2006

The integration of third party intellectual property (IP) : reuse of IP cores

Paul Weil
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Weil, Paul, "The integration of third party intellectual property (IP) : reuse of IP cores" (2006). *Master's Theses*. 2932.

DOI: <https://doi.org/10.31979/etd.53fj-2t3f>
https://scholarworks.sjsu.edu/etd_theses/2932

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

THE INTEGRATION OF THIRD PARTY INTELLECTUAL PROPERTY (IP): REUSE
OF IP CORES

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Paul Weil

May 2006

UMI Number: 1436958

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 1436958

Copyright 2006 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© 2006

Paul Weil

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

Dr. David Parent (Chair)

Dr. Lili He

Dr. Tamara Papalias

APPROVED FOR THE UNIVERSITY

05/08/06

ABSTRACT

THE INTEGRATION OF THIRD PARTY INTELLECTUAL PROPERTY (IP): REUSE OF IP CORES

By Paul Weil

IP reuse is a multi-billion dollar segment of the Integrated Circuit design industry. With its ever growing use in SoC designs, IP quality, integration, and reuse has become an important issue in any successful design. Choosing the wrong piece of IP can increase time to market, or even result in a failed design. The use of such standards as the VSIA Quality IP metric can help in quantifying IP and by doing so, risks can be greatly reduced. By taking a chip through an entire design cycle, and utilizing third party IP within the design, IP quality and reuse can be commented on. It can be shown that with the use of the VSIA Quality IP metric, IP can be successfully quantified, and the design cycle can be shortened with reduced risk. As a result, third party IP can be used with a higher degree of confidence in any design.

Table of Contents

CHAPTER ONE: INTRODUCTION	1
1.1 Statement of the Problem.....	1
1.2 Objectives	1
1.3 Research Question	2
1.4 Research Significance.....	3
1.5 Relevant Definitions/Glossary	4
1.6 Methodology	5
1.7 Scope/Limitations	6
1.8 Overview of Remaining Chapters.....	6
CHAPTER TWO: LITERATURE REVIEW.....	7
2.1 Introduction.....	7
2.2 Overview of IP Core Use	7
2.3 IP Core Quality	9
2.4 Checking For IP Core Reusability	11
2.5 Measuring IP Quality.....	18
2.6 VSIA Quality Metrics Assessment Workbook.....	20
2.7 Summary.....	24
CHAPTER THREE: METHODOLOGY	26
3.1 Introduction.....	26
3.2 Research Approach	26

3.3	Research Strategies/Timeline.....	27
3.4	Research Tactics: Data Samples and Data Collection	28
3.5	Summary	29
CHAPTER FOUR: RESULTS		30
4.1	Introduction.....	30
4.2	IP Blocks	30
4.2.1	ADC/DAC Design IP.....	30
4.2.2	Oklahoma State University Standard Cell Library	33
4.2.3	Tanner High ESD I/O Pads.....	33
4.3	Chip Design	34
4.4	VSIA QIP Data	37
4.4.1	Integration Manual Results	41
4.4.2	Ease of Integration Results	43
4.4.3	Verification Quality Results	44
4.5	IP Integration	48
4.5.1	ADC/DAC Design IP.....	48
4.5.2	Oklahoma State University Standard Cell Library	51
4.6	Summary	53
CHAPTER FIVE: CONCLUSIONS, FUTURE RESEARCH, REMARKS. 54		
5.1	Introduction.....	54
5.2	General Conclusions	54
5.3	Suggestions for Further Research	56

5.4	Some Concluding Remarks.....	57
	References	59
	Appendix A: Chip Design Tutorial	64
	Synthesis	65
	Place and Route.....	73

List of Figures

Figure 1: Sample page from summary report of VSIA QIP metrics	22
Figure 2: 5-bit analog to digital converter IP block.....	31
Figure 3: 5-bit digital to analog converter IP block.....	31
Figure 4: ADC/DAC Design IP directories	32
Figure 5: Completed ASIC chip design.....	36
Figure 6: SJSU ADC/DAC IP VSIA summary report.....	38
Figure 7: Documentation quality section.....	40
Figure 8: Integration manual section	41
Figure 9: Rules not followed in integration manual section	42
Figure 10: Imperatives followed in analog hardware reference manual.....	42
Figure 11: Analog ease of integration section	43
Figure 12: Rules followed in the analog ease of integration section	44
Figure 13: Rules not followed within analog ease of integration section.....	44
Figure 14: Verification section of the VSIA QIP metrics.....	45
Figure 15: Imperatives not satisfied within verification quality section.....	46
Figure 16: Rules satisfied within verification quality section	46

CHAPTER ONE: INTRODUCTION

1.1 Statement of the Problem

Integrating third party Intellectual Property (IP) into large System on a Chip (SOC) designs can be a time consuming, risky, complicated, and difficult task to carry out in an effective and efficient manner. This makes IP quality and use of IP standards an important prerequisite for IP reuse in new designs. With the ever-growing complexity and size of today's chip designs, the use of third party IP has become an essential solution in getting any large chip/SOC design completed on time and within budget. As a result, the quality of the IP core has become the deciding factor in whether to integrate existing IP into the design, and whether it can be incorporated in a correct and timely manner.

1.2 Objectives

By taking several IP blocks through the process of integration in an actual chip design, and documenting the results, some of the typical integration and quality problems can be highlighted. Along with the use of the Virtual Socket Interface Alliance (VSIA) [1] Quality IP (QIP) [2] standard and applying it to the IP blocks, it can be shown that standards can help in choosing the right IP, as well as drive IP quality and reuse. In this paper, an overview of the IP quality assessment issue is examined, as well as why reuse is very important for today's IP core designers—as part of the literature review. More specifically, by actually taking several IP blocks and using them within a chip design, this

paper can then comment on some of the hazards and complications of IP reuse, and the need for industry to be aware of them and their need to address them.

It is the central argument of this paper that: (a) it is profitable to apply quality standards for IP reuse; (b) the assessment of IP cores is now facilitated to the point where standardized forms can help provide quality scores on individual cores and these can then be measured against others using the same scoring system; (c) while the achievement of a “no error” assessment is not yet possible in real time, such a state can be approached within acceptable levels; and (d) the industry needs to be aware of the common pitfalls of third party IP reuse, and be prepared to address them.

The paper also looks at how a set of standards (the VSIA QIP [2] spreadsheet) can help identify issues with IP, and how application of these standards can help select the right IP for reuse.

1.3 Research Question

The key research question examined in this paper is: **Is it possible, using present-day IP testing technology and scoring systems, to determine on a consistent and reproducible basis the quality of re-usable Intellectual Property cores? That is, is it possible to determine quantify IP in a way that can be used for future installations of such re-usable cores?**

1.4 Research Significance

As the re-usable IP core industry increases in size, it will become more and more important to be able to make such quality determinations before the installation process. Making the testing process easier, more standardized and less time-consuming than was previously possible is a necessary first step to closing the gap between ever-increasing demand for System-On-A-Chip (SOC) designs and the ability of engineers to successfully implement the design.

As well, the research will help to determine how to choose/quantify which IP vendors are providing cores of an acceptable quality, something that will become more important as more and more of these vendors enter the market.

Finally, it is hoped that the present research into the testing of re-usable IP cores will eventually lead the manufacturers and vendors of such cores to concern themselves much more with design problems and solutions. It is the belief that the solving of design problems before the cores are released for sale is a better way to ensure the quality of such cores than is the verification process alone. However, knowing that there are verification testing processes out there to help catch errors and defects in IP cores should encourage designers to be more vigilant in the first place—or else end up on the wrong list when it comes to making recommendations for the purchase of future IP cores from the same vendor.

1.5 Relevant Definitions/Glossary

Some definitions of terms used in the paper include:

Design Reuse: The inclusion of previously designed components (blocks of logic or data) in software and hardware. Ideally, design reuse makes it faster and cheaper to design and build a new product, since the reused components will not only be already designed but also tested for reliability. However, that does not guarantee the integrity of the reused component within a new integrated system universe.

Intellectual Property (IP) Core (Block or Macro): A block of logic or data that is used in making a field programmable gate array (FPGA) or application-specific integrated circuit (ASIC) for a product. Universal Asynchronous Receiver/Transmitter (UARTs), central processing units (CPUs), Ethernet controllers, and PCI interfaces can be examples of IP cores. IP cores come in three categories: hard, firm, or soft. Hard cores are physical manifestations of the IP design; they have been made in silicon, are best for plug-and-play applications, and are less portable and flexible than the other two types. Like the hard cores, firm (or semi-hard) cores carry placement data but are configurable to various applications. The most flexible of the three, soft cores exist either as a netlist (a list of the logic gates and associated interconnections making up an integrated circuit) or hardware description language (HDL) code, such as Verilog and VHDL.

Quality: How well an IP core conforms to a set of specifications (customer or vendor) or pre-defined standards. It can be rated qualitatively or quantitatively according to a scoring scheme.

System-On-A-Chip (SoC): The placing of all the necessary electronic circuits and parts for a “system” (such as a cell phone or digital camera) on a single integrated circuit (IC).

1.6 Methodology

The methodology used in this paper is a quantitative one. Basically, the researcher has taken several IP cores or blocks and used them in an actual chip design. By doing so, first hand knowledge and experiences on IP reuse can be gathered. This data can then be applied to the research and commented on. The IP blocks were also tested and quantified using the VSI Alliance (VSIA) [1] Quality IP (QIP) metrics/standard spreadsheet [2]. This is to show that such standards can be applied to any IP core and thus help in choosing a quality core for reuse. At the same time, such setting of standards in a quantitative way should help in raising the standards for future reusable IP core design.

By taking these blocks through the VSIA QIP scores can be determined on their quality. The use of these scores can then be applied to help quantify the IP, as well as highlight where the blocks are deficient in quality. The results from this process can then be used going into the chip design process, letting the designer know where there might be problems during integration of the IP into the design.

A more detailed account of the testing process as well as the particulars of the chip design along with the IP blocks used is presented in the Chapter Three Methodology section.

1.7 Scope/Limitations

The scope of the paper is the area of reusable IP blocks and its intersection with the area of the VSIA QIP spreadsheet as used for testing said IP blocks. The limitation of the paper is its inability to generalize with respect to the ability of the VSIA QIP spreadsheet to capture every aspect of the IP core quality matrix (on a theoretical level). It is, however, the contention of this paper that the VSIA QIP spreadsheet does allow for testing that is of significant standardization within the IP core testing world and that it is thus useful in practical terms for those wishing to quickly and inexpensively determine the quality of their IP cores.

1.8 Overview of Remaining Chapters

The methodology and procedures of the study are presented in Chapter Three, as well as explanations of the effectiveness of using the particular methodology chosen, brief theoretical perspectives, research design (data collection, and data analysis). The results of the methodology are presented in Chapter 4. The final chapter of the study is concerned with the summary of findings, conclusions, and recommendations for further research.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

In this section, a literature review of relevant material is presented. The review is divided into several parts, including: (a) An Overview of IP Core Use; (b) IP Core Quality; (c) Checking For IP Core Reusability; (d) Measuring IP Quality; (e) VSIA Quality Metrics Assessment Workbook [2]; (f) Summary.

2.2 Overview of IP Core Use

IP cores are the result of an imbalance or “design gap” between the amount of silicon available for chip use and the ability of engineers to design and verify new circuits for this silicon architecture: “the difference between the number of available logic gates given the prevalent process generation of the day and the number of gates that present EDA [electronic design automation] tools allow designers to really use” [33].

In the 1980’s gate level design was the norm, with designed gate- and register-level schematics using pencil and paper. Once EDA tools became available during the late 80’s, the tools could then simulate and string together the logic gates previously drawn by hand. This in turn created an increase in the “design gap.” To keep up with this new emerging design gap, a higher level of abstraction was needed for circuit design, and hardware design languages such as Verilog and circuit synthesis tools emerged to fill the gap. Now circuits could be designed from a higher level of abstraction, away from the gate level, thus enabling much larger circuits with higher transistor counts to be

designed. Now in 2006, with ever smaller transistors sizes, predictions are that the ever widening gap will be close to 200 million gates [3]. Smith [3] argues that to close this gap the emergence of electronic system level (ESL) methodology, and the ability to reuse very large functional blocks (IP) will be the answer. ESL methodologies represent an even higher system level abstraction for chip design. At the present moment, ESL methodologies have not been embraced and developed enough for adoption by the EDA companies and industry. As a result IP reuse has become the current forerunner and solution to the industry's "design gap." As Wilson [4] puts even more dramatically:

Simple arithmetic can show the necessity for intellectual property reuse in system-level chip design. A modest system-on-chip requires millions of gates. Each designer is capable of averaging perhaps a few tens of fully verified gates per day. Unless substantial portions of the chip are reused from previous designs or licensed from third parties, either the design team will grow to the Intel-esque scale or the schedule will span a geological time frame.

