KLE Technological University
Creating Value
Leveraging Knowledge

DEPARTMENT OF ELECTRONICS ENGINEERING (VLSI Design And Technology)

Minor Project Report

on

STANDARD CELL DESIGN

By:

1. **Swathi Raghavendra Kulkarni** USN:01FE22BEV007
2. **Hitashi Dinesh Haldonkar** USN:01FE22BEV013
3. **Apoorva Appanna Bhoi** USN:01FE22BEV028

Semester: VI, 2024-2025

Under the Guidance of

Prof.Rajeshwari.Mattimani

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031
2024-2025

DEPARTMENT OF ELECTRONICS ENGINEERING (VLSI DESIGN AND TECHNOLOGY)

CERTIFICATE

This is to certify that project entitled **"Standard Cell "** is a bonafide work carried out by the team of **"Swathi Raghavendra Kulkarni(01FE22BEV007), Hitashi Dinesh(01FE22BEV013), Apoorva Appanna Bhoi(01FE22BEV028)"**. The project report has been approved as it satisfies the requirements with respect to the mini project work prescribed by the university curriculum for BE (VI Semester) in Department of Electronics Engineering(VLSI design and technology) of KLE Technological University for the academic year 2024-2025.

Prof.Rajeshswari.M	**Dr Sujatha Kotabagi**	**B. S. Anami**
Guide	**HOD(VLSI Design And Technology)**	**Registrar**

External Viva:

Name of Examiners **Signature with date**

1.

2.

ACKNOWLEDGMENT

We are thankful to KLE Technological University, Hubballi, for giving us the opportunity to work on our project, "Standard Cell Design" and for providing all the necessary resources and an environment conducive to learning and development. This focus on innovation and practical learning has played a key role in helping us grow and successfully complete the project. We thank our guide, Prof. Rajeshswari Mattimani, for her excellent guidance, continuous encouragement, and availability to solve our problems at any time. Her ability to simplify complex ideas and give constructive feedback has really helped us to overcome many problems in the project. Her guidance has been inspiring during this whole journey. We would like to extend our sincere thanks to our Head of the Department, Dr. Sujatha S. Kotabagi, for her support and encouragement, which have been invaluable in the completion of our project. We also thank Mr. Venugopal D. Kulkarni for his dedication to teaching and helping students with the fundamental concepts, which laid a strong foundation for our project. We are also grateful to our peer group and all the faculty members who urged and cooperated with us during this process. Our thanks extend to teaching assistants and non-teaching staff for support and to all who have directly or indirectly contributed to the successful completion of our project.

-Team 12

ABSTRACT

In modern VLSI design, optimized and reliable standard cells are essential for building efficient digital systems. This project presents a complete design flow for developing and characterizing standard cells using Cadence tools. Standard cells implemented include basic logic gates (NAND, NOR, XNOR), a D Flip-Flop, and a 4-bit Priority Encoder. The workflow involves schematic design, layout creation, DRC/LVS verification, post-layout simulation, and .LIB file generation.

Initially, each cell is designed and simulated at the schematic level for functional correctness. Layouts are then manually created and verified against design rules. Post-layout views are extracted to include parasitic effects, and simulations are repeated to observe timing changes such as increased delay, rise, and fall times. These effects highlight the importance of parasitic-aware design in real-world applications.

Finally, each cell is characterized using Cadence Liberate to generate standard Liberty (.LIB) files, which contain accurate timing, power, and area data. These files are crucial for integration into digital synthesis and ASIC workflows. This project demonstrates a practical and industry-relevant standard cell development methodology, offering insights into timing optimization and physical verification techniques essential in VLSI design.

Contents

List of Figures

Chapter 1

Introduction

In the rapidly advancing field of Very-Large-Scale Integration (VLSI) design, the development of optimized and reliable standard cell libraries is critical. Standard cells form the backbone of digital integrated circuits and are reused across numerous designs to maintain consistency, performance, and scalability. These cells include logic gates, flip-flops, multiplexers, encoders, and other fundamental building blocks that are synthesized and placed to form complex digital architectures.

