IOWA STATE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SENIOR DESIGN PROJECT

TEAM DEC 1515

PROJECT PLAN VERSION 1

Project title: Trusted Electronic Systems

TEAM MEMBERS:

1. Fatema Aftab – Team leader
2. Colton Bachman – Team Webmaster
3. Xin Hu – Team Webmaster
4. Yitao (Tyler) Liu – Key concept holder
5. Qian Chang – Team communication
6. Chao Huang – Team communication

Group email: dec1515@iastate.edu

Client/ Company/ Organization: Dr. Randy Geiger and Dr. Degang Chen

Client Contacts/Advisors:

1. Dr. Randy Geiger

Advisors Email: rlgeiger@iastate.edu

2. Dr. Degang Chen

Advisors Email: djchen@iastate.edu

Abstract

Our senior design project is about hardware Trojans. A hardware Trojan is a malicious, dangerous and an intentional attempt of a designer towards an electronic circuit design resulting in nasty consequences. It is basically a backdoor that can be inserted into a system which can benefit the bad guy in controlling the entire hardware.

Hardware Trojans are capable of defeating nearly all of the security mechanisms these days because almost every hardware system has millions of transistors installed in them. The Trojan might leak important information to a third party, modify how the system functions, or it can also be a simple Denial of Service (DoS).

Table of Contents

Glossary

ALU Arithmetic Logic Unit

ASIC Application Specific Integrated Circuit

CAD Computer Aided Design

CMOS Complementary Metal-Oxide-Semiconductor

CPU Central Processing Unit

DoS Denial of Service

EDA Electronic Design Automation

EM Electron Migration

FPGA Field Programmable Gate Array

HDL Hardware Description Language

IC Integrated Circuit

IP Intellectual Property

IO Input/Output

JTAG Joint Test Action Group specified IC test interface

RF Radio Frequency

RTL Register Transfer Level

SoC System on Chip

TCB Trusted Computing Base

VHDL VHSIC Hardware Description Language

Executive Summary

Scientists have been working on hardware Trojans since last few decades because our world is overwhelmed with technology. Everything we use these days has control over us whether it is just an automation device or a computer. The ability for a common man to trust all these devices is becoming a concern. The security of our devices is making the integrated Circuits (IC) main topic of research for a lot of engineers. This is because ICs are used in procuring and maintain cyber, military, financial, and other critical systems. Therefore, this project mostly focuses on the hardware security, most importantly on identifying methods that can be used to introduce the Trojans into the electronic systems. Moreover, this project also emphasizes on methods for detecting these counterfeited electronic components and devices. We will also study the Trojans that reside in the software which helps trigger the Trojan in the hardware. More and more industries are getting aware of this vulnerability and millions of dollars are being invested to find the way out this problem.

Hardware Trojan insertion into an IC is very easy and can occur at any point of design. We know that there are different steps in making of an IC namely, specification, design, verification and manufacturing. A hardware Trojan can be built in any time and any of these steps. Maintaining a tight eye onto all these processes is very time and money consuming. These days most industries are separating all these different processes and increasing the use of Electronic Design Automation (EDA).

Hardware Trojans are very difficult to spot. They are hard to observe during the primitive IC verification and testing phases. It's tough to find the payloads and trigger states. Most ASIC designs are too huge to do exhaustive testing and verification on them so it is very likely that a lot of Trojans go undetected every day.

For making next generation defensive mechanisms for the development of electronic circuits we need to understand hardware Trojans. Even after a bunch of research, engineers still do not have a system that can ensure the device safety. This makes our project challenging yet very interesting for us because in near future it will be necessary for hardware systems to perform securely in presence of Trojans.

Introduction

In our project we want to introduce a hardware Trojan in a vending machine design. We are going to use a commercial vending machine which works on FPGA and introduce a hardware Trojan. We chose a vending machine design because it is a very applicable and a practical example of a hardware Trojan. This gives us capability of making Trojan as bad as possible and also come up with its detection and prevention methods. The reason we want a vending machine on FPGA is because we all are familiar with the operation of FPGA board after taking CPR E 281 (Digital Logic). The program is Verilog which uses Hardware Descriptive Language (HDL) and this is a programming language we all know.

