# DESIGN AND CHARACTERIZATION OF A STANDARD CELL LIBRARY FOR THE SKYWATER 130NM PROCESS

By

TEODOR-DUMITRU ENE

Bachelor of Science in Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
2018

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2020

DESIGN AND CHARACTERIZATION OF A STANDARD CELL LIBRARY FOR THE SKYWATER 130NM PROCESS

Thesis Approved:

Dr. James E. Stine, Jr.

Thesis Adviser

Dr. Bingzhe Li

Dr. Ickhyun Song

Name: TEODOR-DUMITRU ENE

Date of Degree: DECEMBER, 2020

Title of Study: DESIGN AND CHARACTERIZATION OF A STANDARD CELL LIBRARY FOR THE SKYWATER 130NM PROCESS

Major Field: ELECTRICAL ENGINEERING

Abstract: This thesis discusses the design and characterization of a standard cell library for the SkyWater 130nm process. A detailed overview of the design process is examined, addressing both methodology as well as theoretical arguments behind specific design choices. Results consist of a robust and flexible standard cell kit as well as a complete and customizable design flow. In accordance with the objectives of the Free and Open-Source Silicon Foundation, the standard cell library and design flows discussed herein can be found inside publicly-accessible code repositories alongside the rest of the SkyWater Open Source PDK.

## TABLE OF CONTENTS

# LIST OF TABLES

Table Page

## LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

The MOSFET is the foundation of modern computer design. The digital logic of nearly all computer chips is implemented with MOSFET devices, with roughly 13 sextillion transistors having been manufactured since their invention [1]. Logic gates are a specific arrangement of such transistors that performs a specific boolean logic function. In turn, standard cells are logic gates designed to act as modular interchangeable blocks within a design.

The design process of nearly all chips designed today make use of standard cell methodology. Standard cell methodology consists of the use of standard cells as a convenient abstraction in the design process [2]. High-level physical designers can assemble entire chips out of standard cells without having to immediately concern themselves with the low-level physical design of the transistor devices themselves. In addition, although digital logic is ultimately implemented through the use of MOSFET devices, the use of standard cells allows logic designers to implement desired functions without having to worry about physical implementation. Any boolean logic can be implemented using a small set of standard cells [2], and thus a logic designer may rely on combinations of available standard cells to physically implement their design.

## 1.1 Free and Open-Source Silicon

Despite the constant improvement and advancements in the fields of integrated electronics and digital computing, many of the fruits of this technological advancement have been historically gated behind a growing paywall. A growing movement has emerged lately towards the creation of a free and open-source ecosystem for the design, implementation, and fabrication of silicon-based devices [3].

A particular effort in this direction has been started in 2020 as a result of a collaboration between Google, SkyWater, and numerous other collaborators such as OSU [4]. This Open Source PDK project is dedicated to providing this aforementioned free and open-source ecosystem for all stages of the ASIC design process, including fabrication.

It is the goal of this project to collaborate with the SkyWater Open Source PDK project to first provide an open-source, well-documented, standard cell design flow. Second, the author seeks to create a full, flexible, standard cell kit that can be used according to the needs of any designers interested in working in the Google free and open-source ecosystem. Finally, the work of this project is designed to allow for future iteration not just by the author, but by anyone interested in contributing to this ecosystem.

## 1.2 Organization

The organization of this thesis is as follows: Chapter II will discuss standard cell design criteria. Chapter III will discuss the implementation flow and methodology used in the design of the this standard cell kit. Chapter IV will discuss the results thereby obtained. Finally, Chapter V presents the conclusion to the work in addition to possible future avenues for work on the topic.

# CHAPTER II

# STANDARD CELL DESIGN

The first step of standard cell design is the mapping of a design's desired logical function into a combination of standard cells. This step is commonly referred to as 'synthesis'. The next step consists of the physical placement and routing of a design's standard cells. This is commonly referred to as 'place-and-route'. Both steps are aided by a wide selection of EDA tools and design flows.

Chiefly within synthesis, but also in a reduced capacity within place-and-route, the choice of standard cells for the implementation of each section of the logical design is repeatedly re-visited and optimized. As standard cells are designed to be modular blocks, this replacement of standard cells is a straightforward process. Additionally, within place-and-route standard cells are laid out along certain placement tracks in a design grid, and further routing tracks and design grids are established for each of the metal routing layers available. This greatly simplifies the place-and-route process, allowing EDA tools to process large designs within a reasonable amount of execution time.

