Investigation of Non-Traditional Applications of the Physical Level in Reconfigurable Computing

by

Jacob D. Couch

Dissertation submitted to the Faculty and Guests of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor in Philosophy

in

Computer Engineering

Peter M. Athanas, Chair

Jonathan T. Black

Charles T. Clancy

Robert W. McGwier

Neil J. Steiner

March 22, 2016

Blacksburg, Virginia

Keywords: FPGA, Supply Chain Risk Management, Design Recovery, Ring Oscillator

Investigation of Non-Traditional Applications of the Physical Level in Reconfigurable Computing

Jacob D. Couch

(ABSTRACT)

Multiple research projects are proposed that utilize low-level knowledge of Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) design processes to enable additional research avenues. In order to accomplish these projects, Tools for Open Reconfigurable Computing (TORC) is utilized to provide a robust environment for circuit analysis and modifications. These projects rely on looking at the low-level constructs of the internals of these microchips. Through this knowledge, techniques for performing supply chain evaluations are proposed utilizing a non-binary comparison of multiple characteristic vectors between different FPGA manufacturing lots, and FPGAs that have been exposed to different environmental conditions. Second, techniques are proposed that look at design recovery by performing fuzzy segmentation and fuzzy matching algorithms to a problem area that has traditionally focused on exact graph sub-isomorphism solutions. Through these projects, additional research vectors are opened to protect and analyze the engineering efforts that are exerted in the design of FPGA and ASIC projects.

Investigation of Non-Traditional Applications of the Physical Level in Reconfigurable Computing

Jacob D. Couch

(GENERAL AUDIENCE ABSTRACT)

Field Programmable Gate Arrays (FPGAs) are a flexible class of processors that are commonly utilized in real-time and streaming applications. Although the underlying architecture of these processors is very flexible, constraints are placed on them through the compilation tools that are utilized to program these processors. This work seeks to look at some non-traditional applications that can be enabled by making use of the low-level details in the code that these processors run, instead of relying only on compilation process.

Dedication

This work is dedicated to my grandparents Donald and Virginia Couch from Yankton, South Dakota. Thank you for always believing in my education and supporting it for all of these years. You always had an open ear and great advice through everything. I am forever indebted to your wisdom and strength. May you now together both rest in peace. I love you.

Acknowledgments

I have no way of adequately thanking everyone who has contributed to the completion of this work. It has been a journey through many challenges, but through these challenges, we have all become better people. From Ralph Waldo Emerson "Life is a journey, not a destination".

To my Virginia Tech friends, Kelly Berry, Christine Beauchene, Graham Day, Katie Boggs Forsyth, Carrie Hughes, Chip Jackson, Ellen Messerly, Joe and Kara Newman, Anna Skinner, Britania Vondrasek, both near and far, thanks for all the fun through my many travels between DC and Blacksburg.

To my Virginia Tech and honorary Virginia Tech friends in DC, Jennifer and Stephen Haddock, Erica Hurt, Katie and Kyle McDaniel, and Josh Peters, thanks for all the great friendship as we have all become young professionals in DC.

To Davey Winyall, Lizzy Gallagher, Pipes, and many others who have shared the Ramble residence. Thanks for letting me crash at your place at all hours of the night.

To the members of the Configurable Computing Lab at Virginia Tech including Andrew Love, Kevin Lee, Kevin Zeng, Wenwei Zha, and many others, thanks for all the help and support, but also thanks for all the great times.

To Karl Pereira, I'm so sorry you have left this world so early. Your smile, kind heart, and

beautiful spirit always lifted us up. May you rest in peace, and may your spirit live on in eternity.

To the Bethesda United Methodist Church family, Rev. Kara Scroggins, Rev. Jenny Cannon, Rev. Adam Briddell, Rev. Ron Foster, thanks for all the support throughout all our spiritual joineries, and the welcoming to the DC area. To Troy, Libby, Pat, Rebecca, Nick, Rachel, Chris, Shaw, Sarah, and the young adult community, thanks for always being there. I will always be grateful integrating me into a community and pushing me to become involved in the youth, lead worship services, and to even go to Haiti.

To the Virginia Tech Secure Research Team, Tim Leake, Warren Lucero, Nick Littler, T.J. Beckett, and John Talerico thanks for making this project a success, and putting up with our crazy requests.

To JHU/APL, Andrew J., Angie T., Ben H., Carter W., Charles L., Erin M., Ian Z., Jackie T., Jeanne V., Jim S., John A., Josh S., Leif P., Morgan S., Nick P., Rob R., and Tommy J., thanks for the flexibility, insight, support, and advice.

To the USC/ISI East team, Andrew Schmidt, Ken Zick, Matt Calderon, Scott Stansberry, Steve Crago, and Matthew French, thanks for working with me on this research, opening your facility to me, and creating a great alliance.

To all the contributors to NWL including Dr. William Headley, Joseph Ziegler, and others. Thanks for helping me solve many problems within this project. To Dr. Chris R., thanks for taking a risk with this project. I believe your investment will have a great return. *The true sign of intelligence is not knowledge but imagination–Albert Einstein.*

To Alexandra Poetter and Sonya Rowe, thanks for keeping everything straight. I know we can be difficult at times, and you always kept us going, took care of bureaucratic problems, and ensured the success of the project.

To Ali Sohanghpurwala, thanks for pushing this project to the end. You have be a great friend through both the Masters and Ph.D. I wish you and your family the best of luck on the rest of your Ph.D. journey.

To Zac Oswald, thanks for always providing encouragement and reason through our journeys both to doctorates. It is exciting to think of what the future holds for both of us.

To Josh Moles, thanks for your advice in getting through this research. Only through your advice and paper reviews, could this research be as strong as it is today.

To Daric Adair, I know we have been through thick and thin, but you have always been there to support me. Thanks for listening, and supporting me throughout this process. I am forever indebted to your calmness and wisdom in showing me the path.

To Brad Barrett, thanks for realizing the potential of my master's research, and encouraging me to peruse it further. You provided the pathway to make much of this research possible. I will always be grateful.

To Dr. Elizabeth Reilly, thanks for working on the Design Recovery project with me. Through the combination of an FPGA guy and a graph theory expert, magic can happen.

To Dr. Jason Forsyth, thanks for being a lab-mate, reviewer, and friend. You have made the journey at Virginia Tech memorable, and I will always remember the great discussions we had. Congratulations on your Ph.D. and your new professorship!

To my committee, Dr. Athanas, Dr. Black, Dr. Clancy, Dr. McGwier, and Dr. Steiner, thanks for serving on my committee. Although this dissertation has been unusual, you have all been open to supporting the research. Hopefully, we have paved the path for future graduates students. Because of your service, graduate students are able to become researchers and advance the knowledge of mankind.

To Dr. Robert McGwier, thanks for the advice on the sponsored project. Without your guidance, critical components would not have been fixed and the project may have failed.

To Dr. Neil Steiner, thanks for working on the sponsored project, and being a committee member for my Ph.D. You have always taken the time to work with me, independent of all the other things you had going on. Your expertise of the TORC framework made much of this research possible. I'm so glad we got the opportunity to work together again.

To Dr. Athanas, thanks for always being a trusted advisor for all things. You have been a great source of knowledge in this journey known as graduate school. Thanks for taking on the NWL program, and offering me the opportunity to complete my Ph.D. under your guidance. Although we have not always seen eye-to-eye on everything, you have always been willing to listen. Your generosity and patience will always be felt.

To my family, Jim, Cathy, and Joey Couch, thanks for supporting me while I do these weird computer things. Thanks for supporting me through this journey, through thick and thin. Even though I broke everything in the house, you always gave me more opportunities to learn. I could not ask for more.

To God, the creator, the designer of all, the provider of my strength, and the redeemer of my life through Him. Thanks for giving me the talents to pursue my dreams. Your guidance and inspiration is perfect.

Contents

List of Figures

List of Tables

Glossary

3DES (Triple Date Encryption Standard) A symmetric key encryption standard that is an increase of the strength of the Data Encryption Standard (DES). 69

AES Advanced Encryption Standard. 69

ASIC Application Specific Integrated Circuit. ii, 2, 3, 9, 14, 69

BIL Bitfile Interpretation Library. 70, 82

CRC (Cyclic Redundancy Check) Data validation check utilized in Xilinx's bitstreams to ensure successful transfer to the FPGA.. 79

DARPA Defense Advanced Research Projects Agency. 14

DPA (Differential Power Analysis) A technique that monitors the current draw from an integrated circuit to determine internal operation states. 69

DSP (Digital Signal Processor) Xilinx's physical primitive for performing advance math functions such as multiplies, without utilizing the general slice components. 67

EDIF (Electronic Design Interchange Format) Vendor independent format to programmatically describe a netlist at the logic/generic level. 67, 87

FPGA Field Programmable Gate Array. ii, ix, 2, 9–11, 14, 65–69, 72, 83, 84, 87

FPGA Editor Xilinx tool to directly modify the routing, placement, and instance configurations of NCD designs. 77

HDL (Hardware Description Language) Language to describe hardware circuits implemented on either a ASIC or FPGA. 64, 88

HLS (High Level Synthesis) Hardware design languages that are more abstracted relative to VHDL or Verilog. Examples include LabView FPGA, ImpulseC, Viva.... 67

ILA Chipscope Integrated Logic Analyzer. 19

I/O Input Output. 35, 67

IP (Intellectual Property) Designs provided by external parties that can be integrated into a design. 14, 65, 67

ISA Instruction Set Architecture. 35, 36

ISE Integrated Synthesis Environment. 68, 71

JSF United State's Joint Strike Fighter. 9

JTAG (Joint Test Action Group) Interface utilized to initially program and debug microcontrollers, memories and FPGAs. 35

LUT (Look Up Table) Basic logic block of FPGAs to perform logic operations. 67

MUX Multiplexer. 84

NCD (Native Circuit Description) Xilinx's circuit design format originally from NeoCAD (NeoCAD Circuit Description). It is the output of MAP and PAR, and the input into PAR and Bitgen. Furthermore, it is the binary equivalent of XDL. 72, 74

NeoCAD An independent company purchased by Xilinx in order to utilize their placer and router after the Xilinx placer and router could not compete. 74

NRE Non-Recurring Engineering. 14

OpenPR Open Partial Reconfiguration. 71

PIP (Programmable Interconnect Point) Configuration element for enabling wires on a Xilinx FPGA. 67, 70, 77, 78, 82, 84, 86, 87

PUF Physically Unclonable Function. 15

RAM Random Access Memory. 67

ROM Read Only Memory. 9, 34

SCPA Semiconductor Chip Protection Act. 69

SLICE Xilinx's general purpose logic primitive. 78

SLICEL Xilinx's primitive slice that provides LUTs and a carry chain, but no DDR Memory or Shift Registers. 76

SRAM Static Random-Access Memory. 67, 68, 72

TORC (Tools for Open Reconfigurable Computing) Open-source toolkit developed by ISI and VT. ii, 19, 65, 78, 80, 83, 86

TPM Trusted Platform Module. 13

TRUST Trust in Integrated Circuits. 14

UCF (Universal Constraint File) Xilinx's constraint file for the placement of modules, assignment of ports, and timing constraints. 19

UNISIM (Universal Simulation) Xilinx simulation library for the primitive instances on Xilinx FPGAs. 67, 78, 87

XDL (Xilinx Description Language) Xilinx defined language that defines all instances, configurations, nets, and routes within a design. 64, 71, 72, 74, 75, 77, 84

XDLRC XDL Resource Database. 83, 84

XDLRC (Xilinx Description Language Report) Report generated by Xilinx tools that describes all physical instance and routing information per FPGA. 77, 78

ZIF Zero Insertion Force. 19

Chapter 1

Introduction

In 1908, the Ford Model T was released to the general public[1]. This automobile, in addition to being the first major success of the assembly line, was also one of the first major successes of a commoditized, standardized machine. Standardized machines brought about many of the technical advances of the early twentieth century. As these machines broke down or needed modifications for a specific task, resourceful people developed the skill-set to analyze, modify and repair these machines with their own know-how. This know-how was not always taught by the manufacturers, and in many circumstances was developed through empirical experiments. With many farmers entering the military in World War II, this know-how for repairing machines enabled them to repair their aircraft and weapons with whatever resources were available. This do-it-yourself mentality continued into the Information Age. From the "some assembly required" Apple I, to the GNU foundation's plethora of community-driven open-source software, there are plenty of opportunities for people to modify existing products to improve or specialize their functionality[2, 3].

Unfortunately, these community driven efforts are thwarted by both the complexity and miniaturization of many modern systems in addition to business practices that discourage

this behavior. The complex computer-driven systems of modern automobiles makes previously simple tuning operations from years past quite complex. Even devices such as home appliances, desktop computers, and even some scientific instruments are designed to be replaced when on component failure, instead of repaired as in previous generations of the device.

The Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) communities have experienced a similar transition over the recent generations of devices. FPGA vendors such as Xilinx and Altera once provided substantial documentation into the underlying architecture details. As newer, more complex, devices are introduced, the tooling for low-level manipulations is not adequately updated. This bit-rot limits the research opportunities available for new compilers, prevents non-traditional techniques on FPGAs, and limits security verification of devices. Research into these areas is essential as the growth of FPGAs is limited by exponential FPGA compile time growth, entrance of FPGAs into new markets, and new supply-chain and security requirements.

1.1 Motivation

In hardware and software design, there is a continual push to increase the abstraction of the design in order to improve the efficiency of developers as devices become more and more complex. Furthermore, there is a continued push to reduce many of the complexities of hardware and software design to enable additional developers and to reduce project development time.

FPGAs are a class of Integrated Circuits (ICs) that can be reconfigured in order to enable optimizations to solve a specific problem. To program these devices, traditionally, individual logic gates were programmed within the devices. Although this process provided high-level of

control within the FPGA, it was a tedious process that required substantial time to generate a working design. This tedious process has been abstracted away through modern design languages.

ASICs are a class of ICs that can perform many of the same functions that an FPGA can, though at a lower power and a lower production cost (for sizable quantities). ASICs are not reconfigurable; thus, during the design process FPGAs are utilized as development platforms. An FPGA's design can easily be translated into an ASIC, as the design is generally developed with the same languages, and the low-level differences are abstracted away with modern development tools. Although these abstractions reduce the development time of hardware designs, they hide some of the underlying architecture attributes. This dissertation seeks to evaluate some unique properties of FPGAs and ASICs, their designs, and their applications.

One attribute that is evaluated in this dissertation is the ability of an FPGA to provide measurement values that can be utilized to evaluate its supply chain properties. Groups of electronic recyclers have sprung up due to the large amount of electronic waste that is currently being generated though discarded electronics. Although the recycling of FPGAs may provide a lower cost alternative to a properly managed supply chain, the quality of these parts may be suspect. A technique is proposed in which FPGAs can generate integrity measurement vectors, through a nondestructive, active procedure. Through this procedure and additional analysis of these vectors, the current "health" and other supply chain properties of an FPGA can be derived.

Another technique being evaluated in this dissertation is the use of design recovery mechanisms within FPGA designs. The source code of systems often becomes separated from the distributed binaries for a product. Furthermore, if the source code and documentation for a legacy system is lost, the original functionality and specifications for the system may also be lost. Multiple techniques look to utilize some low-level properties of FPGAs to assist

engineers in recovering the original design. Furthermore, design recovery can be utilized to perform competitive analyses and even assist in Intellectual Property (IP) protection. Although there is existing research in this field, the proposed techniques utilize multiple approximation techniques to provide a more reasonable computational bound in comparison to other current techniques that are utilized in order to provide a mechanism to understand existing legacy systems.

1.2 Dissertation Statement

The current landscape of vendor-controlled toolchains for both ASIC and FPGA markets have limited many of the possible research vectors within the reconfigurable computing community. As these tools have developed, they have also adapted numerous abstraction techniques. Because of these abstraction techniques and the closed ecosystems, there are many assumptions within the hardware development community that are followed, but not openly questioned. This research explores three different pathways of utilizing an FPGA or FPGA/ASIC design tools to provide alternative capabilities that are not commonly utilized today, including design recovery and supply chain integrity.

1.3 Research Problems

This dissertation addresses the following problems:

- **FPGA Supply Chain Risk Management** The lifetime of FPGAs in critical systems is ever increasing. The redesign rate of these systems is always longer than intended. Therefore, older FPGAs may need to be utilized in presently manufactured systems,

due to the substantial overhead of retesting and recertification of these systems. Current mechanisms for the verification of the acquisition of these FPGAs is purely policy and procedure based. Technical solutions to this problem are treated in Chapter 2.

- **HW Design Recovery** The engineer's design is difficult to recover from an unknown circuit utilizing existing techniques. The recovery of this design can be useful for compatibility engineering, competitive analysis, and is even utilized in the prosecution of IP protection lawsuits. This process is currently very time consuming both in engineering and computational time. Technical solutions to this problem are treated in Chapter 3.

- **Additional Topic** The research problem for this topic is discussed in the addendum.

1.4 Research Questions

This research is broken into three major segments. The first two segments are discussed in this document. The third segment is discussed in the addendum to this document. The following are the research questions that guided this research.

- **RQ 1:***How can FPGAs be protected from supply chain vulnerabilities?* Chapter 2 investigates a non-traditional measurement technique that can be utilized to compare FPGAs against other FPGAs in order to determine the homogeneity of a manufacturing batch. These measurements can also be utilized to detect certain types of malicious modifications to the FPGA. With this type of detection both secondary market parts and tampered parts can be identified. This identification assists in ensuring the integrity of the supply chain.

- **RQ 2:***How can the design of a compiled design be recovered in order to evaluate and*

understand the functionality of a specific system? Chapter 3 outlines a technique for the intelligent annotation of various subsets of a design from within a larger recovered design. These subsets contain atomic elements that are reconstructed into a graph that represents the connectivity of these primitives. This graph can the be subdivided and compared against other known graphs to identify common blocks. This technique is an alternative fuzzy approach that has reasonable computational requirements in comparison to existing techniques, which are currently computationally bounded.

- **RQ 3:** *This research question is discussed in the addendum.*

1.5 Contributions

This research focuses on many of the alternative capabilities that are available on FPGA and ASIC platforms. This section outlines the specific contributions of this body of work. The details and techniques of these contributions are within the following chapters.