Process Details

With this project we had to keep in mind three most important aspects of hardware Trojan:

1. Creating a hardware Trojan
2. Detection of Trojan
3. Steps to prevent the Trojan

For all these parts to come together and completing the project we had to understand the classification and trigger mechanisms of the Trojans.

Threats

Hardware Trojans are very obstinate once they are installed in a system they are always there. They are active when they system is turned on. They are able to change any and almost all electronic systems by infecting and changing the behavior of the integrated circuit. Considering the threats, understanding the insertion and activation mechanisms becomes very important. So therefore we researched classification of Trojans and how it affects the system. For our vending machine design we are thinking about making a sequential, always on and externally triggered Trojan.

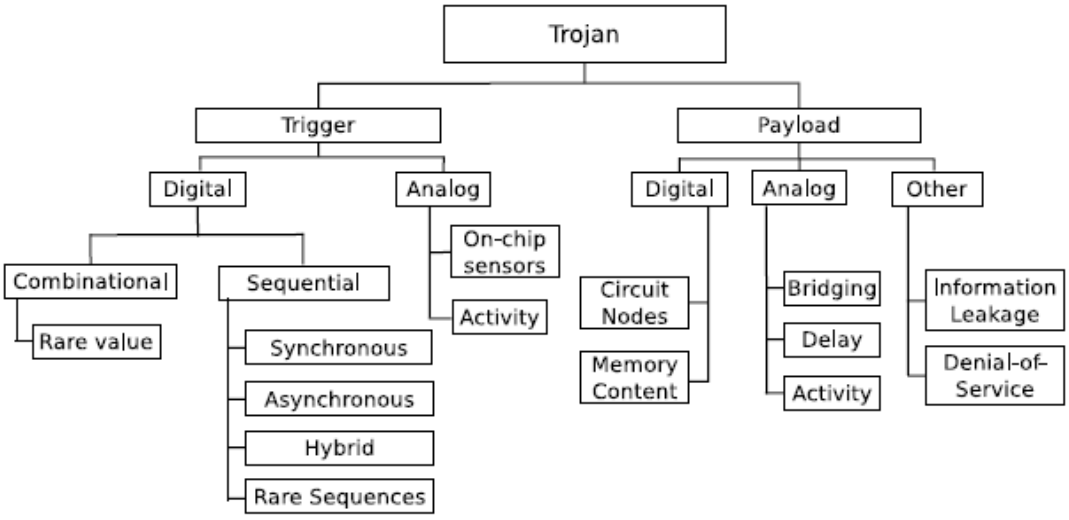


Figure 1: Hardware Trojan Taxonomy: Chakraborty, Narasimhan and Bhunia (2010)

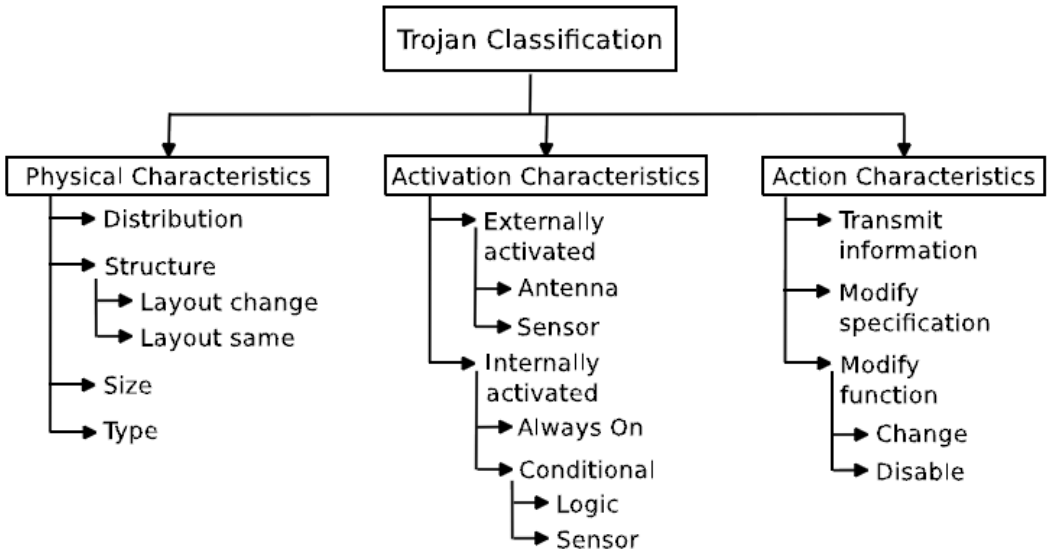


Figure 2: Hardware Trojan Taxonomy: Wang, Tehranipoor and Plusquellic (2008)

