Complete ASIC Flow of I2C communication protocol

Group Name: Section 16 - CND211_Group 4

Supervised by

CND Team

Passionately Implemented by

Student Name	ID
Fayza Ahmed Sayed Ahmed	V23010517
Amr Samy Mohamed	V23010343
Omar Mohamed Shabaan	V23010277
Eslam Hussien Ahmed Solyman	V23010203

Fall 2024

Table of Contents

Table of Figures

Introduction

The I2C (Inter-Integrated Circuit) protocol is renowned for its bi-directional serial bus, offering a straightforward and effective means of data exchange between devices. Its suitability for applications requiring periodic communication over short distances, especially among multiple devices, has made it a go-to choose in various industries. Notably, the I2C standard is a true multi-master bus, integrating collision detection and arbitration to prevent data corruption when multiple masters attempt to control the bus simultaneously.

In this ASIC flow project, the primary objective is to navigate through the process of creating the complete layout (GDS) of an I2C, a two-wire interface, utilizing the provided Process Design Kit (PDK) under nominal supply conditions. It's important to note that the RTL (Register-Transfer Level) for the I2C has already been provided, and the focus of the project is specifically on the ASIC flow.

The tasks involved in this project encompass the synthesis of the provided RTL code, formal verification, static timing analysis, creation of the floorplan and power plan, clock tree synthesis, placing and routing the core, and the design and layout of the pad ring with all necessary inputs and outputs. The final stage of the project involves signoff and the completion of the final layout.

Given that the RTL for the I2C is provided, the project's emphasis on the ASIC flow underscores the essential steps and considerations involved in the physical design and implementation of the I2C protocol within an integrated circuit. This project presents a valuable opportunity to gain practical experience in ASIC flow, layout design, and the integration of a well-established communication protocol into the realm of electronic design.

The I2C protocol's distinct features, including its bi-directional nature, minimal pin count, and support for multi-master bus systems, have secured its widespread adoption across diverse sectors, including consumer electronics, IoT, automotive, aerospace, and industrial equipment.

This report will explore the intricacies of the I2C protocol, the specific ASIC flow tasks involved in the project, and the broader significance of this endeavor within the domain of modern electronic design and communication protocols.

Overview

This project is dedicated to the design and realization of a complete layout (GDS) of the I2C communication protocol using SAED90_EDK. I2C, known as Inter-Integrated Circuit, stands as a bi-directional serial bus solution facilitating efficient data exchange between interconnected devices. Its versatility shines in scenarios where intermittent communication over short distances among multiple devices is essential.

Key Features:

- **Efficiency and Simplicity:** I2C is renowned for its straightforwardness and effectiveness, positioning it as a favored option for a wide array of embedded systems.
- **Multi-Master Support:** The protocol boasts multi-master functionality, allowing multiple devices to seamlessly communicate on the same bus. This is complemented by collision detection and arbitration mechanisms, safeguarding against data corruption during simultaneous bus access by multiple masters.
- **Utilization of SAED90_EDK:** The project harnesses the capabilities of SAED90_EDK throughout the ASIC flow, spanning RTL design, synthesis, place and route, and layout generation phases.
- **Comprehensive Learning Experience:** Participants will develop a profound understanding of the I2C protocol intricacies and ASIC design flow nuances, laying a robust foundation for crafting efficient communication interfaces in future semiconductor endeavors.

Through this project, participants will develop a comprehensive understanding of the I2C protocol and ASIC design flow, laying the foundation for creating robust and efficient communication interfaces in future semiconductor designs. This endeavor not only offers practical experience in ASIC design but also contributes to enhancing knowledge and skills in implementing industry-standard communication protocols for embedded systems.

In addition to its technical focus, this project leverages an extensive suite of Electronic Design Automation (EDA) tools. These include Synopsys Design Compiler (DC) for logic synthesis, Synopsys Formality for formal verification, Synopsys IC Compiler II Library Manager for data setup and dlib creation, Synopsys IC Compiler II for back-end flow, and Synopsys PrimeTime for static timing analysis. Such tools ensure meticulous execution and validation of the ASIC design process, reinforcing the project's commitment to excellence in semiconductor development.

ASIC Flow

The ASIC flow for the design and implementation of the I2C communication protocol using SAED90_EDK involves several key stages, each essential for ensuring the successful integration and functionality of the protocol within the semiconductor design process.

Synthesis of the Code:

The initial stage involves the synthesis of the provided RTL code using tools such as Synopsys Design Compiler (DC) to convert the RTL description of the I2C protocol into a gate-level netlist suitable for subsequent stages of the ASIC flow.

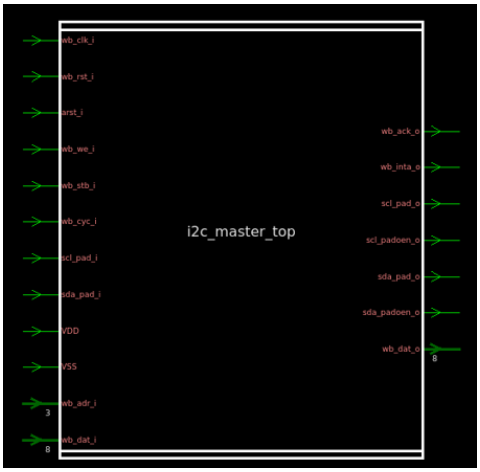

Figure 1: Synthesized View of I2C chip

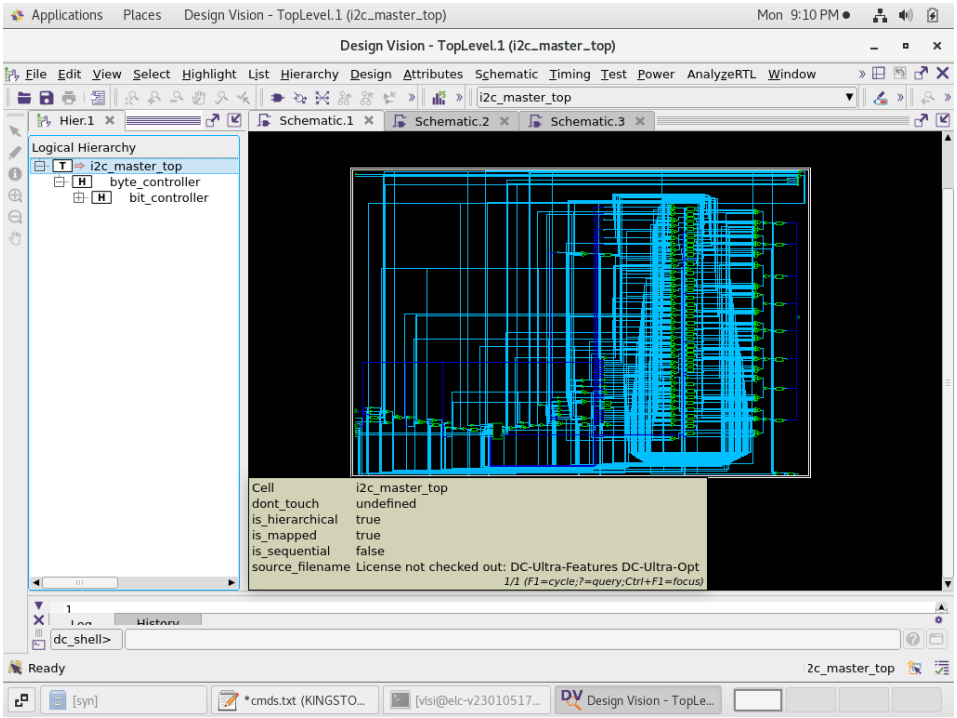

Figure 2: Schematic view of I2C Synthesis Netlist.