## 2.1 Design Goals

The main design goals for standard cell creation are to minimize area and power consumption while maximizing speed. When keeping other design details constant - such as fabrication process, technology node, and supply voltage - prioritizing one of these three goals normally comes at the expense of one or both of the others.

One such possible trade-off is that of lower power versus higher speeds. A straightforward and effective method of performing this trade is by varying the threshold voltage of the transistors composing the cell. A lower threshold voltage will result in higher switching speeds at the expense of higher leakage power. Similarly, a higher threshold voltage will results in lower switching speeds but lower leakage power [5]. The trade-off between these two design goals can be performed while keeping the third goal, area, constant.

Another such possible trade-off is that of lower area versus higher speeds. Should the height of a standard cell be kept constant, a designer can perform this trade-off by providing multiple options of the same standard cell with different drive strengths. Drive strength is a measure of a cell's capability to charge or discharge the capacitance connected to its output port and is directly proportional to the cell's $W/L$ ratio [5]. As such, a higher drive strength cell can be easily constructed by increasing the $W$ parameter of its transistors. Since the height of the standard cells is kept constant one must then typically resort to fingering, which is the connection of multiple transistors in parallel to create a device with a higher effective $W$ [5]. The use of multiple transistors in parallel thus increases the area of the standard cell.

Should the height of a standard cell be allowed to vary, the solution is far more straightforward. The designer can directly increase the $W$ parameter of the transistors that make up a standard cell, resulting in a taller, faster, cell. However, varying the height of standard cells introduces new challenges.

## 2.2 Cell Heights and Tracks

As previously mentioned, however, the place-and-route process inside a standard cell flow makes use of placement tracks and design grids to greatly expedite implementation. Part of the advantage of this grid-based system is to ensure that the power grid spanning the design grid is continuous and does not significantly obstruct routing. As all CMOS logic requires low-impedance access to both supply and ground power, it is typical for these power rails to be situated at identical heights across all standard cells within a library. When implemented as such, standard cells can always be abutted to one another both vertically as well as horizontally, resulting in a highly modular and flexible design.

Having standard cells of varying heights within the same library renders it extremely difficult to take advantage of such a grid-based placement system and is thus highly unadvisable. Thus, modern standard cell libraries are typically defined in terms of their cells' uniform height. This is commonly done using the unit of 'tracks'. Here, tracks refer to the routing tracks of the lowest metal layer available for inter-cell routing. Such a definition is advantageous as it is partially decoupled from the transistor's raw dimensions and thus a sense of consistency between different technology nodes can be contained.

Figure 2.1 provides a visual example of the concept or routing tracks. Shown is an empty cell that is 12 tracks tall, or 12T. The purple lines represent possible metal wires that can be drawn across the cell on the lowest non-interconnect metal layer. To be noted is that although this cell is 12 tracks tall, only 9 total routing wires can actually be accommodated. This is typical, as the power rails necessary for CMOS design take up significant space.

Although enforcing a uniform height across all cells in a library eliminates the straightforward solution for trading lower area vs higher speeds, designers still have many options available, such as those previously listed. In particular, modern standard cell libraries frequently offer multiple variants of threshold voltage for each cell, allowing for easy trade-off of power versus speed within a single design. In addition, standard cell libraries offer a large selection of both logical functions and drive strengths, in order to suit any possible designer needs.

## 2.3 Cell Widths and Filler Cells

As mentioned previous, standard cells must abut both vertically as well as horizontally when physically placed inside of a layout. This is partially in order to ensure continuous power rails, partly to allow for a grid-based placement system, and partly to satisfy manufacturing design rules. Specifically, having small empty gaps not populated by standard cells will result in various design rule check violations in most manufacturing processes. As such, any small horizontal gaps between cells must be 'filled' with specialized cells called filler cells. It is thus evident that a standard cell library must at least provide a filler cell with width equal to the width of the smallest possible such gap.

Should cell widths not conform to a standard pattern the gaps between cells could potentially be any multiple of the smallest allowable manufacturing grid unit, requiring a filler cell of such small dimensions. Considering that the smallest standard cell will typically be two orders of magnitude larger in width than the manufacturing grid unit, even a single gap may require hundreds of filler cells, adding up to unmanageably large numbers of filler cells for an entire design.