Trigger Mechanisms

A trigger mechanism means the activation of a Trojan once it is inserted in to the software/hardware. There are three types of trigger mechanisms:

1. Internally Triggered
   i) Combinational Activation
   ii) Sequential Activation
2. Externally Triggered
3. Always on

Sequential Activation

This Trojan relies on the sequence of events occurring for activation. Mostly sequential activation helps with turning on a certain number of clock cycles.

Externally Triggered

Externally triggered Trojans depend on some interaction of the outside environment with the Trojan. It can be any sort of human interaction with the IC. The target is immediately activated if a button or switch is pressed. A trigger can also come from another component, such as a light sensor, which is connected to the target.

Always on

This is a Trojan which is always on and does not need any activation from any other source. This is never turned off which means that this Trojan is always working once it is inserted in the system. This Trojan is mostly used in leaking data through a circuit which has a side channel inside an IC.

Combinatorial Activation

The Trojan is activated by using a combination of buttons or switches. For example, if the user held down the first, third, and sixth buttons on the vending machine.

Test Plan

Since we have decided to do always on trigger mechanism our Trojan is going to be always on in the vending machine schematic. That means that there is no outside interference needed for the Trojan to get activated. There will be two kinds of Trojans in the vending machine. One Trojan will trick the customers into putting extra quarters/ dimes more than what the product costs. Another Trojan will be activated which will help us take the money out of the vending machine. The first Trojan could be classified into "always on" and the other Trojan will be "externally triggered" Trojan.

Once these two Trojans are inserted into the design we want to make the Trojan more insidious by having the Trojan trick only random customers so it is less obvious to the owner that something is wrong with the machine. We have to trick the owner into believing that the machine can sometimes do Denial of Service by taking extra money or not returning change. The externally triggered Trojan will give us control of how much money we want to take out of the vending machine. It will be activated by entering a sequence of coins which is specifically entered in the machine. For example, putting 5 pennies in a row will activate the sequential Trojan and gives us access to the money in the vending machine.

Our Trojan will be activated using sequential and combinatorial logic. For example, the Trojan is activated if you insert 2 dimes, 1 quarter, and 5 nickels while holding the coin return button.

Trojan Design

We have created four different types of Trojans:

1. Block inputs
2. Block outputs
3. Flush all outputs
4. Set all items price to zero

Trojan1: Block inputs

```
`include "Coninsertsystem.v";
Reg[1:0] trojancase2;
Always @(input1) begin
If(input1[3:2] == trojancase2[1:0]) begin
        CLK <= 0'b1;
      tranjan_state <= 0100'b;
      while(1){do nothing}   .
End else begin
    normal runs
```

Trojan2: Block outputs

```
`include "Changesystem.v";
Reg[1:0] trojancase3;
Always @(change) begin
If(change[4:3] == trojancase3[1:0]) begin
        tranjan_state <= 001'b;
        while(1){do nothing}

End else begin
    normal runs
```

