

**RTL to GDS-II Implementation of a Low Power and  
High Speed 8-Bit Counter Using a Galois LFSR and  
Optimization of 64-bit Counter.**

**MINIPROJECT REPORT**

Submitted by

**LAKSHMI RAJKUMAR M**

**REG NO: MDL24ECVE05**

to

The APJ Abdul Kalam Technological University  
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology

in

VLSI and Embedded Systems



**Department of Electronics and Communication Engineering**

Govt. Model Engineering College

Thrikkakara

June 2025



Department of Electronics Engineering  
Model Engineering College  
Thrikkakara

## BONAFIDE CERTIFICATE

*This is to certify that the Mini Project entitled*  
**RTL to GDS-II Implementation of a Low Power and  
High Speed 8-Bit Counter Using a Galois LFSR and  
Optimization of 64-bit Counter**

*Submitted by:*

**LAKSHMI RAJKUMAR M**

**REG NO: MDL24ECVE05**

*is a bonafide account of the work done by her under our supervision*

Head of the department

**Prof. Anjana Devi S**  
Mini project Co-ordinator /Guide

## **Acknowledgement**

First of all I thank the Almighty, for providing me with the resources, knowledge and guidance to complete this project successfully.

I would like to gratefully thank Chip to Startup(C2S) program of MeitY, Government of India for providing the EDA tools.

I express my deep gratitude towards our honorable Principle, Dr. Mini M G and Head of our Department, Dr. Mini M G for providing me with all the necessary facilities that paved way to a successful project.

I would like to thank Assistant Professor Mrs Anjana Devi S for guiding me every step along the way, providing valuable suggestions and passing on confidence and courage to perform the project in the best way possible.

I would also like to express my sincere gratitude to all teaching and non-teaching staff including research scholars of Department of Electronics Engineering for their sincere directions imparted and the cooperation in connection with the project.

I am also thankful to my parents and my colleagues for their support and suggestion given during the course of the project.

## ABSTRACT

High-speed counters play a crucial role in various digital applications, including frequency measurement, clock generation, and data processing. This project focuses on the RTL-to-GDSII implementation of a high-speed 8-bit counter using a Galois Linear Feedback Shift Register (LFSR) with state extension and a traditional 5-bit binary counter. Typically, an n-bit LFSR generates  $2^n - 1$  unique states, but in this design, state extension techniques are applied to modify the state space to  $2^n$ , ensuring seamless integration with a conventional binary counter.

The proposed design consists of a 3-bit modified-state LFSR and a 5-bit binary counter (2bit, 3bit cascaded), whose outputs are concatenated to form an 8-bit high-speed counter. The use of an LFSR-based approach enables reduced switching activity and improved speed compared to traditional ripple or synchronous counters. The project follows the RTL-to-GDSII flow using Cadence tools, covering critical design steps such as Simulation, logic synthesis, Physical Design.

The proposed design of a 64-bit counter is to optimize area, speed with reasonable power compared to the pre-existing design. The stages of the binary counter are reduced to 2 (i.e. 23,35) with elimination of few AND Gates. This project contributes to high-performance digital design methodologies by demonstrating the feasibility of LFSR-based counters in high-frequency applications while ensuring compatibility with conventional binary counters.

---

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Relevance . . . . .	2
1.2	Objective . . . . .	3
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>4</b>
<b>3</b>	<b>SYSTEM DESIGN</b>	<b>7</b>
3.1	Base Model Design[1] . . . . .	7
3.2	Proposed Design of 64 BIT Counter . . . . .	9
3.3	Proposed Design of 8 BIT Counter . . . . .	11
3.4	RTL to GDSII Design Flow . . . . .	12
3.5	Synthesis Flow . . . . .	13
3.6	Physical Design Flow . . . . .	14
3.7	Verification Platform . . . . .	16
<b>4</b>	<b>Implementation of Base Model 64 BIT Counter[1]</b>	<b>17</b>
4.1	Simulation Result . . . . .	18
4.2	Synthesis Report . . . . .	18
4.3	Inference . . . . .	20

<b>5</b>	<b>Implementation of Proposed 64 BIT Counter</b>	<b>22</b>
5.1	Simulation Report . . . . .	22
5.2	Synthesis Report . . . . .	23
5.3	Inference . . . . .	26
<b>6</b>	<b>RTL to GDSII Implementation of Proposed 8 BIT Counter</b>	<b>27</b>
6.1	Simulation Report . . . . .	27
6.2	Synthesis Report . . . . .	28
6.3	Inference . . . . .	31
6.4	Physical Design . . . . .	32
6.4.1	Floor Plan . . . . .	32
6.4.2	Power Plan . . . . .	33
6.4.3	Placement . . . . .	34
6.4.4	Clock Tree Synthesis . . . . .	36
6.4.5	Routing . . . . .	37
6.4.6	DRC and LVS Check . . . . .	40
6.4.7	GDSII . . . . .	42
<b>7</b>	<b>Results and Conclusion</b>	<b>44</b>
<b>8</b>	<b>Future Scope</b>	<b>46</b>

