important to be able to detect trojans in their dormant form, before they activate and perform malicious actions. In the last experiment, the thermal camera is also used to measure the impact of the trojan in its active form. Table 1 provides information on the type of devices used to carry out the different experiments.

Table 1. Description of Off-The-Shelf Hardware for Hardware Trojan Detection.

Type	Device	Price
Oscilloscope	Infinium series 1 GHz	£4000
Oscilloscope	Digilent OpenScope	£60.00
Infrared Thermometer Gun	Powefix	£17.99
Heat Camera	Flir C2 Camera	£550.00

3.1. Design

Whilst there is much literature available on the topic of hardware trojans and many papers aimed at the various taxonomies of hardware trojans, there are few practical examples to be found. This paper aims at providing thorough details on the creation of a tailored hardware trojan. The factors considered in the experiments and the resultant decisions are shown in the list below;

Main Function The algorithm as a main function running continuously. This function serves by providing information on the board, and executing tasks in a round robin fashion, these tasks run in a loop, this ensures steady power readings.

Physical Representation The trojan designed for the experiments is in the form of a functional trojan as opposed to a parametric trojan. Functional trojans are categorised as a malicious inclusion of components and code as opposed to an alteration to existing code.

Physical Size The trojan designed for the experiments is required to be small and consists of no more than three components, this allows a tight distribution on the board. The creation of a complex design is deemed to be out of the scope of this paper.

Characteristics The trojan is activated by an external trigger to preserve the integrity of the testing results. This provides assurances for the different test results, and allows the monitoring of the free, dormant, and active states. This test falls within the parameters of a conditional trojan.

Malicious Function The trojan is required to have a malicious impact on the board, while more advanced functions such as the leaking of cryptography keys are within the scope of this investigation, simpler functions are privileged to demonstrate the process. Two distinct functions are chosen, the first one causes the board to overheat, while the second one performs an unwanted action on the board.

The design was carried out using the Vivado WebPACK Design Suite, this allowed for high level design, synthesis and implementation.

Currently the use of third party Intellectual Property (IP) Cores to construct large and complex designs is standard practice industry wide. An IP core is a block of logic or code that can be utilised in the design of an ASIC or FPGA, most common components can be found the form of an IP Core and this method of design drastically reduces the amount of time the designer spends writing code. It is equally likely that an attacker inserting malicious code would have had the skill to create and insert their own hand crafted code. A block diagram is shown in Figure 1, the block diagram represent the counter without the trojan inserted.

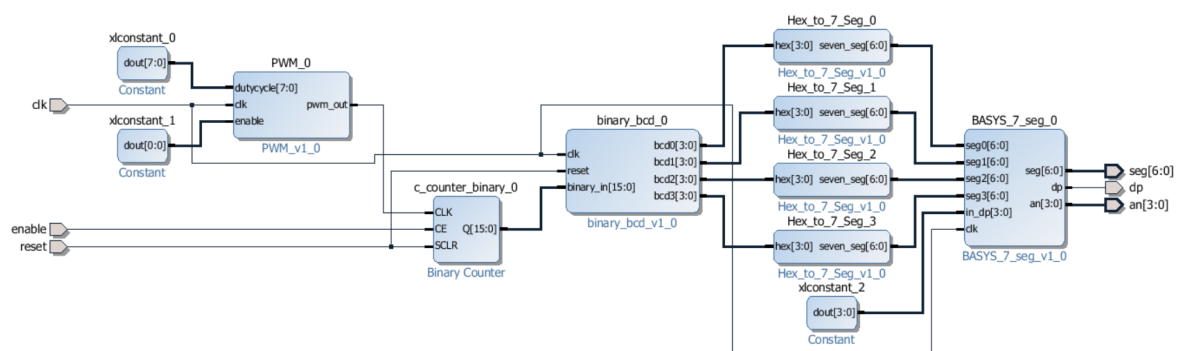


Figure 1. Configured IP Cores Block Diagram.

3.2. Trojan-Free Design

The main function of the trojan-free design was decided to be a counter program, counting from 1 to 9999 on a loop and output to the four seven segment displays present on the Basys 3 board.

The components used in this design operate at a steady rate, this ensures steady power tests taken whilst performing side channel analysis on the clean board, thereby guaranteeing that any fluctuations detected on the infected board are indeed caused by the presence of the trojan. The trojan-free design requirements for the counter can be found below;

- The refresh rate for the 7 segment displays is set to 5000 Hz
- The design utilises all four 7 segment displays
- The 7 segment displays output a count that beginning at 0 and incrementing up to 9999
- Upon reaching maxcount of 9999 the displays resets to 0
- The counter is enabled via a switch
- The counter can be reset via a pushbutton
- The counter will count at a rate of 5 Hz

After the final configuration, it was a matter of creating a top level VHDL wrapper to contain the inputs and outputs of the block design. The wrapper file maps all the I/O ports of the physical board with the I/O described in the constrain file. Finally, the constraints for the project needed to be added to successfully synthesise and implement the design. Constraints files are used to map the input and output ports to specific pins on the FPGA board and can be easily configured by uncommenting a master file and editing the port names to reflect those of your design. Figure 2 shows the trojan-free elaborated design schematic. The schematic is based the RTL file. The schematic identifies pieces of code representing hardware structure. The code further converted (Synthesis) into “technology cells” abstracting elements such as multiplexer, adder, etc. . . .

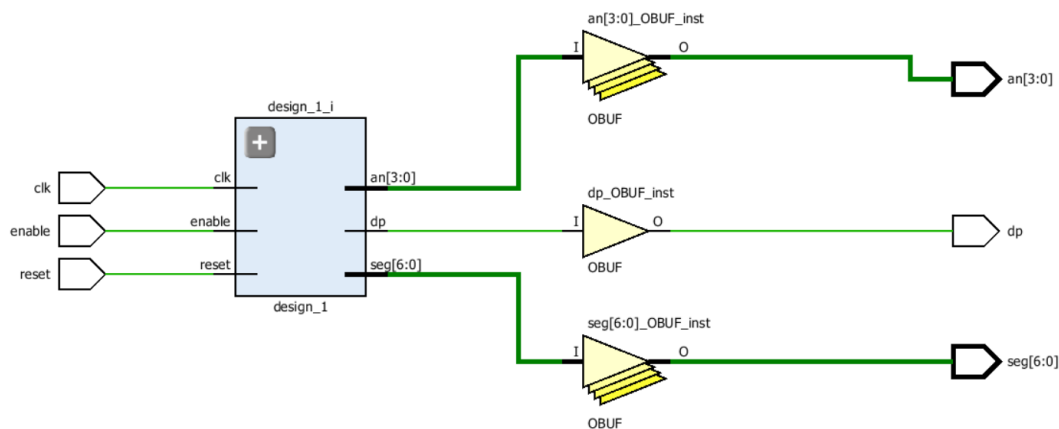


Figure 2. Trojan-free Elaborated Design Schematic Provided by Vivado.