It is trivial to prove that it is necessary and sufficient for a standard cell library to contain a filler cell whose width is the greatest common divisor of all other cell widths. In order to minimize the number of filler cells inserted at the end of a design's place-and-route phase, libraries must be designed so that the greatest common divisor of their cell widths is as large as possible. However, should this restriction be too harsh, significant amounts of space will be wasted by stretching standard cells to equal their width to the nearest multiple of the filler cell's width.

### 2.4 Well and Substrate Taps

The MOSFET device is typically only used for three of its terminals: source, drain, and gate. These three terminals allow it to superficially act as a switch, allowing the implementation of digital logic. However, the fourth terminal of the MOSFET, the body, can significantly affect the function of the device [5]. Most applications seek to minimize this ‘body effect‘ by tying this fourth terminal directly to the rails: supply voltage for PMOS devices, and ground for NMOS devices. This configuration allows all devices in a design to act in similar fashion, with their threshold voltage not being affected by a voltage being applied to the body terminal [5].

The question of how to make this connection is one of design choice. Historically, it was common to have these connections, known as well and substrate taps, situated within each standard cell [2]. However, in modern design nodes, transistor sizes are small enough that it is sufficient to insert well and substrate taps separate from standard cells, using specialized tap cells spaced out at distances recommended by the foundry’s design rules.

As SkyWater 130nm is an older node, the design rules recommend having a tight distribution of well and substrate taps relative to the size of the standard cells. In addition, having built-in well and substrate taps allows for far more flexibility not just in the use of cells, but also in the use of novel EDA tools. With the main design goal of this project being to create a flexible standard cell kit and flow that can be used and iterated upon as easily as possible, the decision was made to include well and substrate taps inside all standard cells.