IP cores (also called macros or blocks) represent the most widely used method of trying to close this gap, which is ever-widening. Theoretically, designers can use these pre-created blocks as part of their System-On-A-Chip (SoC) designs without having to create them themselves, thus saving time and energy involved in essentially re-inventing the core. This practice has spawned a new segment of the industry, the sale of reusable IP cores. According to Gogolewski [5]:

Buying IP is no longer an option. For design teams struggling to meet tight deadlines and budget constraints, buying IP is imperative. IP has become the fundamental foundation for enabling massive system-on-a-chip (SoC) designs.

2.3 IP Core Quality

The use of an existing design component does not necessarily mean that these cores are truly plug-and-play. To quote Gogolewski [5] again:

It wasn't all that long ago that any programmer with a workstation in the second bedroom in their home could whip up some register transfer level (RTL) code, throw together some documentation and call it an IP product. It is obvious that, while this type of IP could promise low cost and instant availability, the quality, support and reusability were questionable at best. And, it's also not hard to see how this impacted functional verification efforts as external and, potentially buggy, IP was getting dropped into chip designs.

While the problem may not be viewed as horrendous, drastic or obvious today, there are still plenty of quality issues that need to be worked out when it comes to IP cores. In fact, even today, there are those who are pessimistic about what is taking place in the industry. According to Naveed Sherwani, president and CEO of ASIC Venture Open Silicon: "The situation is so bad that we consider all IP defective by definition. We have an internal IP quality group that not only investigates the IP and the vendor, but surveys previous users to try to anticipate the problems" [4]. For one thing, there is the issue of compatibility; for another, not all IP cores are created equal: the quality differences between one core and another can be immense.

Thus, it is important to have methods for testing and verifying these cores before they become part of a SOC. On the simplest level, the team designing the SOC has to be able to verify that any changes they've made do not impact on the core:

One way to verify that the new design has not functionally changed from the old one is for the IP provider to supply a test bench and test suite to fully verify the correct operation of the core. Unfortunately, the complete test suite for many cores is too large to be delivered as part of the IP core. So, most IP providers select a subset of the full verification suite that can be run to verify correct operation [6].

Functional verification can become a sticking point when it comes to IP because “engineers have to rely on IP suppliers or other design groups within their enterprise to provide bug-free designs and bug-free models of those designs at multiple levels of abstraction. Complicating the task, the engineers in most cases have no hands-on knowledge of how the IP has been designed and verified” [7].

Before an IP core is even considered for possible reuse, there are several questions that need to be answered. Among them:

- Is the IP core designed for reuse or is it a previous design simply repackaged as an IP core?
- Does the firm offer a complete IP core product line, making it easier for later updates and/or enhancements when needed?
- Does the firm provide strong maintenance and support, including a properly documented way for customer support, forthrightness in disclosing bugs, timeliness of bug fixes, release to add functionality, quick response to issues raised, and strong first-line support?

Only then can the core be considered for testing.

Gajski *et al.* [8] list a series of problems facing both the IP provider and the IP Integrator (user). The principle IP Integrator issues include:

- Exploration: a mechanism is required to determine the design alternatives and the functionality in terms of suitability
- Integration: complicated by a number of issues such as specifications, support, interface design, testing, and verification.

- Standardization: a process to be developed for purchase as well as specifications for simple evaluation.

In 2000, Savage *et al.* [9] rightly point out that successful reuse of IP cores is not the same as software reuse. In the case of software, bugs are quickly spotted when the program is compiled and run; but hardware IP cores can only be truly tested when built—and that could cost between \$500K-\$1M that “can be lost in the blink-of-an-eye as result of one wrong line of RTL code” [9]. Hence, quality standards and testing formats are critical: “With verification devouring more than 70 percent of the project cycle, they [design teams] wondered how much IP was saving them in terms of time and resources” [5].

2.4 Checking For IP Core Reusability

Both IP core reuse and the measurements needed to ensure a quality product have advanced at eye-popping speed—even if measured only from the year 2000 to date. In 1999, Bricaud [10] pointed out what he called the Silicon White Space Hole, “the available silicon left that a system designer has to fill after using his created IP' and memory blocks.”

At the time, Bricaud [10] provided some astounding numbers to crunch: A Silicon White Space Hole in the millions of gates to fill. To produce new code would require thousands of engineer-man-hours—and that still would not close the gap, especially when there was already talk of a 12 million equivalent gate SoC [10]. Thus, the need for quality reusable cores is apparent.

At this point, Bricaud [10] talked mainly of checking to ensure that the cores are in fact reusable. He divided the process up into essentially two steps: the integration step and the verification step. The integration process is a relatively easy one in the sense that it is an on-off two-state system: the blocks can either be integrated or they cannot: “If the blocks have been created and documented for easiness to use, then the integration process is very straightforward. If the blocks are each a separate individual jewel based around performance-area-power, but are isolated islands, then integration will be a nightmare” [10].

The verification process turns out to be much more complex. This was mainly because of the processing power needed to run verification tests—several weeks for just one core with each embedded processor having in the area of 500 test suites [10]. Instead, the process was performed only on the interblocks in what is labeled as “transaction verification.” Transaction verification works by raising the verification effort to a higher level of abstraction. This type of verification uses transactions at the chip system level, such as what kind of data flows through the systems and how it is stored, to improve verification productivity. As a result, an engineer can develop and diagnose tests from system level abstraction.

In 1999, Thomas [11] points out that, with the exponential increase in data with the increase in gates, it is no longer enough to test individual nodes, for example, because “you cannot consider a node on a netlist to be a single point—it may be represented by interconnect that travels all the way across a chip.” Thomas [11] also argues that the acquisition of IP cores for reuse must undergo successive steps—from identification

through a database to determine what is needed to a process of searching for the right IP (first internally, then through partners, trusted outside vendors, and only as a last resort the open market). Throughout all this process, “[a]dherence to standards (such as VSIA and SI2) should be a major factor in IP selection. You will need to set up procedures to handle soft and firm IP to ensure consistency” [11].

According to Savage *et al.* [9], the verification process for IP cores typically concerns itself with four basic types of activities:

- Functionality test development for basic operation verification
- Verification of functionality under unique or unusual conditions, using “deep corner cases”
- Random testing
- Silicon prototyping before product release

Savage *et al.* [9] also insists that “IP *quality* is the cornerstone that will enable the proliferation of SoC semiconductor devices.” However, as of 2002, Seepold *et al.* [12] could still state that no clearly formalized and quantified method of conducting this type of quality control was available. A similar statement in 2002, on the part of Yun *et al.* [13] covered the reuse of hard IP which was described as “largely a manual and error-prone process ... Repairing a hard IP requires the designer to understand the functionality, interfaces and electrical characteristics of complex cores such as microprocessors, MPEG decoders, DMA controllers, etc.”

By 2004, Keating [14] could make the argument that the vendors of commercial IP were making great strides in the improvement of the quality of their products—with

the hope of perfection at the end of it. At the same time, Keating [14] argues that this is a Herculean task that probably will never be achieved, stating that “there is no interesting (that is, complex) piece of code, either in software or in hardware, that is defect-free.” The same argument holds for the verification process: “A reasonably complex design (say, with 100 or more state flops) has a state space that cannot possibly be completely tested. (Even at two billion states per second, covering 2^{100} states would take longer than the life of the universe)” [14]. So the object becomes not cleaning out all the potential bugs but simply catching the most egregious ones. Keating *et al.* [14, 15] lists three best practices to help improve IP core quality:

- Design: the use of best design practices, including proper partitioning and interfacing
- Review: according to Jones [16], conducting a thorough review of both design and code is a more effective and efficient way of uncovering bugs than through testing
- Verification: Despite the importance of the first two methods for quality control, this is where most of the time and energy is spent, with “constrained random testing” rather than directed testing as the principal way of conducting tests.

Despite the fact that the verification methodologies have improved tremendously in just a few years, in most cases in little over a half-decade, that does not mean that error-free IP is closer. According to Laurent Ducousso [7], IP verification manager at STMicroelectronics, “a third of all IP hardware bugs occur in the specification, another

third occur due to missing test cases, and the final third result from missing random tests. Meanwhile, four-fifths of all SoC bugs ... come about because of bugs in IP.”

As Keating [14] points out, “there is no obvious path to intersecting the axis of zero defects. Instead, we seem to be riding an asymptote towards ever-increasing quality, but without ever hitting the axis itself. The state space of interesting designs is still too large for complete coverage in a finite amount of time, and formal tools are a long way from being able to prove all the important properties of a complex design.”

In 2005, Keating [17] presented this problem much more specifically where he argued basically that verification is not capable of addressing the IP quality issue to the desired levels that IP users want. Instead, the emphasis should be on improving design, and on finding better methods to measure design state space and to represent “the state space in a way that leads to more effective reasoning about the design” [17].

According to Bruce Wile [18], ASIC chief engineer for IBM’s Systems and Technology group, exhaustive verification taken one level at a time is essential—and the move up to the next level should only be taken once the bug rate drops to an acceptable number at the previous level: “If you didn’t create the scenario that is going to cause that assertion to go off before you get to silicon, then you’ve missed the bug. Exhaustive verification is much easier to reach at the lower levels.” Assertions are used for debugging by catching can’t happen errors. Within hardware description language (HDL) designs, an assertion is a conditional statement that checks for specific behavior and displays a message if it occurs. Assertions are generally used as monitors looking for bad behavior, but may be used to create an alert for desired behavior as well [19]. Similarly,

John Goodenough [20], director of design technology at ARM Ltd., argues for the need for verification IP, saying that there is no such thing as “correct by design. As a user of IP, you need to understand that that’s the way it is. You need to ask IP vendors what they verified and how they verified it. The most important thing is not what is verified, but what is not verified.”

Gogolewski [5] argues for integrated packaged solutions combining IP with verification IP:

That means that IP providers now are expected to test, package, verify and deliver quality IP that is ready for efficient deployment ... The IP business has matured in the last few years and IP blocks have become an essential part of the SoC design fabric. Change is on the horizon to help design teams more effectively and efficiently implement third-party blocks.

As early as 1999, while listing the challenges involved in the implementing of third-party reusable IP cores, which included the difficulties involved in the verification, testing, and troubleshooting of cores, Mallya *et al.* [18] argued for an almost symbiotic relationship between the core vendor and the core customer or user:

The purchase, implementation, and verification of assigned cores have to be part of a “joint venture” between core vendor and customer ... Problems encountered by designers can be caused by insufficient core design and verification by the IP core vendor, inadequate understanding of the IP core’s function by the customer, or miscommunication between both parties ... Obtaining cores from third-party vendors means the customer must diligently specify core deliverables to the vendor. Designing chips with these cores requires close cooperation with the customer and IP core vendor. Finally, companies designing core-based chips will always have to decide whether to build a particular core themselves or buy it from an IP core provider.

This joint venture is very similar in concept to “extreme programming” [21]. Extreme programming is a software development strategy, where by the programmers work closely together and in conjunction with the customer, to achieve best practices and

results. The software goes through planning, design, test, debugging, and design enhancements. Through the entire design cycle, both the supplier and customer need to work closely together to deliver the final product based on an approach simplicity, communication, and feedback [21].

That quality control of third-party IP cores is extremely important to the industry can be seen by some projections related to the future market for these types of cores. As of 2003, there were more than 200 IP providers, and revenue for the 2004 year-end was placed at \$1.5 billion [22].

This increase in markets and burgeoning market share for IP cores has also led to an increase in the complexity of such cores. According to Ensell [22]: “[U]sers report that third-party IP blocks are becoming substantially more complex, more difficult to integrate and more difficult for the providers to support.” This is a key point in the idea of IP re-use. These cores are being integrated into other designs. Stand alone, they might work fine, but then getting them into a system level with other blocks, there could be potential problems. For example, installing a manufactured processor into a motherboard for a computer. As a whole, the processor has already been designed and verified to work with the motherboard. All the verification and integration tasks have been taken care of. Within the processor, there might be several pieces of IP in the design that might have taken weeks to correctly verify and integrate. A distinction must be made between board level design, and chip level design. Just because the IP by itself works, does not mean that at the system level it will perform as designed. Other players in the industry feel the same way. Among them:

- **Colin Harris**, president of worldwide operations at PMC-Sierra Inc.: “[W]e typically spend about a third as much on IP evaluation as we do on the purchase itself” [4];
- **Ralph von Vignau**, director of technology and standards at Philips Semiconductors: “Clearly, buying IP today is still risky, despite all the internal approaches to evaluation that large companies have put in place” [4].

