

**Developing and Documenting the Chip Design, Fabrication, and Bring-up Process
for a Vertically-integrated Co-curricular**

by

Jackson Macguire Hafele

A Creative Component submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering (Computing and Networking Systems)

Program of Study Committee:
Henry Duwe, Major Professor

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2025

Copyright © Jackson Macguire Hafele, 2025. All rights reserved.

DEDICATION

I would like to dedicate this paper to my lovely fiancé Caitie and my cat Mr. Nutter Butter.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vi
ACKNOWLEDGMENTS	vii
ABSTRACT	viii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. RELATED WORK	3
2.1 Co-curricular Impacts	3
2.2 Vertically-Integrated Curriculum	4
2.3 ASIC Design Process	4
2.4 Documentation	5
2.5 Tutorials	7
CHAPTER 3. MOTIVATION	8
3.1 Lack of Existing Experience	8
3.2 Vertically Integrated	9
3.3 MPW Shuttle Comparison	9
3.4 Active Tools and Processes	11
CHAPTER 4. MOTIVATION IN CONTEXT	12
4.1 ISU Coursework	12
4.1.1 CPRE 2810	13
4.1.2 CPRE 2880	13
4.1.3 CPRE 3810	14
4.1.4 CPRE 4000+	15
4.1.5 EE 4650	15
4.1.6 EE 3300/4350	16
4.2 eFabless	16
4.2.1 OpenMPW	16
4.2.2 ChipIgnite	18
4.2.3 Provided Open Source Tools	19
4.2.4 Provided Documentation	20
4.2.5 Provided Tutorials and Webinars	22
4.3 Student Groups	23

4.3.1	Senior Design Teams	23
4.3.2	First-year Honors Mentor Program	24
4.4	ChipForge	25
4.4.1	Club Structure	25
4.4.2	ASIC Design Cycle	26
4.5	ChipForge Tools	27
4.5.1	Toolchain	28
4.5.2	Command Line Utility	28
4.5.3	Custom Hardware	28
4.5.4	FPGA Gen	31
4.5.5	Camera Interface	31
CHAPTER 5. MAIN IDEA		33
5.1	Structure and Best Practices	33
5.2	Tools Used	34
5.2.1	Markdown Format	34
5.2.2	Docsify	34
5.2.3	Gitlab Pages Hosting	35
5.2.4	CircuiTikz	35
5.3	Documentation Pages	36
5.3.1	Landing Page	36
5.3.2	Introduction	37
5.3.3	Digital	39
5.3.4	Analog	40
5.3.5	Bring-up	41
5.3.6	Hardware	43
5.3.7	Standards	43
5.3.8	Meta	44
CHAPTER 6. EVALUATION		45
6.1	Methods	45
6.1.1	Informal Feedback	45
6.2	Results	46
6.2.1	Informal Feedback	46
CHAPTER 7. FUTURE WORK		48
7.1	Additional Documentation	48
7.2	Maintenance	49
7.3	Active Feedback	50
7.4	Relation to Coursework	51
CHAPTER 8. CONCLUSION		52
REFERENCES		53

LIST OF TABLES

	Page
Table 3.1 MPW Shuttle Services	10

LIST OF FIGURES

		Page
Figure 4.1	Caravel Development Board (Black) and Caravel Breakout Board (White) . . .	18
Figure 4.2	sddec22-17 OpenMPW Fabricated Chips	24
Figure 4.3	ChipForge Design Cycle	27
Figure 4.4	ChipForge Development Board	29
Figure 4.5	ChipForge Memory PMOD	30
Figure 4.6	ChipForge Seven Segment Display PMOD	30
Figure 4.7	Internal FPGA Board	31
Figure 4.8	Bring-up Camera Interface	32
Figure 5.1	CircuiTikz Tri-State Buffer Schematic	36

ACKNOWLEDGMENTS

I would like to take a moment and thank everyone who has guided me through my graduate research. First I would like to thank Dr. Henry Duwe, who has provided excellent teaching and mentorship throughout my time at Iowa State University. I would also like to thank Gregory Ling, who has been an amazing partner in clubs, coursework, and research. I would also like to thank all of the friends and family who have provided me support throughout my education at Iowa State. Finally, I would like to thank all of the students I have worked with, who have the patience to sit through my teaching. This work was supported in part by the U.S. National Science Foundation under Grant 2144757.

ABSTRACT

At Iowa State University (ISU), there is a lack of existing resources and organizations which provide engineering students with hands-on Application Specific Integrated Circuit (ASIC) design experience. With the establishment of an officially recognized student organization, ChipForge, students are now be able to design fully-funded digital and analog ASICs with open-source tools. To support a wide range of students with varying majors, experience levels, and interests, additional documentation is required for ChipForge. Actively maintained documentation is needed by ChipForge members to learn how to translate their skills to use the required open source tools, which are not covered in the ISU curriculum. Additionally, students will need a source of information to learn how to use the internally developed tools from ChipForge for ASIC design, verification, and validation processes.

To create effective documentation, related works are reviewed to determine best practices for software engineering documentation. Additionally, the context of the ECpE curriculum within Iowa State University is considered, with existing classes and work that students have completed through the university. By utilizing existing literature and the catered experience of Iowa State University students, documentation can be made which assists students in the ASIC design process, without becoming overwhelming. Informal feedback has been collected to evaluate the impact of the created documentation, with proposed changes in how to change the methodology of the documentation to ensure it is accessible and beneficial for students in the newly established co-curricular club ChipForge at Iowa State University.

CHAPTER 1. INTRODUCTION

Within the Electrical and Computer Engineering (ECpE) department at Iowa State University (ISU), there is a lack of existing resources and organizations that can teach students about the Application Specific Integrated Circuit (ASIC) design process. Multiple courses exist in the ECpE curriculum that teach students about relevant ASIC topics, including logic design, physical layout, and embedded systems, but each course is separated in its own semester term. There is a severe lack of integration between courses, which is a detriment to the students focused on learning about the full ASIC design process. Students currently do not have the opportunity to iterate on a design between semesters, including architecture design, implementation, fabrication, and validation. With the establishment of the student led organization, ChipForge, students will be able to fully realize the ASIC design process over multiple semesters in a vertically integrated environment.

ChipForge began as an idea supported through multiple iterations of senior design projects within the ISU ECpE curriculum. Senior Design is taken between 2 semesters, for a 2 term design cycle [Iowa State University ECpE Senior Design](#). Within the first semester, students are expected to capture the requirements of project and construct an adequate design which meets all customer requirements. In the second semester, students must implement their design using provided tools to meet all design requirements. As of now, led by ChipForge's faculty advisor, Dr. Henry Duwe, 4 digital ASIC design senior projects have been completed, as well as 1 analog ASIC design project. Currently, there are 2 active digital senior design projects and 1 active analog senior design project. Additionally, a separate group from the First-Year Honors Mentor Program (FHMP) was created for one semester to provide a learning experience for a different user group, which had little to no relevant course experience prior to the semester research. The FHMP group

was tasked with helping Graduate students perform validation (bring-up) on the first received digital ASIC design from the first senior design group.

For both existing groups, including senior design teams and FHMP, it was determined that documentation was required for different reasons. Senior Design teams, which included experienced undergraduate students, who have completed all required coursework, needed information on how to properly set up and use the tools in a consistent manner. Multiple senior design teams struggled with setting up local environments for changing open source tools, which heavily delayed the implementation of senior design projects in the second semester of the course. Additionally, the provided information on how to run the open source tools was sparse, which made it difficult to run verification, physical layout, or design submission checks in a timely manner. The FHMP group had additional challenges alongside learning how to run the tools, which included learning how to code in C and Verilog, the concepts of digital and embedded system design, and how to work through system level architecture designs. Each of these topics are covered in ISU courses, which the FHMP members had not taken yet, which shows a need for relevant design context.

By establishing an official club, ChipForge is required to cater to the varying experience levels of undergraduate students, who are part of different majors and multiple technical levels. Additionally, ChipForge is using tools that are not natively supported by the ISU ECpE department, and are challenging to use, as required by ChipForge's ASIC design solution.

This creative component presents the foundational documentation for ChipForge. This documentation includes an introduction for club members, curated tutorials for each design stage, and steps to run all required open source and internal tools rooted in evidence-based documentation design. This documentation has been through three rounds of evaluation with final results indicating effective documentation that has contributed to ChipForge's first tapeout design in their first semester of the co-curricular's establishment.

CHAPTER 2. RELATED WORK

ChipForge was created under the idea that a co-curricular ASIC design club could provide a unique and effective alternative for undergraduate engineering students to learn about ASIC design in a vertically integrated environment. Multiple research papers have demonstrated positive impacts for co-curricular and student led organizations, as well as vertically integrated learning within university curriculum's. By combining the positive effects of both of these mediums, ChipForge can achieve a new model to teach students about ASIC design at Iowa State University. Additionally, there are many research topics that have studied the effectiveness of different documentation strategies and tutorial writing, which can serve as a guideline for how to implement documentation for ChipForge.

2.1 Co-curricular Impacts

There are multiple opportunities offered to students through university co-curriculars, that are additional to the required university coursework of engineering students. In a study [Powell et al. \(2015\)](#) between 300 clubs across 13 colleges of engineering, multiple core competencies were shared across these co-curriculars, including hands on experience for relevant technical projects, future career opportunities, and the opportunity to participate in student driven leadership teams. Another work [Olewnik et al. \(2015\)](#) shows that despite the large time commitment a co-curricular can include, students report that they are able to successfully connect their club material to their coursework. Additionally, students report receiving learning support from their peers, and a great opportunity to connect with partners and industry professionals. It is also important to consider the nature and goals of a club, which can heavily influence the motivation of students and the structure of a co-curricular. A study [Hinkle and Koretsky \(2019\)](#) interviewed students in three different clubs, which demonstrated different divisions of labor based on the

goals of the co-curricular and students. Depending on the goals and timeline of each club, the division of labor and organization of the club differed greatly. Co-curricular organizations can provide a strong learning environment and opportunity for growth for engineering students, and can vary greatly depending on the goals of the organization.

2.2 Vertically-Integrated Curriculum

Within the context of a university co-curricular, a vertically integrated club will combine multiple technical aspects from varying courses across multiple years of experience levels from undergraduate students. A target experience for a ChipForge member is to participate in an ASIC design from their Freshman to Senior undergraduate years in their curriculum, tying in and adding to relevant experiences from their coursework as they become more experienced. Multiple works outline the positive benefits of vertically integrated programs in traditional university coursework. At the University of Arkon, the Chemical Engineering curriculum integrated a design project where all years of undergraduates must enroll, scaling from freshman to senior level students [Spickard-Prettyman et al. \(2004\)](#). Purdue University has implemented similar research based vertically integrated coursework into their curriculum, and received positive feedback from students as well [Ramirez and Zoltowski \(2022\)](#). Vertically integrated engineering projects have the potential to expose students to a wider range of experiences, that can not be achieved in a normal course curriculum. With ChipForge, students will have the opportunity of applying multiple semesters of courses towards ASIC design, including logic design, architecture design, embedded systems, and physical design layout.

