RTL TO GDSII USING SYNOPSYS TOOLS

A training program report submitted in partial fulfilment
of the requirements for the Award of
Degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

CHANDRAKIRAN G (312422106032)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
St. JOSEPH'S INSTITUTE OF TECHNOLOGY
(An Autonomous Institution)
OMR, CHENNAI – 600 119

2022 – 2026

BONAFIDE CERTIFICATE

Certified that the Training Program report titled “RTL to GDSII Using Synopsys Tools” is the bona fide work of “CHANDRAKIRAN G” (312422106032),who completed the training program under my supervision.

SIGNATURE
Dr. P. LATHA, M.E., Ph.D..,
SUPERVISOR,
Associate Professor,
Electronics and Communication Engineering,
St. Joseph’s College of Engineering,
Old Mamallapuram Road,
Chennai – 600 119.

SIGNATURE
Dr. P.G.V. RAMESH , M.Tech., Ph.D..,
HEAD OF THE DEPARTMENT,
Professor,
Electronics and Communication Engineering,
St.Joseph’s Institute of Technology,
Old Mamallapuram Road,
Chennai – 600 119.

ACKNOWLEDGEMENT

At the outset, I would like to express my sincere gratitude to our beloved ***Chairman Dr. B. BABU MANOHARAN, M.A., M.B.A., Ph.D.***, for his constant guidance and support.

I would like to express my sincere thanks to our respected ***Executive Director, Mrs. S. JESSIE PRIYA, M. Com.***, and our ***Managing Director, Mr. B. SASHISEKAR, M.Sc*** for their kind encouragement and blessings.

I express my sincere gratitude and wholehearted thanks to our ***Principal Dr. S. ARIVAZHAGAN, M. E., Ph.D.***, for his encouragement to make this Training a successful one.

I wish to express our truthful thanks and gratitude to our ***Head of the Department Dr. P G V RAMESH, M.Tech., Ph.D.***, Department of Electronics and Communication Engineering for his constant support and motivation throughout my internship.

I have no words to express my heartfelt thanks to ***Dr. P. LATHA , M.E., Ph.D.., Associate Professor*** for her guidance, support and encouragement in completing the Training and Project within the stipulated time

(Chandrakiran G)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The project demonstrates the design of an 8-bit adder using 1-bit full adders, with the entire design flow spanning from RTL (Register Transfer Level) to GDSII. The design utilizes individual 1-bit full adders to construct the 8-bit adder, ensuring scalability and efficiency. The RTL code was written in VHDL/Verilog, and extensive testbenches were created to verify the functionality of both the full adders and the overall 8-bit adder. Verification covered functional correctness, edge cases, and timing analysis.

Following RTL verification, the design was synthesized using a logic synthesis tool, optimizing for area, power, and timing. The next steps involved creating the floor plan, where placement and power grid planning were crucial to ensure efficient routing. A detailed power plan was implemented to manage both dynamic and static power consumption. Clock Tree Synthesis (CTS) was performed to minimize skew and ensure optimal clock distribution.

Routing was then completed to finalize signal paths, ensuring that all design constraints were met. Basic Static Timing Analysis (STA) was conducted to ensure that the design met timing requirements, followed by comprehensive reporting. Finally, the GDSII file was generated, providing the final layout for manufacturing. The project demonstrates the full ASIC design flow from RTL to physical implementation.

1. INTRODUCTION

What is VLSI?

VLSI, or Very Large-Scale Integration, is a cutting-edge technology in the field of electronics that involves the design and fabrication of integrated circuits (ICs) by integrating millions or even billions of transistors onto a single silicon chip. These chips serve as the "brains" of electronic devices, enabling a wide range of functionalities from complex calculations to high-speed data processing. **The Revolution of VLSI Technology**

VLSI technology has transformed the way we create electronic circuits. Traditionally, electronic devices were built using bulky circuits composed of individual components like resistors, capacitors, and discrete transistors. VLSI replaces these cumbersome designs with highly compact and efficient chips, often no larger than a coin. This miniaturization has led to the development of powerful, portable, and energy-efficient devices that are integral to modern life.

Why VLSI is Important

Miniaturization of Devices:

Impact: VLSI allows for the creation of incredibly small and compact electronic devices. This miniaturization is what makes modern gadgets like smartphones, laptops, and smartwatches possible.

Example: A smartphone today has more computing power than the room-sized computers of the past, all thanks to VLSI technology.

Improved Performance:

Impact: By integrating more transistors onto a single chip, VLSI enhances the performance of electronic devices. This means faster processing speeds, better data handling, and overall improved functionality.

Example: High-performance gaming consoles and graphics cards rely on VLSI to deliver seamless and immersive gaming experiences.

Reduced Power Consumption:

Impact: VLSI chips are designed to be energy-efficient, reducing the power consumption of electronic devices. This makes devices not only faster but also more environmentally friendly.

Example: Energy-efficient processors in laptops and smartphones extend battery life, allowing users to stay connected for longer periods.

Cost Efficiency:

Impact: The mass production of VLSI chips reduces manufacturing costs significantly. This cost efficiency makes advanced electronics more affordable for consumers.

Example: The affordability of smartphones and other consumer electronics is largely due to the costeffective production of VLSI chips.

Key Steps in the VLSI Design Process Specification:

Purpose: Engineers define the functional requirements of the chip. This includes deciding what the chip should do, such as powering a smartphone, processing video, or managing data in a server.

Output: A detailed specification document outlining the chip's intended capabilities and performance metrics.

Design:

Purpose: A blueprint of the chip is created using specialized software tools. This design phase includes both the behavioural and structural aspects of the chip.

Process: Engineers use hardware description languages (HDLs) like Verilog or VHDL to describe the chip's functionality. This high-level design is then synthesized into a gate-level netlist.

Output: A comprehensive design that includes the layout and interconnections of all components on the chip.

Fabrication:

Purpose: The chip is physically built in a fabrication facility, also known as a foundry. This process involves layering and connecting millions of tiny components to create a functional integrated circuit.

Process: The design is translated into a series of photolithographic masks, which are used to pattern the layers of the chip. Advanced manufacturing techniques ensure precision and accuracy.

Output: A fully fabricated chip ready for testing and integration into electronic devices.

Testing:

Purpose: The final chip is rigorously tested to ensure it meets the specified performance and reliability standards. This includes functional testing, electrical testing, and environmental testing.

Process: Automated test equipment (ATE) is used to simulate various operating conditions and verify the chip's functionality.

Output: A certified chip that is ready for mass production and integration into end-user devices. Applications of VLSI

VLSI chips are used in a wide range of devices and industries, such as

1. Consumer Electronics:

- Smartphones (processors, memory, and sensors)
- Laptops and PCs (CPUs, GPUs, and RAM)
- Smart TVs and home assistants

2. Automotive Industry:

- Advanced driver-assistance systems (ADAS)
- Engine control units (ECUs) for efficient vehicle performance
- Electric vehicles (EVs) for battery management

3. Medical Devices:

- Wearable health monitors (like fitness trackers)
- Imaging systems (like MRI machines)
- Portable diagnostic tools

4. Aerospace and Defence:

- Satellite communication systems
- Radar and navigation systems
- High-speed data processing for military applications

5. Communication Networks:

- 5G and wireless communication infrastructure
- Routers and modems for internet connectivity

Conclusion

VLSI is at the heart of modern technology, powering the devices and systems we use daily. Its ability to integrate countless functions into a small chip makes it indispensable for innovation. From healthcare to entertainment, VLSI enables industries to thrive and evolve, pushing the boundaries of what's possible.

2. RTL2GDSII FLOW 2.1 Introduction

The entire Chip life cycle can be characterized by a few key stages that are briefly discussed below. The focus of this report will be the IC design stage

2.1.1 IC Design:

IC design is where the journey begins. It's the process of conceptualizing, defining, and creating the blueprint for an integrated circuit. It can be broadly divided into:

- **System-Level Design**: Translating user requirements and device functionality into system specifications.
- **RTL Design (Digital)**: Using hardware description languages like Verilog or VHDL to describe the logical behaviour.
- **Analog/Custom Design**: Creating precise transistor-level schematics for analog and mixedsignal components.
- **Verification and Validation**: Ensuring correctness through simulation, synthesis, and formal verification.
- **Physical Design**: Transforming logical designs into layouts for fabrication, including placement, routing, and timing closure.

This step combines creativity and engineering precision to design ICs that are smaller, faster, and energy efficient.

2.1.2 IC Manufacturing:

Once the design is finalized, the IC undergoes fabrication in highly specialized semiconductor foundries. This process includes:

- **Wafer Fabrication**: Creating wafers with multiple chips using photolithography, doping, etching, and deposition techniques.
- **Material Science**: Using advanced materials like silicon, gallium arsenide, or even emerging technologies like graphene.
- **Moore's Law in Action**: Shrinking transistor sizes for each generation to pack billions of transistors onto a single chip.
- **Packaging**: Encasing the fabricated chip in protective materials, ensuring electrical connections and heat dissipation.

This is a highly capital-intensive stage, requiring precision and expertise to achieve high yield and quality.

2.1.3 Post Tape-out IC Testing and Validation:

After tape-out (finalizing the layout for fabrication), rigorous testing and validation are performed to ensure the IC functions as intended:

- **Wafer Testing**: Probing individual chips on the wafer to test functionality before cutting them.
- **Package-Level Testing**: Checking electrical, thermal, and mechanical properties after packaging.
- **Burn-In Testing**: Running the chip at elevated temperatures and voltages to screen for early-life failures.
- **Field Tests and Debugging**: Evaluating performance in real-world conditions and fine-tuning as needed.

This stage ensures the IC is reliable and meets the required performance standards before it enters the market.

2.1.4 Integration into Devices:

Once validated, these ICs are integrated into smart devices such as smartphones, IoT devices, and autonomous vehicles. The entire process—from design to post-tape-out testing—is a marvel of modern engineering, pushing the boundaries of what's possible in technology.

The IC Design is the focus of the work we are concerned with here. To begin, we first understand the classification of IC design flows.

1. RTL to GDSII Flow
2. Custom / AMS Design flow

2.2 RTL2GDSII Design Flow:

Of the two available design flows, the RTL2GDSII is the most preferred for reasons mentioned below. While RTL2GDSII is the most preferred, it is by no means a sole option. In fact, any design of an IC can be achieved by any design flow, owning to their own caveats.

Benefits of RTL2GDSII over other methods:

Automation: The RTL-to-GDSII flow is highly automated, leveraging advanced tools for synthesis, placement, routing, timing analysis, and more. This reduces design time and effort, making it ideal for large-scale digital designs.