2.5 Measuring IP Quality

Despite acknowledgement that zero-defect IP is not at present possible (and perhaps might never be), much work has been done on trying to determine whether or not IP quality can be measured in a quantifiable way. In this case, quality refers to how suitable a particular IP core is for a customer’s specific purposes, and how well the core works without critical errors. Progress in this area has been somewhat haphazard: “Although the popularity of using third-party IP has grown, variations in quality remain problematic. Contributing to the situation, IP providers continue to wrestle with trade-offs between making their product available as soon as possible (for revenue-generating purposes) versus releasing product before it has been fully verified” [22].

This has become even more important as more and more complex chip arrays are being created for use in consumer products that demand more and more interactivity and functionality, products such as cell phones that act as digital image recorders and MP3 players. As Kaskowitz [23] puts it: “This further escalates the need to adopt ‘quality’ IP to reduce the risk of design errors due to the functionality requirements for today’s

complex designs...However, for reusable IP to be economically viable, it needs to easily fit into the target system-on-chip (SoC) environment, which requires standard interfaces, complete documentation and an IP verification environment to monitor and exhaustively test the IP functionality.”

One of the things that many IP core users have started to do, in order to create some sort of quality paper trail, is to put together a checklist to help quickly determine whether a particular IP core was suitable or not: “The problem with that solution from the vendor’s perspective, particularly for small vendors, is that each prospective customer has its own checklist. All the checklists may be superficially similar, but each requires a separate effort on the vendor’s part during an evaluation. That has led to calls for a uniform IP evaluation checklist” [4].

The first quality metric to appear was OpenMORE, based on the original *Reuse Methodology Manual* [15]. According to Werner [24]:

The spreadsheet focused on soft IP and while many valuable criteria were raised, the spreadsheet tended to blur the lines between deliverables, documentation, methodology, code requirements, scripting, and tool issues. It also did not completely address enough quantifiable checkpoints to ensure robust code that was reusable across tools and platforms.

It was at this point that the VSIA Quality Study Group and the Development Working Group (DWG) started working on what would become the VSIA QIP Quality Metric.

The VSIA QIP focused on four areas that were determined to be the major factors in IP quality. Labeled the “Quality Axes of the Quality Metric,” these factors were: authoring (design); verification process; IP maturity; and provider capabilities [24]. According to Werner [24]:

The QIP Quality Metric quantitatively provides the information needed for the end users to quickly compare and select suitable IP using a common baseline. QIP addresses the needs of the marketplace by providing an objective means to measure IP, reducing the integration time of quality IP, and providing a measurable method to guide IP vendors in the development of and to showcase the quality of their IP.

To assess any particular IP core, it is necessary to determine and then assign a quality value for each criterion. This is followed by the calculation of an overall quality using the scoring scheme. The use of automatic quality criteria checking allows for a speeded up process, thus lowering costs and producing results that are quantified and reproducible. The VSIA QIP metric has approximately 50% automatic criteria checking, using the following steps:

- Completeness of deliverables
- Coding guideline and design rule check
- Verification qualification
- Synthesis qualification
- Qualification framework

According to Vörg *et al.* [25], the use of this type of quality framework can help lead to “up to 95% integration effort reduction, the prospect of further benefits due to the use of qualification methods right from the start and less customer claims.”

2.6 VSIA Quality Metrics Assessment Workbook

The VSIA QIP Assessment Worksheet v2.0 is a very workable iteration of the original Quality Evaluation Spreadsheet (QES) put together by the VSIA Quality Study Group which came together in late 1999 [26]. The object of the group was to define IP

quality: “Quality is a measurable conformance to (customer or provider) specified requirements” [26].

The QES was designed to help the user evaluate a single IP; compare IP; evaluate entire SoC's. According to Birnbaum *et al.* [26]: “It can be used as an objective measure of quality (mandatory/recommended attributes), or as a more subjective grading system (attribute number grades) ...”

The creation of the VSIA QIP Quality Metric Assessment Workbook dates to the end of 2004. The Quality Metric is described as providing “a set of questions that elicit an objective measure of fitness for a Virtual Component (VC or IP Core)” [27]. The idea is to allow customers to determine differences (good, bad, or simply not compatible) among various IP cores. According to Mike Kaskowitz [28], president of the VSIA and general manager of Mentor Graphics' IP division: “Basically, this is a set of metrics used to assess the quality of IP providers. It lets companies look at several key differentiators and cuts their IP-selection process down to a couple of weeks from three to six months, without having to put them through a rigorous investigation. And it helps minimize risks when purchasing IP cores by helping people better understand the core's components.” Four types of IP cores can be assessed using the QIP: digital soft IP; digital verification IP; software IP; and analog hard IP.

The VSIA QIP 2.0 includes three spreadsheets. One evaluates the vendor's overall capabilities, one evaluates the design and verification practices of the IP design team, and the third evaluates the piece of IP being delivered.

Figure 1 below provides a sample page from the Summary Report section of the workbook. Questions on the spreadsheet are rated as “imperative,” “rule” or “guideline.” This lets users sort out showstoppers from “nice to have” items. The scoring keeps track of how many imperatives, rules, and guidelines are unsatisfied, and uses color coding to differentiate them.

A		B	C
4	Copyright © 2003, 2004, 2005 VSIA	IP Development Assessment	25%
5	Summary Report	Total	41%
6			
7	Release Version 2		
8		Imperatives not satisfied	
9		Rules & Guidelines not satisfied	138
10	Technical Support	Satisfied Imperatives, Rules & Guidelines	
11		Unanswered	0
12		Total	268
13			
14	Type in the name of the IP Vendor	SJSU EE Department	
15	Type in the name or part number of the IP	5 Bit Analog to Digital Converter	
16	Select the type of IP you are evaluating	analog IP	
17	Select the type of assesment you are undertaking	Vendor, Integration & Development	
18	Select the amount of information you wish to display	By category	
19		Score	%
20	Vendor Assessment	57	25%
21	VENDOR: Vendor Assessment	57	25%
22	IP Integration Assessment	0	0%
23	IP Development Assessment	1210	42%
24	ANALOG IP: IP Ease of Reuse (IP Integrator's View)	262	42%
25	ANALOG IP: Design & Verification Quality (IP Integrator's View)		
26	ANALOG IP: Design & Verification Quality (IP Developer's View)		
27	ANALOG IP: Design & Verification Quality (IP Integrator's View)		
28	ANALOG IP: Design & Verification Quality (IP Developer's View)		
29	ANALOG IP: Design & Verification Quality (IP Developer's View)	343	42%
30	ANALOG IP: Design & Verification Quality (IP Integrator's View)		
31	ANALOG IP: Design & Verification Quality (IP Developer's View)		
32	ANALOG IP: Design & Verification Quality (IP Integrator's View)		

Figure 1: Sample page from summary report of VSIA QIP metrics

As indicated, the summary provides a score and percentage scoring breakdown of quality issues such as: IP Ease of Reuse; Design & Verification Quality; IP Maturity; and Vendor Assessment. The Design & Verification Quality is further broken down to

specific Design Quality and Verification Quality percentages. As well, a breakdown of the Imperatives not satisfied; Rules & Guidelines not satisfied; and Satisfied Imperatives, Rules & Guidelines is provided. There are further breakdowns for each of the IP Ease of Reuse, Design Quality, and Verification Quality.

For the purposes of the above type of scoring report, definitions are provided with respect to Imperative, Rule, and Guideline. The item **Imperative** refers to questions that “concern component attributes that must be met; otherwise it may be impossible to use the component within a user project” [27]. The item **Rule** refers to questions that “concern component attributes that should be met. Failure to meet them may significantly impact the cost of using the component within a user project” [27]. The item **Guideline** refers to questions that “concern component attributes that if met will result in a general improvement in usability and maintainability of the component, or provide evidence that good practice has been used in the development of the component” [27]. Among the benefits provided by the VSIA QIP Metric are:

- The ability to quickly eliminate many offerings of IP and thus being able to concentrate on exactly what the user is looking for;
- The ability on the part of the vendor to simply answer a set of questions one time, providing the data needed to start the customizing process of the IP;
- The ability to “drill down” into the various scores and score weights to better quantify the IP.

According to Kathy Werner [4], reuse manager at Freescale Semiconductor Inc.: A big advantage to the VSIA QIP Metric is the communication it creates. Two vendors may

end up with similar overall scores on the metric but with strengths and weaknesses in different areas. With the metric, you have a quantitative way of discussing what the vendor needs to do to better meet your needs.”

- The ability for vendors to see how well they are perceived by customers versus their own perceptions of the products they sell or license;
- The ability to dig deeper once the first-pass checklist has been done.

2.7 Summary

From examination of the progress that has taken place in the design and verification of IP cores in the last half-decade, the literature is in agreement that the quality of reusable IP cores can now be tested to a fairly high degree of quantification. In this literature review, it has been shown that IP core technology has come a very long way within a span of five years or so. What was once a very haphazard process (both for designing and verifying IP cores) is now much more organized and standardized.

In this sense, researchers agree that the VSIA Quality Metric provides a very high level tool with which to determine the quality of an IP core. The use of this tool can in effect make a piece of IP better and more efficient to use. Once the IP has been quantified, possible bottlenecks and potential integration problems can be highlighted. As a result, if the IP is chosen for a design, these issues can be addressed beforehand, possibly reducing integration time and errors. However, most commentators and those within the industry still insist that even the use of such a Metric is only the first step in setting standards for IP core quality. Perhaps a more important one is the creation of an

auditing process: “[D]oing a design review with the vendor, retracing its steps, and comparing them with your own standards is vital. This auditing process can be quite invasive, with members of the user team reviewing design flows and—especially—verification flows in detail with the vendor’s engineering team” [4].

Nevertheless, the application of these standardized VSIA QIP spreadsheets provides a metric for testing of Sock IP cores. In the next chapter, the researcher lays out the methodology for conducting just such testing on several IP cores to determine if such testing is indeed worthwhile on a practical level, rather than simply theoretically. Then once this has been accomplished, the same cores will then be used in an actual chip design, so firsthand data on IP use/reuse can be commented on.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

In this chapter, a methodology is presented as a framework to be used in the conducting of IP core quality tests using the VSIA QIP Metric workbook. Briefly discussed are the research approach, research strategies and research time horizon, and research tactics (data samples, data collection).

3.2 Research Approach

In this study, the theory or research question to be examined is: **Is it possible, using present-day IP testing technology and scoring systems, to determine on a consistent and reproducible basis the quality of reusable Intellectual Property cores? That is, is it possible to determine quality IP in a way that can be used for future installations of such re-usable cores?**

The hypothesis used in the paper is that it is possible to determine the quality of reusable IP cores through the use of a VSIA QIP Metric Workbook.

With the IP blocks quantified in terms of quality and usability by the VSIA QIP, the research then goes one step further and collects data on the actual use and integration of the IP cores in a real world chip design. This will generate data that can then be commented on, and correlated with the VSIA QIP data.

The main research approach this paper adopts a deductive approach rather than an inductive one. A deductive approach is the dominant research approach within the

positivist philosophy, as it encourages development of a conceptual and theoretical structure prior to its testing through empirical observation [29]. The initial stage in the deductive approach is the development of testable propositions from the theory, which is then expressed in operational terms. The propositions or operational hypothesis are examined by using any form of valid empirical inquiry. The results are examined to confirm, modify, or abandon the theory. The deductive approach has some important characteristics, including the collection of quantitative data and testing of the hypothesis by using a highly structured methodology to facilitate replication.

3.3 Research Strategies/Timeline

The research strategy employed in this study is an experimental one.

Experimental studies typically involve the following:

- (i) the definition of a theoretical hypothesis;
- (ii) the selection of samples of individuals from known populations;
- (iii) the allocation of samples to different experimental conditions;
- (iv) the introduction of planned change on one or more of the variables;
- (v) the measurement on a small number of the variables; and
- (vi) control of other variables.

All these conditions are met in the introduction of two IP cores into the VSIA QIP Metric in order to determine their quality or lack thereof.

The time horizon of the present research is a cross-sectional one rather than longitudinal. In this instance, the research is a “snapshot” of a particular phenomenon or

set of phenomena at a particular time—a cross-sectional slice of IP core quality. There is no intent to measure the change and development in research issues over time.

3.4 Research Tactics: Data Samples and Data Collection

The raw primary data is from four pieces of IP. The first two pieces of IP are from former San Jose State University students. These blocks were designed, and documented, as IP blocks for use by other SJSU students. By taking these blocks through the VSIA QIP—for the purposes of this study, scores can be determined, and the use of these scores can then be used to quantify the IP, as well as highlight where the blocks are deficient in quality. The third and fourth pieces of IP are a standard cell library for synthesizing and placing the digital part of the chip design, and a pad frame for fabrication of the final chip design. These two pieces of IP will not be included in the VSIA QIP data collection. The VSIA QIP has not matured enough to be able to include these two types of IP. As mentioned before, the VSIA QIP is valid for digital soft IP, digital verification IP, software IP, and analog hard IP. The standard cell library and the pad frame do not fall into any of these categories, but their use and integration into the chip design will still be commented on in this research. The idea is to be able to quantify, at the end of the process, the right piece of IP for a project, through the use of the VSIA QIP Metrics, as well as from first hand experience.