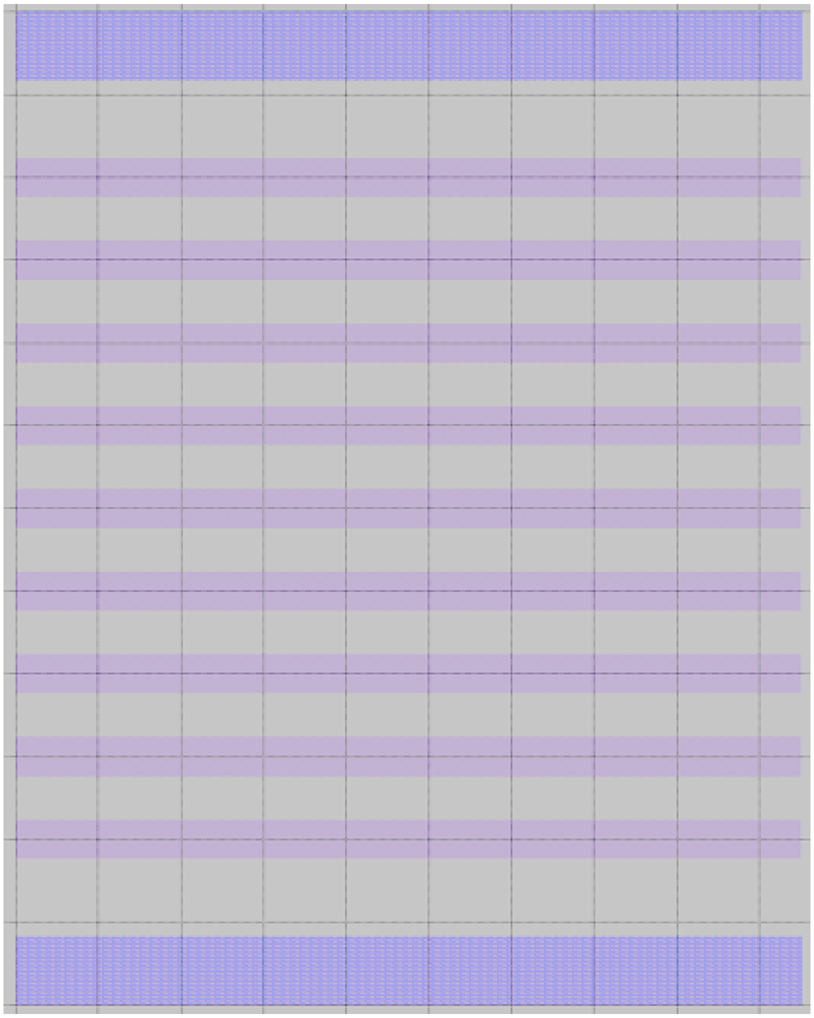

Figure 2.1: Routing tracks for a 12-track cell

## 2.5 OSU Standard Cell Kit Goals

As previously discussed, the goal of this project is to create a full, flexible, standard cell kit that can be adapted to fit the needs of any designers working in the Google free and open-source ecosystem. Given the standard cell design goals and restrictions discussed in this section, the initial project goal became to create four different cell libraries, each with a different track height. In addition, each standard cell library was envisioned as consisting of three different variants: high speed, medium speed, and low speed. Choosing a desired track height would allow the designer to make an initial choice for prioritizing area versus speed, while the option of multiple cell speed variants within each library allows the designer flexibility within their design. Table 2.1 illustrates the initial design goal for the sky130_osu standard cell kit.

| Library Variant | | Cell Height | MOSFET Threshold |
|---|---|---|---|
| sky130_osu_18T | _hs | 18T (6.66 µm) | low Vt |
| | _ms | | regular Vt |
| | _ls | | high Vt |
| sky130_osu_15T | _hs | 15T (5.55 µm) | low Vt |
| | _ms | | regular Vt |
| | _ls | | high Vt |
| sky130_osu_12T | _hs | 12T (4.44 µm) | low Vt |
| | _ms | | regular Vt |
| | _ls | | high Vt |
| sky130_osu_9T | _hs | 9T (3.33 µm) | low Vt |
| | _ms | | regular Vt |
| | _ls | | high Vt |

Table 2.1: List of cells proposed for creation.

# CHAPTER III

## IMPLEMENTATION FLOW AND METHODOLOGY

Standard cell design begins with a schematic, which can be drawn on paper or using an EDA tool. The standard cell designer then converts this schematic into a physical layout using a layout editor. The direct output of this process is a GDSII file, a universally-used layout file standard. Creating a physical layout is the bulk of the work, but once complete, this design must be extracted, analyzed, and characterized.

First, the GDSII layout file is extracted into a spice schematic. A basic spice extraction is sufficient for performing layout-versus-schematics checks and simulations, but to ensure analysis and characterization is accurate, parasitic extracted (PEX) must be performed. Parasitic extraction annotates the spice deck with extracted resistances, capacitances, and parasitic devices such as diodes.

Once a spice extraction with annotated parasitic information is performed, it can be used to perform charaterization on the cell. Characterization consists of running large numbers of spice simulations in order to obtain good approximations for the cell's behavior in a large range of conditions. The output of characterization is typically in the form of a Liberty Timing Format file.

In parallel with the spice extraction process, a Liberty Exchange Format view, LEF, is extracted from the GDSII layout. This view is designed with place-and-route in mind, where only the back-end-of-line (metal) layers are relevant. In order to simply the job of the place-and-route tool, a LEF view contains only information that is directly relevant to place-and-route.

Once these extractions and characterizations have been completed, cells can be tested by implementing a design through synthesis and place-and-route. Results from this implementation can be used to further refine the cells in order to improve their capabilities in a realistic scenario.

## 3.1 Layout

For the purposes of this project, and indeed all standard cell design, Magic is the preferred layout creation tool. Magic is an open-source tool maintained by Tim Edwards [6]. While commercial tools are well-suited for large-scale layout editing, Magic is extremely well-suited for cell design, and has been used for such purposes by the author on technology nodes as small as 14nm.

In general, Magic has multiple features that make it highly attractive for the manual creation of small designs. One key feature is automatic, real-time, DRC checking. Another key advantage of Magic is the ease with which it can be scripted. For example, Magic allows the designer to not worry about fabrication-specific layers that are irrelevant to the cell design process. Instead, the designer can opt to simply create a script that automatically appends such layers to the final output, reducing the complexity and clutter of their design.

## 3.2 SPICE

SPICE (Simulation Program with Integrated Circuit Emphasis) is an often-used protocol and programming setup that is useful for simulating small circuits at the transistor level with predefined models. Transistors for a given technology are modeled in SPICE using model files that contain hundreds of parameters describing the physical behavior of the transistor. These parameters provide sufficient accuracy to check the functionality and the performance of a design both in digital and analog fields.

### 3.3 Liberty Exchange Format

As previous mentioned, the Liberty Exchange Format (LEF) view is designed with place-and-route in mind. While a layout contains a large variety of layers that are required for fabrication, the place-and-route process only concerns itself with the placement and routing of standard cells. As long as the underlying cell layout is valid, all extraneous information can be abstracted away.

