Possible Organization for Writing a Thesis including a LaTeXFramework and Examples

by

A Graduate Advisor
B.Sc., University of WhoKnowsWhere, 2053
M.Sc., University of AnotherOne, 2054

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Whichever

Possible Organization for Writing a Thesis including a LaTeXFramework and Examples

by

A Graduate Advisor
B.Sc., University of WhoKnowsWhere, 2053
M.Sc., University of AnotherOne, 2054

Supervisory Committee

Dr. R. Supervisor Main, Supervisor
(Department of Same As Candidate)

Dr. M. Member One, Departmental Member
(Department of Same As Candidate)

Dr. Member Two, Departmental Member
(Department of Same As Candidate)

Dr. Outside Member, Outside Member
(Department of Not Same As Candidate)

Supervisory Committee

Dr. R. Supervisor Main, Supervisor
(Department of Same As Candidate)

Dr. M. Member One, Departmental Member
(Department of Same As Candidate)

Dr. Member Two, Departmental Member
(Department of Same As Candidate)

Dr. Outside Member, Outside Member
(Department of Not Same As Candidate)

ABSTRACT

This document is a possible Latex framework for a thesis or dissertation at UVic. It should work in the Windows, Mac and Unix environments. The content is based on the experience of one supervisor and graduate advisor. It explains the organization that can help write a thesis, especially in a scientific environment where the research contains experimental results as well. There is no claim that this is the *best* or *only* way to structure such a document. Yet in the majority of cases it serves extremely well as a sound basis which can be customized according to the requirements of the members of the supervisory committee and the topic of research. Additionally some examples on using LaTeXare included as a bonus for beginners.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

my cat, Star Trek, and the weather, for supporting me in the low moments.

Supervisor Main, for mentoring, support, encouragement, and patience.

Grant Organization Name, for funding me with a Scholarship.

I believe I know the only cure, which is to make one's centre of life inside of one's self, not selfishly or excludingly, but with a kind of unassailable serenity-to decorate one's inner house so richly that one is content there, glad to welcome any one who wants to come and stay, but happy all the same in the hours when one is inevitably alone.

Edith Wharton

DEDICATION

Just hoping this is useful!

Chapter 1

Introduction

The term *Trojan Horse* or *Trojan* has become a modern metaphor for a deception where by an unsuspecting victim welcomes a foe into an otherwise safe environment. [10] Though modern civilization rarely has need for large walls we are similarly surrounded. Not by stone and mortar but by the technology we so heavily rely on. These days it is more common to come across a piece of equipment with some form of computer in it than without. They provide us entertainment, education, security, monitor our health, grow our food and more. Our reliance make us susceptible to their compromise. Since the dawn of the computer we have dealt with software threats; we are almost as good as protecting ourselves against them as they are at attacking us. In recent years a new incarnation of danger has emerged; in hardware. In this new arena of attack and defend those who seek to defend are far behind.

1.1 Motivation

In the summer of 2007 an Israeli military action referred to as Operation Orchard commenced. A group of F-15I Ra'am fighter jets from the Israeli Air Force 69th Squadron took off to attack a suspected nuclear reactor in neighboring Syria. In the flight path was a Syrian radar station which boasted 'state-of-the-art' aircraft detection and neutralization technology. The Israeli war planes were able to approach and destroy the installation undetected. Though never proven it is commonly accepted that the detection mechanism was deactivated by a back-door circuit inserted into the radar system. [7] In 2011 over 1300 cases were reported to the Electronic Resellers Association International (ERAI) of modified ICs and the occurrence has been going

up. [1] Integrated Circuits (IC) are a large part of modern life yet it is easy to forget that they drive virtually every piece of technology used today. Ensuring their ICs run our devices as expected is vital in the digital era. Since their discovery there has been concerted effort to detect hardware trojans but the innate complexity of ICs makes it difficult. [4]

1.2 Contributions

1.3 Organization

Chapter 2

Hardware Trojans

2.1 Background

Integrated Circuits (IC) are continuously decreasing in size whilst increasing in complexity. These trends require ever more people and sophisticated means of manufacture which in turn creates security vulnerabilities. Products developed by semiconductor companies generally compromised in one of two ways. First, due to the complexity it is rare for a product to be managed within a single company. Frequently, steps in the production-chain are outsourced. It is within these 'third-party' contributors that products can be maliciously modified. Secondly, for various reasons, employees of trusted contributors have been known to make modifications. [2] These modifications are known as hardware Trojans. ICs are an integral part of every facet of the modern world. Proper application of a Trojan can provide information, control of mechanical systems, surveillance and more to an unauthorized party.