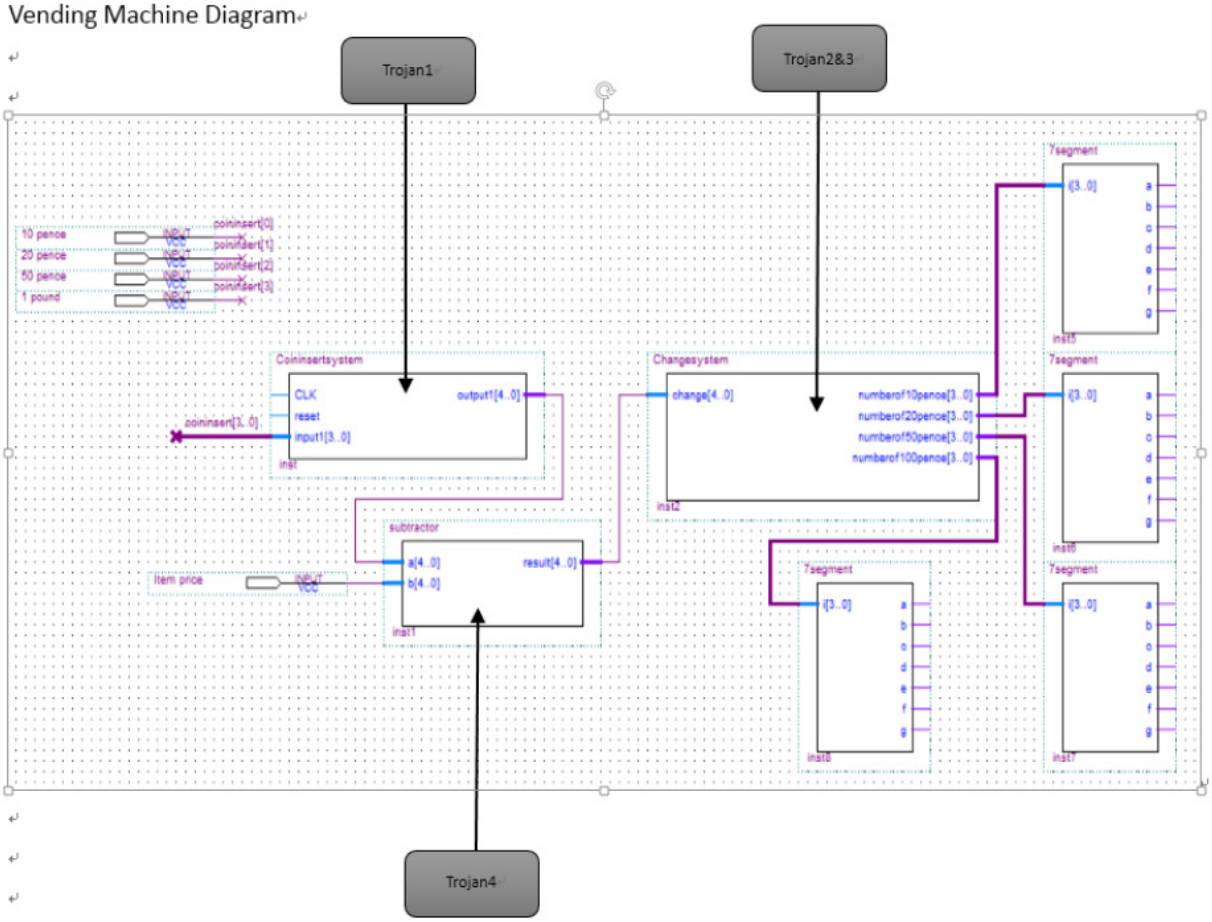
Trojan3: Flush all outputs

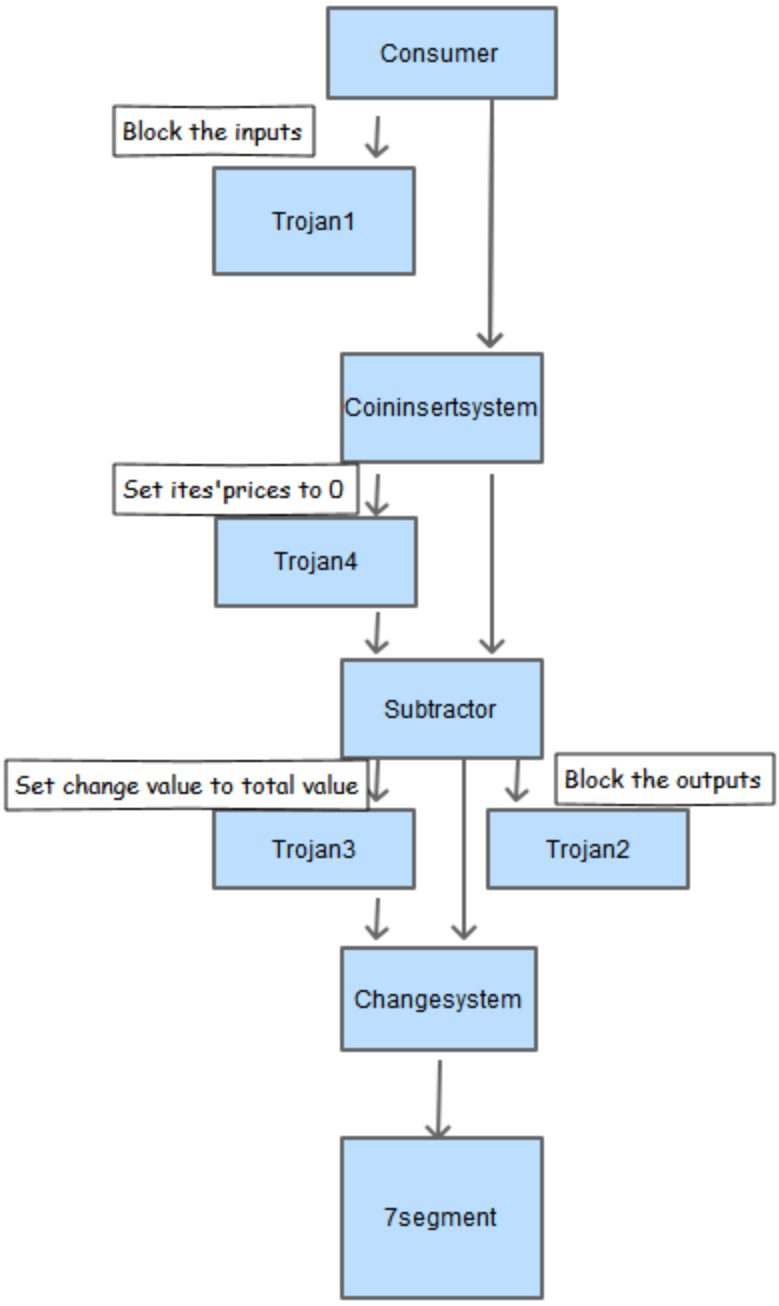
```
`include "Changesystem.v";
Reg[1:0] trojancase4;
Always @(output) begin
If(input[3:2] == trojancase4[1:0]) begin
        tranjan_state <= 010'b;
        output <= total;
        input1 <= 0`b1;
        o_clk <= 1'b1;
        reset <= 1'b1;
End else begin
    normal runs
```

Trojan4: set all item price to 0;

```
`include "subtraor";
Reg[1:0] trojancase5;
Always @(output1) begin
If(input[3:2] == trojancase5[1:0] begin
        Trojan_state <= 011'b;
        Item_price <= 0'b1;
        B <= Item_price;
        P_clk <= CLK;
End else begin
    Normal run subtractor;
```

Vending Machine Diagram




Consumer
Block the inputs
Trojan1
Coininsertsystem
Set ites'prices to 0
Trojan4
Subtractor
Set change value to total value
Block the outputs
Trojan3
Trojan2
Changesystem
7segment