- *A method that utilizes asynchronous circuits in FPGAs to provide information about the supply chain characteristics of a specific die.* This method is capable of identifying lifecycle events such as a reflow of an FPGA within an existing system. This may be indicative of tampering within the system, and can be a trigger for additional supply chain actions. Another lifecycle evaluation is to determine the homogeneity of a manufacturing batch of FPGAs. This is utilized to assist in detecting counterfeit FPGAs that could compromise the integrity of a supply chain.

- *A method for the segmentation and non-graph sub-isomorphism pattern matching of recovered hardware designs.* By re-purposing an existing imaging processing algorithm for image segmentation, recovered designs can be partitioned into a collection of seg-

ments that are easier to analyze in order to gain a better understanding of the original design. This segmenting reduces the computational complexity of current algorithms in this space. The current graph subgraph isomorphism algorithms for design recovery are computationally expensive, and are not optimized for the types of designs that are generated by commercial FPGA compilers and High level Synthesis (HLS) tools for ASICs. This pattern matching method compares vectorized segments against a library of other well known segments in order to remove commonly known segments from the collection of unknown segments.

- Additional contributions that are listed in the addendum.

Additional work was completed in the realization of these techniques on real hardware, and it is discussed within the Appendix. The specific details are restricted due to the sensitivity of the platform, and existing Non-Disclosure Agreements (NDAs) that are in place. Although this work brings completeness to this dissertation, its lack of inclusion should not distract from the fundamental research contributions described henceforth. Furthermore, it is not categorized as one of the major contributions of this dissertation.

1.6 Dissertation Organization

This dissertation is organized into three research chapters, with additional background support located in the appendix. Within each research chapter there is a section that is an analysis of the current state of the research problem.

Chapter 2 is a discussion on supply chain and device lifecycle risks for FPGAs and a technique to mitigate some of these risks. Chapter 3 is a discussion on techniques to perform design recovery for FPGAs and ASICs. Chapter 4 references an addendum that is a discussion

of a unique technique that was developed to provide a solution to a sponsor. Within the restricted appendix is additional applied research that was in support for the above chapters.

Chapter 2

Supply Chain Integrity Measurement of FPGAs

Sections of this chapter were co-authored by J. Arkoian.

Publications:

J.D. Couch, "Characterization and Verification of FPGA Die for SCRM", *Defense Technical Information Center (DTIC)*[Not publicly available], 2014

J. Couch, J. Arkoian, "An Investigation into a Circuit Based Supply Chain Analyzer for FPGAs", in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Lausanne, 2016, pp. 1-9. [Online]. Available: https://doi.org/10.1109/FPL.2016.7577335

Provisional Patent: J.D. Couch "Physically Unclonable Functions for Supply Chain Risk Management in FPGAs",June 22, 2017

2.1 Introduction

Systems that are designed with FPGAs have lifespans that range from a few years, to multiple decades. To reduce non-recurring engineering time, these systems may be built using proven, existing designs. Even though these existing designs are known to work, modifying the design to utilize a modern FPGAs, even with identical code, can require substantial testing and certification resources. Therefore, older FPGAs are still required to be procured and used in modern systems, because they are no longer available from the manufacturer, they may come from less than fully trusted suppliers in order to continue production of a product.

With FPGAs becoming more common in consumer, industrial, and military systems, this problem of a limited supply of older parts will only become more prevalent. This research provides a technique to evaluate FPGA dies throughout their life cycle, and detect attributes about the "lot" identity and environmental exposures. A "lot" of FPGAs is defined as those that are manufactured in a single manufacturing batch, and carry identical lot codes and manufacturing dates. It is hypothesized that there are slight variations in the manufacturing process and the environment of manufacturing that can be revealed by the slight changes in a FPGA die. This is accomplished by programming multiple Ring Oscillators (ROs) on an FPGA, and then measuring their oscillating frequencies with high precision. The integrity of a specific die relative to the lot of delivery and the integrity of a lot of FPGAs are evaluated. This technique does not evaluate the board that the FPGA is connected to nor the bitstream that is loaded on the FPGA, but it provides a capability to detect modifications to the physical FPGA through the supply chain.

2.1.1 Motivation

FPGA vendors are staying competitive with ASIC vendors by consistently updating their design with new features, smaller feature sizes, faster executions, and larger fabrics. New FPGA generations are generally released by manufacturers every two years with a manufacturing cycle of five to seven years[4]. In addition, many modern systems are utilizing an increasing number of FPGAs in their designs. For example, the United State's Joint Strike Fighter (JSF) aircraft contains over 200 FPGAs, out pacing the number of custom ASICs [5]. Currently, the supply of FPGAs for this project are well secured, but mission critical systems in the past have been compromised from faulty FPGAs[6]. If a design in an existing system requires an older FPGA, and proper supply chain preparations were not conducted for this system, it is possible that additional FPGAs may need to be acquired through secondary market channels.

A secondary market FPGA may be required to sustain the operation of a system due to an exhaustion of other options. This problem will only continue to grow as mission critical systems are increasingly expected to outlive their original designed lifespan in this budget conscious environment. Because of the changing mission requirements, the correct end of life orders cannot be assured. This implores the need for a mechanism to integrate and validate the integrity and health of both existing FPGAs in the field and shelf lots of FPGAs.

2.1.2 Goals

The goal of this research is to provide product owners additional tools to evaluate the "health" of their products. This "health" is defined as ensuring the correct operation of the FPGA to authorized users throughout its lifetime, while not inducing any additional effects that may be advantageous to an adversary. FPGAs that have been recycled or exposed to

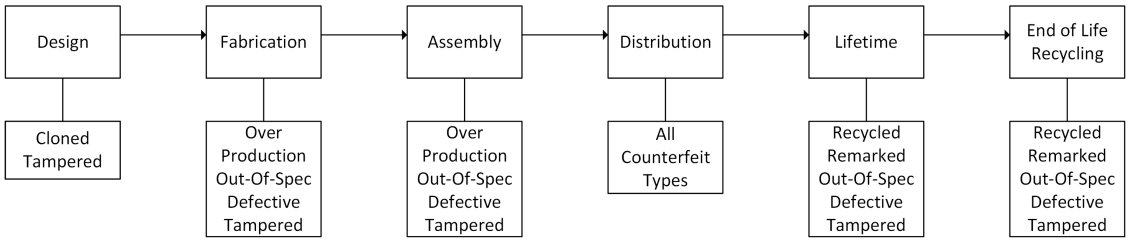


Figure 2.1: Generic six-stage supply chain example with attack vectors[7].

extreme environmental conditions are less likely to maintain this standard of "health".

Figure 2.1 depicts a general microelectronic system life-cycle and potential attack vectors at each stage of the life-cycle[7]. This life-cycle is further adapted for an FPGAs in Figure 2.2[8]. For example, in the manufacturing stage, the manufacturer could place a malicious modification within the FPGA. This research seeks to provide an out-of-band evaluation mechanism at the procurement, system assembly, system deployment, and system operation levels that will detect modification.

In the procurement stage, the FPGAs can be evaluated within each lot. If the uniformity of the measurements from within a lot is not as expected, the lot can be marked as suspect. At the system assembly and deployment stage, the FPGA's lot integrity can be periodically measured. If changes in the measurement are detected between measurements at different inspection points, especially if handled by an untrusted actor or unknown environmental conditions, the FPGAs can be marked as suspect and the system can be evaluated. Finally, there is a lofty goal that throughout device operation, the device can measure itself and identify if it is beginning to break down, providing early warning for a potential upcoming failure. The foundation of this goal is presented in this chapter, but its realization proposed for future work.

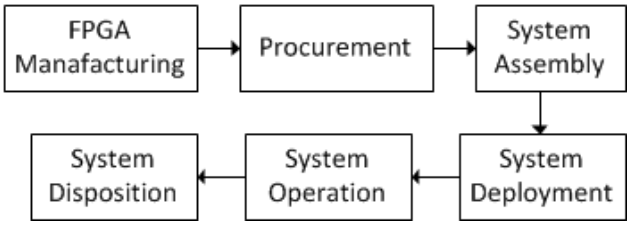


Figure 2.2: FPGA based six-stage supply chain example[8].

2.2 Background

This research provides an additional vector to evaluate the "health" of an FPGA. Existing examples of compromised systems, the current state of counterfeit market, and the lack of non-destructive die integrity measurements lead to the developing the need for this research.

2.2.1 Example System

One example of improper FPGAs being used in mission critical systems is the ice detection system for the P-8A aircraft[6]. The P-8A aircraft deicing system relies on FPGAs to control the heaters on the wing surfaces. When the P-8A was processed for a midlife upgrade, the deicing system was refurbished. Aircraft that contained this upgraded deicing system began to have early life failures of this system. After analysis of the system, it was determined that the FPGAs that were used in this system were purchased from a secondary market vendor who supplied recycled FPGAs. Because the chips had already been used they did not exhibit the traditional "Infant Mortality" failures that would be expected in burn-in as seen in Figure 2.3[9]. This lack of "Infant Mortality" was covered by a continued out wear out failure. In the case of the P-8A aircraft, the parts on the aircraft were farther to the right on the wear out failure graph than expected. This resulted in numerous mid-life failures that were unexpected. This ultimately resulted in a replacement of all affected components to ensure the safety of the aircraft.

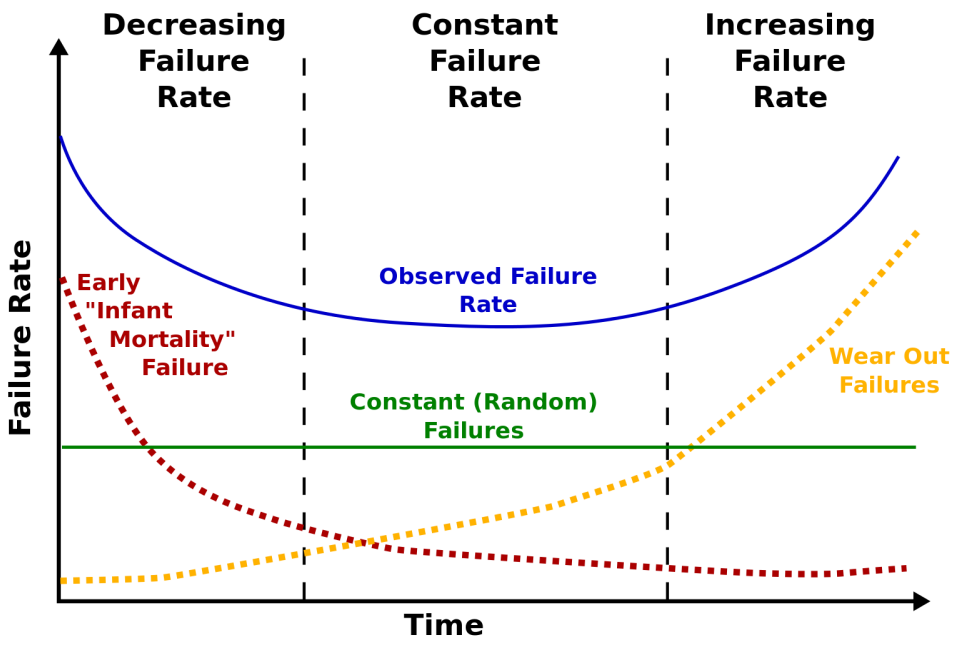


Figure 2.3: Bathtub curve for failure analysis.

2.2.2 Status of Counterfeit Electronics

In 2008 it was estimated that counterfeit and pirated products impacted the global economy by $775 billion, with a potential to double by 2015[10]. Although this accounts for all types of counterfeit and pirated products, counterfeit and pirated electronics are a $169 billion segment of this problem[11]. Furthermore, counterfeit programmable logic is a $2 billion segment of concern.

The types of defects that are the motivation for this research are defects that affect the reliability and usability of the system. Figure 2.4 outlines a taxonomy of potential problems with non-genuine parts[12]. Any of these failure modes could be a function of an insecure supply chain.

In order to combat these risks, verification procedures are introduced to ensure the integrity

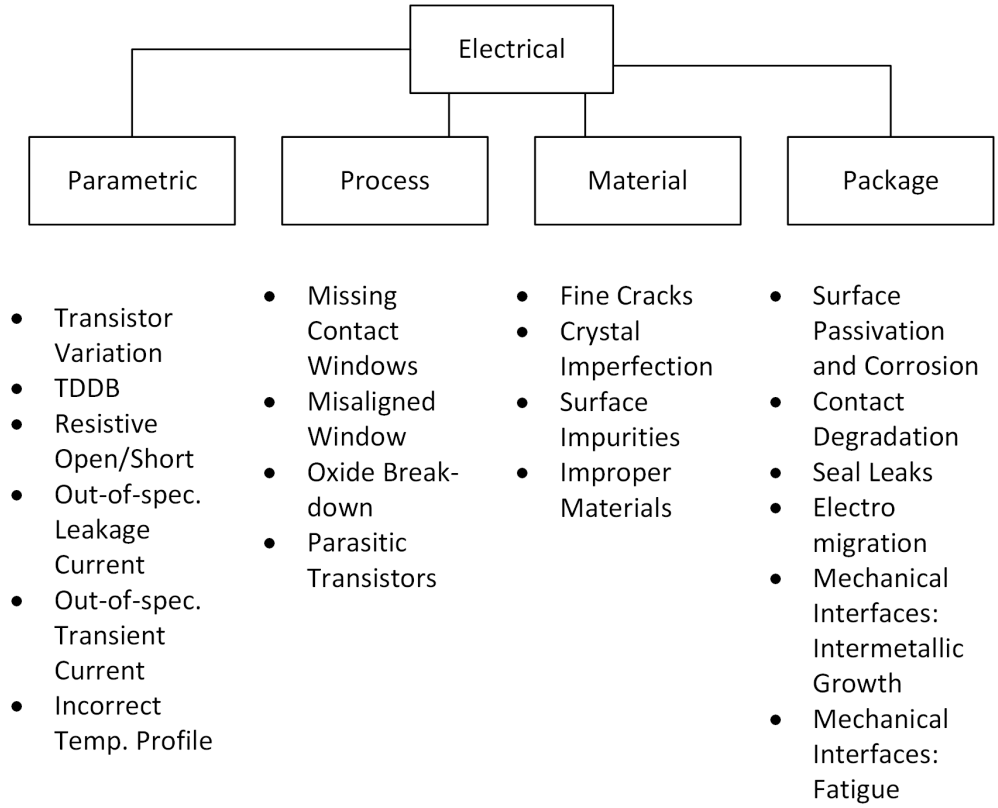


Figure 2.4: Taxonomy of electrical defects in counterfeit components[12].

of the supply chain. Typically, in most supply chain operations, a physically identifiable artifact is attached to the device of interest. This could be anything from a serial number to an anti-tamper or genuine seal. This artifact is not easily replicated, and provides a quick, non-destructive mechanism to identify the integrity of the device through all stages of the supply chain. It does not actively measure the integrity of the internal components of device.

The software stack that is on these devices may also be digitally signed in order to discourage any non-approved software modifications to the software. Examples of this technique range from utilizing Trusted Platform Modules (TPMs) to ensure a secure boot, to blowing a fuse on the processor when an unapproved operation occurs, to disabling any trusted functionality, similar to Samsung's Knox system[13, 14].

If these physically identifiable artifacts do not provide sufficient protection, or are not applicable to the architecture of the device, destructive methods may be employed to evaluate the device. Some of these methods include polishing and imaging the silicon dies within the device and chemical analysis of the components on the die[12].

2.2.3 Existing Research

This research relies on the use of asynchronous fingerprinting functions to help identify supply chain risks. Existing techniques generally fall into two categories: supply chain policies and FPGA die unique Physically Unclonable Functions (PUFs).

Supply Chain Risk Mitigation

There are still outstanding risks from cloned and counterfeit FPGAs[15]. Substantial research and operational policies have been completed to attempt to mitigate the risk of supply chain vulnerabilities[16]. These techniques include proper procurement practices, supplier

selection, counterfeit component disposition, and disposal of electronic waste. In general, this research is primarily focused on ensuring trusted suppliers at all stages of the production cycle, or providing safeguards to detect non-compliance in the supply chain.

Compromises in the supply chain may be done for malicious intent to ensure "modified" supplies are used in a system. This would enable an adversary to affect the reliability of a system. Compromises may also be for economic advantages by utilizing gray market supplies when other supplies are either not available or at a higher price point. This type of compromise was estimated to be a $58 billion dollar market in 2007, with expectations for growth[17]. This is not a recent trend. There have been numerous cases of counterfeit parts as the manufacturing of "American" products have been transfered to international factories[18].

The concept of counterfeit materials with similar operating properties has been a long term research problem. Originally, this problem was a concern because the owners of the IP were not receiving proper compensations for the Non-Recurring Engineering (NRE) costs[19]. Furthermore, the end consumers were expecting support for the counterfeit parts. One recent example of this was the FTDI FT232RL chips that were routinely counterfeited. FTDI released a driver update that disabled the functionality of the counterfeit chips[20]. In this situation, the consumers who were affected by this update had purchased in their belief legitimate products, which unbeknown to them contained a counterfeit part.

Throughout the device life-cycle there are many points where the supply chain could be compromised allowing for malicious modifications, which may introduce uncertainty into the system. The Defense Advanced Research Projects Agency (DARPA) Trust in Integrated Circuits (TRUST) project was tasked with looking into types of malicious modifications that could be inserted into both an FPGA die and an ASIC design [21]. Through this project, numerous attack vectors were discovered in the life-cycle of an FPGA.

In general, the mechanisms to reduce the risk of a breach in the supply chain revolve around ensuring trusted suppliers and verification of activities in all components throughout the life-cycle of the platform. Although this approach reduces and mitigates many of the available attack vectors, it does not provide a mechanism to measure and characterize the properties of the actual platform to determine if the supply chain has been subverted.

Physical Markings

The common way to ensure that parts within a supply chain can be validated at a later time is to place tamper evident artifacts on the device. This could be as simple as a serial number on the device, device tracking systems, and even Radio Frequency IDentification (RFID) trackers. These simple methods are being expanded to include tamper evident seals, embedded Ultra Violet (UV) labels, and even a micro-tag as exhibited by the DARPA Supply Chain Hardware Integrity for Electronics Defense (SHIELD) project[22]. There have even been attempts at attaching holograms to chips that can only be activated by a specified laser[23]. Although all of these methods can be used to improve the integrity of the supply chain, they require either substantial chain-of-custody arrangements, substantial individual device tracking efforts, or a passive appendage that does not actually measure the operational characteristics of the die.