2.2 Topology

The discussion, detection and evaluation of hardware trojans requires a comprehensive means of description. Several hardware trojan taxonomies have been proposed [3, 6, 8, 9]. In [9], trojans were organized based solely on their activation mechanisms. A taxonomy based on the location, activation and action of a trojan was presented in [6], [3]. However, these approaches do not consider the manufacturing process. Another taxonomy was proposed in [8] which employs five categories: insertion, abstraction, activation, effect, and location. While this is more extensive than

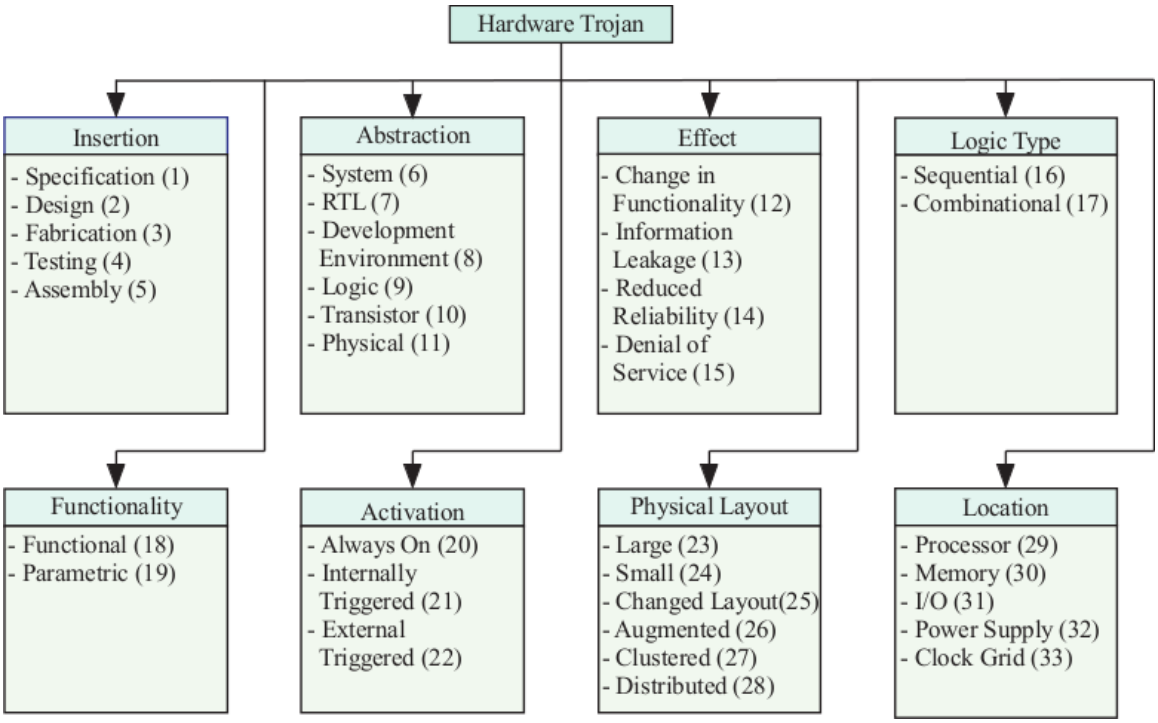

Figure 2.1: The thirty-three attributes of the hardware trojan taxonomy in [5].

previous approaches, it fails to account for the physical characteristics of a trojan. An additional taxonomy was proposed in [5] which considers all attributes a hardware trojan may posses. This taxonomy is the most comprehensive and was selected as the means of description for this work. It is comprised of thirty-three attributes organized into eight categories as shown in Fig. 2.1. These categories can be arranged into the following four levels as indicated in Fig. 2.2.

1. The **insertion** (chip life-cycle) level/category comprises the attributes pertaining to the IC production stages.

2. The **abstraction** level/category corresponds to where in the IC abstraction the trojan is introduced.

3. The **properties** level comprises the behavior and physical characteristics of the trojan.

4. The **location** level/category corresponds to the location of the trojan in the IC.

The properties level consists of the following categories.

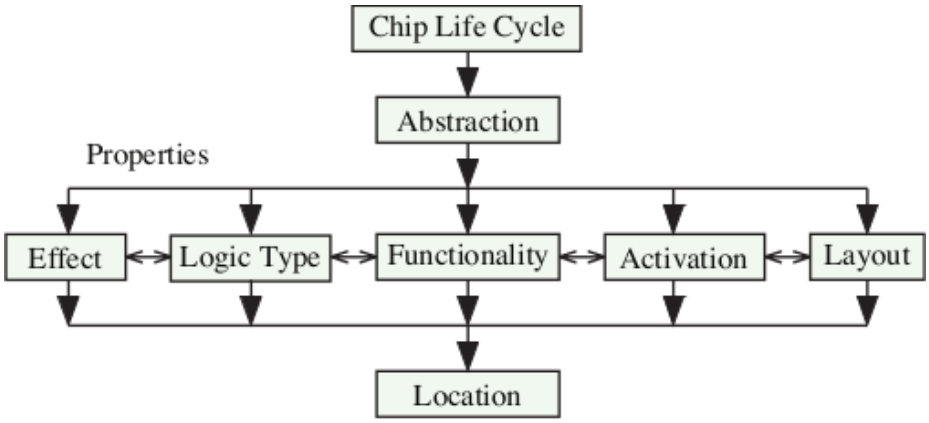

Figure 2.2: The hardware trojan levels [5].