The process of creating a high-quality standard cell involves several key stages: schematic design, layout creation, simulation (both pre- and post-layout), physical verification (Design Rule Check — DRC, and Layout Versus Schematic — LVS), and characterization. Accurate timing and power models are essential for these cells, and they are captured in Liberty (`.LIB`) format through a characterization process. This `.LIB` file is used during logic synthesis, timing analysis, and the place-and-route stages of ASIC development.

This project focuses on implementing a comprehensive design flow for selected standard cells such as NAND, NOR, XNOR gates, a D Flip-Flop, and a 4-bit Priority Encoder. Using Cadence Virtuoso, Assura, Spectre, and Liberate, we simulate, verify, and characterize each cell, ultimately producing technology-accurate `.LIB` files suitable for digital design flows.

1.1 Motivation

In today's digital era, the demand for faster, smaller, and more power-efficient integrated circuits has made optimized standard cell design a critical aspect of VLSI development. Standard cells form the backbone of digital systems, and their performance directly impacts the overall functionality, power consumption, and speed of complex chips.

While commercial libraries exist, designing custom cells allows for better control over transistor sizing, power optimization, and layout efficiency, making them more suitable for specific applications. It also provides deeper insight into how post-layout parasitics affect circuit behavior, which is essential for accurate timing analysis.

Another key motivation is to gain hands-on experience with industry-standard tools such as Cadence Virtuoso, Assura, and Liberate. This project enables us to follow a professional VLSI design flow—from schematic creation to layout, simulation, and characterization—bridging the gap between academic concepts and real-world chip design practices.

1.2 Objectives

The objective of this project is to design, simulate, and characterize a set of standard cells—NAND, NOR, XNOR gates, a D Flip-Flop, and a 4-bit Priority Encoder—using

Cadence design tools. The workflow includes creating transistor-level schematics, verifying functionality through simulation, designing layouts, and performing DRC and LVS checks to ensure physical correctness.
Post-layout simulations are carried out using parasitic-extracted views to analyze timing parameters like rise time, fall time, and delay. Each cell is then characterized using Cadence Liberate to generate accurate .LIB files, which are essential for digital synthesis and timing analysis. This project also aims to develop familiarity with the complete VLSI cell design flow, reflecting industry practices.

1.3 Literature survey

Standard cell libraries are an integral part of digital VLSI design. They consist of pre-designed and pre-characterized logic gates and sequential elements that can be reused across different designs. The use of standard cells significantly improves design productivity, enables automation in synthesis and layout, and ensures consistency in functionality and performance. The design, simulation, and characterization of these cells have been extensively studied and documented in both academic research and industrial practice.
Books such as "CMOS VLSI Design: A Circuits and Systems Perspective" by Neil Weste and David Harris provide a foundational understanding of how standard cells are constructed from CMOS transistors, covering both static CMOS logic and transmission gate logic. They also explain the importance of transistor sizing and switching behavior on delay, power, and area—critical factors in standard cell design.
Several research papers and technical reports highlight the role of layout parasitics in degrading circuit performance. Post-layout parasitic extraction allows designers to model the real-world resistance and capacitance effects that are introduced during fabrication, significantly impacting rise/fall times and overall delay. This emphasizes the need for post-layout simulation and characterization for accurate timing analysis.
In modern design workflows, EDA tools such as Cadence Virtuoso are widely used for schematic capture and layout design, while Assura is employed for physical verification through DRC (Design Rule Check) and LVS (Layout Versus Schematic). Spectre, a part of the Cadence simulation suite, enables analog and digital transient simulations for verifying signal integrity and functionality.
For characterization, Cadence Liberate is a robust and industry-accepted tool used to generate Liberty (.LIB) files, which include detailed timing, power, and area models of standard cells under various operating conditions. These files are crucial for synthesis tools like Synopsys Design Compiler and for accurate static timing analysis (STA) during the implementation flow.
Previous student and industry projects have also explored custom cell development, reporting that manually designed standard cells allow for better optimization and understanding of trade-offs in digital design. The process of developing and characterizing a cell library is not only educational but also aligns with practices followed in semiconductor companies, making it highly relevant for training and research purposes.
This project builds upon these foundational concepts and tools, integrating schematic design, physical layout, post-layout extraction, and characterization into a complete standard cell design methodology.

