TRUSTED SOFTWARE ENGINE AND PCB DESIGN FOR DATA CONSISTENCY CHECKING OF COMMERCIAL OFF-THE-SHELF (COTS) HARDWARE

by

Raymond Alexander DelVecchio

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Fall 2009

TRUSTED SOFTWARE ENGINE AND PCB DESIGN FOR DATA CONSISTENCY CHECKING OF COMMERCIAL OFF-THE-SHELF (COTS) HARDWARE

by

Raymond Alexander DelVecchio

Approved: _____
Fouad Kiamilev, Ph.D.
Academic Advisor in charge of thesis work

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Michael J. Chajes, Ph.D.
Dean of the College of Engineering

Approved: _____
Debra Hess Norris, M.S.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGEMENTS

First and foremost, thanks to my advisor, Dr. Fouad Kiamilev. I am positive he is one of the finest people to work with at the University because he accepts students from various backgrounds and encourages everyone to focus on their strengths and work on what they enjoy. In addition to being a great advisor, he is also is a great man who was always there to give advice to all the members of our research group, whether it was related to electrical engineering or life in general.

Thanks to all the members of CVORG for making the lab an enjoyable place to work. Special thanks to Ryan Hoover and Rodney McGee for the help with this particular project and with all other work along the way. I can't forget Nick Waite – his crazy and brilliant ideas were a driving force for nearly all CVORG projects and he was always around to explain theories and ideas in a practical and easy to understand manner.

To all my friends, even if you did not know the details of my journey you were always there to push me in the right direction and take my mind off of some of my struggles and I won't forget that.

Last, but certainly not least, thank you to my family. They have always supported me with whatever decisions I make in life, and without them I wouldn't be where I am today. I've been blessed with a great group of people to call my family and I'm grateful to have them in my life every day.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

- **AC** – Alternating Current
- **AES** – Advanced Encryption Standard
- **ASIC** – Application Specific Integrated Circuit
- **BGA** – Ball Grid Array
- **BRAM** – Block Random Access Memory
- **COTS** – Commercial Off-The-Shelf
- **DC** – Direct Current
- **DCE** – Data Communications Equipment
- **DDR SD RAM** – Double Data Rate Synchronous Dynamic RAM
- **DoS** – Denial of Service
- **DTE** – Data Terminal Equipment
- **EDK** – Embedded Development Kit
- **EM -** Electromagnetic
- **FIFO** – First In, First Out
- **FPGA** – Field Programmable Gate Array
- **FPU** – Floating Point Unit
- **FSL** – Fast Simplex Link
- **HDL** – Hardware Description Language
- **I/O** – Input/Output
- **IC** – Integrated Circuit
- **IP** – Intellectual Property
- **IR** – Infrared

- **ISE** – Integrated Software Environment
- **JTAG** – Joint Test Action Group
- **LCD** – Liquid Crystal Display
- **LED** – Light Emitting Diode
- **PCB** – Printed Circuit Board
- **PLB** – Processor Local Bus
- **PROM** – Programmable Read Only Memory
- **RAM** – Random Access Memory
- **RISC** – Reduced Instruction Set Computer
- **ROM** – Read Only Memory
- **RTL** – Register Transfer Level
- **SD** – Secure Digital
- **VGA** – Video Graphics Array
- **VHDL** – VHSIC Hardware Description Language
- **VHSIC** – Very High Speed Integrated Circuit
- **XPS** – Xilinx Platform Studio

LIST OF FIGURES

ABSTRACT

Recent trends in technology have pushed the majority of ASIC fabrication overseas. This high volume market leaves devices vulnerable to attack by adversaries who could potentially alter the design at the hardware level while at the foundry. This type of emerging threat, known as a hardware Trojan, can leave mission critical government or financial systems vulnerable to attacks that can lead to system failure.

For much of the previous decade, software was the main focus of computer security, but the past few years have ushered in a new wave of hardware security research to safeguard against such attacks.

This thesis provides insight into how hardware Trojans are classified, in addition to providing examples of exploits that can lead to sensitive information leakage in an encryption system.

A Trojan detection system is proposed for a COTS AES encryption component, which is accompanied by a modular stacked PCB design to implement such a system.

1 BACKGROUND

One of the major advancements in the field of electrical engineering was the shift from vacuum tubes and component transistors to integrated circuits. Since the inception of the IC, electrical systems have exponentially improved in speed, size, power consumption and complexity. This has led to globalization and mass manufacturing techniques, which have effectively driven down the price of fabrication in addition to shortening the design cycle [1].

Today, many companies choose to outsource the fabrication of their ASIC designs to foundries outside of the United States due to the lower associated costs. While this is desirable for saving on expenses, it may introduce unwanted security concerns that can compromise the intended functionality [2].

In general, there are three main types of security concerns when discussing an embedded system: confidentiality, integrity and availability [3]. Confidentiality deals with keeping illicit users from acquiring secure or sensitive data that is stored within the system. Data integrity means that secure data is maintained and not distorted or removed, so as not to lose information that is essential to the functionality of the system. From the definition of each, it can be noted that confidentiality is usually about protecting a system from unauthorized reading and data integrity measures should safeguard against unauthorized writing. The last security concern is availability, which means that an authorized user should be able to access the system promptly if necessary. The main focus behind this concern is to circumvent a DoS attack.

Even if the system is completed with the aforementioned security measures in mind, the design still must be submitted to the foundry for fabrication. Due to this, one must be suspicious of sets of people on the other end who could potentially be adversaries willing to tamper with the original design. With the source files in hand, they can alter components of the design or add malicious circuitry to the IC mask that will affect the system under specific and rare conditions. This type of malicious alteration is known as a hardware Trojan, and it can pose a serious threat to secure military or financial systems, among other commercial systems and consumer products.

The dangerous aspect of this threat is the stealthy nature of the attack. It does not take much of an alteration to inflict serious damage to a complex electrical system. A Trojan that is confined to a small area of a chip can easily go undetected among millions or billions of transistors [4]. Once activated, the Trojan can inflict damage at a number of levels, from completely disabling system operation, to leaking confidential information, or even altering vital computations that are necessary for security purposes [2]. The type of attack is completely up to the adversary and is only limited by their creativity and imagination, which makes defending such an attack a tricky proposition.

Hardware Trojans are difficult to test for after fabrication since they will usually lay dormant until they are activated by a statistically improbable condition that standard input test vectors may not cover. They may also be triggered by a timer or specific counter value, known as a time bomb Trojan, which cannot be tested simply by driving the inputs and observing the outputs.

Reverse engineering or destructive testing could be used to check for design inconsistencies, but this is complicated and costly. Furthermore, it may be the case that only a few chips in each batch have been tampered with, making testing less feasible and more burdensome.

Trusted foundries within the United States exist, but the expense of custom designing and fabricating each chip in an elaborate system would be exorbitant. Generally the types of projects that use such costly methods of fabrication are funded by government subsidies where mission critical and military applications must have the highest level of security [2].

Given that hardware is the foundation upon which software is built, a security breach at the hardware layer renders software protection ineffective. With this type of emerging threat, more focus is being shifted towards hardware security. As research progresses, more methods of examining and detecting hardware Trojans will come about to maintain security throughout the design process. It is important to keep in mind, though, that the cycle of security is never ending, as attackers will always search for clever ways to circumvent secure channels and defenses.

2 HARDWARE TROJANS

Before determining a method or procedure to safeguard against malicious attacks, it is first necessary to take a more detailed look at how such hardware Trojans can be classified. In general, hardware Trojans are distinguished from each other based on three characteristics: physical, activation and action [5]. These will be discussed further and a depiction of the characteristic flow chart can be seen below in Figure 2-1.

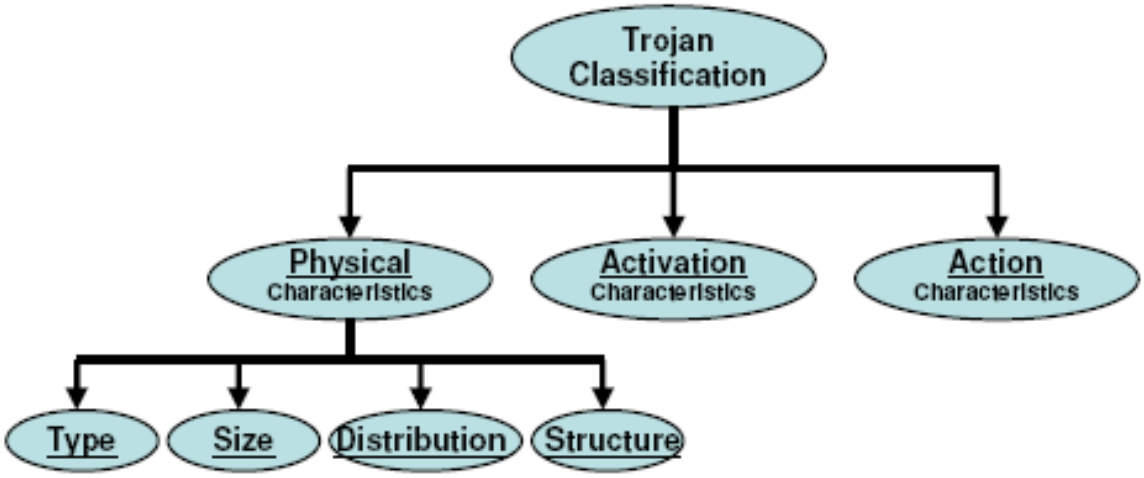

Figure 2-1 – Classification of Hardware Trojan Characteristics [5]

2.1 Physical Characteristics

There are four main physical characteristics to take into account when classifying hardware Trojans, including type, size, distribution and structure. A visual representation of the physical characteristics is shown below in Figure 2-2.

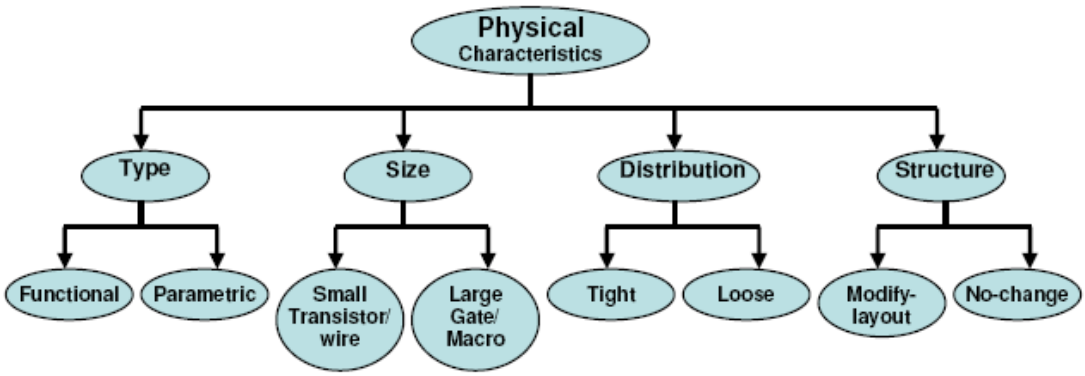

Figure 2-2 –Physical Characteristics of Hardware Trojans [5]

The type category can be divided into two main classes – functional and parametric. A functional type Trojan is one where the adversary adds extra circuitry or logic to an existing design, while a parametric type Trojan is one where the parameters of the design are altered to decrease the expected reliability or functionality of the original design. An example of a parametric Trojan is a wire that is fabricated thinner than usual such that over time the metal connection is degraded and signal integrity is compromised.

The size of the Trojan relates to how large the area of alteration is, whether this change is functional or parametric. Some Trojans only require a few transistors in a single location, whereas others are more complex.