Scalability: Modern digital chips, like microprocessors and GPUs, contain billions of transistors. Managing such complexity at the transistor level would be impractical. RTL-to-GDSII enables designers to handle this complexity efficiently.

Time **Efficiency**: Since the process is automated, the turnaround time from specification to fabrication is much shorter than the manual, detail-oriented custom IC flow.

Reuse of IPs: The RTL-to-GDSII flow allows for easy reuse of pre-designed intellectual property (IP) blocks, further accelerating the design process.

Cost-Effectiveness: For digital ICs, the automated nature of RTL-to-GDSII reduces the need for manual intervention, which can lower costs in terms of labour and time.

Focus on Functionality: Designers can focus more on defining the functionality using RTL (Register Transfer Level) without getting into the nitty-gritty of transistor-level design.

The RTL2GDSII Design flow itself is further divided into many sub-categories and each one of the cans be as vast as an individual domain.

At high-level, it is divided into two distinct categories,

- Front End Design
- Back End Design

2.2.1 Front-End Design

Front-End Design involves defining the functionality and behaviour of the integrated circuit. It is focused on the logical and architectural aspects of the design. Here's a deeper look at its key components:

1. **System-Level Design:**
    - High-level architecture and specification of the IC based on its intended application. ○ Defines how different components (e.g., processors, memory, communication interfaces) interact with each other.
2. **RTL Design (Register Transfer Level):**
    - The functionality of the chip is described using hardware description languages (HDLs) such as Verilog or VHDL. ○ The RTL code defines how data flows between registers and how operations are performed.
3. **Functional Verification:**
    - Ensures that the RTL design works correctly according to the system-level specifications. ○ Techniques include simulation, formal verification, and emulation to identify bugs early in the process.
4. **Logic Synthesis:**
    - The RTL code is converted into a gate-level netlist using logic synthesis tools.
    - Maps the high-level design to basic logic gates while optimizing for power, performance, and area (PPA).
5. **Static Timing Analysis (STA):**
    - Verifies that the design meets timing requirements (setup and hold times). ○ Ensures the design operates reliably at the target clock frequency.

6. **Design for Testability (DFT):**
    - Adds test structures (e.g., scan chains, Built-In Self-Test) to the design to facilitate testing during manufacturing.
    - Ensures defects in the chip can be identified efficiently post-manufacturing.

2.2.2 Back-End Design

Back-End Design focuses on the physical implementation of the IC. It involves translating the logical design (netlist) from the Front-End stage into a physical layout that can be fabricated. Here's a detailed breakdown of this crucial stage:

1. **Floor planning:**

- This is the initial step where the chip's overall layout and organization are determined.
- The locations of major components (macros, standard cells, I/O pads, etc.) are defined based on connectivity and performance.
- Objectives include minimizing wire length, optimizing area, and ensuring proper power and clock distribution.

2. **Placement:**

- The standard cells (logic gates, flip-flops, etc.) are placed within the defined floorplan.
- Placement tools optimize for area, timing, and power consumption.
- Considerations like congestion and design rule compliance are considered.

3. **Clock Tree Synthesis (CTS):**

- A robust clock distribution network (clock tree) is designed to minimize clock skew and ensure that all parts of the chip are synchronized.
- This step is critical for timing closure, ensuring the IC operates correctly at the desired clock frequency.

4. **Routing:**

- After placement, the connections between components (nets) are physically routed using metal layers.
- Tools ensure routing adheres to design rules (e.g., spacing and layer constraints).
- Routing is optimized for signal integrity, power efficiency, and minimal delays.

5. **Power Planning:**

- Proper power distribution is ensured to minimize voltage drops and prevent hot spots.
- Power grids are designed to supply adequate power to all components of the chip.

6. **Design Rule Check (DRC):**

- A verification step to ensure the design adheres to the manufacturing process's design rules.
- Any violations are corrected before moving forward.

7. Equivalence Checking:

- Ensures that the physical layout matches the original netlist generated during synthesis.
- This is a critical step to confirm that no logical discrepancies exist in the design.

8. Parasitic Extraction:

- Parasitics (resistance, capacitance, and inductance) introduced by the physical layout are extracted.
- Accurate parasitic information is essential for post-layout simulations.

9. Timing Signoff:

- Using Static Timing Analysis (STA), the design is analysed to ensure it meets all timing requirements, including setup and hold times.
- Iterative adjustments may be made to meet these constraints.

10. Power and Signal Integrity Analysis:

- Verifies that the power distribution network can handle the chip's power demands without excessive noise or voltage drops.
- Signal integrity checks ensure there are no issues like crosstalk or electromagnetic interference.

11. Final GDSII Generation:

- The verified physical layout is converted into GDSII format, the standard file format used for IC fabrication.
- This marks the "tape-out" phase, where the design is finalized and sent to the foundry for manufacturing.

This complete design flow is achieved with help of the Synopsys EDA tools, namely

- Synopsys VCS®
- Synopsys Verdi®
- Synopsys Design Compiler®
- Synopsys IC Compiler™ II
- Synopsys PrimeTime® Suite

3. PROJECT IMPLEMENTATION AND DISCUSSION

The Implementation of the Project ‘8-bit Full Adder using 1-bit Full Adder’ and its results, resolutions, optimizations, reports and more are discussed in detail

3.1 Front End Design

Steps such as specification, design, verification of HDls and synthesis of RTL is categorized here

3.1.1 Specification

The design flow of all IC design methodologies always starts with the specifications, where the functionality and operating conditions among other parameters relevant to the product's use are formally specified.

Project Name:

8-bit Full Adder

Description:

An 8-bit full adder constructed from cascading 1-bit full adders. **Project**

Hierarchy:

Top Level: adder_pr

Submodules: adder_1b

Module Port specifications: 'adder_1b'

Table 3-1: adder_1b port specifications

Name	Type	Width	Description
a	Input	1	'a' input to the adder
b	Input	1	'b' input to the adder
cin	Input	1	Carry input to the adder
sum	Output	1	Resulting sum of the a and b ports
carry	Output	1	Carry out from the adder
clk	Input	1	Clock signal, used to synchronise the circuit
rst	Input	1	Reset signal, used to recover to a known state

Module Port specification: 'adder_pr'

Table 3-2: adder_pr port specification

Name	Type	Width	Description

a	Input	8	a input to the adder
b	Input	8	b input to the adder
cin	Input	1	Carry into the adder
sum	Output	8	Resulting sum of the a and b ports
carry	Output	1	Carry out from the adder
clk	Input	1	Clock signal, used to synchronise the circuit
rst	Input	1	Reset signal, used to recover to a known state

Functionality:

Add the given two 8-bit numbers, together with the carry in, and generate the sum and carry out.

This type of adder is also called as aa Ripple Carry Adder (RCA), as every sum bit can be immediately calculated but the carry bits need to be propagated through each full adder, before the final carry out can be considered to be a valid bit. This also introduces some amount of propagation delay.

3.1.2 RTL Design **1-bit**

Full Adder:

A Full Adder be built using AND, OR, and XOR gates. Simplest implementation would be to build a half adder and then cascade them.

Likewise, an 8-bit Full Adder is obtained by cascading 8 1-bit full adders in sequence.

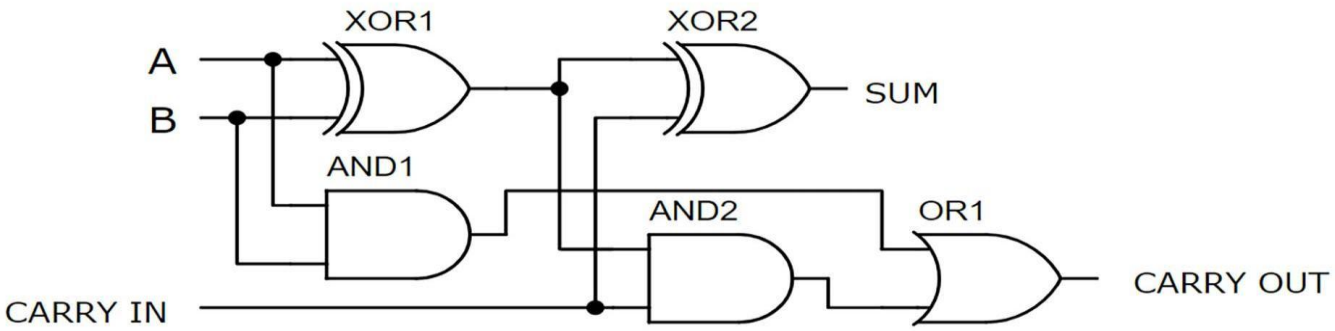


Figure 3-1: 1-bit Full Adder

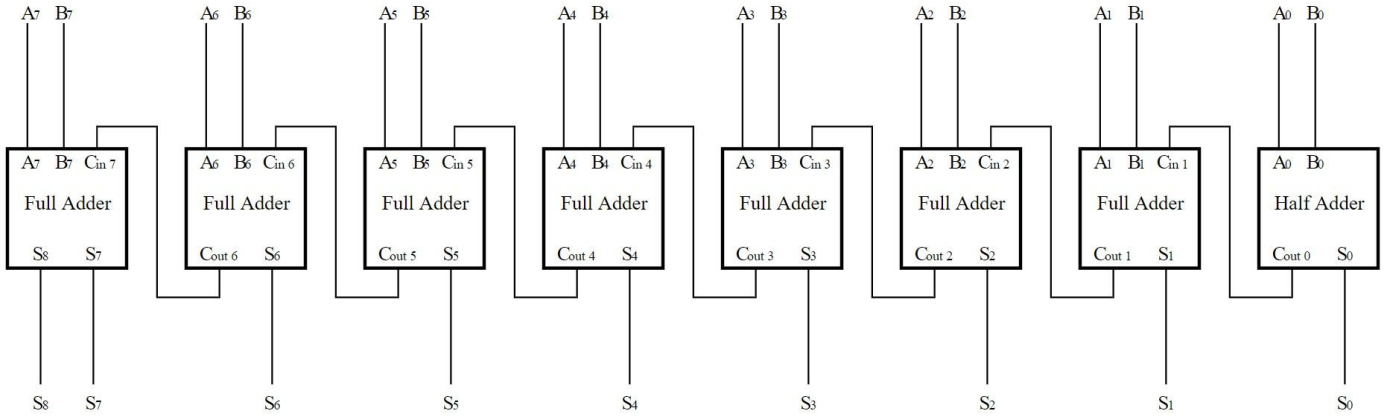


Figure 3-2: 8-bit Full Adder from 1-bit Full Adder

Verilog HDL Code for 1-bit Full adder:

```
 `default_nettype none

module adder_1b (
input wire clk,
input wire cin,
input wire rst,
input wire a, b,
output reg sum, carry
);


always @ (posedge clk or posedge rst) begin
if (rst) begin
    {carry, sum} <= 0;
end
else begin
    {carry, sum} <= a + b + cin;
end
end

endmodule
```