1.4 Problem statement

In the context of advanced VLSI design, the accuracy and reliability of standard cells are of paramount importance. Existing cell libraries may not always meet the specific requirements of a given design in terms of timing, power, or area. Additionally, without proper post-layout

simulation and characterization, the actual performance of cells may deviate from expectations, leading to design failures at the integration stage.
This project addresses these challenges by designing custom standard cells from scratch, simulating their behavior pre- and post-layout, and generating characterized .LIB files. By doing so, we ensure that the cells are functionally correct, physically realizable, and performance-accurate, making them suitable for integration into real-world ASIC and SoC designs.

1.5 Organization of the report

Chapter 2: The chapter introduces an overview of the model, discussing functionality and design architecture.
Chapter 3: This chapter brings out the results obtained for our model and verifies the results.
Chapter 4: This is the conclusion chapter of the report and discusses the future scope of the project.

Chapter 2

System design

2.1 Functional block diagram

The functional block diagram of the project represents the step-by-step design and characterization flow of standard cells. Each block in the flow corresponds to a specific process stage, highlighting the use of Cadence tools for schematic design, layout, verification, and simulation, followed by characterization using Cadence Liberate. Below is a high-level description of the flow:

- **Schematic Design Block:**
  Design of standard cells (NAND, NOR, XNOR, D-FF, and 4-bit Priority Encoder) at the transistor level using Cadence Virtuoso.
- **Simulation Block:**
  Pre-layout simulations using Spectre to extract rise time, fall time, and propagation delay. Functional verification is carried out to ensure logical and timing correctness.
- **Layout Block:**
  Manual layout design of each cell to ensure optimal area and proper connectivity. Metal layers, diffusion regions, and poly layers are used to define the physical structure.
- **DRC and LVS Verification Block:**
  DRC (Design Rule Check) ensures the layout conforms to fabrication rules.
  LVS (Layout Versus Schematic) confirms that the layout matches the original schematic.
- **Parasitic Extraction and Post-Layout Simulation Block:**
  Use of Cadence Assura to extract parasitic components and generate `av_extracted` views. Post-layout simulations are performed to analyze timing degradation due to parasitics.
- **Abstract View and Symbol Generation Block:**
  Logical views, symbols, and pin definitions are created for each standard cell. Abstract views are generated to facilitate digital synthesis and integration into larger designs.
- **Characterization Block:**
  Setup of the Cadence Liberate environment with required input files such as `.sp`, `include_tt`, and template files. The `char.tcl` script is used to characterize each cell and generate Liberty (`.lib`) files under specific operating conditions (1.8 V, 25°C).