2.3 ASIC Design Process

ChipForge will follow through the full ASIC design process, starting from an initial design, through ASIC fabrication, and finishing with post silicon validation, or bring-up. [Edwards et al. \(2021\)](#) describes the process of open source ASIC design, which mirrors similar steps to commercial ASIC design, except with the use of free, open-source tools. An initial design would

begin by defining project requirements and defining the system architecture, without the implementation of any cells or blocks. Next, Hardware Description Language (HDL) modules would be created for digital designs, and analog circuits would be defined for analog blocks. Both HDL source files and analog schematics can be simulated with different tools, to verify expected functional behavior before synthesis or layout. Next, digital designs will synthesize source HDL code to create a gate level netlist of standard cells, which are used by a foundry to fabricate an ASIC. The analog cells will manually be laid out, and utilize netlists from the verified schematics. Both digital and analog cells will be placed, either through the use of automatic EDA tools, or done manually in the case of some analog designs. Next, after all gate level or analog cells are placed down from the corresponding netlists, routing will occur which connects the wires between each gate and cell. Finally, an output GDSII file is created, which defines each layer of an ASIC design to describe how a target project should be fabricated at a foundry. Design Rule Check (DRC) and Layout Versus Schematic (LVS) checks can also be done to verify a design was synthesized, placed, and routed properly.

2.4 Documentation

Documentation can vary highly, depending on the style, technical aspects, and needs of its users. Multiple works have been studied, including surveys from developers in academia, industry, and online, to determine general best practice techniques for documentation, what processes should be supported, and what pitfalls to avoid.

Understanding what aspects of documentation are most valuable to engineers and developers is critical to providing effective work for ChipForge. [Forward and Lethbridge \(2002\)](#) determined that the most effective attributes of documentation include relevant content, up-to-date information, availability, examples, and organization of sections. Additionally, another key point highlighted was to focus on creating documentation over maintaining it, assuming the documentation tools should enable designers to easily create and publish more documentation as needed in the future. [Plösch et al. \(2014\)](#) states that in another developer survey, the most

important documentation attributes include accuracy, clarity, readability, structuredness, and understandability. Each of these aspects can vary depending on the relevant technical topics and user groups, which must be considered with ChipForge. [Aghajani et al. \(2019\)](#) scraped artifacts from multiple online sources, including mailing lists, code repositories, and online forum posts. The most discussed topics centered around software documentation included correctness, completeness, and up-to-dateness.

Multiple works have been created which review and analyze the process of writing documentation, which is critical to ensure that documentation will be continued on after its initial writing. [Aghajani et al. \(2020\)](#) discovered that issues related to the process of documentation were a lower priority than issues related to documentation content. However, the primary issue with documentation processes stemmed from not having enough time in development to create documentation. Additionally, other concerns with documentations include lack of organization or automation in producing documentation. [Visconti and Cook \(2002\)](#) developed a documentation maturity model, and found through surveying multiple industry projects that many documentation efforts were lacking, including limited recognition in value of documentation or a defined process for documentation. [Dagenais and Robillard \(2010\)](#) shows that reference documentation for how to use tools was the easiest type to create, but took more time to maintain. Additionally, getting started documentation for first time users is a very beneficial way to train new designers in a project or organization.

[Uddin and Robillard \(2015\)](#) cites the most referenced challenges with documentation for both content and presentation categories. It was determined that the most frequent software documentation issues related to content included incompleteness, ambiguity, and unexplained examples. In terms of presentation, frequent problems arose with documentation relating to excessive amounts of documentation or fragmented information across multiple pages.

2.5 Tutorials

A critical component to ChipForge documentation will be tutorials, which are text based documents that provide a guideline for new members to use ASIC design tools. [Head et al. \(2020\)](#) found that key aspects to programming tutorials includes keeping examples consistent, starting with a reference implementation, propagating tutorial changes, and testing tutorials after writing. [Kim and Ko \(2017\)](#) determined through the analysis of multiple existing tutorials that several trends existed, showing that most tutorials teach from the ground up, require learners to write programs, and provide feedback for actions completed. Additionally, some negative trends were common, including missing guidance for common errors, no personalization on prior experience, or explanation of why a concept is useful. [Sillito et al. \(2012\)](#) describes attributes of recognized code examples on online forums, including concise code, providing step-by-step solutions, and highlighting critical elements of needed questions. Each of these aspects of tutorials are critical to consider within the context of ASIC design and ChipForge, which will need to cater to a variety of student backgrounds and experience levels.

CHAPTER 3. MOTIVATION

As a student driven co-curricular, ChipForge can provide early technical development of skills, mentorship, and end to end ASIC design experiences that are not available in the existing curriculum. Additionally, each relevant course taught for Electrical and Computer Engineering students at ISU teaches only one part of the design cycle at a time in a semester, such as designing embedded software for validation or logic design to build a digital ASIC. There are also sections of the design cycle which are lacking from the ISU curriculum, such as ASIC validation or digital verification, that can be taught in ChipForge with proper documentation.

The chosen solution (eFabless) to design and fabricate ASICs means that ChipForge is required to use unfamiliar open source tools, which are not taught in any coursework. This will lead to constant work to maintain, due to the rapidly changing nature of the open source environment. An effective system is required to enable documentation that can be easily updated and added upon as tools update.

3.1 Lack of Existing Experience

One main reason for the growing interest in an ASIC design club is due to the lack of any existing clubs which focus on any aspect of VLSI design within Iowa State University. There are multiple existing clubs [Iowa State University Student Organizations](#) which focus on discrete level Printed Circuit Board or embedded systems design and validation, but none centered around the topic of ASIC design. This has posed a challenge for students who are studying VLSI design at Iowa State, with the goal of receiving internships before their official graduation, as many students use co-curricular organizations to build technical skills and a diverse alumni network. With the lack of any other existing ASIC organizations at ISU, there is also a lack of resources to utilize and provide to club members to assist in the ASIC design process. This identifies the need

for a solid form of documentation, that can be actively maintained within ChipForge by its own members, built from the ground up.

3.2 Vertically Integrated

A critical reason that ASIC design cannot be fully realized within the existing ISU curriculum is due to the limitation of time in a per-semester course, which is around 4 months long. While there are multiple courses that teach students critical concepts about digital logic design, physical design, analog design, and embedded validation, these courses are loosely connected and do not tie together a full project. By utilizing the skills within each independent course, students can apply their knowledge within ChipForge to realize a vertically integrated ASIC design process across a 2 year period. This also provides students with the opportunity of an iterative design process which can organically grow and change over time, which would provide an experience closer to industry work. Since this design process requires filling in gaps between multiple classes, documentation will be required to provide relevant context for students with multiple levels of experience and backgrounds, as well as to "bridge the gap" between the Iowa State University curriculum.

3.3 MPW Shuttle Comparison

There are multiple different companies which offer shared Multi Project Wafer (MPW) submissions, in which an ASIC design can be submitted as part of a shared mask. The main benefit of this is the reduced cost, which means universities, research programs, and startup companies can submit ASIC designs at an improve amortized rate. Table 3.1 outlines the comparison between multiple MPW submission. The chosen MPW shuttle program of ChipForge is ChipIgnite, which is a paid version of the previously sponsored shuttle program OpenMPW, which was sponsored by Google eFabless. OpenMPW and ChipIgnite offer the cheapest price per mm^2 of any researched MPW shuttles. Additionally, eFabless provides multiple design repositories and submission requirements to help users create and submit an ASIC design. Tiny Tapeout [Tiny Tapeout](#) is another company which uses the ChipIgnite shuttle to provide a more

accessible design cycle, which is primarily targeted towards high school and university programs. Tiny Tapeout includes multiple projects on one ASIC design, which is apart of the shared MPW shuttle, to reduce the cost of low area designs. The large drawback to this is that the price per mm^2 is much larger than a full ChipIgnite submission. On the other hand, the same set of tools and processes as eFabless can be used. Another company which offers shared MPW submission is Muse Semiconductor [Muse Semiconductor](#), which provides competitive rates to the ChipIgnite program for their 180nm process, with a minimum area of $5mm^2$, which is half the delivered size of both OpenMPW and ChipIgnite. Compared to ChipIgnite, the 180nm Muse Semiconductor shuttle delivers less packaged chips, and does not include a suite of supported design repositories or tutorials, like eFabless does. While the same open source tools could be used, without the existing design repositories from eFabless, it would be much more difficult to design and verify an ASIC.

Table 3.1 MPW Shuttle Services

Shuttle	Price (\$/mm ²)	Area (mm ²)	Quantity	Node (nm)	Foundry
OpenMPW	0	10.8	20	130	SkyWater
ChipIgnite	926	10.8	100	130	SkyWater
180 MS RF G	1,250	5	40	130	TSMC
Tiny Tapeout (L)	4,167	1.2	5	130	SkyWater
Tiny Tapeout (M)	5,000	0.4	3	130	SkyWater
65 MS RF GP	5,800	1	100	65	TSMC
Tiny Tapeout (S)	6,250	0.08	1	130	SkyWater

There are multiple existing works which have used the open source SkyWater 130nm PDK and open source tools supported by eFabless, in the same process [Herman et al. \(2023\)](#); [Chen et al. \(2023\)](#); [Cirimelli-Low et al. \(2023\)](#); [Oliveira and Brito-Filho \(2022\)](#). Unfortunately, these works do not provide any additional documentation or resources to assist ChipForge students, demonstrating a need for more resources to support the co-curricular.

3.4 Active Tools and Processes

ChipForge has committed to using eFabless and its supported ASIC design and verification tools [OpenROAD](#); [GTKWave](#); [Magic VLSI](#), which means the co-curricular is susceptible to actively changing tools and processes that are required to be updated for submission approval. The precheck tool is used to verify that an open source ASIC design using OpenROAD, the packaged synthesis and physical layout tool, is compatible with the SkyWater 130nm open Product Development Kit, as well as other design constraints. When these tools change, a new tag is released for the design repositories by eFabless, which will require a solution of active documentation to maintain and communicate to members within ChipForge when a new release is made. Additionally, the functionality of the tools and processes may change, meaning more documentation may need to be changed or updated internally, to meet ChipForge's own deadlines and design cycle.

CHAPTER 4. MOTIVATION IN CONTEXT

Since the required documentation for the ASIC design process is targeted towards a student led co-curricular at Iowa State University, it is important to consider the specific context of the curriculum and student groups that will be involved. Since undergraduate level students will not have all taken the same set of courses at the same time, it is important to recognize the topics of each core class, and how it can impact the students learning. Additionally, the context of different existing groups can be explained, with different senior design groups or a 1 semester Honors project, which have varying levels of experience.

It is also important to understand the processes and open source tools provided by eFables, which are required to be used to generate and submit an ASIC design. Additionally, the internal tools solely for ChipForge must be understood to properly document their purpose and functionality [Ling \(2024\)](#). Both sets of tools will provide the framework for a vertically integrated ASIC design process, which includes ASIC design, fabrication, and bring-up.

4.1 ISU Coursework

The most relevant courses for students interested in being apart of the vertically integrated co-curricular ChipForge at Iowa State University is CPRE 2810, 2880, and 3810 [Iowa State University Courses and Programs](#). These classes provide the foundation for digital logic and embedded systems design, which is imperative for iterating on an ASIC design process and being involved in the bring-up process. For more experienced members, there are multiple additional senior (CPRE 4000+) courses that can be taken to provide more experience in a system architect role in the design process.

For greater focus in the VLSI design process, for both digital and analog flows, EE 3300, 4350, and 4650 can be taken. Two of these courses, EE 3300 and 4350 provide a background in analog

design, which is possible for a subset of the eFabless MPW shuttle submissions. EE 4650 provides additional teaching material on other aspects of digital ASIC design, with a larger emphasis on physical design and timing constraints, versus the CPRE courses functional logic design.

By utilizing the publicly available coursework materials, and by understanding the overall structure of the ECpE programs, we can better reference contextual information within the documentation targeted towards students apart of ChipForge.