---

# List of Figures

3.1	Base Model Design[1] . . . . .	8
3.2	Proposed Design of 64 BIT Counter . . . . .	9
3.3	Circuit Diagram of 64 BIT Counter . . . . .	10
3.4	Proposed Design of 8 BIT Counter . . . . .	11
3.5	Circuit Diagram of 8 BIT Counter . . . . .	12
3.6	Synthesis Flow . . . . .	14
3.7	Physical Design Flow . . . . .	15
4.1	Base Model of 64 BIT Counter[1] . . . . .	17
4.2	Simulated Output . . . . .	18
4.3	Synthesized Netlist . . . . .	19
4.5	Power Report of Base Model . . . . .	19
4.4	Area Report of Base Model . . . . .	20
4.6	Timing Report of Base Model . . . . .	20
5.1	Simulated Output of LFSR . . . . .	22
5.2	Simulated Output of 64 BIT Cascaded Counter . . . . .	23
5.3	Simulated Output of 64 BIT Cascaded Counter in Cadence .	23
5.4	Synthesized Netlist . . . . .	24
5.5	Area Report . . . . .	24

5.6	Power Report . . . . .	25
5.7	Timing Report . . . . .	25
6.1	Simulated Output of LFSR . . . . .	27
6.2	Simulated Output of 8 BIT Cascaded Counter . . . . .	28
6.3	Simulated Output of 8 BIT Cascaded Counter in Cadence . .	28
6.4	Synthesized Netlist . . . . .	29
6.5	Area Report . . . . .	29
6.6	Timing Report . . . . .	30
6.7	Power Report . . . . .	31
6.8	Floor Plan (a) . . . . .	32
6.9	Floor Plan (b) . . . . .	33
6.10	Power Plan . . . . .	34
6.11	Standard Cell Placement . . . . .	35
6.12	Layout after Standard Cell Placement . . . . .	35
6.13	CTS . . . . .	36
6.14	CTS Distribution . . . . .	37
6.15	Routing . . . . .	38
6.16	WNS (Setup) . . . . .	39
6.17	WNS (Hold) . . . . .	39
6.18	DRC Check . . . . .	41
6.19	LVS Check . . . . .	42
6.20	GDSII . . . . .	43

# Chapter 1

## INTRODUCTION

With the rapid advancement of digital electronics, especially in domains like communication systems, embedded devices, and high-performance computing, there is an increasing demand for high-speed counters. These counters play a critical role in ensuring accurate timing, synchronization, and control in modern digital systems. Traditional binary counters, while functionally adequate, often fall short in meeting the stringent performance and area constraints required in current technologies.

High-speed counters are versatile components that find utility across numerous application domains:

**Clock Division:** High-speed counters are commonly used to divide the frequency of a master clock signal, enabling the generation of lower-frequency clocks needed for subsystems with slower operating requirements.

**Timers:** Counters serve as the backbone of timer modules in microcontrollers and processors, facilitating precise time-delay generation, scheduling tasks, and timeout operations.

**Frequency Measurement:** Counters are essential in digital frequency meters, where they count the number of signal transitions in a given time interval to measure frequency accurately.

**Program Counters (PC):** In microprocessors and digital signal processors

(DSPs), the program counter keeps track of instruction execution by incrementing on each clock cycle or control signal.

**Bit Synchronization:** High-speed counters help maintain synchronization between transmitted and received data in serial communication systems, ensuring that bits are correctly aligned and interpreted.

**Random Number Generators:** Many hardware-based random number generators (RNGs) incorporate counters as a part of the entropy-generation mechanism, especially in pseudo-random number generation techniques.

**Automated Test Equipment (ATE):** In semiconductor and board-level testing, counters are used to sequence test vectors and collect response timings, enabling thorough and high-speed validation of digital systems.

## 1.1 Relevance

In modern VLSI (Very-Large-Scale Integration) design, the demand for high-speed counters has increased significantly due to the need for faster, more efficient digital systems. Traditional binary counters, although simple and functionally adequate, often suffer from limitations such as lower counting rates, high propagation delay, and increased area requirements, making them less suitable for high-frequency or low-power applications. As digital designs continue to scale, there is a growing shift towards more advanced counter architectures that can meet stringent performance and power constraints.

One of the most promising alternatives is the counter designed using a Galois Linear Feedback Shift Register (LFSR). This approach is highly relevant in modern digital circuit design because of its significant advantages in terms of speed, power efficiency, and reduced hardware complexity when compared

to conventional binary counters. The structure of a Galois LFSR allows bit-wise feedback and parallelism, which helps eliminate the long carry chains present in traditional counters, thereby reducing the critical path delay and allowing the circuit to operate at higher clock frequencies.

Galois LFSR-based counters are widely used across several domains, including cryptography, where they serve as compact and efficient pseudo-random sequence generators; error detection and correction systems, where they facilitate mechanisms such as cyclic redundancy checks (CRC); and in digital systems requiring randomization or variability, such as pseudo-random number generation. Due to their low switching activity and minimal hardware overhead, they are also highly suitable for power-sensitive applications, making them an ideal choice in portable and battery-powered devices.

Moreover, these counters are well-suited for implementation on ASIC and FPGA platforms. Their simplicity allows for highly optimized synthesis, resulting in reduced usage of logic resources and routing complexity. In addition, LFSR counters can be easily integrated with clock gating techniques, which further enhances power efficiency by minimizing unnecessary switching activity. Their ability to maintain high-frequency operation while consuming less area and power highlights their importance in modern VLSI designs.

## 1.2 Objective

The objective of this project is to design and implement a high-speed 8-bit counter using a Galois Linear Feedback Shift Register (LFSR) with state extension and to design a 64-bit optimized counter.

# Chapter 2

## LITERATURE REVIEW

This literature review provides an overview of the relevant research and its contributions to the project's goals.