Figure 3-3: Verilog Code for 1-bit Full Adder

Verilog HDL Code for 8-bit Full Adder using 1-bit Full Adders:

```
`default_nettype none
`include "adder_1b.v"

module adder_pr (
input wire clk, cin,
input wire rst,
input wire [7:0] a, b,
output reg [7:0] sum,
output reg carry
);

wire [7:0] sumw;
wire cw0, cw1, cw2, cw3, cw4, cw5, cw6, cw7;

adder_1b inst0 ( clk, cin, rst, a[0], b[0], sumw[0], cw0);
adder_1b inst1 ( clk, cw0, rst, a[1], b[1], sumw[1], cw1);
adder_1b inst2 ( clk, cw1, rst, a[2], b[2], sumw[2], cw2);
adder_1b inst3 ( clk, cw2, rst, a[3], b[3], sumw[3], cw3);
adder_1b inst4 ( clk, cw3, rst, a[4], b[4], sumw[4], cw4);
adder_1b inst5 ( clk, cw4, rst, a[5], b[5], sumw[5], cw5);
adder_1b inst6 ( clk, cw5, rst, a[6], b[6], sumw[6], cw6);
adder_1b inst7 ( clk, cw6, rst, a[7], b[7], sumw[7], cw7);

always@ (posedge clk or posedge rst) begin
    if (rst) begin
        sum <= 0;
        carry <= 0;
    end
    else begin
        sum <= sumw;
        carry <= cw7;
    end
end

endmodule
```

Figure 3-4: 8-bit Full Adder using 1-bit Full Adder

3.1.3 RTL Verification

Front-End RTL Verification refers to the process of ensuring that a digital design, expressed at the Register Transfer Level (RTL) using hardware description languages (HDLs) such as Verilog or VHDL, meets its intended functional specifications. This verification is a critical step in the pre-silicon design lifecycle, as it identifies and rectifies potential design flaws before the design is sent for physical implementation.

8-bit Full Adder testbench:

```
1  `default_nettype none
2  `timescale 1 ns / 10 ps
3  `include "adder_pr.v"
4
5  module tb;
6
7  reg [7:0] a, b;
8  reg clk, rst, cin;
9  wire [7:0] sum;
10 wire carry;
11
12 integer i, file ,console;
13 reg [8:0] exp, tcase, tt;
14
15 adder_pr inst0 ( clk, cin , rst, a, b, sum, carry);
16
17 initial clk = 0;
18 always #5 clk = ~clk;
19
20
21 task add;
22 input [8:0] exp_val;
23 if ({carry, sum} != exp_val)
24     $fdisplay(console, "[%t] a = %d ( %b ) b = %d ( %b ) cin = %d sum = %d ( %b ) carry = %d, # expected {carry, sum} = %d ( %b )",
          $realtime, a, a, b, b, cin, sum ,sum, carry, exp ,exp);
25 else begin
26     $fdisplay(console, "[%t] a = %d ( %b ) b = %d ( %b ) cin = %d sum = %d ( %b ) carry = %d ", $realtime,  a, a, b, b, cin, sum ,sum,
          carry);
27     tcase = tcase + 1;
28 end
29
30 endtask
31
```

Figure 3-5: 8-bit Adder testbench – Part 1 / 2

```
32  initial begin
33      $fsdbDumpvars();
34      $timeformat(-9, 2, "ns", 10);
35      file = $fopen("adder_pr_sim.log");
36      console = file | 32'b1;
37      rst = 0; a = 0; b =0; cin = 0; tcase = 0; tt = 16; #10;
38      // reset
39      rst = 1; #20;
40      @(posedge clk);
41      rst = 0;
42      @(posedge clk);
43      $fdisplay(console, "################################################################################");
44      $fdisplay(console,  "\t 8-bit adder using 1-bit full adder");
45
46      for (i = 0; i < tt; i = i + 1) begin
47          a = $random; b = $random; cin = $random;
48          repeat (9) begin
49              @(posedge clk);
50          end
51          exp = a + b + cin;
52          add ( exp );
53      end
54      $fdisplay(console,  "\t Test cases : %d. PASSED : %d", tt, tcase);
55      $fdisplay(console, "################################################################################");
56      $finish;
57  end
```

Figure 3-6: 8-bit Adder testbench – Part 2 / 2

Running the simulation via VCS and Verdi:

Compiling the Source files:

File Structure (rtl_simulation):

```
.
├── adder_1b.v
├── adder_pr_tb.v
└── adder_pr.v
```

Procedure to compile the Verilog sources:

- Open terminal, ensuring the files are proper, as shown
- In the terminal, invoke the VCS tool using following command

- vcs -full64 adder_pr_tb.v -lca -kdb -

debug_access+all Break down:

-full64: Instructing the tool to use 64-bit version software

-lca: (Low-Cost-Access) License Type specification

-kdb: Knowledge Database, enables use of databases such as fsdb (Fast Signal Database) -

debug_access+all: Enable all debug-related features

- The above command will analyse, lint, and compile the testbench, by extension the source files too. The compiled output will be available under the executable named “simv”

- ./simv verdi Break down:

  -Runs the compiled output file using Verdi Tool, also generates the .fsdb files

- verdi -ssf ./novas.fsdb -nologo Break down:

  -ssf: specifies the 'signal save file', which is 'novas.fsdb'

  -nologo: suppresses the Synopsys Logo from appearing

  - This command will launch the GUI of verdi tool, where we can view the waveforms and schematics.

```
################################################################################

                        8-bit adder using 1-bit full adder

[  135.00ns] a =  36 ( 00100100 ) b = 129 ( 10000001 ) cin = 1 sum = 166 ( 10100110 ) carry = 0

[  225.00ns] a =  99 ( 01100011 ) b =  13 ( 00001101 ) cin = 1 sum = 113 ( 01110001 ) carry = 0

[  315.00ns] a = 101 ( 01100101 ) b =  18 ( 00010010 ) cin = 1 sum = 120 ( 01111000 ) carry = 0

[  405.00ns] a =  13 ( 00001101 ) b = 118 ( 01110110 ) cin = 1 sum = 132 ( 10000100 ) carry = 0

[  495.00ns] a = 237 ( 11101101 ) b = 140 ( 10001100 ) cin = 1 sum = 122 ( 01111010 ) carry = 1

[  585.00ns] a = 198 ( 11000110 ) b = 197 ( 11000101 ) cin = 0 sum = 139 ( 10001011 ) carry = 1

[  675.00ns] a = 229 ( 11100101 ) b = 119 ( 01110111 ) cin = 0 sum =  92 ( 01011100 ) carry = 1

[  765.00ns] a = 143 ( 10001111 ) b = 242 ( 11110010 ) cin = 0 sum = 129 ( 10000001 ) carry = 1

[  855.00ns] a = 232 ( 11101000 ) b = 197 ( 11000101 ) cin = 0 sum = 173 ( 10101101 ) carry = 1

[  945.00ns] a = 189 ( 10111101 ) b =  45 ( 00101101 ) cin = 1 sum = 235 ( 11101011 ) carry = 0

[ 1035.00ns] a =  99 ( 01100011 ) b =  10 ( 00001010 ) cin = 0 sum = 109 ( 01101101 ) carry = 0

[ 1125.00ns] a =  32 ( 00100000 ) b = 170 ( 10101010 ) cin = 1 sum = 203 ( 11001011 ) carry = 0

[ 1215.00ns] a = 150 ( 10010110 ) b =  19 ( 00010011 ) cin = 1 sum = 170 ( 10101010 ) carry = 0

[ 1305.00ns] a =  83 ( 01010011 ) b = 107 ( 01101011 ) cin = 1 sum = 191 ( 10111111 ) carry = 0 [

 1395.00ns] a =   2 ( 00000010 ) b = 174 ( 10101110 ) cin = 1 sum = 177 ( 10110001 ) carry = 0

[ 1485.00ns] a = 207 ( 11001111 ) b =  35 ( 00100011 ) cin = 0 sum = 242 ( 11110010 ) carry = 0

                        Test cases :  16. PASSED :  16

################################################################################
```

Figure 3-7: Testbench Results

The above text-based output can be obtained from running the 'simv' file. To view the actual waveform based output and also the RTL level schematic, we use the Verdi tool.

Once the Verdi tool is opened select the module of concern, right click it then aadd it to Waveform window.