Here is the synthesis for the byte controller block:

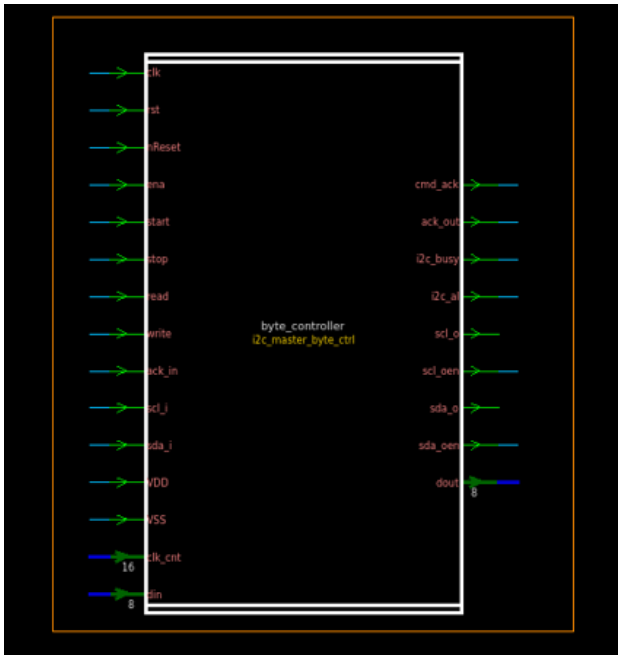

Figure 3: Synthesized View of byte_controller block.

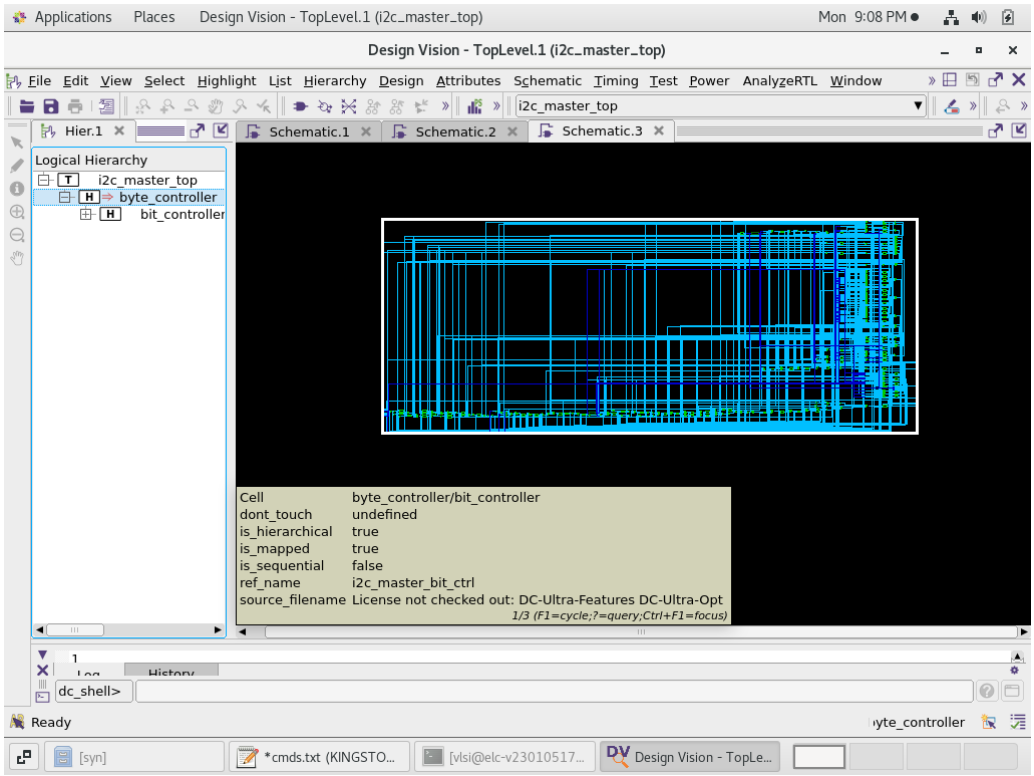

Figure 4: Schematic view of byte_controller Synthesis Netlist.

Here is the synthesis for the bit controller block:

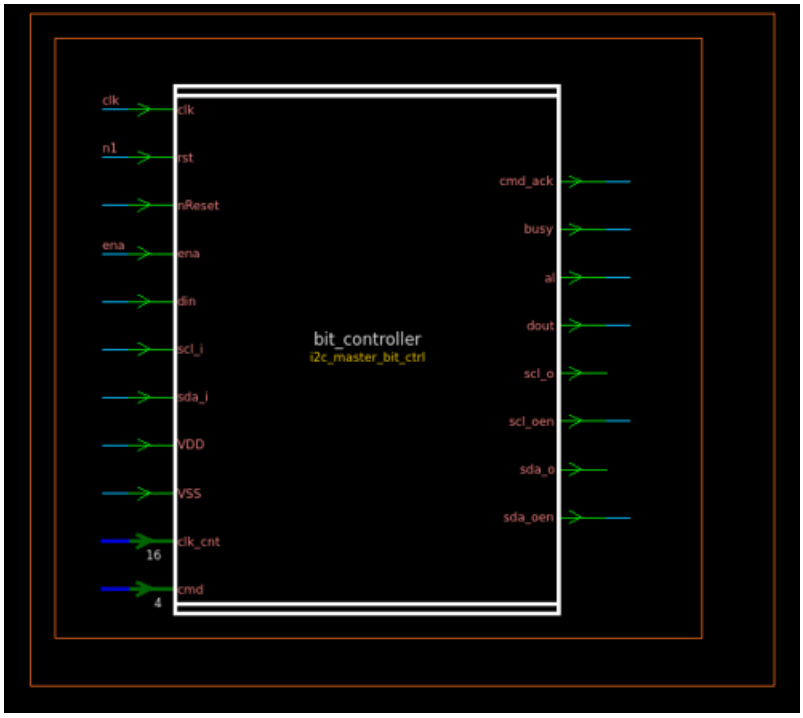

Figure 5: Synthesized View of bit_controller block.