In their work titled “64-bit High Speed Counter with Galois LFSR”, Rohith Bathula, Sunil Kumar Amgoth, and Deepak Ramesh Salgar (2024) proposed a novel architecture that integrates a Galois Linear Feedback Shift Register (LFSR) with conventional digital counting logic to achieve high-speed and resource-efficient counting. The design incorporates a T Flip-Flop, which functions either as a frequency divider or as a control mechanism to regulate the counting process. A 58-bit counter is employed to track events or timing information with high resolution, while the Galois LFSR enhances the randomness and efficiency of the overall system. The final 64-bit output, denoted as  $Q[63:0]$ , renders the design particularly well-suited for large-scale counting applications such as cryptographic systems, error detection and correction, and precision frequency measurement. The integration of the LFSR not only improves speed and reduces hardware complexity but also offers significant advantages in power efficiency, making the design relevant for modern high-performance VLSI systems [1].

In another significant contribution, Hyungjoon Bae, Yujin Hyun, Suchang

Kim, and Sangsoo Park (2023) proposed a High-Speed Counter with Novel LFSR State Extension, published in IEEE Transactions on Computers. Their architecture introduces a two-stage counter design composed of a Lower 6-bit Counter and a Higher 58-bit Counter. The lower counter utilizes a 6-bit Galois LFSR integrated with a state detection circuit that not only governs its own counting behavior but also controls the activation of the higher-stage counter. The higher 58-bit counter is designed as a pipelined binary counter, partitioned into 14-bit and 16-bit segments, which are selectively enabled by pipeline enable (PEN) signals. This staged and conditional activation mechanism ensures efficient power consumption and high-speed operation. The novel LFSR state extension enhances performance by optimizing both transition logic and enable signal generation, making the architecture particularly suitable for applications that require ultra-fast counting with minimal latency, such as digital signal processing and high-resolution time measurement [2].

Jayasurya K and Ajith R (2024), in their work “Area and Power Optimized RTL to GDS II Flow” presented at the 2024 5th International Conference on Circuits, Control, Communication and Computing, examined an optimized RTL-to-GDS II design flow targeting improvements in power efficiency and area reduction. Utilizing Cadence Genus for synthesis and Cadence Innovus for physical implementation, the authors used a telecommunication receiver core as a case study. The core integrated critical blocks such as a phase-locked loop, signal processing unit, and a power management module. Their optimized design achieved a 1730 micro-meter sq reduction in chip area, a 33 percent increase in positive slack (enhancing timing margins), and a 25 percent reduction in power consumption. This study demonstrates the effectiveness of a carefully tuned physical design methodology in meet-

ing aggressive design goals, particularly in communication-centric SoCs. The principles and tools applied here are directly relevant to the backend implementation of high-performance counters and FSMs where area and power efficiency are crucial design metrics [3].

Amit Shohal and Jasbir Kaur (2024), in their paper “Efficient RTL to GDS II Flow for Finite State Machine Integration: A Physical Design Approach” presented at the 2024 IEEE 5th India Council International Sub-sections Conference (INDISCON), focused on streamlining the backend implementation of digital designs, particularly finite state machines (FSMs). Their proposed methodology outlines a complete physical design flow beginning with RTL simulation using Cadence NC Launch to verify functional correctness. The synthesis phase is carried out using Cadence Genus, which transforms the RTL into a gate-level netlist and generates reports on power, area, and timing. The design targets a 90nm technology node and proceeds through placement and routing using Cadence Innovus. The culmination of the flow results in the generation of a GDS II file, ready for fabrication. This structured and tool-integrated approach emphasizes the importance of accurate verification, efficient resource utilization, and timing closure in achieving reliable physical design outcomes—principles that are critical in the development of high-performance digital counters and FSM-based control logic in VLSI systems [4].

# Chapter 3

## SYSTEM DESIGN

### 3.1 Base Model Design[1]

The block diagram represents a hierarchical digital system divided into two main subsections. Subsection-1 consists of a 6-bit Galois Linear Feedback Shift Register (LFSR), a T flip-flop, and a combination of logic gates. The LFSR generates pseudo-random or sequential bit patterns and is enabled by a control signal derived from the T flip-flop, which itself is driven by the CLOCK and COUNT signals through an OR gate. The LFSR produces six outputs ( $Q[0]$  to  $Q[5]$ ), which are processed through several AND gates to detect specific patterns or conditions. These outputs are then used to control the second part of the system.

Subsection-2 is a 58-bit binary counter that is segmented into four parts, likely of sizes 14, 14, 14, and 16 bits, respectively. The enable signals for these segments are derived from the logic in Subsection-1, ensuring that counting only occurs when certain LFSR conditions are met. This structure allows the wide binary counter to be selectively enabled, reducing unnecessary switching activity and potentially saving power. The combined output of the system

is a 64-bit value, where the lower 6 bits ( $Q[5:0]$ ) come from the LFSR and the upper 58 bits ( $Q[63:6]$ ) come from the binary counter. The system is globally controlled by RESET, CLOCK, and COUNT signals, providing reset functionality, timing, and counting control. Overall, this design likely serves as a low-power pseudo-random counter or pattern generator, suitable for applications in cryptography, testing, or digital system control.