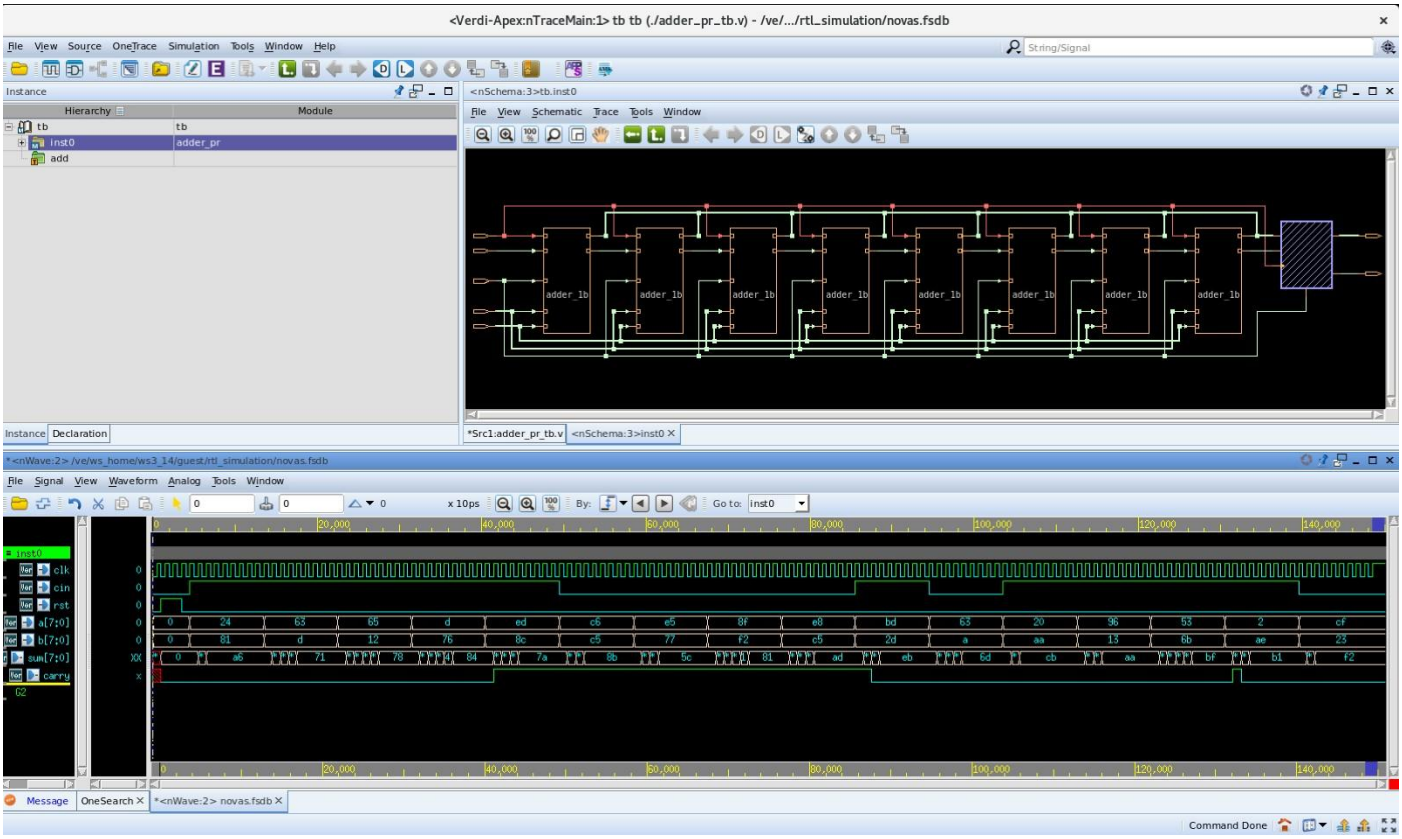


Figure 3-8: RTL schematic and Simulation waveform

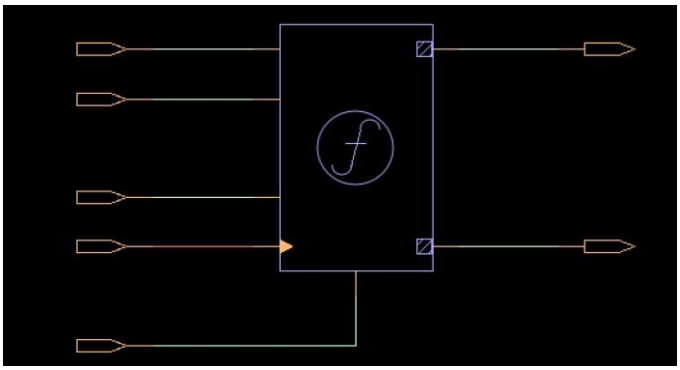


Figure 3-9: RTL Schematic of 'adder_1b'

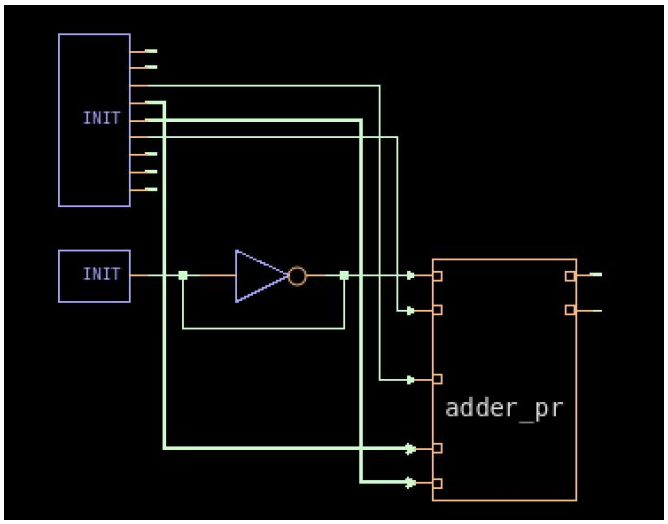


Figure 3-10: RTL Schematic of testbench

3.1.4 RTL Synthesis

Logic synthesis is a critical process in the front-end design flow of digital systems, tasked with translating high-level RTL (Register Transfer Level) descriptions into a gate-level representation that adheres to the constraints of a specific technology library. This process serves as the bridge between the functional abstraction of a design and its physical realization.

Logic Synthesis Using Design Compiler:

The following commands are used to do the logic synthesis in the Design Compiler.

- dc_shell

  Invoke the design compiler from inside the DC folder, where the script to run dc, and few other default scripts are present.

  We set the PDK path, point the source files, set constraints, the generate the schematic and approximate reports of various parameters like Timings, Power, Area etc...

- source run_dc.tcl where run_dc.tcl is our scripts.

Once this process is complete, we can move on to viewing the schematic and reports

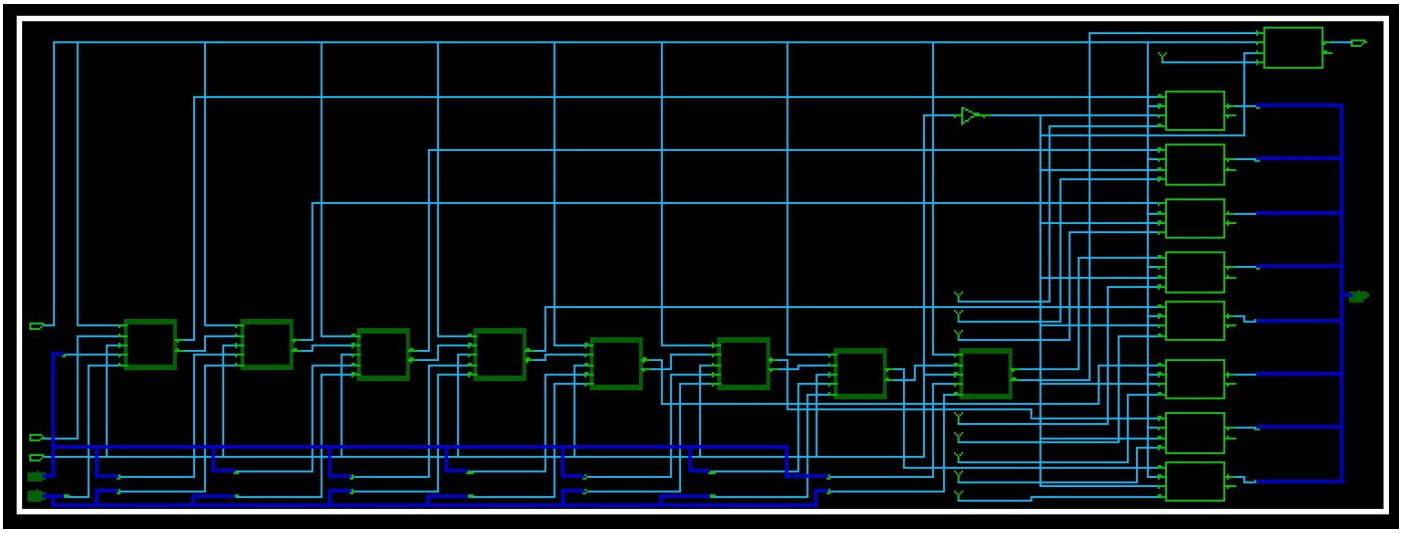


Figure 3-11: Mapped Netlist Schematic of RTL design

3.2 Back End Design

3.2.1 Floor Planning

We use the ICC2 Tool, i.e. the Integrated Circuit Compiler 2 Tool to achieve Floor planning.

Similar to the DC flow, we first go to a separate folder, in this case the ICCII folder, then invoke the icc2_shell which has a bunch of predefined environment variables for us.

- Icc2_shell
- Source ./script/floorplanning.tcl
  Where all the floor planning information, such as shape of core die, relative size of it, placement of IO pins, etc... are all defined.

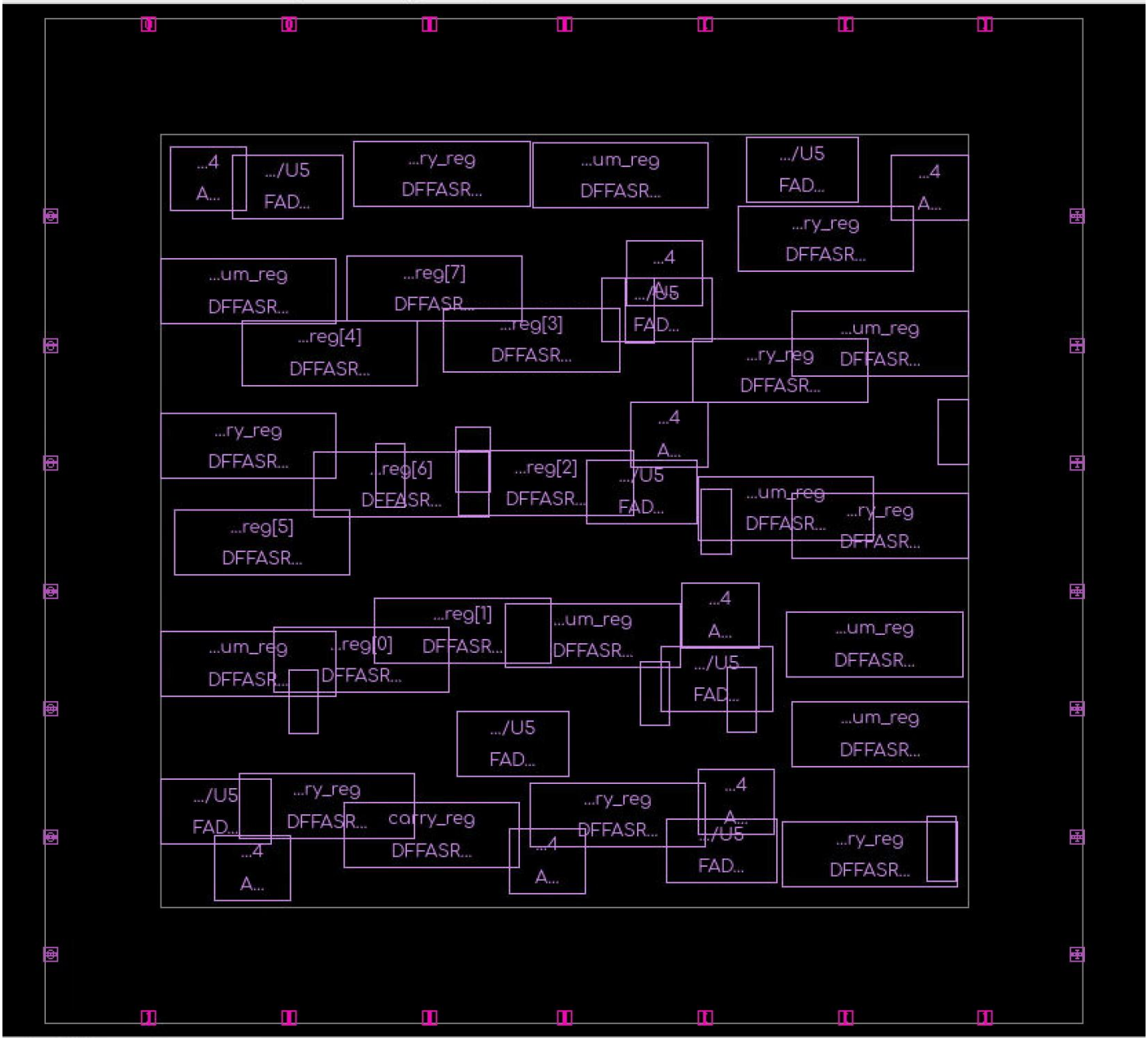


Figure 3-12: Floorplan of the RTL Design

3.2.2 Placement

Back-End Placement in IC physical design involves assigning exact locations to standard cells within the chip layout to optimize performance, timing, and area while ensuring routing feasibility. It includes:

- Global Placement: Roughly positioning cells to minimize wirelength.
- Detailed Placement: Refining cell positions to align with timing and design rules.
- Legalization: Ensuring cells are placed without overlaps and comply with manufacturing constraints.

Placement impacts chip timing, power efficiency, and manufacturability, preparing the design for routing and optimization.

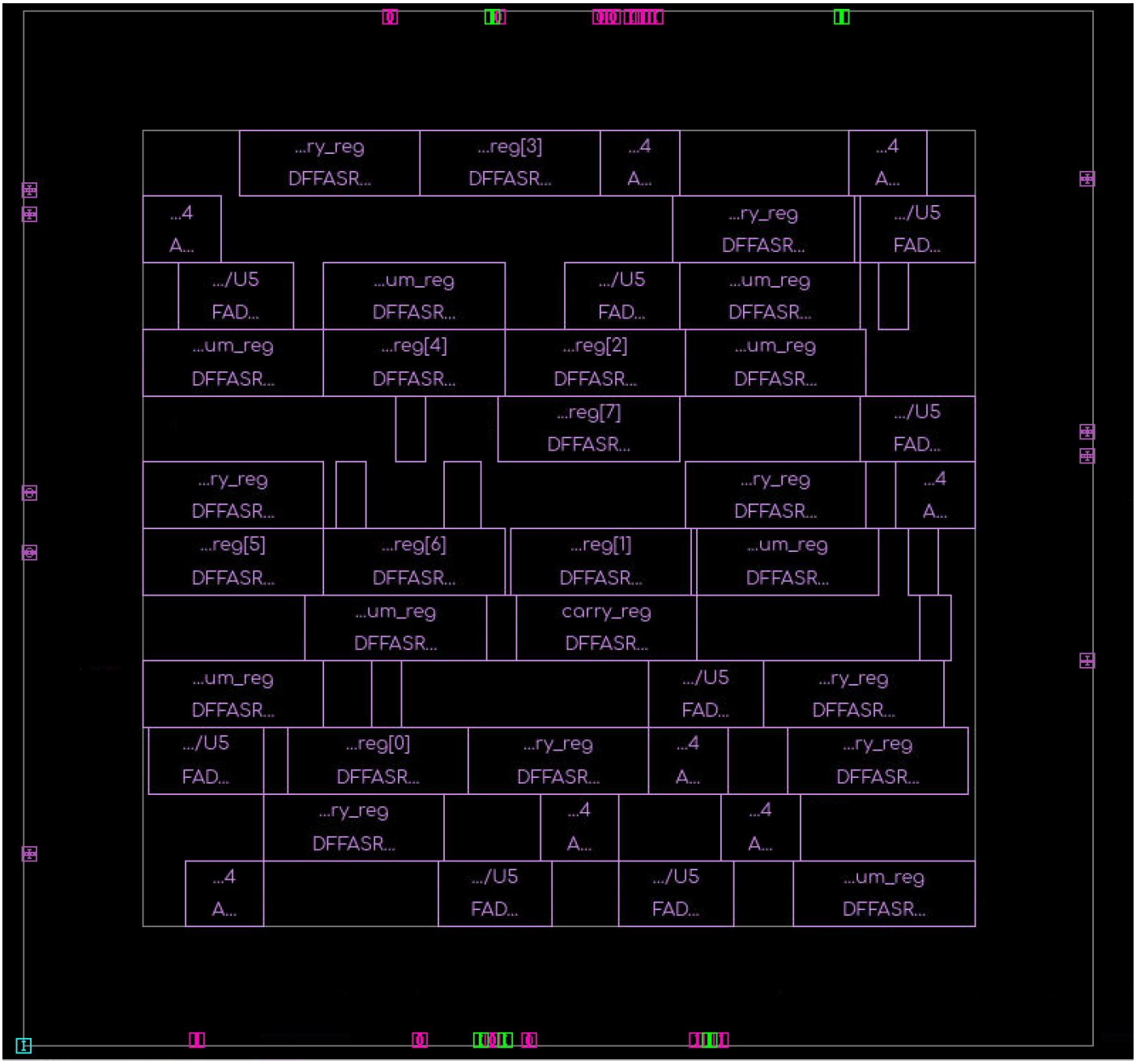


Figure 3-13: Placement of Cells

3.2.3 Power Planning

Back-End Power Planning is a critical stage in the physical design process, focused on ensuring stable and efficient power delivery across the chip while meeting performance and reliability constraints.

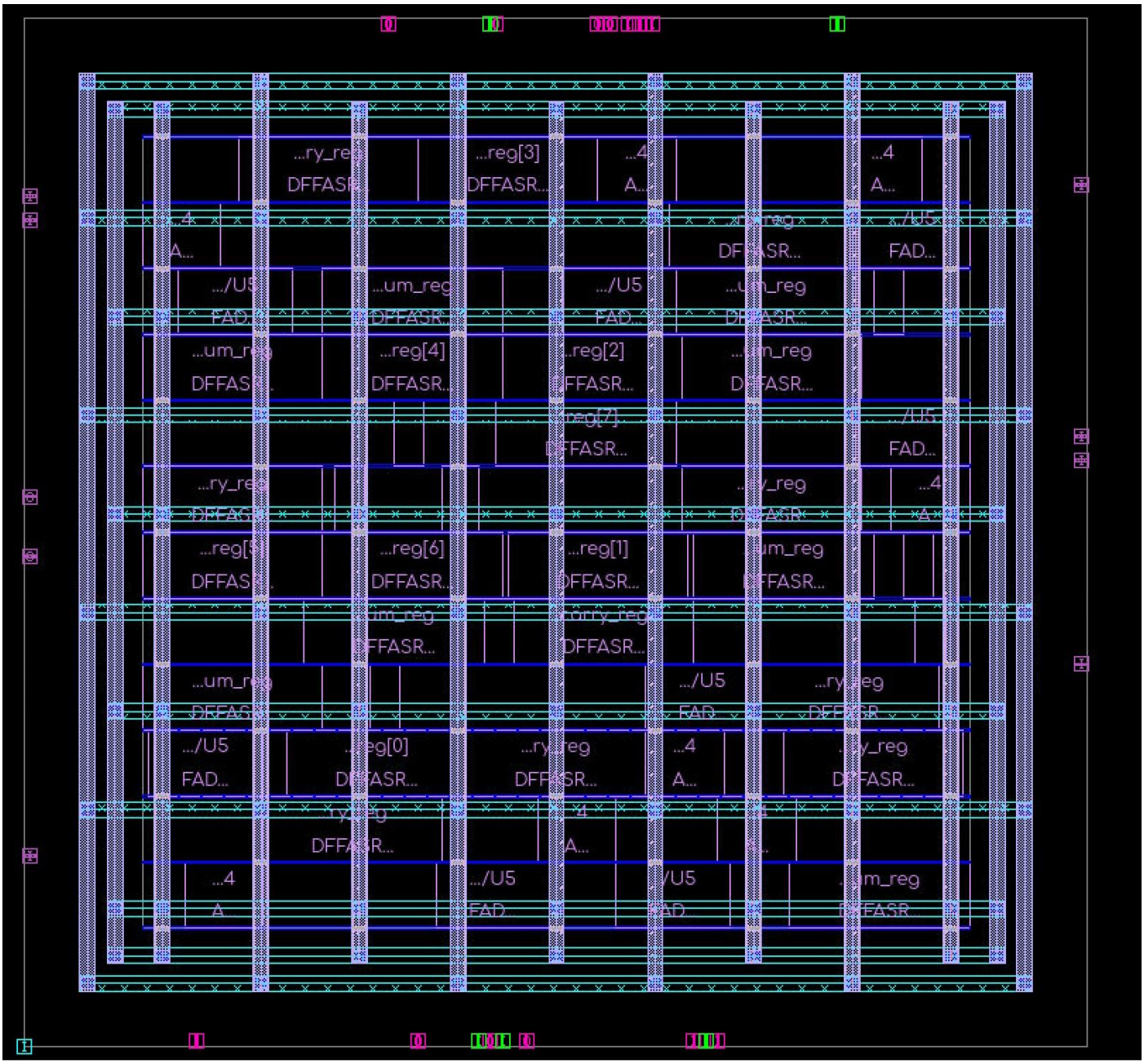


Figure 3-14:Layout of the RTL design at the Post - Power Planning stage

3.2.4 Clock Tree Synthesis

Back-End Clock Tree Synthesis (CTS) is a crucial step in the physical design flow, responsible for generating a balanced clock tree to distribute the clock signal across the chip efficiently and with minimal delay. It ensures that all sequential elements (e.g., flip-flops) receive the clock signal with minimal skew and jitter while meeting performance constraints.