4.1.1 CPRE 2810

CPRE 2810, Digital Logic, is the first digital design class that students will take in the ECpE curriculum at Iowa State University. This course is required for both Computer and Electrical Engineering students, with a prerequisite requirement of Sophomore classification. CPRE 2810 teaches about basic boolean arithmetic, combinational and sequential circuit design, and finite state machines. Additionally, students have a lab component where they integrate their digital designs on an Altera DE2-115 FPGA Development board, to validate their designs in real time after simulation. This class provides an opportunity for all students to start at "Level 0", assuming no background knowledge is required before the course begins. As learned from previous groups, it was discovered to be extremely beneficial for potential ChipForge members to come in with at least the foundational knowledge. As a supplement, ChipForge documentation can include references to this course to use as a crutch to better guide students in the digital ASIC design process.

4.1.2 CPRE 2880

Embedded Systems 1, or CPRE 2880, is the first embedded systems class in the ECpE department. Like CPRE 2810, this course is required for both Computer and Electrical Engineering students. Students are required to take either CPRE 2810 first, or an Electrical Engineering coding course which teaches best practices in C. Within the course, students will design embedded C code for a TM4C123 micro controller, which is attached to a Roomba to

simulate a robot mission on Mars. This course teaches students foundational practices for real-time embedded systems design and debugging practices. All of the skills learned in this course can be translated to verifying and validating an ASIC design by writing software for the open source RISC-V processor which is apart of the eFabless Caravel Core, labeled the Management Core. As this class is expected to be taken by Sophomore level students, it is again encouraged that ChipForge members have taken the course before entering the club, as it is essential for understanding the motivation of design and bring-up concepts. With this in mind, documentation can reference the taught information from CPRE 2880 and build off what is already given for students to build off of.

4.1.3 CPRE 3810

CPRE 3810, Computer Organization and Assembly Level Programming, is intended to be taken after CPRE 2880 and is meant for Junior or Senior level engineering students. This course is only required for Computer Engineering students, and not Electrical Engineering students. Students will learn about assembly programming with the MIPS Instruction Set Architecture, processor design, and cached memory systems. In the lab portion of the course, students will implement a MIPS processor which can run 30+ different MIPS instructions. These instructions can vary between arithmetic, instruction branching, and memory interface operations. The greatest benefits a student can have after this course is more experience coding in an HDL language to implement a digital design, as well as working through more complex system level problems, which could pose a challenge for a larger system wide ASIC design within ChipForge. It is expected that more experienced members who are leading projects within ChipForge have taken this course. In terms of documentation, the context of system level designs and challenges should be considered to reflect the level of complexity students are comfortable with.

4.1.4 CPRE 4000+

There are multiple senior level Computer Engineering courses at Iowa State University which can help students learn more about embedded systems, digital design, and system level challenges, which build on the skills of all the aforementioned courses. Such courses include CPRE 4800, 4870, and 4880 [Iowa State University Courses and Programs](#). These courses include a mixture of software and hardware design with micro controllers and FPGAs in a more complex system than any previous class. For example, in CPRE 4800, Graphics Processing and Architecture, students are placed in 4 person groups and expected to create an FPGA design for a graphics processor. Similar to CPRE 3810, these courses can be used to build on system level design skills for team leaders within ChipForge. For documentation purposes, these courses are important to recognize, since they represent the upper limit of what an undergraduate student is capable of learning through the existing Iowa State University curriculum.

4.1.5 EE 4650

EE 4650, Digital VLSI Design, is an Electrical Engineering course that focuses on other aspects of digital design than the Computer Engineering counterparts mentioned beforehand. In this course, students focus more on the comparisons between power, area, and delay, which are extremely beneficial for the physical layout design within the ASIC design process. Since this is a minimal focus in the other CPRE courses, it may be beneficial for students with a focus in VLSI design to take this course, to experience a breadth of information and learn more about another section of the design process, which is fairly limited in coverage in other courses. Understanding that the digital physical design is limited is important in the context of documentation, since more work will need to be done to explain the basics of physical design, its constraints, and how to work through the process specifically with eFabless's supported tools.

4.1.6 EE 3300/4350

Both EE 3300 and 4350 focus on Analog VLSI design, which is another optional design path that has been explored in past senior design teams. ChipForge students interested in Analog Design should take these courses to gain a better understanding of how to design and verify different analog cells. It is important to recognize that like the digital design courses, the analog design courses taught at Iowa State University use a different set of tools, meaning documentation should focus on explaining the basics of how to set up and interface with the open source tools provided by eFabless.

4.2 eFabless

eFabless [eFabless](#) is the current chosen solution of ChipForge to participate in Multi Project Wafer (MPW) Shuttle submissions, where students can design, fabricate, and validate open source ASICs with an acceptable turn around time. Within the following sections, a description of the different eFabless MPW shuttle descriptions, provided tools, and provided documentation will be outlined. This will drive a large portion of the ChipForge documentation, which is dependent on the tools and processes that are required by eFabless.

4.2.1 OpenMPW

The OpenMPW program through eFabless conducted 8 individual shuttle programs at varying submission dates, which took on 40 different ASIC design projects [eFabless](#). Each shuttle was sponsored by Google, and required each submission to be open source and available to the public. Each design was required to use the Caravel framework, which was a wrapper design in which both user digital and analog ASIC could be inserted within. Within the Caravel framework is an open source RISC-V processor called the Management Core, which is used to interface both to external I/O pins independent of the user design, alongside interfacing directly with the custom user area design between 128 logic analyzer pins and a Wishbone bus interface.

The OpenMPW solution was beneficial for multiple senior design projects at Iowa State University since it required no monetary support for the customer, which led to more possible groups to undergo the process. 5 different senior design teams, 4 digital and 1 analog, went through the design process assuming their project could pass the Caravel Precheck. This precheck was used to ensure that a unique design was created, constraints were not violated, and that the design would be able to be successfully fabricated. This acted as a quality gate for a free, sponsored event.

The first digital senior design project, sddec22-17, was able to successfully create and submit an approved design, which was a SHA-1 hashing IP block. More detail on this design will be explained later, but it is important to state that ChipForge has received the fabricated ASIC design within a QFN-50 packaged ASIC. Additionally, ChipForge received a Caravel Breakout Board, which places the fabricated and packaged QFN-50 ASIC onto a compatible M.2 card, and an updated revision of the Caravel Development Board, as seen in Figure 4.1. The Caravel Development board includes voltage regulators, a USB connector to program the Management Core, and 38 GPIO pins to interface between either the Management Core or user area design. The schematic is open source and capable to be modified, and due to the delivered ASIC being its own board, ChipForge can also manufacture its own custom hardware to interface with the PCB via the M.2 connector.

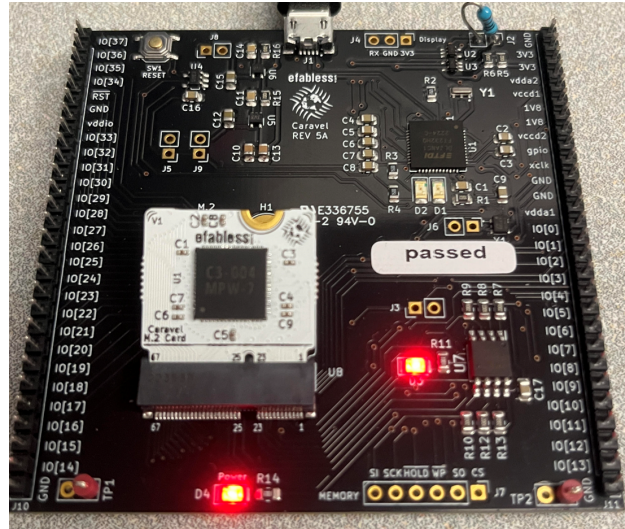


Figure 4.1 Caravel Development Board (Black) and Caravel Breakout Board (White)

4.2.2 ChipIgnite

ChipIgnite is the paid alternative to OpenMPW, which has an accelerated turn around time and additional support from eFabless [eFabless](#). Each MPW Submission utilizes the same set of open source tools and submission requirements for precheck, and costs \$10,000 per MPW order. Within each order, 100 packaged QFN-50 chips are delivered along with 5 Caravel Development Boards to validate a ChipForge design. The IO interfaces remain the same as the OpenMPW shuttle as well. Google has ended support for the OpenMPW shuttle, meaning it is the job of ChipForge to secure funding to continue creating chips. Funding has been secured to order 6 different ASIC designs through the National Science Foundation and the ChipIgnite program, which will be intended to be used over a 3 year period. By utilizing the experiences of senior design teams through the OpenMPW shuttle, the same tools and processes can be documented and utilized with the ChipIgnite program.

4.2.3 Provided Open Source Tools

All required open source tools and requirements are provided by eFabless, which are used for both their OpenMPW shuttle and ChipIgnite submissions. Some tools are shared between the digital and analog design process, but it is important to recognize which ones are not, to ensure each tool and flow is referenced step by step in documentation. Additionally, a submission precheck is provided by eFabless to ensure that a design can be successfully submitted and fabricated.

The digital design process combines multiple existing tools into the Caravel harness, which is a wrapper that a digital design can be inserted into. Within the caravel harness is the Management Core, which can be used to interface a digital or analog design via 128 Logic Analyzer pins, or 38 GPIO pins, which are connected externally to the Management Core. Source HDL modules are written in Verilog, and can be simulated with the open source tool Icarus Verilog [Icarus Verilog](#). Simulating waveforms are crucial, to ensure a digital designs performs the expected functionality for modules and a system level design. There are fixed commands within the caravel repository to run waveform simulations, which are required in documentation. To verify simulated waveforms, the waveform viewer GTKWave [GTKWave](#) is used by eFabless. This is not bound by the open source tools, and additional waveform viewers could be used. OpenROAD [OpenROAD](#) is used for end to end ASIC generation, including synthesis, place and route, and timing analysis. This is advertised as a one stop solution to save time for designers within the caravel project. This process is called hardening by eFabless, and it configured with a yaml source file for each hardened module. Design Rule Checks and Layout Versus Schematic checks are set in place to verify a generated ASIC design will meet fabrication requirements and not fail. Magic [Magic VLSI](#) can optionally be used to create custom digital cells to be placed in the user area design of an ASIC, by creating custom routing for a logic cell or module. Finally, KLayout [KLayout](#) is used to view the generated GDSII files from the OpenROAD output, which shows the final placement and routing of the target design module or full ASIC. Like GTKWave, KLayout is not required to be used in the open source flow, and other open source layout viewers

could be chosen. Specific commands to run waveform simulation and hardening must be included in documentation for both experienced and inexperienced members within ChipForge. Alongside this, information on how to create a configuration file for OpenROAD, where to place files within the project directory, and how to create a testbench must also be included.

The analog design process with eFabless uses multiple other open source tools to complete analog designs within the Caravel user wrapper. Schematics for cells, to be placed in the user area, are designed using XSchem [Xschem](#). These schematics can be created hierarchically, similar to the digital design flow with Verilog source files. Specific libraries within the SkyWater 130nm open source PDK are required to create schematic cells, which will need to be properly documented. To simulate schematics, the analog waveform simulator Gaw [Gaw](#) is used, which requires additional configuration to setup within XSchem. The layout tool Magic [Magic VLSI](#), which can be used optionally for the digital process, is required for the analog design flow to place and layout all designed analog cells. Magic has a very unique set of commands and controls, which would be useful to include in documentation. The tool Netgen [Netgen 1.5](#) is used to perform Layout Versus Schematic checks of the generated spice files from XSchem versus the generated netlists post layout from Magic. Each of these tools requires relevant commands to start and source the tools, as well as information to navigate the tools. This information would be relevant for any member in ChipForge, since it is not taught in the ISU curriculum for any of these specific tools.

For both the digital and analog design processes, design submission is determined with the Caravel Precheck. This tool will be run within the caravel design repository and evaluate if the design has been changed from the sample, contains no DRC or LVS violations, and ensures a generated top level ASIC design exists.