Counterfeit devices can be detected after the fact through non-invasive techniques such as chemical analysis of the packaging, or invasive and destructive techniques such as Scanning Electron Microscope (SEM) imaging. Furthermore, all of these techniques are external protections that are applied to the device, but do not evaluate the performance characteristics of the device.

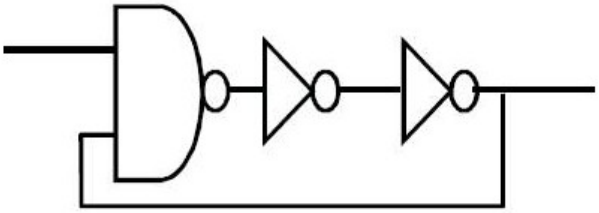
Physically Uncloneable Functions

Numerous organizations have conducted research into PUFs for determining unique keys per die within an FPGA[24, 25, 26]. This research is primarily focused on determining an authentication token for a specific FPGA die. Many types of PUFs currently exist, including ring-oscillator, butterfly, and arbiter PUFs [27, 28]. In all of these PUF techniques, it is a binary comparison between two structures within the same FPGA. This is done to mitigate many environmental factors, such as voltage and temperature variations, that may influence the measurement. In this experiment, it is a value measurement of the frequency of oscillation calculated by utilizing a stable external oscillator as a reference.

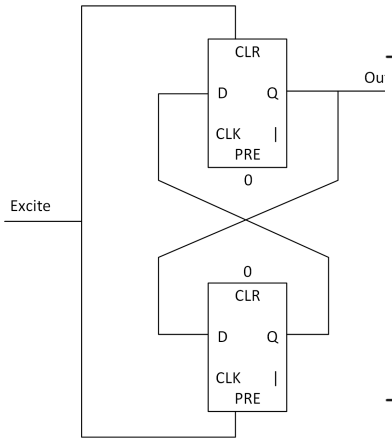
Within this experiment, the ring oscillator will be used as the measurement apparatus because it allows for easy comparison to an external oscillator. Furthermore, this comparison can be measured on a numeric scale in comparison to the binary output of other PUF options.

These PUFs are required to generate unique key material for the specific die, while mitigating for differences in temperature and voltage environmental conditions[29]. To accomplish this, most of these techniques rely on generating a single bit by comparing a pair of ring oscillators. For each of these ring oscillators, the ring size generally varies between three and twenty five elements. Research has also been conducted to ensure the integrity of these measurements throughout the life cycle of the FPGA[30].

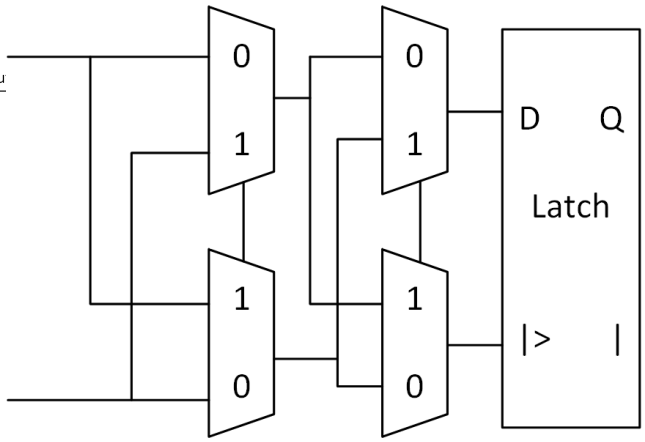
J. Graf proposed a method for supply chain integrity utilizing multiple PUF measurements across the FPGA[34]. These measurements were initially collected at manufacturing time, and then the FPGA would re-authenticate utilizing these measurements one by one during operation. Although this work places the utilization of PUFs in a framework that is compatible with the current needs of supply chain problems, it does not evaluate utilizing PUFs to determine the integrity of a manufacturing lot, or the health of a specific FPGA.



(a) Ring oscillator PUF[31].



(b) Butterfly PUF[32].



(c) Arbiter PUF[33].

Figure 2.5: PUF variations

Yu and Devadas conducted research that generated a method of combining PUF measurements to create additional unique material for a specific FPGA[35]. This research is useful when placed in conjunction with Graf's research to ensure there is enough unique material for the life of the system; however, it does not address the integrity of a manufacturing lot or the specific health of an FPGA.

Dogan's recent research proposes detecting recycled FPGAs by identifying a slowdown in ROs within an FPGAs due to aging[36]. This work is similar to the work presented in this chapter. This chapter seeks to extend this work by looking at lot classification, and to provide better classification of aging dies through the use of a Support Vector Machine (SVM).

The body of existing research on utilizing PUF techniques to determine the relationship of FPGAs with regard to their supply chain properties is primarily focused on individual die identification. Although this is the gold standard in individual die tracking, the overhead for this is substantial. Furthermore, it requires a priori knowledge of every FPGA that may be investigated in the future. Finally, it still leaves a hole in the supply chain before the first measurement. For the purposes of this work, the use of PUFs for individual, unique die identification is considered out-of-scope. The research in this section seeks to begin initial investigations into the feasibility of using this specific timing information for conducting supply chain inquiries.

Foundry Identification

Wendt contributed a method for identifying foundries by looking at the delay of certain gates across a wide variety of supply voltages on ASICs[37]. Through this technique, values were measured from multiple ASICs based on the completion time of a subset of the ISCAS89 and

ITC99 benchmark suite. These measurements were then passed to a SATisfiability (SAT) solver utilizing the Kolmogrov-Smirnov test. Although Wendt was successful in foundry identification of ASICs, it was never applied to "lot" identification of FPGAs. Although varying the supply voltage to generate additional data could be used on an FPGA, the magnitude of the voltage range performed in this experiment is unlikely to work on an FPGAs due to the design requirements of an FPGA holding the configuration in Static Random-Access Memory (SRAM) cells.

2.3 Technique and Implementation

Typical designs of FPGAs consist of synchronous logic that is interconnected through the routing network within the FPGAs die. Through the compilation process, the longest delay paths are identified and the design is optimized such that these paths satisfy the timing model for the FPGA. Only under worst case situations of temperature, voltage, and manufacturing defects will the FPGA actually approach these timing estimations. Otherwise, the FPGA will complete the required logic equation for a clock cycle and store the result in a state. By removing the synchronous structures from the design, and introducing multiple, free-running, asynchronous structures, the specific timing characteristics of both the routing and logic on an FPGAs can be measured.

Through the process of designing adequate ring oscillators with appropriate placement constraints, a chip-wide measurement can be derived for the FPGA. By placing multiple structures of different lengths and routing constraints, this technique can generate additional measurement material that can be used in the analysis of the die. All measurements in this project were conducted with an external oscillator as a reference clock, and the frequency of oscillation of the internal structure generated the material for analysis as shown in Figure

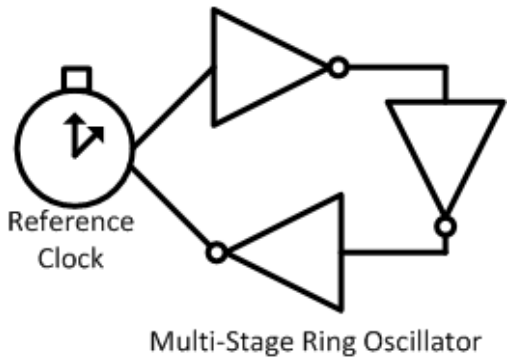


Figure 2.6: Simplified example of experiment.

2.6.

With this measurement, a unique fingerprint can be derived for a FPGA die. In existing PUF techniques, typically oscillators are only compared against other oscillators on the die to determine the key material. In this supply chain based measurement, the least significant portions of the measurement cannot be used because they uniquely identify the individual die, but more significant portions of the measurement can be used. These portions provide insight into the supply chain properties of this FPGA.

For the experiments in this chapter all measurements were conducted on Virtex FPGA (XCV1000-4C)[38]. This FPGA was released in 1999; however, as of 2015 it is still used in many systems. In addition, this FPGA utilizes a 220nm feature size, which is quite large with respect to modern FPGAs. This FPGA was selected as it is still used in existing designs, and multiple lots of manufacturing are still available for procurement. Although the use of this larger feature size may contribute to changes in the results, research from Sedcole and others suggest that the variance in the die will continue to be expected in more modern FPGAs [39]. The process of "lot" identifier generators in modern FPGAs may change in comparison to these older FPGAs, which may cause these identifiers to become ambiguous.

2.3.1 Experiment Setup

The experiment was conducted using seven lots of FPGAs acquired through different suppliers. The number of FPGAs within each lot vary from a single FPGA to 60 FPGAs, for a total of 102 FPGAs tested in this experiment. Each FPGA was programmed with a bitstream containing eight ROs that were executed in series. Each RO was manually routed within the FPGA using placement and routing constraints through a Universal Constraint File (UCF). In addition, the Tools for Open Reconfigurable Computing (TORC) framework was used to provide final placement of some instances[40].

Manual placement of each RO was required to eliminate the otherwise automatic optimization of the hardware logic layout and routing. The placement goals were designed to ensure that all regions of the FPGA die are used. This provides a solid, random, and statistically valid sampling of the FPGA die by the set of ROs . These representative eight ROs, denoted as I-VIII in this chapter, form a basis for a statistically representative fingerprint of the FPGA and the manufacturing lot to which it belongs. The ROs I-VIII are shown as delay 0-7 respectively in Figure 2.7.

For each FPGA, 128 measurements were collected. Sixteen measurement values were stored for each RO. The synchronous reference clock was operating at 33MHz. Each RO executed for a fixed period, and the number of cycles completed by the RO structure was recorded. The results were stored in an Chipscope Integrated Logic Analyzer (ILA) block and were recovered through ChipScope debugging tools[41]. Through the post processing the normal distribution of each FPGA and RO was verified.

The Xilinx Virtex BG560 Prototype Platform was used for all measurements in this study[42]. A Zero Insertion Force (ZIF) socket was used to mount the FPGA, ensuring stability for many of the external components of this system. The test apparatus used by this research

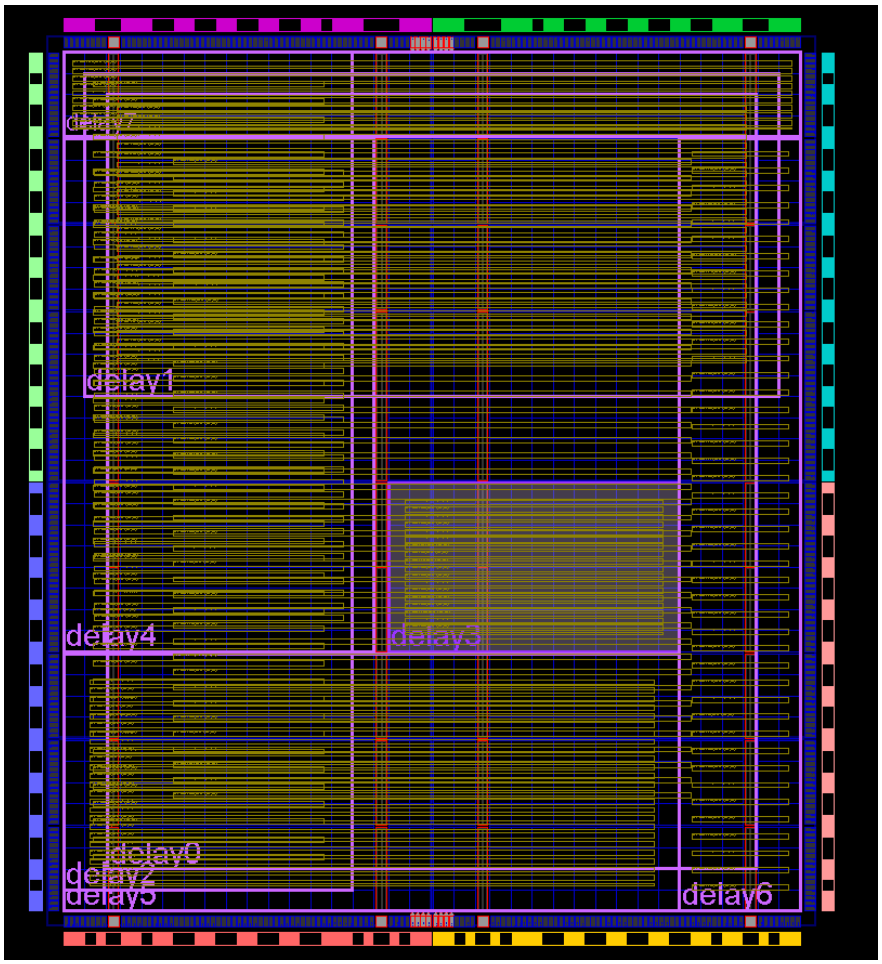


Figure 2.7: Sample PlanAhead layout for eight ring oscillators (RO)s. Each ring oscillator is constrained in the delay boxes.

Figure 2.8: Test apparatus for supply chain RO measurements.

is shown in Figure 2.8. By utilizing this platform, in addition to benchtop power supplies, potential variances due to irregularities in the Printed Circuit Board (PCB) board, voltage supply and regulators, and clock variance were minimized.

Baseline measurements

The baseline measurements were collected for each of the eight lots. The distribution of FPGAs in each lot is shown in Table 2.1. The lot identifiers were recorded from the manufacturer's markings on the FPGAs. Although this is not necessarily an indication of which wafer the FPGA came from, it is indicative of an approximate date for which the FPGA was manufactured.

Each FPGA was compared to all other FPGAs within the lot to ensure similarity of the measurements. Then each lot was compared against each other lot to evaluate if each lot could be uniquely identified.

Manufacture Lot ID	856a	551a	056a	576a	600a	639a	953a
Internal Lot ID	A	B	C	D	E	F	G
Number of Die	60	1	1	24	1	13	2

Table 2.1: Number of die per lot in experiment.

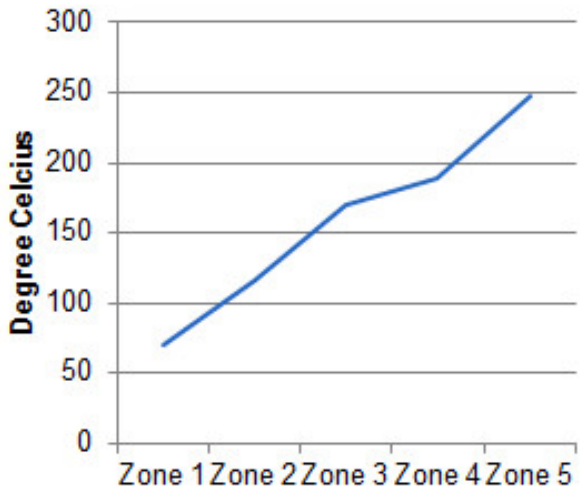


Figure 2.9: Lead free heating profile used in experiment.

Altering the Baseline

The baseline performance of the unaltered, new FPGA required simulating a rework that would be present in a secondary market part. It is known that traditional PUFs can become degraded through the aging process[43]. In this experiment, the ability to detect and quantify this aging of the FPGA is measured. If abnormal aging is detected, the device can be marked for further investigation as it may have been subject to a compromise in its integrity. One possible cause of this compromise could be a reflow if the FPGA was removed and replaced on the PCB board.

Modification to the FPGA RO was accomplished through subjecting the FPGAs under test to a heating profile common to a lead-free profile in a reflow oven (Figure 2.9). For this experiment, twelve FPGAs were selected from Lot A (856a). These FPGAs were placed in

the reflow oven and subjected to the temperature profile discussed above. In order to bake the FPGAs while retaining the integrity of the balls on the ball grid array of the FPGA, the chips were placed with their metal tops facing down and placed within a tray to pass through the reflow oven for five minutes. After each baking cycle the FPGAs were then placed in the test apparatus, they were measured, and the results were recorded. This simulates the heat profile that would be necessary to solder an FPGA to a PCB board, and then again the profile needed to remove the FPGA from the PCB board.

2.3.2 Measurement

In order to ensure that the ROs are statistically a normal Gaussian distribution, sixteen measurements were made for each of the eight ROs within an FPGA lot and the results were plotted on a Quantile-Quantile plot against a normal distribution (Figure 2.10). The resulting Q-Q plot in this example in one RO that has a coefficient of determination (R^2) of 0.9823, suggesting high degree of confidence that the results of ROs from FPGAs within a lot are normally distributed. The coefficient of determination for the remaining lots are shown in Table 2.2. All lots and ROs exhibited similar distributions, with larger lots contributing to increased confidence in a normal distribution. All outliers that are outside 10% of the median are thrown out.

2.4 Results and Analysis

The results presented in this framework are broken into two major sections. The first hypothesis test is to determine if the lot of manufacturing can be determined by evaluating the individual FPGA dies. The second hypothesis test is to determine if modifications to

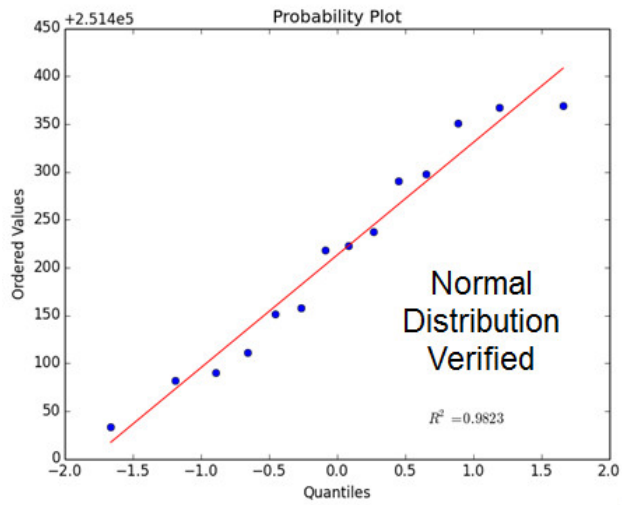


Figure 2.10: Quantile-Quantile plot across all measurements.