- The **effect** describes the disruption or effect a trojan has on the system.
- The **logic type** is the circuit logic that triggers the trojan, either combinational or sequential.
- The **functionality** differentiates between trojans which are functional or parametric.
- The **activation** differentiates between trojans which are always on or triggered.
- The **layout** is based on the physical characteristics of the trojan.

The relationships between the trojan attributes shown in Fig. 2.1 can described using a matrix $\mathbf{R}$ [5]. Entry $r(i,j)$ in $\mathbf{R}$ indicates whether or not attribute i can lead to attribute j. For example, $r(2,3) = 1$ indicates that design (attribute 2) can lead to fabrication (attribute 3). This implies that if an IC can be compromised during the design phase (attribute 2), it may influence the fabrication phase (attribute 3).

The matrix $\mathbf{R}$ is divided into sub matrices as follows

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_1} & \mathbf{R_{12}} & 0 & 0 \\ 0 & \mathbf{R_2} & \mathbf{R_{23}} & 0 \\ 0 & 0 & \mathbf{R_3} & \mathbf{R_{34}} \\ 0 & 0 & 0 & \mathbf{R_4} \end{bmatrix}$$

$$\mathbf{R} =$$

A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	1	0	0	0	1	0	0	0	0	0																						
2	0	0	1	0	0	0	1	0	0	0	0																						
3	0	0	0	1	0	0	0	0	0	0	1																						
4	0	0	0	0	1	1	0	0	1	0	0																						
5	0	0	0	0	0	1	0	0	0	0	0																						
6						0	1	0	0	0	0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0					
7						0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	0	0					
8						0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1					
9						0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0					
10						0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	0	1	1	0					
11						0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	1	1					
12												0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13												0	0	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1
14												0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1
15												0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
16												1	0	0	1	0	0	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1
17												1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18												1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19												0	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1	1
20												1	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
21												1	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1
22												1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1
23												1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0	0	0
24												1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	1
25												1	0	0	1	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	1	1
26												1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
27												1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1
28												1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	1
29																																	
30																																	
31																																	
32																																	
33																																	

where $\mathbf{R_1}$, $\mathbf{R_2}$, $\mathbf{R_3}$ and $\mathbf{R_4}$ indicate the attribute relationships within a category. For example, $\mathbf{R_1}$ is given by

$$\mathbf{R_1} = \left[\begin{array}{c|ccccc} A & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 0 & 1 \\ 5 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Submatrix $\mathbf{R_{12}}$ relates the attributes of the insertion category to the attributes of the abstraction category. An example of this submatrix is

$$\mathbf{R_{12}} = \left[\begin{array}{c|cccccc} A & 6 & 7 & 8 & 9 & 10 & 11 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 1 & 0 & 0 & 1 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

2.3 Field Programmable Gate-Arrays (FPGA)

An IC belongs to one of two categories. An Application Specific Integrated Circuit (ASIC) or a Field Programmable Gate-Array (FPGA). An ASIC is manufactured once and is immutable; its hardware is permanently printed in silicon. FPGAs are reconfigurable. They are comprised of an array of Programmable Logic Devices (PLD). A PLD is a component whose functionality is dependent on a set of configuration options. Each PLD in and FPGA device receives a series of configuration instructions in the form of a binary message. A single FPGA can contain hundreds, or even thousands of PLDs, each of which can be one of hundreds of possible different types. The messages from all PLDs comprise the configuration Bitstream for the whole device. FPGA developers create processor designs using a programming language; the design is then downloaded or "configured" onto the device via the Bitstream. The design process is simpler and modifications can be made after manufacture. Because of this, FPGAs are becoming the IC of choice in larger scale productions however these same features increase their vulnerability.

Chapter 3

Automated Trojan Detection

3.1 Methodology

Figure 3.1 provides a visual representation of the basic concept assumed for the purposes of this work. All stages of production of an FPGA implementation are done "in-house" with the exception of the fabrication process. It is assumed that any trojan discovered is inserted in the fabrication phase; all other stages are trusted. The method of automated trojan detection described in this work would take place in the 'testing' phase of the life-cycle.