4.2.4 Provided Documentation

Multiple sources of online documentation already are published and maintained, which can be used as a basis for the foundation of ChipForge's documentation. This documentation is

particularly beneficial in outlining how to run the specific open source tools and architecture required by eFabless, but is lacking in any contextual information that would be useful for students within the ISU ECpE curriculum.

eFabless hosts multiple websites with repositories linked to documentation. One specific page outlines the Caravel Harness [Caravel Harness](#), which overviews the total frame of the ASIC design, including the digital/analog user design area, the open source RISC-V Management Core, and the IO interface between both modules. This page is useful in outlining what types of pin interfaces the designers have access to, an explanation for project directory structures, and how to get started with building an initial design. Unfortunately, this page is not well maintained, and contains multiple broken hyperlinks to other pages, as well as some out of date information relating to creating configuration files for hardening and which versions of tools to use.

Another documentation page that eFabless published is the Caravel Management SoC page [Caravel Management SoC - Litex](#). This page overviews details specifically within the Management Core IP block, outlining a more detailed overview of the system architecture within the open source RISC-V core. This is crucial information that is relevant for any ChipForge member or ASIC design team that is using the Management Core to validate an ASIC design. Within the Management Core, multiple memory mapped register blocks can be modified to configure UART, SPI, GPIO, and a TIMER module. The Management SoC page provides memory addresses for multiple registers within these blocks, and in some cases descriptions of how to configure the modules properly. But, this is not true for all cases, in which it will require more effort from ASIC designers to determine functionality.

The final main documentation page published by eFabless is the Caravel User Project page [Caravel User Project](#). Caravel User Project is the main design repository that digital design teams will work inside of to develop a digital ASIC design to be submitted to eFabless, either for an OpenMPW or ChipIgnite order. This page outlines how to integrate Verilog source designs, simulation, hardening, timing analysis, and running the precheck analysis tool. This is the most robust and beneficial page for digital designers to walk through, and is more up to date than the

previously referenced Caravel Harness documentation page published by eFabless. Notably, there is not a matching documentation page for the twin analog design repository, Caravel User Project Analog, in which an analog user design project is inserted into the Caravel framework.

Another important documentation page to note is the OpenROAD documentation [OpenROAD documentation](#). As there are a limited number of classes within the ISU curriculum that focus on digital physical design, this information is critical for any senior design team. There are multiple useful examples of how to configure the hardening files for synthesis, place and route, and timing analysis when designs utilize OpenROAD within the Caravel User Project design repository. This page does not outline how to run the specific commands within the eFabless framework, however. Relevant information and pages from OpenROAD could be referenced in ChipForge documentation, cutting out what only is relevant for the digital flow within eFabless.

ASIC designs are submitted to the SkyWater foundry to be fabricated, with the open source 130nm Product Development Kit. Google has released an alpha build for documentation outlining the 130nm PDK library [SkyWater SKY130 PDK](#). There is some information on Design Rule Checks and lists of cells that are included in each of the 5 provided libraries in the PDK. However, multiple pages are marked as in progress or empty, and there are 2 warnings on the front page that the documentation is still a work in progress. The documentation can be built locally from the git repo for a more up to date copy, as done by a previous senior design team, but information is still sparse.

4.2.5 Provided Tutorials and Webinars

Alongside eFabless providing multiple pages of documentation, they also have created multiple video based tutorials and webinars, which are on their YouTube channel [Efabless YouTube Channel](#). Many webinars are between 30 to 90 minutes long, and cover topics such as circuit simulation, layout design, and how to setup an introductory project. There are multiple different web series which include introductory videos, as well as more in depth topics for advanced users. The generalized technical webinars are most beneficial to teach best practice

design techniques for ASIC design processes. However, if the introduction and tutorial videos are not updated and re-uploaded, it is likely that they will fall out of date and not be relevant for any user. It is important to consider the medium of documentation and how easy it is to maintain, which is more difficult in a video, since it requires the entire contents to be recorded again.

4.3 Student Groups

There are multiple existing groups of students who have worked through ASIC design projects with eFabless's provided tools, prior to the creation of ChipForge. Each group provides a different experience level and need for information within the context of their project, which is important to recognize to determine how to frame ChipForge's documentation goals and structure.

4.3.1 Senior Design Teams

Multiple senior design teams [Iowa State University ECpE Senior Design](#) have completed their 2 semester coursework for both the digital and analog flow. 4 Senior design teams have completed the digital ASIC design process using eFabless, while 1 team has completed an analog design. Additionally, there is both a digital and analog senior design team in their second semester currently. Each team has been required to use the tools from eFabless, since each design was to be apart of either the OpenMPW sponsored shuttle or as a piece of the ChipIgnite program that will be ordered in the future. As part of some of the senior design teams deliverables, documentation on how to use the tools and perform validation, assuming the designed ASIC could be ordered, was required by the senior design advisor and customer, Dr. Henry Duwe. Documentation was delivered in a PDF format, and created using a Microsoft Word document. There were no requirements in formatting or standard for any of the documentation or deliverables received, which could vary between teams. Through many senior design processes, the teams identified a slow start which was due to setting up a development environment and learning how to use the open source tools and commands. Through the use of proper documentation, this can be standardized for both the digital and analog flows.

In Spring 2023, ChipForge acquired the first digital senior design team's (sddec22-17) fabricated ASIC design, which included a SHA-1 hashing IP block, that was intended to be used as part of a bitcoin mining system. The received dies are represented in Figure 4.2. As referenced before, eFabless also sent 1 Caravel development board and 10 breakout boards which contained a M.2 connector with the packaged QFN-50 ASICs soldered on. The successful design and fabrication of the first senior design teams ASIC opens up more opportunity for ChipForge, in the form of post silicon validation.

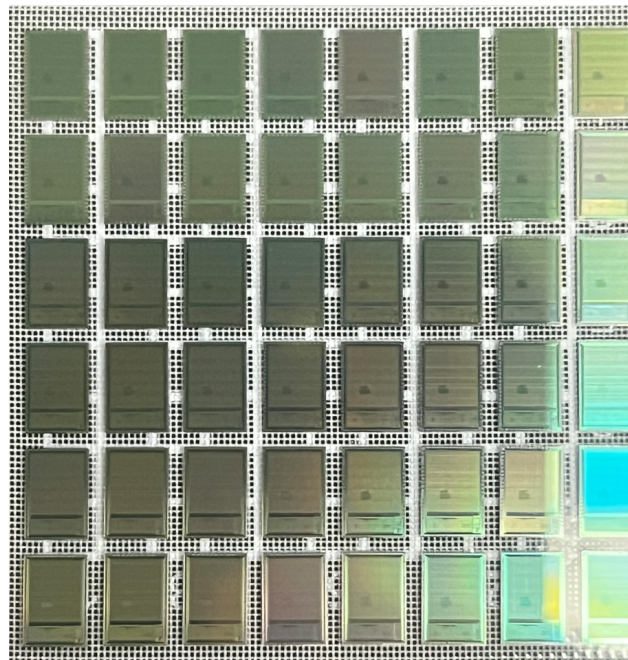


Figure 4.2 sddec22-17 OpenMPW Fabricated Chips

4.3.2 First-year Honors Mentor Program

The First-Year Honors Mentor Program (FHMP) provided Freshman undergraduate students with the opportunity to be involved in research with professors and graduate students at Iowa State University within the College of Engineering. ChipForge was able to determine critical information that would be required to bootstrap inexperienced students who have never learned or worked on any stage of ASIC design before. During the Spring 2023 semester, 5 students

assisted ChipForge in the bring-up process of the first received SHA-1 hashing ASIC from the first digital senior design team. This provided students with the opportunity to learn more about digital design and embedded systems validation before their classes. But, this also posed a challenge as it was evident that more background information would need to be required on documentation, to help bootstrap more inexperienced members within the club, who are not senior level undergraduate students. This will require more example code, references to ISU coursework, and explanations of designs than the senior design teams within the documentation.

4.4 ChipForge

In Summer 2024, ChipForge became recognized as an official Registered Student Organization within Iowa State University. As follows, members have begun to be officially added to ChipForge, and will be expecting guidance on processes, team organization, and determined roles within the co-curricular. The following sections will outline different aspects of the clubs management, which can be embedded into documentation to assist onboarding prospective members.

4.4.1 Club Structure

The general structure of the club will follow at minimum what is required by Iowa State University [ISU Starting a New Student Organization](#). At minimum, it is required that a club president and treasurer are appointed to a cabinet and involved each semester. Additionally, the club must maintain at least 5 active members, which can include leadership and a single club advisor. Students can be added on Iowa State University's student organization database, to receive emails for updates from ChipForge leadership. Currently, there are no requirements for dues, and no rules to determine who an active member is. Meetings are expected to be held weekly, and treated as long form workdays in which students can work on their respective task in the ASIC design cycle, which will be covered shortly.

Some goals of ChipForge include maintaining student involvement, fostering growth, and developing the strength of the ASIC design presence within Iowa State University. With the

identified lack and interest of ASIC design targeted primarily towards undergraduates, ChipForge seeks to foster this interest and provide opportunities for self driven students to learn with self driven projects. The projects and outcomes of ASIC designs will heavily depend on the personal interest and performance of each individual student, but ChipForge will aim to make any project as accessible as possible. This includes creating and supporting internal tools to accelerate the design process [Ling \(2024\)](#), as well as creating effective documentation to provide common information to any interested student. While there will be multiple stages of each ASIC design, the targeted documentation will provide sections to cater towards each stage within the ASIC design process.

4.4.2 ASIC Design Cycle

As stated previously, funding has been secured through a National Science Foundation grant for 6 independent ASIC designs over a 3 year period with the eFabless ChipIgnite solution. The exact amount of funding was targeted to ensure that ChipForge could meet the target design cycle, as depicted in [Figure 4.3](#). ChipForge has identified an initial design and validation cycle of 2 years, or 4 semesters, for a single chip. Each following semester, a new design cycle will begin, which creates a pipelined process.

Per semester, the following 4 project steps are conducted in order

1. High-Level Design: System level design and capturing requirements
2. Implementation: Implement digital/analog design and verification
3. Fabrication: Submit chip to eFabless for fabrication
4. Bring-up: Validate ASIC with RISC-V Management Core and internal tools

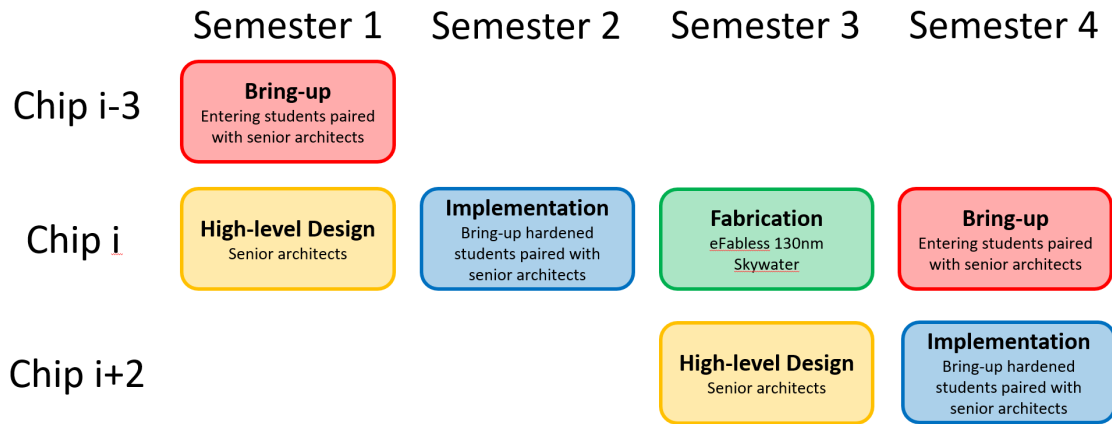


Figure 4.3 ChipForge Design Cycle

It is important to consider the previous context of the ISU curriculum and the students capabilities for each of the design phases. Based on the experience provided in the initial classes such as CPRE 2810 and 2880, younger members in ChipForge could contribute most effectively to the Implementation and Bring-up stages of the design. More experienced students that have taken more classes can contribute to the system level design within the High-level Design step, as well as providing assistance in the ASIC physical design [OpenROAD](#) and ensuring eFabless precheck is met. During the Fabrication of an ASIC, club members can assist other projects, which would be in the other stages of a design due to the pipelined nature of the project cycle. It is important to consider the context of the design cycle, and that the ChipForge documentation must cater to all aspects of the design cycle at once, since multiple members will be looking for different types of tools and accessible information.