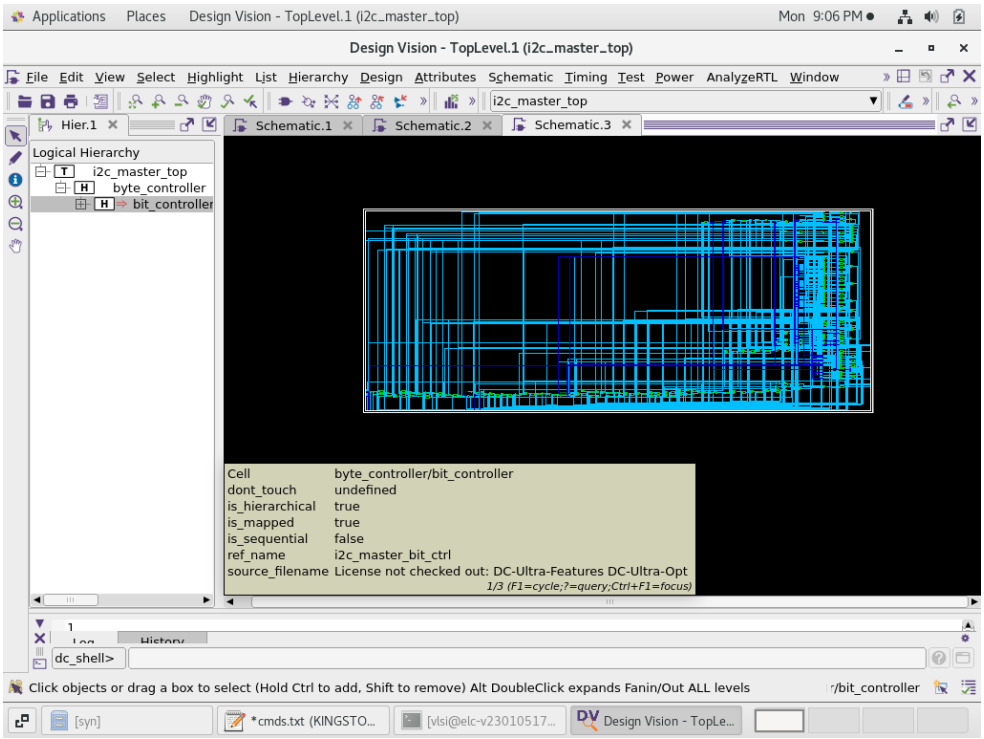

Figure 6: Schematic view of bit_controller Synthesis Netlist.

Performing Static Timing Analysis:

Static timing analysis is performed to assess the timing characteristics of the design, ensuring that the I2C protocol meets the required timing constraints. Performing Static Timing Analysis (STA) is a crucial step in the design process to ensure that the timing characteristics of the design meet the required constraints. In this case, the STA is conducted to assess the timing characteristics of the I2C protocol. The analysis includes running several reports to evaluate different aspects of the design. These reports are generated as follows:

- **report_area:** ./report/synth_area.rpt
- **report_cell:** ./report/synth_cells.rpt
- **report_qor:** ./report/synth_qor.rpt
- **report_resources:** ./report/synth_resources.rpt
- **report_timing -max_paths 10:** ./report/synth_timing.rpt

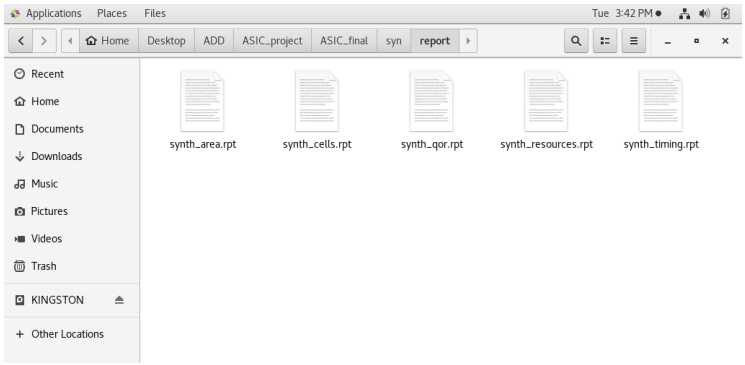

Figure 7: Reports after synthesis.

After running these reports, it is found that all 10 timing paths meet the required constraints, and the slack is positive.

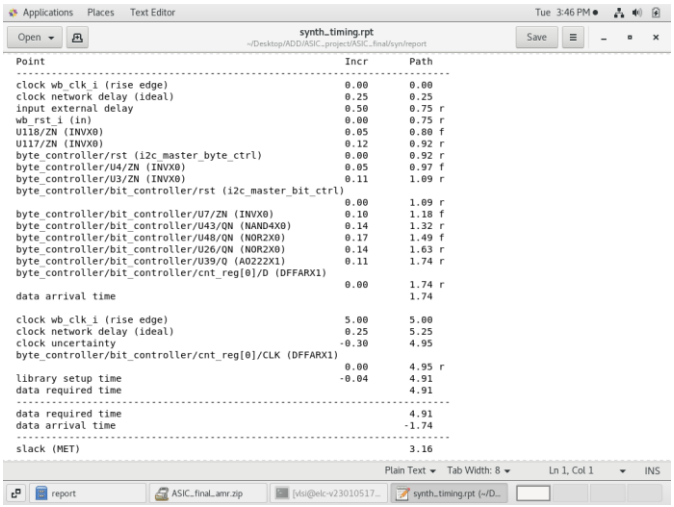

Figure 8: STA report snippet

This comprehensive analysis ensures that the design not only meets the timing requirements but also provides insights into the area, quality of results (QoR), and resource utilization, which are essential for optimizing the overall design.

Formal Verification:

Formal verification is conducted to rigorously validate the correctness of the design and ensure that it adheres to the specified properties and requirements. Tools like Synopsys Formality are employed for this critical verification process. We followed a systematic approach to formal verification, adhering to the steps recommended by Synopsys Formality:

- ✓ **Guide:** We started by selecting the appropriate setup verification file (default.svf) in the synthesis directory, providing the necessary configuration for the verification process.
- ✓ **Refinement (Ref):** All RTL files pertaining to the design were included, ensuring that the complete design description was available for analysis.
- ✓ **Implementation (Impl):** We read the design files and database libraries, setting up the top-level design correctly to establish the environment for verification.
- ✓ **Setup:** This involved configuring the tool and specifying the properties and requirements to be verified, including constraints and design specifications.
- ✓ **Matching (Match):** Formality analyzed the design against its refined version, ensuring functional equivalence and consistency between the RTL and gate-level representations.

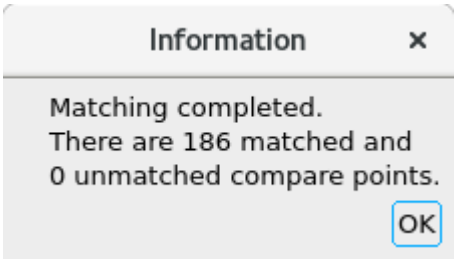

Figure 9: Formality Matching step succeeded!

- ✓ **Verification (Verify):** The tool rigorously verified the design against the specified properties, thoroughly checking for any violations or discrepancies.

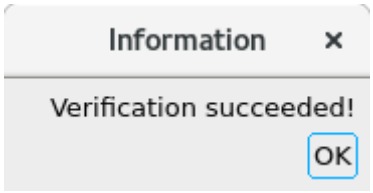

Figure 10: Formality Verification step succeeded!

- ✓ **Debug:** In case of any issues or failures encountered during verification, we engaged in debugging activities to identify and resolve the underlying causes, ensuring the correctness of the design.