The VSIA QIP on which the IP core data is tested consists of a Microsoft Excel worksheet that is filled out by the developer or integrator. As well, a Users' Manual is included with the Worksheet. As the questions are answered, a score is tallied, with

certain questions being listed as **imperatives**, or **rules**, or **guidelines**. Comments are also included with most questions. Thus the VSIA QIP is a set of metrics to help quantify and choose the right piece of IP. The VSIA QIP requires the use of excel 2002 or greater, with macros enabled.

Secondary data will be collected during the integration on the IP blocks during the chip design phase. This data will include any issues and pitfalls encountered during the integration and use of the IP blocks. This secondary data will be used to actually comment on the integration and use of the selected IP blocks. This data will be correlated with the VSIA QIP results showing how the VSIA QIP can actually help make a piece of IP better by highlighting possible issues before integration, as well as commenting on the issues of IP reuse as a whole.

3.5 Summary

In this chapter, the study has looked at the research methodology used, the research approach (deductive); the research strategies and time horizon (experimental and cross-sectional); and the research tactics for gathering and inputting data.

In the next section, the findings of the data from the VSIA QIP, as well as from the IP chip integration phase are examined in order to help determine whether or not IP core quality can be examined in a quantifiable and standardized way through the VSIA QIP metric, as well as comment on the actual issues and pitfalls experienced during the IP chip integration, and correlate this with the data generated from the VSIA QIP metric.

CHAPTER FOUR: RESULTS

4.1 Introduction

In this chapter, the IP blocks used in the research will be presented, along with the chip design used in the integration of the IP. Both the results of the data input (within the VSIA QIP Worksheet) and the findings resulting from IP integration are presented. This has been done to help determine whether or not IP core quality can be examined in a quantifiable and standardized way through the VSIA QIP Metric —and to highlight and comment on IP reuse issues in industry.

4.2 IP Blocks

4.2.1 *ADC/DAC Design IP*

The first two IP blocks used in this research were designed by students Daniel A. Brisco, Steven J. Corriveau, and Shawn E. Smith [30], and was provided by Dr. David Parent of San Jose State University [31]. The IP was packaged as an “ADC/DAC Design IP” [31] that “...consists of an ADC and a DAC for use in future DSP [digital signal processing] design projects at SJSU” [30]. The two physical IP blocks are composed of a 5-Bit Analog to Digital converter (ADC) as well as a 5-Bit Digital to Analog converter (DAC). Both pieces of IP are categorized as Hard IP. Hard IP is any IP verified in silicon and available as a physical layout in GDSII form. Figure 2 shows the layout of the 5-bit analog to digital converter.

Figure 2: 5-bit analog to digital converter IP block

Figure 3 shows the layout of the 5-bit digital to analog converter.

Figure 3: 5-bit digital to analog converter IP block

The IP was designed and manufactured using the AMI06 0.6 μm CMOS process. The IP also included a document with an overview of the design process for both pieces of IP; examples of the circuits used in their construction, as well results from testing and

verification. The IP blocks were manufactured through MOSIS [34] onto silicon and had been received and tested by the students before publication and release of their IP.

The IP was delivered as several Cadence Virtuoso Custom Design Platform [33] design directories (Figure 4).

Figure 4: ADC/DAC Design IP directories

The directories were composed of all three students work, along with all the cells, views, and testbenches created during the development and verification of the IP blocks. Also included with the IP was the final documentation of the IP authored by the students. The total IP package came across as complete, including all needed files and information for successful integration into a design.

4.2.2 Oklahoma State University Standard Cell Library

The third piece of IP used in this research was the Oklahoma State University Standard Cell Library [34]. It was designed and implemented by several Illinois Institute of Technology Electrical Engineering students and their professor Dr. James Stine [35]. This library of digital cells contained the physical logic gates used by the tools for the synthesis and then the placing and routing of the design. This standard cell library is free for use by Universities and is in the public domain. The library contained cells for four different technologies, but for this research the AMI06 0.6 μm CMOS process was used.

The library was complete and contained all the needed cells for a digital design. It was composed of 32 digital cells, ranging from D-type flip flops, latches, buffers, to AND, NAND, NOR, OR, XOR, AOI, and OAI gates. This library had been completely verified and tested during its creation and implementation for correct logic, timing and best area. The library distribution included all needed files for synthesis and place and route with different industry tools, documentation, and the needed files for correct export of the resulting design. Also included with the library distribution were reference designs and design flows for use with the library. The library was very professionally designed and implemented.

4.2.3 Tanner High ESD I/O Pads

The fourth and final piece of IP used in this research was the Tanner High Electrostatic Discharge (ESD) Input Output (I/O) pads [36]. These pads were the final piece of the chip design, and were used for the final connections before sending the chip

design out for fabrication. These pads were available directly from MOSIS [32], and are required for final submission for chip fabrication. They are fully documented and verified by Tanner Research Inc. [36]. These are industry grade I/O pads, and as a result are very professional and reliable.

4.3 Chip Design

The chip design used in the research was composed of a main digital DSP block that includes the use and implementation of the IP blocks under study. The digital DSP block was a 5-bit low pass finite impulse response (FIR) digital filter with built in self test (BIST), and utilized the ADC IP block on the inputs, and the DAC IP on the outputs (Figure 5). The filter was designed as a 16Khz low pass filter for audio applications. The FIR block was implemented and designed using the Verilog hardware description language. With the inclusion of the IP blocks under study, the entire chip design composed an ASIC DSP chip from taken from Verilog and IP, through to fabrication and silicon. Standard design practices were used during the design phase of the chip. Practices such as standard naming practices, version control, and RTL and netlist verification.

The design was taken through an entire design flow as follows:

- RTL implementation and verification
- RTL synthesis and verification
- Chip floorplanning and powerplanning
- Place and route of digital DSP block

- IP block integration
- IP block routing
- Final verification and tape out

For the purposes of this study, the most important and relevant part of the design flow was the IP block integration. The chip design was undertaken as to be able to gain first hand knowledge and experiences in IP integration and reuse, and as a result, be able to comment on them and apply them to the research and the data compiled from the VSIA QIP metrics. The final instance count for the DSP filter core was 498 instances, which encompasses several thousand transistors. Including the used IP, this was a quite a large design, and encompassed the majority of the available silicon within the chip package. The final ASIC DSP chip was implemented within a 40 pin package (Figure 5).

The entire design implementation length, not including design research, totaled about 8 days, with the longest part being the IP integration, and the debugging it entailed. It can be broken down as follows:

- RTL synthesis and verification: 1 day
- Chip floor planning, power planning, and place and route: 1 day
- IP block integration and routing: 3 days
- IP block debugging: 2 days
- Chip final verification and tape out: 1 day

The integration of the IP during the design cycle took up more than half of the total time for the implementation of the entire chip. The chip was successfully designed and implemented and released to MOSIS [32] for fabrication (Figure 5).

Figure 5: Completed ASIC chip design

The final chip design for this research sent off for fabrication consisted of the following blocks (Figure 5):

1. The I/O pads
2. The 5-bit analog to digital (ADC) converter IP block used in this research
3. The 5-bit low pass finite impulse response (FIR) digital filter

4. The 5-bit digital to analog converter (DAC) IP block used in this research
5. Another student design included on the chip for fabrication, but not part of this research
6. Compilation of different metals from AMI process used as fill for fabrication rule purposes

As can be seen in Figure 5, almost the entire chip core was utilized for this research.

Item number 5 represents another SJSU student design that was included within the chip core, as there was some area available for its inclusion. This design was completely separate from any of the functionality of the designed DSP chip used for this research, and utilized its own I/O pins.

The final design was completed successfully, and followed standard design flows and practices through to completion.

4.4 VSIA QIP Data

The VSIA QIP metrics were used to quantify two of the IP blocks used for this research, the SJSU ADC and DAC [30]. Vendors can use the VSIA QIP for their own in house use to have as a guideline for IP development. It questions the vendors on their own practices, and can measure the IP's fitness for reuse. Integrators use it to quantify IP that they might be using for a SoC design, or evaluating for purchase. The QIP is used as an objective tool, as it does not give a pass or fail score. The score is used as a guideline to help make an informed decision on the IP's quality [27]. The VSIA QIP could even be

used in a university setting to help teach robust design practices during a design processes, such as the SJSU ADC/DAC IP [30] used for this research.

After reviewing the IP and included documentation, the researcher filled out the VSIA QIP for the blocks in order to get a snap shot of the overall quality of the IP.

Figure 6 shows a snapshot of the summary page for the IP used in this research.

A		B	C
4	Copyright © 2003, 2004, 2005 VSIA	IP Development Assessment	26%
5	Summary Report	Total	41%
7	Release Version 2		
9		Imperatives not satisfied	
10		Rules & Guidelines not satisfied	138
11		Satisfied Imperatives, Rules & Guidelines	
12		Unanswered	0
13		Total	268
14	Type in the name of the IP Vendor	SJSU EE Department	
15	Type in the name or part number of the IP	5_Bit Analog to Digital Converter	
16	Select the type of IP you are evaluating	analog IP	
17	Select the type of assesment you are undertaking	Vendor, Integration & Development	
18	Select the amount of information you wish to display	By category	
19		Score	%
20	Vendor Assessment	67	25%
21	VENDOR VENDOR ASSESSMENT		
22	IP Integration Assessment	11	11%
23	IP Development Assessment	1,111	1,111%
24	ANALOG IP: IP Ease of Reuse (IP Integrator's View)	262	42%
25	ANALOG IP: Ease of Reuse (IP Integrator's View)		
26	ANALOG IP: Ease of Integration (IP Integrator's View)	188	43%
27	ANALOG IP: Design & Verification Quality (IP Developer's View)	343	42%
28	ANALOG IP: Design Quality (Internal Design)		
29	ANALOG IP: Design Quality (Design Data)	62	25%
30	ANALOG IP: Design Quality (Design Data)	236	25%
31	ANALOG IP: Design Quality (Design Data)		
32	ANALOG IP: Design Quality (Design Data)	55	22%

Figure 6: SJSU ADC/DAC IP VSIA summary report

From first inspection of the scores, the IP did not score very well across the board, and shows quite a few **imperatives** and **rules** that were not satisfied. But as stated above, the scores are to be taken objectively. The IP was designed and built by undergraduate university students for a final project. Quite a few of the questions within the VSIA QIP

are not applicable in this setting. The only verification protocols that were applied to the SJSU ADC/DAC IP [30] were written down and documented in a tutorial written by Dr. David Parent titled “Using IIT’s Standard Cell Library to Synthesize Top Down Designs Authored with Verilog” [37]. Though these verification protocols are sufficient for such a university design such as used in this research, they would only be a small subset of what would be used in an industry designed and verified chip. The students doing the design also did not have any company standards to adhere to, and they cannot be represented as vendors. In an industry setting, these would all be valid and important sections and considerations, but for this research specific items within these two sections will not be discussed in detail. The important sections of the VSIA QIP used for this research were the documentation quality, ease of integration, and verification quality. These three sections deal directly with the overall quality and implementation of the IP. These would have the most impact on the ease of reuse of the IP within the chip design used for this research.

All three of these sections did not score above 50%, so it is important to see where exactly the IP is deficient from looking at the scoring closely. Figure 7 shows a snapshot of the scoring for the documentation quality section. This is an important part of any piece of IP, as it is the main tool available to the IP integrator.