4.5 ChipForge Tools

ChipForge has developed multiple internal tools to assist in streamlining the process for ASIC development within Iowa State University [Ling \(2024\)](#). These tools are intended to be used specifically for ChipForge members and interested parties within Iowa State University. Such tools include an actively managed toolchain which will setup an environment to use all of

eFabless's required open source tools, as well as custom Printed Circuit Board hardware to assist in validation and future project development. Additionally, a script to generate a FPGA project based on a target ASIC design within the Caravel framework has been created, to allow for more opportunity of ASIC verification before design submission and fabrication.

4.5.1 Toolchain

As identified from the senior design teams, one of the largest setbacks in the open source ASIC design process with eFabless was accessing and configuring the open source tools. Multiple teams created local environments on different machines and operating systems, which led to inconsistencies when sourcing and running the tools. As a solution, ChipForge has established a common remote server that students can SSH into and source all required tools with one command [Ling \(2024\)](#). Information will be required to be documented on how to SSH and access the remote server, where it is physically located on campus, and how to activate the environment for the supported tools. After the tools are sourced, the documentation can step through each following step as normal, assuming the same commands to simulate, generate, and submit ASIC designs as supported by eFabless.

4.5.2 Command Line Utility

A command line utility **chipforge** was created [Ling \(2024\)](#) to support a collection of common python tools used by ChipForge members. By creating a command line utility, ChipForge members will have the capability of working with a shared set of tools, which can be managed by a limited, more experienced set of designers. This ensures that each ChipForge members if using the same process, which can have one set of writing to outline installation and usage.

4.5.3 Custom Hardware

A need was identified for custom Printed Circuit Board (PCB) hardware within ChipForge to validate the first batch of OpenMPW ASICs from the first digital senior design team [Ling \(2024\)](#).

The desire was to expand the I/O capabilities of the provided eFabless development board in through the addition of Digilent PMOD ports [Digilent](#). This allows ChipForge to further validate software on the Caravel Management Core with additional hardware interfaces. An example of the ChipForge development board is shown in Figure 4.4. Documentation will be required to provide ChipForge members access to the repository or PCB designs as well as how to view the generated outputs for PCB orders or designated parts. Additionally, port numbers should be listed to describe the pin interfaces between the PMOD ports and the packaged ASIC within the eFabless Breakout Board (In White).

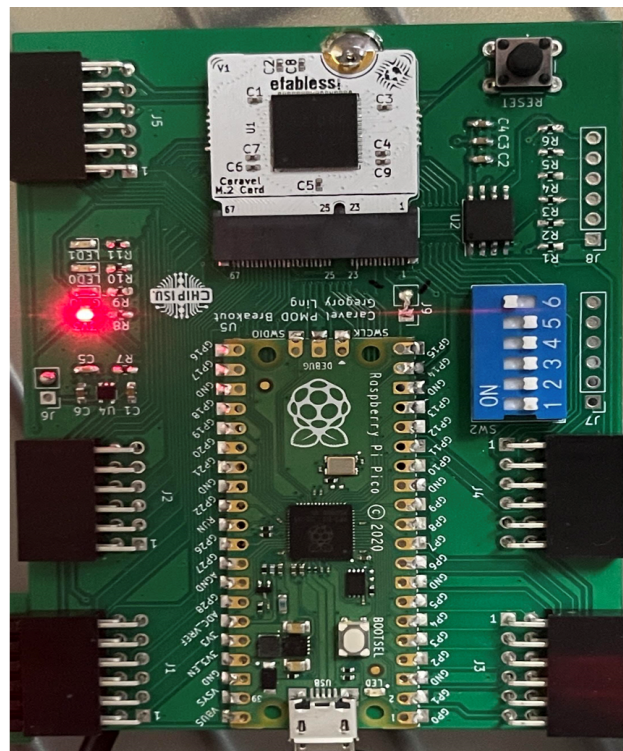


Figure 4.4 ChipForge Development Board

To pair with the designed ChipForge development board, two independent PMOD designs have been completed [Ling \(2024\)](#). The first design is referenced in Figure 4.5, which includes 2 chips of 512Kbit SRAM flash and 1 chip of 4 MBit NOR flash. These memory chips provide additional opportunities for ChipForge members to validate software on existing recieved chips, to

further test the open source RISC-V Management Core. The second designed PMOD is a seven segment display interface, which uses shift registers to store the values to display for each seven segment display. The seven segment displays may be daisy chained, by utilizing the shift register design. An example PMOD is shown in Figure 4.6. Like the memory PMOD, the intent of the seven segment display PMOD is to further validate the design of the Management Core, as well as providing another design to challenge ChipForge students, as we only have a single chip to validate currently.

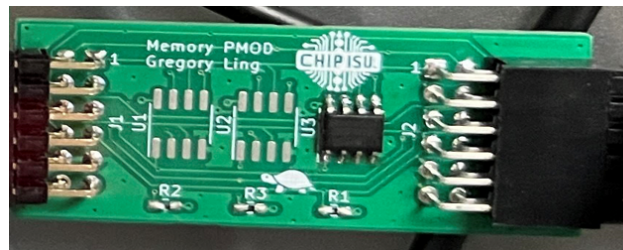


Figure 4.5 ChipForge Memory PMOD



Figure 4.6 ChipForge Seven Segment Display PMOD

For documentation purposes, it is important to signify where the source repository and information is for both PMOD designs, as well as their I/O interfaces and the functionality of each design. With the proper background context on how to modify and order the PCB design, as well as how to program each PMOD, ChipForge members will be capable of following through more sample tutorials.

4.5.4 FPGA Gen

ChipForge has created a script to create a Vivado FPGA project and synthesize a digital ASIC design which exists in the Caravel design repository [Ling \(2024\)](#). This is beneficial since it allows students to synthesize a design (albeit in a different function than in the OpenROAD ASIC flow) and verify software interactions with the target design. A sample image of the currently supported ChipForge FPGA, an Arty A7, is shown in Figure 4.7. In terms of documentation, it must be stated what commands need to be run to generate an FPGA project within the Caravel structure, and the context of what an FPGA is and how it is used within ChipForge.

Additionally, it would be helpful to identify which board is the FPGA for ChipForge members, who may be inexperienced with the different supported development boards in the co-curricular.

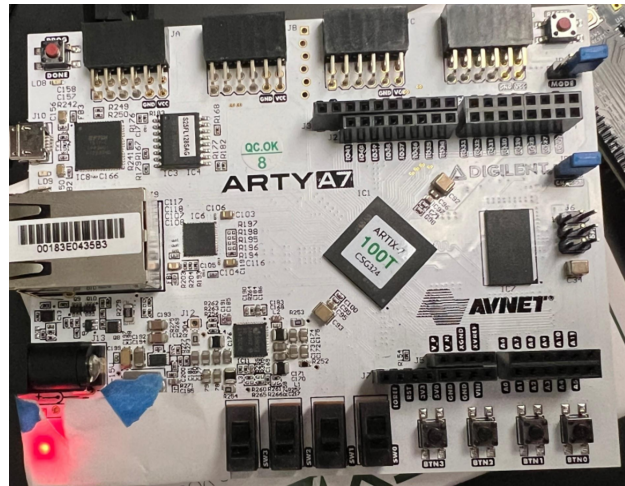


Figure 4.7 Internal FPGA Board

4.5.5 Camera Interface

To support off campus verification and validation of ASIC designs, a camera interface was created and setup within the ChipForge lab at Iowa State University [Ling \(2024\)](#). A camera was determined a necessity due to the visual nature of the seven segment PMOD PCB and the lights that are driven on all three of the supported development boards. An example of the camera

interface is shown in Figure 4.8. Documentation will require access information for this camera interface, including VPN information, a website address, and login information.

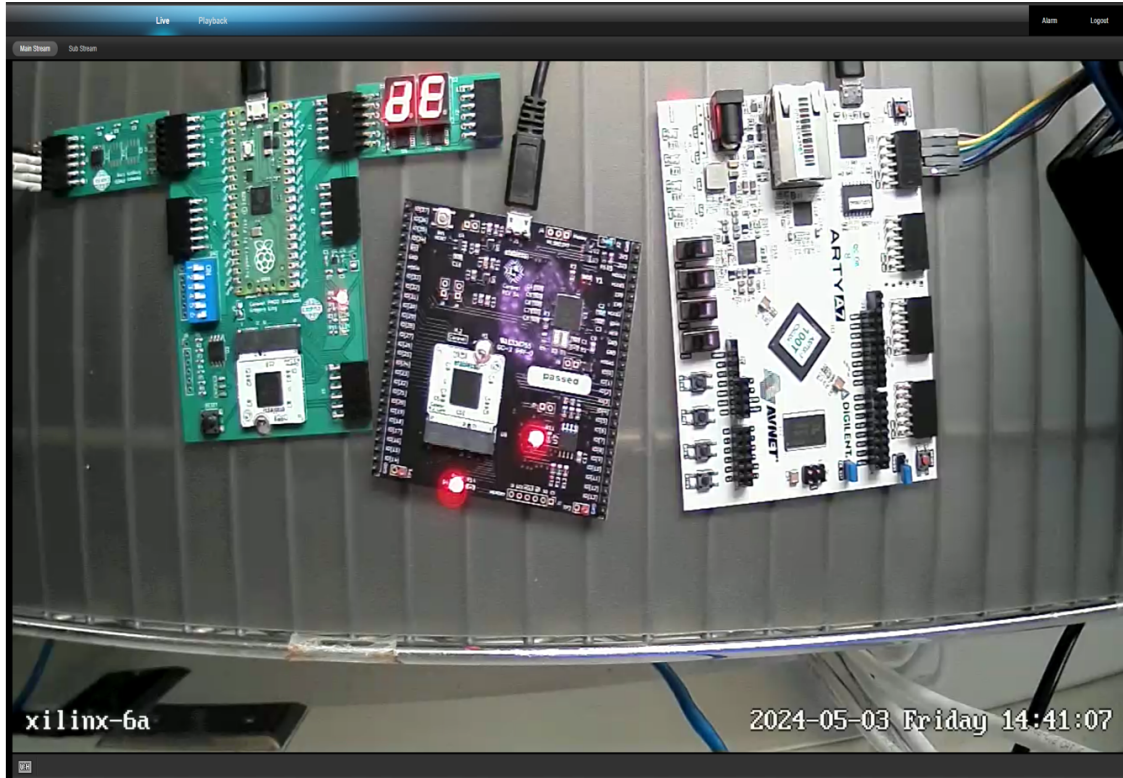


Figure 4.8 Bring-up Camera Interface

CHAPTER 5. MAIN IDEA

Documentation and tutorials are a necessity in ChipForge due to the high technical barrier of ASIC design and requirement of using unfamiliar open source tools with eFabless's design flow. By utilizing related works on documentation and tutorial best practices, a concise outline can be made to produce effective documentation within the context of the ISU curriculum for ChipForge members. This section will outline the design principles of the documentation and tutorials, the tools used to achieve these goals, and what information is included in the documentation to support ChipForge members.