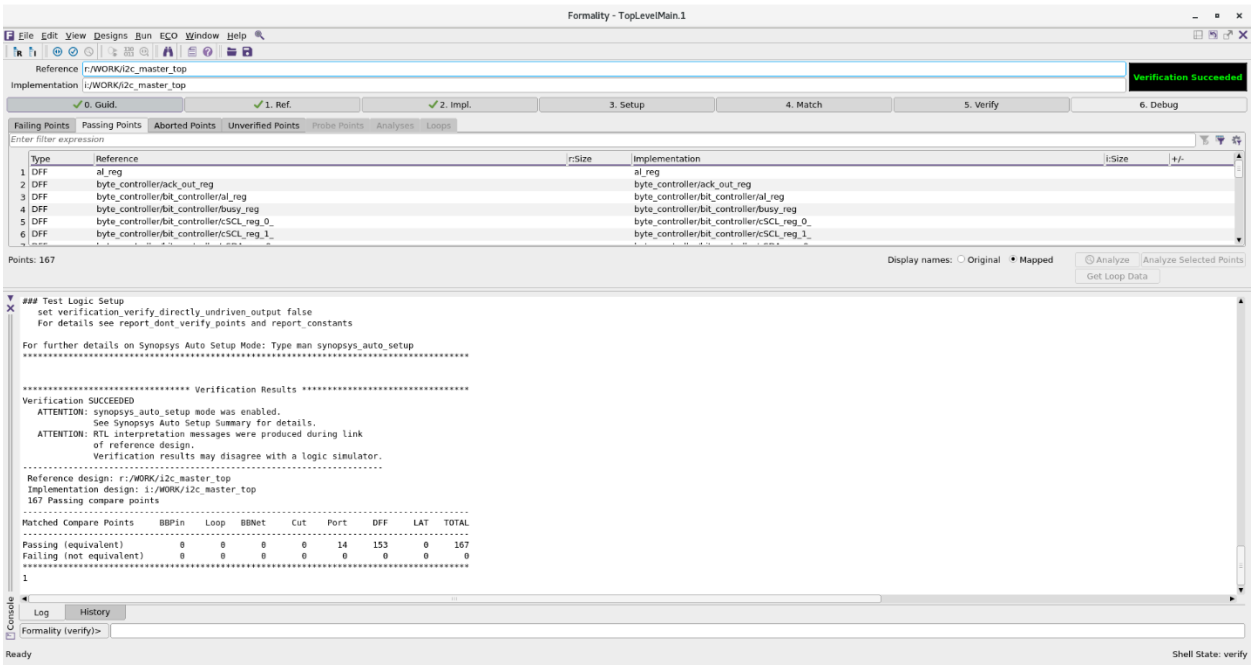

Figure 11: Formal verification Succeeded!

NDM creation and Data setup:

Data setup involves preparing the design data and configuring the environment for subsequent physical design steps. This includes organizing input files, such as netlists and libraries, and setting design parameters and constraints.

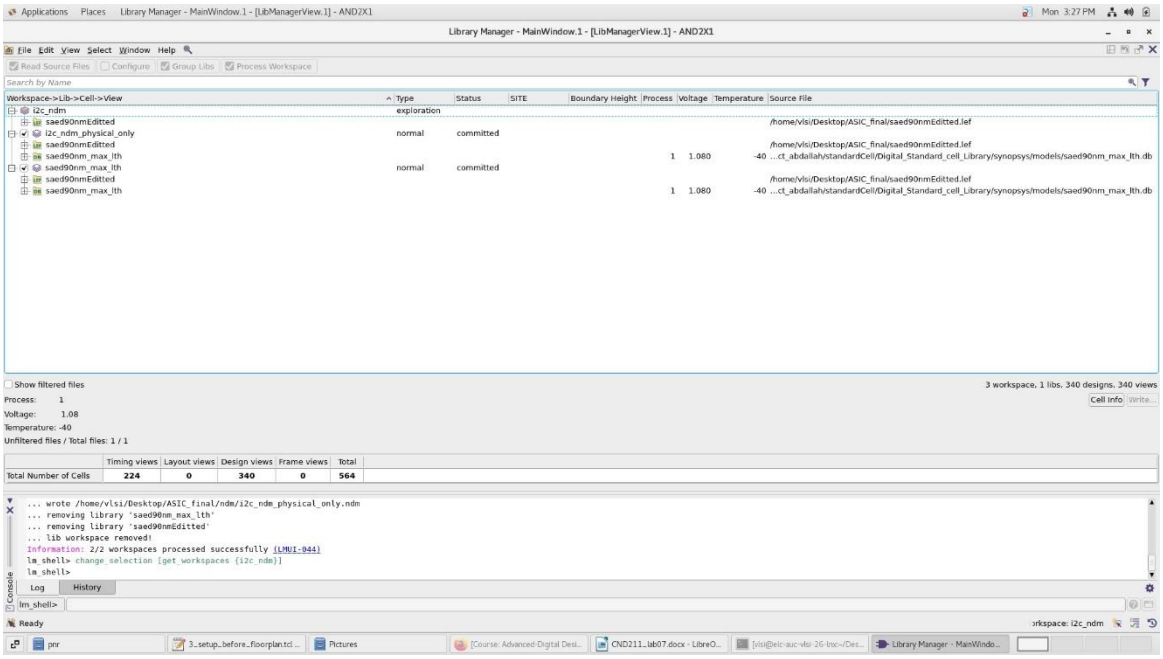

Figure 12: ndm created successfully!

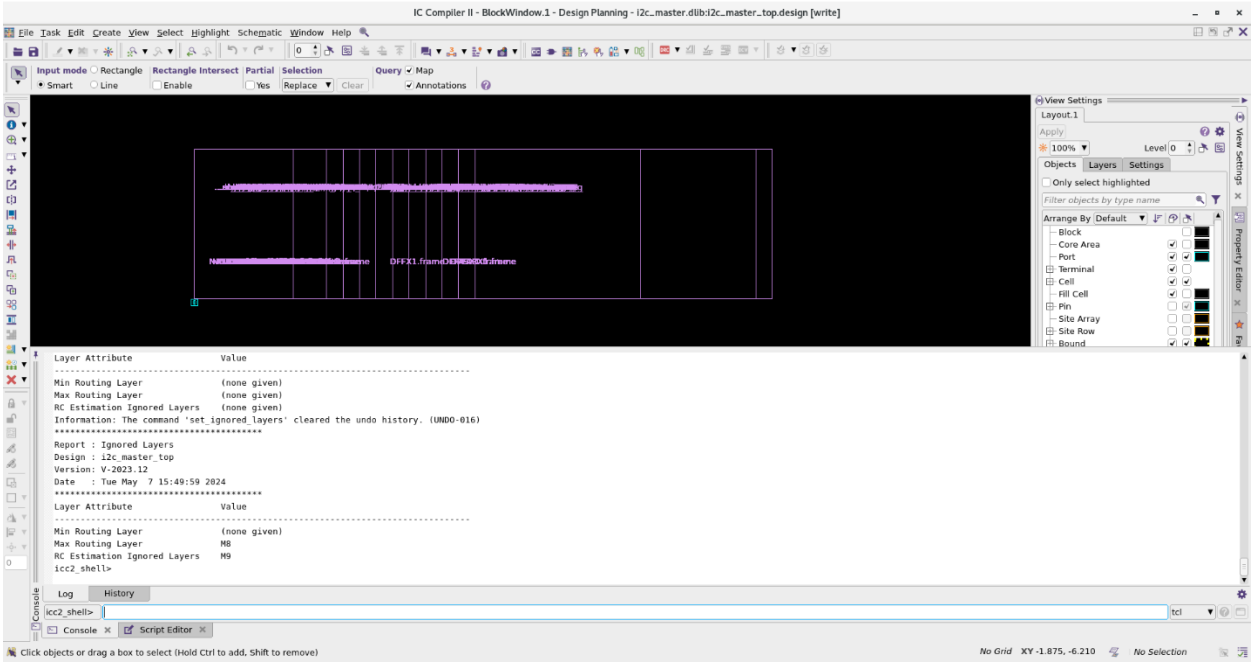

Figure 13: After running Setup script.