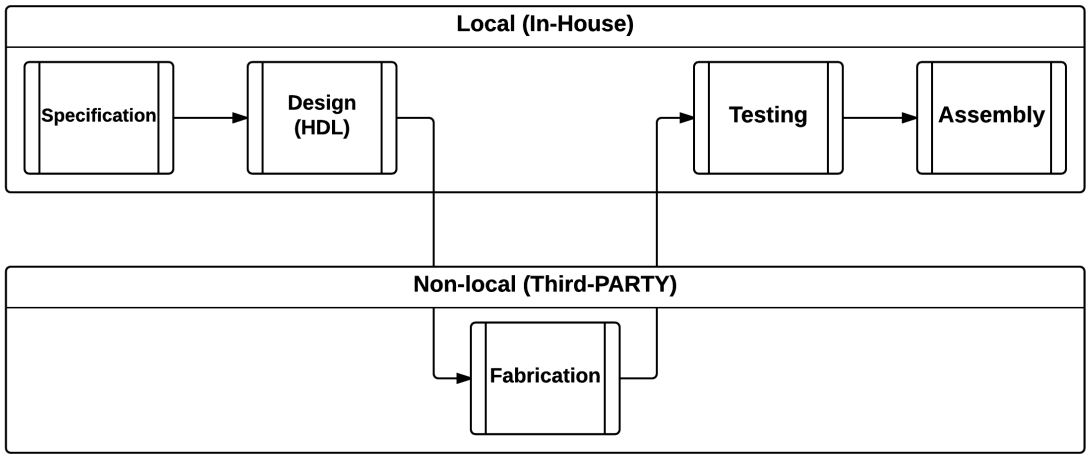

Figure 3.1: FPGA Life-Cycle

Figure 3.2 shows an overview of the trojan detection scheme. FPGA designs are written in a Hardware Description Language (HDL). *Xilinx* provides a series of User-