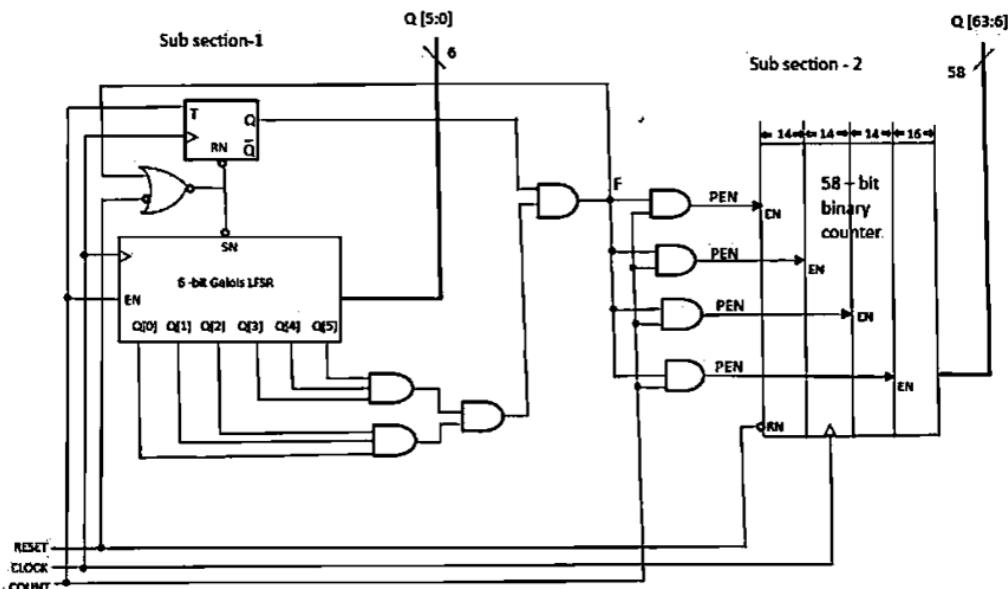


Figure 3.1: Base Model Design[1]

### 3.2 Proposed Design of 64 BIT Counter

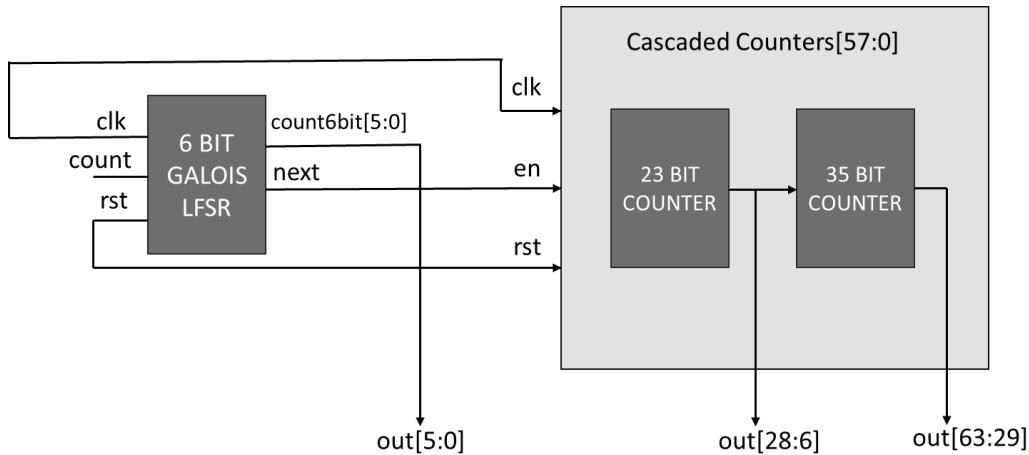


Figure 3.2: Proposed Design of 64 BIT Counter

The block diagram illustrates a compact and hierarchical 64-bit counting architecture that combines a pseudo-random number generator with cascaded counters for efficient large-range counting.

On the left side, the system begins with a 6-bit Galois Linear Feedback Shift Register (LFSR). This LFSR generates a sequence of pseudo-random values and plays a critical role in enabling or controlling the operation of the subsequent counter stages. It is often used for spreading the switching activity to reduce power consumption or for generating unique patterns for system control or testing.

On the right side, the diagram shows a block labeled Cascaded Counters [57:0], indicating a 58-bit counter implemented in two stages. The first stage is a 23-bit counter, and its overflow or terminal count output is used to enable the second stage—a 5-bit counter. These two counters are arranged

in a cascaded fashion, meaning the second counter increments only when the first one completes its full count cycle. This structure ensures an efficient division of counting logic and helps manage area and timing more effectively.

Together, the output from the LFSR (6 bits) and the cascaded counters (58 bits) forms a complete 64-bit output. This type of architecture is suitable for applications requiring wide counters with controlled or conditional enabling, such as low-power digital systems, hardware random number generation, or testing environments using pseudo-random sequences. The use of LFSR in the front stage also improves entropy and reduces power by avoiding constant toggling in large counters. The polynomial equation for 6 bit LFSR is  $(x)^6 + (x)^5 + 1$ .