Table 2.2: Coefficient of Determination per RO per lot.

Lot ID	A	B	C	D	E	F	G
RO-I	0.9737	0.9244	0.9389	0.9470	0.9461	0.9613	0.9779
RO-II	0.9253	0.9539	0.9803	0.9739	0.8874	0.6313	0.9783
RO-III	0.9466	0.9697	0.9056	0.9537	0.9734	0.9831	0.9628
RO-IV	0.9844	0.9827	0.8362	0.8744	0.9663	0.9725	0.9801
RO-V	0.9020	0.9772	0.9703	0.9419	0.8084	0.9728	0.9760
RO-VI	0.9809	0.9742	0.9789	0.9786	0.9653	0.6830	0.9797
RO-VII	0.9165	0.9634	0.9764	0.7739	0.9308	0.9870	0.7824
RO-VIII	0.9418	0.7923	0.9787	0.9825	0.9759	0.9194	0.9823

an individual FPGA can be detected from placing the FPGA into a reflow oven to simulate tampering.

2.4.1 Lot Classification

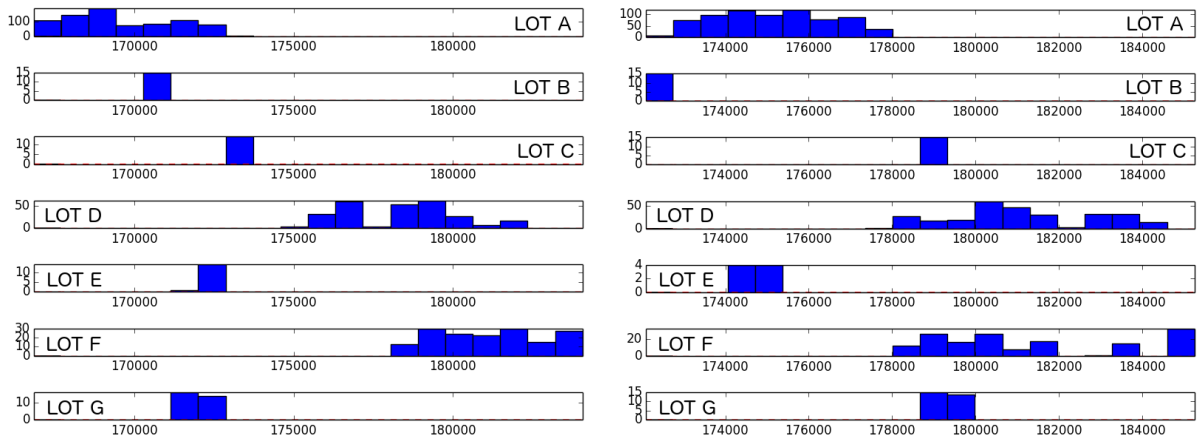


Figure 2.11: Results from seven different lots while running a) RO-IV and b) RO-VII The x-axis is frequency in Hertz and the y-axis is occurrence number per bin in the histogram. Note: the standard deviations for Lots A, D, and F are similar. Furthermore, notice how the lots migrate from different measurements. For example, Lot G is within the band of Lot A for RO-IV, but is within the band of Lots C, D, and F for RO-VII.

The results from each measurement were averaged per RO per lot. Figure 2.11 shows a histogram of the results from the measurement of all FPGAs per lot for RO-IV and RO-VII. The x-axis is defined as the the frequency of oscillation. Twenty possible bins were defined in this experiment with the range specified as the highest and lowest value across all ROs, all measurements. The y-axis is defined as the number of members of each bin. The specific mean values and standard deviation of the lot classifications are found in Table 2.3 and Table 2.4 respectively.

The data was then processed by a SVM in order to determine if the recorded data was classifiable[44]. The data from all measurements was categorized by the manufacturing

Table 2.3: Mean frequency [Hertz] per RO per lot.

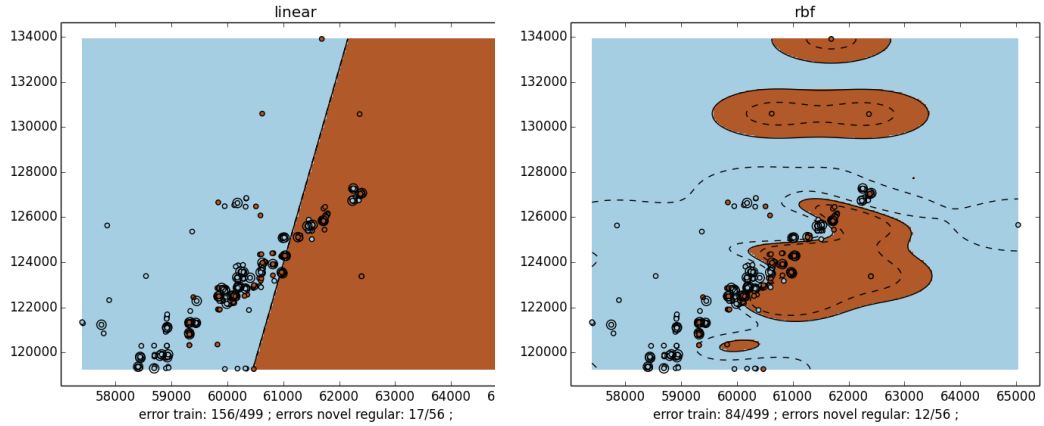
Lot ID	A	B	C	D	E	F	G
RO-I	169593	170644	173169	173169	178364	172067	181137
RO-II	173980	177534	178509	178509	186136	178990	187813
RO-III	175163	172149	178777	178777	181099	174728	181338
RO-IV	175009	173396	177198	177198	181000	174658	182629
RO-V	370429	375162	375162	377691	394339	380245	396661
RO-VI	345597	350500	350500	354269	364518	351980	368747
RO-VII	719379	720009	720009	734784	757801	731173	764563
RO-VIII	728314	721558	721558	735785	750384	724138	752367

Table 2.4: Standard Deviation [Hertz] per RO per lot.

Lot ID	A	B	C	D	E	F	G
RO-I	1666	40.5	141	1750	47.8	1763	250
RO-II	1267	18.0	62.3	1806	37.3	1775	528
RO-III	1380	48.5	82.4	1712	38.6	2340	502
RO-IV	1314	26.5	671	1773	52.6	2214	441
RO-V	2563	114	197	3637	351	4568	232
RO-VI	2633	105	83.6	2764	98.4	3420	250
RO-VII	5061	194	114	6359	566	6786	2291
RO-VIII	6035	533	262	6814	281	6608	4430

lot. Each of the ROs is considered a dimension within the SVM. In general there are 15 measurements per die per RO. The SVM is then trained with a random subset of the data per the training ratio. The remaining data is then used as the nominal data for testing the SVM.

The recorded values were processed through a Radial Basis Function (RBF) and Linear kernels within the SVM. Within the RBF, γ values of .000005, .000001, .0000005, and .0000001 were used to compensate for the large values of the RO frequency. Training ratios of .3, .5, and .7 were used. An example of the results in two dimensions is shown in Figure 2.12. In this example, RO-I is compared with RO-V. The linear function generates a line that performs the classification for anything above or below that line. Although this is basic, the RBF kernel is needed to perform an optimal classification. The γ variable for the RBF is a



(a) Linear (b) Radial Basis Function (RBF)

Figure 2.12: Results from the SVM utilizing .9 training ratio between Lots A (blue) and E(brown), and RO-I on the Y axis in Hertz and RO-V in Hertz on the X axis.

function of how tight the classification shapes are around the training data.

Because eight different ROs were used in this experiment, each RO is treated as a dimension. Each lot was compared against every other lot individually. The success ratios for both the training results are shown in the upper right half of the table and nominal classification results are shown in the bottom left half of the table.

The SVM is evaluated with a training ratio of {.3, .5, and .7} The gammas were varied from {.000001, .000005. .000001, .00000005 and .0000001} The RBFs with a γ less or equal to .000001 have the optimal results for all test runs. The specific results for these experiments is shown in Appendix A

The use of the SVM results in an error rate of less than 1% for most RBF kernels and a marginally higher error rate for the linear kernel. All of these results suggest that "lot" classification works. The specific values of the failure rates is shown for training values in Table 2.5 and for nominal values in Table 2.6.

Furthermore, it is likely that some of these lots came from the same foundry. Thus, the

results support classification within a foundry.

Table 2.5: Average failure rate of classification on training data for lot identity based classification.

	Linear	RBF .000005	RBF .000001	RBF .0000005	RBF .0000001
.3	62/2760	0/2760	0/2760	0/2760	0/2760
.5	76/4184	0/4184	0/4184	0/4184	0/4184
.7	128/6120	0/6120	0/6120	0/6120	0/6120

Table 2.6: Average failure rate of classification on nominal data for lot identity based classification.

	Linear	RBF .000005	RBF .000001	RBF .0000005	RBF .0000001
.3	128/6120	227/6120	100/6120	89/6120	84/6120
.5	70/4184	115/4184	54/4184	47/4184	43/4184
.7	45/2760	52/2760	17/2760	18/2760	11/2760

2.4.2 Die Life Cycle Classification

The chip baking that was conducted on this portion of the research both simulates the process of attaching and removing the FPGA to a PCB board. Generally as circuits age, the transition speed of the transistors slows. This is a function of a gradual degrading of the PN junction[45, 46]. Furthermore, it is hypothesized that the changes in the initial burn in of the chips is a function of impurities in the manufacturing process. It is further hypothesized that these impurities may vary over time in the manufacturing process and slight changes in the manufacturing environment.

When looking at all the data commingled there is a general trend of slower operation of the transistors; however, this data has such a large standard deviation, that patterns cannot be found by looking at the holistic lot. When the individual die are baselined and then evaluated, the effect is easier to detect. In Figure 2.13, it can be noted that the expected

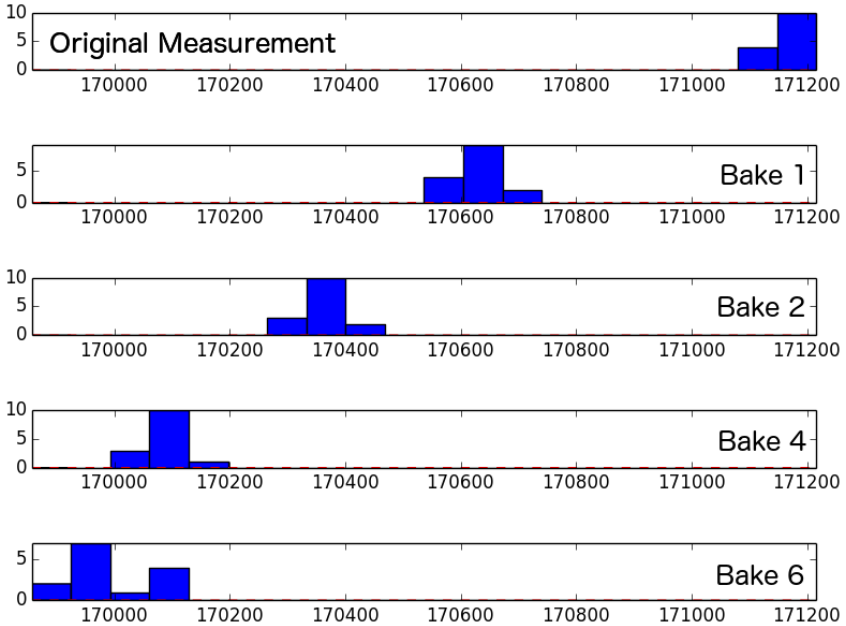


Figure 2.13: Results of heating cycles on Lot A. The x-axis is frequency in Hz and the y-axis is the occurrence number per bin of the histogram. Note: The general left shift after each baking cycle indicates a slowing of the ring oscillator.

left shift of the cycle frequency is observed indicating a slowdown of the RO. Again if the distribution is not normal, then suspicion should be raised about the integrity of the FPGAs.

With the data discovered in this phase, it is difficult to tell if an individual FPGA has been modified by just looking at a single measurement without a baseline. When looking across a lot of FPGAs which pass the previous experiment, deviations can be noticed which may indicate an outlier or a larger than expected deviation. This can help identify a suspect delivery of FPGAs.

Table 2.7: Mean frequency [Hertz] per RO per temperature exposure for twelve dies from Lot A.

Bake	0	1	2	4	6
RO-I	170184	170134	169380	169062	168985
RO-II	174419	174107	173570	172928	173114
RO-III	174743	174505	173835	173504	173398
RO-IV	175162	174818	174051	174111	173994
RO-V	370037	369792	368646	368749	367994
RO-VI	347217	346567	345300	344470	344803
RO-VII	722872	720716	718859	717264	717663
RO-VIII	728853	727760	723927	723409	722166

Table 2.8: Standard deviation [Hertz] RO per temperature exposure for twelve dies from Lot A.

Bake	0	1	2	4	6
RO-I	1407	1482	1307	1258	1365
RO-II	669	1359	750	771	945
RO-III	885	1261	905	860	890
RO-IV	1173	1395	778	1047	1252
RO-V	1386	2034	1566	1820	1654
RO-VI	2051	2713	2003	2424	2054
RO-VII	4007	5500	3508	4355	3826
RO-VIII	5309	6138	4985	4652	5231

The same SVM is used as described in Section 2.4.1. The SVM is evaluated at four different training ratios {.3, .5, .7 , .9} and three different γ values {.00005, .00001, .000005}. The linear SVM is also evaluated for .3. Its results were poor so it was not continued for further evaluations. The RBF function was adequate in identifying a less than 5% error rate on most RBF kernels with .00005, .00001 and .000005 γ values. The exact rate of classification failures is shown in Table 2.9 for training data and Table 2.10 for nominal data.

Table 2.9: Average failure rate of classification on training data for temperature exposure based classification.

	Linear	RBF .00005	RBF .00001	RBF .000005
.3	390/1080	1/1080	1/1080	1/1080
.5	N/A	0/1800	3/1800	6/1800
.7	N/A	0/2520	2/2520	8/2520
.9	N/A	0/3240	3/3240	10/3240

Table 2.10: Average failure rate of classification on nominal data for temperature exposure based classification.

	Linear	RBF .00005	RBF .00001	RBF .000005
.3	1098/2520	454/2520	287/2520	235/2520
.5	N/A	242/1800	165/1800	113/1800
.7	N/A	122/1080	92/1080	77/1080
.9	N/A	51/360	39/360	32/360

2.5 Future Work

There is substantial additional work that can conducted in creating both lot based measurements, and age based measurements of FPGAs. Further work can be completed with the SVM to discriminate against an unknown lot, and to increase the size of known lots, and the variety within FPGAs families. Furthermore, the value of γ will need to be further investigated based on the variance of the FPGAs within a lot per architecture.

Another application of this research can be in the protection of intellectual property without incurring the overhead of individual RO signatures as proposed in [26, 47]. In this scenario, only a specific lot of FPGAs would be authorized to execute through a run-time measurement. There is an additional investigation that is required to determine how much variance should be expected due to temperature and voltage changes. Initial research has been conducted by Maiti in identifying these environmental changes [31], but it did not go into attempting to utilize this change to determine the supply chain "health" as a function of these results.

In this research, a dedicated development board was used for all measurement. Additional research should be conducted to determine the effects of using different PCB boards for measurements. It is believed that if the clock and power for these measurements could be provided by high precision sources instead of traditional on board oscillators and the voltage supplies are of adequate quality, this research could be continued across existing soldered chips. If this step is successful, this measurement technique could migrate from an out-of-band measurement to an active measurement that could deny access to an FPGAs based system based on a failed "health" measurement. Furthermore, it could be used to predict an exact failure time.

PUFs are currently used in sensitive systems to provide a guarantee that the specified bit-stream will only operate on authorized silicon dies. This technique requires measuring of each FPGA that is produced, and then placing this measurement in an authorized measurement list. This research may provide an avenue to better characterize multiple manufactured lots, and authorize these lots while not authorizing many secondary market chips. Although this solution may not be perfect, it could provide a challenge to the counterfeit parts market.

2.6 Conclusion

This research provides a non-invasive technique to identify suspect FPGAs dies in the supply chain through FPGA fingerprinting analysis. This technique can be extended to evaluate FPGA life-cycle integrity against a known lot baseline. Lot integrity can be measured, and a baseline established. Once a baseline is formed, specific FPGAs that are marked as being from a known lot can be evaluated for truth. Lot baselines that are found to have a high standard deviation can be flagged for further investigation as a suspicious finding. Recycled FPGAs may be detected if the recycler does not ensure lot and environmental integrity in the newly packaged lots. FPGAs that have been exposed to strong environmental conditions such as heat, can also be detected if sufficient baselines exist.

Although this work focused on a small set of seven different lots, it is expected that with the approximately 1% error rate of the lot classification and the 10% error rate of the die classification procedures that additional FPGAs lots could be added to the set and proper classification could be observed. Furthermore, because of the continued variance in modern manufacturing processes, there is expected to be continued variance in the ROs[39]. It is unknown though if the pattern of classification by lot identification will hold up as the process of assigning these identifiers, and the potential variance between lot identifiers in specific modern fabrication processes is unknown.

This work is also is in contrast to both existing PUF research as the existing research seeks to identify individual dies, and maximize uniqueness and repeatability of these measurements. Furthermore, this work is focused on ensuring the values do not change as a result of aging, where this work is focused on measuring these changed values. It also is in contrast to foundry identification techniques, as they are focused on only the identity of the foundry, but not the lot, or changes within the die due to aging.

Table 2.11 compares and contrast various techniques for evaluating the integrity of FPGAs. The *Modified Ring Oscillator* proposed in this work fills a niche of non-destructive techniques that perform a course evaluation of the integrity of the FPGA. Although this work may not be sufficient on its own, it can be used in conjunction with other techniques to improve the current state of supply chain risk management.

This research contrasts with the existing body of research in focusing on the "health" of an FPGA with respect to its lot identity, and the number of times it has been thermally cycled to the levels of a flow or reflow of the solder.

PUFs are focused on individual die identification. If the proper infrastructure is in place to measure, maintain, and authenticate the PUFs, this can be a great tool to ensure an FPGAs has not been substituted in an existing system. PUFs do not provide protection against systems that the integrity of the FPGAs cannot be ensured before it is initially received, e.g. a gray market acquisition.