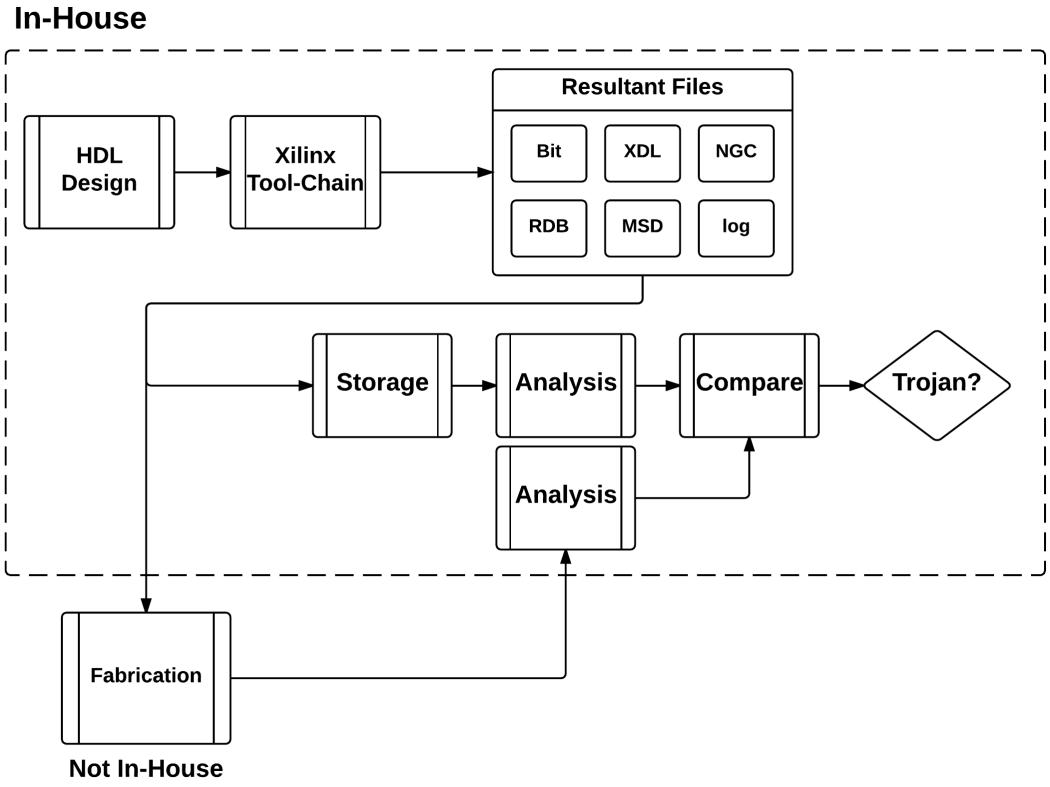

Figure 3.2: Methodology Overview

Interface (UI) or command line tools to process the design known as the 'tool-chain'. The tool chain generates a series of files that are used for a variety of purposes. The Bit file is a binary representation of the design to be implemented. It is referred to as the Bitstream or Configuration Bitstream and is the final form which is loaded into the device. This Bit file is also one of the primary files sent to the fabrication house where it will be implemented onto the batch of devices ordered. The resultant files are kept in secure storage while a copy is sent to be fabricated; these 'clean' copies are reffered to as Golden. Though it is known that fabrication houses will often attempt to make optimizations on designs this methodology requires that no such efforts are made. When the completed batch of fabricated chips are returned the Bitstream is extracted via the method described in section 3.4. That which is extracted is referred to as the Target Bitstream. The Golden and Target Bitstreams are analyzed in conjunction to detect differences, described in section 3.5. Any discovered differences are then attributed to the corresponding component in the architecture, described in

section 3.6. Finally, descriptive attributes presented in section 2.2 are returned to the user, described in section 3.7.

3.2 FPGA Architecture and Configuration

A *Xilinx* Field Programmable Gate-Array (FPGA) is comprised of a matrix of blocks referred to as the 'gate-array' and is shown in Figure 3.3. A device can contain anywhere from a couple hundred to a few thousand blocks; they are arranged into columns by type. A block will consist of one or multiple tiles depending on the type. A tile is a component specific to a particular function such as Input-Output (IO), design logic, memory...etc but their detailed functionality can be configured by the user. An FPGA may have over one-hundred different types of tiles however each column is comprised entirely by a single block type. Columns are separated by clock regions as shown by the dashed lines in Figure 3.3. Each region is an independent array of blocks that uses a dedicated clock resource; this minimizes clock skew from causing undesired timing delays. Though each tile has a designated purposes (ex.

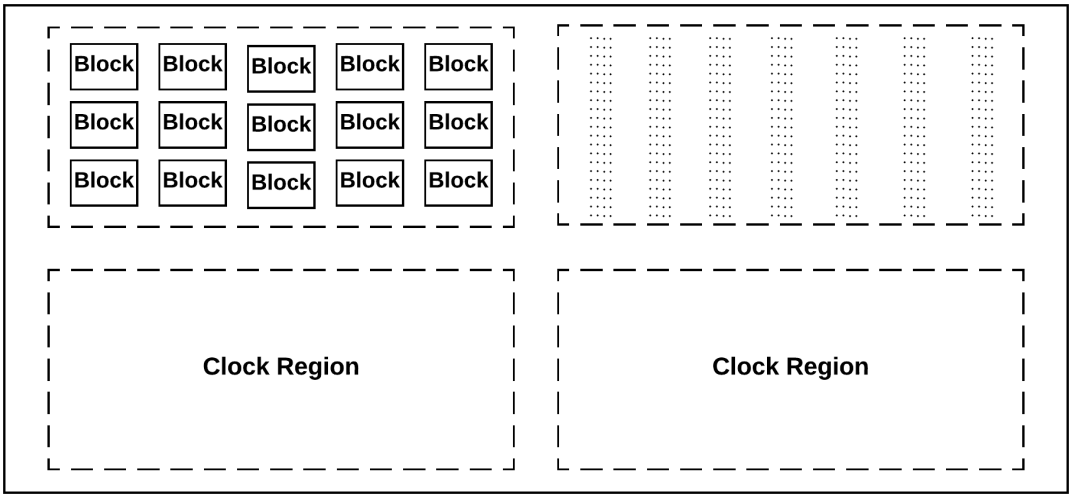

Figure 3.3: Rudimentary Layout of a Virtex Gate-Array

Input-Output (IO), Configurable Logic (CL), memory...etc) their functionality can be configured by the user; this is how designs are implemented on a device. The configuration of each tile is dictated by the Bitstream. To improve performance the contents of the gate-array is referred to as dynamic. A dynamic device is unable to