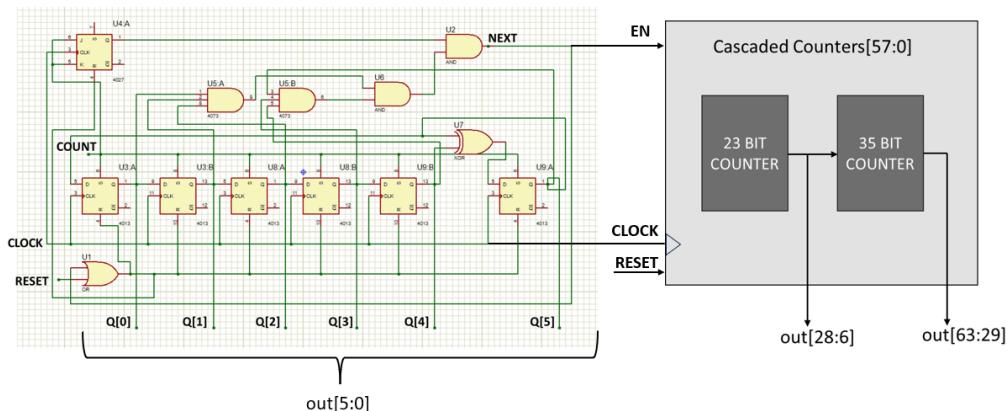


Figure 3.3: Circuit Diagram of 64 BIT Counter

### 3.3 Proposed Design of 8 BIT Counter

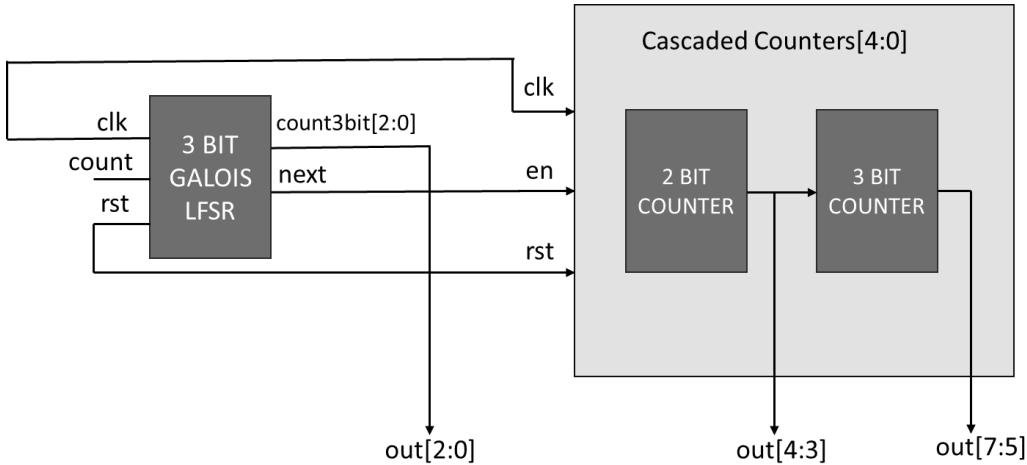


Figure 3.4: Proposed Design of 8 BIT Counter

The block diagram illustrates a compact, low-bit version of a pseudo-random enabled cascaded counter system, combining a Galois LFSR and hierarchical counters to produce a 5-bit output.

On the left, the 3-bit Galois LFSR is responsible for generating a pseudo-random sequence. This LFSR acts as a control logic block, likely used to gate or enable the operation of the downstream counters. The use of a Galois LFSR helps distribute switching activity, enhancing power efficiency and introducing controlled randomness in the count pattern.

On the right, within the Cascaded Counters [4:0] block, there are two counters arranged in a series: a 2-bit counter followed by a 3-bit counter. The 2-bit counter acts as the first stage, and its overflow (or terminal count) triggers the second stage—the 3-bit counter. This cascading method allows the design to count beyond the range of each individual counter while min-

imizing active switching. Together, these counters produce a 5-bit output, covering a counting range from 0 to 31.

The total system output is formed by concatenating the LFSR and counter outputs or using the LFSR solely as a control mechanism. This small-scale version of the earlier 64-bit design is suitable for applications in digital testing, simple random sequence generation, or energy-efficient low-bit-width counters in resource-constrained environments. The polynomial equation for 3 bit LFSR is  $(x)^3 + (x)^2 + 1$ .

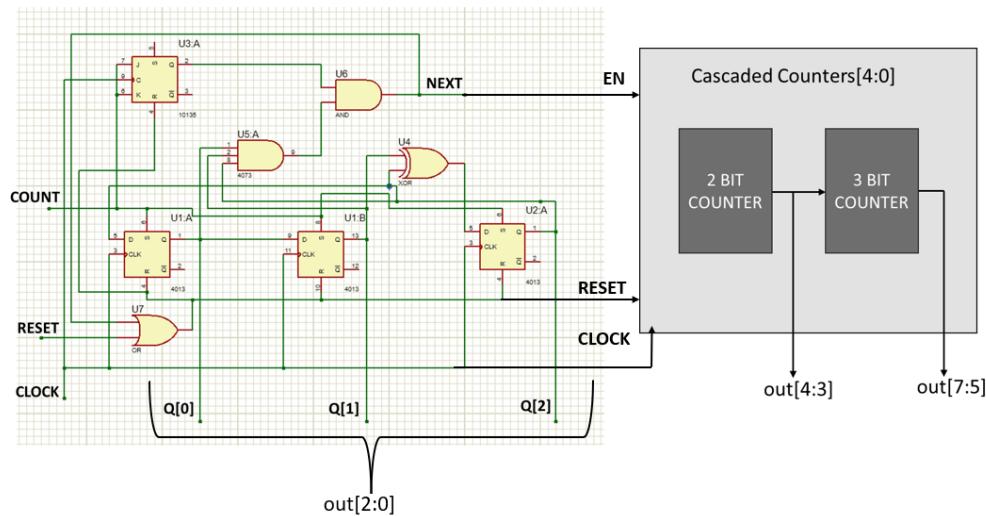


Figure 3.5: Circuit Diagram of 8 BIT Counter

### 3.4 RTL to GDSII Design Flow

The RTL to GDS (Register Transfer Level to GDSII) design flow is a comprehensive process used in digital VLSI (Very Large Scale Integration) design to transform a high-level hardware description into a final physical layout that can be fabricated on a silicon chip. The flow begins with RTL design,