A cell's LEF view thus provides the place-and-route tool with a view with only the routing metal layers are present. The LEF view also specifies the location and identity of all of a cell's pins, as well as antenna information the place-and-route tool uses to detect and avoid antenna violations.

In addition to the cell-specific information encoded in its LEF view, the LEF file format provides separate technology-specific information to the place-and-route tool. This consists of information about each metal layer such as their width and spacing design rules, their resistivity, their capacitance, and a list of valid vias between each metal layer.

### 3.4 Liberty Timing Format and Characterization

While SPICE models can be highly accurate, for the purpose of implementation simpler models are necessary. Characterization creates tables of values under a variety of conditions for characteristics such as rise and fall delay, setup and hold timing constraints, and power consumption. Synthesis and place-and-route flows use these simple tables to interpolate values that may appear in a realistic design. While interpolation is far less accurate than simulation, it is a necessary trade-off for the implementation and optimization of physical designs.

Table 3.1 list the multiple corners and modes that were used to characterize the standard cell kit created as part of this project.

| Corners | Voltage (V) | Temperature (°C) |
|---|---|---|
| FF | 1.56 | -40 |
| FF | 1.76 | -40 |
| FF | 1.95 | -40 |
| FF | 1.95 | 100 |
| FF | 1.95 | 150 |
| SS | 1.28 | -40 |
| SS | 1.44 | -40 |
| SS | 1.60 | -40 |
| SS | 1.60 | 100 |
| SS | 1.60 | 150 |
| TT | 1.20 | 25 |
| TT | 1.35 | 25 |
| TT | 1.44 | 25 |
| TT | 1.50 | 25 |
| TT | 1.62 | 25 |
| TT | 1.68 | 25 |
| TT | 1.80 | 100 |
| TT | 1.80 | 150 |
| TT | 1.80 | 25 |
| TT | 1.89 | 25 |
| TT | 2.10 | 25 |

Table 3.1: List of modes and corners chosen for cell characterization

### 3.5 Implementation and Testing

To test the functionality and performance of the cells, the full library was used as the basis for a series of full design flow runs. The design flow involves several steps that each exercise different components of the standard cells. A design starts out with a hardware description using a hardware descriptive language (HDL) such as Verilog or VHDL. The HDL describes the logic in a high-level fashion as well as how data flows from register to register within the design. In this state, the logic is not directly mappable to cells in the library. Synthesis is the process that converts an HDL design to logic using only valid cells within the library. This design, much like a schematic, does not contain any information about the location of the cells nor how they will connect together physically.

Place-and-route is the process that creates a physical layout of the cells, including how they will connect together and how large the design will be. As the name would suggest, placement calculates where cells from the library should be physically placed in a design to allow for maximum performance while also allowing for maximum wiring flexibility. In addition to these factors, power consumption and clock timings are also taken into account to ensure the resulting design will function if manufactured. The routing component involves drawing the wires between the placed cells and making sure there are no collisions, lengthy wires that could cause timing issues, and unroutable situations that result in impossible designs. At the end of this process, a final GDSII layout is created that contains all of the material layer designs and connections needed to fabricate the design and create a real, functioning chip.

# CHAPTER IV

# RESULTS

Tables 4.1 through 4.4 list the cells currently available as part of the OSU standard cell kit contribution to the SkyWater Open Source PDK project. Each cell is replicated across three different track libraries - 18T, 15T, 12T - as well as three different threshold voltage variants - hs, ms, ls - resulting in a total of 9 cell sets and 576 total cells. While the 9 different cell sets are similar, their layout is not identical and manual redesign was required in each case. As can be seen, a wide complement of digital logic cells is present, in addition to all necessary physical cells required for actual implementation of a design.