Some design drafts for our Vending machine codes:

```
module money_input (M1,M2,M3,CLK,control);
    //create a control bit that control the delay of two money inputs
    input M1,M2,M3,TJset, CLK;
    output control;
    reg control;
    reg [9:0] counter;
    reg [1:0]flag;

    initial counter = 0;
    initial flag=00;
    always @ (M1 or M2 or M3 or TJset) begin
        if (M1|M2|M3)
            flag=01;

        else if(TJset)
            flag = 11;
    end

    always @ (posedge CLK) begin
        if (flag==01)
            counter <= counter + 1;
            else if(flag == 11)
            counter = 0; //reset the counter if trojan set, it used for always block ed
conter bit to 0;
    end

    always @ (counter) begin
        if (counter == 1)
            control= 1;
        else if (counter >= 1000)
            counter <=0;
        else
            control=0;
    end
endmodule
```

```
module input_to_7_bits(Plus_Subtract,MI1,MI2,MI3,P0,P1,P2,P3,P4,P5,P6,b0,b1,b2,b3,b4,b
/*
convert the input data to 7 bits
*/
    input Plus_Subtract,MI1,MI2,MI3,P0,P1,P2,P3,P4,P5,P6, TJ1;
    output b0,b1,b2,b3,b4,b5,b6;
    reg b0,b1,b2,b3,b4,b5,b6;
    always @ (Plus_Subtract or MI1 or MI2 or MI3 or P0 or P1 or P2 or P3 or P4 or P5 o
 begin
        if(Plus_Subtract==0)begin
            case({MI1, MI2, MI3})
                3'b001:begin b0=0;b1=0;b2=0;b3=0;b4=0;b5=0;b6=1;end // 5 cents
                3'b010:begin b0=0;b1=0;b2=0;b3=0;b4=0;b5=1;b6=0;end //10 cents
                3'b011:begin b0=0;b1=0;b2=0;b3=0;b4=1;b5=0;b6=1;end //25 cents
                3'b100:begin b0=0;b1=0;b2=0;b3=1;b4=0;b5=1;b6=0;end //50 cents
                3'b101:begin b0=0;b1=0;b2=1;b3=0;b4=1;b5=0;b6=0;end //1 dollar
                3'b110:begin b0=1;b1=1;b2=0;b3=0;b4=1;b5=0;b6=0;end //5 dollar
            endcase
        if( Plus_Subtract==0||TJ1)begin
            case({MI1, MI2, MI3})
                3'b001:begin b0=1;b1=0;b2=0;b3=0;b4=0;b5=0;b6=1;end // 5 cents
                3'b010:begin b0=1;b1=0;b2=0;b3=0;b4=0;b5=1;b6=0;end //10 cents
                3'b011:begin b0=1;b1=0;b2=0;b3=0;b4=1;b5=0;b6=1;end //25 cents
                3'b100:begin b0=1;b1=0;b2=0;b3=1;b4=0;b5=1;b6=0;end //50 cents
                3'b101:begin b0=1;b1=0;b2=1;b3=0;b4=1;b5=0;b6=0;end //1 dollar
                3'b110:begin b0=1;b1=1;b2=0;b3=0;b4=1;b5=0;b6=0;end //5 dollar
            endcase
        //changed for set all money input to 20 dollars
        end
        else begin
            b0=P0;
            b1=P1;
            b2=P2;
            b3=P3;
            b4=P4;
            b5=P5;
            b6=P6;
        end
    end
endmodule
```

```
module LCD_7_segment(A,B,C,D,E,F,G,Z,Y,X,W,);
    input Z,Y,X,W, tj2;
    output A,B,C,D,E,F,G;
    reg A,B,C,D,E,F,G;
    always @(A or B or C or D or E or F or G)
    begin
    if(tj2==0)begin
        case ({Z,Y,X,W})
            4'b0000:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0001:begin A='b1;B='b0;C='b0;D='b1;E='b1;F='b1;G='b1; end
            4'b0010:begin A='b0;B='b0;C='b1;D='b0;E='b0;F='b1;G='b0; end
            4'b0011:begin A='b0;B='b0;C='b0;D='b0;E='b1;F='b1;G='b0; end
            4'b0100:begin A='b1;B='b0;C='b0;D='b1;E='b1;F='b0;G='b0; end
            4'b0101:begin A='b0;B='b1;C='b0;D='b0;E='b1;F='b0;G='b0; end
            4'b0110:begin A='b0;B='b1;C='b0;D='b0;E='b0;F='b0;G='b0; end
            4'b0111:begin A='b0;B='b0;C='b0;D='b1;E='b1;F='b1;G='b1; end
            4'b1000:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b0; end
            4'b1001:begin A='b0;B='b0;C='b0;D='b0;E='b1;F='b0;G='b0; end
            4'b1010:begin A='b0;B='b0;C='b0;D='b1;E='b0;F='b0;G='b0; end
            4'b1011:begin A='b1;B='b1;C='b0;D='b0;E='b0;F='b0;G='b0; end
            4'b1100:begin A='b0;B='b1;C='b1;D='b0;E='b0;F='b0;G='b1; end
            4'b1101:begin A='b1;B='b0;C='b0;D='b0;E='b0;F='b1;G='b0; end
            4'b1110:begin A='b0;B='b1;C='b1;D='b0;E='b0;F='b0;G='b0; end
            4'b1111:begin A='b0;B='b1;C='b1;D='b1;E='b0;F='b0;G='b0; end
            endcase

            if(tj2==1)begin
        case ({Z,Y,X,W})
            4'b0000:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end //set
            4'b0001:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0010:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0011:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0100:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0101:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0110:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b0111:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1000:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1001:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1010:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1011:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1100:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1101:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1110:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            4'b1111:begin A='b0;B='b0;C='b0;D='b0;E='b0;F='b0;G='b1; end
            endcase
        end
    endmodule
```

Non-functional Requirements/ Project Issues

A hardware Trojan is easy to create for a designer but very difficult to detect it. The Trojan is mostly hidden in the schematic and we have to find a way to make it as least obvious as possible. The code can change the way the Trojan works on FPGA, but detecting the Trojan in the code is going to be a challenge. Hybrid triggers that combine all the other trigger mechanisms will make the job of finding the Trojan more difficult. Another problem with detection is that every IC works in a different way and has a different objective to fulfill in order for it to work. Making a detection method for a general Trojan is almost impossible.