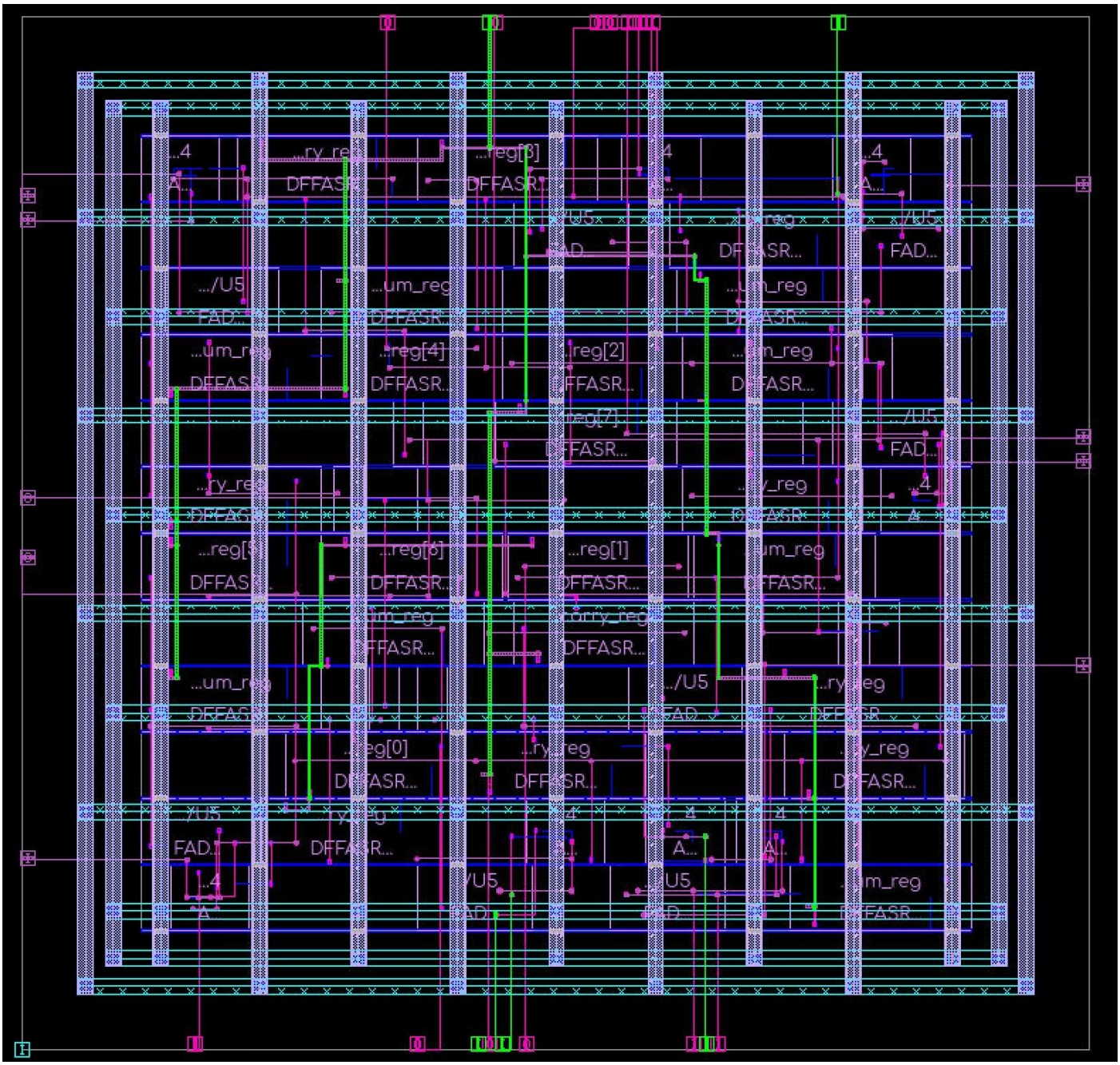


Figure 3-15: Layout Post CTS

3.2.5 Routing

Back-End Routing is the process of connecting all placed standard cells, macros, and I/O pins using metal layers according to the design rules and constraints. It ensures that all nets (signals) are physically routed while meeting timing, power, and signal integrity requirements.

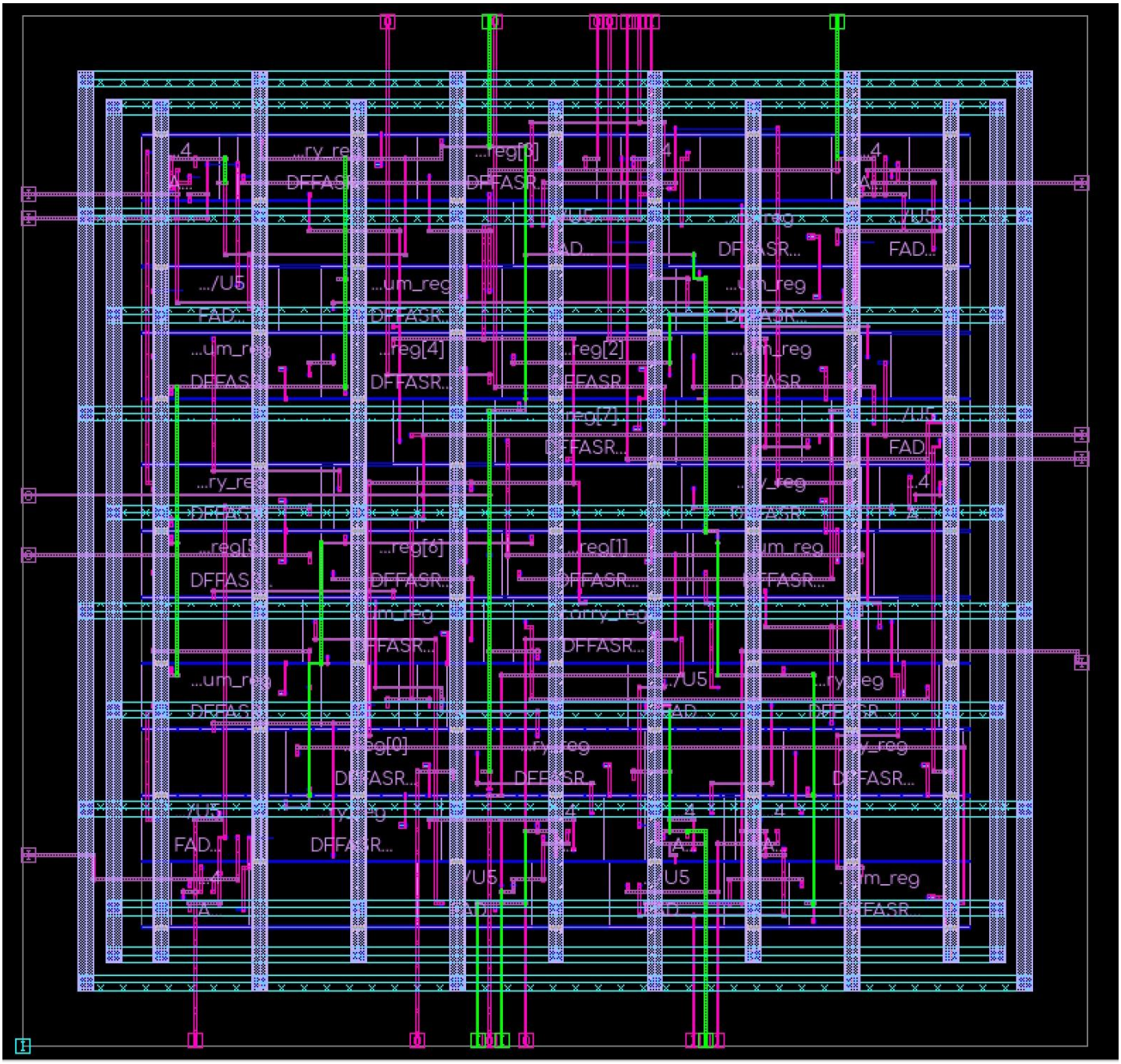


Figure 3-16: Final Layout with Routing

3.3 Reports

Report Summary (DC):

Power:

Parameter	Value
Number of ports	84
Number of nets	150
Number of cells	83
Number of combinational cells	25
Number of sequential cells	25
Number of buffers/inverters	9
Number of references **Area:**	10

Parameter	Value (sq. units)
Combinational area	76.751487
Buffer/Inverter area	11.690624
Non-combinational area	190.608001
Net Interconnect area	35.985573
Total cell area	267.359488
Total area **Timing:**	303.345061

Slack

Max Path	0.76
Min Path	0.18

Report Summary (ICC):

Utilization:

Parameter	Value

Utilization Ratio	0.6353
Total Area	420.8625
Total Capacity Area	420.8625
Total Area of Cells **Wire Length:**	267.3595

Type	**Value**
TotalH	322.322
TotalV	373.530
Total	695.851

Parameter	**Value (pW)**	**Percentage**
Cell Internal Power	5.85e+06	63.4%
Net Switching Power	3.38e+06	36.6%
Total Dynamic Power	9.23e+06	100.0%
Cell Leakage Power	5.99e+08	-

Slack	
Max Path	0.76
Min Path	0.10

All crucial information of the design can be viewed using the following command ➲