The distribution of the Trojan accounts for where the Trojan resides in the physical layout. If several transistors were added, but placed evenly across the design, this would be known as a loose distribution. On the other hand, if all of the changes were made in a confined layout area, the distribution would be considered tight.

Lastly, the structure class is used to define whether the physical layout of the original design must be regenerated in order to fit the Trojan. For parametric Trojans, it may be easier to keep the original IC dimensions the same, since this type of attack generally involves existing design components. Alternatively, a functional Trojan that has a larger number of transistors may affect the layout of the rest of the design. This is not desirable since the change in layout can also change power and delay characteristics, thus increasing the chances of detection. When implementing this type of hardware Trojan in an attack, the size, distribution and structure are all key factors to the adversary.

Examples of the physical characteristics of hardware Trojans are demonstrated in Figure 2-3.

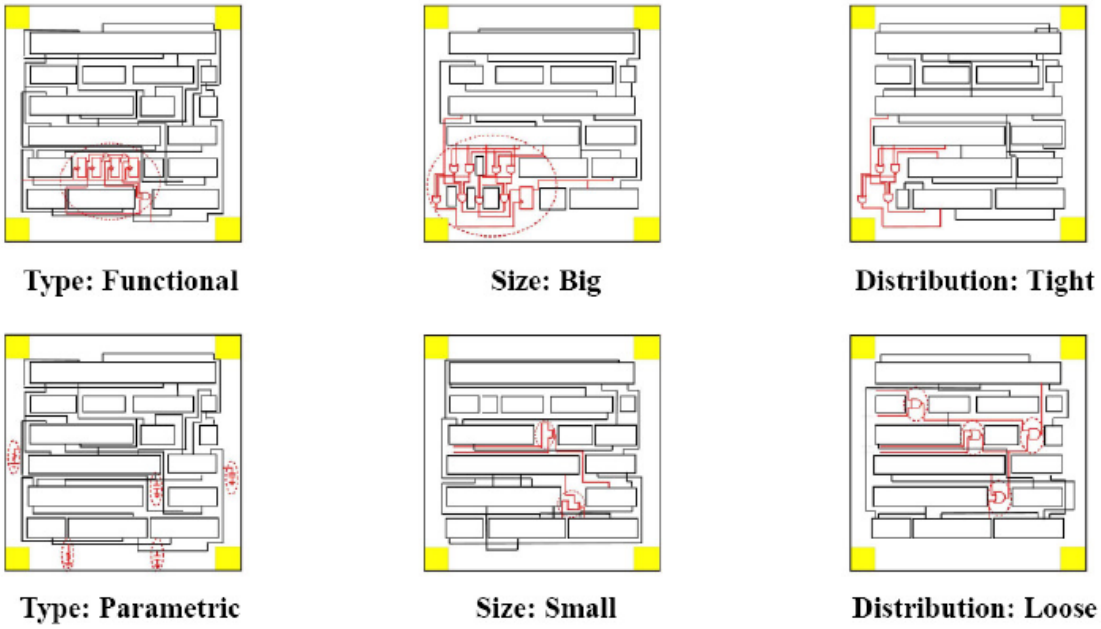

Figure 2-3 – Examples of Trojans with Various Physical Characteristics [5]

2.2 Activation Characteristics

The activation characteristics describe the condition or set of conditions which will trigger the illicit function associated with the hardware Trojan. Because the adversary wants to be as stealthy as possible, it is important to ensure the Trojan is not activated during IC testing or normal user operation. Due to this, the activation sequence of the Trojan is chosen carefully to be statistically improbable.

This class can be broken down into two main subclasses: externally activated and internally activated. Figure 2-4 below shows the activation characteristics.

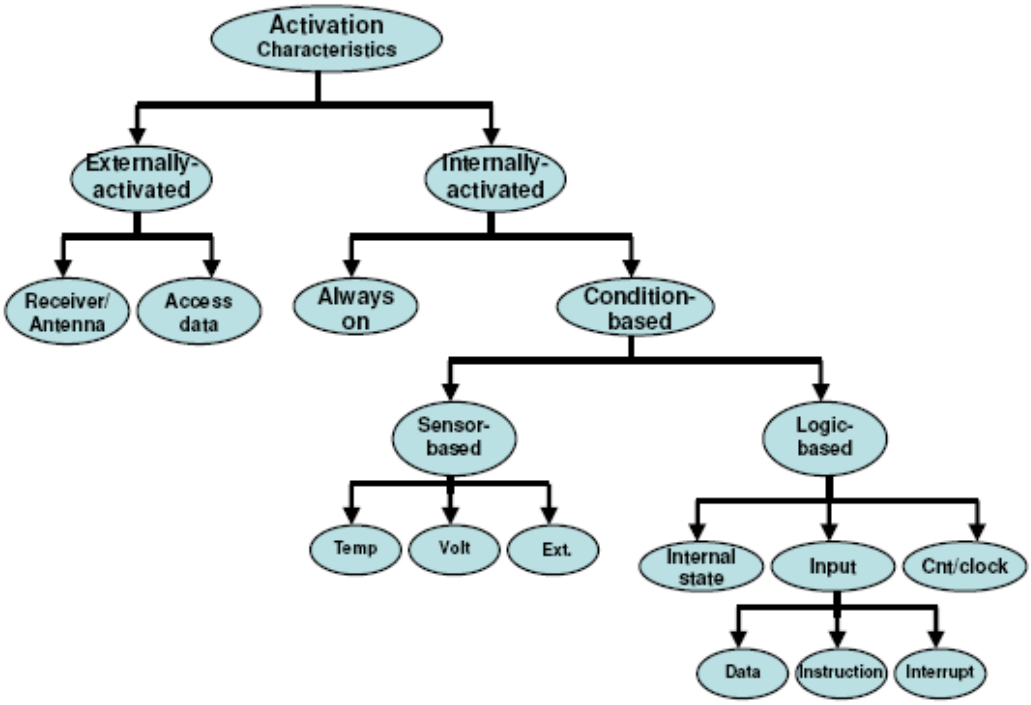

Figure 2-4 – Activation Characteristics of Hardware Trojans [5]

An externally activated Trojan is one where the adversary can activate the Trojan at the time of their choosing. This implies that the adversary has a covert way to access the system, whether it is through a transmitter-receiver antenna pair, or backdoor access to a networked system.

An internally activated Trojan can be further divided into two subcategories: always-on and condition-based.

The best example of an always-on internally activated Trojan is a parametric type alteration. A wire or transistor whose parameters were changed does not need a specific trigger; instead the trigger lies in the fact that the geometry of the component has changed and may cause havoc if it does not behave as expected.

A condition-based Trojan is one where a specific condition must be met to be successfully activated. This condition could be met from a certain input sequence, a counter value, a sensor reading, etc. To implement such an attack, a functional type Trojan is needed due to the extra logic to check for the triggering condition.

2.3 Action Characteristics

Once the Trojan has been activated, it can then perform its intended malicious action. These actions can be grouped into three broad categories: modify-specification, modify-function and transmit information. These are displayed below in Figure 2-5.

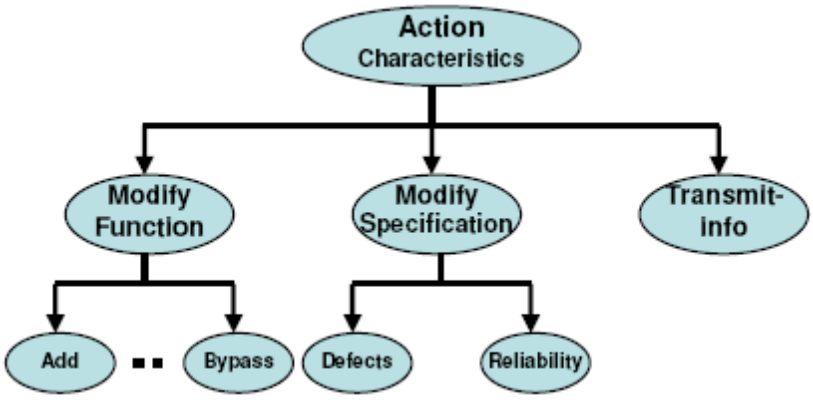

Figure 2-5 – Action Characteristics of Hardware Trojans [5]

The modify-specification class is related to parametric type Trojans. As stated before, parametric Trojans alter the geometry of a transistor or wire in an existing design. These could be considered the stealthiest since the number of gates is not

changed, but the action capabilities are inherently limited. Most modify-specification Trojans will end up causing irregular behavior or even system failure.

A modify-function action is one where extra logic or bypassing logic can be used to alter the intended functionality of the system. These types of Trojans can seemingly inflict the most damage to the system due to the endless attack possibilities, but they tend to be the riskiest in the eyes of the adversary, since the extra logic can be easier to detect than a parametric Trojan.

A transmit information Trojan is self explanatory. This type of alteration would generally be used in a system that stores sensitive information, such as encryption keys or passwords. Once activated, the adversary can gather this information and use it in any way they choose.

3 TESTING PLATFORM

3.1 Trojan Attacks

- Spartan-3E Starter Kit – FPGA – XC3S500E-FG320
- AES Encryption & Decryption Core – Written in Verilog
- Optical Voice Link Kit from Industrial Fiber Optics
- Radio receiver tuned to approximately 50 MHz
- Fluke Ti50 Thermal Imager

AES encryption was chosen as a baseline application for further experiments. Open source hardware implementations of the 128-bit AES encryption and decryption cores were obtained from [6].

The hardware Trojans that were added to the base AES designs were implemented on a Spartan-3E Starter Kit board. This development board provides several easy to use hardware peripherals that make it an ideal choice for a testing platform. The starter board houses a 320-pin BGA packaged FPGA. This can be programmed using a standard HDL or with the C programming language if a MicroBlaze soft processor core is implemented.

The main memory components of the board include a 4 Mbit Platform Flash PROM for storing the FPGA bit configuration, and 64 MB of DDR SDRAM for program memory.

Some of the peripherals on the board include: a PS/2 mouse and keyboard connector for user input, a 2x16 LCD screen for user output, a VGA port to interface with a monitor or other display device, an Ethernet port for network connectivity, one

DCE and one DTE RS-232 connector for serial communication, and several expansion connectors for board to board communication [7]. The Spartan-3E starter board can be seen below in Figure 3-1.