2.2 Design alternatives

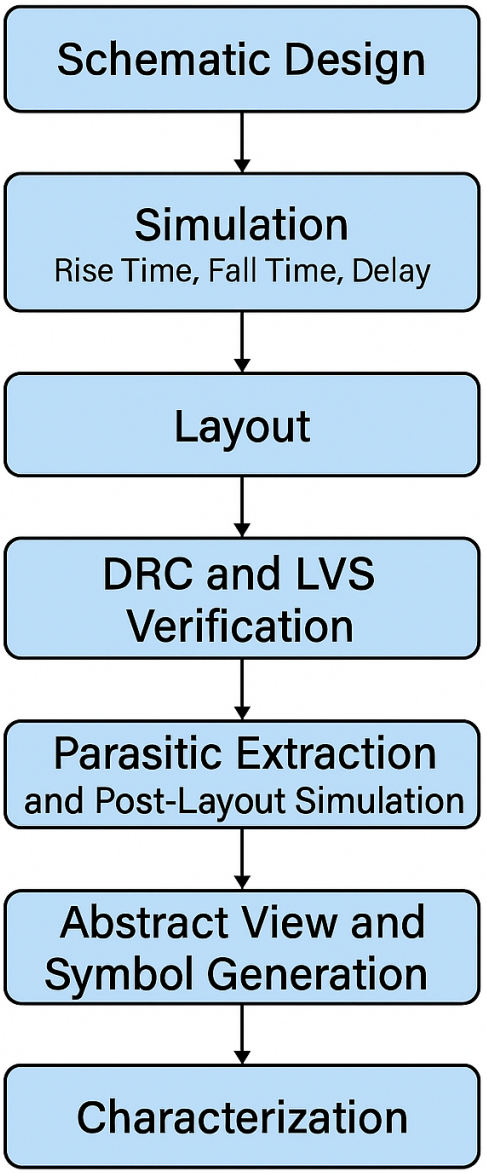

Figure 2.1: Block Diagram

2.3 Final Design

Throughout the project, various tools and design approaches were considered. The final methodology was selected based on reliability, educational relevance, and compatibility with available infrastructure. Below are the key decisions and their justifications:

- **Tool Choice:**
  While Synopsys and Mentor Graphics tools were potential alternatives, Cadence tools were selected due to their academic availability, seamless integration, and comprehensive suite for the entire VLSI design flow.
- **Design Style:**
  Static CMOS logic was chosen over dynamic logic or pass-transistor logic. This choice enhances design reliability and simplifies the layout process while maintaining noise margins and power efficiency.
- **Layout Method:**
  Manual layout was preferred over automated layout generation. Manual techniques provided better control over parasitic elements, diffusion sharing, and cell compactness—key factors in optimized standard cell design.
- **Simulation Flow:**
  Both pre-layout and post-layout simulations were performed. This dual-step simulation flow allowed the observation of real-world effects, such as delay degradation due to parasitic resistances and capacitances extracted from the layout.
- **Characterization Tools:**
  Cadence Liberate was used for cell characterization. It was selected for its compatibility with Cadence schematic and layout data, and for its ability to automatically generate Liberty (`.lib`) files required for timing analysis and integration into digital design workflows.

Chapter 3

Implementation details

3.1 Specifications and Final System Architecture

This project focuses on the development and characterization of the following standard cells:

- **Logic Gates:** NAND, NOR, XNOR
- **Sequential Cell:** D Flip-Flop
- **Combinational Block:** 4-bit Priority Encoder

Design Specifications

- Technology: Generic 180nm CMOS process
- Supply Voltage: 1.8 V
- Operating Temperature: 25°C
- Design Style: Static CMOS
- Simulation Tool: Cadence Spectre
- Layout Verification: Cadence Assura (DRC, LVS)
- Characterization Tool: Cadence Liberate
- Output Format: Liberty (.LIB) files

Final System Architecture

The overall design flow consists of the following integrated stages:

1. **Schematic Design:** Transistor-level logic implementation using Cadence Virtuoso.
2. **Simulation:** Functional verification using testbenches to extract delay, rise time, and fall time.
3. **Layout Design:** Manual layout creation with strict adherence to DRC rules.
4. **Verification:** DRC and LVS checks to ensure correctness and manufacturability.
5. **Parasitic Extraction:** Generation of extracted views for post-layout simulation.

6. **Symbol and Abstract Generation:** Creation of views required for integration in digital design flows.
7. **Characterization:** Generation of `.lib` files using Cadence Liberate and timing scripts.

Key Files Used

- `.lib` (Liberty File): Contains timing, power, and area information of the standard cell. Essential for synthesis and timing analysis using tools like Synopsys Design Compiler and PrimeTime.
- `.sp` (SPICE Netlist): Describes the circuit at the transistor level; used for simulation and input to Liberate during cell characterization.
- `include_tt` (Technology Model): Provides SPICE models for NMOS and PMOS devices under typical process-voltage-temperature (PVT) conditions (1.8 V, 25°C). Required for accurate simulation and characterization.
- `.tcl` (Script File): Automates the Liberate characterization process by defining cell behavior, specifying input conditions, and linking netlists and models.

These files are interconnected and form the backbone of the VLSI standard cell design and characterization flow. The `.sp` and `include_tt` files serve as input to both simulation and characterization, the `.tcl` script orchestrates the characterization process, and the `.lib` file is the final deliverable representing the fully characterized behavior of each standard cell.