ID	Description	Assessment	Score	Assessment Comments	Importance
1	IP Base of Reuse (IP Integrator's View)		200		
1.1	IP Integration Manual		10		
1.1.1.1	Is an IP integration manual or chapter available?	y			Imperative
1.1.1.2	Does the IP integration section define the deliverables?	n		Does not list what and where each block resides, or their naming convention.	Imperative
1.1.1.3	Does the document fully define the interfaces to the IP?	n		The pins were not labeled on the actual physical IP, making determining their location extremely difficult.	Imperative
1.1.1.4	Does the document define naming conventions?	n			Rule
1.1.1.5	Is there a means of verifying the completeness of the deliverables?	n			Rule
1.1.1.6	Does the document describe how to instantiate the IP component?	n			Rule
1.1.1.7	Does the document provide floorplanning guidelines?	n			Rule
1.1.1.8	Does the document detail external design guidelines or restrictions?	n			Rule
1.1.1.9	Does the document describe a System Level Model and how to use it?	y		The document does a good job of explaining the functionality of the blocks.	Rule
1.1.1.10	Does the document provide instructions for how to build an environment using this component?	n			Rule
1.1.1.11	Does the document provide technology information, area and power?	y			Rule
1.1.1.12	Is the sense remote to IP?	n			Optional
1.1.1.13	Has the maximum resistance been specified?				
1.1.1.14	Is there a tool compatibility list in the implementation guide?	y			Guideline
1.1.2	Analog Hardware Reference Manual		40		
1.1.2.1	Is an IP reference manual available?	y		Contains datasheet only.	Imperative
1.1.2.2	Does the document contain a functional description of the IP?	y			Imperative
1.1.2.3	Does the document contain an Architectural Overview?	y		It provides a pin out diagram, but does not specifically describe each function. The documentation also does not divulge that there is a set pin for the IP, until the reader gets to the verification chapter.	Imperative
1.1.2.4	Does the document fully define the interfaces to the IP?	n			Imperative
1.1.2.5	Does the document detail the level of ESD protection provided?	n		Not applicable	Imperative
1.1.2.6	Does the document contain timing diagrams?	n	0		Rule
1.1.2.7	Does the document contain electrical characteristics?	y			Imperative
1.1.2.8	Does the document contain a floorplan abstract?	n		Does not give any layout information at all.	Imperative
1.1.2.9	Does the document describe operating and ancillary modes?	n			Imperative
1.1.2.10	Does the document define the bonding and package requirements and guidelines?	n		not applicable	Imperative
1.1.3	Release Notes Document		0		
1.1.3.1	Are there release notes available?	n		not applicable	Imperative
1.1.3.2	Are enhancements and improvements described?				
1.1.3.3	Are bug fixes described by the revision notes?				
1.1.3.4	Are known, unresolved and outstanding issue described?				
1.2.1	Configurability and Parameterization		0		

Figure 7: Documentation quality section

Within the document, red reflects an **imperative** that must be followed, and infers a “Danger” signal. If not followed, poor quality IP can be inferred and the IP is deemed unacceptable. Orange is a **rule** that infers “Caution” if not followed. Green infers a “Go” and represents a **rule** or **imperative** that has been followed [27]. Provided after each question is an assessment notes section for the IP integrator to make comments as the IP is being evaluated.

The documentation quality is made up of three sections; IP Integration Manual, Analog Hardware Reference Manual, and Release Notes Document (Figure 7). The IP

scored very badly in both the Integration Manual as well as the Analog Hardware Manual. The Release Notes Document was not applicable for this piece of IP.

4.4.1 Integration Manual Results

There were two main **imperatives** not satisfied within the Integration manual section that stood out the most to the researcher (Figure 8).

21	1.1.1.2	Does the IP integration section define the deliverables?	n	Does not list what and where each block resides, or their naming convention.	Imperative
22	1.1.1.3	Does the document fully define the interfaces to the IP?	n	The pins were not labeled on the actual physical IP, making determining their location extremely difficult.	Imperative

Figure 8: Integration manual section

When it came to the deliverables, the IP documentation did not specify where to find each block within the deliverables, along with what their names were within the directories. As the IP was delivered as several directories (Figure 4), each containing different cells used in the final implementation, this was an important consideration. This failure to follow, provide, and document standard naming practices could lead to difficulties tracking down the final iteration of the IP during integration. Regarding the interfaces of the IP, the physical pins on the DAC IP were not clearly labeled, which could lead to big problems during integration. The integrator would not know where to hook up the pins on the physical IP. Once again this shows a failure to follow, provide, and document standard naming practices during the design flow. These two not satisfied **imperatives** could represent significant problems during the integration of the IP into any design, and are definite “Danger” signals.

Several important **rules** were also not followed by the IP within the integration manual section (Figure 9).

24	1.1.1.5	Is there a means of verifying the completeness of the deliverables?	n		Rule
25	1.1.1.6	Does the document describe how to instantiate the IP component?	n		Rule
26	1.1.1.7	Does the document provide floorplanning guidelines?	n		Rule

Figure 9: Rules not followed in integration manual section

There was no mention within the IP documentation of how to verify if all the correct blocks and cells were delivered with the IP. Furthermore there was no mention of how to actually instantiate the component, along with guidelines on how best to prepare a floorplan using the IP. These rules are not necessarily deal breakers, but are definite “Caution” signals to the researcher, for without these pieces of information, the actual IP integration could become cumbersome, and lengthy.

On the positive side, within the Analog Hardware Reference Manual Section of the VSIA QIP, the IP did satisfy several of the important **imperatives** (Figure 10).

1.1.2.1	Is an IP reference manual available?	y	Contains datasheet only.	Imperative
1.1.2.2	Does the document contain a functional description of the IP?	y		Imperative
	Does the document contain an Architectural Overview?	y	It provides a pin out diagram, but does not specifically describe each function. The documentation also does not divulge that there is a set pin for the IP, until the reader gets to the verification chapter.	Imperative
1.1.2.3				

Figure 10: Imperatives followed in analog hardware reference manual

The IP documentation included a reference section that did include a good functional description of the IP blocks, along with an architectural overview of how the IP was developed and constructed. This is important information critical for correct IP integration, as some types of circuit architectures are not compatible with others, and the IP integrator needs this information up front in the design cycle. Even though part of the Architectural review was lacking in content, this can be construed as a “Go” for this IP.

4.4.2 Ease of Integration Results

Within the Ease of Integration section of the VSIA QIP metrics, the most important section to the researcher was the Analog Ease of Integration section (Figure 11). These rules represent the physical aspect of the IP, and how it was constructed.

119	1.2.9	Analog Ease of Integration Rules		85		
119	1.2.9.1	Is the datecode of the model library clearly identified?	n		Not applicable	Imperative
120	1.2.9.2	Are all I/O modeled in simulation?	y			Imperative
121	1.2.9.3	Are all I/O included or specified (where placed remotely from IP)?	n		Not applicable	Imperative
122	1.2.9.4	Are all names consistent with circuit schematics and layout?	y			Rule
123	1.2.9.5	Are effective ground senses routed away from gnd busses where IR errors or IR noise injection could come into play?	n	0		Rule
124	1.2.9.6	If the pad is external to the IP, is separate routing specified in the integration manual?	n	0		Rule
125	1.2.9.7	If the reference circuit ground should be Kelvin sensed at the bandgap or initiating reference, is the ground configured as such?	n	0		Rule
126	1.2.9.8	Do digital signals cross supply boundaries?	n			Optional
127	1.2.9.9	Are level shifters used for all digital signals which cross supply boundaries?				
128	1.2.9.10	Are power-ground pairs protected by diodes or are diodes specified if the I/O is remote?	y			Imperative
129	1.2.9.11	Are multiple analog supplies diode protected between supplies (e.g. VDDA to VDD)?	y			Imperative
130	1.2.9.12	Are reference inputs filtered and sufficiently buffered?	n		Not applicable	Imperative
131	1.2.9.13	Are the signal polarities coming into and leaving the block as defined in the naming standard?	y			Imperative
132	1.2.9.14	Do critical signals need to be shielded?	n			Optional
133	1.2.9.15	Have they been adequately shielded?				
134	1.2.9.16	Are the specified bias currents verified to match those specified in the Integration Manual?	n		No Mention of Biasing within Documentation	Imperative
135	1.2.9.17	Are there latch-up and guard rings on devices connected to I/O? Or is there at least 200 ohms in series with the output?	y			Imperative
136	1.2.9.18	Are there metal runners from subcircuits to pad?	n			Optional
137	1.2.9.19	Have loading conditions and IR drops been evaluated?				
138	1.2.9.20	Are outputs/inputs from subcircuits wide enough?	y			Imperative
139	1.2.9.21	Do all tracks in the IP satisfy electromigration requirements?	y			Imperative
140	1.2.9.22	Has the IP been simulated with extracted parasitics including all fill patterns?	y			Imperative
141	1.2.9.23	Are undefined or ill-conditioned input conditions modeled according to guidelines?	n	0		Guideline
142	1.2.9.24	Is the power down leakage current specified?	n	0		Rule
143	1.2.9.25	Is the quiescent supply current specified for each power/ground track?	n			Imperative
144	1.2.9.26	Can the simulation method demonstrate specifying disturbance signature to which the IP is sensitive?	n	0		Rule
145	1.2.9.27	Can the simulation method demonstrate specifying the IP resilience level to such disturbances?	n	0		Rule
146	1.2.9.28	Does the simulation view feature input noise ports?	n	0		Rule
147	1.2.9.29	Does the simulation view feature output noise ports?	n	0		Rule
148	1.2.9.30	Does the simulation view feature a noise transfer model to specific performances?	n	0		Rule

Figure 11: Analog ease of integration section

From first glance, there are several **imperatives** that are not satisfied, but most of them ended up not applicable to the IP. More important were the **imperatives** that had been satisfied. The most important **imperatives** for the researcher had to do with the routing and the simulation of the physical effects of the IP (Figure 12).

130	1. 2. 9. 20	Are outputs/inputs from subcircuits wide enough?	y		Imperative
131	1. 2. 9. 21	Do all tracks in the IP satisfy electromigration requirements?	y		Imperative
132	1. 2. 9. 22	Has the IP been simulated with extracted parasitics including all fill patterns?	y		Imperative

Figure 12: Rules followed in the analog ease of integration section

Having confirmation that the IP was routed with IR drop and electromigration effects taken into account is very important with any analog design. More importantly is having the information that the IP was simulated with extracted parasitics for more accurate final results. Both of these results represented a “Go” for this IP.

Several **rules** that were not followed within this section needed a closer inspection regarding their possible impact on the integration of the IP (Figure 13).

136	1. 2. 9. 26	Can the simulation method demonstrate specifying disturbance signature to which the IP is sensitive?	n	Q	Rule
137	1. 2. 9. 27	Can the simulation method demonstrate specifying the IP resilience level to such disturbances?	n	Q	Rule
138	1. 2. 9. 28	Does the simulation view feature input noise ports?	n	Q	Rule
139	1. 2. 9. 29	Does the simulation view feature output noise ports?	n	Q	Rule

Figure 13: Rules not followed within analog ease of integration section

By taking these **rules** objectively, the researcher came to the conclusion that these were more out of scope for this design and not applicable. Simulating noise aspects of the IP, along with disturbance signatures were not part of the verification protocols outlined within the SJSU tutorial [39] that the designers used. While definitely cause for a “Caution” light, they were deemed to be of not to much concern for the IP’s intended use in this research.

4.4.3 Verification Quality Results

The verification section of the VSIA QIP was one of the more important sections of the metrics for this research, and for any integrator considering a piece of IP. The integrator needs to know that all aspects of the IP have been tested and verified for

functionality before it can be successfully integrated into any SoC design. With this information, the IP can be used and integrated with a higher degree of confidence.

Overall the IP scored quite well within the verification quality section of the VSIA QIP metrics (Figure 14).

276	2.3	Verification Quality		28		
276	2.3.1	Verification		48		
277	2.3.1.1	Does the IP have error-free DRC and LVS logs?	y			Imperative
278	2.3.1.2	Have start-up and operating conditions been checked to avoid balanced differential common-mode latch-up?	n	0		Rule
279	2.3.1.3	Have you verified that matched transistors under metal is not problematic?	n	0	Not applicable	Rule
280	2.3.1.4	Have you verified that the power and ground busing within the IP can handle the peak current requirements at maximum frequency?	y			Rule
281	2.3.1.5	Are all test modes of the IP described including how to access them, specific instructions on their use and what tests can be run in each mode?	y		Only 1 mode	Rule
282	2.3.1.6	Are methods provided for testing or directly accessing the analog circuitry in the IP, including the signals and controls available, access registers and how to make measurements and use the results?	y			Rule
283	2.3.1.7	Are details of all characterization tests to be carried out on the IP described, including required external circuitry, estimated times to complete the tests and any assumptions made about the test environment?	y			Rule
284	2.3.1.8	After physical implementation of the IP, were all necessary functional and parametric performances simulated and validated against the schematic, including the effects of parasitic capacitance and resistance?	y			Rule
285	2.3.1.9	Are all IP specifications which cannot be guaranteed by the described tests explicitly labeled?	n	0		Rule
286	2.3.1.10	Can the simulation method demonstrate noise simulation at the device level?	n	0	Not applicable	Rule
287	2.3.1.11	Can the simulation method demonstrate noise simulation at the IP level, taking into account the noise characteristics of the lower levels?	n	0	Not applicable	Rule
288	2.3.1.12	Can the simulation method demonstrate noise simulation to check jitter and/or phase noise performances of clock systems?	n	0	Not applicable	Rule
289	2.3.1.13	Can the simulation method demonstrate the simulation of local dispersion with random distribution of parameters?	n	0	Not applicable	Rule
290	2.3.1.14	Can the simulation method demonstrate the IP sensitivity to local dispersion and its impact on yield?	n	0	Not applicable	Rule
291	2.3.1.15	Has the simulation been demonstrated for all process corners, voltages, and temperatures?	n			Imperative
292	2.3.1.16	Can the simulation demonstrate the worst case functional margin for head room corner?	n			Imperative
293	2.3.1.17	Is the circuit simulated with system models (cables and connectors)?	y		Simulated on a testboard, connected to digital oscilloscope	Imperative
294	2.3.1.18	If applicable, is the circuit simulated for suspend, resume and inactive modes?	n	0		Rule
295	2.3.1.19	If applicable, is the circuit simulated for mode switching glitch checking?	n			Imperative
296	2.3.1.20	Is the circuit simulated with imposed supply noise?	n	0		Rule
297	2.3.2	Coverage		10		
298	2.3.2.1	Has the design been verified to run over the specified voltage tolerance range?	y			Imperative
299	2.3.2.2	Has the design been verified to run over the specified temperature tolerance range?	n			Imperative
300	2.3.2.3	Has the design been verified to run over the specified process conditions?	n			Imperative
301	2.3.2.4	Has the design been specified to run over specified and documented PVT's?	n			Imperative
302	2.3.2.5	Has the design been analyzed for safety issues due to component failures? (e.g. combustible components)	n			Imperative

Figure 14: Verification section of the VSIA QIP metrics

There were several **imperatives** not satisfied that stood out and required further inspection (Figure 15). By taking the **imperatives** not satisfied objectively, and putting

them into the context of by whom the IP was developed, and what the IP will be used for, they were deemed satisfactory by the researcher.