3.3. Unsigned Multiplier

As previously discussed the trojan code used in this project was handcrafted to mimic the techniques that would be used by an attacker, although it is possible that malicious code could be contained within a third-party IP core.

The design to be used for the trojan is in the form of an unsigned multiplier that operates on a combination of buttons and switches and output onto the LED. The required inputs of both switches and a button can be said to represent the usually hard to guess trigger. These inputs are often used in conditional trojans.

The trojan was designed with these specifications in mind.

- There shall be 15 input switches and 1 input button
- There shall be a start button and a reset button
- The multiplier will output in binary via LEDs
- The multiplier shall utilise two finite state machines
- The multiplier should have a malicious effect on the board

FPGA boards do not tend to contain many dedicated multipliers with the Basys 3 containing only four. This means that when creating a design that requires a multiplier most designers will compensate for this by creating their own. As such, the presence of this code would not be likely cause suspicion. The junction temperature of the board is the highest temperature that a board can operate safely at, it is calculated using Equation (1).

$$T_j = T_a + (\theta_{J_a} \times P_d) \quad (1)$$

Let T_a be the ambient temperature expressed in °C, let θ_{J_a} be the Junction-to-ambient thermal resistance, let P_d be the core power, finally let T_j be the junction temperature. The relationship between the thermal parameters of the board can be expressed as show in Equation (2), let θ_{J_a} be the junction-to-ambient thermal it defines the difference between junctions temperatures and the ambient temperature when the device dissipates 1Watt of power, hence is expressed in °C/W resistance, let θ_{J_c} be the junction-to-case thermal resistance and is expressed in °C/W and θ_{C_a} be the case-to-ambient thermal resistance and is expressed in °C/W.

$$\theta_{J_a} = \theta_{J_c} + \theta_{C_a} \quad (2)$$

The design for the trojan also utilises two finite state machines, This is necessary for the multiplier as it requires several inputs. This can be seen in Figure 3 (Highlighted in Red). For the purposes of this trojan a Mealy machine was deemed to be most appropriate. Generally Mealy machines have fewer states than Moore machines, the output changes at the clock edges in a Mealy machine and react faster to inputs. This fits well with the requirements of the trojan as it must have several inputs.

Once the trojan designed the next step is to create a working constraints file as this is necessary to test the functionality of the trojan.

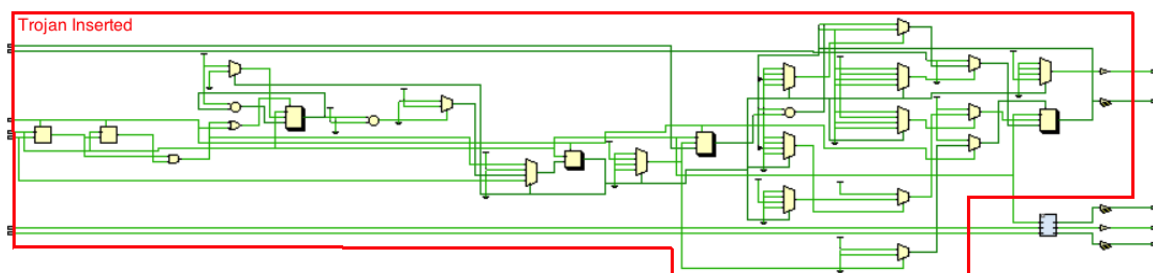


Figure 3. Trojan-inserted Elaborated Design Schematic.

3.4. The Trojan

Having designed and built both the counter and multiplier circuits the next step is to integrate the code from the multiplier into the top level VHDL wrapper created for the counter circuit. The counter signal is asynchronous. The counter activates on the first flip-flop. The subsequent flip-flops are clocked by using the output of the previous flip-flop.

The additional ports required by the multiplier was to be added to the original design for the counter. Whilst this is straight forward for most of the ports, the clk port in the multiplier was deleted as there was already a clock in place and they had been set to the same value, meaning that the “trojan” was now utilising the clk port of the counter.

The second amendment that had to be made was the reset port for the multiplier, as again the counter already had a reset port. While it may have been possible to allow both to use the same button, the ability to reset the trojan alone is important to the integrity of the testing and results. To circumvent this the reset port for the multiplier was renamed as *reset₂* and would be mapped to a different push button. As the multiplier is unsigned it was also necessary to include the *IEEE.numeric_std.alllibrary* and is working synchronously.

As the code to be inserted was still referencing the reset port it was necessary to go through all the processes and change the reset port to the newly created *reset₂* port. The next step was to insert the code for the processes that would be utilised by the trojan, this code was again inserted into the architecture body of the program.

Finally, the constraints file was updated to include the ports utilised by the 'trojan'. At this point it became evident that there were not enough input switches to accommodate both designs, the multiplier requires all sixteen switches as it takes inputs in binary form and the counter also requires an input switch.

As the switch was essential to the counters enable signal, a push button would only increment on each button press, the multipliers *input₁* was mapped to a push button. This made the input pattern required by the trojan a combination of buttons and switches followed by a specific button press.

3.5. Elaborating the Designs

Although the code has now been created for both the trojan-free and trojan-inserted boards there are several more phases to the process of creating the final bitstream file that will be used to program the board. The first of these stages is that of elaborating the design. The elaboration process essentially takes the code provided and changes it from RTL into a variety of visual representations.

3.6. Synthesising the Designs

Synthesis refers to the process of transforming the design from an RTL design into a gate level representation. Synthesis is run after the relevant design and constraint files have been added to the project and will only be successful if the files added contain no syntax or constraints errors. Following a successful synthesis, it is possible to access the synthesised design and view the schematics of the design along with various other representations.

3.7. Implementing the Designs

Having successfully synthesised the design the next step was to run it through the implementation process. Implementation is essentially equivalent to routing and placing the design on the board. The implementation process places the netlist onto the board in a virtual environment and connects them in accordance with the constraints file. If it is possible to safely place the design on the board and it will then be possible to generate a bitstream file which can then be placed on the FPGA board.