where the functionality of the chip is described using hardware description languages like Verilog or VHDL. This RTL code is then verified through simulation to ensure correct logic behavior. Next, the design enters the synthesis phase, where the RTL code is converted into a gate-level netlist using standard cell libraries. This netlist is further subjected to Design for Testability (DFT) insertion to enhance test coverage. Following synthesis, the design proceeds to the physical design stage, starting with floorplanning (deciding the placement of blocks), placement (arranging standard cells), and clock tree synthesis (ensuring proper clock distribution). Then, routing is performed to connect the components using metal layers. After routing, the design undergoes physical verification steps such as DRC (Design Rule Check), LVS (Layout vs. Schematic), and timing analysis to ensure it meets all manufacturing and performance requirements. Once the design is finalized and verified, it is exported in GDSII format, which is the standard file format used by foundries to fabricate the chip. This RTL to GDSII flow ensures that a functional, manufacturable, and optimized chip layout is produced from the initial behavioral description.

### 3.5 Synthesis Flow

The synthesis flow in Cadence is a critical step in the digital design process, where high-level RTL (Register Transfer Level) code is converted into an optimized gate-level netlist using standard cell libraries. In Cadence tools such as Genus Synthesis Solution, the flow begins by importing the RTL code written in Verilog or VHDL, along with the design constraints specified in the SDC (Synopsys Design Constraints) format. These constraints typically include timing requirements, clock definitions, and I/O specifications. Once the RTL

is read and elaborated, the synthesis tool performs various optimizations to meet area, power, and timing goals. This includes logic optimization, technology mapping to standard cell libraries, and resource sharing. The tool generates a gate-level netlist that represents the circuit using components from the target technology library. GPDK90 is the library file used. At the end of the flow, the synthesized netlist is saved, along with constraint and report files, and passed on to the physical design tools such as Innovus for further implementation.

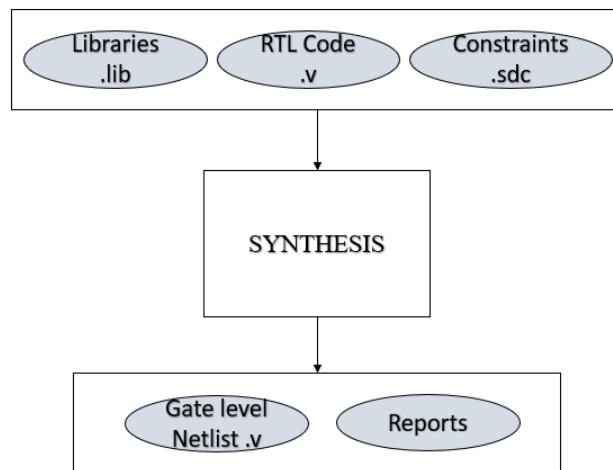


Figure 3.6: Synthesis Flow

## 3.6 Physical Design Flow

The physical design flow in Cadence, typically carried out using the Innovus Implementation System, involves transforming a synthesized gate-level netlist into a final layout ready for fabrication. This process begins after logic synthesis and focuses on placing and routing the actual physical components on a silicon chip while meeting design constraints for performance,

power, and area. The first step is floorplanning, where the overall chip architecture is defined, including the placement of major blocks (macros), I/O pads, and power structures. Next, power planning is done to create the power distribution network, ensuring reliable power delivery to all parts of the chip. Following this, placement arranges the standard cells from the netlist within the floorplan area, optimizing for area utilization and timing. After placement, clock tree synthesis (CTS) is performed to distribute the

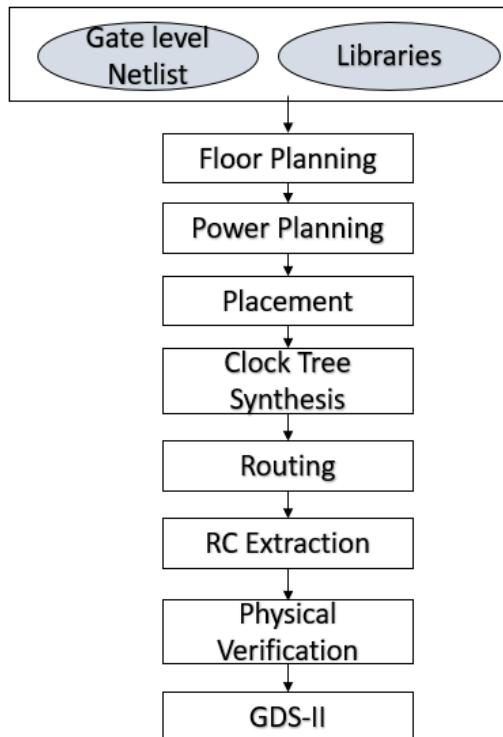


Figure 3.7: Physical Design Flow

clock signal uniformly and minimize skew across the design. Then, the design proceeds to routing, which connects all placed cells and macros according to the netlist using metal layers while avoiding congestion and meeting design rules. Once routing is completed, the design undergoes physical verification,

which includes checks like Design Rule Check (DRC) and Layout Versus Schematic (LVS) to ensure manufacturability and correctness. Additionally, Static Timing Analysis (STA) and signal integrity checks are performed to validate timing closure and electrical performance. After passing all verification steps, the final layout is exported in GDSII format, the standard used for chip fabrication.