Creating the Floorplan:

Floor planning is an essential phase in the physical design of integrated circuits (ICs), involving three main steps. First, the floor plan is initialized, defining the chip's physical boundaries and size. Next, placement is determined, deciding where each functional block should go within the chip area to optimize performance. Finally, pin constraints are set to specify the locations and characteristics of input/output connections, ensuring proper connectivity with external devices. Through careful floor planning, designers create an organized layout that meets performance, power, and manufacturing requirements effectively.

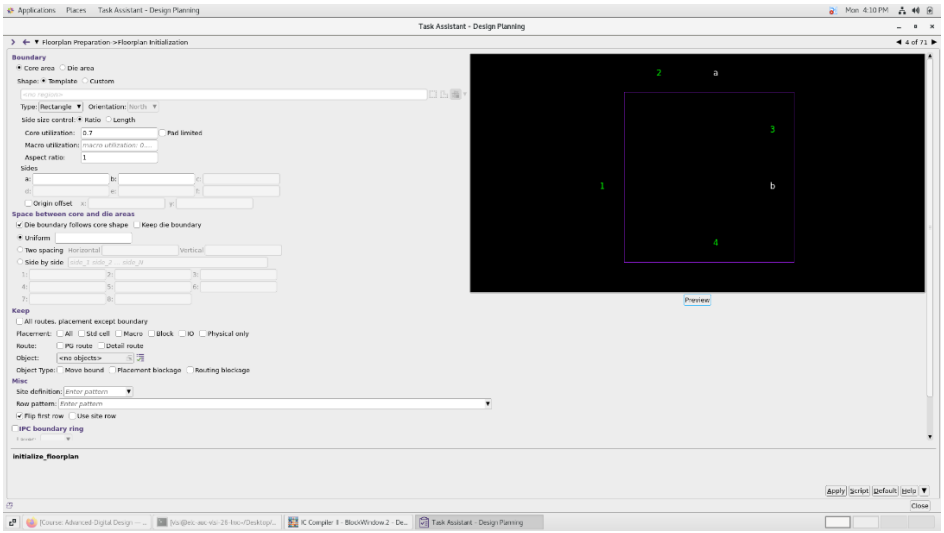

Figure 14: Initialize floor plan.

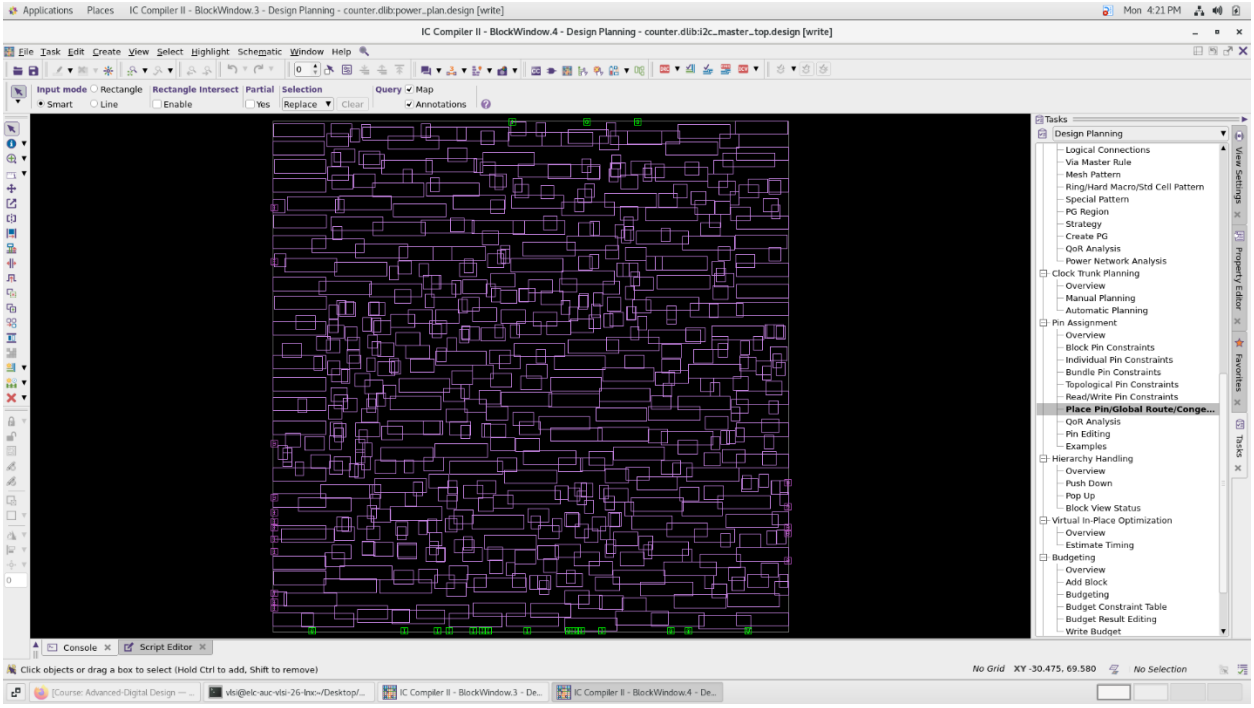

Figure 15: Initial Placement and Pins Constrains

Creating the Power Plan:

Power planning is a critical aspect of integrated circuit (IC) design, focusing on the efficient distribution and management of power throughout the chip. In power planning, various techniques are employed to ensure stable and reliable power delivery to all parts of the chip. One commonly used method is the implementation of power rings, which encircle the chip's periphery and serve as a primary power distribution network.

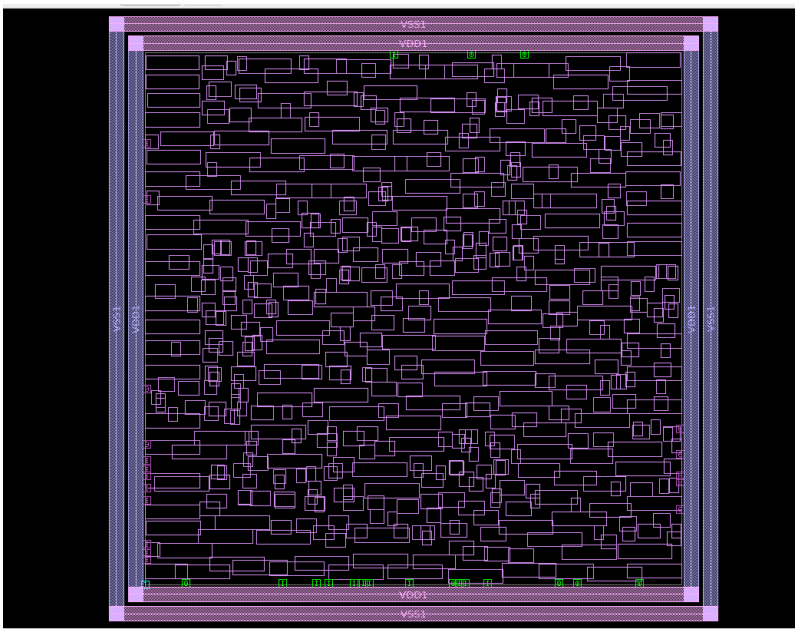

Figure 16: Power Ring.

Another approach involves utilizing power meshes, which are distributed across the chip's interior to provide uniform power distribution.