3.8. Generating Bitstream File and Programming the Board

Once the design has passed the required phases it is possible to generate a bitstream file, this is the file that used to program the FPGA board.

Before programming the board, it was necessary to ensure that the board was correctly configured. The board can be programmed to accept a file or to perform a Built In Self Test (BIST), by default it is set to BIST to enable programming mode on the board.

4. Malware Detection Methodologies

The scenario presented in this manuscript for the trojan detection assumes that the board has been infected during production, hence the full design of the board is known. It is also assumed that we have a free trojan board at our disposal. This could be a prototype.

4.1. Side Channel Analysis

Side channel analysis techniques are employed in order to detect fluctuations in variables such as power and path delay, for this investigation it was decided to test power levels using an oscilloscope.

Two oscilloscopes were used for the investigation, the first one was an Infinium series 1 GHz 4 GSa/s oscilloscope from Agilent. While the second one was an OpenScope 2 scope channel with 12 bits at 2 MHz as shown in Table 1.

By analysing the schematics of the Artix 7 board, we identified the power in/out pins to the FPGA chip. As these pins are likely to be utilised by any design they were most likely to exhibit power fluctuations caused by the trojan. In a real world scenario, the authors of a circuit to be able to locate and test multiple pins and capacitors against a golden design in order to detect a trojan.

In order to detect power fluctuations, we set a trigger on the oscilloscope. The trigger allows the user to set a limit on readings. If the voltage measured goes above the set trigger level then the oscilloscope will stop the measurement. The trigger can be dragged and to fit measurements for pins on the trojan-free board, if the trojan boards readings exceed those of the trojan-free board, we could assume that the board was infected. This experiment was realised with the two oscilloscope, over fifteen times, to ensure accurate readings.

4.2. Temperature Readings

In order to obtain temperature readings during the testing phase of the investigation an infrared thermometer and a heat camera were used on the boards Artix 7 FPGA chip. With the trojan active, it is assumed that during normal operation, the trojan-free board will consume less power, hence the chip will be cooler. Moreover, even in its dormant form fluctuations caused by the presence of the trojan may be detectable.

The Infrared Thermometer Gun used is described in Table 1. The thermometer is an infrared thermometer which utilises sensors to acquire and measure infrared radiation from the surface it is aimed at, and from this radiation it determines temperatures. The device can detect and record temperatures ranging from $-50\text{ }^{\circ}\text{C}$ to $+380\text{ }^{\circ}\text{C}$ and as such it fits within the expected temperature range of the FPGA chip. The FLIR C2 heat camera used is described in Table 1. The Flir C2 Camera is a standard off-the-shelf heat camera, with a sensitivity of $0.10\text{ }^{\circ}\text{C}$ and works within the $-10\text{ }^{\circ}\text{C}$ to $+150\text{ }^{\circ}\text{C}$ ($14\text{ }^{\circ}\text{F}$ to $302\text{ }^{\circ}\text{F}$) The IR sensors provides 80×60 (4800 measurement pixels) and works within as spectral range of $7.5\text{--}14\text{ }\mu\text{m}$.

4.3. Testing Process

There were several factors considered for the testing process, firstly after each test the board is left to cool down for fifteen minutes before being re-tested. This ensure that the results are not skewed by remnant heat. Misleading results could potentially affect the camera and heat gun readings, hence the cool down period. Secondly, ten readings where taken on both boards with 30 s intervals. All results are then compared. The infrared thermometer is being held at 15 cm from the board for all readings, while the heat camera is held at 20 cm from the board.

5. Results

5.1. Elaborated Design Results

Once the design has been elaborated there are several different graphical representations that can be accessed; the default is the schematic of the design. The schematic representation of the design is a pre-optimized design and is comprised of generic symbols such as AND gates, OR gates, adders and multipliers, it can be useful in helping uncover errors early in the design process. Figures 2 and 3 show the trojan-free design schematic and the trojan-inserted schematic. Another interesting representation that can be accessed is the design hierarchy, this is a graphical representation of the hierarchy of the design. The top level VHDL should always be at the top of the hierarchy. The final representation that

is of interest at elaboration level is that of the I/O pin mapping. This is a graphical representation of the constraints files with each of the mapped pins being marked, as shown in Figures 4 and 5. Although the differences are subtle close inspection reveals the extra pins being utilised by the trojan in the bottom left and right sections of Figure 5 (Highlighted in Red). The differences in the figures lie in the I/O pins (highlighted in red). These pins represent the extra ports being used by the trojan. The I/O Planning view layout represents graphically the various I/O, clock and logic objects present in the design. Through these representation, one can make design decisions and optimisations. In these Figures, we aim at showing the little difference between the trojan-free and trojan-inserted design. Further highlighting the requirements of both an expert eye to differentiate between both I/O mappings and the need for an off-the-shelf approach to detect trojans.

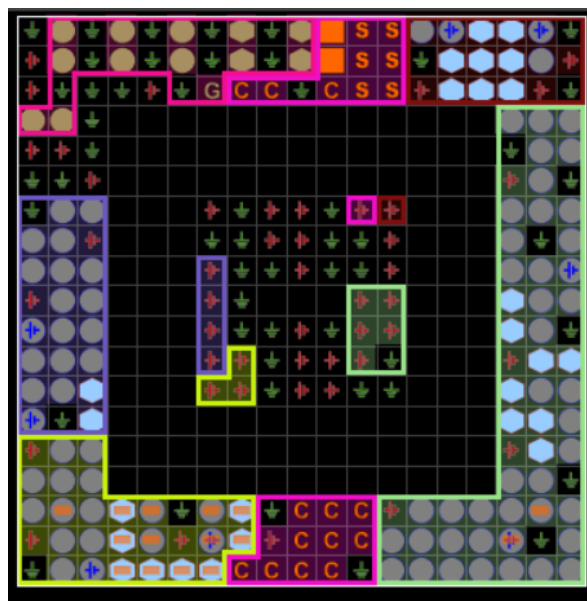


Figure 4. I/O pin mapping representation of the trojan-free design.