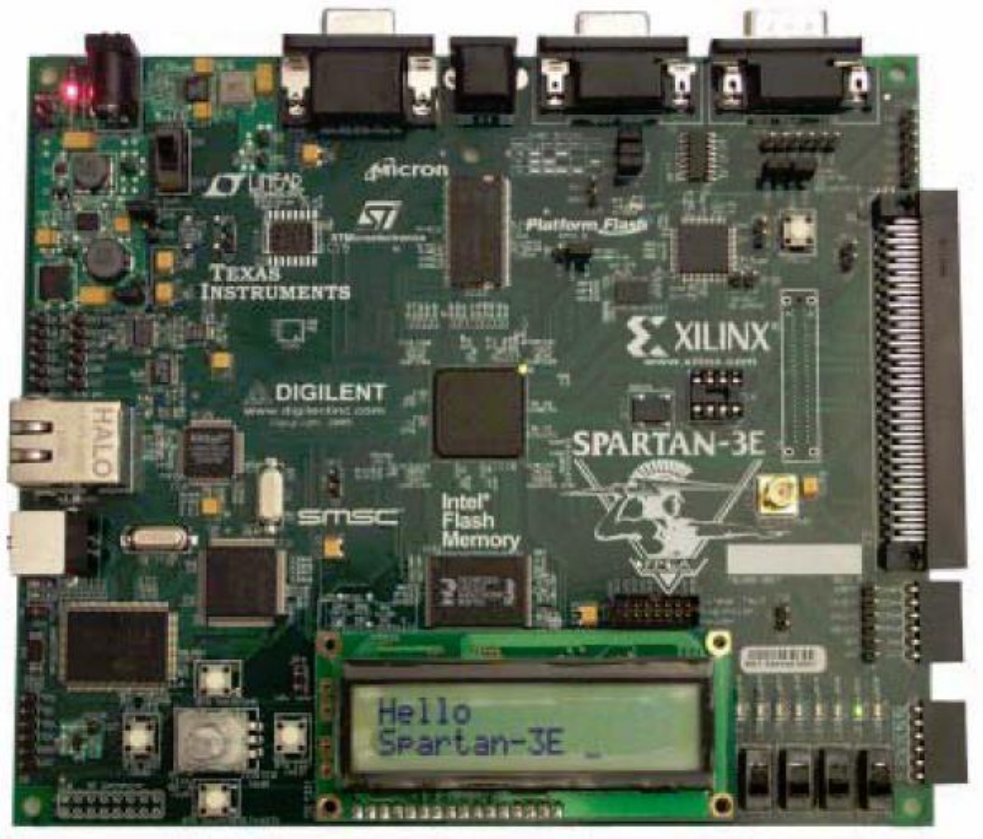

Figure 3-1 – Spartan-3E Starter Kit Board

An Optical Voice Link kit, as shown in Figure 3-2, was chosen to convert light information to audio information, so that the resulting signal could be played through a speaker. The details of the optical hardware Trojan will be discussed further in the following section.

A simple radio receiver that can be tuned to 50 MHz is needed to obtain the EM signal radiating from the board. The EM Trojan will be explained in the next section.

Figure 3-2 – Optical Voice Link Kit by Industrial Fiber Optics

A Fluke thermal imager was used as part of a demonstration of thermal Trojans. The thermal imager uses the IR radiation of objects within its field to provide a high resolution representation of the heat signature seen by the camera. For the thermal Trojans, the Fluke was used to detect the fluctuation of heat from various components on the board. The sources of heat were used as a method to covertly and cleverly leak information out of the board, which will be discussed in the following section.

Figure 3-3 – Fluke Ti50 IR Thermal Imager

3.2 Trojan Detection

- Spartan-3E Starter Kit – FPGA – XCS1600E-FG320
- Custom FX2 Connectors
- MicroBlaze Soft 32-bit RISC Processor
- AES Encryption & Decryption Core
    - Untrusted Board - Written in Verilog, Simulates COTS AES Encryption Chip
    - Trusted Board – Written in C, MicroBlaze Software Checking Engine

Two Spartan-3E Starter Kit boards were needed to implement the Trojan detection system. The Hirose 100-pin FX2 expansion header was used as the interface between the two boards via a custom FX2 connector. A picture of the expansion header can be seen below in Figure 3-4.

The FX2 connector has 43 I/O pins for communication, with many of the 100 pins grounded to allow for high speed signaling and greater signal integrity. In addition, two voltages are provided to the connector: the main 5V power supply and either 2.5V or 3.3V, which is selected through the use of a jumper.

For this system, one Spartan-3E board will act as a COTS chip that only houses an encryption and decryption core, while the other board contains a security module. It was determined that for rapid development, the MicroBlaze soft processor core could be used to expedite program development. In order to perform security checking against the hardware encryption results, a software implementation of AES encryption found in [8] was used.

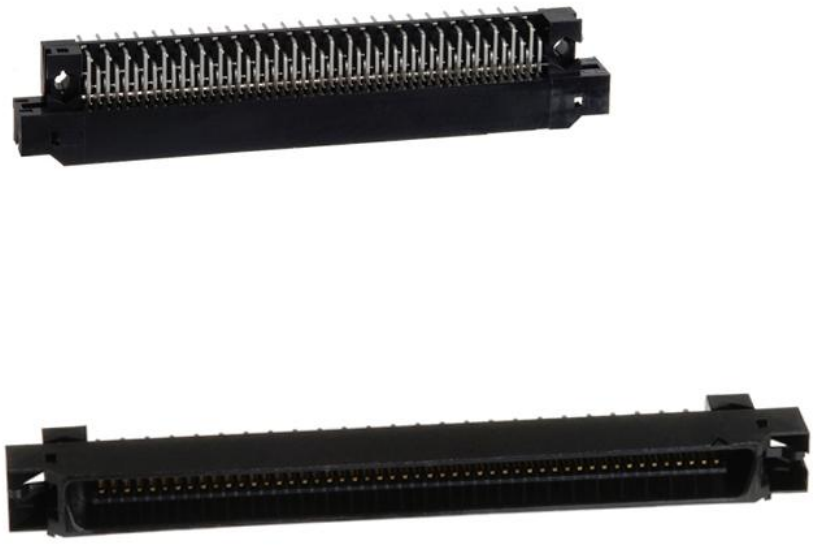

Figure 3-4 – FX2 Right Angle Connector

4 ATTACKS & EXPLOITS

The first objective throughout the course of this research was to develop examples of hardware Trojans that were not only lightweight, but also stealthy in the way they leaked sensitive information. As such, the main focus for these attacks was the action characteristics of the Trojans.

With the Spartan-3E FPGA development board running an AES-128 encryption application, several modifications were made to the VHDL design. Knowing that the security of the application is dependent upon the secrecy of the key, the objective was to find clever and secretive ways to leak the key out of the board. It is important to note that while triggering is an issue when inserting a hardware Trojan, the malicious operations were deemed far more important. Because of this, the triggering methods will not be discussed in detail.

4.1 Optical Hardware Trojan

The first Trojan attack uses a spare LED on the development board to leak the key. In order to maintain secrecy of this attack, a protocol for transmitting the key was set. Each bit of the key would be transmitted for a half second interval, where a bit ‘0’ is represented by an LED switching rate of 2 kHz and a bit ‘1’ is represented by an LED switching rate of 4 kHz. Using the receiver from the Optical Voice Link Kit by Industrial Fiber Optics, the transmitted light information is converted to an electrical signal and played through a speaker, as shown in Figure 4-1. With two high

switching rates, the LED appears to be solidly lit to the user, but the tone differences are clearly heard when played through the speaker.

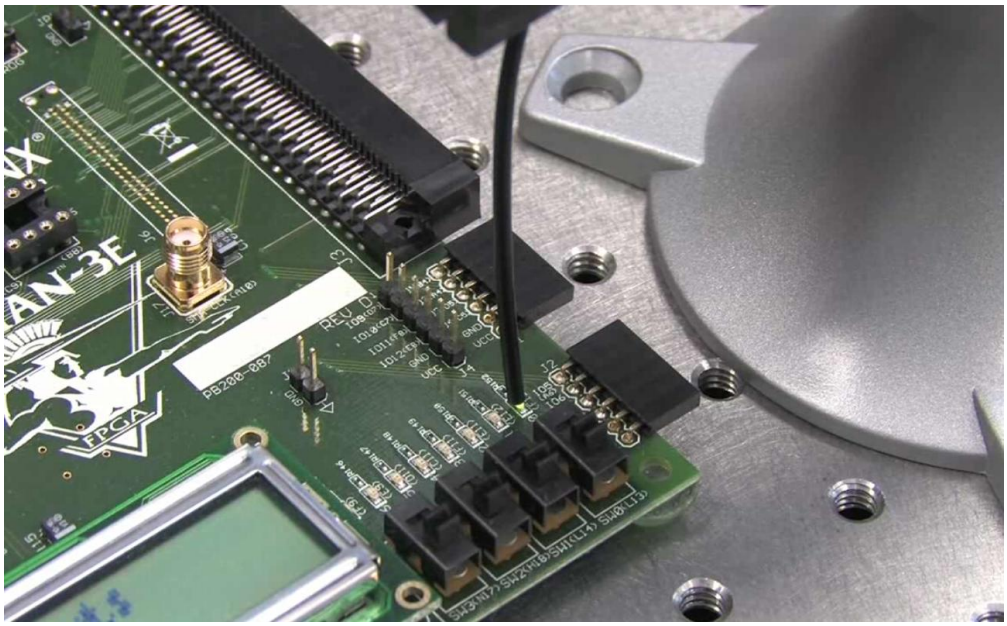

Figure 4-1 – Optical Trojan Demonstration

4.2 EM Hardware Trojan

The second attack focuses on transmitting the key using electromagnetic waves. The protocol for transmitting the key is similar to the optical Trojan, where each bit is transmitted for a half second interval. Now, though, the key is mapped to an unused output pin where an external wire can be attached to act as an antenna. A bit '1' is represented by switching the output pin at a rate of 50 MHz, and a bit '0' is represented by a constant voltage on the output pin. Using a radio receiver tuned near 50 MHz, the key transmission can be heard once the Trojan has been triggered. The captured signal is shown below in Figure 4-2.

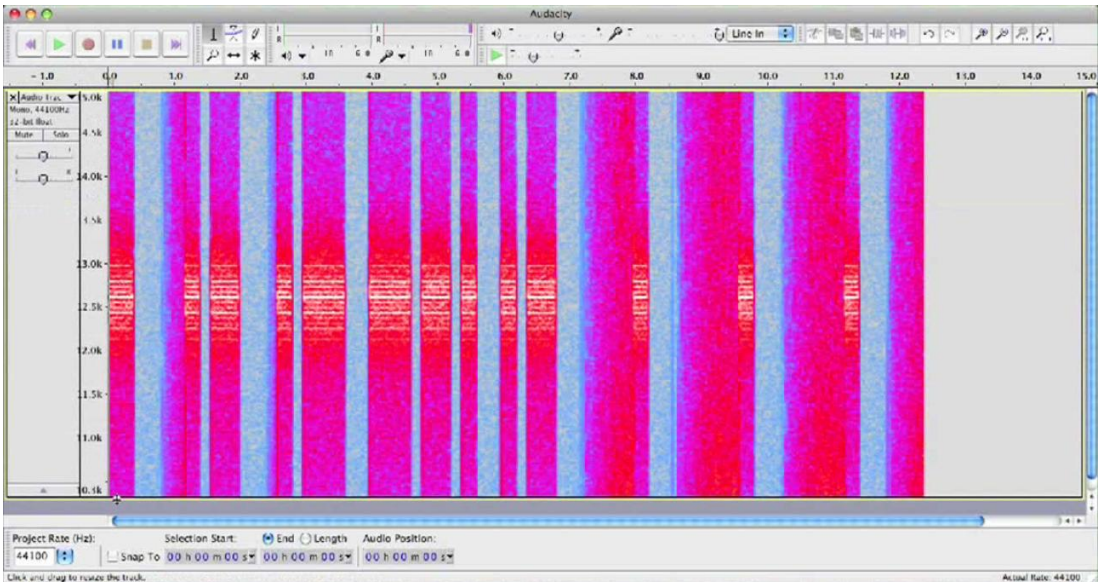

Figure 4-2 – Captured EM Signal from Radio Receiver in Audacity

4.3 Thermal Hardware Trojan

The last two attacks focus on leaking the key thermally. For the first attack, an unused resistor on the board was targeted. For transmitting a bit '1', current is flowing through the resistor and therefore the resistor is dissipating heat for that half second interval. When transmitting a bit '0', there is no current flowing through the resistor and hence no heat is dissipated. These thermal patterns can be observed by using an IR camera and the key can then be extracted. A picture of the thermal camera detecting a heated resistor is displayed below in Figure 4-3.