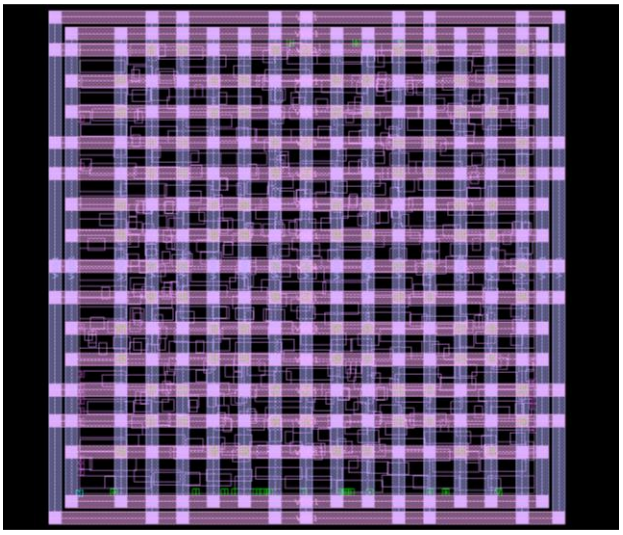

Figure 17: Power Mesh.

Rail planning is also essential, ensuring that power supply rails are strategically placed and sized to meet the dynamic power requirements of different chip regions.

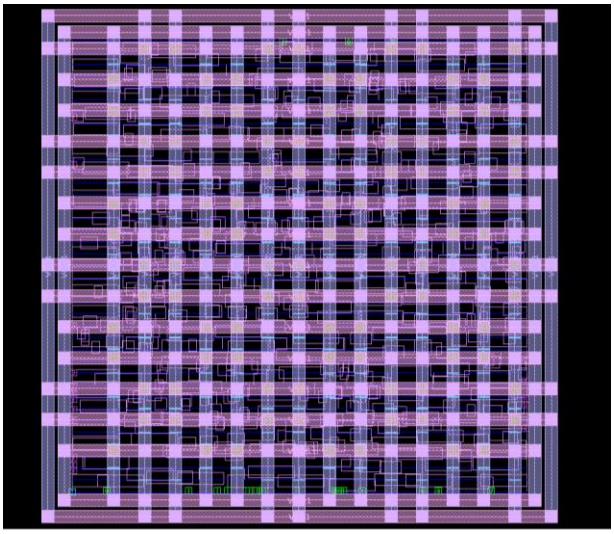

Figure 18: Rail planning.

Moreover, addressing **IR drop** is crucial to mitigate voltage variations caused by resistive losses in power distribution networks.

Pin density and ***cell density*** considerations play a significant role in power planning as well, determining the distribution of power and ground pins across the chip and optimizing the placement of logic cells to minimize power consumption and maximize performance. Through meticulous power planning, designers can achieve efficient power delivery, reduce IR drop effects, and enhance overall chip performance and reliability.

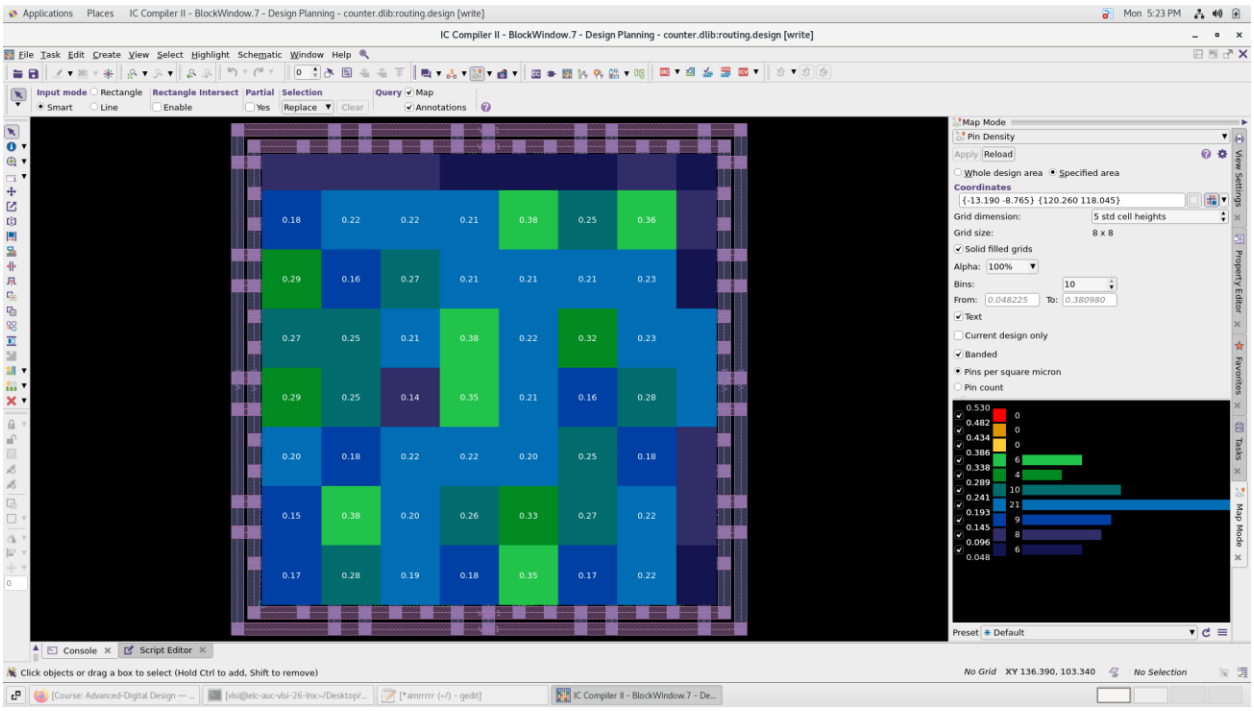

Figure 19: Pin density.