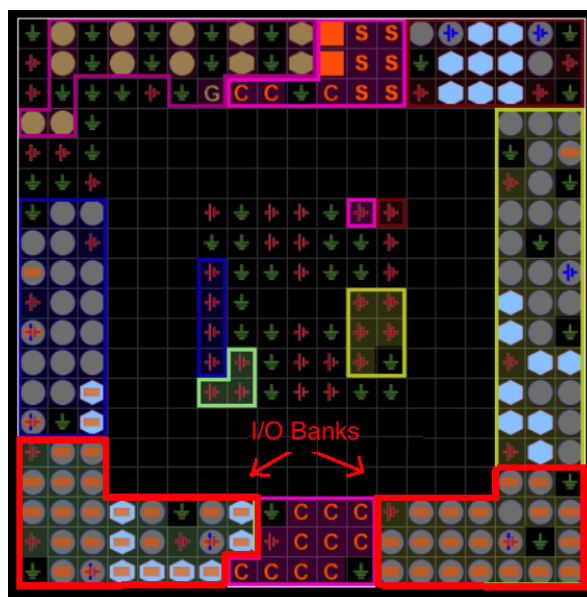


Figure 5. I/O pin mapping representation of trojan-inserted design.

5.2. Synthesised Design Results

Synthesis transforms the RTL design into a gate level netlist consisting of primitives such as Look-Up-Tables (LUTs) and as such offers new and updated representations. The default representation at this stage is a floor plan of the FPGA that can be zoomed into and examined in detail. The floorplans of the trojan-free and trojan-inserted circuits are shown in Figures 6 and 7. The presence of the trojan can be seen most clearly in Figure 7 sections X0Y0, X1Y0 and X1Y1. The differences between Figures 6 and 7 are the coloured sections, these indicate different layout with more or less cells being used. While it is unlikely to detect the trojan through the visualisation of the floor plan, the Figures aim at providing an overview of potential differences in the design.

The hierarchy of the design can also be accessed and examined; when examining the design it is found that both the trojan-free and trojan-inserted designs now contained significantly higher numbers of leaf cells. The hierarchal representation is shown in Figures 8 and 9. The Hierarchical Design enable to partition a design into small manageable modules that can further be processed independently from each other. This view also allows out-of-context (OOC) optimisation, when building a larger piece of software, or when building a more sophisticated trojan.

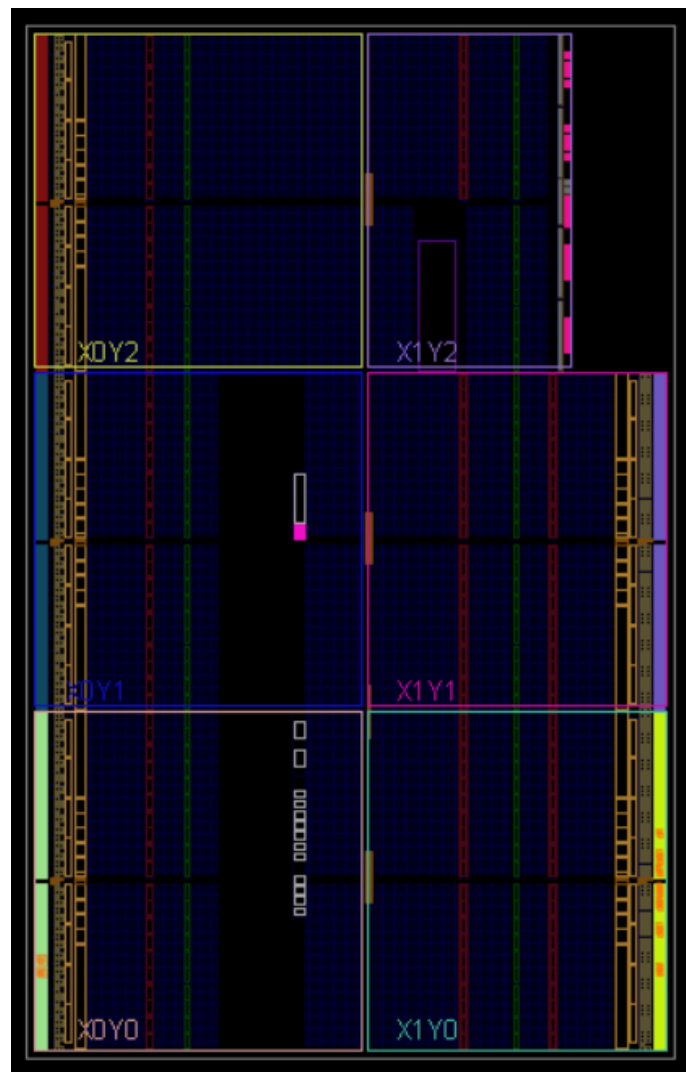


Figure 6. Floorplan-trojan-free.

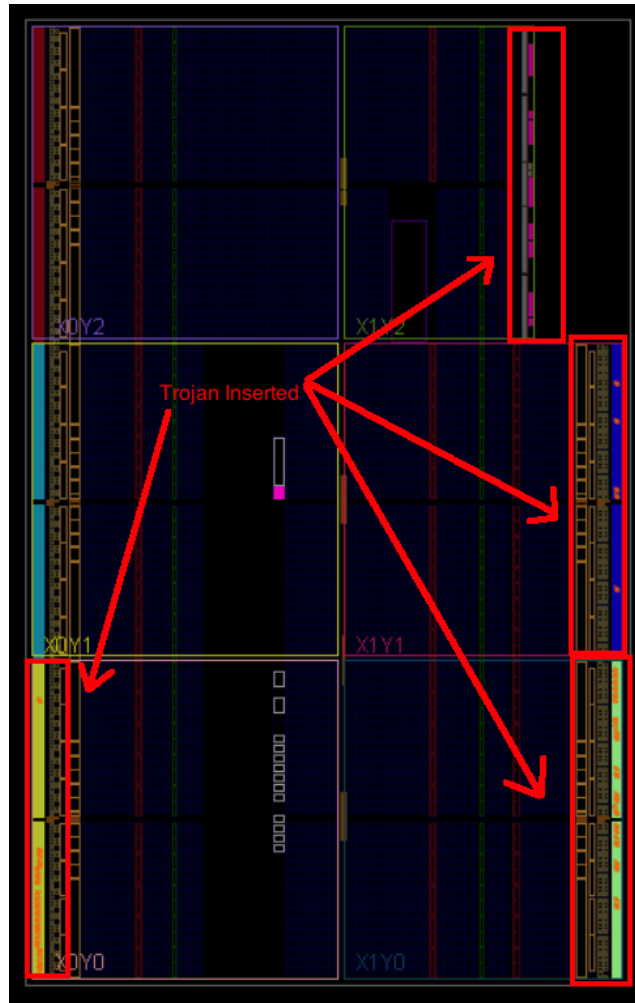


Figure 7. Floorplan-trojan-inserted.