Figure 4-3 – Resistor Dissipating Heat Captured by the Thermal Imager

Our second thermal attack uses the FPGA itself as the leakage medium. Instead of controlling an external component or pin like the previous attacks, this Trojan implements several dummy registers internal to the FPGA which are switched at a rate of 50 MHz to represent a bit '1', and set to zero for a bit '0'. When transmitting the key, the heat emitted from the switching activity within the internal circuitry is clearly visible with the IR camera. Figure 4-4 below shows the FPGA when saturated with operations.

Figure 4-4 – FPGA Dissipating Heat Captured by the Thermal Imager

5 TROJAN DETECTION SYSTEM

The Trojan detection system demonstrated here follows a trusted hardware design methodology. Because it is difficult to test a COTS IC for added Trojan hardware, an extra hardware layer is introduced between the commercial chip and the user board to act as an intermediary for communication.

Through the perspective of the user, the trusted layer is transparent. It monitors the data to and from the user board, and if it detects any suspicious behavior, operation is halted. This layer also stores any sensitive data, such as encryption keys.

For the purposes of this demonstration, the suspicious behavior was defined as information leakage through the AES encryption or decryption text pins. The trusted module runs a software version of AES encryption and decryption, and compares the results against the hardware results. If these results do not match, the determination is that the AES hardware chip has been compromised.

All three layers are simulated using two Spartan-3E starter kit boards. The first board incorporates the user module and trusted module, while the second board implements the untrusted AES core. The two boards communicate with each other over a serial interface, through a custom FX2 connector. The system setup is shown in Figure 5-1, while a block diagram of this design can be seen below in Figure 5-2.

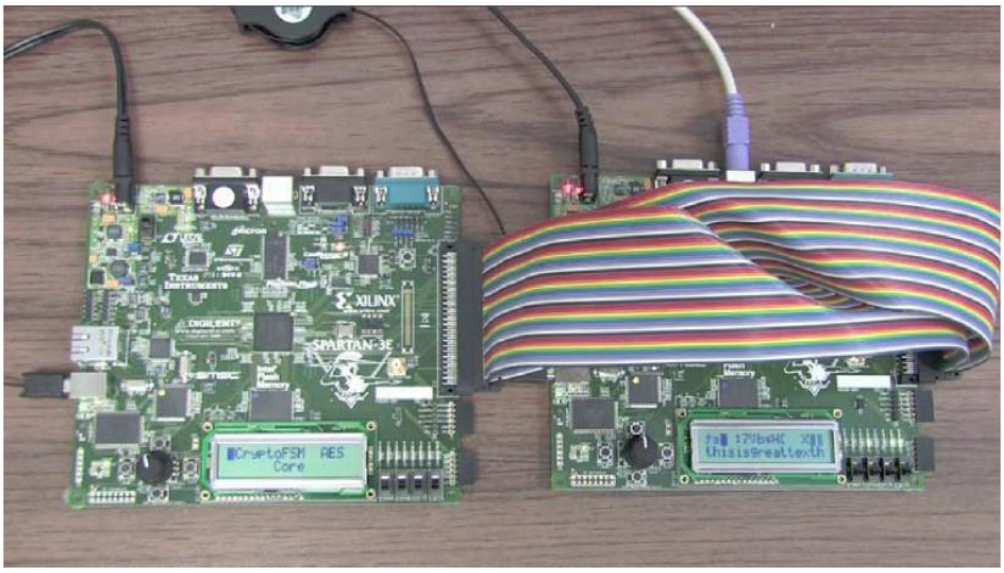

Figure 5-1 – Trojan Detection System Setup Using Spartan-3E Starter Boards

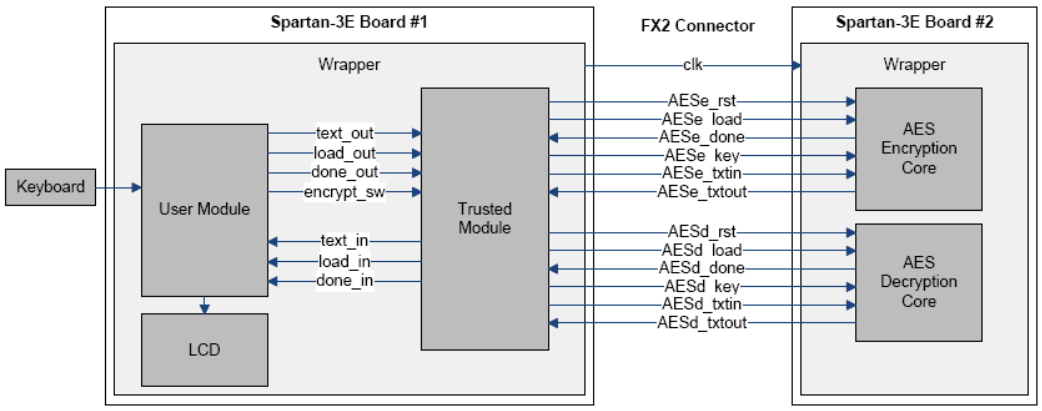

Figure 5-2 – Block Diagram of Trojan Detection System

5.1 MicroBlaze Processor

The design of this system was based heavily around the MicroBlaze 32-bit RISC soft processor core. This processor was ideal for a number of reasons.

Xilinx Platform Studio (XPS) software allows the user to build a hardware system through the Base System Builder Wizard. This makes the hardware design process effortless, abstracting away minor details that could become a source of human error. Through the wizard, the user may select a number of different options, such as the size of local memory (BRAM), clock speed, cache, FPU, along with adding several pre-defined peripherals.

Because the MicroBlaze processor is implemented completely using RTL on an FPGA, the architecture of the processor can be reconfigured while in the field. This distinctive ability to update the configuration without physically replacing the IC is what makes FPGA technology promising for future hardware and software design.

Once the hardware system is in place, the MicroBlaze core promotes rapid development through the ease of writing software in the C programming language.

What makes it even more powerful is the ability to add custom hardware peripherals. By attaching a hardware peripheral to a bus, the user can send data to the peripheral through a software function call, and let the hardware perform the computationally intensive tasks. In some applications, this can result in a massive performance increase when compared to using a microprocessor alone.

The two main buses which can be used to interface with hardware peripherals are the Processor Local Bus (PLB) or the Fast Simplex Link (FSL) bus. Most third-party IP cores will connect to the PLB, but the FSL bus allows for a lower latency connection to the MicroBlaze core. The FSL bus is a FIFO-based communication

channel that provides high speed access to the MicroBlaze core; up to 16 master and 16 slave devices can be attached to the bus.

5.2 User Module

The user module is written in VHDL and is connected to the MicroBlaze core through the FSL bus. This peripheral houses a controller for keyboard input and LCD output, which are supported on the Spartan-3E Starter Board.

The user module prompts the user to type in a string to be encrypted via the keyboard, which is displayed on the LCD. Once the data is buffered in the user module, it is sent to the MicroBlaze core through an FSL bus transfer.

After the user string is sent for encryption, the user module will wait for the results and display the encrypted text on the LCD.

5.3 Trusted Module

The trusted module consists of the MicroBlaze software application, and a VHDL hardware peripheral that acts as a parallel to serial converter.

As mentioned previously, the keyboard input is gathered by the user module and then transferred to the MicroBlaze core by an FSL get function. From this point, the input string is written to the FSL bus to the trusted peripheral to be serialized and sent to the second board which contains the AES core. At this time, the same data string is encrypted using a software version of AES written in C.

Once the encryption process is completed, the resulting data is sent back to the trusted module of the first board, where the hardware encryption results are checked against the software encryption results. If these strings match, then the system is operating correctly; otherwise the system is stopped due to the data inconsistency which indicates that the system has been compromised or attacked.

A block diagram of the system created using Xilinx Platform Studio is displayed below in Figure 5-3.

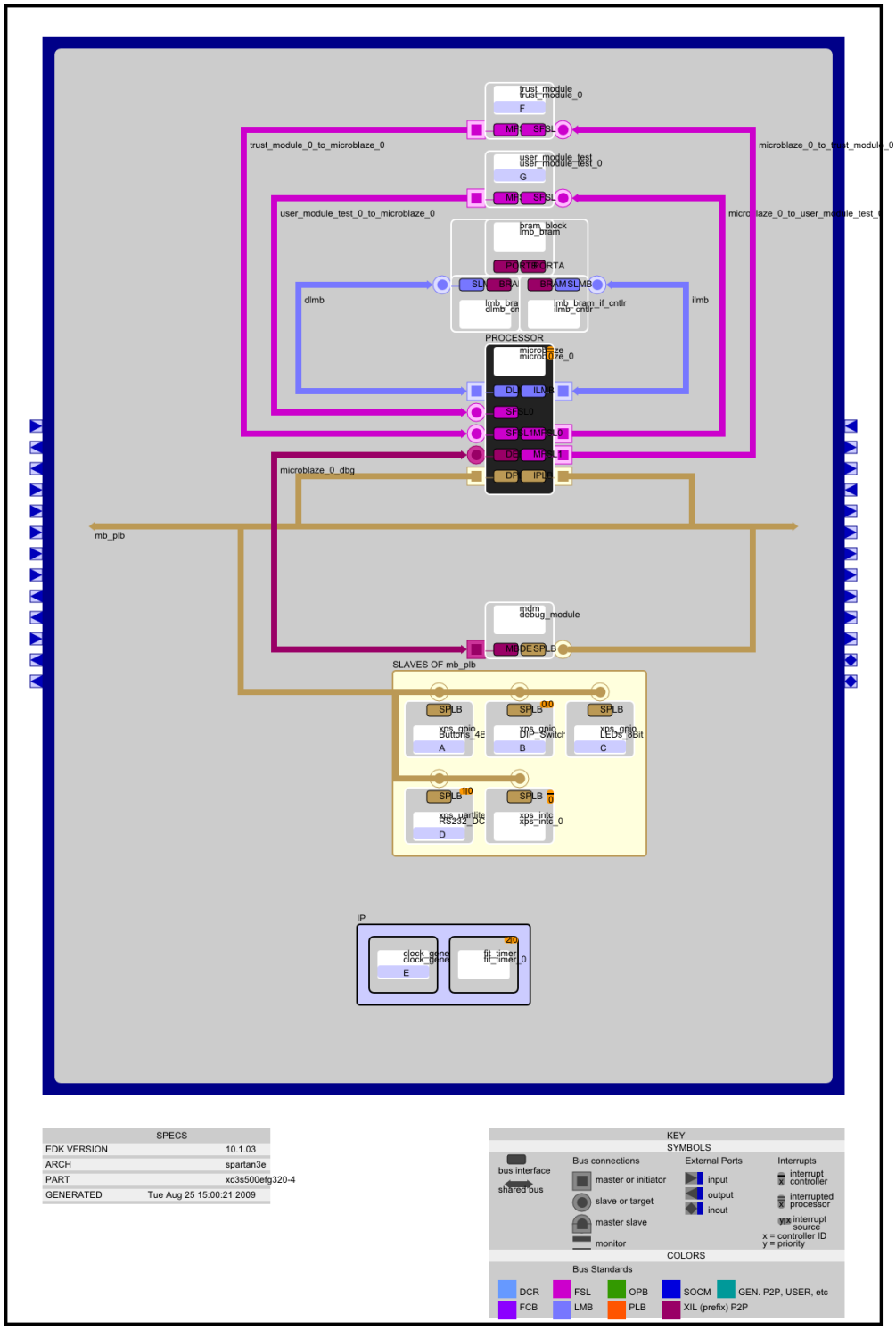

Figure 5-3 – Block Diagram of MicroBlaze System

6 PRINTED CIRCUIT BOARD DESIGN

To be consistent with the Trojan detection system concept, the goal was to design a modular and stackable printed circuit board that could be programmed to behave as any of the three layers shown in Figure 6-1 below.