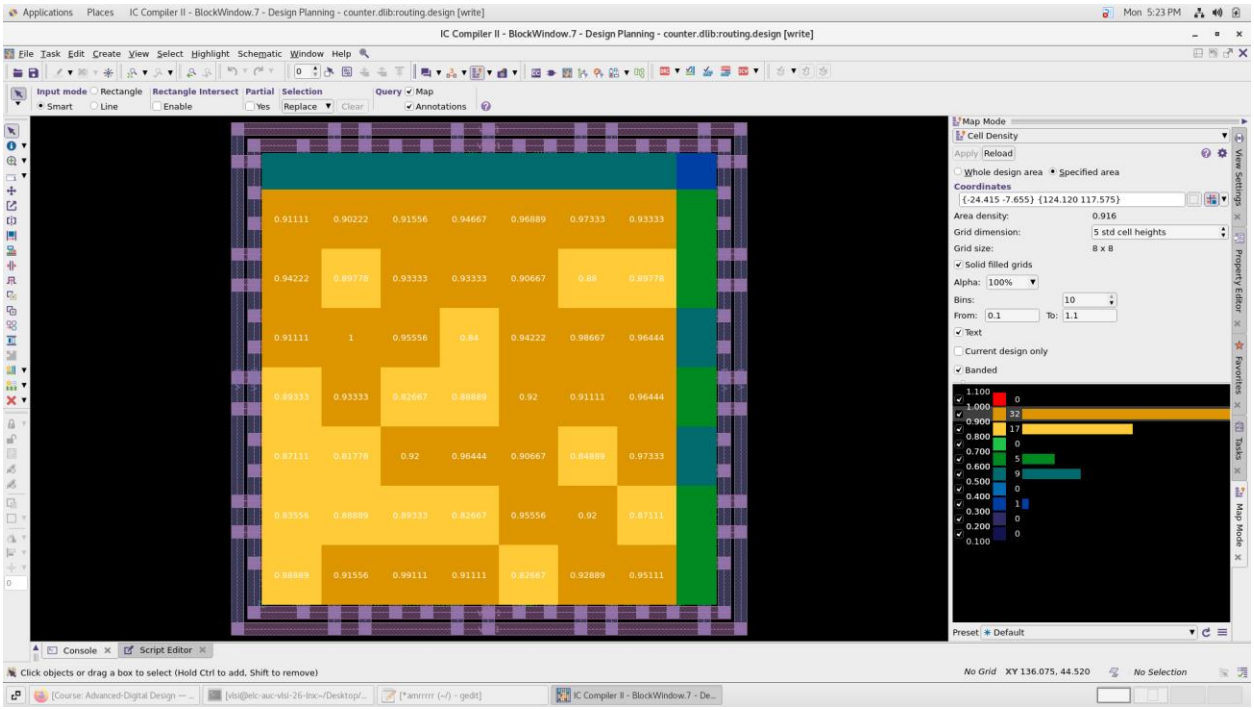

Figure 20: Cell density.

Performing Clock Tree Synthesis:

Clock tree synthesis is a crucial step in ensuring the efficient and reliable distribution of clock signals throughout the design. Through tools like Synopsys IC Compiler II, the generation of an optimized clock distribution network is carried out to minimize clock skew and attain desirable clock characteristics.

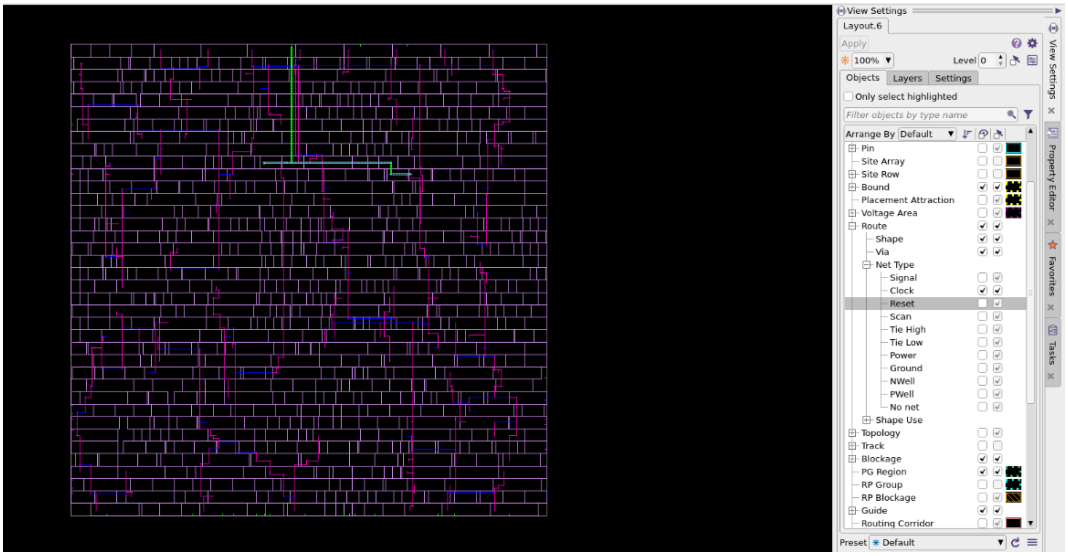

Figure 21: Clock Tree Synthesis.

Routing:

The I2C protocol is routed within the chip, employing advanced route methodologies to optimize performance, minimize signal delays, and adhere to design rules and constraints.

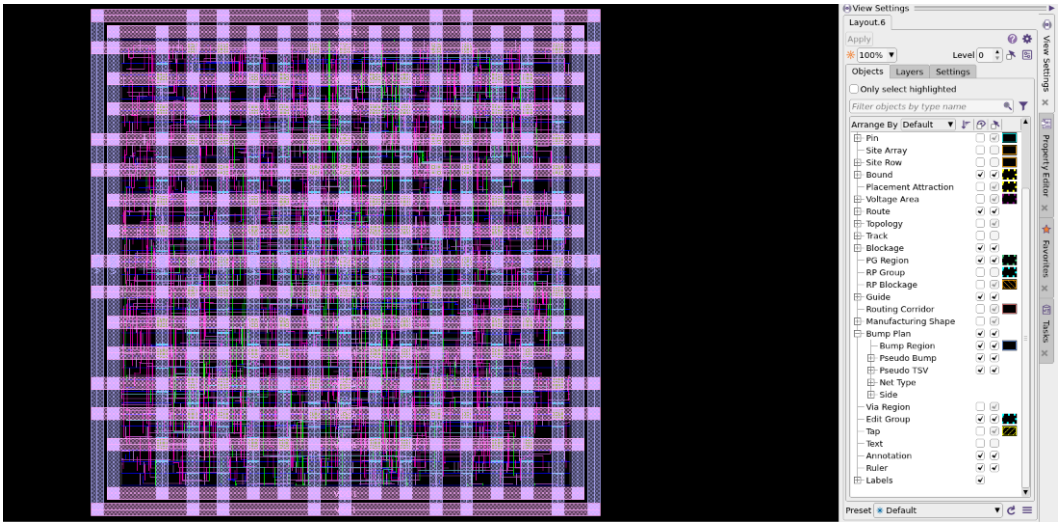

Figure 22: Routing.

Prime Time validation:

PrimeTime is a static timing analysis tool used in digital design to verify that timing requirements are met. It analyzes timing paths, identifies critical paths, and detects potential timing violations. PrimeTime provides detailed timing reports, facilitating design refinement and ensuring design reliability and performance.

Initially, we encountered hold violations, which we subsequently addressed by inserting buffers to resolve the issue.

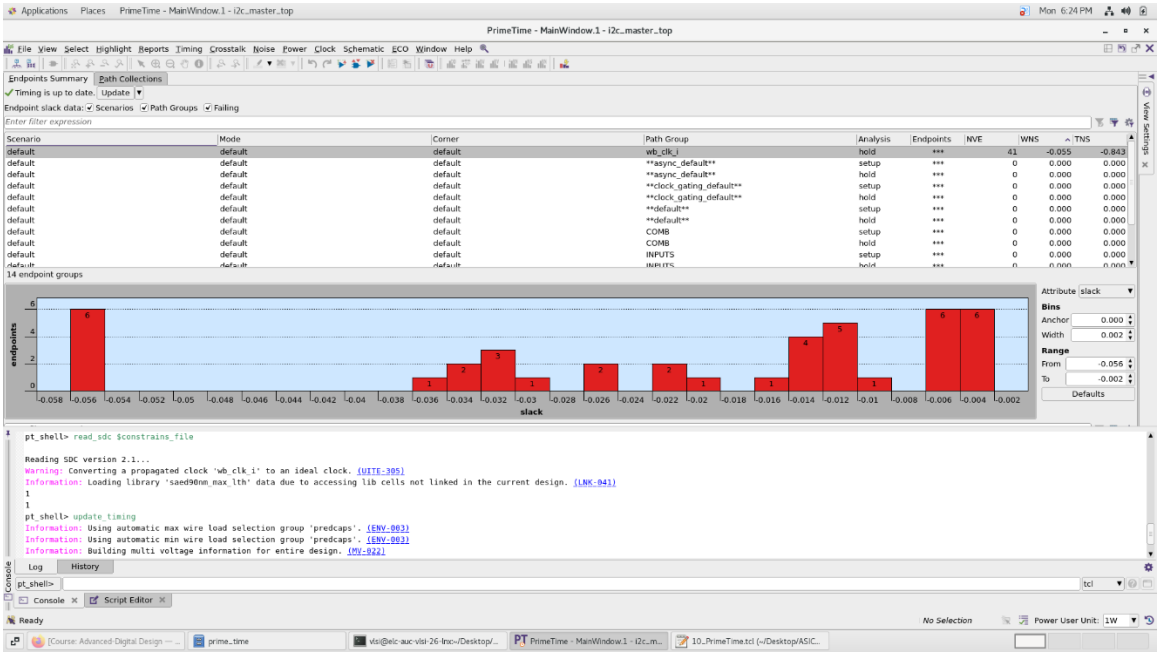

Figure 23: Prime Time before fixing Hold violations.