3.2 Algorithm

The project does not involve a traditional software algorithm but follows a procedural VLSI design flow. This flow ensures that each standard cell is correctly designed, verified, and characterized for integration into digital systems. The steps can be summarized as follows:

1. **Design Schematic:** Create the transistor-level schematic for the desired logic cell (e.g., NAND gate) using Cadence Virtuoso.
2. **Simulate the Schematic:** Use Spectre to verify functional correctness and extract timing parameters such as rise time, fall time, and propagation delay.
3. **Draw Layout:** Design the physical layout in Virtuoso ensuring adherence to design rules and optimal area usage.
4. **Run DRC and LVS:** Perform Design Rule Check (DRC) and Layout Versus Schematic (LVS) using Cadence Assura to ensure that the layout matches the schematic and complies with fabrication constraints.
5. **Extract Parasitics:** Use Assura to generate `av_extracted` views that include parasitic resistance and capacitance components.
6. **Post-Layout Simulation:** Simulate the parasitic-extracted views using Spectre to evaluate real-world timing effects such as delay degradation.
7. **Generate Symbol and Abstract View:** Create the symbol and abstract view of the standard cell using the Abstract tool for use in digital design environments.
8. **Set Up Liberate Environment:** Prepare the necessary files including the SPICE netlist (`.sp`), technology model (`include_tt`), and template files.

9. **Run Characterization Script:** Execute the `char.tcl` script in Cadence Liberate to characterize the cell and generate the corresponding Liberty (`.lib`) file.

Each of these steps is an essential part of the VLSI standard cell implementation pipeline, ensuring that the final cells are not only functionally correct but also optimized and physically realizable for larger digital system integration.

Chapter 4

Optimization

4.1 Introduction to optimization

In VLSI design, optimization refers to the process of improving design parameters such as area, power, speed, and timing to meet performance targets while ensuring manufacturability. During standard cell design, optimization plays a key role in transistor sizing, layout compactness, and delay reduction. The goal is to achieve efficient cells that consume minimal resources and operate reliably across different process, voltage, and temperature (PVT) conditions.

4.2 Types of Optimization

Optimization in VLSI standard cell design is essential to achieve a balance between performance, area, and power. The following types of optimization techniques were considered during the project:

Timing Optimization: Aims to reduce propagation delay, rise time, and fall time of standard cells. This is achieved through careful transistor sizing, minimizing parasitic capacitances, and optimizing the signal paths.

Area Optimization: Focuses on minimizing the layout footprint of a cell. Efficient use of silicon area leads to higher integration density, reduced chip cost, and better scalability.

Power Optimization: Targets the reduction of both dynamic and leakage power. Techniques include minimizing switching activity, using minimum-sized transistors in non-critical paths, and applying threshold voltage control strategies.

Parasitic Optimization: Reduces the impact of interconnect resistance and capacitance, which can degrade signal integrity and increase delay. This is achieved by optimizing metal routing, avoiding unnecessary overlaps, and minimizing long wire segments.

Design Rule Optimization: Ensures that the layout complies with foundry-defined Design Rule Check (DRC) constraints. Adhering to these rules helps avoid costly rework and ensures the design is manufacturable.

4.3 Selection and Justification of Optimization Method

In the context of this project, two primary optimization objectives were prioritized: **timing** and **area**. These goals were selected based on their direct impact on performance, manufacturability, and scalability in real-world VLSI systems.

Why Timing Optimization?

- Accurate timing is fundamental for the correct operation of synchronous digital systems. Delays in critical paths can lead to setup or hold time violations in sequential circuits.
- Post-layout simulations revealed increased delays due to parasitic resistance and capacitance. To mitigate this, transistor sizing and interconnect layout were refined to minimize rise and fall times.
- Optimizing timing ensures that the designed standard cells can operate reliably at target clock frequencies and integrate smoothly into larger digital systems without violating timing constraints.
- Logical effort and path balancing techniques were considered where applicable to improve transition times and reduce skew.

Why Area Optimization?