5.1 Structure and Best Practices

In Chapter 2, multiple positive and negative aspects of documentation and tutorial writing were reviewed [Forward and Lethbridge \(2002\)](#); [Plösch et al. \(2014\)](#); [Aghajani et al. \(2020\)](#); [Uddin and Robillard \(2015\)](#). Based on those principles, the following core ideas of ChipForge documentation and tutorials will be considered when writing:

- **Keeping it Simple:** Text and examples will be concise to ensure students do not get lost through the technical flows, while providing additional contextual information in other documentation pages on an as needed basis.
- **Ease of Maintenance:** Documentation will be accessible and up to date, assuming changes to the open source process and need of information for ChipForge
- **Directed Access:** Documentation and tutorials will be organized into multiple pages, depending on the technical flow of the target design (Digital, Analog, Bring-up, onboarding) to ensure accessibility and organization.

- **Contextual Information:** Information will be considered within the context of the ISU curriculum, which is not currently possible in existing documentation from eFables or other sources.

Each of the defined principles will be considered when determining the tools used and design of documentation and tutorials for ChipForge members to outline an ASIC design process.

5.2 Tools Used

Multiple tools were selected on the foundation of the design principles **Keeping it Simple** and **Ease of Maintenance**. Tools support the writing, building, and publishing of the documentation of ChipForge. By utilizing an effective set of tools, the target design principles can be achieved for ChipForge documentation and tutorials.

5.2.1 Markdown Format

The selected format for writing documentation and tutorials is Markdown source files. Markdown provides basic text formatting, the ability to create lists, and organization between variable sizes of headers. In the most simple implementations, markdown source code can be written like a text file. But, with minimal learning, or **Keeping it Simple**, basic syntax can enable additional information to be highlighted and emphasized within the documentation. The ease of learning Markdown will enable more ChipForge members to update documentation, achieving **Ease of Maintenance**. By using markdown files, authors are not limited to proprietary software such as Microsoft Word files, which could limit editing and up-to-dateness of the documentation. Markdown files are also additionally easier to maintain within a design repository with git, in which changes to documentation can be reviewed with pull requests.

5.2.2 Docsify

Docsify [Docsify](#) is an open source documentation generator that will parse Markdown files and generate HTML files to be hosted for documentation. Docsify provides multiple supported

themes, plugins, and deployment through GitHub Pages to support documentation with lightweight building. This solution provides minimal additional overhead, which reduces the amount of tooling that authors need to learn to update documentation, ensuring the principle **Keeping it Simple**. Additionally, Docsify does not need to be reconfigured in a source repository after it is setup, guaranteeing **Ease of Maintenance**.

5.2.3 Gitlab Pages Hosting

Docsify enables automatic building and deployment through the use of either GitHub or GitLab Pages. GitLab is supported by the ISU ECpE department, and can be used for free to maintain a design repository for documentation and tutorials. By enabling GitLab Pages for the documentation repository, source Markdown code with documentation will be automatically rebuilt after being pushed to master by Docsify. After being built, the generated HTML build will be hosted automatically, ensuring that no author has to manually build and publish a new version of the wiki, guaranteeing **Ease of Maintenance**.

5.2.4 CircuiTikz

CircuiTikz [Redaelli et al. \(2024\)](#) is a schematic diagram tool that can be used to draw circuits in LaTeX documents. This package has been integrated into the ChipForge documentation, to enable rendered circuit schematics embedded within the Markdown source code. Figure 5.1 represents a sample circuit of a tri-state buffer, which would model how a GPIO port works on an ASIC design. This tool provides the ability to give contextual information to inexperienced members of ChipForge, who are going through the tutorials for the first time. By using **Contextual Information**, another method of teaching can be achieved in the tutorials with visual schematic diagrams.

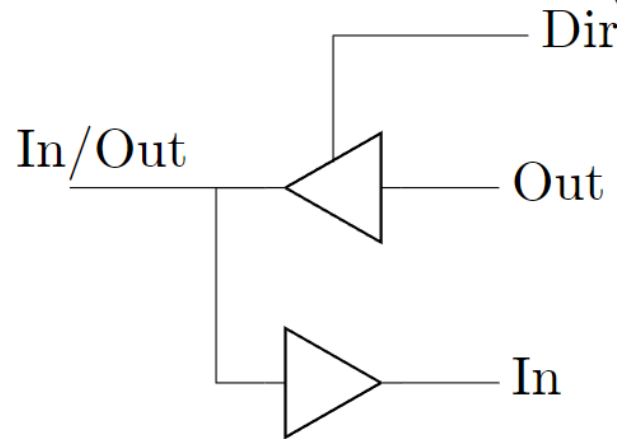


Figure 5.1 CircuiTikz Tri-State Buffer Schematic

5.3 Documentation Pages

ChipForge documentation has been broken into multiple subsections including a landing page, onboarding introduction, technical flows, and documentation maintenance. Segmenting documentation helps to achieve **Directed Access**, as it provides readers with the opportunity to more easily navigate the multiple pages. The ChipForge documentation is published on Gitlab Pages and is publicly available [Hafele et al. \(2024\)](#). An overview of each page, and a description of why it is needed, will be outlined in each of the following subsections.

5.3.1 Landing Page

By default, readers will arrive at the Home page, which is the original index of the documentation wiki. This page provides a description of each segment of the wiki, posed as questions for the reader. The first page that is referenced is the Introduction page, intended as a one time read for new members of ChipForge. After that, the Project Specific pages are referenced, including Bring-up, Digital, Analog, and Hardware wiki pages. Each of these pages outlines tool and ASIC design process details that are required to either design or validate an ASIC within ChipForge. Next, a link to Standards is included, which guides users to coding, file

naming, and git standards that are expected of ChipForge members. Finally, a link to a Meta page is included, which instructs authors of the documentation wiki how to manage the page, and where updates should be provided for each relevant technical topic. The main goal of the Landing Page is to provide **Directed Access** to ChipForge members, so that whether they are a new member, part of a technical team, or on the leadership cabinet, they can easily identify where to navigate within the wiki.

5.3.2 Introduction

Introduction is used as a first time onboarding tool for New Members of ChipForge. Since ChipForge members will have varying experience levels, or none, using Linux terminals or git, a baseline assumption of no experience should be made. For more experienced students, supplementary information can be skipped as needed. This does come at the cost of supplemental information being seen as "bloat" for experienced members, but it would come at the cost of fragmenting the documentation by splitting up on boarding between experience levels.

The Introduction Index page offers a guideline for how new members should navigate through each page. The first required page is New Members, which will step students through installing all required tools needed by a designer in ChipForge. Two additional pages are included as a reference for tools on an as needed basis, which are pointed to in each of the other Technical Flow pages. There are an additional two pages to provide supplementary information on how to use git and Linux command line interfaces. The organization of the index pages is structured similarly for all other wiki sections, and supports **Directed Access**.

On-boarding is started with students beginning at a New Members page, which outlines the following:

- VPN access
- On-Campus lab access
- Remote VDI Tool

- Git installation
- Remote server access
- SSH Authentication

Each of the above items are a necessity for each student, since the tools and processes are used for a variety of technical tasks. By guiding each ChipForge member through each section, it can be assumed that they have access to each of the above tools. A concise explanation is used to justify the usage and purpose for each tool to **Keep it Simple** and avoid bloat within the documentation. Since each ChipForge member is expected to step through this page, it is important to keep it as simple as possible to keep students moving through on-boarding.

Additional pages are included to reference other internal ChipForge tools such as the Toolchain and chipforge Command Line Interface [Ling \(2024\)](#). These are listed as optional in the index, and can be referenced for first time users when they proceed to the following Technical segments of the wiki. The additional tools are not included in Technical pages since they are referenced multiple times. For example, the Toolchain is used to source all project dependencies for the digital and analog design flows, as well as for sourcing code to validate a design.

Finally, Supplementary pages are included to provide an introduction to how to use Git and use a Linux command line interface. Assuming a wide range of experience levels, some students may have never used either of the required tools, while some are more experienced, including usage in advanced courses or industry. To streamline the process, the information is included for members who need it, but it is listed as optional, to not distract experienced students who are onboarding. These design principles support both **Keeping it Simple** and **Directed Access**.

After new ChipForge members have on-boarded through the Introduction pages, they are ready to proceed to one of the Technical segments of the wiki, which depends on what their starting project is. Each Technical page has its own independent track of pages and tutorials, and can be followed through from start to finish for their sample design flows.

5.3.3 Digital

The Digital section is prompted for ChipForge members who are looking to learn more about the digital design process of creating an ASIC with open source tools. It is assumed that students who are going through the digital design flow have taken atleast the first digital design course, CPRE 2810, at Iowa State university. Additionally, it would be ideal to have taken CPRE 3810 and other senior level digital design courses, but it is not required.

The first step on the index page guides readers to a set of curated tutorials, with a clear "Start Here" indicator, enabling **Directed Access**. The goal of the tutorials is to provide multiple sample Verilog designs that students can use to run through the full ASIC design process, including RTL level simulation, synthesis, gate level simulation, physical design, and precheck analysis. To achieve this, the complexity of the Verilog designs is kept to a minimum, to ensure there is a low barrier of entry for any student attempting to learn the tools. If students need more background information on how to implement Verilog designs, testbenches, or digital design topics, the pages Verilog and Verilog Testbench are included after the curated tutorials. These additional pages can provide additional **Contextual Information** for inexperienced students, but not get in the way as bloat for more experienced members.

Three tutorials were created as part of the Digital tutorial set, including a counter, ripple carry adder, and open source memory interface, OpenRAM. Each tutorial begins with a set of pre-requisites, which link back to the Introduction page, to ensure that students have all of the necessary tools installed. If this is the case, then students will be presented with a Design Background for each tutorial, including relevant contextual information for the target design that is being simulated and generated. Then, readers are prompted to clone the tutorial design repository that is maintained by ChipForge, and source all required open source tools through the internally support Toolchain [Ling \(2024\)](#). After that, the open source tools can be utilized through the same flow as supported by eFabless. Through each step of simulation and design, context to each step is given to explain what is being run with each step, to provide additional **Contextual Information** for students who have been apart of a digital design course previously.

The intent is to link what has been previously taught and experienced in the existing ISU curriculum, and connect that to the new tools and flows of the open source ASIC design process. At the end of each tutorial, readers are prompted to go to the next tutorial to learn about a different design and gain more experience with the flow.

Since the supported open source tools require specific commands to be run, additional pages that are broken apart were designed to provide experienced members with an accessible way to run the tools. This was something previous senior design teams desired, and has enabled current teams to interface with the tools at an accelerated rate. The separate pages include Pre-Layout simulation, physical design, post layout simulation, and precheck analysis. The intent is to keep these pages concise so students can access the desired commands needed to run the tools, and exclude contextual information that was referenced in the tutorials. This achieves the design target of **Keeping it Simple**, while still enabling the tutorials to provide **Contextual Information**.

A Reference page is included in the Digital section of the wiki to provide supplementary links to past senior design document deliverables, eFables provided documentation, and many references cited within this paper. This is included in its own section to support **Directed Access**, as students reported the large set of links was unfit to be placed in the same section as the commands to run tools or background design information. Additionally, a page for Troubleshooting has been created, but is currently empty. The intent is that when future members find common bugs, ChipForge can add them to a shared page so the issue is addressed and a workaround is found.