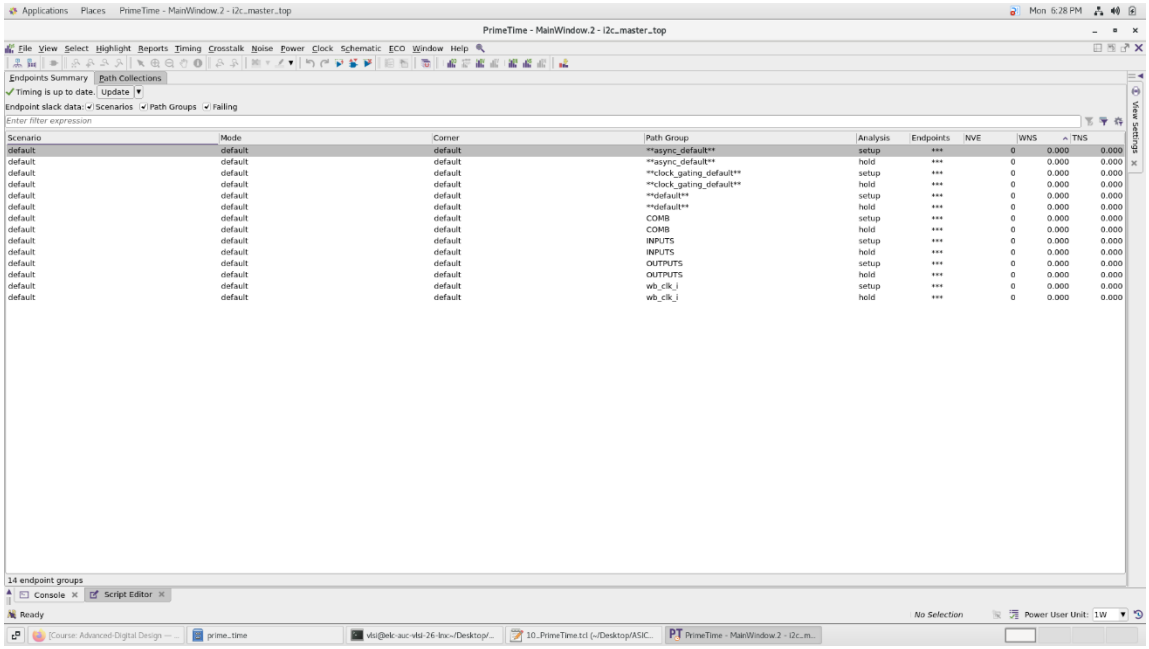

Figure 24: Prime Time after fixing Hold violations.

Final View:

This is the final view after inserting the buffers using that command:

```
fix_eco_timing -type hold -methods insert_buffer -buffer_list {NBUFFX1 NBUFFX2 NBUFFX3}
```

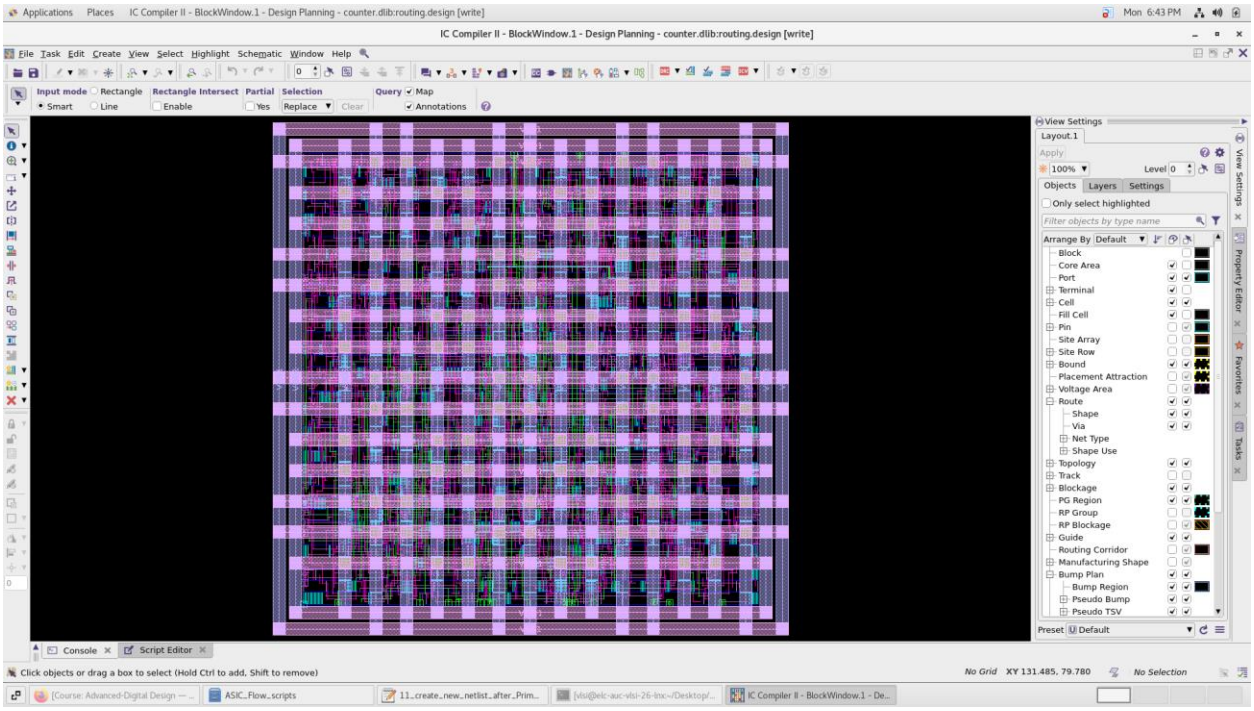

Figure 25: Final View

This comprehensive ASIC flow ensures that the I2C communication protocol is meticulously designed, verified, and integrated into the semiconductor design, setting the stage for its successful deployment in diverse electronic systems.

Conclusion

Our journey through the ASIC flow has revealed the intricate process of turning an idea into a real chip. From RTL design to synthesis, and from floor planning to routing and signoff, each step played a vital part in creating a flawless design that meets strict power, area, and timing requirements. Ultimately, our experience with the I2C protocol in the ASIC flow showcases the seamless integration of communication protocols with digital design methods, paving the way for modern, efficient, and high-performance integrated circuits, driving innovation in the field of electronics.