Physically identifiable artifacts such as DARPA Shield, serial numbers, and anti-tamper stickers are a cost effective way to track FPGAs, can provide some protection against gray market acquisitions if the artifact is attached at the factory. If this was not completed at the factory, then there is a gap in the integrity of the supply chain that cannot be easily mitigated.

Scanning electron microscopes can provide an in-depth analysis of a device, and can go looking for malicious modifications, but it is destructive and very time consuming to complete these types of analysis. Furthermore, it can't easily detect recycled parts that have no malicious defects, but may exhibit an earlier than expected life cycle failure.

Chemical analysis can detect if the chemical composition of the device has deviated from the expected compounds. This is useful to detect if the chip has been repackaged. Chemical

Table 2.11: Implementation challenges of counterfeit avoidance techniques.

Avoidance Technique	Reliability	Destruct-iveness	Implementation Difficulty	Detection Difficulty	Implementation Cost	Identification Mechanism
Physically Unclonable Functions	Medium	None	Medium	Low	High	Individual die
Physically Identifiable Artifact	Low	None	Low	Low	Low	Individual die
Scanning Electron Microscope	High	Yes	None	High	None	Difference between two dies
Chemical Analysis	Medium Low	Yes	Medium	High	Medium	Difference between two dies
Foundry Identification[37]	Medium Low	None	Low	Medium	Medium	Manufacturing foundry
Recycled FPGA Detection[36]	Medium Low	None	Low	Medium	Medium	Golden model comparison
Modified RO: Lot-ID	Medium Low	None	Low	Medium	Medium	Manufacturin lot
Modified RO: Life cycle	Medium Low	None	Low	Medium	Medium	Individual die re-flow

analysis could detect a reconstituted lot if the chemical makeup of the packaging changes between production runs. It will be unable to detect if the FPGAs has be reflowed multiple times.

Foundry identification techniques, if successfully ported to FPGAs, could potentially be used to identify the foundry of manufacturing. This data could be used to identify a reconstituted lot, but the granularity would be substantially reduced. With additional research, the granularity could be increased and provide similar results to this research.

The recycled FPGAs detection research completed similar research to this body of work. Its criteria for detecting recycled parts is an accelerated aging process where the chip is heated to 125°C for multiple hours to simulate extended use of the FPGA. This is in contrast to the 5 minute aging process used here to simulate the soldering of the chip onto a PCB board. Furthermore, the lot discrimination process is not analyzed in the recycled FPGAs detection work, but the technique could easily be ported to conduct this operation.

The above techniques are compared with an existing survey in Table 2.12[7]. The bottom four techniques all rely on detecting timing characteristics from a chip, thus the similar results. The difference is in the granularity of the use case.

Through the Modified Ring Oscillator research, presented here, the granularity can be applied to specific manufacturing lots, as opposed to foundry identification. In addition, the Modified Ring Oscillator can detect changes in a die that are exhibited during the initial burn in cycle, a use case that the referenced solutions do not address. This fills a niche of non-destructive techniques that perform a course evaluation of the integrity of the FPGA. Although this work may not be sufficient for complete lot identity, and ensuring supply chain integrity on its own, it can be used in conjunction with other techniques to mitigate some of the risks of the current state of supply chains.

Table 2.12: Assessment of different IDs [7]. First four rows are from reference

IDs	Reliability	Uniqueness	Unclonable	Manufacturability	Cost effectiveness	Ease-of-Use
QR codes (Physical Artifact)	Not verified	Medium	Medium	Not verified	Not verified	High
DNA markings	Low	Low	Low	Low	Low	Medium
Nanorods (Physical Artifact)	Not verified	High	High	Not verified	Not verified	Medium
Physically Uncloneable Functions	Not verified	High	High	Not verified	Not verified	Medium
Scanning Electron Microscope	High	N/A	N/A	N/A	Low	High
Chemical Analysis	Medium Low	Low	Low	High	Low	Low
Foundry Identification[37]	Medium Low	Low	Low	N/A	Medium	Medium High
Recycled FPGA Detection[36]	Medium Low	Medium Low	Medium	N/A	Medium	Medium High
Modified RO: Lot-ID	Medium Low	Medium Low	Medium	N/A	Medium	Medium High
Modified RO: Life cycle	Medium Low	Medium Low	Medium	N/A	Medium	Medium High

Chapter 3

FPGA/ASIC Design Recovery

Sections of this chapter were co-authored by E. Reilly and M. Schuyler.

Publications: B. Barrett, J. Couch, "Advanced DEsign Recovery (ADER)", *Defense Technical Information Center (DTIC)* [Not publicly available], 2014

J. Couch, E. Reilly, M. Schuyler, B. Barrett, "Functional Block Identification in Circuit Design Recovery", in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, McLean, VA, 2016, pp. 75-78. [Online]. Available: https://doi.org/10.1109/HST.2016.7495560

Provisional Patent: J. Couch, E. Reilly, M. Schuyler, B. Barrett, "Fuzzy Identification of Intellectual Property (IP) Blocks within Circuit Designs",November 26, 2015

3.1 Introduction

Due to the increased use of both soft and hard IP blocks within designs, the continued push to HLS development environments and the verification of post-synthesis gate-level netlists is not only uncommon, but challenging to conduct. Hardware designs that have compiled to remove all annotations and hierarchy are rarely verified to ensure that the intended design is free from any non-intended logic, either malicious or benign.

This problem was investigated by DARPA through the TRUST project[48, 5]. There was excellent research conducted through this effort, but it was primarily focused on detecting malicious modifications to a design. Although that is one use of design recovery techniques, the design recovery problem also routinely arises in compatibility testing, copyright investigations, and Trojan detection[49, 50, 51].

Many times in the design recovery process, custom ASICs and FPGAs are treated as black boxes in comparison to typical commodity hardware and software designs. This black box provides a gap in analyzing the entire system, though there are a few efforts that have focused on addressing this shortcoming. This work presents a technique to assist in this design recovery by identifying known blocks within a flattened and non-annotated FPGA or standard cell ASIC design. Once the blocks are identified and annotated, an engineer can work by hand through the remaining design to perform a complete design recovery. This research is not interested in steps that are required to get to this point. After the execution of this research, the design can then be further analyzed and simulated in other higher level tools such as simulators and emulators with stronger knowledge of the appropriate test vectors to exercise the portion of the design of interest.

The concepts of design recovery and reverse engineering are quite common throughout both the hardware and software industry. The first use of the term “reverse engineering” was by

M.G. Rekoff who defined it as "the process of developing a set of specifications for a complex hardware system by an orderly examination of specimens of that system"[52]. At each stage of a project, design recovery is conducted to gain a better understanding of the design, how it has been implemented, and how it can interact with external systems. For the purposes of this paper, the definitions of forward engineering, reverse engineering, design recovery, restructuring, and re-engineering are defined in [53].

Section 3.2 provides an overview of the current state of both hardware and software reverse engineering efforts. Section 3.3 provides an overview of current design recovery techniques for flattened netlists. Section 3.4 outlines the algorithms and techniques utilized in this research to advance the netlist recovery process. Section 3.5 outlines the results from these algorithms. Sections 3.6 provides an overview of future research and tools that can be enabled from this research.

3.2 Design Recovery Background

The design recovery problem is typically broken into three major areas, hardware and circuit board design recovery, custom ASIC and FPGA design recovery, and software design recovery. Insight into the operations of a system can be gleamed from all three of these techniques, but at times a critical piece of information is missing without a specific type of design recovery.

3.2.1 Hardware Approaches

Hardware reverse engineering has revolved around gaining an understanding of how a piece of hardware functions. The major categories of hardware reverse engineering include product

tear-downs, system-level analysis, process analysis, and circuit extraction[54]. Specifically through this process, higher-level concepts such as a mask Read Only Memory (ROM), identification of known blocks, and high-level interconnection are discovered. Various imaging processes are utilized, including polishing and optical microscopes, X-Ray machines, and SEM. Although these processes are time consuming, there is ample research in the field to guide a reverse engineer to a greater understanding of the design.

An entire industry has emerged as numerous entities seek to recreate design information from existing hardware devices. Some of these entities include Chipworks and UBM TechInsights. Because of this known capability, forward engineers of sensitive designs have begun to utilizing techniques to better protect their hardware design. This back-and-forth will continue as additional countermeasures and understanding of the hiding measures are better understood.

3.2.2 Software Approaches

The software reverse engineering industry has largely relied on the discovery of a binary file, and then reverse engineering its contents. The process of discovering this file could be as simple as it existing on easily readable media, to extraction from on-board memory, to extraction of the memory through the Joint Test Action Group (JTAG) interface or other common extraction techniques[55]. One of the most popular tools for assisting in this effort is IDA Pro, produced by Hex-Rays[56].

With IDA Pro, reverse engineers are able to take binary files compiled for many known Instruction Set Architecture (ISA) and identify the instructions that are utilized in the execution of this code. This process, known as disassembly, currently contains support around 100 ISAs[56]. With this information, it is possible to begin to understand the flow

of the program, when and where external Input Output (I/O) is accessed, and potential execution paths. Furthermore, IDAPro contains a decompiler that is capable of both 32-bit and 64-bit x86 and ARM ISAs. With the decompiler, instead of receiving a disassembled view of the assembly, the decompiler will attempt to regenerate pseudo-C to represent the binary file. This pseudo-C is semi-human readable, and can provide additional information to the reverse engineer on the operation of this binary file. With this type of knowledge, even higher abstractions can occur to identify known blocks of code to reduce the search space[57]. There are other tools available that can support alternative ISAs, but they are either not widely available or are not mature enough to assist in the reverse engineering of most binary files.

3.3 Background of HDL Design Recovery

Substantial research has been conducted on restoring flattened netlists into a higher level analysis. One recent major program in this field, the DARPA Integrity and Reliability of Integrated Circuits (IRIS), was tasked with determining the function of digital, analog and mixed signals integrated circuits[58].

The DARPA IRIS program expanded on the DARPA TRUST program to look into specific methods of design recovery within a netlist. Much of this work was on identifying exact replications of known circuits within the design through optimizations of the designs. Additional research in the field of Hardware Description Language (HDL) design recovery was focused on Finite State Machine (FSM) recovery and subgraph isomorphism solving.

3.3.1 Graph Sub-isomorphism

Graph sub-isomorphism is the traditional method of identifying common subgraphs within a larger graph[59]. The computational complexity of a complete subgraph isomorphism is NP-complete[60]. Numerous researchers have contributed efforts into accelerating this process.

Gemini, and its follow-on work SubGemini, are projects that utilize a Computer Aided Design (CAD) viewpoint of a circuit and identify subcircuits within a design[61, 62]. Gemini was one fo the first subgraph isomorphism applications applied to circuits for verification. It is specifically looking for missing nodes within a known graph. This work was extended by SubGemini utilizing a faster subgraph isomorphism algorithm. Although this research is older, its performance is on the order of identifying a 3000 node subcircuit within a 50000 node circuit in 30 minutes. This tool does not incorporate a fuzzy matching component.

SubIslands also extended the Sub-isomorphism research by combining graduated assignment matching techniques, error propagation and delayed decision making, and a bipartite graph labeling algorithm. This work is another attempt in optimizing the NP-complete subgraph isomorphism problem[63].

Other work such as hashing subgraphs, and fuzzy attributed graph matching have been utilized to assist circuit design recovery efforts[64, 65].

In general, all of these techniques rely on a solid understanding of the logical boundaries of a sub-circuit, and are looking for an exact match. Modern compilers, especially when working with standard cells or FPGA primitives, perform optimizations that are different on each compilation. These optimizations cause exact matching to falter, as many of the artifacts that are keypoints of the matching are no-longer in the same form.

3.3.2 Finite State Machine Recovery

One way to perform a higher level analysis is to attempt to reconstruct the FSMs of the design. Shi et al. published an algorithm to identify a state machine from a flattened netlist[66]. This technique is based on identifying the cycles within the netlist that are associated with appropriate registers. Once this is accomplished, these cycles are analyzed to determine if they could be a potential FSM. The recovery of these FSMs can be useful in determining the control flow of a design, especially within glue logic.

This technique is limited to identifying the FSMs without regard to the outlying computational and data paths. Although this technique is useful in the design recovery process, it is focusing on a different portion of the problem compared to the segmentation and pattern matching presented here.

3.3.3 Behavioral Pattern Matching

Li proposes a technique that involves matching sub-circuits against abstract library components and then performing data mining to aggregate these components into larger logic blocks[67]. This basic technique was introduced and later extended by Pelz[68, 69], but was extended in more recent research attempts[70]. This technique requires that the maximum clique be calculated for all comparisons to generate all signal correspondences. This is an NP-complete problem that Li admits is only scalable to a graph of a few hundred nodes. Furthermore, Li assumes that the circuit is already segmented into appropriate modules, something that would be carried through many compilation tools, but considered out of scope for this research.

Another approach for IP reuse is proposed by Zeng[71]. In this approach, variants of the subgraph isomorphism are evaluated, in which minimal performance improvements are iden-

tified. These improvements are based around optimizing logic equations within multiple sub-clusters of an IP block. This enables IP blocks to be reused in subsequent compile operations without recompiling the IP block. In this research, the IP blocks are generally limited to at most 1000 gates. In addition, this technique requires that the entire decomposition tree of an IP block be held in memory.

Additional approaches include gate matching by Whitman; formal method matching of designs by Li, and behavioral pattern mining by Li[72, 73, 74]. These library and mining techniques serve as an inspiration for the library within the matching section discussed in Section 3.4.3.

All of these approaches rely on a pattern matching identification of small sub-circuits, and then combine these sub-circuits together into larger circuits that can become functional units.

3.3.4 DARPA IRIS

Through the DARPA IRIS program, techniques were required to perform higher level analysis of netlists in order to identify malicious sub-circuits within the netlist. This program focused on accelerating the NP-complete subgraph isomorphism problem.

WordRev is a product of the DARPA IRIS program that looks to identify similar wire paths within a design and combine them into a bus[75]. The idea is that by taking these individual bit paths, they can be combined to provide word paths of execution. Although this technique reduces the computational complexity of the design, its implementation is still based on exact matching of designs.

Subramanyan, et. al., utilize the K-Cut algorithm for segmentation of the design graph, and then utilize aggregation, word propagation, module generation, and library matching to identify modules under the DARPA IRIS program[76]. The cutting algorithm is designed to

cut the graph into many small slices such that each bit-path is an individual cut. By cutting the graph this fine, the individual slices can then be recombined based on an aggregation algorithm. Once the words are rejoined, they can be recombined into larger word operations and finally matched to a library.

Luna Technologies also developed a system for identifying modifications between golden designs and implemented designs, similar to their works on DARPA TRUST[77].

These techniques, along with many other techniques developed under DARPA IRIS are effective in performing design recovery of ASICs with adequate bounding of the problem and computational resources. Techniques such as this are useful in Trojan detection, as the Trojan may be quite small. As designs are commonly being compiled with HLS tools, and the complexity of the architectures are increasing, the results from the compiler are generally nondeterministic. Without this deterministic result, most of these tools are unable to identify the sub-blocks of interest. Therefore, these exact matching techniques may not be as useful in future endeavors.

3.4 Approach

This research looks primarily at the interconnection of various nodes within a circuit as the attributes. The goals of this research are to perform intelligent classification of regions of a netlist in order to reduce the size of a design that must be subject to blind design recovery efforts. This is in contrast to many existing approaches which rely on existing "golden designs" and/or exact subgraph matching. To accomplish this, two major algorithms are used to perform segmentation of the design followed by a fuzzy matching algorithm. The overview of the components in this approach is shown in Figure 3.1.

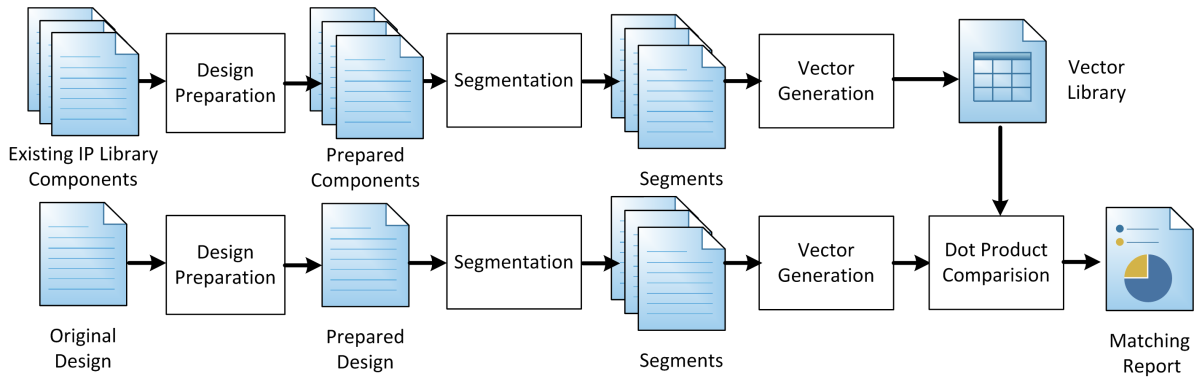


Figure 3.1: Full transformation of a circuit design.

3.4.1 Design Preparation

Before segmentation and matching can be conducted, the design must be prepared such that all logic and memory elements are broken into atomic units. These elements will become the nodes in the graph that represents the design. The nodes are also characterized into a limited character alphabet that contains nodes such as memory, logic, and external connections. Although this generalization is a lossy operation, it reduces the complexity of the problem for later analysis.

The design is also modified to identify all memory operations, and place an additional memory element after each memory element. This operation is conducted to ensure that for all segmentations there is a memory element on the boundary instead of leaving the indeterminism if the memory element should be segmented into Cluster A or Cluster B. This concept is outlined in Figure 3.2.