In order to test the cells, the same RISC-V single-cycle processor design was implemented using each of the 9 cell-sets. The schematic of the design used can be seen in Figure 4.1, and the place-and-route implementation results can be seen in Table 4.5. Of note in the implementation results is that the power, PDP, values are highly inaccurate. This is in part due to difficulty in estimating average power consumption of a design without a target use-case, and partly due to deficiencies in the current device models available in the SkyWater 130nm PDK.

| Cell Name | Description |
|---|---|
| ADDFXL | 3-2 counters |
| ADDFX1 | |
| ADDHXL | 2-2 counters |
| ADDHX1 | |
| AOI21XL | AND-OR-INVERT gate |
| AOI22XL | |
| OAI21XL | OR-AND-INVERT gate |
| OAI22XL | |
| NAND2XL | NAND cells |
| NAND2X1 | |
| NOR2XL | NOR cells |
| NOR2X1 | |
| MUX21XL | 2-1 muxes |
| MUX21X1 | |
| TBUFIXL | Active-high inverting tri-state buffer |
| TBUFIX1 | |
| TNBUFIXL | Active-low inverting tri-state buffer |
| TNBUFIX1 | |

Table 4.1: Current selection of standard cells - Part 1.

| Cell Name | Description |
|---|---|
| INVXL | Inverters |
| INVX1 | |
| INVX2 | |
| INVX4 | |
| INVX6 | |
| INVX8 | |
| BUFXL | Buffers |
| BUFX1 | |
| BUFX2 | |
| BUFX4 | |
| BUFX6 | |
| BUFX8 | |
| ADD2XL | AND gates |
| ADD2X1 | |
| ADD2X2 | |
| ADD2X4 | |
| ADD2X6 | |
| ADD2X8 | |
| OR2XL | OR gates |
| OR2X1 | |
| OR2X2 | |
| OR2X4 | |
| OR2X6 | |
| OR2X8 | |

Table 4.2: Current selection of standard cells - Part 2.

| Cell Name | Description |
|---|---|
| DFFXL | Positive edge D flip-flop |
| DFFX1 | |
| DFFNXL | Negative edge D flip-flop |
| DFFNX1 | |
| DFFSXL | D flip-flop with asynchronous active-low set |
| DFFSX1 | |
| DFFRXL | D flip-flop with asynchronous active-low reset |
| DFFRX1 | |
| DFFSRXL | D flip-flop with asynchronous active-low set and reset |
| DFFSRX1 | |

Table 4.3: Current selection of standard cells - Part 3.

| Cell Name | Description |
|---|---|
| ANT | Antenna diode cell |
| ANTFILL | Fill cell with same footprint as ANT |
| FILLX1 | Filler cells |
| FILLX2 | |
| FILLX4 | |
| FILLX8 | |
| FILLX16 | |
| FILLX32 | |
| TIEHI | Used to tie a net off to VDD through a FET |
| TIELO | Used to tie a net off to GND through a FET |
| DECAPXL | Decoupling capacitor |
| DECAPX1 | |

Table 4.4: Current selection of standard cells - Part 4.