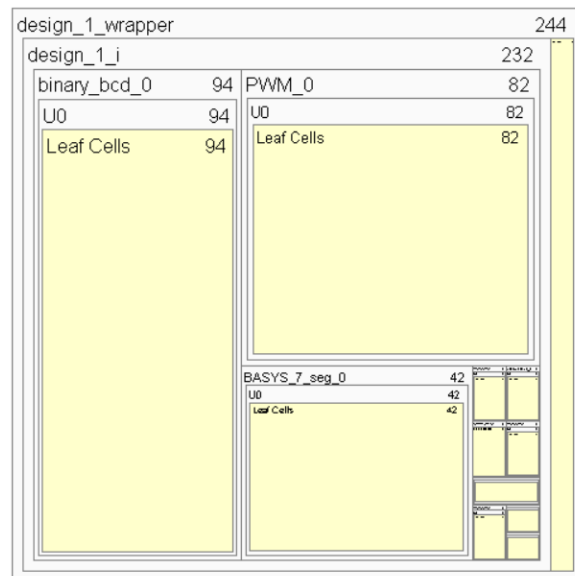


Figure 8. Design hierarchy-trojan-free.

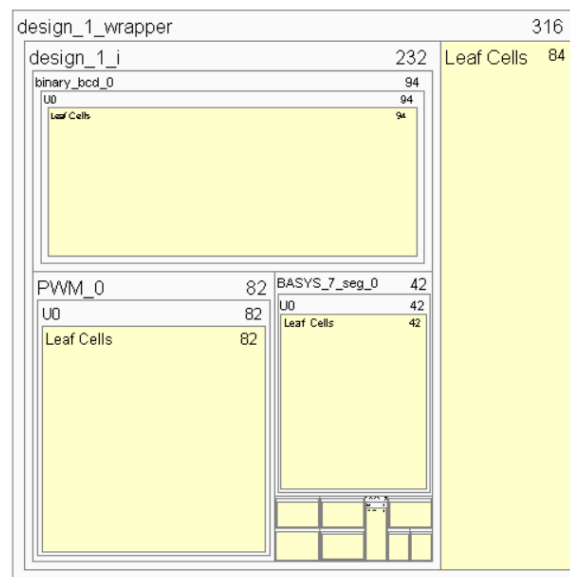


Figure 9. Design hierarchy-trojan-inserted.

5.3. Implemented Design Results

The successful completion of implementation allows access to further high level representations and the default presentation is a floorplan of the FPGA. The floorplans produced after implementation are more detailed than those produced during synthesis as the design has now been placed and routed in accordance with the constraints file.

5.4. Power Reports

The post implementation power reports summarise various factors such as on chip power and junction temperature. These reports were accessed to determine that the trojan-inserted board would indeed have the intended malicious purpose; to cause a junction overheat within the board. It is worth noting that the design is fully placed and routed during implementation so the report is indicative of the expected effect of the functioning design. The summary results for both the trojan-free and trojan-inserted boards are shown in Figures 10 and 11. These results can be used by the attacker to have an idea of the power required when designing the trojan. A high power consumption when active will render the stealthiness of the trojan useless, however, in this work we aim at detecting the trojan in a dormant state, hence, the power consumption during its active state does not affect the detection of the trojan, it is merely an indication of the stealthiness in an active state.

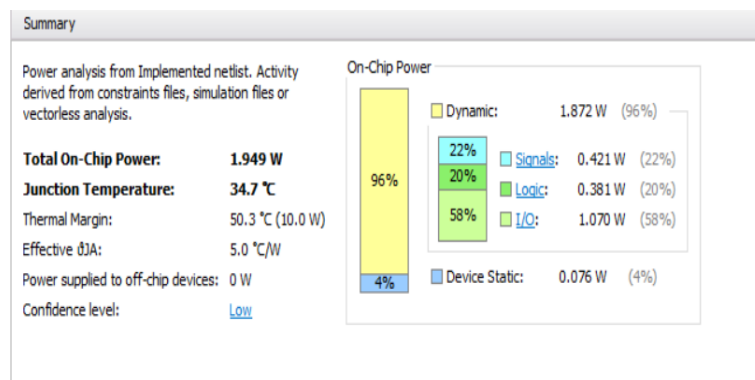


Figure 10. Trojan-free summary results.

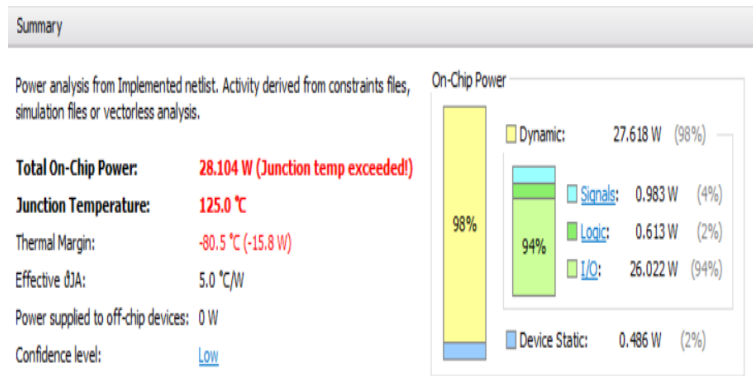


Figure 11. Trojan-inserted summary results.

5.5. Side Channel Analysis Results

During the side channel analysis a number of pin out were tested, the main difference was shown on the C101 capacitor. The C101 capacitor located next to the FPGA chip on the board [29]. The results are shown in Table 2. The fluctuations can also be seen in Figures 12 and 13 both figures shows oscilloscope readings. the trojan-free board requires 3.49 v while the trojan insert reading demonstrates that the board requires 3.6 v. The trojan was dormant at all times, during the readings, when active the trojan required over 3.9 v. This significant fluctuation was repeated over ten times, and provided us with the same reading during each iteration, essentially demonstrating that minimalist trojans do require more power and can be identified using side channel analysis with off-the-shelf oscilloscope.

Table 2. Results of C101 Capacitor.