291	2.3.1.1c	Has the simulation been demonstrated for all process corners, voltages, and temperatures?	n		Imperative
292	2.3.1.1d	Can the simulation demonstrate the worst case functional margin for head room corner?	n		Imperative
299	2.3.2.2	Has the design been verified to run over the specified temperature tolerance range?	n		Imperative
300	2.3.2.3	Has the design been verified to run over the specified process conditions?	n		Imperative
301	2.3.2.4	Has the design been specified to run over specified and documented PVT's?	n		Imperative
302	2.3.2.5	Has the design been analyzed for safety issues due to component failures? (e.g. combustible components)	n		Imperative

Figure 15: Imperatives not satisfied within verification quality section

The simulation process corners, voltages, and temperatures, along with the worst case functional margin for head room corner, were not applicable for this piece of IP. These are verification tests that were available to the IP designers, but had not yet been required or implemented. Further more, the chip design would not be used in any extreme settings, making these **imperatives** not applicable. While these would be “Danger” lights for industry and for large scale chip production, for this research they were deemed acceptable.

The **rules** that were satisfied were much more important and relevant to this section of the research (Figure 16).

2.3.1.4	Have you verified that the power and ground busing within the IP can handle the peak current requirements at maximum frequency?	y		Rule
2.3.1.5	Are all test modes of the IP described including how to access them, specific instructions on their use and what tests can be run in each mode?	y		Rule
2.3.1.6	Are methods provided for testing or directly accessing the analog circuitry in the IP, including the signals and controls available, access registers and how to make measurements and use the results?	y		Rule
2.3.1.7	Are details of all characterization tests to be carried out on the IP described, including required external circuitry, estimated times to complete the tests and any assumptions made about the test environment?	y		Rule
2.3.1.8	After physical implementation of the IP, were all necessary functional and parametric performances simulated and validated against the schematic, including the effects of parasitic capacitance and resistance?	y		Rule

Figure 16: Rules satisfied within verification quality section

The included IP documentation included several sections describing the tests performed, as well as the setup and schematics used for the testing. The designers even went so far as to design a test board specifically for the testing of the fabricated chip [30]. All necessary functional aspects of the physical design were verified against the schematic, correctly characterized, as well as simulated using all extracted parametric effects and resistances from the final physical layout. All of these satisfied **rules** inferred a “Go” for this IP.

As can be noticed from the filled out VSIA QIP pages, a determination was made that the blocks vary across the board in terms of overall quality. There were several **imperatives** that were not met within the VSIA QIP metrics, along with several important **rules** that were also not satisfied. These unmet **imperatives** highlighted possible problems that may be encountered during the use and integration of the IP. When all of these factors were taken objectively, the researcher made the determination that the IP was sufficient for use with the chosen chip design. The documentation was lacking, including clear and concise naming conventions, along with correct labeling of the physical IP, but included enough information to use correctly. The more important analog integration rules had been followed, and the most important verification steps had been taken to verify that the IP would perform as represented.

That is the key point to be made with respect to the VSIA QIP worksheet. It can be used to provide a comparative breakout of both good and the bad points—both for individual cores and as a way to match one core with another. As a result of this, the VSIA QIP can in effect make the chosen IP better, in that it highlights possible areas of

concern, preparing the integrator for these problems if and when they arise. It can be used to develop high quality IP, but it can also be used to quantify IP, and as a result make the IP more efficient to use and integrate. In addition to, the VSIA QIP has also shown it can help the end-user/integrator choose whether the IP under evaluation is the best, or at the least sufficient, piece of IP for his/her design.

4.5 IP Integration

The actual use and integration of the IP during the chip design was an important part of the research. Even though the VSIA QIP results were deemed satisfactory enough by the researcher to go ahead with the IP integration, it also informed the researcher that there could be potential problems with the integration of the IP into the chip design.

4.5.1 ADC/DAC Design IP

Some of the more important highlighted problems from the VSIA QIP metrics were as follows:

- Lack of integration section defining deliverables clearly
- Lack of the use of clear and concise naming conventions
- Lack of verification of the completeness of the IP deliverables
- Lack of complete documentation of all IP interfaces

These all became valid concerns as the IP was integrated into the chip design.

As previously presented, the IP was delivered as several directories for use within the Cadence Virtuosos Custom Design Platform [33]. The delivery itself was acceptable, as

this was the tool that the researcher used for the chip design, but if the IP integrator was using a different design environment, this would pose a serious problem. Upon further investigation of this issue, the researcher learned that the IP was available in GDSII form, thus clearing up this potential problem for future IP integrators. Using the IP as delivered across several hierarchical directories posed quite a problem for the researcher. As there was no defined list of deliverables within the IP documentation, the researcher had no idea from which directory to find the finalized IP. As can be seen from Figure 4, the delivered IP contained three directories containing the IP files: AMI06, AMI06_backup, and AMI06_DAC. To further complicate matters; the final IP was composed of cells taken from each of the three directories. This again posed quite a big challenge when it came time to write out the final GDSII file of the final chip design for fabrication. As the IP was actually made up of cells spread across all three directory hierarchies, the tool had trouble compiling all of these files across these directories, and as a result the GDSII file was being corrupted. This resulted in corrupted and invalid design data being released to the foundry for fabrication. This was caught by the researcher, and with considerable help from his professor, the problem was corrected. But this debugging process took two days to accomplish, and if it had not been caught, the chip design would have failed as a direct result of poor IP design. If the IP had followed clear standard naming conventions, included a documented section clearly defining the deliverables, and a method to verify that the deliverables were all complete and un-corrupted within the package, the researcher would have been able to identify the key parts of the IP, where they resided, and been able to rectify the problem in a much faster manner. If this was an industry chip

design, the time to market deadline could have been missed, and financial losses could have been substantial, all as a direct result of using the wrong piece of IP.

Due to the lack of complete documentation of all of the IP interfaces, the integration and routing of the IP blocks into the chip design took close to three days to accomplish. This was more than a third of the entire design cycle! The exact physical location of the pins on the DAC IP [30] was not documented anywhere within the IP deliverables, nor was it labeled on the physical layout of the IP (Figure 3). Due to this fact, the researcher had no knowledge of where to route and hook up the pins of the DAC IP [30] to the rest of the chip design. Fortunately the researcher located within one of the IP deliverables directories a routed example of the DAC IP [30] block, with fully routed input and outputs. With this information, along with the pin out diagram from the documentation, the researcher was able to reverse engineer the pin location information for successful routing of the IP into the chip design. Due to the lack of floorplanning information, the researcher also had no good idea on how to best orient the IP for best routing and placement. Both of these issues took quite a large section of the design cycle to accomplish, bringing the IP integration to a halt, which could have been avoided by the consistent use and documentation of standard naming conventions and practices. Without the routed example found within IP directory hierarchy, placed there intentionally or as an example, the researcher would not have been able to implement the IP core correctly without further assistance, and in a timely manner.

For this research, these were the most important and costly outcomes of the physical implementation and integration of the IP within the chip design. Due to these

problems, the total debugging time due to the integration of the IP was a total of 5 days, which equaled 62% of the total design cycle. This is a significant result, and highlights one of the main issues with IP reuse. The debugging of problems due to poorly documented IP, along with poorly packaged IP.

The VSIA QIP metrics highlighted possible concerns in all of these areas. Due to this fact, the researcher was aware of the documentation shortcomings, the failure of the physical IP not being labeled correctly, and the failure to follow and document standard naming conventions within the IP design, and thus was prepared for meeting these challenges during the IP integration into the chip design. Without the use of the VSIA QIP metrics to evaluate the IP beforehand, and not knowing of these possible problems before integrating the IP into the design, the design cycle would have been significantly longer, and the successful outcome of the chip design could have been further compromised.

4.5.2 Oklahoma State University Standard Cell Library

Though the OSU standard cell library was not able to be quantified by the VSIA QIP metrics, it was still an important piece of IP used for this research and bears some mentioning. The standard cell library could be construed as the most important piece of IP for any ASIC design. It represents the physical portion of the digital part of the design. How well these logic cells are designed directly relates to how well the circuit will perform in terms of timing and yield. A lot of work and verification goes into characterizing standard cell libraries for physical effects and timing before their release. This may be one of the reasons that they are not part of the VSIA QIP metrics. Industry

has reached a point in where there is a high amount of confidence in the standard cell library vendor's products, so much so, that they are generally used "right out of the box."

Even though the OSU standard cell library was not designed by an industry vendor, it was still used with a high degree of confidence in this research. It performed well and as expected. But it did have a potentially serious problem during its subsequent use. After all synthesis and place and routing had been finished, the place and route tool outputted a GDSII file of the FIR digital filter design. This file represented the physical aspect of the fully placed and routed design. This same file was then input into the Cadence Virtuoso Design Platform [33] environment for the final integration of the analog hard IP, as well as for final sign off and tape out of the chip for fabrication.

Bundled with the OSU standard cell library was a layermap file that defines all the physical layers of the design process used in the library, such as the metals and different doped regions of silicon. This file was used for correct translation of the GDSII file for input between tools. This file was either corrupted or when it was created, it was created with some unnoticed errors. As a result, when the FIR digital filter was inputted into the Cadence Virtuoso Design Platform [33] environment for final IP integration, the FIR digital filter design was missing a complete layer. This was not caught by the researcher, but by his professor when he went to review the final design. Debugging this error took the most part of a day to complete. When it was realized that the layermap file was incorrect, a substitute layermap file supplied with the Cadence Virtuoso Design Platform [33] environment was used to rectify the problem. Even though this was a simple error to fix, and there was nothing actually wrong with the functionality of the

original FIR digital filter design. If it had not been caught, the information sent out for fabrication would have been incomplete and corrupted, resulting in a complete failure of the chip design. This also brought the design cycle to a complete halt, and resulted in a longer design cycle due to this bug.

4.6 Summary

In this Chapter, the study has looked at the IP used in the research, the chip design that was used in the integration of the IP, along with the findings and results of the IP core tests performed through the use of the VSIA QIP metrics worksheet. These results were objectively broken down and commented on, and though the IP was found to be lacking in several areas, based on the VSIA QIP metric results, the IP was deemed sufficient for use within this research. The researcher also commented on the IP integration within the chip design and correlated it with the findings from the VSIA QIP metric results.

CHAPTER FIVE: CONCLUSIONS, FUTURE RESEARCH, REMARKS

5.1 Introduction

In this chapter, the study provides some general conclusions with respect to the IP core data and the VSIA QIP worksheet's ability to determine quality. As well, some future research directions are briefly mapped out—and the researcher offers some recommendations and comments on IP re-use, and the benefits of using the VSIA QIP to quantify IP.

5.2 General Conclusions

The paper has demonstrated all three of its objectives, showing (a) that quality standards are applicable and profitable for IP reuse; (b) that IP core assessment has been automated to the point where standardized forms are available so that IP cores can be measured and quantified; and (c) that the industry has reached a state whereby acceptable standard levels are widely available and implemented, with the sole provision that the “error-free” assessment of IP cores is not an achievable goal at the present time—and perhaps will never be.