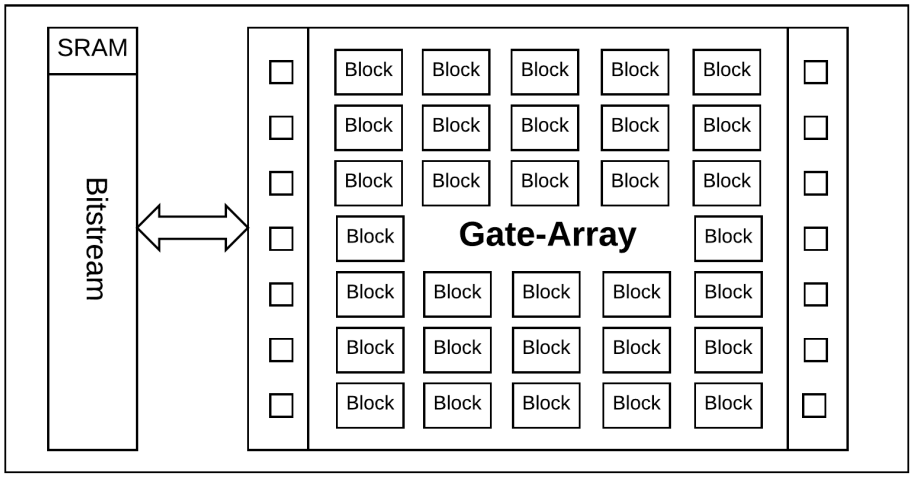

Figure 3.4: FPGA Device Layout

retain the contents of its memory when it looses power. To prevent having to plug in a device and download the configuration every time it is powered on, an external static memory device (i.e. retains its contents with loss of power) holds the Bitstream. When an FPGA is powered on the Bitstream is loaded from the external memory (often Static RAM (SRAM)) into the gate-array, as can be seen in Figure 3.4.

3.3 The FPGA Bitstream

The *Xilinx* bitstream is organized into 'frames'. A frame is a string of single bits that span from the top to the bottom of a clock region of a device as seen in the top-right quadrant of Figure 3.3. Frame affects every block in a column and multiple horizontally adjacent frames are required to configure an entire column. They are uniquely identified by a 32-bit address and are the smallest addressable element. The composition of the frame address is fairly consistent across the *Xilinx* catalog however there are small differences between device families. The following is the structure of the Virtex-5 family frame address scheme according to [11]. The make-up of a frame address is shown in Table 3.1. The Block Address (BA) identifies the block type.

Table 3.1: Frame Address

Unused								BA			T	Row Address					Major Address								Minor Address						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

- BA 0: Logic type.
- BA 1: Block RAM (BRAM).
- BA 2: BRAM Interconnect.
- BA 3: BRAM non-configuration frame.

The logic block contains the columns which provides the primary configuration for the device (CLBs, IOBs... etc). The BRAM columns initialize the memory for the device while the BRAM Interconnect columns configure how the logic of the design interacts with the BRAM.

In the case of the Virtex-5 family each clock region is composed of twenty blocks in a column separated by a horizontal clock bus as shown in Figure 3.5. Each row

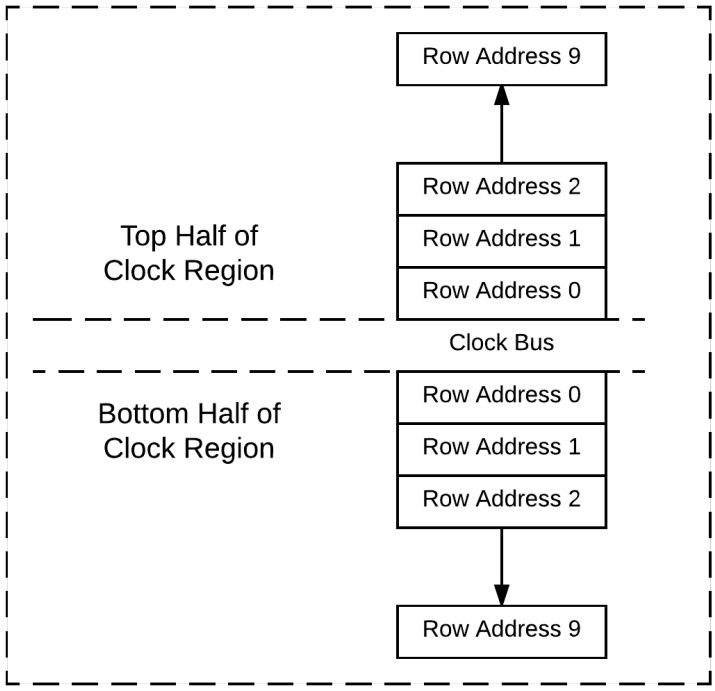

Figure 3.5: Row Order of Virtex-5 Clock Region

of blocks is given a row value in it's address that increments away from the clock bus starting at 0. The frame address includes a Top indicator bit in position 20 that indicates whether the specified row is above or below the horizontal clock bus. The major address specifies the block within the row. These addresses are numbered from

Table 3.2: Number of Frames (minor addresses) per Column

Block	Number Of Frames
CLB	36
DSP	28
BRAM	30
IOB	54
Clock	4

left to right and begin at 0. The minor address indicates the frame number within a column. Table 3.2 provides the number of frames per column type. The number of frames given in Table 3.2 corresponds to the block in the column. As described in section 3.2 a block may contain multiple tiles. In a CLB column a block consists of an interconnect column and a CLB. To further understand how frames configure tiles a mapping must be made between each frame and the corresponding tile. This is described in section 3.6.