C101 FPGA Power in	Trojan-Inserted	Trojan-Free
V amptd	3.665 V	3.493 V
V avg	-87.77 mV	-29.631 mV
V p-p	3.665 V	3.493 V

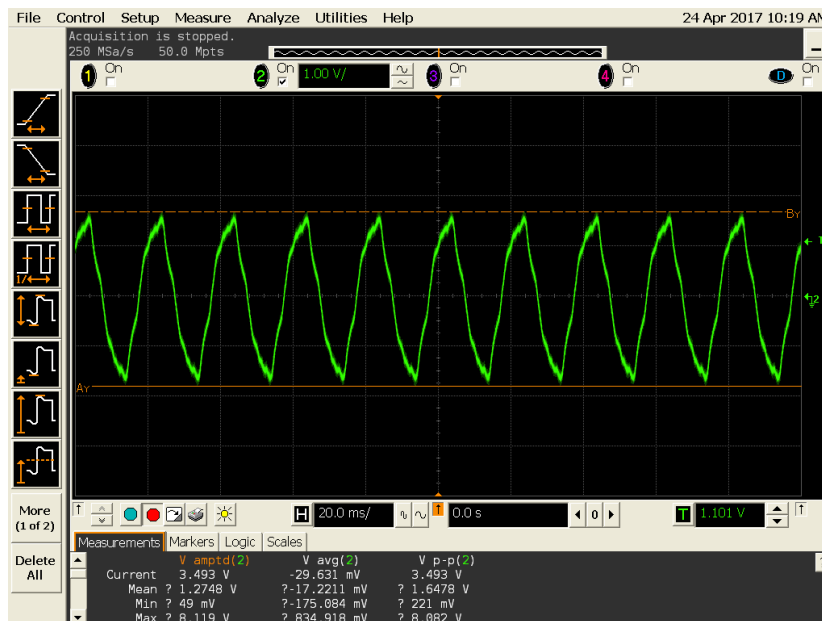


Figure 12. Trojan-free readings of the C101 Capacitor Charging and Discharging.

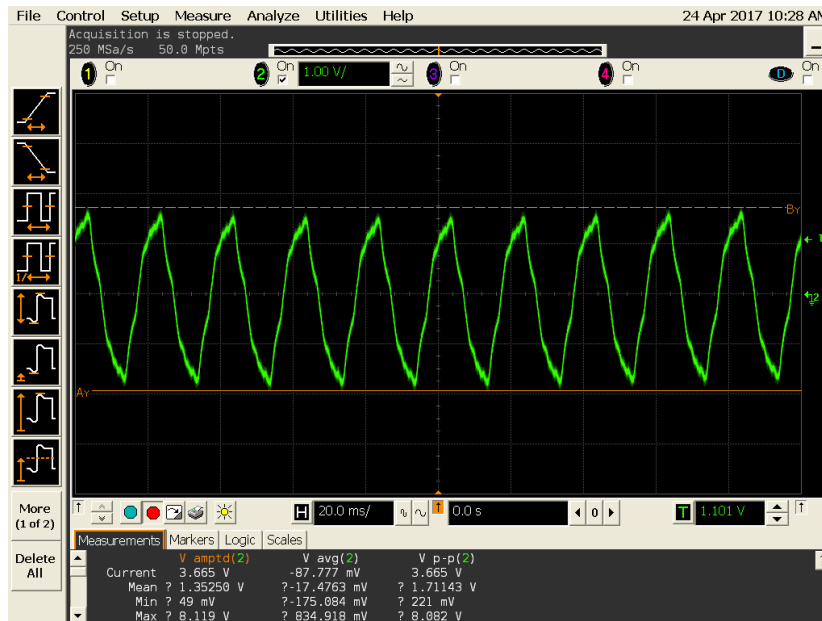


Figure 13. Trojan-inserted readings of the C101 Capacitor Charging and Discharging.

5.6. Infrared Thermometer Results

The results of the infrared thermometer showed a gradual increase in temperature over time for both boards, however the trojan infected board had a higher temperature, the results are shown in Table 3.

Table 3. Infrared thermometer results.

Run No.	Trojan in °C	Trojan-Free in °C
1	17.9	17.4
2	18.1	17.4
3	18.2	18.1
4	18.4	18.2
5	18.6	18.4
6	19.0	18.6
7	19.2	18.8
8	19.2	19.0
9	19.5	19.2
10	19.6	19.4

As shown in Table 3 the temperatures taken on the trojan-infected board are on average 0.2 °C above those of the trojan-free board. While the increases in temperature over time can be attributed to the counter program as it runs continuously, the fact that the trojan-inserted board was tested first negates this as a reason for the different temperatures. Prior to testing the trojan-inserted board, there had been no power to the board for over an hour so residual temperature from another test can also be ruled out. These factors combined with the recorded drop in temperature of the trojan-free board can be considered to be proof of the presence of the trojan.

5.7. Heat Camera

The board was also subject to a heat camera test in order to detect the trojan in three different states (Free, Sleeping and Active). The results of the heat camera confirmed the results obtained through the infrared thermometer. As shown in Figure 14 the temperature increases over time. Figure 14a from left to right presents the readings for the trojan-free, sleeping and active. Figure 14b

provide a close up reading of the trojan in sleeping mode over a period of 10 min. Finally, Figure 14c provides close up reading of the active trojan board over a period of 10 min. As previously observed in Section 5.6 it is possible to notice an obvious difference in temperature between the trojan-free and trojan sleeping boards.