5.3.4 Analog

The structure of the Analog section is very similar to the Digital, changing what tools are used in the analog open source ASIC design process, and what contextual information is supported along with it. A sample tutorial is prompted for first time viewers of the Analog section, who want to learn more about analog design. The tutorial includes walking readers

through the design process of a simple inverter design, in which a schematic will be created, simulated, and implemented in physical design. Users will learn how to use Design Rule Checks and Layout Versus Schematic to analyze and compare their pre and post layout designs. It is assumed that students have taken at least one Analog VLSI course at Iowa State University, EE 3300, to have the proper background knowledge for each of the tutorial steps.

Like the Digital section, multiple independent pages are created to break down parts of the Analog design process, to **Keep it Simple** and provide **Directed Access** for experienced analog ASIC designers apart of ChipForge. These pages break down the separate design steps such as creating a schematic, simulating a schematic, physical layout, running DRC and LVS, and integrating a designed cell in the overall user design. An additional page was created, which was based on the previous work of an analog senior design team, which overviews how to implement ReRAM into a design, which is a future potential design option for members of ChipForge. This provides **Contextual Information** based on the past experience of senior design teams, that can be utilized for future ASIC MPW shuttles. Finally, there is again a reference page included with multiple links to other eFabless and SkyWater PDK resources, to be used as needed by students.

5.3.5 Bring-up

The Bring-up section of the wiki is focused around writing C code for the Management Core within the ASIC design, as well as providing tutorials to step through the process of validating the first received ASIC, the SHA-1 hashing IP block. Like the Digital and Analog sections, Bring-up begins with a set of guided tutorials, that are indicated as the first step on the index page, supporting **Directed Access**. The provided tutorials demonstrate how to clone a tutorial project, how to flash code to one of the bring-up development boards, and **Contextual Information** on how to write embedded systems code in C for the Management Core. In these tutorials, it is assumed that students have taken at least the first digital logic course CPRE 2810 and the first embedded systems course CPRE 2880, to have the proper background information to go through the design steps. By using the baseline curriculum at Iowa State taught to Sophomore

level students, code and digital designs can be explained in a more concise manner, to **Keep it Simple**. At the end of the tutorials, students can simulate designs to see how the Management Core interacts with the digital design in the user design area. Additionally, students have the option of generating a bitstream and loading their design to an FPGA [Ling \(2024\)](#). At the end of each tutorial, additional exercises are provided as a challenge to students to further their programming skills, utilizing their experience through the tutorial and previous coursework.

A wiki page demonstrating the different functionalities of the open source RISC-V Management Core is included, as a reference for any student involved in the bring-up process. Since the Management Core is new for any student in the ISU curriculum, it is essential to include as a reference to show students how to reference GPIO, Interrupts, and SPI. This information is primarily provided in the form of tables, to **Keep it Simple**. Additional links are included as reference to other eFables documentation for users.

A firmware page was created to teach ChipForge members about general embedded systems design principles. This can include introductory questions, assuming no prior knowledge is known about software design. Since validation is expected to be the first semester of any ChipForge members design experience, it is important to also cater to students who have less experience, or have potentially not taken either the first digital logic or embedded systems course. To build off this, a Reference page was included to outline simple digital design circuits using CircuiTikz [Redaelli et al. \(2024\)](#). This includes an explanation of basic logic gates, D Flip Flops, Shift Registers, and a Tri-State Buffer. It was discovered that students validating a design needed more background on the digital design, so this was chosen to be included under Bring-up instead of Digital. It is assumed that students in the digital design stage have taken at least one digital design course. Both of these pages provide additional **Contextual Information** that is catered towards students in the Iowa State curriculum, and can be built upon as needed at the request of students.

5.3.6 Hardware

As previously referenced, ChipForge has developed multiple Printed Circuit Board projects to expand the capabilities of ASIC validation [Ling \(2024\)](#). The Hardware section provides a link to the design repository of these projects, an explanation of each file that is generated, what to use to order a PCB or components, and a link to install the used open source schematic capture and PCB layout tools (KiCAD). Since there will be minimal active work designing PCBs for ChipForge in the future, and is not intended for the design flow of new members, no tutorials exist currently for PCB time. However, in the future, if interest is met, tutorials or development can be added to the Hardware section of the website. For now, Hardware provides the resources students need to fabricate and assemble future PCBs to be used to validate ASIC designs. By including the Hardware section independently to Bring-up, where the flow is focused on embedded systems design, this page achieves **Directed Access**.

5.3.7 Standards

The Standards pages are used to define ChipForge best practices for ASIC designs and software design. This includes a page for example Verilog coding standards, which can be used for the digital design flow. This provides example coding snippets for students creating digital designs, and assumes students have completed the first digital logic course, CPRE 281, to read the code. In a similar manner, a page for C coding standards has been created, and is currently empty. The goal is to add more to this page as ChipForge grows, to better define expected coding practices for students involved in the Bring-up process. This information would be relevant for any designer in Bring-up, or for any student who has taken the first embedded systems course, CPRE 288, at ISU. In a similar manner, pages for File Naming and Git have been created, which are planned to be used for project level organization as ChipForge grows. These standards have not yet been defined, due to the club being in its early stages. By summarizing all expected standards of ChipForge members, **Directed Access** is supported, as the standards are unrelated to running tools in each design flow.

5.3.8 Meta

The Meta section of the ChipForge wiki contains information needed for members who want to maintain documentation or the team. This includes an Information Technology page, which demonstrates what teams and organizations new ChipForge members and leadership should be added to. For example, a new ChipForge member must be added to the Microsoft Teams group to communicate with other members of the club. Another pages outline how to use Docsify to build the documentation wiki, and how to install the correct plugins to use CircuiTikz. Finally, an Updates page provides guidelines for what specific Markdown pages to edit when future tools or processes change. Each of these pages is primarily for the Leadership cabinet within ChipForge, and is intended to make the process of updating the documentation and managing ChipForge easier. Due to this, the Meta section provides **Ease of Maintenance** for the future authors of ChipForge.

CHAPTER 6. EVALUATION

6.1 Methods

The most beneficial feedback for the quality and contents of the documentation come from the users who are utilizing it, which includes senior design teams, First Honors Mentor Program students, and ChipForge members. Up until now, the primary form of feedback has been derived informally from discussions with students utilizing the commands and tutorials.

6.1.1 Informal Feedback

By utilizing the most relevant aspects of documentation determined by developer surveys, along with the past experiences of senior design teams, documentation and tutorials were initially constructed to meet those goals. These contents have been used and reviewed by multiple different follow on groups affiliated with ChipForge, which can derive how to improve future documentation and tutorials.

Initially, the first round of documentation was derived from the experience and deliverables of following digital senior design teams. This created a baseline for the Digital design flow that each of the 4 Digital senior design team would use to realize an independent design targeted for tapeout. As part of developing the documentation, meetings were held to gather feedback on the quality of the documentation with the Fall 2024 senior design team, which was tasked with delivering a framework for ChipForge's first tapeout design. This helped to model the most critical aspects of the Digital design flow that is documented on ChipForge in Spring 2024.

During the middle of the Spring 2024 term, weekly meetings were also scheduled to work with and gain feedback with the 5 First Honors Mentor Program students related to the Bring-up documentation. This group had limited engineering experience due to being new to the ECpE

program, which can help to provide motivation for what contextual information that is required to support students who are less familiar with ASIC design bring-up.

A third round of feedback was derived from the 2 Analog senior design teams, who were utilizing the limited set of commands and tutorials that were provided in the Analog section of the documentation. This provides another user group to explore if the given documentation will suffice, or how the utilization of the less familiar tools can be improved.

6.2 Results

The best form of feedback we have currently collected is the informal feedback from multiple affiliated groups with ChipForge, after the initial documentation and tutorials were created. As of now, the anonymous survey with various user groups has not been conducted. In the future, results can be utilized to drive critical improvements or additions to the existing documentation for ChipForge.

6.2.1 Informal Feedback

The informal feedback collected from Digital senior design teams, FHMP, and analog senior design teams helped to drive the motivation for how to further improve the documentation, as well as what principles to focus on in our final documented deliverable up to the end of the Fall 2024 semester. In general, positive feedback was given, with an indication that the documentation and tutorials have helped various groups achieve functioning designs at a faster rate.

As documentation was being updated, Digital senior design teams reported that the digital tutorials were a supportive introduction in how to walk through the required tools and end to end digital ASIC design flow. Teams reported that the barrier for entry was much lower, which was possible by utilizing the past experience and deliverables of previous senior design teams, alongside iterative feedback on how to improve organization. Students also appreciated weekly meetings to explain the flows alongside the documentation, which would be a similar role to leadership members on ChipForge. Finally, students appreciated the fact that documentation was

quickly updated after problems or feedback were reported, demonstrating support for the students. In terms of items that were improved through this process, the documentation was originally less segmented, which led to users becoming lost in the navigation, and slowing down their process to find commands to run the tools. This drove the need for the Keep it Simple and Directed Access principles of our documentation design, which was improved afterwards.

The FHMP group provided a different set of results, explaining what they remembered as the most impactful from their semester working with the bring-up process for ChipForge's first fabricated ASIC. Students reported that their most memorable experiences included working with a logic analyzer to measure pins, explaining Verilog on a whiteboard, using Git, and soldering tester circuit boards together. All of these experiences were novel for the Freshman honors students demonstrating that these skills were valued and interesting. It is also important to consider that these skills should be highlighted in the documentation to provide better context for the bring-up and digital design sections to further support those teachings from ChipForge leadership. There were multiple technical sections students found were lacking however, including the commands for how to utilize revision control with Git. To solve this, a reference page was included in the Introduction section of the documentation for basic commands to use git.

The two current Analog senior design teams have been utilizing the analog documentation and inverter tutorial since it has been created. Students have discovered errors in the commands, including a command with a space between two characters that should not have existed. The second active analog senior design team has successfully cloned down the documentation repository and published the change directly, supporting the documentation principle for Ease of Maintenance.

CHAPTER 7. FUTURE WORK

Given the initial development of documentation for ChipForge, the largest contribution that can be done is further development and additions to the documentation. This can include more contextual information or tutorials for each of the ASIC design flows. Additionally, ChipForge leadership will need to actively maintain the documentation to ensure it does not go stale. Formal survey feedback can also be utilized to better cater the documentation to both ChipForge and senior design teams in the future. Finally, documentation can be used to help support students interested in ASIC design in the existing ISU coursework.

7.1 Additional Documentation

Currently the most well maintained and documented sections of the ChipForge website includes the digital ASIC design process and the bring-up validation process. This is due to the experience of both graduate researchers who founded the club, and from what was followed in past senior design projects. Keeping this in mind, it would be ideal to further develop the processes and documentation around the analog design flow and validation process. Additionally, further background on basic embedded systems and digital/analog VLSI design would be beneficial for younger students, who do not have much coursework under their belt as a newer student in the ISU curriculum.

With the creation of a second analog senior design team, there is growing interest from ECpE students to follow through the analog ASIC design process. As stated before, this includes a wide range of tools, which are currently factored into the supported toolchain. While these tools are currently supported to be ran internally, there is little background information on how to properly utilize them or best design practices, due to lack of experience of the students writing the documentation. The hope is that future leadership teams or optimistic club members will

develop this documentation the first time an analog ASIC design is conducted within ChipForge with the ChipIgnite program.

From our survey results, it was discovered that there was less context information surrounding the tutorials than we anticipated. This is likely due to the minimal previous experience in basic design courses of some students, due to the broad range of a university co-curricular. Due to this, it would be ideal to add more background information on digital design or embedded information, to explain in better detail basic functionalities such as what an adder is or how address mapped memory works. This information could be provided at the beginning or end of a tutorial, and be actively referenced by leadership members during training workshops.

Other proposed ideas for documentation have included a page for outreach, which could be used to gather interest and support for ChipForge within Iowa State University. The documentation pages could be used to help guide ChipForge leadership in how to go through outreach processes, as well as an appealing page to attract students to the club. This could act as a landing page for next steps into the club, rather than the current landing page which acts as a guide to further technical information.