The general discussion of IP quality stressed to point that, because re-use is so important today, the issue of quality standardization has come to the forefront of IP core discussions. As well, once one accepts the fact that quality IP is important, then it seems fairly obvious that setting the right standards can further help identify, choose, and develop good IP.

In the thesis research itself, the results of actual IP blocks being tested, using the VSIA QIP metrics, has been presented and shown to be able to quantify IP cores with a strong degree of probability. It has been shown that the VSIA QIP weighted scoring system, has worked as advertised, and with these scores analyzed in an objective way to determine what differentiates good from bad quality IP. These scores helped the researcher understand where the chosen IP for this research was deficient, and prepared him to address these problems during the actual IP integration into the chip design. Without this forehand knowledge, the design cycle would have been severely impacted and extended, thereby possibly missing the tape out date all together and resulting in possible failure of the entire chip design. In effect, the use of the VSIA QIP can help make a chosen piece of IP better, in that it highlights the problems before integration, thus reducing possible integration problems, and reduces debugging time.

Further more, with the actual use and integration of the IP into a real world chip design, general first hand statements of IP re-use can be made by the researcher:

- The IP must be well documented for correct usage and integration into designs.
- The IP deliverables must be arranged in a coherent and logical way for easy access and understanding.
- The use and documentation of clear and concise naming standards must be followed.
- Hard IP must include information and examples for the floorplanning and routing of the IP.

A document and standard such as the VSIA QIP is a very powerful tool that can be used in many settings, from industry to academia. In industry it could be used to help the development of in house IP by setting standards for the design teams follow and adhere to. This would make the integration of the IP created using the VSIA QIP standard more efficient and cost effective to use by different teams within the company. If used by IP vendors, it would create a higher degree of confidence in their IP by potential customers and integrators with the knowledge that the IP has been designed using an industry standard. In a university setting, it would be a good document for the same reasons, making utilization, and integration of the created IP easier for other students to use in projects and research. In addition to, it will also teach the students, and future engineers, best practices to follow during the design and creation of IP, which in turn they will take with them out into industry.

5.3 Suggestions for Further Research

One of the things that was not touched upon in this particular research was some type of analysis as to how usage and trust issues pertaining to the score can be addressed. In other words, one can argue that there is a gap between the marking of certain scores in a spreadsheet and the actual performance testing of the individual IP cores.

As well, a much larger sample of IP cores needs to be run through the VSIA QIP metrics [2] worksheet before a truly generalized set of results can be entertained.

The IP tool environment needs to be well documented within the IP, and its impact addressed. Not everyone in industry uses the same EDA tools for their

development and design of their chips. Because of this fact, future IP needs to take this into account and document the tools used in the IP's development, and reduce the possible impact if the IP is used in a different environment than the one it was developed on. This can be done by the use of widely accepted industry standards, and the reduction of the use of proprietary standards. The VSIA QIP metrics does touch on this area, but was not discussed in this research.

Finally, more research can be undertaken in an effort to match these scoring results with actual SoC outcomes developed with real world industry wide IP cores. In other words, does the test score reflect the IP core's performance under real conditions?

5.4 Some Concluding Remarks

It can be safely concluded that the use of the VSIA QIP worksheet is indeed a powerful step forward when it comes to identifying IP core quality before the cores are actually put in place in an SoC environment. We have seen how a score is created and how a better score can reflect a better IP core. That, however, does not indicate right off the bat that the particular core is the right one for a particular integrator or developer.

The VSIA QIP provides a further breakdown of strengths and weaknesses that an integrator can then match to his or her needs, and objectively assessed for each particular design. IP re-use will become a large sector of the Integrated Circuit Design industry as time moves forward. With the use of such metrics as the VSIA QIP, risk in the use of third party IP can be reduced to an acceptable level. IP re-use is an important and major

part of industry today. With transistors getting smaller and smaller, and designs getting bigger and bigger, IP re-use will have a larger and larger part to play in SoC chip designs.

References

- [1] VSI Alliance, Driving System Chip Productivity, 2005.
[Online]. Available: <http://vsi.org>. [Accessed Sept. 10, 2005].
- [2] VSI Alliance, “*QIP (Quality IP) Metric Version 2.0*” 2005.
[Online]. Available: http://vsi.org/qip_metric. [Accessed Sept. 10, 2005].
- [3] D. Malinak. “EDA in the Doldrums?” *Electronic Design*, June 16, 2003. [Online].
Available: <http://www.elecdesign.com/Articles/Index.cfm?ArticleID=5247>.
[Accessed Sept. 10, 2005].
- [4] R. Wilson. “The search for semiconductor IP intensifies,” *EETimesOnline*, June 20, 2005. [Online]. Available:
<http://www.us.designreuse.com/news/news10690.html>. [Accessed Sept 10, 2005].
- [5] M. Gogolewski. “Intellectual property now part of the design fabric,” *TechOnline*, May 18, 2005. [Online]. Available:
http://www.techonline.com/community/tech_group/38109. [Accessed Sept 11, 2005].
- [6] MIPS Technologies, Inc. “Choosing an intellectual property core,” July, 2002.
[Online]. Available:
http://www.mips.com/content/PressRoom/TechLibrary/WhitePapers/choosing_ip.
[Accessed Sept 11, 2005].
- [7] M. Santarini. “DAC panelists bemoan functional-verification ‘hell’,” *EDN: Voice of the Electronics Engineer*, June 16, 2005. [Online]. Available:
<http://www.edn.com/article/CA608694.html>. [Accessed Oct 10, 2005].
- [8] D.D. Gajski, A.C. Wu, V. Chaiyakul, S. Mori, T. Nukiyama, & P. Bricaud,
“Essential issues for IP reuse,” in *ASP-DAC (Asia and South Pacific Design Automation Conference)*, Yokohama, Japan, 2000, pp. 37-42.

- [9] W. Savage, J. Chilton, & R. Camposano, "IP reuse in the System on a Chip era," in *13th International Symposium on System Synthesis*, 2000, pp. 2-7.
- [10] P.J. Bricaud, "IP reuse creation for System-on-a-Chip design," in *IEEE 1999 Custom Integrated Circuits Conference*, 1999, pp. 395-401.
- [11] T. Thomas, "Technology for IP reuse and portability," *IEEE Design & Test of Computers*, (October-December), 1999, pp. 7-13.
- [12] R. Seepold, N. Martinez Madrid, A. Vörg, W. Rosenstiel, M. Radetzki, P. Newmann, and J. Haase, "A qualification platform for design reuse," presented at Proceedings of the International Symposium on Quality Electronic Design, 2002, pp. 1-6.
- [13] C. Yun, X. Wie, & L. Ping, "Hierarchical hard IP reuse in SOC," *IEEE*, 2002, pp. 1449-1453.
- [14] M. Keating, "The IP quality revolution," presented at Proceedings of the Fifth International Symposium on Quality Electronic Design, 2004.
- [15] M. Keating, & P. Bricaud, *Reuse Methodology Manual*. Boston, MA: Kluwer Academic Publishers, 2002.
- [16] C. Jones, *Applied Software Measurement: Assuring Productivity and Quality*. New York: McGraw Hill Text, 1996.
- [17] M. Keating, "IP quality: a design, not a verification problem," presented at Proceedings of the Sixth International Symposium on Quality Electronic Design, 2005.
- [18] S. Mallya, & K. Yamada, "Chip design challenges Using hard and soft third-party IP," *Electronic News (North America)*, 1999, pp. 20-21.

- [19] R. Stolzman, "Understanding Assertion-Based Verification," *TechOnLine*, August 16, 2002. [Online]. Available: http://www.techonline.com/community/related_content/21077. [Accessed Oct 10, 2005].
- [20] D. McGrath, "IP verification panel advocates standards," *EETimesOnline*, June 15, 2005. [Online]. Available: <http://www.eetimes.com/conf/dac/showArticle>. [Accessed Oct 10, 2005].
- [21] R. Jeffries, "What is Extreme Programming," *XProgramming.com*, November 8, 2001. [Online]. Available: <http://www.xprogramming.com/xpmag/whatisxp.htm>. [Accessed Feb 9, 2006].
- [22] J. Ensell, "Addressing the IP quality problem: FSA and VSIA link hands to tackle the issue," *Chip Design Magazine* (February-March), 2004. [Online]. Available: <http://www.chipdesignmag.com/print.php?articleId=18>. [Accessed Oct 10, 2005].
- [23] M. Kaskowitz, "Standards-based IP," *EETimesOnline*, May 28, 2004. [Online]. Available: <http://www.eet.com/showArticle>. [Accessed Oct 11, 2005].
- [24] K. Werner, "Can IP quality be objectively measured?" presented at Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum, 2004, pp. 1-2.
- [25] A. Vörg, M. Radetzki, & W. Rosensteil, "Measurement of IP qualification costs and benefits," presented at Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2004.
- [26] M. Birnbaum, & C. Johnson, "VSIA quality metrics for IP and SoC," presented at International Symposium on Quality Electronic Design, 2001, pp. 279-283.
- [27] P. Rose, *QIP Quality Metric User Guide, Rev. No: 2.0*. Los Gatos, CA: VSI Alliance, Inc., 2004.

- [28] J. Berman, "Metrics measure quality of IP companies and cores," *EDN: Voice of the Electronics Engineer*, June 6, 2005. [Online]. Available: <http://www.edn.com/article/CA606306.html?spacedesc=newProducts>. [Accessed Oct 12, 2005].
- [29] J. Gill, and P. Johnson, *Research methods for managers*. London: Paul Chapman Publishing, 1997.
- [30] D. Brisco, S. Corriveau, & S. Smith, *ADC/DAC Design IP*. San Jose State University, CA. May 26, 2005. [Online]. Available: <http://www.engr.sjsu.edu/dparent/ICGroup/Case%20Studies/a2d5bit.pdf>. [Accessed Sept 12, 2005].
- [31] D. Parent, "Home page - San Jose State University, Dr. David Parent," 2005. [Online]. Available: <http://www.engr.sjsu.edu/dparent/>. [Accessed Sept. 10, 2005].
- [32] MOSIS, Integrated Circuit Fabrication Service, 2005. [Online]. Available: <http://www.mosis.org/>. [Accessed Sept. 10, 2005].
- [33] Cadence Virtuoso Custom Design Platform, San Jose, CA. 2005. [Online]. Available: http://www.cadence.com/products/custom_ic/index.aspx?lid=cic. [Accessed Sept. 10, 2005].
- [34] Oklahoma State University, Oklahoma Standard Cell Library, Version 2.4, 2005. [Online]. Available: <http://vcag.ecen.okstate.edu/projects/scells/>. [Accessed Sept. 10, 2005].
- [35] J. E. Stine, J. Grad, I. Castellanos, J. Blank, V. Dave, M. Prakash, N. Iliev, and N. Jachimiec, "A Framework for High-Level Synthesis of System-on-Chip Designs," *International Conference on Microelectronic Systems Education*, IEEE Computer Society, pp. 11-12 2005.

- [36] Tanner Research Inc., High ESD I/O Pad Libraries.
[Online]. Available:
http://www.tanner.com/CES/products/tech_setups/high_esd.htm. [Accessed Sept 10, 2005].
- [37] D. Parent, *Using IIT's Standard Cell Library to Synthesize Top Down Designs Authored with Verilog*. San Jose State University, CA. [Online]. Available:
<http://www.engr.sjsu.edu/dparent/ICGroup/Tutorials/MOD4.pdf>. [Accessed Sept 10, 2005].

Appendix A: Chip Design Tutorial

The entire 5-bit FIR filter design was coded using the new Verilog 2001 features, which includes the new signed arithmetic feature (2's compliment). This signed feature can be used on reg and net declarations, as well as on module port declarations. This new feature was used throughout the coding of this design, to designate that all additions and multiplies were going to be signed. This in turn lets the tool know to use signed adders and multipliers. This takes care of a lot of the extra coding that might need to be done to make sure that the signed arithmetic is handled correctly. Once this was implemented, the coding of the Verilog for the design was very straight forward. The total line for the Verilog code was approximately 100 lines.

The first task was to set up the correct size registers for the delay elements, the addition and multiply registers, the coefficient registers, and the input and output registers. Once these had been calculated, it was a straight forward process to handle the multiplication of the coefficients, then the addition/summation of the products. One last piece of coding that had to be implemented was the sign extension of the products for their final addition/summations. With this coding in place, it was then a simple task to implement the pipeline/delay registers, along with the dataflow aspect of the filter design. Some simple signals were added to the design, such as a clock enable signal, along with the reset signal. All in all, the coding was a straight forward piece of work that was easy to implement, with any coding errors fixed during synthesis.