All net interconnections within the design are replaced with undirected edges in the graph. The choice of undirected edges is used as both the segmentation and matching algorithms selected expect undirected graphs. Furthermore, designs should be segmented if there are separate clock domains by the clock domain. Once this is completed, the weight of the remaining clock edges within a single domain should be reduced. All of these modifications

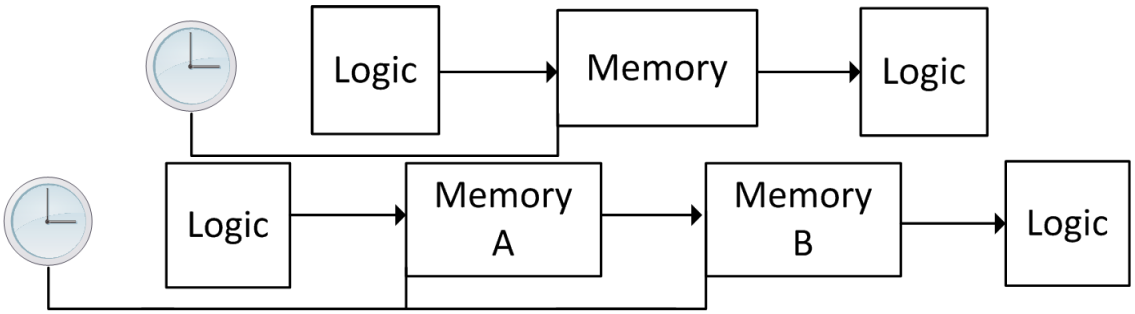


Figure 3.2: Insertion of the additional memory element: (a) original (b) modified.

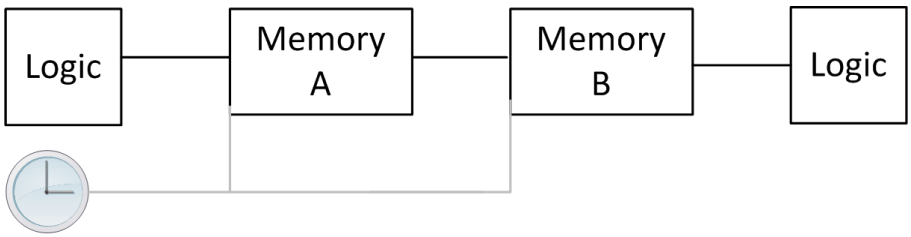


Figure 3.3: Full transformation of a circuit design with the devalued clock edges.

are shown in Figure 3.3. The specific details of the transformation process are located in the limited distribution Appendix B.

3.4.2 Segmentation

Although there are no exact rules for hardware design, the continued presence of core-based designs and HLS tools provide more and more designs that are based on clusters of highly interconnected logic and memory elements. A recent research effort in reconfigurable computing showed that many modern designs are highly interconnected clusters are then sparsely connected either to a common bus or to other highly interconnected clusters[78].

Given a graph as described above, the goal is to partition the graph such that the clusters are relatively high density, while the number of edges between clusters is sparse. The result is a number of smaller functional component designs, which may then be compared with other smaller known components. For graph partitioning, the Normalized Cut (Ncut) algorithm

developed by Shi and Malik for image segmentation is used [79].

Assume that for nodes u, v in a given graph, the edge between them has a weight denoted by $w(u, v)$. Let A, B be a partition of the vertices of the graph. Then, the cut of A and B is

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v). \tag{3.1}$$

The cut between two sets measures the number of edges that need to be removed from the graph to completely disconnect A and B. Similarly, Shi and Malik define an association value for a set A with the entire vertex set, V, as the following:

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t). \tag{3.2}$$

The association measures the total connection of set A within the given graph. Then, Ncut is defined as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V))}. \tag{3.3}$$

The first term can be thought of as the proportion of set A's connections in the graph that need to be removed in order to disconnect A from B. The second term is the same for set B. Minimizing this term results in a cut that considers both within cluster similarity and total dissimilarity of clusters in the graph. Though minimizing the Ncut is NP-complete, Shi and Malik provide an efficient algorithm to discover an approximate solution. See [79] for more details.

Because of the unknown optimal sizing of the clusters, multiple passes of the Ncut algorithm are completed across the design. All the clusters that are derived from this algorithm are recorded and placed into a library for the matching algorithm. The library clusters are sourced from common IP sources. They were recompiled optimized with different parameters

to introduce minor non-deterministic modifications to the test article. Clusters were also introduced into the library that were not part of the test design. This is done to simulate blind design recovery. The specific implementation details of the design segmentation algorithm are located in the Design Segmentation section in the limited distribution Appendix B.

3.4.3 Matching

Once the design is segmented, the resulting subgraphs, also referred to as clusters, are compared to existing designs in a library in order to identify functional components within the larger original design. Figure 3.4 demonstrates this concept.

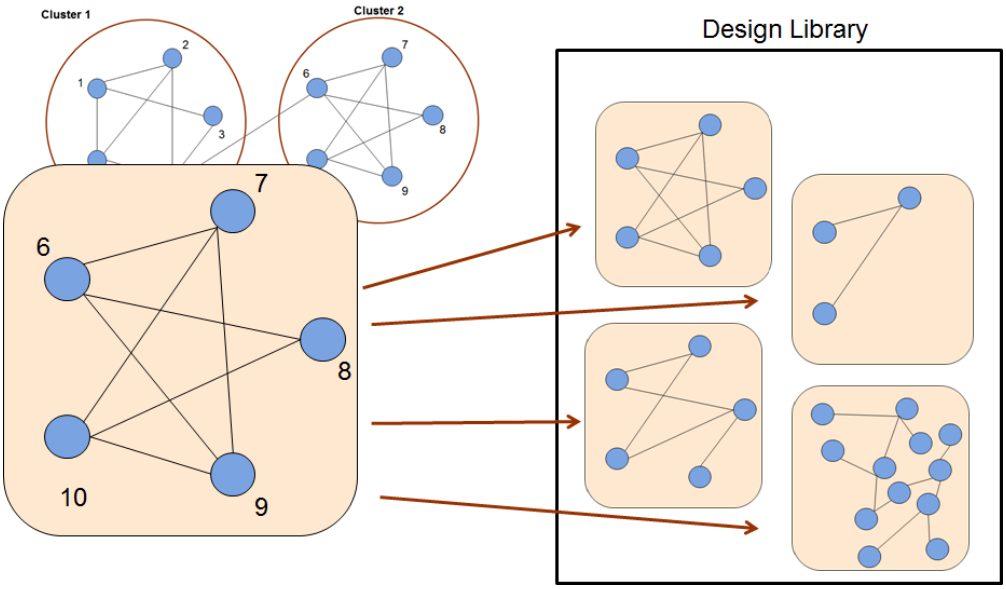


Figure 3.4: After segmentation, each cluster is compared with every design in a design library to identify known functional components.

In its exact form, this is the Graph Isomorphism problem, which is NP-complete. In this chapter, an approximate matching approach is taken by first embedding each graph into a low-dimensional vector space (a lossy step) for comparison. This is desirable to allow for slight differences in designs due to compiler or programmer differences. Specifically, the

concept of dot product representations is used for embedding. In 2010, Scheinerman and Tucker proposed this 'vertices to vectors' approach as a way of embedding the most relevant pieces of a graph's connectivity into low-dimensional vector space, allowing for geometric analysis methods [80]. Given a graph, each vertex i is assigned a vector x_i in $\mathbf{R}^d$ for a user selected dimension d such that, for distinct vertices i and j, the dot product $x_i \cdot x_j$ is approximately equal to the weight of the edge between i and j. For a Vertex i, the magnitude of x_i reflects how connected i is in the graph. The angle between two vectors corresponds to the connectivity of the vertices. For example, two vertices with little in common will be assigned vectors that are roughly orthogonal. For a simple example, see Figure 3.5. Scheinerman and Tucker propose an iterative algorithm to calculate the dot product representation of a graph in [80]. This algorithm was implemented in MATLAB with dimension equal to 50.

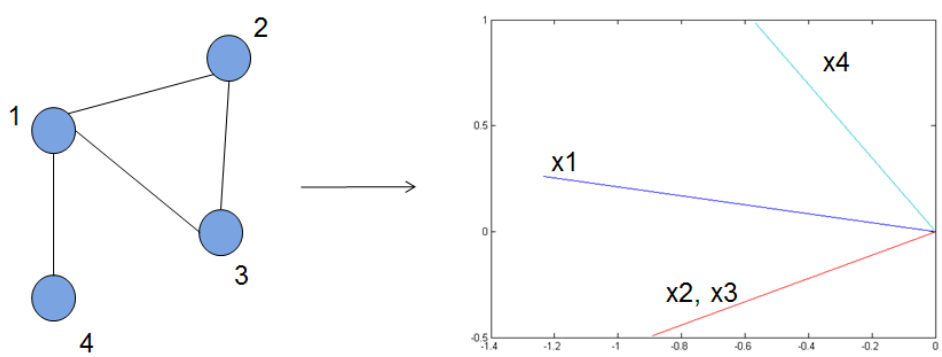


Figure 3.5: In this dot product representation example, each vertex in the graph is assigned a vector in $\mathbf{R}^2$. Notice that Vertices 2 and 3 are assigned the same vector since they play the same role in the graph. The vector for Vertex 4 is almost orthogonal to x_2 and x_3 since the only thing $2, 3, 4$ have in common is Vertex 1, whose vector is between the other vectors (and largest).

After embedding into $\mathbf{R}^d$, two sets of vectors are compared. This can be done with various techniques. In this chapter, the Procrustean transformation is used to align the sets of vectors and find the norm of the difference between the two sets. See Horn and Johnson's

text for more information [81].

By embedding each sub-circuit into a low dimensional space and using geometric analysis methods, the total circuit of over 10000 nodes with sub-circuits of over 1000 nodes were able to be analyzed. Not only is this an improvement over existing techniques, but it also allows for fuzzy design matching to account for compiler and minor design differences.

3.5 Results

For the purpose of these results, a synthetic, sample design was constructed that contained two different Advanced Encryption Standard (AES) cores, two different Fast Fourier Transform (FFT) cores, and two small microprocessor control blocks. These cores were pulled from publicly available IP libraries. This combined design resulted in over 14000 nodes in the design. Additional cores were added to the design library including other types of FFT and AES cores, along with other microprocessors to provide additional comparison vectors for the matching algorithm. The specific details of the cores and architectures is located in the limited distribution Appendix B.

3.5.1 Segmentation

The Ncut algorithm was applied to the overall design in order to identify relevant smaller functional elements in the design. The largest design considered has 14972 vertices and 62527 edges. A mid-2015 Macbook Pro computer with a 2.8 GHz core i7 processor with 16 GBs of RAM for all experiments. This design ran in 5.8 seconds.

Table 3.1 shows results of running the Ncut algorithm on the large design for 11 clusters. Notice that the AES blocks mostly segment into their own individual clusters. There is no

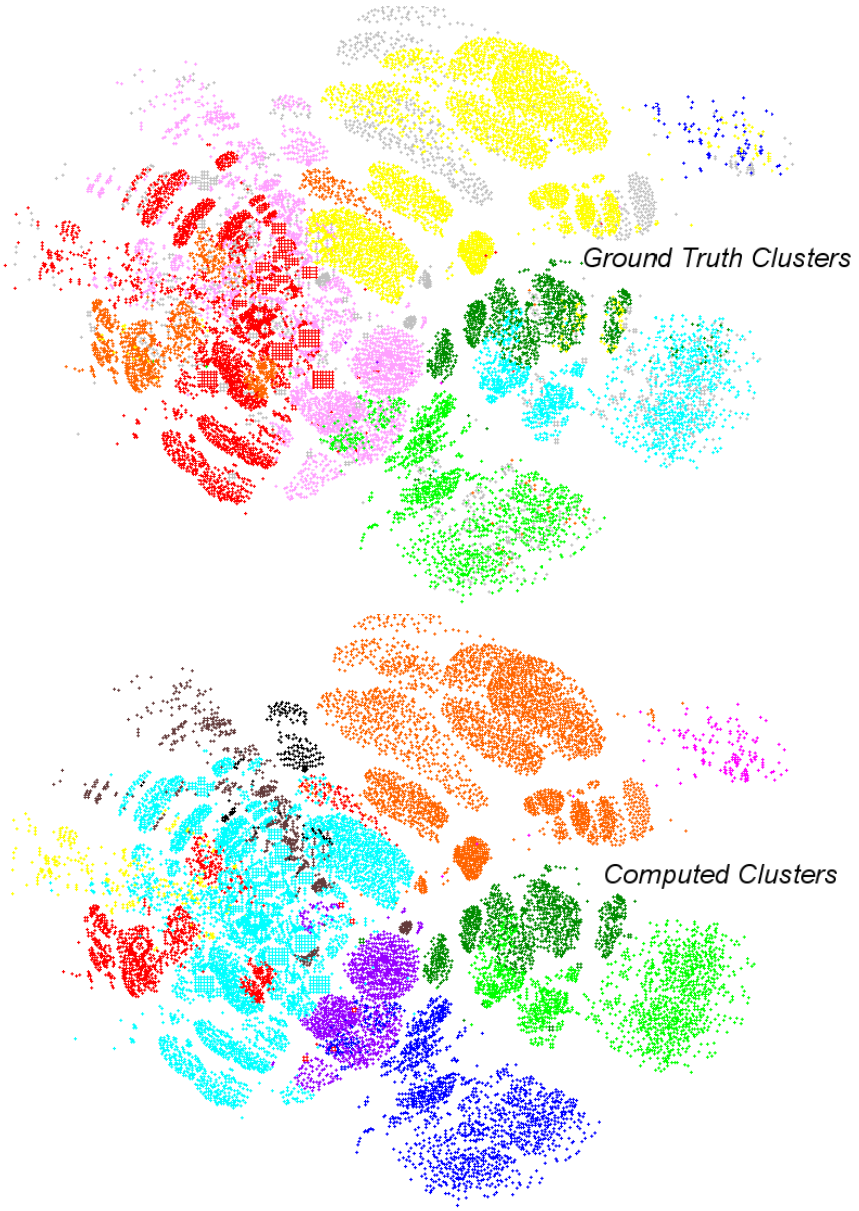


Figure 3.6: Graphical comparisons of ground truth segmentation(a) and the calculated segmentation (b) of the reference design. Note: The colors only indicate segmentation boundaries, not specific segments. For example, a strong match in the segmentation can be identified between the aqua ground truth cluster and the lime green computed cluster.

Table 3.1: Segmentation results.

Partition	Cls1	Cls2	Cls3	Cls4	Cls5	Cls6	Cls7	Cls8	Cls9	Cls10	Cls11
Control Core A	0	72	0	0	0	0	0	0	9	0	0
Control Core B	22	47	0	52	0	0	0	0	2800	0	768
Memory A	0	0	0	0	0	0	0	0	0	0	0
FFT A	0	0	0	0	239	0	1	0	4	12	2420
Memory B	0	0	0	0	0	0	0	0	0	0	2
FFT B	0	0	486	0	0	172	1	0	2	1179	739
Memory C	0	0	0	0	0	0	1	0	0	0	1
AES A	0	0	0	7	0	0	1149	1	0	0	0
AES B	0	0	0	927	0	0	24	10	0	0	2
AES C	3	0	0	1	0	0	0	1255	0	0	1
AES D	915	0	0	3	0	0	0	22	17	0	32

The large test design has 11 subdesigns totaling 14972 vertices. Dark blocks indicate segments where the majority of the original subdesign landed within the segmentation. Lighter blocks indicate partial matches.

concern with stray vertices here or there since the matching step is fuzzy. The other large pieces are two control cores and two FFTs. Each control core mostly has its own segment (clusters 2 and 9), though Core B loses a significant 768 nodes to Cluster 11, as well. The largest segments of the FFTs correspond to clusters 10 and 11. The FFT subdesigns are a bit messy. Each one has 'rogue segments' in clusters 3, 5, or 6 indicating that FFTs may inherently be composed of smaller functional units. In this case, the smaller functional FFT units should be included in the matching design library instead of the FFT itself. Nonetheless, this segmentation is sufficient for the matching algorithm to extract the main pieces of the overall design. A visualization of the matching algorithm is provided in Figure 3.6.

3.5.2 Matching

The output of the previous step is several smaller clusters, which are then compared to designs in a library. Embedding each cluster into $\mathbf{R}^d$ is the most time consuming step of the process. In the specific matching described in this chapter, the dimension $d = 50$ was used to provide sufficient information about each partition, while reducing the computational complexity. Calculating the dot product representation of a cluster with 1423 vertices and 3975 edges takes roughly 5.5 seconds using MATLAB.

The test library contains 16 designs containing all the known functional elements in the overall design, and designs similar to the known designs to test the robustness of the algorithm. The designs in the library have between 117 and 4608 vertices and between 371 and 10786 edges after the initial design preparations.

An initial test of the matching algorithm showed that each subdesign matches with itself in the design library. This was a necessary step since the dot product representation algorithm is lossy and iterative. Thus, results may vary slightly between runs of the same design. Through multiple iterations, it was discovered that the difference across runs is negligible and using dimension 50 still allows for ample differentiation between designs.

Next, the proposed end-to-end segmentation and matching algorithm was used. Each cluster in Table 3.1 was run through the proposed matching algorithm. The matching algorithm proved to be robust to incorrect vertex groupings. For example, Cluster 11 consists primarily of FFT A with significant contributions from Control Core B and FFT B. Nonetheless, the matching algorithm correctly identifies Cluster 11 as being most similar to the FFT A subdesign. Similarly, Cluster 9 was identified as Control Core B, and clusters 1, 4, 7, 8 were all identified as AES blocks. Thus, all the major elements of the design were identified, except for FFT B. Clusters 2, 3, 5, and 6 are all relatively small clusters and did not match

well with anything. In fact, this is what would be expected to happen since those clusters are small pieces broken off of the FFT blocks. This is where a reverse engineer would analyze the overall design to realize that is where the small unmatched blocks are coming from. The results of the matching are shown in Table 3.2. The matching column is the scalar sine difference between an exact match, and the attempted match of the library cluster vs the segmented cluster. Although there are mismatches in the final table, the majority are a function of the small sizes of the memory blocks that are subsets of the control cores and the FFT. Overall, these results are promising and demonstrate how robust and effective the proposed matching algorithm can be.

Table 3.2: Co-similarity of designs.