### 3.7 Verification Platform

The RTL design and verification were carried out using Verilog, with simulation performed using Cadence Incisive, synthesis using Cadence Genus, and physical design using Cadence Innovus.

# Chapter 4

## Implementation of Base Model 64 BIT Counter[1]

The circuit was implemented using Xilinx Vivado, Simulated in Cadence nclauch, Synthesized using Cadence Genus.

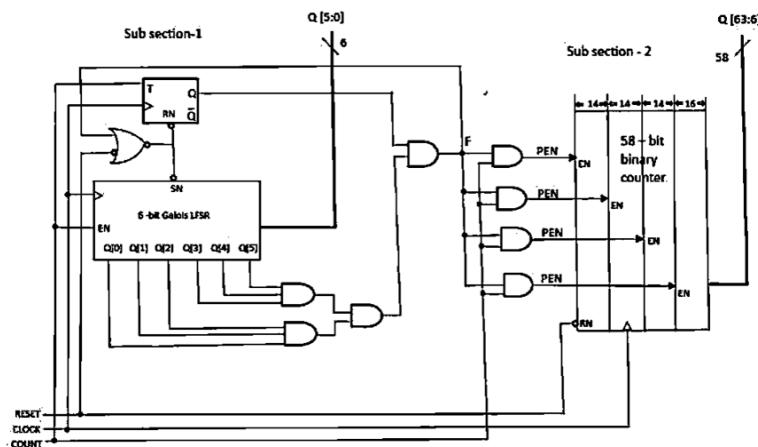


Figure 4.1: Base Model of 64 BIT Counter[1]

## 4.1 Simulation Result

The Simulated output of Base model Counter design is done with Cadence incisive tool.

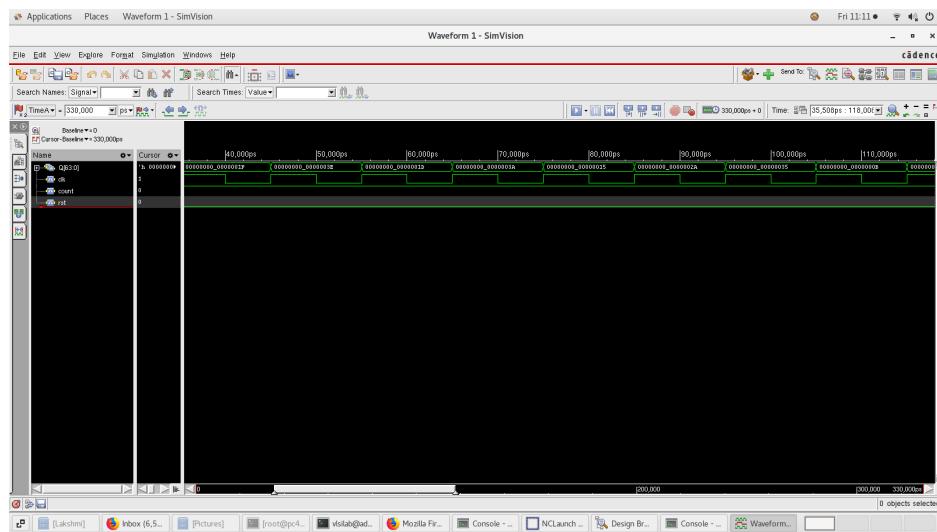


Figure 4.2: Simulated Output

## 4.2 Synthesis Report

In the base model design of 64 BIT counter, the synthesis is done in Cadence Genus, the synthesized Netlist is observed along with Area, Power and Timing Reports.