- In high-density digital ICs, standard cells are replicated thousands to millions of times. Reducing the footprint of individual cells leads to better silicon utilization.
- Compact layouts help in minimizing wire lengths, which in turn reduces parasitic capacitance and improves routing efficiency.
- Manual layout allowed precise control over spacing, diffusion sharing, metal usage, and pin alignment — all of which contributed to a more compact and efficient cell design.
- Area-efficient cells also contribute to lowering fabrication costs and improving power density per mm^2.

Consideration of Other Optimization Types

Although timing and area were prioritized, other forms of optimization were recognized and noted for future scope:

- **Power Optimization:** While not the main focus in this phase, power reduction is crucial for battery-operated and thermally constrained systems. Techniques such as multi-threshold CMOS (MTCMOS), clock gating, and dynamic voltage scaling (DVS) could be integrated in future iterations.
- **Parasitic and Routing Optimization:** These were partially addressed through careful layout practices, but more advanced routing tools and metal layer optimization can further enhance signal integrity.
- **Design Rule Optimization:** Ensured all layout structures adhered to fabrication constraints to avoid costly rework and to guarantee yield.

Conclusion

The chosen optimization methods—timing and area—provided the best trade-off between performance and resource utilization in the current project scope. These optimizations enhance the standard cells' suitability for practical integration into digital synthesis flows and ASIC development pipelines.

Chapter 5

Results and discussions

5.1 Result Analysis

The designs for basic gates (NAND, NOR, XOR, XNOR), a D Latch, and a 4-bit Priority Encoder were successfully realized and implemented. The below important observations were made:

- DRC (Design Rule Check) Clean:

All the layouts were DRC-clean, confirming design rule compliance and manufacturability.

- LVS (Layout Versus Schematic) Matched:

Layouts were compared with their respective schematics, and netlists matched exactly, ensuring functional correctness.

- Parasitic Extraction and Timing Analysis:

Parasitic RC extraction was conducted for every layout. Data extracted was utilized to calculate propagation delays and to describe the performance of every cell.

- .lib File Generation:

Liberty (.lib) files were generated successfully for every cell utilizing characterized timing, area, and functional information. These files facilitate integration into standard-cell-based synthesis streams.

- Area and Power Observations:

Area utilization was optimal for simple gates. Power consumption was calculated using capacitance and transition activity, and trends were shown to be as expected between different cell types.

- Functional Simulation:

Post-layout and behavioral simulation ensured the logical operation of all cells to be correct, confirming structure and functionality of both. [a4paper,12pt]article geometry array booktabs longtable caption titlesec margin=1in

- Characterization Summary: NAND, NOR, XOR, XNOR, D Flip-Flop, and Encoder

1)Common Library Parameters

2)Power Pins

3)Cell-Level Summary

4)Timing Summary

Parameter	Value
Voltage Unit	1 V
Time Unit	1 ns
Leakage Power Unit	1 nW
Default Voltage	1.8 V
Default Temp	27 °C
Delay Model	Table Lookup
Max Transition	1.5 ns
Fanout Load	1

Pin	Type	Voltage
VDD	Primary Power	1.8 V
VSS	Primary Ground	0 V

Cell	Leakage Power(nW)	Function	Timing Sense	Max Cap(pF)
NAND	0.0687322	(!A) + (!B)	Negative Unate	0.6
NOR	0.0714341	(!A * !B)	Negative Unate	0.6
XOR	0.0244035	(!A * B) + (A * !B)	Non-Unate	0.6
XNOR	0.0340114	(!A * !B) + (A * B)	Non-Unate	0.6
D Flip-Flop	0.425627	Edge-triggered Storage	Non-Unate	0.6
Encoder	Estimated ∼0.05	4-to-2 Priority Encode	Combinational	—

Cell	Rise Delay(ns)	Fall Delay(ns)	Rise Transition(ns)	Fall Transition(ns)
NAND	0.346–10.511	0.183–5.437	0.574–21.041	0.318–10.707
NOR	0.582–19.19	0.138–3.54	1.08–39.94	0.21–6.44
XOR	0.20–18.5	0.15–3.2	0.8–35	0.2–5.9
XNOR	0.25–19.0	0.17–3.5	1.0–37	0.3–6.2
D Flip-Flop	0.388–10.25	0.295–3.50	∼0.62–21	∼0.24–6.45
Encoder	0.45–5.0	0.3–3.2	0.7–15.0	0.4–8.5