3.4 Bitstream Extraction

In order to detect any trojans in the Target device the configuration Bitstream will need to be recovered. As mentioned in section 3.2 the Bitstream is stored in a memory unit external to the gate-array. All *Xilinx* devices provide a feature known as Readback. There are two styles of Readback; Readback verify and Readback Capture. The Readback capture method provides a large quantity of debug information which is not needed; Readback verify will be used. Readback verify is the process where the device is put into a 'frozen' state during run-time and all of the configuration bits are returned from the gate-array to the SRAM. The results can then be uploaded to a Personal Computer (PC) for analysis. This process overwrites the original frame data in the SRAM with the values which actually configured the device. By using this method rather than simply reading the SRAM it ensures that what is tested in section 3.5 is actually what configured the device. This minimizes risk of tampered external memory units or configuration mechanics.

3.5 Frame Comparison

3.6 Tile Mapping

3.7 Trojan Attributes

Chapter 4

Experiments

Assuming you have some experimental results to support your claims this is where all the data is reported. There are a few issues you should consider before dumping a lot of stuff here, or it will lose its effectiveness.

First of all you must describe precisely the experimental setup and the benchmarks you used. In any scientific discipline an experimental result is only good if it is reproducible. To be reproducible then somebody else must have sufficient details of the setup to be able to obtain the same data. Thus the first section in this chapter is a super precise history of the decisions made towards experimentation, including mentions of the paths which became infeasible. The setup must be valid and thus your description of it must prove that it is indeed sound. At times, terrifying times, when writing this section, both supervisor and student realize belatedly that something is missing and more work needs to be done!

The second portion of this chapter is dedicated the the actual results. At least two issues arise here:

1. Should all the data be reported here or should some be placed in the Appendix?
2. Should this be an exposition of the raw facts and data or should it include its analysis and evaluation?

There are no definite answers here, but I follow a few rules.
Should all the data be reported here or should some be placed in the Appendix?

- If there is a large number of tables of data, it might be better to present here only a handful of the most significant ("best") results, leaving all the rest of the data in the Appendix with proper linkages, as it would make the chapter

so much more easily readable (not to mention limiting the struggle with a word processor for the proper placement of tables and text).

- Use an example throughout, call it a "case study" to make it sound better, so that all the data and results are somehow linked in their logic, and even better if this is one of the examples you used in Chapter 2 to describe the original problem.

- Highlight in some manner the important new data, for example the column of your execution speed where all the numbers are much smaller. Make the results highly easy to read!

- It is normally expected that data should be presented only in one form and not duplicated, that is, you are not supposed to include both a table of raw numbers and also its graphical representation from some wonderful Excel wizard. I tend to disagree. I would not wish to see every results repeated in this manner, but some crucial ones need to be seen in different manners, even with the same information content, in order to show their impact. One good trick is to place the more boring tables in the Appendix and use wonderful graphs in this chapter.

- This is the one chapter where I would splurge and use colour printing where necessary, as it makes an *enourmous* difference.

Should this be an exposition of the raw facts and data or should it include its analysis and evaluation?

- Is the evaluation of the data really obvious? For example you have 10 tables to show that your chemical process is faster in development and gives purer material - you may simply need to highlight one column in each table and state the obvious.

- Most results are not that obvious even if they appear so. Moreover this is where you are comparing your *new* results to data from other people. I usually describe other people's work at this point and make comparisons. That is why I prefer to talk about the analysis and evaluation of the results in a separate chapter.

- There is absolutely no clear structure here which is best.

Chapter 5

Evaluation, Analysis and Comparisons

For a Master's research this chapter represents the critical part where **you** are truly evaluated to determine whether you should be given your degree. Even more so for a PhD. Consider carefully what the University calendar states regarding the expectations for a master's thesis, paraphrased here.

1. *A Masters thesis is an original lengthy essay.* The main implication here is that the essay is original, that is, it is completely newly written by you and does not contain any writings from others unless precisely quoted. Any paraphrased items must be cited.

2. *It must demonstrate that:*

   - students understand research methods;
   - students are capable to employ research methods;
   - students demonstrate command of the subject.

3. *The work may be based on:*

   - original data;
   - original exercise from scholarly literature;
   - data by others.

4. *The work must show that:*

- appropriate research methods have been used;
- appropriate methods of critical analysis supplied.

5. *The work must contain:*

   - evidence of some new contribution;
   - evidence of a new perspective on existing knowledge.

Only the last point uses the attribute *new* and it refers almost entirely to giving a new perspective and analysis, even if based on data from others. This truly implies that this current chapter on evaluation and analysis of results is the most important and must be written with care. You are demonstrating here that, even if given data and methods from others, your skills of critical judgment and analysis are now at the level that you can give professional evaluations.