ID	True	Calculated	Match
1	AES D	AES	0.11994
2	Control Core A	Memory	0.34893
3	FFT B	AES	0.44194
4	AES B	AES	0.11806
5	FFT A	Memory	0.57913
6	FFT B	Memory	0.33189
7	AES A	AES	0.065193
8	AES C	AES	0.066218
9	Control Core B	Control Core	0.083697
10	FFT B	AES	0.080711
11	FFT A	FFT	0.057055

Matching against the library of known blocks. Here the recovered sub-circuits are compared against a slightly reduced library of possible sub-circuit designs.

3.6 Future Work

Through this work, it can be shown that both fuzzy segmentation and fuzzy matching can be used to assist in the design recovery process. By capturing common components from various

IP libraries, a reasonably sized library can be constructed to assist in the initial triage of future unknown designs. The machine time that is required to generate the library, and to do the corresponding matches is reasonable for most moderate size FPGA and ASIC designs of hundreds of thousands of nodes to millions of nodes. With the continued use of precompiled IP blocks, it is reasonable to expect the library to contain thousands to tens of thousands of vectors to match against while still having acceptable execution time. Furthermore, these vectors can be further evaluated and matched utilizing many methods that are common to software design recovery processes without revisiting the subgraph isomorphism problem.

This research was conducted within MATLAB without utilizing parallel execution. Many of the segmentation, library generation, and matching functions could be parallelized to increase execution speed. The MATLAB data structures generated in this research were not memory efficient, and could be improved through an optimized implementation.

3.7 Conclusion

This work alone does not provide all the tools and resources required to perform design recovery within an circuit, but it provides a new approach of combining a segmentation algorithm with a matching algorithm to assist in design recovery efforts when the subcomponents are not known a priori.

Table 3.3 provides an overview of the existing techniques discussed in Section 3.3.

The subgraph isomorphism class of techniques, Sub-isomorphism, Gemini, SubGemini, and SubIsland, all generally rely on finding a common clique within the circuit. Once this is complete, existing matching algorithms can be utilized. All of these techniques inherently rely on exact matches through subgraph isomorphism. Gemini and SubGemini were able to

Table 3.3: Comparison of circuit design recovery techniques.

Technique	Size of subgraph	Exact Match	Speed	Size Reduction	Specific or Generic
Graph Sub-isomorphism[60]	Hundred	Yes	Slow	No	Generic
Gemini[61]	Multi-Hundreds	Mostly	Slow	No	Generic
SubGemini[62]	Thousands	Mostly	Moderate	No	Generic
SubIslands[63]	Thousands	Yes	Moderate	No	Generic
Finite State Machine Identification[66]	Tens of Thousands	N/A	Moderate	Yes	Specific
Behavioral Pattern Matching[68][69][70]	Hundreds	Yes	Moderate	Yes	Generic
Data Mining[74] Library Generation[72]	Tens-Thousands	Varies	Moderate	No	Generic
IP-Reuse[71]	Hundreds	No	Slow	No	Generic
WordRev[75]	Tens of Thousands	N/A	Fast	Yes	Generic
K-Cut[76]	Hundards of Thousands	N/A	Fast	Yes	Generic
Segmentation and Matching	Tens of Thousands	No	Fast	Yes	Generic

easily identify divergence from the exact match, but this is not enough variance for blind design recovery, especially with compiled and optimized designs. The segmentation and matching work presented here is a replacement for these techniques on blind design recovery efforts.

The FSM Identification technique is a complement to the segmentation and matching work. It can be used to identify glue logic that may be in place between the cores that the segmentation and matching algorithms identify. This algorithm is speedy, but it is optimized to find specific subgraph structures.

The Behavioral Pattern Matching algorithms focus on the identification of small sub-circuits, and then build up to larger circuits. This relies on the ability to identify these small circuits, which may be lost in modern compilation and optimization tools. Thus, these tools generally require exact sub-circuit matches.

Data Mining is a complement to this research that can be used to improve the library of items to match against. It is utilized in conjunction with matching algorithms to increase the size and efficiency of the design library.

The IP-Reuse research provides the concept of switchable components that are functionally equivalent. This technique is useful for optimization, circuit diversification, and reverse engineering techniques. Although this technique provides functionally equivalent representations of a logic block, it does not optimize the search space to accelerate sub-circuit identification.

WordRev provides another complement to the segmentation and matching algorithm. WordRev could be applied to all library circuits and unknown circuits for reduction. some artifacts such as the importance of large buses could be lost in this effort though. This could lead to a reduction in the effectiveness of both segmentation and matching.

K-Cut provides another segmentation methods in comparison to the Ncut segmentation in general it supplies similar results; however, in its use with DARPA IRIS, it was optimized to generate smaller blocks than were used here.

Although substantial research exists on techniques to identify known or unknown sub-circuits within a design, they are generally focused on exact sub-circuit matches through subgraph isomorphism. These existing techniques are typically used for test vector generation and golden design verification. This research shows the viability of utilizing fuzzy matching by reducing the alphabet of the nodes, providing a reasonable segmentation method, and vectorizing the elements to provide a linear matching algorithm. The techniques shown above are aimed at assisting blind design recovery efforts by reducing the human time that is required to gain an understanding of the design while also providing reasonable computation time, of a few seconds, for the analysis with a precomputed library.

Chapter 4

Additional Sponsored Research

This portion of the dissertation is located at the Virginia Tech Secure Research Office.

Chapter 5

Conclusion

Researchers, designers, and even operators occasionally have the need to look deeper into a product to understand the underlying design. Both hardware and software vendors are continuing to restrict this access. This research provides multiple use cases where this knowledge can be utilized to enable capabilities that can increase the security posture of a product through verifications of both hardware and software designs. I hope that through this research, the community will appreciate the importance of low level research on available real devices and tools.

5.1 Contributions

This research focuses on many of the alternative capabilities that are available on FPGA and ASIC platforms. This section outlines the specific contributions previously discussed.

- *A method that utilizes asynchronous circuits in FPGAs to provide information about the supply chain characteristics of a specific die.* This method is capable of identifying

lifecycle events such as a reflow of an FPGA within an existing system. This may be indicative of tampering within the system, and can be a trigger for additional supply chain actions. Another lifecycle evaluation is to determine the homogeneity of a manufacturing batch of FPGAs. This is utilized to assist in detecting counterfeit FPGAs that could compromise the integrity of a supply chain.

- *A method for the segmentation and non-graph sub-isomorphism pattern matching of recovered hardware designs.* By re-purposing an existing imaging processing algorithm for image segmentation, recovered designs can be partitioned into a collection of segments that are easier to analyze in order to gain a better understanding of the original design. This segmenting reduces the computational complexity of current algorithms in this space. The current graph subgraph isomorphism algorithms for design recovery are computationally expensive, and are not optimized for the types of designs that are generated by commercial FPGA compilers and HLS tools for ASICs. This pattern matching method compares vectorized segments against a library of other well known segments in order to remove commonly known segments from the collection of unknown segments.

- Additional contributions that are listed in the addendum.

Through the supply chain integrity measurer, existing techniques for determining the uniqueness of an FPGA are being repurposed in order to provide unique contributions in manufacturing lot identification, and aging identification to provide a factor of confidence in the acquisition and life-cycle of FPGA based devices. Current research is limited to unique die identification and foundry identification. The Modified Ring Oscillator supply chain integrity measurer expands these existing capabilities to offer an active mechanism to measure the integrity of an FPGA. This combined with proper SVM analysis shows a 1% error rate for

lot classification and a less than 5% error rate for individual die tracking through the burn in process. These combined techniques provide a new active measurement mechanism, for supply chain protection techniques in a new problem area.

The process of blind design recovery for ASICs and FPGAs is currently limited to spot solutions that identify and collapse specific artifacts, or fall back to a variant of subgraph isomorphism. This research opens the door to a fuzzy approach of segmentation and matching that further constrains the computational time of subgraph identification. Although this research has only been applied to a limited subset of potential designs, the techniques that were utilized to not rely on any specific artifacts of a specific architecture or subgraph. Through additional research, the generality of this technique can be further realized.

All the techniques presented throughout this dissertation are proof-of-concepts. Additional work is needed to collect data about the ecosystems of operation, and outliers that may be present in these ecosystems. Without a better understanding of these ecosystems, there will be oversights which can easily cause errors for users. That all being said, this work provides additional motivation for continuing low level ASIC and FPGA research, as the field is not completely understood.

Bibliography

[1] F. M. Company, "Model T Facts — Ford Media Center." [Online]. Available: https://media.ford.com/content/fordmedia/fna/us/en/news/2013/08/05/model-t-facts.html

[2] S. Wozniak, "A Chat with Computing Pioneer Steve Wozniak : NPR." [Online]. Available: http://www.npr.org/templates/story/story.php?storyId=6167297

[3] G. Foundation, "gnu.org." [Online]. Available: https://www.gnu.org/gnu/gnu.html

[4] O. Scekic, "FPGA Comparative Analysis," Tech. Rep. [Online]. Available: http://home.etf.rs/~vm/os/vlsi/razno/Altera%20vs%20Xilinx%20(Scekic)%20.pdf

[5] D. Kaufman, "An Analytical Framework for Cyber Security," DARPA, Tech. Rep., 2011. [Online]. Available: http://www.darpa.mil/WorkArea/DownloadAsset.aspx?id=2147484449

[6] J. Villasenor and T. Mohammad, "The Hidden Dangers of Chop-Shop Electronics," *IEEE Spectrum*, 2013. [Online]. Available: http://spectrum.ieee.org/semiconductors/processors/the-hidden-dangers-of-chopshop-electronics

[7] M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits*. Cham: Springer International Publishing, 2015. [Online]. Available: http://link.springer.com/10.1007/978-3-319-11824-6

[8] S. Drimer, "Volatile FPGA design security a survey." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.105.3354

[9] G. Klutke, P. Kiessler, and M. Wortman, "A critical look at the bathtub curve," *IEEE Transactions on Reliability*, vol. 52, no. 1, pp. 125–129, mar 2003. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1179819

[10] D. Chardonnal, "Impacts of counterfeiting and piracy — I," 2011.

[11] IHS, "Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market," 2012. [Online]. Available: http://press.ihs.com/press-release/design-supply-chain/top-5-most-counterfeited-parts-represent-169-billion-potential-cha

[12] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, Feb. 2014. [Online]. Available: http://link.springer.com/10.1007/s10836-013-5430-8

[13] ISO/IEC, "11889-1:2009 - Information technology – Trusted Platform Module." [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=50970

[14] G. Gruman, "The truth about Samsung Knox for Android security — InfoWorld," 2013. [Online]. Available: http://www.infoworld.com/article/2612731/mobile-security/mobile-security-the-truth-about-samsung-knox-for-android-security.html

[15] S. Trimberger and J. Moore, "FPGA Security: Motivations, Features, and Applications," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1248–1265, Aug 2014.

[16] H. Livingston, "Securing the DOD Supply Chain from the Risks of Counterfeit Electonic Components," *BAE Systems*, 2010.

[17] K. AGMA, "Effective channel management is critical in combating the gray-market and increasing technology companies bottom line. Retrieved July 27, 2011," 2008. [Online]. Available: https://scholar.google.com/scholar?cluster=16967389256156197837&hl=en&as_sdt=20000005&sciodt=0,21#0

[18] M. Pecht and S. Tiku, "Bogus!" *IEEE Spectrum*, vol. 43, no. 5, pp. 37–46, May 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1628506

[19] D. Duhan and M. Sheffet, "Gray markets and the legal status of parallel importation," *The Journal of Marketing*, 1988. [Online]. Available: http://www.jstor.org/stable/1251451

[20] S. Gallagher, "FTDI on counterfeit chip bricking: Our intentions were honorable — Ars Technica," 2015. [Online]. Available: http://arstechnica.com/information-technology/2014/10/ftdis-anti-counterfeiting-efforts-sit-between-a-rock-and-a-hard-place/

[21] K. Bernstein, "Trusted Integrated Circuits (TRUST)," DARPA, Tech. Rep., 2015. [Online]. Available: http://www.darpa.mil/Our_Work/MTO/Programs/Trusted_Integrated_Circuits_%28TRUST%29.aspx

[22] ——, "Supply Chain Hardware Integrity for Electronics Defense (SHIELD)," 2014.

[23] L. Li, "Technology designed to combat fakes in the global supply chain," *Business Horizons*, vol. 56, no. 2, pp. 167 – 177, 2013, {SPECIAL} ISSUE: {PROTECTING} {YOUR} {INTELLECTUAL} {PROPERTY} {RIGHTS}. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007681312001668

[24] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 148–160.

[25] A. Maiti and P. Schaumont, "Improving the quality of a physical unclonable function using configurable ring oscillators," in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*. IEEE, 2009, pp. 703–707.

[26] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, *FPGA intrinsic PUFs and their use for IP protection*. Springer, 2007.

[27] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*. Springer, 2010, pp. 3–37.

[28] S. Morozov, A. Maiti, and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," in *Reconfigurable Computing: Architectures, Tools and Applications*. Springer, 2010, pp. 382–387.

[29] X. Wang and M. Tehranipoor, "Novel physical unclonable function with process and environmental variations," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10. 3001 Leuven, Belgium: European Design and Automation Association, 2010, pp. 1065–1070. [Online]. Available: http://dl.acm.org/citation.cfm?id=1870926.1871187

[30] M.-D. Yu and S. Devadas, "Secure and Robust Error Correction for Physical Unclonable Functions," *Design Test of Computers, IEEE*, vol. 27, no. 1, pp. 48–65, Jan 2010.

[31] A. Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive," *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, Oct. 2010. [Online]. Available: http://link.springer.com/10.1007/s00145-010-9088-4

[32] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*. IEEE, Jun. 2008, pp. 67–70. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4559053

[33] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual conference on Design automation - DAC '07*. New York, New York, USA: ACM Press, Jun. 2007, p. 9. [Online]. Available: http://dl.acm.org/citation.cfm?id=1278480.1278484

[34] J. Graf, J. Hallman, and S. Harper, "Trust in the FPGA Supply Chain Using Physically Unclonable Functions," in *GoMACTech 2010*, November 2010.

[35] M. Yu and S. Devadas, "Recombination of physical unclonable functions," 2010. [Online]. Available: http://dspace.mit.edu/handle/1721.1/59817

[36] H. Dogan, D. Forte, and M. M. Tehranipoor, "Aging analysis for recycled FPGA detection," in *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, oct 2014, pp. 171–176. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6962099

[37] J. B. Wendt, F. Koushanfar, and M. Potkonjak, "Techniques for Foundry Identification," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference - DAC '14*. New York, New York, USA: ACM Press, jun 2014, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=2593069.2593228

[38] *Virtex 2.5 V Field Programmable Gate Arrays*, Ds003 ed., Xilinx Inc. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds003.pdf

[39] P. Sedcole and P. K. Cheung, "Within-die delay variability in 90nm FPGAs and beyond," in *2006 IEEE International Conference on Field Programmable Technology*. IEEE, dec 2006, pp. 97–104. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4042421

[40] N. Steiner, A. Wood, H. Shojaei, J. Couch, P. Athanas, and M. French, "Torc: towards an open-source tool flow," in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*. ACM, 2011, pp. 41–44.

[41] *ChipScope Pro 10.1 Software and Cores User Guide*, UG029 ed., Xilinx Inc., 2008. [Online]. Available: http://www.xilinx.com/ise/verification/chipscope_pro_sw_cores_10_1_ug029.pdf

[42] *Virtex BG560 Prototype Platform*, Xilinx Inc. [Online]. Available: {}http://www.xilinx.com/products/boards-and-kits/hw-afx-bg560-100.html

[43] A. Maiti and P. Schaumont, "The Impact of Aging on a Physical Unclonable Function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1854–1864, sep 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6612683

[44] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[45] S. Keane and C. Kim, "Transistor aging," *IEEE Spectrium*, 2011. [Online]. Available: http://spectrum.ieee.org/semiconductors/processors/transistor-aging

[46] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," in *VLSI Test Symposium, 2007. 25th IEEE*, May 2007, pp. 277–286.

[47] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, June 2008, pp. 67–70.

[48] S. Adee, "The hunt for the kill switch," *Spectrum, IEEE*, 2008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4505310

[49] G. R. Ignatin, "Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility," *University of Pennsylvania Law Review*, vol. 140, no. 5, pp. 1999–2050, 1994. [Online]. Available: http://www.jstor.org/stable/3312440?seq=1#page_scan_tab_contents

[50] J. Kumagai, "Chip detectives [reverse engineering]," *IEEE Spectrum*, vol. 37, no. 11, pp. 43–48, 2000. [Online]. Available: http://dl.acm.org/citation.cfm?id=369402.369415

[51] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5342394

[52] M. G. Rekoff, "On reverse engineering," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 2, pp. 244–252, Mar. 1985. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6313354

[53] E. Chikofsky and J. Cross, "Reverse engineering and design recovery: a taxonomy," *IEEE Software*, vol. 7, no. 1, pp. 13–17, Jan. 1990. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=43044

[54] C. Clavier and K. Gaj, Eds., *Cryptographic Hardware and Embedded Systems - CHES 2009*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5747. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-04138-9

[55] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," *...Hardware and Embedded Systems-CHES 2009*, 2009. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-04138-9_26

[56] C. Eagle, *The IDA Pro Book, 2nd Edition: The Unofficial Guide to the World's Most Popular Disassembler*. No Starch Press, 2011. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=3nPAM3AZ1foC&pgis=1

[57] M. F. et. al. Lospinuso, "Apparatus and Method for Identifying Related Code Variants in Binaries," Mar. 2014. [Online]. Available: https://www.google.com/patents/US20140068768

[58] K. Bernstein, "Integrity and Reliability of Integrated Circuits (IRIS)," 2011. [Online]. Available: http://www.darpa.mil/program/integrity-and-reliability-of-integrated-circuits

[59] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, jan 1976. [Online]. Available: http://dl.acm.org/citation.cfm?id=321921.321925

[60] M. Garey and D. Johnson, "Computers and intractability; a guide to the theory of NP-Completeness," 1979. [Online]. Available: http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=COLPOS.xis&method=post&formato=2&cantidad=1&expresion=mfn=005118

[61] C. Ebeling, "GeminiII: a second generation layout validation program," in *[1988] IEEE International Conference on Computer-Aided Design (ICCAD-89) Digest of Technical Papers*. IEEE Comput. Soc. Press, 1988, pp. 322–325. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=122520

[62] M. Ohlrich, C. Ebeling, E. Ginting, and L. Sather, "Design Automation, 1993. 30th Conference on," pp. 31–37, 1993.