7.2 Maintenance

As was discovered through the multiple iterations of the analog and digital senior design projects, the provided open source tools, including OpenROAD, Magic, and GTKWave will be updated often. While this is beneficial for the open source community and future ChipForge designs, it will require leadership to actively monitor and provide updates to documentation when these tools change. Luckily, git hashes are provided in the subrepos, which indicate which release tag is used for new iterations. This should help indicate when tools change, using the support of revision control. An example of maintenance could be updating the tutorials for some of the common commands that are used to generate a digital simulation, which are required to be used from the provided caravel framework and synthesis tools.

In a similar manner, the ChipForge team will be required to provide updates for the internal supported tools that were created only for ChipForge use. This responsibility should fall on the team member who is actively providing updates for these tools, who should release proper documentation updates to the website at the same time. This will enable all existing and future ChipForge members to proceed with using both bug fixes and new feature updates.

Finally, repository organization, club management, and wiki management will need to be maintained on the documentation website. Repository organization can pertain to how ChipForge organizes its code base within Gitlab. If a source repository is renamed or moved, it is imperative that the related tutorials are also updated immediately, guaranteeing that active members will not have clone errors when trying to pull down projects. Additionally, relevant on boarding information will need to be maintained, to ensure students stay engaged and active during the on boarding process. This includes keeping up to date contact information of club leadership, meeting locations, and relevant tools to get started. Wiki management upkeep can include information on how to build the documentation website using Docsify, which is its own open source project which can change as well.

7.3 Active Feedback

Active feedback will be required within ChipForge, to further engage with club members and determine what is beneficial in the current documentation and what should be changed or added. Similar to the surveys conducted with existing students, more surveys should be regularly taken, and feedback should be encouraged within club meetings both during and after on boarding. This will ensure that new members remain active and engaged, and are supported in an acceptable co-curricular learning environment. The changes required to modify documentation will be heavily dependent on the feedback given, and how frequent a similar sentiment is shared between club members. It may be challenging to meet everyone's needs, as each club member will have varying levels of experience.

Future anonymous surveys are planned to determine the qualitative aspects of the created documentation and tutorials as well. These surveys will be conducted on a per-semester basis with different users from senior design groups and ChipForge members to gather results on the quality of the deliverables of this work.

Some proposed questions are as follows:

- What aspects of ChipForge documentation did you find most beneficial/challenging?
- What pages of the documentation did you reference most often?
- Was there any relevant information that was missing from the documentation?
- Are there any improvements you could suggest to the organization of the documentation?

7.4 Relation to Coursework

The documentation and tutorials can also be utilized in the established ECpE courses at Iowa State University. Students can utilize the available tools and funding of ChipForge, along with the existing design IP developed as part of lab material, to deliver more tapeout ready designs.

Currently, in ChipForge's first tapeout, two designs which were derived from Coursework were created. One design was a Multiply and Accumulate module from a lab in CPRE 4870, Hardware Design for Machine Learning, as well as an Arithmetic Logic Unit from CPRE 3810, Computer Organization and Assembly Level Programming. With continuous support of ChipForge documentation and tutorials, both students and professors could benefit from further integration of ChipForge in the ECpE curriculum.

CHAPTER 8. CONCLUSION

With the growth of open-source ASIC design at Iowa State University, the need for documentation has been met for the currently supported tools and processes. Through the studied documentation strategies and techniques, the first version of ChipForge documentation has been created to assist students of varying skill levels and needs. With the collection of feedback from current user groups, key strengths and weaknesses in the documentation has been identified, indicating future improvements. With this documentation in place, and future formal survey results, the next group of ChipForge leadership members can provide updates to the documentation and tutorials as needed. Coinciding with other research [Ling \(2024\)](#), ChipForge has successfully completed their first submission ready tapeout design to eFabless as of the Fall 2024 semester. Moving forward, ChipForge will work on developing additional tapeout designs while validating the returned fabricated and packaged ASICs. By utilizing the given documentation, students are capable of fully realizing an open source vertically integrated ASIC design flow.

REFERENCES

- Aghajani, E., Nagy, C., Linares-Vásquez, M., Moreno, L., Bavota, G., Lanza, M., and Shepherd, D. C. (2020). Software documentation: the practitioners’ perspective. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, ICSE ’20*, page 590–601, New York, NY, USA. Association for Computing Machinery.
- Aghajani, E., Nagy, C., Vega-Márquez, O. L., Linares-Vásquez, M., Moreno, L., Bavota, G., and Lanza, M. (2019). Software documentation issues unveiled. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1199–1210.
- Caravel Harness. Caravel harness. <https://caravel-harness.readthedocs.io/en/latest/>. Accessed: 2024-3-8.
- Caravel Management SoC - Litex. Caravel management soc - litex. <https://caravel-mgmt-soc-litex.readthedocs.io/en/latest/>. Accessed: 2024-3-8.
- Caravel User Project. Caravel user project. <https://caravel-user-project.readthedocs.io/en/latest/>. Accessed: 2024-3-8.
- Chen, P.-H., Tsao, C., and Raina, P. (2023). An open-source 4×8 coarse-grained reconfigurable array using skywater 130 nm technology and agile hardware design flow. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.
- Cirimelli-Low, J., Khan, M. H., Crow, S., Lonkar, A., Onal, B., Zonenberg, A. D., and Guthaus, M. R. (2023). Sram design with openram in skywater 130nm. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.
- Dagenais, B. and Robillard, M. P. (2010). Creating and evolving developer documentation: understanding the decisions of open source contributors. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE ’10*, page 127–136, New York, NY, USA. Association for Computing Machinery.
- Digilent. Digilent pmod™ interface specification 1.2.0.
- Docsify. Docsify. <https://docsify.js.org/>. Accessed: 2024-27-7.
- Edwards, R. T., Shalan, M., and Kassem, M. (2021). Real silicon using open-source eda. *IEEE Design Test*, 38(2):38–44.
- eFabless. eFabless. <https://efabless.com/>. Accessed: 2024-27-7.

- Efabless YouTube Channel. Efabless youtube channel.
https://www.youtube.com/@efabless_channel. Accessed: 2024-3-8.
- Forward, A. and Lethbridge, T. C. (2002). The relevance of software documentation, tools and technologies: a survey. In *Proceedings of the 2002 ACM Symposium on Document Engineering, DocEng '02*, page 26–33, New York, NY, USA. Association for Computing Machinery.
- Gaw. Gaw - gtk analog wave viewer. <https://gaw.tuxfamily.org/>. Accessed: 2024-3-8.
- GTKWave. Welcome to gtkwave. <https://gtkwave.sourceforge.net/>. Accessed: 2024-28-7.
- Hafele, J., Ling, G., and Duwe, H. (2024). Chipforge documentation.
<https://doi.org/10.5281/zenodo.14279431>.
- Head, A., Jiang, J., Smith, J., Hearst, M. A., and Hartmann, B. (2020). Composing flexibly-organized step-by-step tutorials from linked source code, snippets, and outputs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20*, page 1–12, New York, NY, USA. Association for Computing Machinery.
- Herman, K., Montanares, M., and Marin, J. (2023). Design and implementation of integrated circuits using open source tools and sky130 free pdk. In *2023 30th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*, pages 105–110.
- Hinkle, C. M. and Koretsky, M. D. (2019). Toward professional practice: student learning opportunities through participation in engineering clubs. *European Journal of Engineering Education*, 44(6):906–922.
- Icarus Verilog. Icarus verilog. <https://bleyer.org/icarus/>. Accessed: 2024-3-8.
- Iowa State University Courses and Programs. Iowa state university courses and programs.
<https://catalog.iastate.edu/azcourses/>. Accessed: 2024-27-7.
- Iowa State University ECpE Senior Design. Iowa state university ecpe senior design.
<https://seniord.ece.iastate.edu/>. Accessed: 2024-27-7.
- Iowa State University Student Organizations. Iowa state university student organizations.
<https://www.stuorg.iastate.edu/>. Accessed: 2024-27-7.
- ISU Starting a New Student Organization. Starting a new student organization.
<https://studentengagement.iastate.edu/starting-new-student-organization>.
 Accessed: 2024-27-7.
- Kim, A. S. and Ko, A. J. (2017). A pedagogical analysis of online coding tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE '17*, page 321–326, New York, NY, USA. Association for Computing Machinery.

- KLayout. Klayout. <https://www.klayout.de/>. Accessed: 2024-3-8.
- Ling, G. (2024). Design and organization of a bring-up first vertically-integrated chip design and fabrication co-curricular.
- Magic VLSI. Magic vlsi layout tool. <http://opencircuitdesign.com/magic/>. Accessed: 2024-28-7.
- Muse Semiconductor. Muse semiconductor. <https://www.musesemi.com/>. Accessed: 2024-4-8.
- Netgen 1.5. Netgen 1.5. <http://opencircuitdesign.com/netgen/>. Accessed: 2024-3-8.
- Olewnik, A., Chang, Y., and Su, M. (2015). Co-curricular engagement among engineering undergrads: do they have the time and motivation? 10.
- Oliveira, W. M. M. and Brito-Filho, F. A. (2022). Development of a baseband amplifier for a 130 nm cmos process using open-source design flow. In *2022 6th International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*, pages 1–5.
- OpenROAD. The openroad project. <https://theopenroadproject.org/>. Accessed: 2024-27-7.
- OpenROAD documentation. Openroad documentation. <https://openroad.readthedocs.io/en/latest/>. Accessed: 2024-3-8.
- Plösch, R., Dautovic, A., and Saft, M. (2014). The value of software documentation quality. In *2014 14th International Conference on Quality Software*, pages 333–342.
- Powell, H., Williams, R., Weikle, R., and Brandt-Pearce, M. (2015). Towards a t shaped electrical and computer engineering curriculum: A vertical and horizontally integrated laboratory/lecture approach. 122.
- Ramirez, N. and Zoltowski, C. B. (2022). First-year experiences – how the vertically integrated projects (vip) model addresses grand challenges and abet outcomes. In *2022 IEEE Frontiers in Education Conference (FIE)*, pages 1–4.
- Redaelli, M. A., Lindner, S., Erhardt, S., and Giannetti, R. (2024). Circuitikz manual. <https://texdoc.org/serve/circuitikzmanual.pdf/0>.
- Sillito, J., Maurer, F., Nasehi, S. M., and Burns, C. (2012). What makes a good code example? a study of programming qa in stackoverflow. In *Proceedings of the 2012 IEEE International Conference on Software Maintenance (ICSM)*, ICSM '12, page 25–34, USA. IEEE Computer Society.
- SkyWater SKY130 PDK. Skywater sky130 pdk. <https://skywater-pdk.readthedocs.io/en/main/index.html>. Accessed: 2024-3-8.

Spickard-Prettyman, S., Qammar, H., Broadway, F., Cheung, H., and Evans, E. (2004). The impact of vertical integration of design teams on the chemical engineering program. In *34th Annual Frontiers in Education, 2004. FIE 2004.*, pages T2G/15–T2G/19 Vol. 1.

Tiny Tapeout. Tiny tapeout. <https://tinytapeout.com/>. Accessed: 2024-4-8.

Uddin, G. and Robillard, M. P. (2015). How api documentation fails. *IEEE Software*, 32(4):68–75.

Visconti, M. and Cook, C. (2002). An overview of industrial software documentation practice. In *12th International Conference of the Chilean Computer Science Society, 2002. Proceedings.*, pages 179–186.

Xschem. Xschem : schematic capture and netlisting eda tool.
<https://xschem.sourceforge.io/stefan/index.html>. Accessed: 2024-3-8.