Functional Requirements

A commercial FPGA vending machine has to work with the Trojans inside it. It has to give us the results we expect or the Trojan is useless. The Trojan will be inserted using the Verilog HDL, which is something we all can manage as a team. Once the Trojan is inserted and we start working on the second aspect of the project, which is detection, we would already know what to look for in the code. This helps us in making the detection strategy because we know where and how the Trojan acts and what changes can be made in the system for the detection of it. The prevention methods are pretty standard we want to expand more on that as we progress in the project.

Risks

The biggest risk we have in this project is that we do not know if we would succeed in detecting the Trojan which is the hardest part of our project. The detection methods have been researched a lot but still it is very hard to detect Trojans. However, since our group is the ones putting the Trojans in the first place, we know what to look for in the code. We believe that will be able to find a way out of this.

Detection

Once a Trojan is detected it is easy to remove from the IC or not use that IC to avoid consequences. Research tells us that detection is easy post fabrication because after

fabrication it is very difficult to spot a Trojan since everything is small. Moreover, RTL design review and simulation are used to detect the Trojan.

There are several ways to detect the hardware Trojan. In our project we plan to use non- destructive method. We want to use side channel signal analysis for detecting the Trojan.

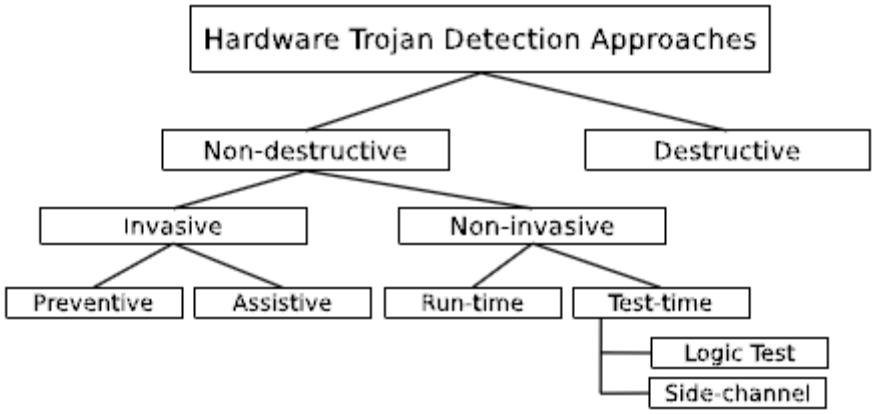


Figure 3: Hardware Trojan Detection Techniques: Chakraborty, Narismhan and Bhunia