report_design –all

```
****************************************
Report : design
        -library
        -netlist
        -floorplan
        -routing
Design : adder_pr
Version: V-2023.12-SP4
Date   : Sat Mar 15 05:50:35 2025
****************************************
--------------------------------------------------------------------------------
                            LIBRARY INFORMATION
--------------------------------------------------------------------------------

Search path : .

Units :
    time                : 1.00ns
    resistance          : 1.00MOhm
    capacitance         : 1.00fF
    voltage             : 1.00V
    current             : 1.00uA
    power               : 1.00pW

Tech file : /slowfs/cae613/lucass/NDMS/saed32/work/icc2_frame/data/TF/tmp.tf

Total number of standard cells  = 286

Total number of dont_use lib cells      = 7
Total number of dont_touch lib cells    = 7

Total number of buffers         = 11
Total number of inverters       = 15
Total number of flip-flops      = 106
Total number of latches         = 12
Total number of ICGs            = 12


Library : saed32rvt_c
  File path : /ve/ws_home/ws3_14/Risi/ref/lib/ndm/saed32rvt_c.ndm
  Source .db libs :
    saed32rvt_ff0p85v125c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p85v125c.db
    saed32rvt_ff0p85v25c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p85v25c.db
    saed32rvt_ff0p85vn40c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p85vn40c.db
    saed32rvt_ff0p95v125c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p95v125c.db
    saed32rvt_ff0p95v25c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p95v25c.db
    saed32rvt_ff0p95vn40c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff0p95vn40c.db
    saed32rvt_ff1p16v125c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff1p16v125c.db
    saed32rvt_ff1p16v25c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff1p16v25c.db
    saed32rvt_ff1p16vn40c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ff1p16vn40c.db
    saed32rvt_ss0p75v125c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ss0p75v125c.db
    saed32rvt_ss0p75v25c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ss0p75v25c.db
    saed32rvt_ss0p75vn40c : /remote/cae1107/libraries/SAED/SAED32_EDK/lib/stdcell_rvt/db_nldm/saed32rvt_ss0p75vn40c.db
```

```
--------------------------------------------------------------------------------
                            NETLIST INFORMATION
--------------------------------------------------------------------------------

CELL INSTANCE INFORMATION
--------------------------------------------------------------------------------
Cell Instance Type          Count % of         Area  % of siteAreaPerSite
                                  total             total
--------------------------------------------------------------------------------
TOTAL LEAF CELLS               50  100      267.359   100 unit:1052

  Standard cells               50  100      267.359   100 unit:1052
  Filler cells                  0    0        0.000     0
  Diode cells                   0    0        0.000     0
  Hard macro cells              0    0        0.000     0
  Soft macro cells              0    0        0.000     0
    Black box cells             0    0        0.000     0
  Analog block cells            0    0        0.000     0
  Pad cells                     0    0        0.000     0
    Flip-chip pad cells         0    0        0.000     0
  Cover cells                   0    0        0.000     0
  Flip-chip driver cells        0    0        0.000     0
  Corner pad cells              0    0        0.000     0
  Pad spacer cells              0    0        0.000     0
  Others                        0    0        0.000     0

Special cells                   0    0        0.000     0
  Level shifter                 0    0        0.000     0
  Isolation                     0    0        0.000     0
  Switch                        0    0        0.000     0
  Always on                     0    0        0.000     0
  Retention                     0    0        0.000     0
  Tie off                       0    0        0.000     0

LVT                             0    0        0.000     0
HVT                             0    0        0.000     0
Normal VT                       0    0        0.000     0
Others                         50  100      267.359   100 unit:1052

Netlist cells                  50  100      267.359   100 unit:1052
Physical only                   0    0        0.000     0

Fixed cells                     0    0        0.000     0
Moveable cells                 50  100      267.359   100 unit:1052

Placed cells                   50  100      267.359   100 unit:1052
Combinational                  25   50       76.751    28 unit:302
Sequential                     25   50      190.608    71 unit:750
Others                          0    0        0.000     0
```

```
Special cells                   0    0        0.000     0
  Level shifter                 0    0        0.000     0
  Isolation                     0    0        0.000     0
  Switch                        0    0        0.000     0
  Always on                     0    0        0.000     0
  Retention                     0    0        0.000     0
  Tie off                       0    0        0.000     0

LVT                             0    0        0.000     0
HVT                             0    0        0.000     0
Normal VT                       0    0        0.000     0
Others                         50  100      267.359   100 unit:1052

Netlist cells                  50  100      267.359   100 unit:1052
Physical only                   0    0        0.000     0

Fixed cells                     0    0        0.000     0
Moveable cells                 50  100      267.359   100 unit:1052

Placed cells                   50  100      267.359   100 unit:1052
Combinational                  25   50       76.751    28 unit:302
Sequential                     25   50      190.608    71 unit:750
Others                          0    0        0.000     0

Buffer                          0    0        0.000     0
Inverter                        9   18       11.691     4 unit:46
Buffer/inverter                 9   18       11.691     4 unit:46

Spare cells                     0    0        0.000     0
ICG cells                       0    0        0.000     0
Flip-flop cells                25   50      190.608    71 unit:750
Latch cells                     0    0        0.000     0
Antenna cells                   0    0        0.000     0

Mux logic                       0    0        0.000     0
Double height                   0    0        0.000     0
Triple height                   0    0        0.000     0
More than triple height         0    0        0.000     0

Logic Hierarchies          8

REFERENCE DESIGN INFORMATION
--------------------------------------------------------------------------------------------------
Number of reference designs used:5
--------------------------------------------------------------------------------------------------
Name            Type          Count     Width    Height        Area   PinDens SiteName siteArea
--------------------------------------------------------------------------------------------------
DFFASRX1_RVT    lib_cell         25      4.56      1.67       7.624     1.049 unit           30
INVX0_RVT       lib_cell          8      0.76      1.67       1.271     3.148 unit            5
AO222X1_RVT     lib_cell          8      1.98      1.67       3.304     2.724 unit           13
FADDX1_RVT      lib_cell          8      2.89      1.67       4.829     1.450 unit           19
INVX2_RVT       lib_cell          1      0.91      1.67       1.525     2.623 unit            6
```

NET INFORMATION

NetType	Count	FloatingNets	Vias	Nets/Cells
Total	71	0	944	1.420
Signal	68	0	395	1.360
Power	1	0	292	0.020
Ground	1	0	180	0.020
Analog Signal	0	0	0	0.000
Analog Ground	0	0	0	0.000
Analog Power	0	0	0	0.000
Clock	1	0	77	0.020
Tie Low	0	0	0	0.000
Tie High	0	0	0	0.000
Others	0	0	0	0.000

NET FANOUT AND PIN COUNT INFORMATION

Fanout	Netcount	netPinCount	NetCount
<2	34	<2	0
2	8	2	34
3	24	3	8
4	0	4	24
5	0	5	0
6-10	2	6-10	2
11-20	0	11-20	0
21-30	1	21-30	1
31-50	1	31-50	0
51-100	1	51-100	2
101-500	0	101-500	0
501-1000	0	501-1000	0
>1000	0	>1000	0

PORT AND PIN INFORMATION

Type	Total	Input	Output	Inout	3-st	Power	Ground
Total	364	281	83	0	0	50	50
Macro	0	0	0	0	0	0	0
Ports	30	21	9	0	0	1	1

FLOORPLAN INFORMATION

CORE AND CHIP AREA INFORMATION

Core Area is : 420.862
Chip Area is : 703.102

```
SITE ROW INFORMATION
--------------------------------------------------------------------------
Site Name    Width    Height   Total Rows     Total Tiles          Area
--------------------------------------------------------------------------
unit          0.15      1.67           12            1656       420.862
--------------------------------------------------------------------------

BLOCKAGE INFORMATION
-------------------------------------------------
Blockage Type                Count          Area
-------------------------------------------------
Hard placement                   0         0.000
Soft placement                   0         0.000
Hard macro                       0         0.000
Partial placement                0         0.000
Register                         0         0.000
Placement allow Buffer Only      0         0.000
Placement allow RP Group Only
                                 0         0.000
RP Group                         0         0.000
Category                         0         0.000
Routing                          0         0.000
Routing for Top                  0         0.000
Routing For Design Rule          0         0.000
Shaping                          0         0.000
-------------------------------------------------

POWER DOMAIN INFORMATION
------------------------------------------------------------------------
Power Domain Name    VA Name         Primary Power Net Primary Ground Net
------------------------------------------------------------------------
DEFAULT_POWER_DOMAIN DEFAULT_VA      VDD               VSS
------------------------------------------------------------------------

VOLTAGE AREA INFORMATION
-----------------------------------------------------------------------------------------------
VA Name          Number             Area       Target    bbox    bbox    bbox    bbox
              of shapes                    Utilization     llx     lly     urx     ury
-----------------------------------------------------------------------------------------------
DEFAULT_VA            1          420.862         1.00    3.00    3.00   23.98   23.06
-----------------------------------------------------------------------------------------------

GROUP BOUND INFORMATION
---------------------------------------
No Group Bound exists

EXCLUSIVE MOVEBOUND INFORMATION
---------------------------------------
No Exclusive MoveBound exists.
```

```
HARD AND SOFT MOVEBOUND INFORMATION
-----------------------------------------
No Hard Or Soft MoveBound exists.

ROUTE GUIDE INFORMATION
-------------------------------------------------
Route Guide Type             Count          Area
-------------------------------------------------
Extra Detour Region              0         0.000
Over icovl CellLayers            0         0.000
River Routing                    0         0.000
Area Double Via                  0         0.000
Access Preference                0         0.000
Default                          0         0.000
Switched Direction Only          0         0.000
Max Patterns                     0         0.000
Others                           0         0.000
-------------------------------------------------

MULTIBIT REGISTER INFORMATION
-----------------------------------------------
No Multibit cells exist

MULTIBIT LS/ISO CELLS INFORMATION
-----------------------------------------------
No Multibit cells exist

RP GROUP INFORMATION
-----------------------------------------
No RP Group exists

LAYER INFORMATION
-------------------------------------------------------------------------------
Layer Name Direction Ignored Pitch  default minWidth minSpacing    sameNet
                                      Width                     MinSpacing
-------------------------------------------------------------------------------
M1         Hor       NO      0.15     0.05     0.05       0.05       0.00
M2         Ver       NO      0.15     0.06     0.06       0.06       0.06
M3         Hor       NO      0.30     0.06     0.06       0.06       0.06
M4         Ver       NO      0.30     0.06     0.06       0.06       0.06
M5         Hor       NO      0.61     0.06     0.06       0.06       0.06
M6         Ver       NO      0.61     0.06     0.06       0.06       0.06
M7         Hor       NO      1.22     0.06     0.06       0.06       0.06
M8         Ver       NO      1.22     0.06     0.06       0.06       0.06
M9         Hor       NO      2.43     0.16     0.16       0.16       0.00
MRDL       Ver       NO      4.86     2.00     2.00       2.00       0.00
-------------------------------------------------------------------------------
--------------------------------------------------------------------------------
                              ROUTING INFORMATION
--------------------------------------------------------------------------------

Total wire length = 749.36 micron
```

Total number of wires = 561
Total number of contacts = 944

FINAL WIRING STATISTICS

Signal Wiring Statistics

Metal layer	Num wires	% of total#	Wire length	% of total length
PO	0	0.00%	0.00	0.00%
M1	24	5.26%	10.96	1.66%
M2	288	63.16%	328.91	49.83%
M3	135	29.61%	297.54	45.08%
M4	9	1.97%	22.67	3.43%
M5	0	0.00%	0.00	0.00%
M6	0	0.00%	0.00	0.00%
M7	0	0.00%	0.00	0.00%
M8	0	0.00%	0.00	0.00%
M9	0	0.00%	0.00	0.00%
MRDL	0	0.00%	0.00	0.00%

Clock Wiring Statistics

Metal layer	Num wires	% of total#	Wire length	% of total length
PO	0	0.00%	0.00	0.00%
M1	0	0.00%	0.00	0.00%
M2	47	44.76%	5.37	6.02%
M3	37	35.24%	29.18	32.68%
M4	21	20.00%	54.73	61.30%
M5	0	0.00%	0.00	0.00%
M6	0	0.00%	0.00	0.00%
M7	0	0.00%	0.00	0.00%
M8	0	0.00%	0.00	0.00%
M9	0	0.00%	0.00	0.00%
MRDL	0	0.00%	0.00	0.00%

P/G Wiring Statistics

Metal layer	Num wires	% of total#	Wire length	% of total length
PO	0	0.00%	0.00	0.00%
M1	13	17.33%	272.69	25.51%
M2	28	37.33%	19.73	1.85%
M3	0	0.00%	0.00	0.00%
M4	0	0.00%	0.00	0.00%
M5	0	0.00%	0.00	0.00%
M6	9	12.00%	202.38	18.93%
M7	12	16.00%	281.71	26.35%
M8	13	17.33%	292.63	27.37%
M9	0	0.00%	0.00	0.00%
MRDL	0	0.00%	0.00	0.00%

Shape Pattern Wiring Statistics

Metal layer	Num shapePatterns	% of total#	Wire length	% of total length
PO	0	0.00%	0.00	0.00%
M1	0	0.00%	0.00	0.00%
M2	0	0.00%	0.00	0.00%
M3	0	0.00%	0.00	0.00%
M4	0	0.00%	0.00	0.00%
M5	0	0.00%	0.00	0.00%
M6	0	0.00%	0.00	0.00%
M7	0	0.00%	0.00	0.00%
M8	0	0.00%	0.00	0.00%
M9	0	0.00%	0.00	0.00%
MRDL	0	0.00%	0.00	0.00%

All the PG shape patterns stand for 0 shapes.