[63] N. Rubanov, "SubIslands: the probabilistic match assignment algorithm for subcircuit recognition," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 1, pp. 26–38, Jan. 2003. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1158251

[64] T. Portegys, "General Graph Identification By Hashing," *arXiv preprint arXiv:1512.07263*, 2015. [Online]. Available: http://arxiv.org/abs/1512.07263

[65] M. Sugeno, "A similarity measure of fuzzy attributed graphs and its application to object recognition," in *Proceedings of IEEE 5th International Fuzzy Systems*, vol. 2. IEEE, 1996, pp. 767–772. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=552277

[66] Y. Shi, C. W. Ting, B. H. Gwee, and Y. Ren, "A highly efficient method for extracting FSMs from flattened gate-level netlist," in *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, 2010, pp. 2610–2613.

[67] W. Li, Z. Wasson, and S. Seshia, "Reverse engineering circuits using behavioral pattern mining," *Hardware-Oriented Security and ...*, pp. 83–88, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6224325

[68] G. Pelz and U. Roettcher, "Circuit comparison by hierarchical pattern matching," in *1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers*. IEEE Comput. Soc. Press, 1991, pp. 290–293. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=185256

[69] ——, "Pattern matching and refinement hybrid approach to circuit comparison," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 2, pp. 264–276, 1994. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=259949

[70] W. Li, Z. Wasson, and S. A. Seshia, "Reverse engineering circuits using behavioral pattern mining," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, Jun. 2012, pp. 83–88. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6224325

[71] K. Zeng, "Enhancing Productivity with Back-End Similarity Matching of Digital Circuits for IP Reuse," Jun. 2013. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/23144

[72] J. Whitham, "A Graph Matching Search Algorithm for an Electronic Circuit Repository," *Univ. of York*, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.5417&rep=rep1&type=pdf

[73] W. Li, "Formal Methods for Reverse Engineering Gate-Level Netlists," 2013. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-222.pdf

[74] W. Li, Z. Wasson, and S. A. Seshia, "Reverse engineering circuits using behavioral pattern mining," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, Jun. 2012, pp. 83–88. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6224325

[75] W. Li and A. Gascon, "Wordrev: Finding word-level structures in a sea of bit-level gates," *2013 IEEE International Symposium on Hardware-Oriented Security and Trust*, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6581568

[76] P. Subramanyan, N. Tsiskaridze, W. Li, A. Gascon, W. Y. Tan, A. Tiwari, N. Shankar, S. A. Seshia, and S. Malik, "Reverse Engineering Digital Circuits Using Structural and Functional Analyses," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 63–80, Mar. 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6683016

[77] J. Graf, S. Harper, and L. Lerner, "The Integrity of FPGA Designs: Capabilities Enabled by Unlocking Bitstreams and 3rd-Party IP," Mar. 2012. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA561734

[78] T. Frangieh and P. Athanas, "A design assembly framework for FPGA back-end acceleration," in *2012 International Conference on Reconfigurable Computing and FPGAs*. IEEE, Dec. 2012, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6416718

[79] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000. [Online]. Available: http://www.computer.org/portal/web/csdl/doi?doc=abs/proceedings/cvpr/1997/7822/00/78220731abs.htm

[80] E. Scheinerman and K. Tucker, "Modeling graphs using dot product representations," *Comput Stat*, vol. 25, 2010. [Online]. Available: http://link.springer.com/article/10.1007%2Fs00180-009-0158-8#page-1

[81] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

Appendices

Appendix A

Results from SVM experiments

Below are the results of the supply chain integrity measurer utilized for both lot classification and individual die life cycle classification. The upper triangle of the table is the training data error ratios, and the lower triangle of the table is the nominal data error rations. The main diagonal of the table is not defined as this is not a proper comparison.

A.1 Lot Classification

Below are the results of the SVM that was applied as each of the 8 lots were compared to each other.

Table A.1: SVM success ratios with linear kernel, .7 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/882	0/262	0/262	72/388	0/262	0/273
B	0/378	N/A	0/640	0/640	0/766	7/340	26/651
C	0/113	0/275	N/A	0/21	0/147	0/21	0/31
D	0/113	1/275	0/9	N/A	0/147	0/21	0/31
E	35/167	0/329	0/63	0/63	N/A	0/147	0/157
F	0/113	1/275	0/9	0/9	0/63	N/A	0/31
G	0/117	8/279	0/14	0/14	0/68	0/14	N/A

Table A.2: SVM success ratios with RBF kernel, .000005=γ, .7 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/882	0/262	0/262	0/388	0/262	0/273
B	15/378	N/A	0/640	0/640	0/766	0/340	0/651
C	1/113	0/275	N/A	0/21	0/147	0/21	0/31
D	0/113	1/275	2/9	N/A	0/147	0/21	0/31
E	8/167	12/329	0/63	0/63	N/A	0/147	0/157
F	2/113	0/275	2/9	0/9	1/63	N/A	0/31
G	3/117	1/279	2/14	0/14	2/68	0/14	N/A

Table A.3: SVM success ratios with RBF kernel, .000001=γ, .7 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/882	0/262	0/262	0/388	0/262	0/273
B	3/378	N/A	0/640	0/640	0/766	0/340	0/651
C	0/113	0/275	N/A	0/21	0/147	0/21	0/31
D	0/113	1/275	1/9	N/A	0/147	0/21	0/31
E	5/167	5/329	0/63	0/63	N/A	0/147	0/157
F	0/113	0/275	0/9	1/9	0/63	N/A	0/31
G	0/117	0/279	0/14	1/14	0/68	0/14	N/A

Table A.4: SVM success ratios with RBF kernel, .0000005=γ, .7 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/882	0/262	0/262	0/388	0/262	0/273
B	3/378	N/A	0/640	0/640	0/766	0/340	0/651
C	0/113	0/275	N/A	0/21	0/147	0/21	0/31
D	0/113	1/275	1/9	N/A	0/147	0/21	0/31
E	5/167	5/329	0/63	1/63	N/A	0/147	0/157
F	0/113	0/275	0/9	0/9	0/63	N/A	0/31
G	0/117	0/279	0/14	2/14	0/68	0/14	N/A

Table A.5: SVM success ratios with RBF kernel, .0000001=γ, .7 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/882	0/262	0/262	0/388	0/262	0/273
B	2/378	N/A	0/640	0/640	0/766	0/340	0/651
C	0/113	0/275	N/A	0/21	0/147	0/21	0/31
D	0/113	0/275	0/9	N/A	0/147	0/21	0/31
E	6/167	2/329	0/63	0/63	N/A	0/147	0/157
F	0/113	1/275	0/9	0/9	0/63	N/A	0/31
G	0/117	0/279	0/14	0/14	0/68	0/14	N/A

Table A.6: SVM success ratios with linear kernel, .5 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/630	0/187	0/187	51/277	0/187	0/195
B	2/630	N/A	0/457	0/457	0/547	5/457	20/65
C	0/188	0/458	N/A	0/15	0/105	0/15	0/22
D	0/188	1/458	0/15	N/A	0/105	0/15	0/22
E	50/278	0/548	0/105	0/105	N/A	0/105	0/112
F	0/188	3/458	0/15	0/15	0/105	N/A	0/22
G	0/195	13/465	0/23	1/23	0/113	0/23	N/A

Table A.7: SVM success ratios with RBF kernel, .000005=γ, .5 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/630	0/187	0/187	0/277	0/187	0/195
B	35/630	N/A	0/457	0/457	0/547	0/457	0/65
C	4/188	0/458	N/A	0/15	0/105	0/15	0/22
D	2/188	2/458	2/15	N/A	0/105	0/15	0/22
E	15/278	24/548	1/105	2/105	N/A	0/105	0/112
F	1/188	1/458	2/15	3/15	6/105	N/A	0/22
G	5/195	1/465	2/23	4/23	3/113	2/23	N/A

Table A.8: SVM success ratios with RBF kernel, .000001=γ, .5 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/630	0/187	0/187	0/277	0/187	0/195
B	19/630	N/A	0/457	0/457	0/547	0/457	0/65
C	1/188	0/458	N/A	0/15	0/105	0/15	0/22
D	1/188	1/458	1/15	N/A	0/105	0/15	0/22
E	9/278	10/548	1/105	2/105	N/A	0/105	0/112
F	0/188	0/458	1/15	1/15	1/105	N/A	0/22
G	0/195	0/465	1/23	4/23	0/113	1/23	N/A

Table A.9: SVM success ratios with RBF kernel, .0000005=γ, .5 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/630	0/187	0/187	0/277	0/187	0/195
B	16/630	N/A	0/457	0/457	0/547	0/457	0/65
C	1/188	0/458	N/A	0/15	0/105	0/15	0/22
D	0/188	1/458	1/15	N/A	0/105	0/15	0/22
E	10/278	8/548	1/105	1/105	N/A	0/105	0/112
F	0/188	0/458	0/15	1/15	1/105	N/A	0/22
G	0/195	0/465	1/23	4/23	0/113	1/23	N/A

Table A.10: SVM success ratios with RBF kernel, .0000001=γ, .5 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/630	0/187	0/187	0/277	0/187	0/195
B	2/630	N/A	0/457	0/457	0/547	0/457	0/65
C	0/188	0/458	N/A	0/15	0/105	0/15	0/22
D	0/188	1/458	0/15	N/A	0/105	0/15	0/22
E	32/278	2/548	0/105	0/105	N/A	0/105	0/112
F	0/188	4/458	0/15	0/15	0/105	N/A	0/22
G	0/195	0/465	0/23	2/23	0/113	0/23	N/A

Table A.11: SVM success ratios with linear kernel, .3 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/378	0/112	0/112	36/166	0/112	0/117
B	6/882	N/A	0/274	0/274	0/328	8/274	18/279
C	0/263	2/641	N/A	0/9	0/63	0/9	0/13
D	0/263	2/641	0/21	N/A	0/63	0/9	0/13
E	82/389	0/767	0/147	0/147	N/A	0/63	0/67
F	0/263	9/641	0/21	0/21	0/147	N/A	0/13
G	0/273	25/651	0/32	2/32	0/158	0/32	N/A

Table A.12: SVM success ratios with RBF kernel, .000005=γ, .3 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/378	0/112	0/112	0/166	0/112	0/117
B	86/882	N/A	0/274	0/274	0/328	0/274	0/279
C	5/263	4/641	N/A	0/9	0/63	0/9	0/13
D	3/263	2/641	2/21	N/A	0/63	0/9	0/13
E	24/389	35/767	2/147	3/147	N/A	0/63	0/67
F	4/263	3/641	2/21	4/21	6/147	N/A	0/13
G	17/273	1/651	2/32	4/32	15/158	7/32	N/A

Table A.13: SVM success ratios with RBF kernel, .000001=γ, .3 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/378	0/112	0/112	0/166	0/112	0/117
B	23/882	N/A	0/274	0/274	0/328	0/274	0/279
C	1/263	1/641	N/A	0/9	0/63	0/9	0/13
D	2/263	1/641	1/21	N/A	0/63	0/9	0/13
E	16/389	15/767	1/147	1/147	N/A	0/63	0/67
F	2/263	1/641	1/21	4/21	1/147	N/A	0/13
G	15/273	0/651	1/32	4/32	8/158	1/32	N/A

Table A.14: SVM success ratios with RBF kernel, .0000005=γ, .3 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/378	0/112	0/112	0/166	0/112	0/117
B	20/882	N/A	0/274	0/274	0/328	0/274	0/279
C	1/263	1/641	N/A	0/9	0/63	0/9	0/13
D	2/263	1/641	1/21	N/A	0/63	0/9	0/13
E	17/389	14/767	1/147	2/147	N/A	0/63	0/67
F	1/263	0/641	0/21	4/21	1/147	N/A	0/13
G	15/273	0/651	0/32	4/32	3/158	1/32	N/A

Table A.15: SVM success ratios with RBF kernel, .0000001=γ, .3 training ratio

Lot ID	A	B	C	D	E	F	G
A	N/A	0/378	0/112	0/112	0/166	0/112	0/117
B	2/882	N/A	0/274	0/274	0/328	0/274	0/279
C	0/263	0/641	N/A	0/9	0/63	0/9	0/13
D	2/263	1/641	0/21	N/A	0/63	0/9	0/13
E	60/389	4/767	0/147	2/147	N/A	0/63	0/67
F	0/263	7/641	0/21	2/21	0/147	N/A	0/13
G	0/273	2/651	0/32	2/32	0/158	0/32	N/A

A.2 Die Life Cycle Classification

Below are the measurements of a single subsection of a lot that was run through a thermal oven multiple times to simulate the process of soldering the part on a PCB and removing the part from the PCB. Bake 0 is a part that has never been soldered, and bake 6 is part that has been soldered and removed from the PCB 3 times.

Table A.16: SVM success ratios with linear kernel and .3 training ratio

Bake	0	1	2	4	6
0	N/A	46/108	44/108	41/108	46/108
1	120/252	N/A	39/108	48/108	22/108
2	92/252	120/252	N/A	40/108	35/108
4	95/252	136/252	137/252	N/A	29/108
6	134/252	77/252	105/252	82/252	N/A

Table A.17: SVM success ratios with RBF kernel, .00005=γ, and .3 training ratio

Bake	0	1	2	4	6
0	N/A	0/108	0/108	0/108	0/108
1	20/252	N/A	0/108	0/108	0/108
2	39/252	34/252	N/A	1/108	0/108
4	52/252	32/252	59/252	N/A	0/108
6	62/252	32/252	69/252	55/252	N/A

Table A.18: SVM success ratios with RBF kernel, .00001=γ, and .3 training ratio

Bake	0	1	2	4	6
0	N/A	2/108	0/108	0/108	0/108
1	16/252	N/A	0/108	0/108	0/108
2	33/252	35/252	N/A	1/108	0/108
4	43/252	19/252	54/252	N/A	0/108
6	50/252	13/252	55/252	19/252	N/A

Table A.19: SVM success ratios with RBF kernel, .000005=γ, and .3 training ratio

Bake	0	1	2	4	6
0	N/A	6/108	0/108	0/108	0/108
1	21/252	N/A	0/108	0/108	0/108
2	21/252	22/252	N/A	1/108	0/108
4	17/252	14/252	48/252	N/A	0/108
6	28/252	8/252	42/252	14/252	N/A

Table A.20: SVM success ratios with RBF kernel, .00005=γ, and .5 training ratio

Bake	0	1	2	4	6
0	N/A	0/180	0/180	0/180	0/180
1	14/180	N/A	0/180	0/180	0/180
2	25/180	18/180	N/A	0/180	0/180
4	21/180	18/180	26/180	N/A	0/180
6	32/180	18/180	40/180	30/180	N/A

Table A.21: SVM success ratios with RBF kernel, .00001=γ, and .5 training ratio

Bake	0	1	2	4	6
0	N/A	1/180	0/180	0/180	0/180
1	12/180	N/A	0/180	0/180	0/180
2	18/180	18/180	N/A	2/180	0/180
4	11/180	16/180	21/180	N/A	0/180
6	12/180	15/180	26/180	16/180	N/A

Table A.22: SVM success ratios with RBF kernel, .000005=γ, and .5 training ratio

Bake	0	1	2	4	6
0	N/A	4/180	0/180	0/180	0/180
1	11/180	N/A	0/180	0/180	0/180
2	15/180	13/180	N/A	2/180	0/180
4	9/180	12/180	16/180	N/A	0/180
6	6/180	6/180	16/180	9/180	N/A

Table A.23: SVM success ratios with RBF kernel, .00005=γ, and .7 training ratio

Bake	0	1	2	4	6
0	N/A	0/251	0/251	0/251	0/251
1	6/109	N/A	0/251	0/251	0/251
2	17/109	9/109	N/A	0/251	0/251
4	11/109	9/109	12/109	N/A	0/251
6	16/109	9/109	21/109	12/109	N/A

Table A.24: SVM success ratios with RBF kernel, .00001=γ, and .7 training ratio

Bake	0	1	2	4	6
0	N/A	2/251	0/251	0/251	0/251
1	5/109	N/A	0/251	0/251	0/251
2	15/109	8/109	N/A	0/251	0/251
4	7/109	8/109	13/109	N/A	0/251
6	10/109	7/109	16/109	3/109	N/A

Table A.25: SVM success ratios with RBF kernel, .000005=γ, and .7 training ratio

Bake	0	1	2	4	6
0	N/A	5/251	0/251	0/251	0/251
1	7/109	N/A	0/251	0/251	0/251
2	14/109	7/109	N/A	3/251	0/251
4	7/109	8/109	12/109	N/A	0/251
6	6/109	4/109	9/109	3/109	N/A

Table A.26: SVM success ratios with RBF kernel, .00005=γ, and .9 training ratio

Bake	0	1	2	4	6
0	N/A	0/324	0/324	0/324	0/324
1	3/36	N/A	0/324	0/324	0/324
2	8/36	4/36	N/A	0/324	0/324
4	5/36	4/36	5/36	N/A	0/324
6	7/36	4/36	8/36	3/36	N/A

Table A.27: SVM success ratios with RBF kernel, .00001=γ, and .9 training ratio

Bake	0	1	2	4	6
0	N/A	1/324	0/324	0/324	0/324
1	3/36	N/A	0/324	0/324	0/324
2	7/36	3/36	N/A	2/324	0/324
4	3/36	3/36	7/36	N/A	0/324
6	4/36	2/36	6/36	1/36	N/A

Table A.28: SVM success ratios with RBF kernel, .000005=γ, and .9 training ratio

Bake	0	1	2	4	6
0	N/A	7/324	0/324	0/324	0/324
1	2/36	N/A	0/324	0/324	0/324
2	6/36	2/36	N/A	3/324	0/324
4	2/36	4/36	6/36	N/A	0/324
6	3/36	2/36	5/36	0/36	N/A

Appendix B

Design Recovery Details

The contents of this section are under non-disclosure agreements and are archived at the Virginia Tech Secure Research Office.

Appendix C

Design Recovery Implementation

The contents of this section are under non-disclosure agreements and are archived at the Virginia Tech Secure Research Office.