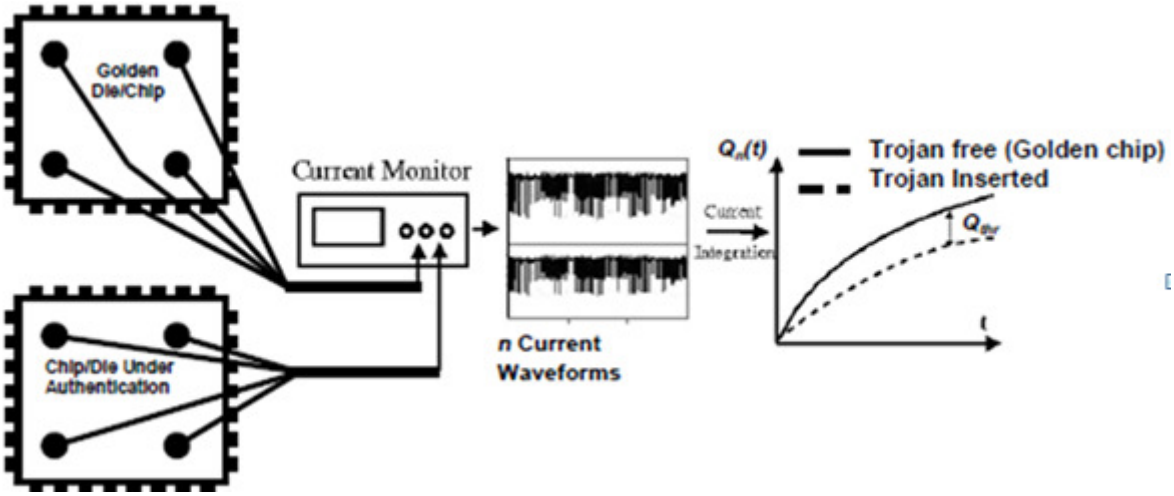


Figure 4: Current (charge) integration method

The diagram above shows that we can detect a Trojan by looking at the current increase/ decrease. The current waveform can tell us where the Trojan is and what is it doing to the system. For example if the die includes four power ports the golden die can be found by using exhaustive test for several randomly selected dies. A hardware

Trojan can be also detected by comparing the current integration method by comparing the results of all the patterns. If the same results occur within the range of variations then that die is considered Trojan free.

Trojans can also be detected by process of elimination which means that all the dies which are Trojan free can be eliminated and the ones which have variations in the current waveform can be tested again.

Prevention

Prevention from Trojans can be started from the very beginning which is the making if an IC. The different stages are:

1) Prevention at Design
2) Prevention at Fabrication
3) Prevention at Post Fabrication

Prevention at Design

Prevention at design can be done by avoiding the untrusted EDA tools or by having third party Intellectual Property (IP) in the design team. The way to avoid Trojans at design stage is by having all hardware resources on clock cycles. All the designs should require complete functionality of the IC. Using CAD tools to create a design and build it gives a check to the functionality and satisfaction that it meets the requirements.

Prevention at Fabrication

Prevention at Fabrication can be done by having a list of security related properties. If both the IP Consumer and Producer agree on those terms this makes the room for Trojans on fabrication level way less.

Prevention at Post Fabrication

When a reconfigurable logic is placed between the outputs of some ICs and outputs this disguises the design from attacker who has access to RTL. This leaves the attacker uncertain about how the IC works and therefore helps us to prevent the system from Trojans

Cost Considerations

The only cost we can think of at this point is the FPGA board on which we will put the Trojan and rest is done in Verilog which we have access to from the school computers. Once we know what kind of FPGA board we need we plan to ask Dr. Geiger and Lee to help us with it.

Project Schedule

Week	**Plan**
Jan 12- 17	Learning about the project in class and getting introduced to the material needed for the class
Jan 18- 24	Finalize the project and meet with advisors and team members
Jan 25-31	Research on hardware Trojans and find out as much information as we can on the topic in general
Feb 1-7	Research more on hardware and software side of Trojans and find connections of how it can be dangerous to the computer systems.
Feb 8-13	Read about the preventions and ways to detect hardware Trojans. Find creative ways of being cautious and defending the computer systems
Feb 15- 20	Narrowed down the Trojans and find out as a team which one we are able to make and prevent at the same time. If something like this is created it has to be rectified in some way
Feb 21- 28	Find ways to create hardware Trojan and find ways to detect it while making it
March 1-7	Design and test the hardware Trojan
March 8-14	Board layout

March 15-21	Finalize board layout and design
March 22-28	Complete testing and coding
March 29- April 4	Debugging and enhancing the circuit by making sure the Trojan is more than just a denial of service
April 5 -11	Things needed for fabrication
April 12-18	Having documentation ready which has to cite. Writing paper about what the project is
April 19-24	Working on prevention of the Trojans and preparing for the presentation
April 26- May 1	Dead week Preparing and giving the final presentation

Conclusion

Hardware Trojans are a continuing threat to our society. They compromise the security and functionality of our electronic systems. The threats can be as dangerous as deadly and as little as a denial of service. They are easy to activate and hard to catch. So it is important for us to think of something which would stop this from spreading and come up with some counter measures to avoid them.

References

Chakraborty, R. S., Narasimhan, S. & Bhunia, S. (2010) Hardware trojan: Threats and emerging solutions. http://www.trust-hub.org/resources/113.

Wang, X., Tehranipoor, M. & Plusquellic, J. (2008) Detecting malicious inclusions in secure hardware: Challenges and solutions, IEEE International Workshop on Hardware-Oriented Security and Trust 0, 15–19.

Tehranipoor, Koushanfar (2010) Rice University: A survey of Hardware Trojan taxonomy and Detection, IEEE CS and the IEEE CASS