Horizontal/Vertical Wire Distribution

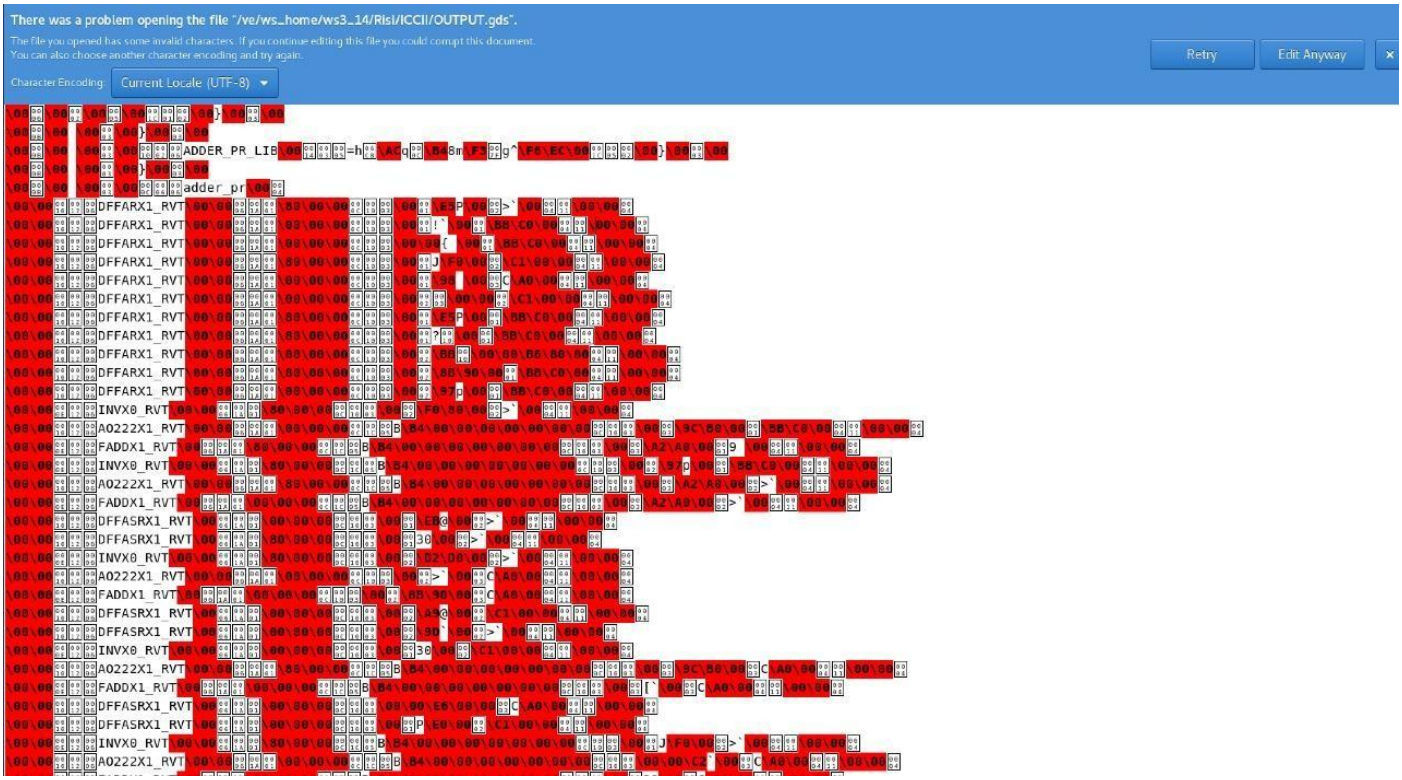
Metal layer	Hor. length	% of hor.	Ver. length	% of ver.
PO	0.00	0.00%	0.00	0.00%
M1	10.96	3.49%	0.00	0.00%
M2	7.90	2.51%	321.00	92.87%
M3	295.11	93.85%	2.43	0.70%
M4	0.46	0.15%	22.22	6.43%
M5	0.00	0.00%	0.00	0.00%
M6	0.00	0.00%	0.00	0.00%
M7	0.00	0.00%	0.00	0.00%
M8	0.00	0.00%	0.00	0.00%
M9	0.00	0.00%	0.00	0.00%
MRDL	0.00	0.00%	0.00	0.00%

FINAL VIA STATISTICS

Via layer	Via def name	Count	% of layer vias
VIA1	VIA12SQ_C	203	64.65%
VIA1	VIA12SQ_C(rot)	27	8.60%
VIA1	VIA12BAR_C	2	0.64%
VIA1	VIA12SQ_C(1X3)	59	18.79%
VIA1	VIA12SQ_C(1X2)	23	7.32%
Double via conversion rate for layer VIA1 = 26.11% (82 / 314 vias)			
Among them, double via conversion rate of detail route vias for layer VIA1 = 9.02% (23 / 255 vias)			
VIA2	VIA23SQ_C(rot)	223	79.08%
VIA2	VIA23SQ_C(1X3)	59	20.92%
Double via conversion rate for layer VIA2 = 20.92% (59 / 282 vias)			
Among them, double via conversion rate of detail route vias for layer VIA2 = 0.00% (0 / 223 vias)			
VIA3	VIA34SQ_C	39	39.80%
VIA3	VIA34SQ_C(1X3)	59	60.20%
Double via conversion rate for layer VIA3 = 60.20% (59 / 98 vias)			
Among them, double via conversion rate of detail route vias for layer VIA3 = 0.00% (0 / 39 vias)			

```
VIA4         VIA45SQ_C(1X3)                        59      100.00%
  Double via conversion rate for layer VIA4 = 100.00% (59 / 59 vias)
    Among them, double via conversion rate of detail route vias for layer VIA4 = 0.00% (0 / 0 vias)

VIA5         VIA56SQ_C(1X3)                        59      100.00%
  Double via conversion rate for layer VIA5 = 100.00% (59 / 59 vias)
    Among them, double via conversion rate of detail route vias for layer VIA5 = 0.00% (0 / 0 vias)

VIA6         VIA67LG_C(2X1)                        54      100.00%
  Double via conversion rate for layer VIA6 = 100.00% (54 / 54 vias)
    Among them, double via conversion rate of detail route vias for layer VIA6 = 0.00% (0 / 0 vias)

VIA7         VIA78LG_C(2X2)                        78      100.00%
  Double via conversion rate for layer VIA7 = 100.00% (78 / 78 vias)
    Among them, double via conversion rate of detail route vias for layer VIA7 = 0.00% (0 / 0 vias)


Custom Via Statistics

No custom vias found in the design.
Overall double via conversion rate = 47.67%
  Among them, overall double via conversion rate of detail route vias = 4.45% (23 / 517 vias)
  *Detail route vias are vias whose shape_use is detail_route.

Via Matrix Statistics

No via matrices found in the design.

FINAL DRC STATISTICS

DRC-SUMMARY:
        @@@@@@@ TOTAL VIOLATIONS =      0
Total number of nets = 71
0 open nets, of which 0 are frozen
Total number of excluded ports = 0 ports of 0 unplaced cells connected to 0 nets
                                 0 ports without pins of 0 cells connected to 0 nets
                                 0 ports of 0 cover cells connected to 0 non-pg nets
Total number of DRCs = 0
Total number of antenna violations = antenna checking not active
1
```

3.4 GDSII Output



4 CONCLUSION AND FUTURE SCOPE

The successful completion of the RTL-to-GDSII flow for an 8-bit full adder in this training program demonstrates a strong understanding of the ASIC design process using Synopsys tools. Each stage, from RTL design to physical layout, was systematically executed, providing hands-on experience in digital design and ASIC development.

The hierarchical approach, using a 1-bit full adder as the building block, ensured modularity and scalability. Synthesis with Synopsys Design Compiler (DC) produced an optimized gate-level netlist. Functional verification using Verdi confirmed correctness, preventing design flaws before physical implementation. Timing constraints were defined using Synopsys Design Constraints (SDC).

Physical design in IC Compiler (ICC) covered floor planning, placement, and power planning while ensuring power integrity. Clock Tree Synthesis (CTS) minimized skew, and routing optimized connectivity. Static Timing Analysis (STA) validated performance.

Final metrics showed an optimal balance between area, power, and performance, with a total cell area of 267.36 sq. units, dynamic power of 9.23e+06 pW, and a utilization ratio of 0.6353, indicating efficient resource allocation.

Future Scope

Future improvements could focus on low-power design techniques such as clock gating to reduce power consumption. Exploring advanced process nodes (e.g., 10nm, 7nm) could enhance area efficiency and performance.

Adding Design for Testability (DFT) features like scan chains would improve post-fabrication testing. Expanding to a 16-bit or 32-bit adder would help analyze scalability challenges.

Integrating the adder into a larger arithmetic unit, such as an ALU, would make it more applicable to processors and signal processing units. FPGA prototyping could further validate the design before fabrication.

This training program reinforced key concepts in VLSI and ASIC design, establishing a strong foundation for tackling more complex projects.