5.2 Discussion on optimization

Optimization of VLSI layout design is essential to attaining a balance between performance, area, and power. Throughout the duration of this project, a number of areas of optimization were recognized and explored partially:

- Layout Area Optimization:

Through the strategic placement of transistors and reducing routing congestion, cell area was minimized without taking a toll on performance. An attempt was made to preserve symmetry in transistor location to provide signal integrity and improved matching.

- Power Efficiency:

Utilization of minimum-sized transistors in non-critical paths resulted in minimizing static and dynamic power consumption. Load capacitance optimization and node switching activity were major contributors to power-aware design.

- Timing Optimization:

Parasitic extraction enabled us to find critical paths in the design. We minimized delay in cells such as the priority encoder and D latch by minimizing fan-out and gate sizing optimization in time-critical paths.

- Cell Architecture Optimization

Logical effort and gate sizing methods were applied selectively to enhance drive strength and delay reduction. The D latch, for example, was optimized via pass-transistor logic to minimize transistor count and switching delay.

- Design Reusability and Modularity

Every cell was designed to be reusable to allow easy integration into a larger standard cell library. Similar design styles and pin positions were used to promote greater compatibility and ease of automated place and route.

Chapter 6

Conclusions and future scope

6.1 Conclusion

We successfully designed and implemented the physical designs of several logic gates such as NAND, NOR, XOR, XNOR, and sequential elements like the D Latch and a 4-bit Priority Encoder using typical VLSI design processes. Each of the designs was carefully checked for Design Rule Check (DRC) and Layout Versus Schematic (LVS) correctness to guarantee correctness and manufacturability. In addition, we derived the timing and functional parameters to create corresponding .lib files that are required for timing analysis and integration into digital design flows.
This project exposed us to the complete custom layout design process with hands-on experience, further understanding the physical and logical side of VLSI design. It also illustrated how essential precise standard cell design is for meeting performance, area, and power requirements in bigger digital systems.

6.2 Future scope

- Expansion of Standard Cell Library: The project can be expanded to incorporate more advanced gates and flip-flops to create a full custom standard cell library that can be used with automated synthesis tools.
- PVT Corner Characterization: The existing .lib files can be characterized further at numerous Process, Voltage, and Temperature (PVT) corners for strong performance under different conditions.
- Integration into Digital Flow: The developed cells can be integrated into a digital design flow with Cadence or Synopsys for ASIC prototyping, supporting end-to-end verification from RTL to GDSII.
- Power and Area Optimization: The future work can concentrate on power and area optimization for the smallest power consumption and area, which are crucial parameters in commercial IC designs.
- Automation: Leveraging the use of layout automation tools and Python scripting could cut design time quite dramatically and enhance repeatability for subsequent designs.

6.3 Bibliography

[1] N. H. E. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 5th ed. Pearson, 2022.

[2] R. J. Baker, CMOS: Circuit Design, Layout, and Simulation, 2nd ed. Wiley-IEEE Press, 2008.

[3] A. Chandrakasan, W. J. Bowhill, and F. Fox, "Design of high-performance microprocessor circuits," IEEE Press, 2001.

[4] Synopsys Inc., Liberty File Format (Version 2.0) Specification, Synopsys, 2020. [Available through Synopsys support or NDA license].

[5] Cadence Design Systems, Virtuoso Layout Suite User Guide, 2021. [Available from Cadence under academic license].

[6] C. Mead and L. Conway, "Introduction to VLSI Systems," Addison-Wesley, 1980. [Seminal work on structured VLSI design].

[7] S. M. Kang and Y. Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, 3rd ed. McGraw-Hill, 2003.

[8] Cadence Design Systems, Cadence Liberate Characterization User Guide, 2021. [Covers setup and use of Liberate for .lib file generation].