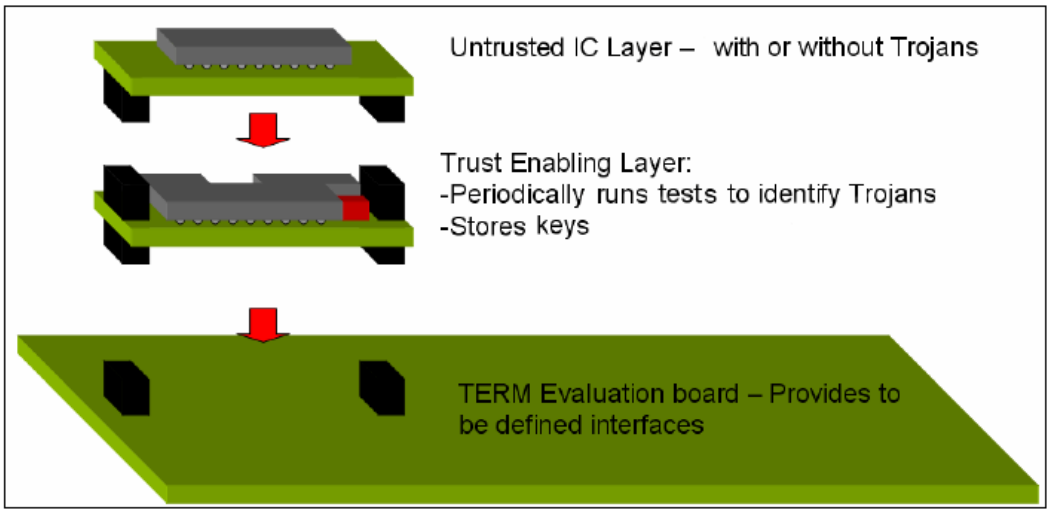

Figure 6-1 – Trust Enabling Functionality for a Stacked PCB

Since the Trojan detection system was built on the Spartan-3E Starter Kit board, it was logical to use the schematics of that design as a base for the custom PCB. Using this methodology, it was possible to design a very complex system in a fraction of the time.

The program that was selected for the custom layout was Cadence Allegro PCB design package. This software allows for an organized and robust design flow, from creating symbols for each part, to schematic design and physical layout.

Xilinx Platform Studio and PlanAhead were also used in conjunction to define and choose the FPGA pins for all peripheral components. Using these two programs together drastically reduced the chances of human error while dealing with all connections to a 208-pin FPGA.

6.1 Component Selection

The Spartan-3E Starter Kit board uses a 320-pin BGA FPGA package. This type of footprint requires special soldering techniques that were not practical for the purposes of this project. Due to this, a 208-pin surface mount package was chosen as a substitute. This FPGA provides a maximum of 158 I/O pins to implement several standard peripherals as well as custom parts. The FPGA is driven by a 50 MHz oscillator. A diagram of the XC3S500E-5PQ208C footprint can be seen below in Figure 6-2.

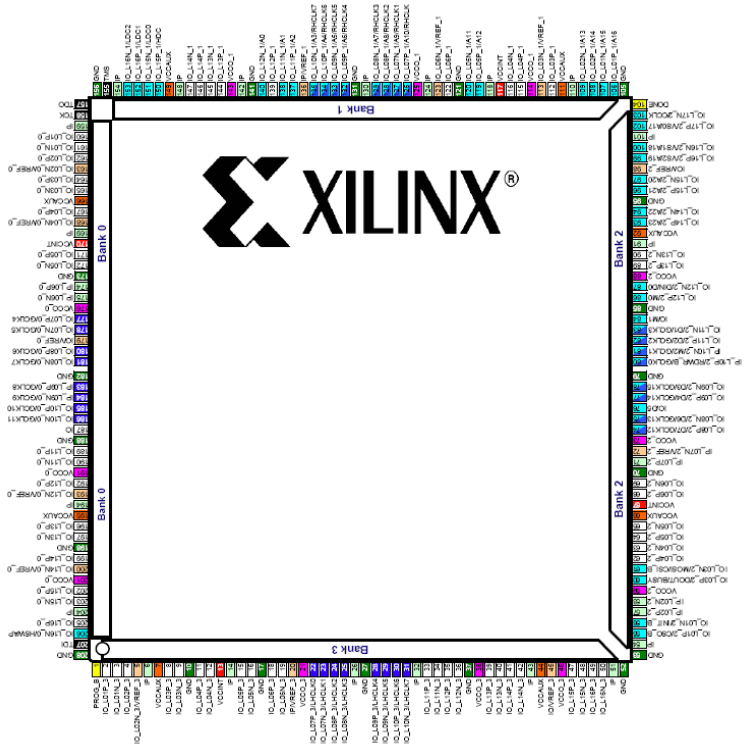

Figure 6-2 – PQ208 FPGA Footprint

Power is supplied to the board through an AC-DC wall transformer, with an output of 5V DC. From this power source, three Texas Instruments PTH04070W adjustable switching regulators are used to deliver the necessary voltages to the individual board components and FPGA. The output of each regulator is determined by an external resistance, and the three output voltages needed to operate the board are 3.3V, 2.5V and 1.2V.

The PCB houses three types of memory: PROM, DDR SDRAM and a Secure Digital (SD) card slot. The main use for the Xilinx Platform Flash PROM is to store the FPGA bit configuration file. Without this type of read-only memory, the FPGA would have to be manually configured each time the board was turned on. The PROM

allows the FPGA to be automatically configured and ready to use without the need to program each time the system is powered.

The Micron Technology DDR SDRAM provides 512MB of external memory. The amount of internal program memory is inherently limited when using the FPGA as a soft processor. This allows more than sufficient memory to write code without worry of exceeding the space limitations.

Adding the SD card slot increases the flexibility and potential application of the system. The main motivation for including an SD card was to be able to run Linux within the embedded system. For the purposes of this project, running Linux was not necessary, but the physical connections were simple enough to add for possible future use.

To support communication between the PCB and a computer serial port, a ST3232B serial transceiver is needed. This low power chip complies with the RS-232 standard output voltage levels.

In order to implement modular stacking capability, FX2CA2-100 straight plug and receptacle connectors are placed on opposite ends of the board; one on the top and one on the bottom as shown in Figure 6-3 below. With this setup, every board added to the stack must be rotated 90º to match its accepting connector. To reduce any stress from stacking, rubber feet could be added to the end of the board such that they match the FX2 connector height.

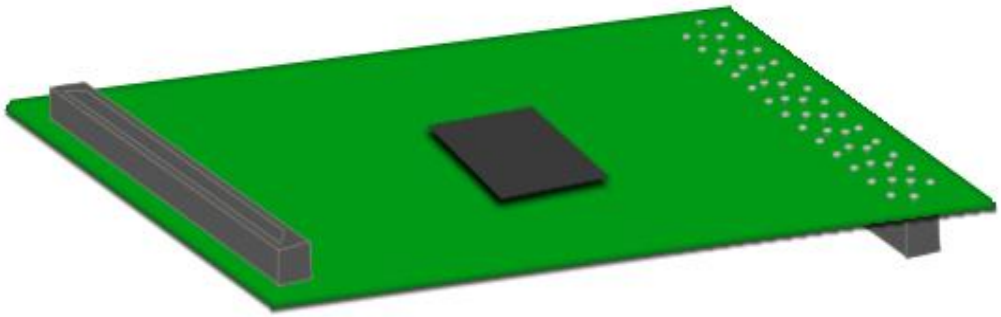

Figure 6-3 – Idea for Modular Stacked PCB

To program the board, a JTAG header was incorporated in the design. Both the FPGA and PROM are daisy-chained in the JTAG programming chain, making either chip capable of accepting the compiled bitstream. The serial JTAG interface is comprised of four ports per chip, providing an easy and lightweight mechanism to access multiple ICs on the PCB. A diagram depicting the connection of JTAG devices can be seen below in Figure 6-4.

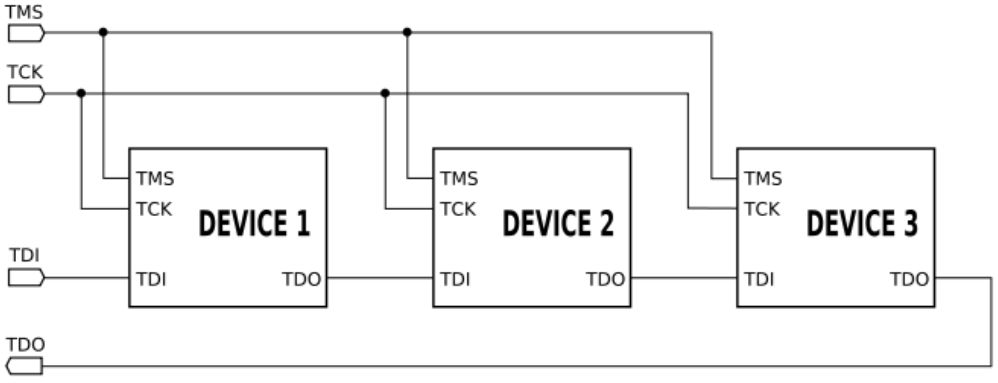

Figure 6-4 – Block Diagram of JTAG Device Chain [9]

The remaining ports were dedicated for user I/O, such as switches, buttons and LEDs. These make user interaction much easier, along with providing a simple way to test and debug programs after they are downloaded to the board.

6.2 Design Considerations

Throughout the design process, there were several important factors to consider when laying out the PCB. Ultimately the decision was made to fabricate a four-layer board, so the first major deliberation was how to arrange the components and assign voltage levels to the power planes.

The top and bottom layers of the board were reserved for the routing of copper traces, while the two inner layers were power planes. As stated previously, there are three main voltages supplied to various components on the board: 1.2V, 2.5V and 3.3V. With one of the power planes reserved for ground, the remaining power plane had to be assigned to one of the three aforementioned voltages.

Given that the FX2 connector I/O could operate at either 2.5V or 3.3V, it was necessary to look at the other connections to make a final determination of the power plane voltage. The IC that uses the most FPGA connections is the DDR SDRAM, which operates at 2.5V. Based on this, a decision was made to set the FX2 operating voltage to 2.5V and assign the power plane to this same voltage. With so many connections at 2.5V, it becomes advantageous to reserve the power plane as opposed to routing each signal from the voltage regulator output.

The remaining power signals must be routed to different locations on the board. To provide a wider path for current to flow, all power traces were set to be larger than the standard signal traces. The traces from the 5V power jack input were made to be 40 mils, as they only need to travel to the three voltage regulators. The last two power

traces for the 1.2V and 3.3V lines were expanded to 12 mils to match the width of the IC pins to which they were connecting. All other standard signal traces were kept at the default value of 6 mils.

The final layout of the PCB can be seen below in Figure 6-5 with all the major components highlighted. The bottom of the board was left open for routing and extra resistors and capacitors.