Things are slightly different for a PhD. According to the Graduate Calendar:
a doctoral dissertation must embody original work and constitute a significant contribution to knowledge in the candidate's field of study. It should contain evidence of broad knowledge of the relevant literature, and should demonstrate a critical understanding of the works of scholars closely related to the subject of the dissertation. Material embodied in the dissertation should, in the opinion of scholars in the field, merit publication.

The general form and style of dissertations may differ from department to department, but all dissertations shall be presented in a form which constitutes an integrated submission. The dissertation may include materials already published by the candidate, whether alone or in conjunction with others. Previously published materials must be integrated into the dissertation while at the same time distinguishing the student's own work from the work of other researchers. At the final oral examination, the doctoral candidate is responsible for the entire content of the dissertation. This includes those portions of co-authored papers which comprise part of the dissertation.

The second paragraph makes it clear that one must emphasize what is new and different from others, without arrogance, yet without being too subtle either. The first paragraph implies that for a PhD it is required that one approached an important open problem and gave a new solution altogether, making chapters 3, 4, 5 all part of the body of research being evaluated. In fact at times even the problem may be entirely new, thus including chapter 2 in the examination. This is in contrast to a Master's degree where the minimum requirement is for chapter 5 to be original.

Chapter 6

Conclusions

My first rule for this chapter is to avoid finishing it with a section talking about future work. It may seem logical, yet it also appears to give a list of all items which remain undone! It is not the best way psychologically.

This chapter should contain a mirror of the introduction, where a summary of the *extraordinary* new results and their wonderful attributes should be stated first, followed by an executive summary of how this new solution was arrived at. Consider the practical fact that this chapter will be read quickly at the beginning of a review (thus it needs to provide a strong impact) and then again in depth at the very end, perhaps a few days after the details of the previous 3 chapters have been somehow forgotten. Reinforcement of the positive is the key strategy here, without of course blowing hot air.

One other consideration is that some people like to join the chapter containing the analysis with the only with conclusions. This can indeed work very well in certain topics.

Finally, the conclusions do not appear only in this chapter. This sample mini thesis lacks a feature which I regard as absolutely necessary, namely a short paragraph at the end of each chapter giving a brief summary of what was presented together with a one sentence preview as to what might expect the connection to be with the next chapter(s). You are writing a story, the *story of your wonderful research work*. A story needs a line connecting all its parts and you are responsible for these linkages.

Appendix A

Additional Information

This is a good place to put tables, lots of results, perhaps all the data compiled in the experiments. By avoiding putting all the results inside the chapters themselves, the whole thing may become much more readable and the various tables can be linked to appropriately.

The main purpose of an Appendix however should be to take care of the future readers and researchers. This implies listing all the housekeeping facts needed to continue the research. For example: where is the raw data stored? where is the software used? which version of which operating system or library or experimental equipment was used and where can it be accessed again?

Ask yourself: if you were given this thesis to read with the goal that you will be expanding the research presented here, what would you like to have as housekeeping information and what do you need? Be kind to the future graduate students and to your supervisor who will be the one stuck in the middle trying to find where all the stuff was left!

Bibliography

[1] Celia Gorman. Counterfeit chips on the rise. IEEE Spectrum: Technology, Engineering, and Science News, 5 2012. url: http://spectrum.ieee.org/computing/hardware/counterfeit-chips-on-the-rise.

[2] N. Jacob, D. Merli, J. Heyszl, and G. Sigl. Hardware trojans: current challenges and approaches. *IET Computers Digital Techniques*, 8(6):264–273, 2014.

[3] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor. Trustworthy hardware: Identifying and classifying hardware trojans. *Computer*, 43(10):39–46, Oct 2010.

[4] H. Li, Q. Liu, J. Zhang, and Y. Lyu. A survey of hardware trojan detection, diagnosis and prevention. In *2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, pages 173–180, Aug 2015.

[5] S. Moein, S. Khan, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi. An attribute based classification of hardware trojans. In *Computer Engineering Systems (ICCES), 2015 Tenth International Conference on*, pages 351–356, Dec 2015.

[6] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic. Power supply signal calibration techniques for improving detection resolution to hardware trojans. In *2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 632–639, Nov 2008.

[7] H.-S. Philip Wong Subhasish Mitra and Simon Wong. Stopping hardware trojans in their tracks. IEEE Spectrum: Technology, Engineering, and Science News, 1 2015. url: http://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks.

[8] Xiaoxiao Wang, M. Tehranipoor, and J. Plusquellic. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 15–19, June 2008.

[9] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty. Towards trojan-free trusted ics: Problem analysis and detection scheme. In *2008 Design, Automation and Test in Europe*, pages 1362–1365, March 2008.

[10] Michael Wood. *"In search of the Trojan war.* The British Broadcasting Corporation, 1 edition, 1998.

[11] Xilinx. *Virtex-5 FPGA Configuration User Guide*, v3.11 edition, Oct 2012.