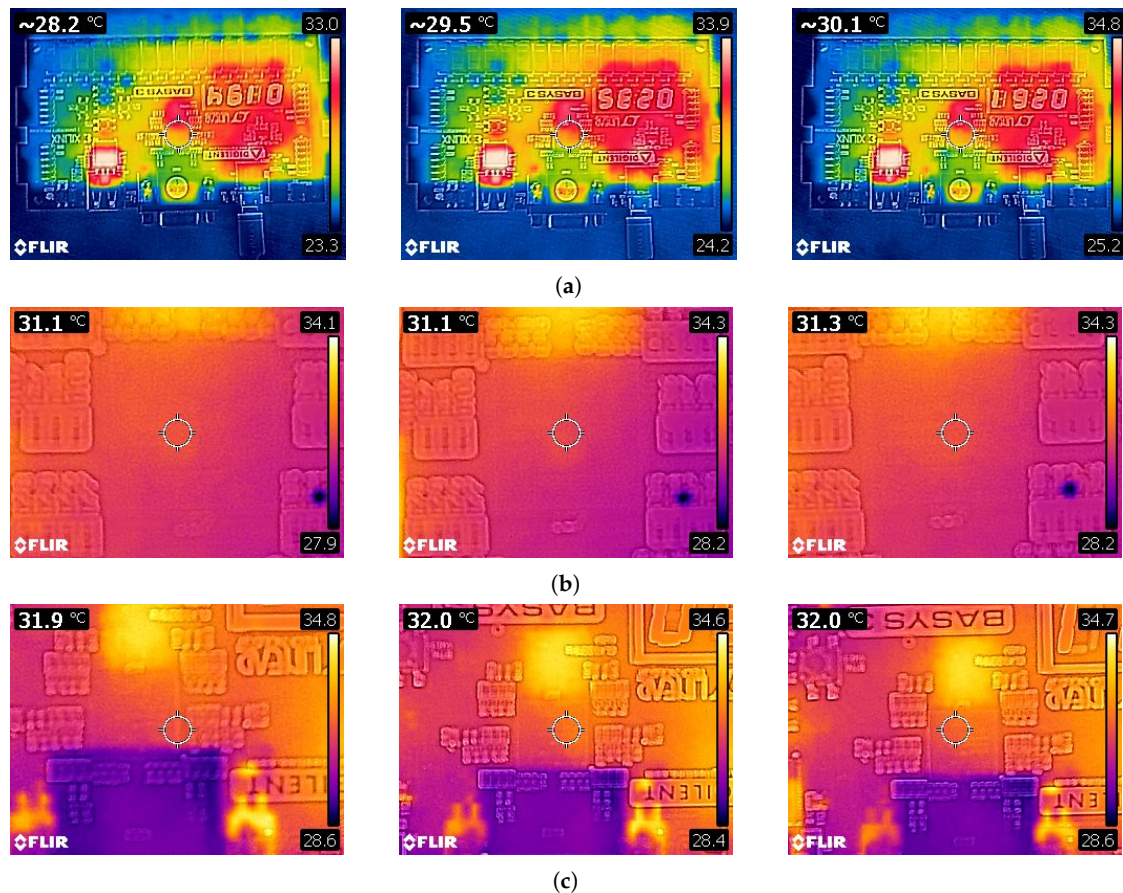


Figure 14. The set of Figure (a) from right to left represents the readings for the trojan free, sleeping and active respectively. The set of Figure (b) demonstrate a close up representation of the trojan free, sleeping and active on the board, while the set of Figure (c) provide close up temperature readings of the trojan active on the board over a period of 10 min.

Note that there is a difference between the readings of the heat gun and the heat camera of approximately 10 °C; we expect this to be due to the calibration of both the camera and the heat gun. While the camera had calibration options only a laboratory multi-point calibration can provide a high accuracy, moreover, the detector can experiences different temperatures due to ambient temperature changes and to the heat dissipation of the material. However, this does not make the results less relevant. The temperature difference remains constant across the devices and provides higher temperatures for trojan in sleep mode and in active mode.

6. Discussion

In this section, the results are discussed in detail. Our test provided the corner-stone for hardware trojan creation and detection using off-the-shelf technologies against a golden model. The practical work presented in this manuscript demonstrated the feasibility of hardware trojan detection using side-channel attacks, a heat camera, and a heat gun. While the settings of these tests where pre-defined, it is not un-common for this type of work as suggested in [30].

The three tests required to provide readings between the trojan-inserted and trojan-free boards w to determine the existence of the trojan. In an industrial setting, it is unlikely to suffice, however, the results presented in this paper provide evidence that trojans can be detected using off-the-shelf hardware due to the increase in accuracy of the devices and the access to technologies previously reserved to specialized industries, and which are now becoming commodities.

6.1. Thermal Camera

The thermal camera provided extensive results of the three stages of the trojan. This technique provides a good visual representation, demonstrating the presence of the trojan on the board. However, as aforementioned, in order for the camera to provide accurate results a trojan-free board is needed. While this technique is accurate, when used as a stand-alone test it requires the camera to be well calibrated in order to provide the best results. In the setting presented in this manuscript, both the trojan taxonomy and its purpose where known, hence, during this white-box testing, the full calibration of the camera could be overlooked.

6.2. Heat Gun

The heat gun was able to demonstrate the presence of a hardware trojan in different settings, while the heat gun was accurate enough to detect the temperature variation, as for the heat camera, this test requires to have free trojan board for comparison. As aforementioned, this is not uncommon for this type of work [30]. While the results provided by the heat gun are notable, they differ greatly from the thermal camera. It was expected from the start for the thermal camera to outperform the heat gun, due to the low accuracy of the heat gun.

6.3. Side Channel Analysis

The side channel analysis techniques was employed in attempts to locate the trojan whilst in its dormant form. Measurements were taken from several pins and capacitors during this process, whilst most of them showed little difference in reading only the C101 capacitor demonstrated a significant difference.

7. Conclusions

This work provides sharper bounds for the case of detection of hardware trojans using off-the-shelf devices, allowing to reduce the costs associated with trojan detection. The increasing number of devices being produced by untrusted foundries puts critical infrastructure at the center of attention. In this manuscript, we highlighted the dilemma of finding a one fits all solution to the problem finding hardware trojans fitting different taxonomies. To this end, we presented the corner stone for the detection of hardware trojans using off-the-shelf devices. We successfully demonstrated the ability of off-the-shelf devices to detect trojans in different settings, namely: sleeping and active. We believe that our practical work has the enormous potential in the successful detection of hardware trojans. In the future we will aim at developing techniques to use thermal imaging for the detection of large scale hardware trojan infection and explore other trojan taxonomies in more intricate designs and with advanced malicious purposes. While we believe this technique is fully applicable to FPGAs, the technique might not be well suited for the denser ASICs and slight modifications might be required both in the methodology and in the off-the-shelf tool used. Moreover, we believe that this method could be used widely with the democratisation of specialised off-the-shelf hardware, following Moore's law with a higher detection accuracy and better thermal imaging capabilities. Future work will compare the technique proposed against smaller known trojans and the process variation and manufacturing variation will be taken into account. Furthermore, the number of test vectors for Vivado power estimator will be increased in order to increase its accuracy.