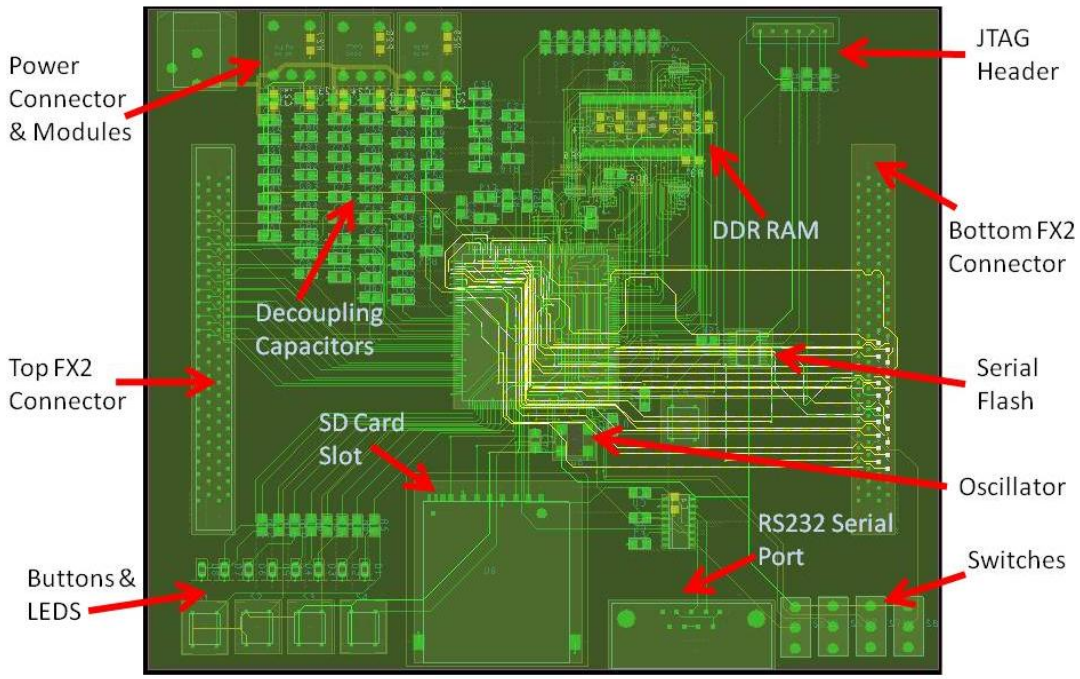

Figure 6-5 – Final PCB Layout

7 CONCLUSIONS

Hardware Trojans were defined and classified, and several examples of how these alterations can potentially inflict damage to a system were displayed. The leakage mediums for these attacks were chosen to be covert and clever, to show how a bit of creativity from an adversary can make a secure system vulnerable.

This work provides one possible solution to check for hardware that has been compromised. With the abundance of attack possibilities, though, further research is necessary to find a generalized solution to multiple types of attacks. However, the multiple layer method demonstrated in this work shows promise, as a physical trusted layer separates the COTS chip with the user board.

A modular stacked PCB design was proposed to implement a layered Trojan detection system. The design was based off the Spartan-3E starter board schematics to accelerate the design cycle.