The CAD tool for synthesis used for the design was RTL Compiler 5.2 by Cadence Design Systems. This is the new breed of synthesis tools that is, from the ground up, inherently different than Synopsys Design Compiler. It was designed with entirely new synthesis algorithms that use different approaches to synthesis. RTL Compiler does what is known as Global Synthesis, where by it takes the whole design into account, speed, area, timing, etc. As a result, it consistently gets better results than Design Compiler. That is why it was chosen as the tool for this design. It also has very powerful tools built in that can choose the best type of datapath elements to use, based on the constraints given the tool. It is a very advanced tool with many aspects to it. As a result, the tool chose the datapath elements (adders, multipliers), as it is setup to maximize the design, and choose the appropriate parts for a successful synthesis run. RTL Compiler does not need to be “scripted” as much as Design Compiler. Almost all of the important attributes and synthesis enhancements are turned on by default, and taken care of “under the hood.”

Synthesis

The basic RTL Compiler work flow is as follows:

- Load HDL files and Technology Libraries
- Perform Elaboration
- Apply Constraints
- Apply Optimization Settings
- Synthesize

- Analyze
- Export Design

RTL Compiler (RC) is started/invoked from the UNIX prompt by typing RC:

```
Unix> rc
```

By default RC will create a logfile titled rc.log, and a command file titled rc.cmd within the current working directory. The logfile will contain the log of the entire RC run, while the cmd file will contain a log of all commands passed to RC.

The majority of RC settings are done via design attributes. These attributes set up the tool with the correct settings for synthesis. The technology library is one such attribute, and the one used for this design was the OSU Standard cell library, AMI06 library. All information regarding the standard cell libraries, their contents and use, can be found on their website, <http://vcag.ecen.okstate.edu/projects/scells/>. RC only takes liberty format technology libraries (.lib) and the .lbr format. To set the target technology library type the following in RC:

```
rc: / > set_attribute library osu05_stdcells.lib
```

This will point the tool to the standard cells to be used for mapping the design to the chosen technology.

The next step is to read in the verilog RTL file of the design and to elaborate it:

```
rc: /> read_hdl -v2001 filter5.v
rc: /> elaborate filter5.v
```

The -v2001 switch was used in this instance, as the newer verilog 2001 extensions were used in the verilog coding of the FIR filter design. The elaborate command builds the data structures, infers registers in the design, and checks the semantics of the verilog

code. If there is anything wrong with your verilog, RC will inform you. Once the design is successfully elaborated, RC will have an internally created structure for the verilog design. This internal structure can be navigated similar to a UNIX environment directory structure by using commands as `ls` and `cd` to navigate. There will be a design section where the elaborated design resides and can be viewed hierarchically, a technology library section, where you can view the loaded in libcells and all their attributes, as well as other sections (if used) for design for test, low power, etc.

With the design elaborated within the tool, the next step is to apply the design constraints, such as the clock waveforms, input and output delays, and any other constraints. This can be done through the use of an SDC (Synopsys Design Constraints) file, which is an industry, standard, or through the RC interface itself:

```
rc: / > define_clock -name clk -period 10000 [find / -port clk]
rc: / > external_delay -in 1000 -clock [find / -clock clk] /designs/filter5/ports_in/*
rc: / > external_delay -out 1000 -clock [find / -clock clk] /designs/filter5/ports_in/*
```

The first command defines the name of the clock to be used within RC, the period of the desired clock, along with applying it to the name of the `clk` port within the elaborated verilog design using the `tcl find` command. RC uses `PICOSECONDS` as its time base and not `NANOSECONDS` like other tools. All times, and capacitances, within RC are reported with picoseconds and picofarads. Thus, the above clock period represents a 100MHZ clock waveform, with a default duty cycle of 50%. The next two commands set an input and output delay for the design of 1 nanosecond.

Once the design constraints have be set on the design, further optimization constraints can be set. This would include any low power settings such as clock gating,

multi-vt optimization, multi-supply/source voltage flow, design for test, and many others.

None of these were used in this design. One optimization that was used was to create and define timing groups into three distinct cost group paths: input-to-register/clock (I2C), register-to-register (C2C), and register-to-output (C2O):

```
rc: /> define_cost_group -name I2C -design filter5
rc: /> define_cost_group -name C2O -design filter5
rc: /> define_cost_group -name C2C -design filter5
rc: /> path_group -from [all::all_seqs] -to [all::all_seqs] -group C2C -name C2C
rc: /> path_group -from [all::all_seqs] -to [all::all_outs] -group C2O -name C2O
rc: /> path_group -from [all::all_inps] -to [all::all_seqs] -group I2C -name I2C
```

The first set of commands defines the name of the cost group, while the second defines the path group for each. By creating these cost/path groups, the worst timing path in each group drives the synthesis, and each will be worked on separately from each other.

Now that the technology libraries have been loaded into RC, the verilog RTL design has been loaded in and elaborated, the clock period and input/output delays defined, it is now time to synthesize the design. The synthesis is composed of three main operations. Synthesize to generic, whereby RC synthesizes the design to primitive verilog gates (and, or, nor) and generic sequential elements. The next stage is the mapping stage, whereby RC maps the generic design to the actual technology library cells. The final stage is an incremental optimization stage, where by RC tries different optimizations on localized parts of the design to incrementally optimize, first for timing, second for power, and finally for area. This can be done by issuing the following commands:

```
rc: /> synthesize -to_generic
rc: /> synthesize -to_mapped -no_incremental
rc: /> synthesize -to_mapped -incremental
```

The above commands will first synthesis to generic, and if you wish, you can inspect the design before synthesizing the mapped cells. If you want to go straight to gates, you only need to do:

```
rc: / > synthesize -to_mapped
```

RC will automatically go through the synthesize to generic step through to incremental optimization. The benefit of breaking it down into its three parts, is that you can inspect the design at each stage, write out the netlist and timing reports, and make adjustments as necessary. During synthesis, if there are any problems, RC will output a message, and it will also output messages and on the progress of the run.

Once there has been a successful synthesis run, it is time to analyze the result, see if the design meets the designed for specs, and output the final netlist:

```
rc: / > report timing > filter5_timing.rpt  
rc: / > report area > filter5_area.rpt  
rc: / > report gates > filter5_gates.rpt  
rc: / > report power > filter5_power.rpt  
rc: / > report qor > filter5_qor.rpt
```

The above commands will print out reports on the timing, area, the number of gate and type used, the power consumption of the design, along with a quality of results (qor) report. The qor report will show all of the above reports in a short summary. You can use all of this information to decide if the design is meeting all of its specifications. If it is not, another synthesis job will need to be run with different settings to meet the desired specifications. One might need to change the clock period, or even run the synthesis job at a higher effort.

Once the design has been correctly synthesized into a gate level netlist, the netlist needs to be written out along with all of the design constraints used in the synthesis run.

```
rc: / > write_hdl > filter5.vg  
rc: / > write_sdc > filter5sdc
```

The resulting files will be a gate-level netlist containing the entire design mapped to the provided technology library cells, along with an sdc constraints file, that contains all of the timing and input and output delay information, along with any other design constraints that were applied to the design. This final netlist can then be used within any simulation tool, such as Modelsim, to verify its functionality using the same testbench created for verifying the original verilog RTL. Another tool for checking the gate level netlist is Cadence Conformal Logic Equivalence Checking tool (LEC). The tool compares the RTL vs. the gate level netlist for logic equivalency. It is a quick and easy way to verify that the synthesized netlist logically matches the supplied verilog RTL. RC also has one more feature, in that it can output all the needed configuration files for the

Cadence First Encounter Place and Route tool:

```
rc: / > write_encounter -design filter5 -basename filter5/filter -lef  
osu05_stdcells.stacks.lef
```

The two above commands will point RC to the physical lef needed for the place and route tool, and will also write out a folder containing all the needed information for starting the place and route tool. Included inside this folder is a configuration file that will load the netlist, sdc file, constraints, and libraries into the tool automatically. The synthesis of the design is now complete.

All of the above commands can be combined into a script text file and loaded into RC upon starting the tool, or from within the tool itself:

```
UNIX> rc-f my_script.g  
or  
rc: /> include my_script.g
```

A template that can be used as a starting point for creating your own rc compile script, containing all the basic commands and attributes, can be written out from rc by entering the following command:

```
rc: /> write_template -outfile my_template.g
```

Using this template is the best starting point for putting together a complete script for a successful synthesis run. Below are the contents of the script used for compiling the FIR filter used in this thesis:

```
##### Template Script for RTL->Gate-Level Flow  
##### Include TCL utility scripts..  
include load_etc.tcl  
##### Set up  
set DESIGN filter5  
set SYN_EFF medium  
set MAP_EFF medium  
set DATE [exec date +%m%d.%H%M]  
##### The following variables are used for diagnostic purposes only.  
##### They do not have any effect on the results of synthesis.  
##### Setting 'map_fancy_names' to 1 tells the tool to name the combinational  
##### cells based on the criteria that was used when selecting them.  
set map_fancy_names 1  
##### Setting 'iopt_stats' to 1 prints out statistics during incremental  
##### optimization.  
set iopt_stats 1  
  
set _OUTPUTS_PATH outputs  
set _REPORTS_PATH reports
```

```

set_attribute lib_search_path
/home/paulweil/school_files/osu_stdcells/lib/ami05/signalstorm
set_attribute information_level 7 /

set_attribute library osu05_stdcells.lib

#### Turn on TNS, affects global and incr opto
set_attribute endpoint_slack_opto true /

read_hdl -v2001 filter5_final.v
elaborate $DESIGN
puts "Runtime & Memory after 'read_hdl'"
timestat Elaboration

check_design -unresolved

set_attribute preserve true /libraries/osu05_stdcells/libcells/PAD*

define_clock -n clk -p 33333 [find / -port clk]

external_delay -in 1000 -clock [find / -clock clk] /des*/filter5/ports_in/*
external_delay -out 1000 -clock [find / -clock clk] /des*/filter5/ports_out/*

report timing -lint

if {[llength [all::all_seqs]] > 0} {
    define_cost_group -name I2C -design $DESIGN
    define_cost_group -name C2O -design $DESIGN
    define_cost_group -name C2C -design $DESIGN
    path_group -from [all::all_seqs] -to [all::all_seqs] -group C2C -name C2C
    path_group -from [all::all_seqs] -to [all::all_outs] -group C2O -name C2O
    path_group -from [all::all_inps] -to [all::all_seqs] -group I2C -name I2C
}

#### Synthesizing to generic
synthesize -to_generic -eff $SYN_EFF
puts "Runtime & Memory after 'synthesize -to_generic'"
timestat GENERIC

#### Synthesizing to gates
synthesize -to_mapped -eff $MAP_EFF -no_incr
puts "Runtime & Memory after 'synthesize -to_map -no_incr'"
timestat MAPPED

```

```

#### Incremental Synthesis
synthesize -to_mapped -eff $MAP_EFF -incr
puts "Runtime & Memory after incremental synthesis"
timestat INCREMENTAL

report timing > $_REPORTS_PATH/${DESIGN}_timing.rpt
report area > $_REPORTS_PATH/${DESIGN}_area.rpt
report gates > $_REPORTS_PATH/${DESIGN}_gates.rpt
write_hdl > $_OUTPUTS_PATH/${DESIGN}.vg
write_sdc > $_OUTPUTS_PATH/${DESIGN}.sdc
write_script > $_OUTPUTS_PATH/${DESIGN}.script
report power > $_REPORTS_PATH/${DESIGN}_power.rpt
report qor > $_REPORTS_PATH/${DESIGN}_qor.rpt

set lef_list
/home/paulweil/school_files/osu_stdcells/lib/ami05/lib/osu05_stdcells.stacks.lef

write_encounter design $DESIGN -basename filter5/filter -lef $lef_list

puts "Final Runtime & Memory."
timestat FINAL
puts "=====
puts "Synthesis Finished ....."
puts "=====

quit

```

Place and Route

Below are two links to excellent tutorials in the use of Cadence First Encounter Place and Route tool. Both tutorials were used in the chip design flow for this thesis, and that is why they are being included here.

The first is from one of the authors of the Oklahoma State University Standard Cell Library, Johannes Grad. It is a very sort and succinct tutorial on how to get up and running with the tool, using the encounter configuration file to set up the tool, then moves through the place and route flow from floorplanning to final output of a GDSII file for the

foundry. Included are also example files so that the tutorial can be run by anyone on their own setup.

<http://www.chiptalk.org/modules/wfsection/article.php?articleid=1>

The second link is to a tutorial from a University of Southern Carolina professor by the name of Dr. Jason Bakos. It goes through the entire design flow from floorplanning through to final output, but also spends a little more time on talking about clock tree synthesis and meeting timing constraints. At the end of the tutorial is also a brief explanation on how to import the final design back into Cadence Virtuoso.

<http://www.cse.sc.edu/~jbakos/612/tutorials/encounter.shtml>