Author Contributions: Conceptualization, C.R. and X.B.; Methodology, C.R. and X.B.; Software, C.R.; Validation, C.R.; Formal Analysis, C.R.; Investigation, C.R., X.B. and A.S.; Resources, X.B.; Data Curation, C.R.; Writing—Original Draft Preparation, C.R.; Writing—Review & Editing, C.R., X.B. and A.S.; Visualization, C.R.; Supervision, X.B. and A.S.; Project Administration, X.B.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Keysight Technologies for their valuable input and the oscilloscope used in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tehranipoor, M.; Koushanfar, F. A survey of hardware trojan taxonomy and detection. *IEEE Des. Test Comput.* **2010**, *27*. [[CrossRef](#)]
2. Huffmire, T.; Brotherton, B.; Sherwood, T.; Kastner, R.; Levin, T.; Nguyen, T.D.; Irvine, C. Managing security in FPGA-based embedded systems. *IEEE Des. Test Comput.* **2008**, *25*. [[CrossRef](#)]
3. Huffmire, T.; Irvine, C.; Nguyen, T.D.; Levin, T.; Kastner, R.; Sherwood, T. *Handbook of FPGA Design Security*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
4. Akram, R.N.; Markantonakis, K.; Holloway, R.; Kariyawasam, S.; Ayub, S.; Seeam, A.; Atkinson, R. Challenges of security and trust in avionics wireless networks. In Proceedings of the 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 13–17 September 2015; p. 4B1-1.
5. Omoogun, M.; Ramsurrun, V.; Guness, S.; Seeam, P.; Bellekens, X.; Seeam, A. Critical patient eHealth monitoring system using wearable sensors. In Proceedings of the 2017 1st International Conference on Next Generation Computing Applications (NextComp), Mauritius, 19–21 July 2017; pp. 169–174. [[CrossRef](#)]
6. Bellekens, X.; Seeam, A.; Nieradzinska, K.; Tachtatzis, C.; Cleary, A.; Atkinson, R.; Andonovic, I. Cyber-physical-security model for safety-critical iot infrastructures. In Proceedings of the Wireless World Research Forum Meeting, Santa Clara, CA, USA, 21–23 April 2015.
7. Robinson, W.H.; Reece, T.; Mahatme, N.N. Addressing the challenges of hardware assurance in reconfigurable systems. In Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Las Vegas, NV, USA, 22–25 July 2013; p. 1.
8. Department of Justice Press Release. VisionTech Administrator Sentenced to Prison for Role in Sales of Counterfeit Circuits Destined to US Military. Available online: <https://www.ice.gov/news/releases/visiontech-administrator-sentenced-prison-role-sales-counterfeit-circuits-destined-us> (accessed on 17 September 2017).
9. Mitra, S.; Wong, H.S.P.; Wong, S. Stopping hardware Trojans in their tracks. *IEEE Spectr.* **2015**. Available online: <https://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks> (accessed on 22 July 2018).
10. Narasimhan, S.; Du, D.; Chakraborty, R.S.; Paul, S.; Wolff, F.; Papachristou, C.; Roy, K.; Bhunia, S. Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010; pp. 13–18.
11. Cao, C.; Guan, L.; Liu, P.; Gao, N.; Lin, J.; Xiang, J. Hey, you, keep away from my device: Remotely implanting a virus expeller to defeat Mirai on IoT devices. *arXiv* **2017**, arXiv:1706.05779.
12. Pyrgas, L.; Pirpilidis, F.; Panayiotarou, A.; Kitsos, P. Thermal Sensor Based Hardware Trojan Detection in FPGAs. In Proceedings of the 2017 Euromicro Conference on Digital System Design (DSD), Vienna, Austria, 30 August–1 September 2017; pp. 268–273.
13. Nowroz, A.N.; Hu, K.; Koushanfar, F.; Reda, S. Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2014**, *33*, 1792–1805. [[CrossRef](#)]
14. Forte, D.; Bao, C.; Srivastava, A. Temperature tracking: An innovative run-time approach for hardware Trojan detection. In Proceedings of the 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 18–21 November 2013; pp. 532–539. [[CrossRef](#)]

15. He, J.; Zhao, Y.; Guo, X.; Jin, Y. Hardware Trojan Detection Through Chip-Free Electromagnetic Side-Channel Statistical Analysis. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 2939–2948. [[CrossRef](#)]
16. Shende, R.; Ambawade, D.D. A side channel based power analysis technique for hardware trojan detection using statistical learning approach. In Proceedings of the 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), Hyderabad, India, 21–23 July 2016; pp. 1–4. [[CrossRef](#)]
17. Xiao, K.; Forte, D.; Jin, Y.; Karri, R.; Bhunia, S.; Tehranipoor, M. Hardware Trojans: Lessons learned after one decade of research. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2016**, *22*, 6. [[CrossRef](#)]
18. Kumar, R. *Fabless Semiconductor Implementation*; McGraw-Hill, Inc.: New York, NY, USA, 2008.
19. Adee, S. The hunt for the kill switch. *IEEE Spectr.* **2008**, *45*, 34–39. [[CrossRef](#)]
20. Yeh, A. Trends in the global IC design service market. *DIGITIMES Res.* **2012**. Available online: <https://www.digitimes.com/news/a20120313RS400.html?chid=2> (accessed on 22 July 2018).
21. Abramovici, M.; Bradley, P. Integrated circuit security: New threats and solutions. In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, Knoxville, TN, USA, 13–15 April 2009; p. 55.
22. Chakraborty, R.S.; Narasimhan, S.; Bhunia, S. Hardware Trojan: Threats and emerging solutions. In Proceedings of the High Level Design Validation and Test Workshop (HLDVT), San Francisco, CA, USA, 4–6 November 2009; pp. 166–171.
23. Collins, D.R. *Trust in Integrated Circuits*; Technical Report; Defense Advanced Research Projects Agency Arlington Va Microsystems Technology Office: Arlington, VA, USA, 2008.
24. Wolff, F.; Papachristou, C.; Bhunia, S.; Chakraborty, R.S. Towards Trojan-free trusted ICs: Problem analysis and detection scheme. In Proceedings of the Conference on Design, Automation and Test in Europe, Munich, Germany, 10–14 March 2008; pp. 1362–1365.
25. Sanno, B. *Detecting Hardware Trojans*; Ruhr-University: Bochum, Germany, 2009.
26. Banga, M.; Hsiao, M.S. A region based approach for the identification of hardware Trojans. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 9 June 2008; pp. 40–47.
27. Jin, Y.; Makris, Y. Hardware Trojan detection using path delay fingerprint. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 9 June 2008; pp. 51–57.
28. Wang, L.W.; Luo, H.W. A power analysis based approach to detect Trojan circuits. In Proceedings of the 2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, Xi'an, China, 17–19 June 2011; pp. 380–384. [[CrossRef](#)]
29. Digilent Basys 3 Artix7 FPGA Board. Available online: https://reference.digilentinc.com/_media/basys3_basys3_sch.pdf (accessed on 7 July 2018).
30. Bao, C.; Forte, D.; Srivastava, A. Temperature tracking: Toward robust run-time detection of hardware Trojans. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1577–1585. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).