APPENDIX A – PCB Design Files

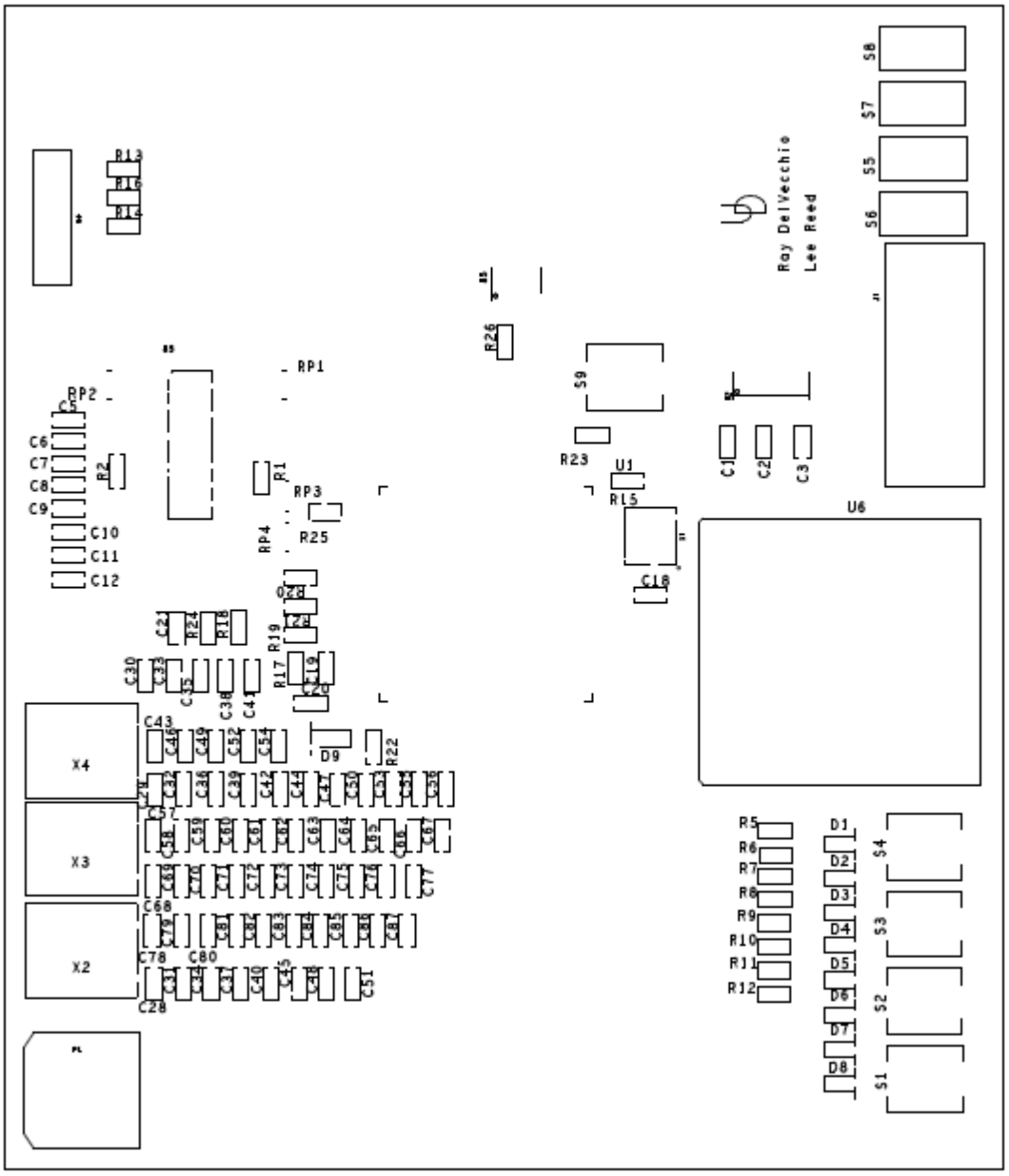

Figure 0-1 – PCB Top Silkscreen Layer

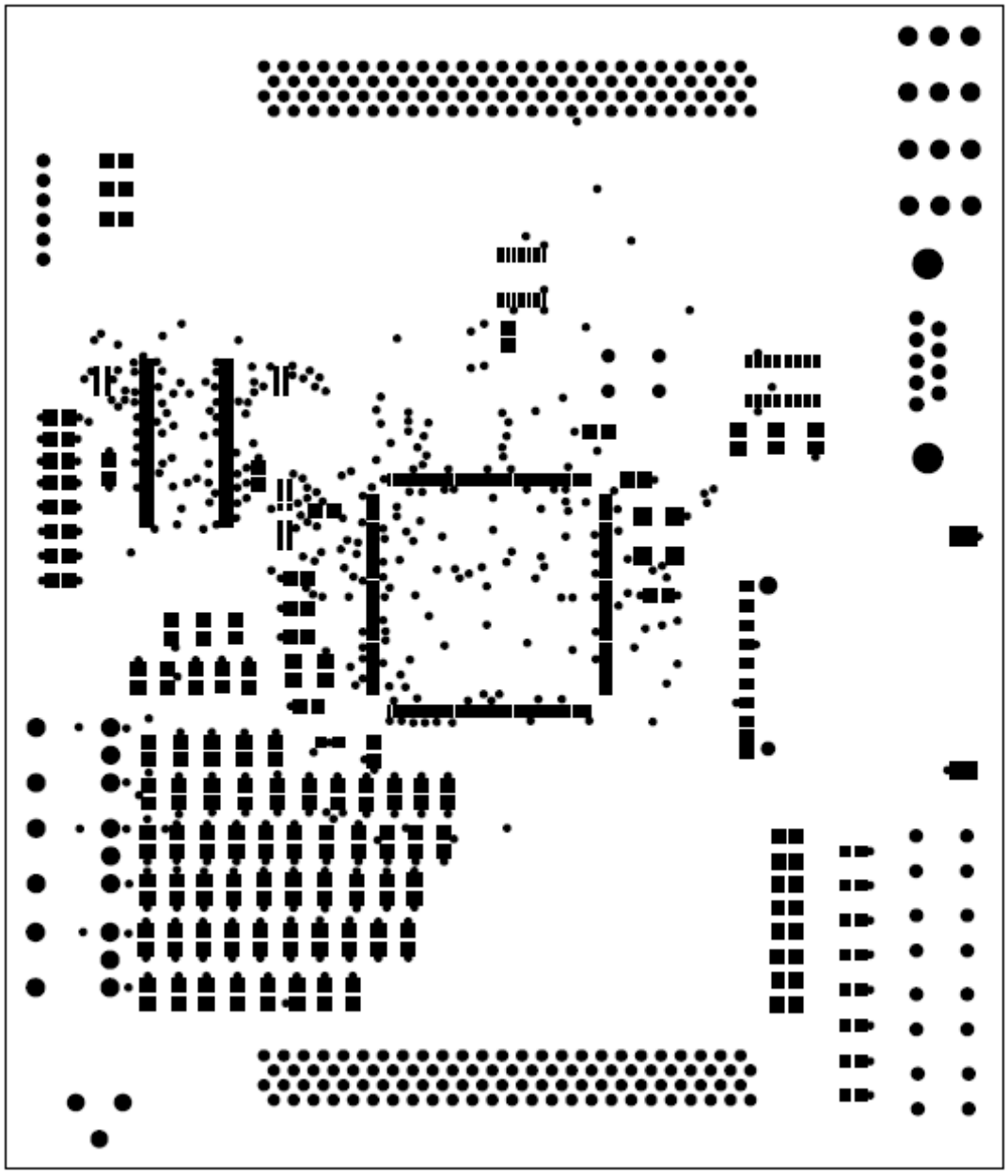

Figure 0-2 – PCB Top Soldermask Layer

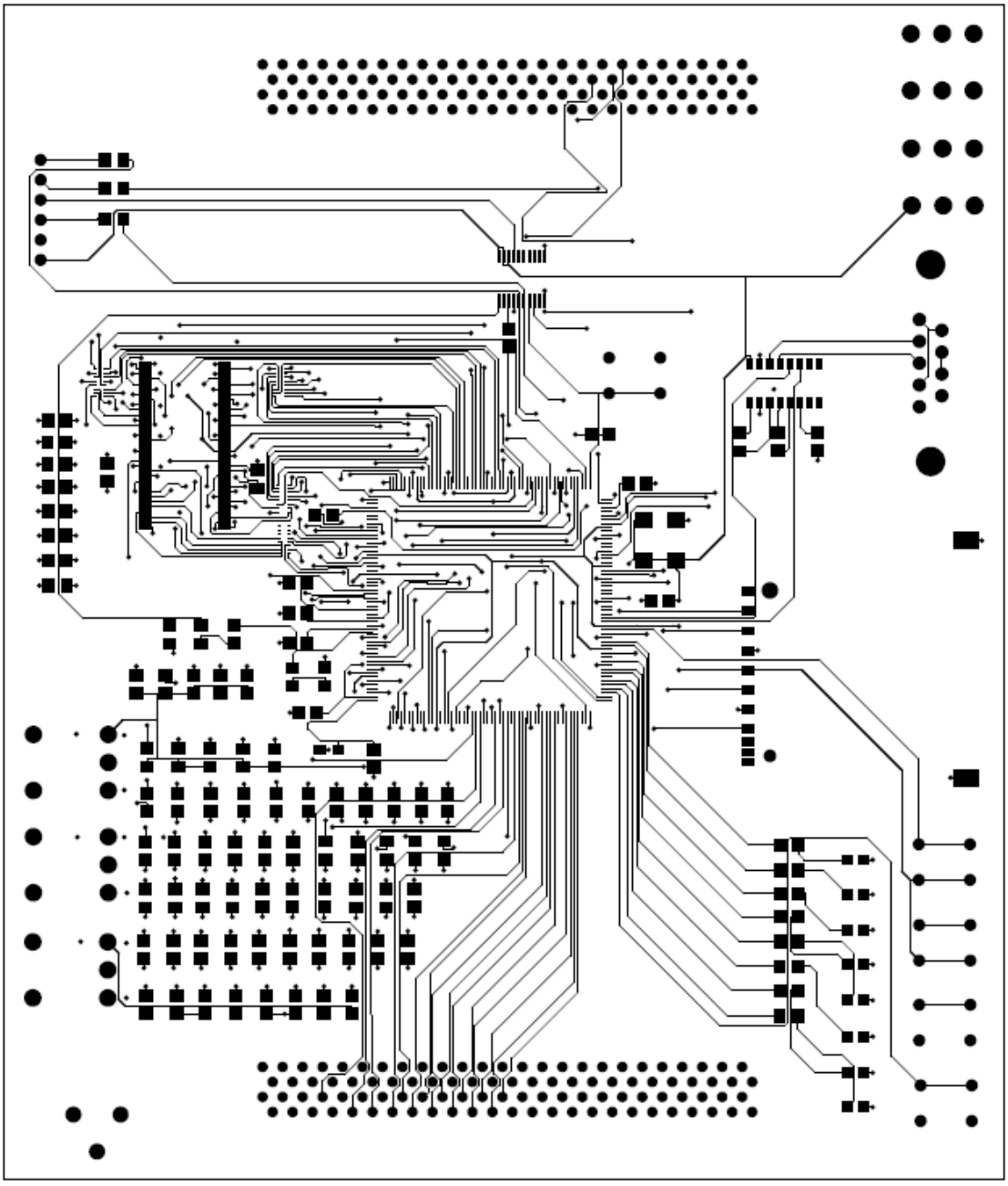

Figure 0-3 – PCB Top Copper Layer

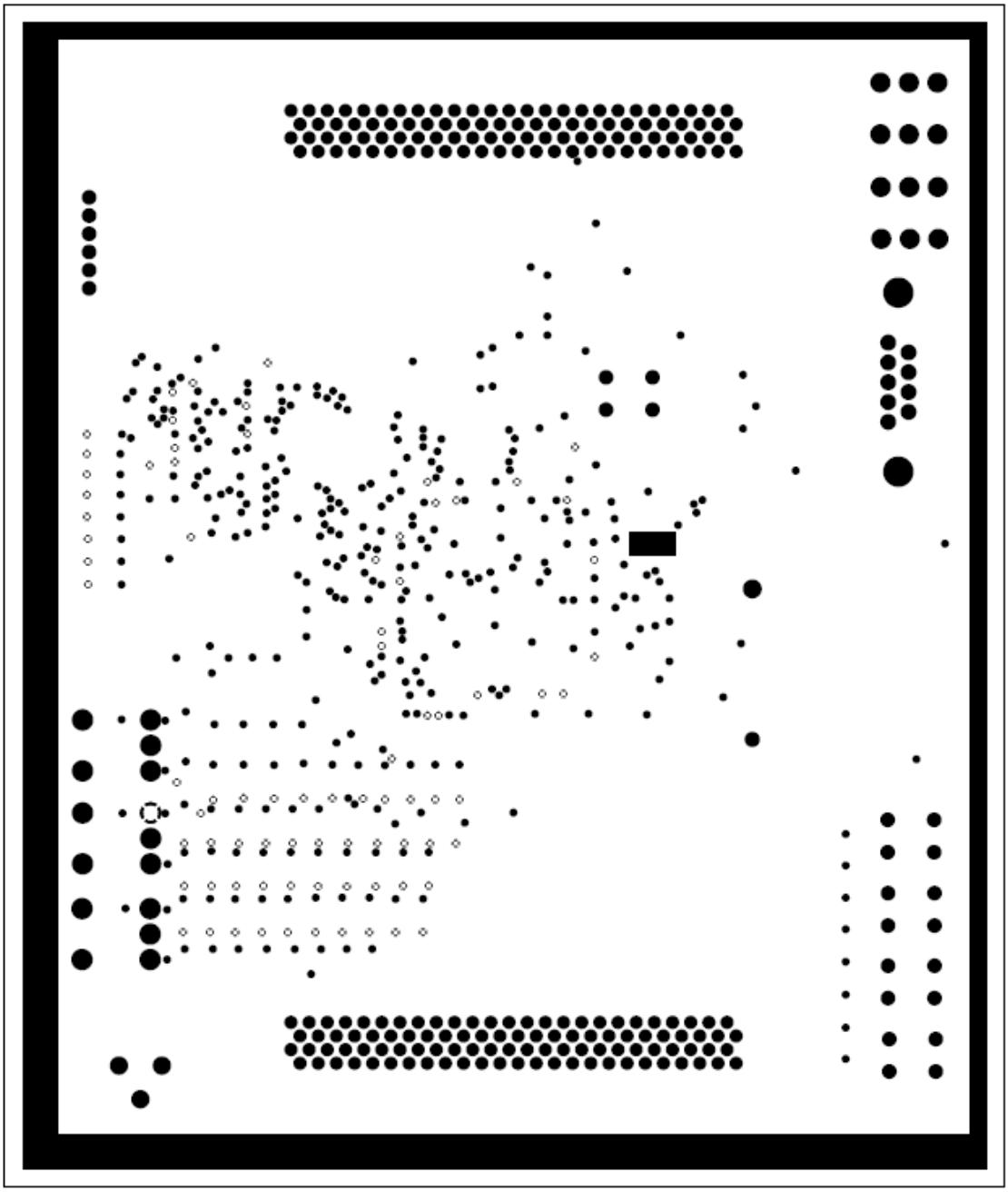

Figure 0-4 – PCB VCC2V5 Internal Power Plane Layer

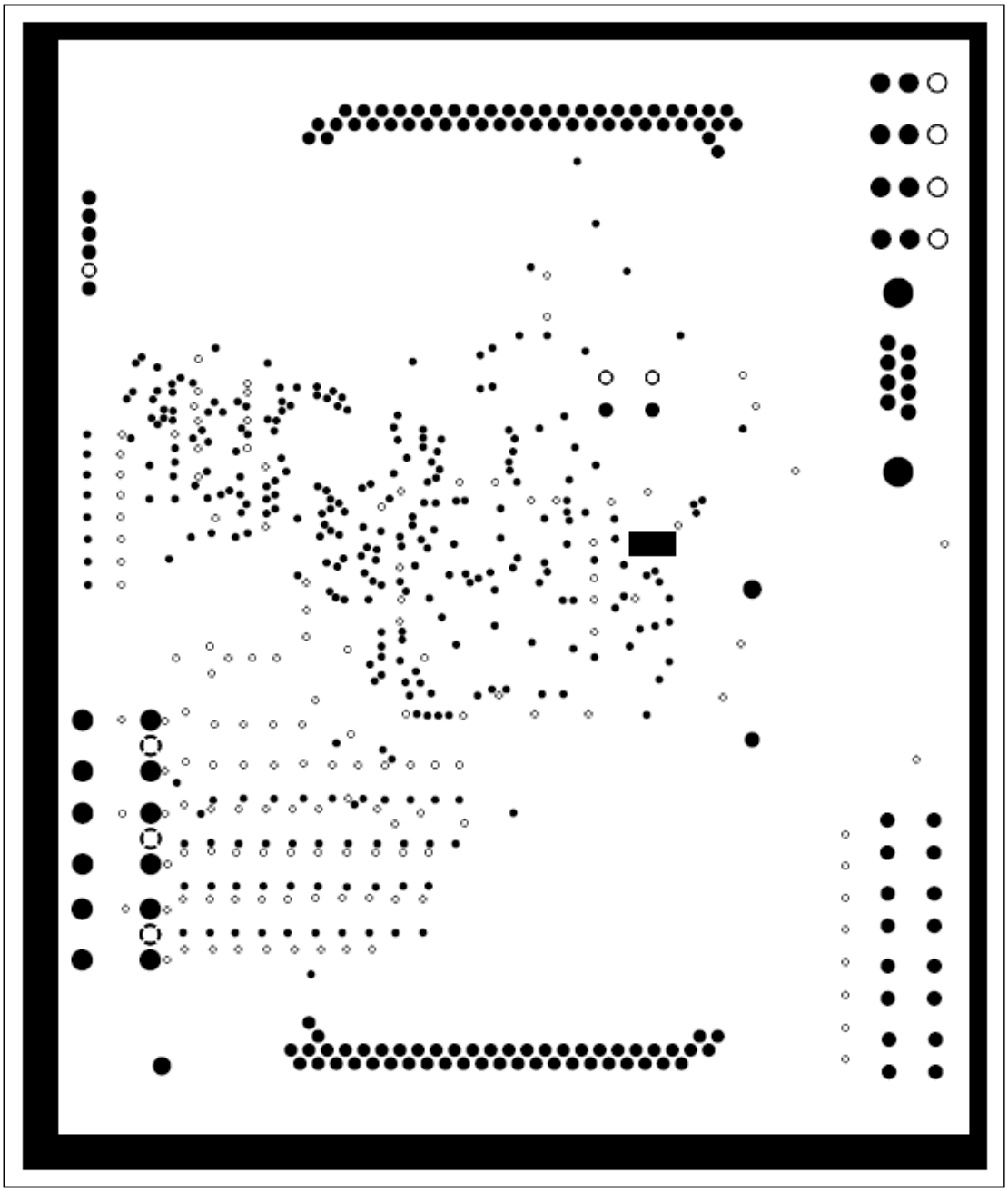

Figure 0-5 – PCB GND Internal Power Plane Layer

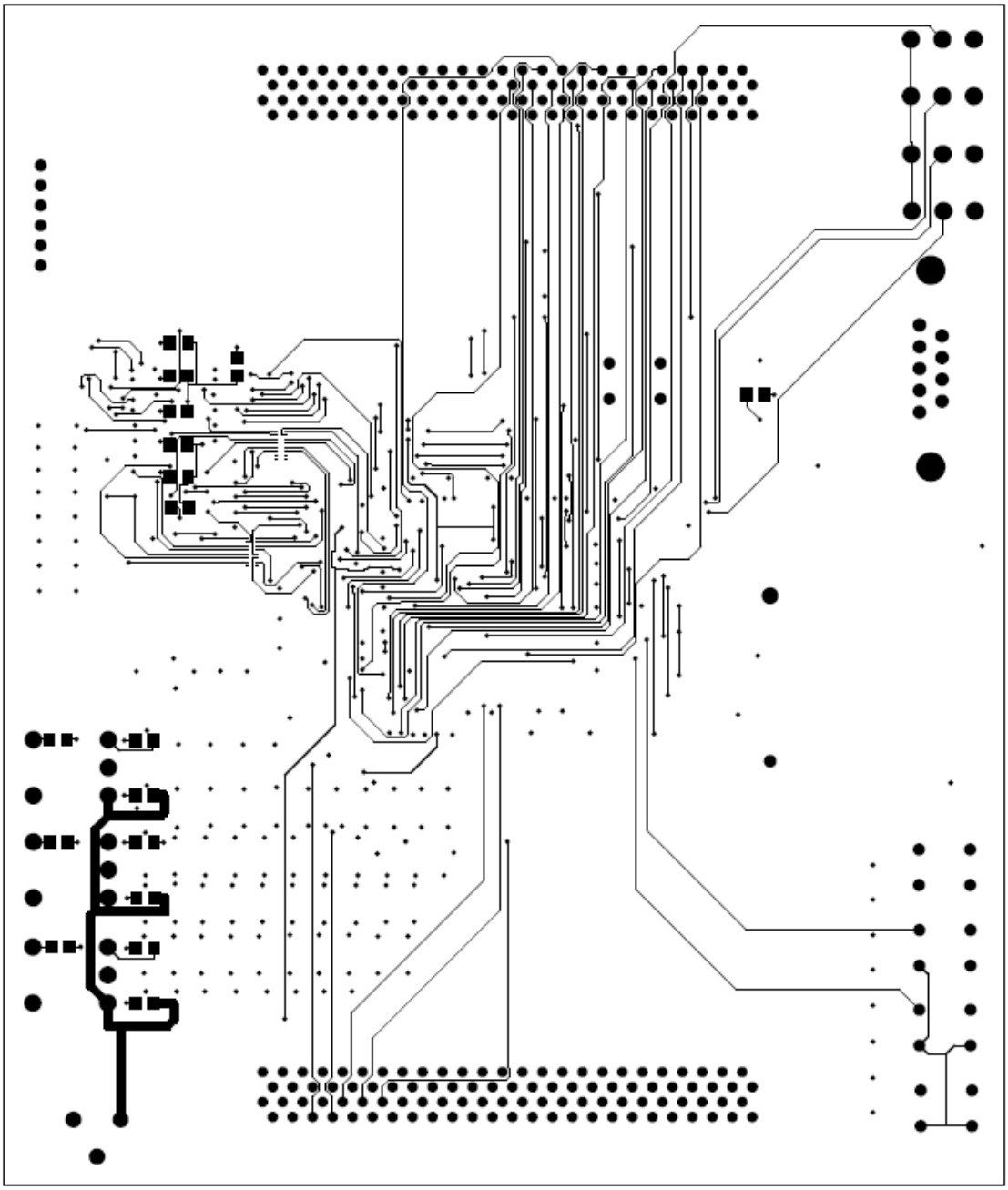

Figure 0-6 – PCB Bottom Copper Layer

Figure 0-7 – PCB Bottom Soldermask Layer

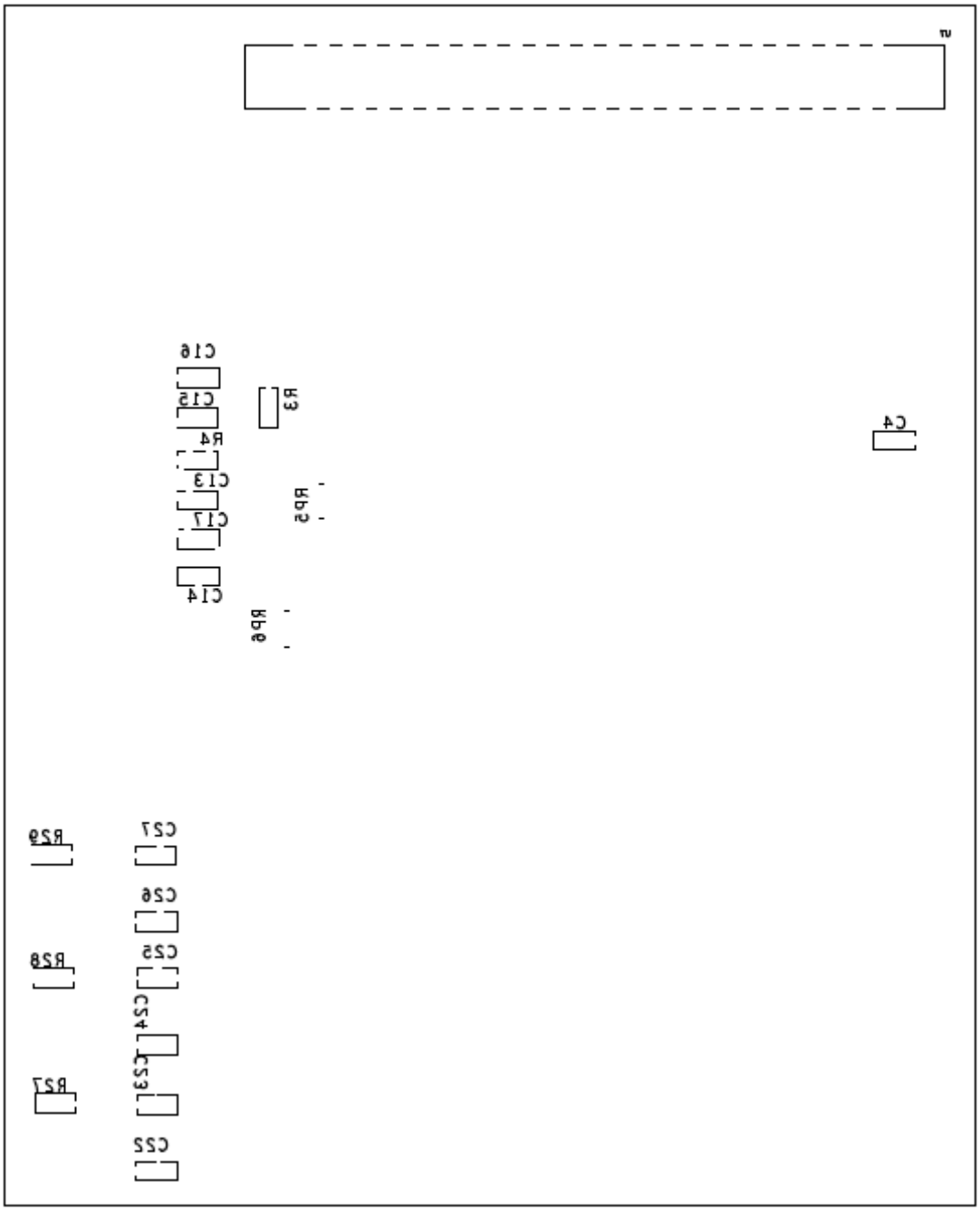

Figure 0-8 – PCB Bottom Silkscreen Layer

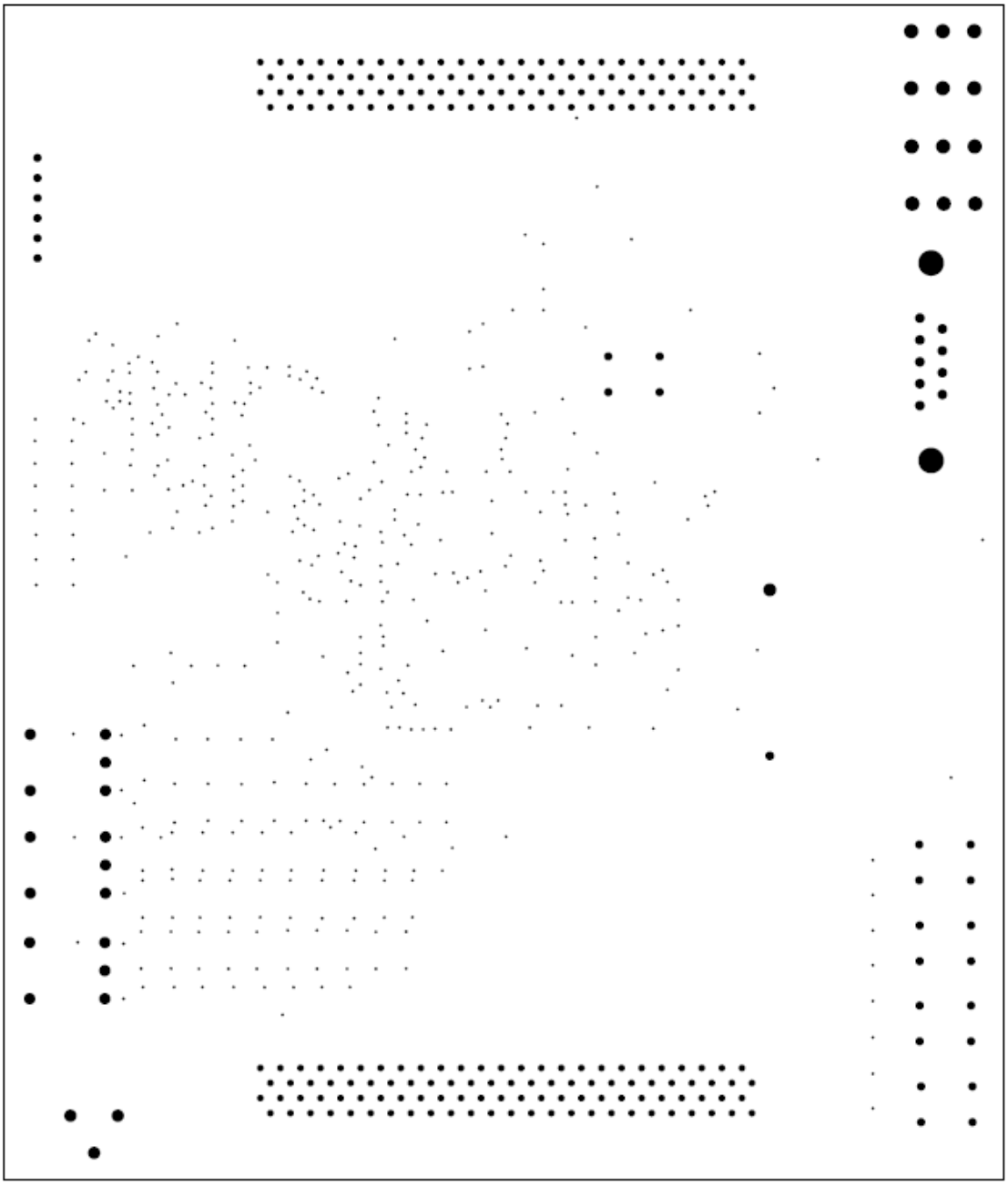

Figure 0-9 – PCB Drill File

REFERENCES

[1] Reza Rad, Jim Plusquellic, and Mohammad Tehranipoor, "Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008.

[2] Dakshi Agrawal, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar, "Trojan Detection Using IC Fingerprinting," in *IEEE Symposium on Security and Privacy*, 2007.

[3] Srivaths Ravi, Anand Raghunathan, and Srimat Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," in *17th International Conference on VLSI Design*, 2004.

[4] Mainak Banga and Michael S. Hsiao, "A Region Based Approach for the Identification of Hardware Trojans," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008.

[5] Xiaoxiao Wang, Mohammad Tehranipoor, and Jim Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008.

[6] Rudolf Usselmann. (2007, May) AES (Rijndael) IP Core : Overview. [Online]. http://www.opencores.org/project,aes_core

[7] Xilinx, "Spartan-3E Starter Kit Board User Guide", March 9, 2006.

[8] Niyaz PK. Advanced Encryption Standard (AES) Implementation in C/C++. [Online]. http://www.hoozi.com/Articles/AESEncryption.htm

[9] Wikipedia. Joint Test Action Group. [Online]. http://en.wikipedia.org/wiki/Joint_Test_Action_Group