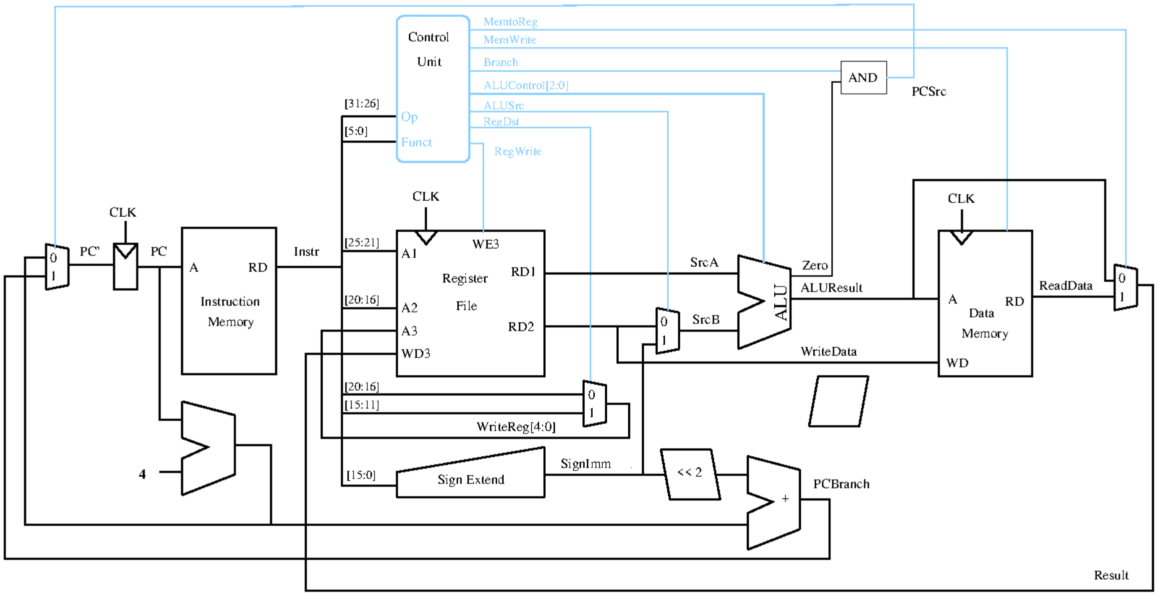

Figure 4.1: OSU RISC-V single-cycle processor

| Library Variant | | Frequency (MHz) | Area ($\mu m^2$) | PDP (pJ) |
|---|---|---|---|---|
| sky130_osu_18T | _hs | 410.56 | 156,190 | 310.82 |
| | _ms | 380.00 | 157,540 | 164.76 |
| | _ls | 320.41 | 158,170 | 163.98 |
| sky130_osu_15T | _hs | 401.66 | 133,010 | 374.04 |
| | _ms | 363.49 | 132,640 | 198.77 |
| | _ls | 310.49 | 132,840 | 169.61 |
| sky130_osu_12T | _hs | 372.89 | 106,630 | 444.57 |
| | _ms | 345.32 | 106,800 | 449.84 |
| | _ls | 285.80 | 107,270 | 352.64 |

Table 4.5: Performance of cells on RISC-V core implementation.

# CHAPTER V

## CONCLUSION AND FUTURE RESEARCH

The first goal of this project has been met, as an entire standard cell design and characterization flow has been published as one of the repositories making up the SkyWater Open Source PDK project, sky130_osu_sc. The design found therein contains all the implementation steps detailed in the previous section. All individual steps are documented and have Makefile-based flows that allow for straightforward execution. Examples are provided, and all scripts are customizable and parameterizable. As of the publication of this thesis, the flow created by this project is in use by two unrelated university research groups who seek to create their own standard cells in the SkyWater 130nm technology node.

The second goal of the project, to create a full, flexible, standard cell kit has also been used. Currently three libraries exist, each with a different track height and each containing three threshold voltage cell variants. The initial goal was for the creation of a fourth, 9T, cell track height library. While this effort has not been abandoned, it is currently in the process of development. Future goals include the completion of this fourth cell library.

While the standard cell kit currently extant is complete and functional, there are still numerous ideas for further improvements of its functionality and capability. Some of these ideas involve further reducing the size of the cells and increasing their speed through design revisions, while others involve the addition of design-specific cells as suggested by the current user-base of the standard cell kit. Future revisions will involve the testing and implementation of such design improvements.

A further goal that is in active development is the physical fabrication of test designs using the standard cell kit created in the course of this project. At the time of the publication of this thesis, an active effort is being made to submit a design for fabrication on SkyWater's first free-of-charge publicly-available shuttle run.

As part of the SkyWater Open Source PDK project and a member of the project's Slack group, I provide daily support and suggestions to individuals seeking to use our cells and our flow, enabling iteration on my work by other interested individuals. Continued support in this manner will be provided for the duration of the project.

# REFERENCES

[1] D. Laws, "13 sextillion counting: The long winding road to the most frequently manufactured human artifact in history,"

[2] D. Jansen, *The electronic design automation handbook*. Springer, 2011.

[3] "The free and open source silicon foundation." https://fossi-foundation.org/.

[4] "Skywater open source pdk." `https://github.com/google/skywater-pdk`, 2020.

[5] Y. Tsividis, *Operation and modeling of the mos transistor*. Oxford University Press, 3 ed., 2010.

[6] T. Edwards, "Magic," 8.3.

[7] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff.," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006.

[8] Synopsys, "Hspice," 2018.09.

[9] Synopsys, "Design compiler," 2019.12.

[10] C. D. Systems, "Abstract," 2004.1.

[11] C. D. Systems, "Liberate," 19.2.

[12] C. D. Systems, "Innovus," 19.1.

# VITA

Teodor-Dumitru Ene

Candidate for the Degree of

Master of Science

Thesis: DESIGN AND CHARACTERIZATION OF A STANDARD CELL LIBRARY FOR THE SKYWATER 130NM PROCESS

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2020.

Completed the requirements for the Bachelor of Science in Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2018.

Experience:

Graduate Research Assistant - VLSI Computer Architecture Research Group
OSU
May 2019 - December 2020