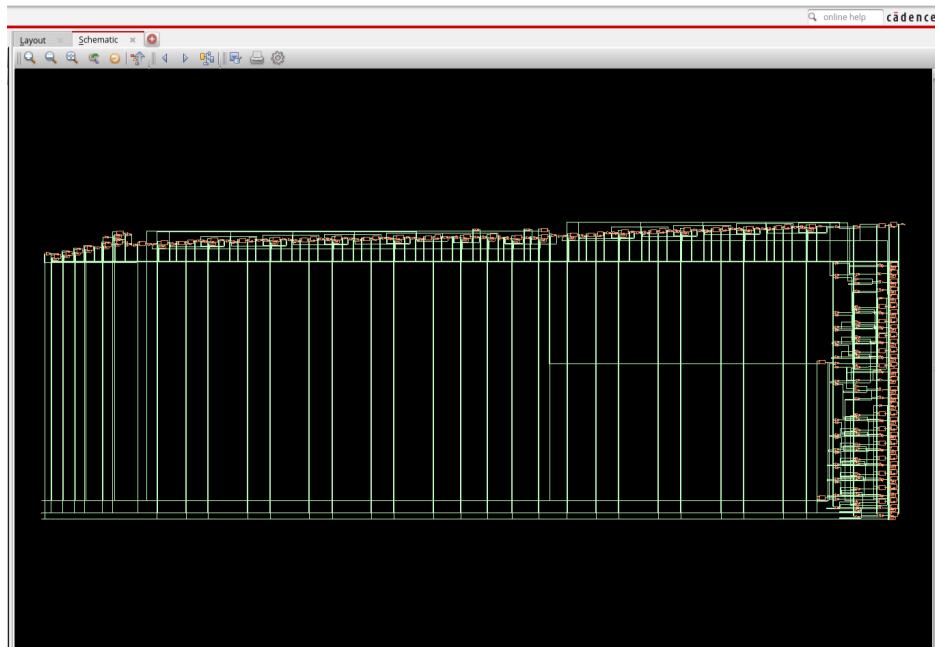


Figure 4.3: Synthesized Netlist



Figure 4.5: Power Report of Base Model

```

root@adrecolab:/home/lakshmi/Desktop/test
File Edit View Search Terminal Help
OAII211X1      8    42.386  slow
OAII211XL     16   84.773  slow
OAII21X1       2    9.083  slow
OAII2BB1XL    24  127.159  slow
OR2X1          1    4.541  slow
OR4X1          2   13.624  slow
SDFFRHDX1    35  874.219  slow
SDFFSHQX1     6  163.498  slow
XOR2XL         1    8.326  slow
-----
total           253 2352.445

      Type      Instances     Area     Area %
-----
sequential      69 1594.788  67.8
inverter        32   72.662   3.1
logic           152 684.995  29.1
physical_cells    0   0.000   0.0
-----
total           253 2352.445 100.0
genus:root: 11>

```

Figure 4.4: Area Report of Base Model

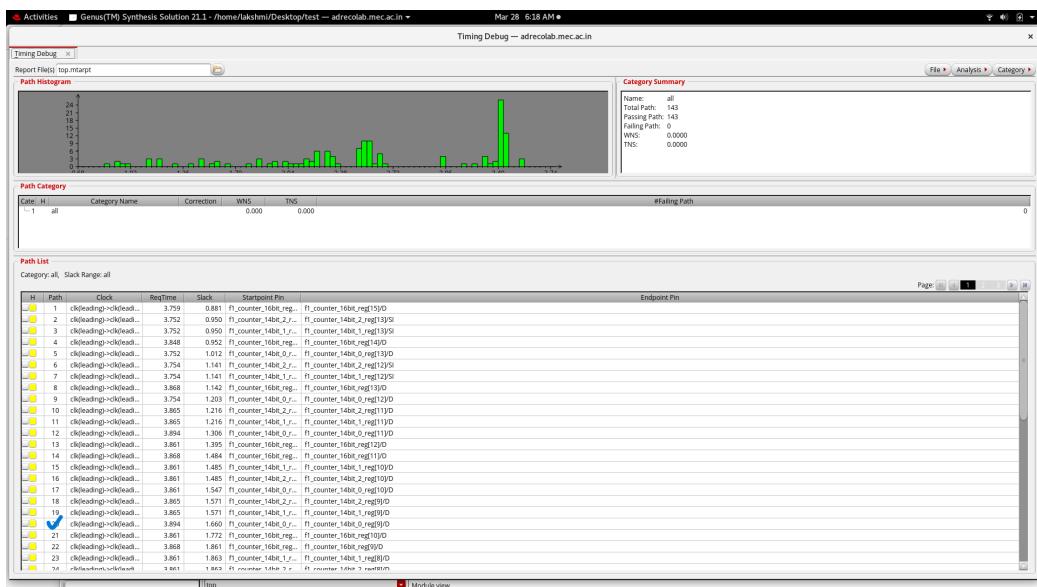


Figure 4.6: Timing Report of Base Model

### 4.3 Inference

Based on the synthesis and timing analysis results, the design demonstrates efficient power, area, and performance characteristics. The total power consumption is 743,549.755 nW, indicating low power operation suitable for

energy-efficient applications. The design occupies a compact area of 2352.445  $\mu\text{m}^2$ , which reflects good utilization of silicon resources. The maximum required time is 3.894 ns, and the design achieves a positive slack of 1.66 ns, confirming that all timing constraints are comfortably met. Overall, the results suggest that the design is well-optimized in terms of power, area, and timing performance.

# Chapter 5

## Implementation of Proposed 64 BIT Counter

In the proposed design of 64 BIT counter, the Verilog code is written and simulated in Xilinx Vivado, Cadence Incisive, Synthesis is performed using Cadence Genus. Analysed the Power, Area and Timing Reports.

### 5.1 Simulation Report

Simulation is performed in Xilinx Vivado and Cadence Incisive.



Figure 5.1: Simulated Output of LFSR

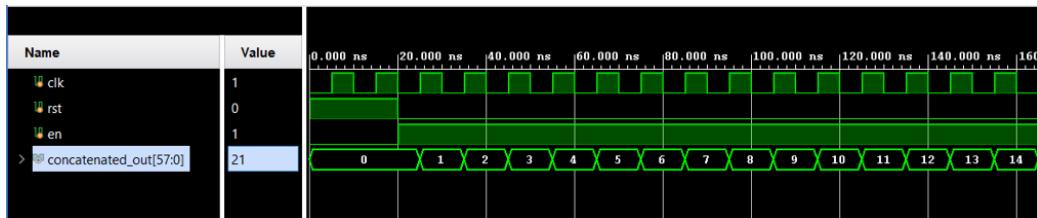


Figure 5.2: Simulated Output of 64 BIT Cascaded Counter

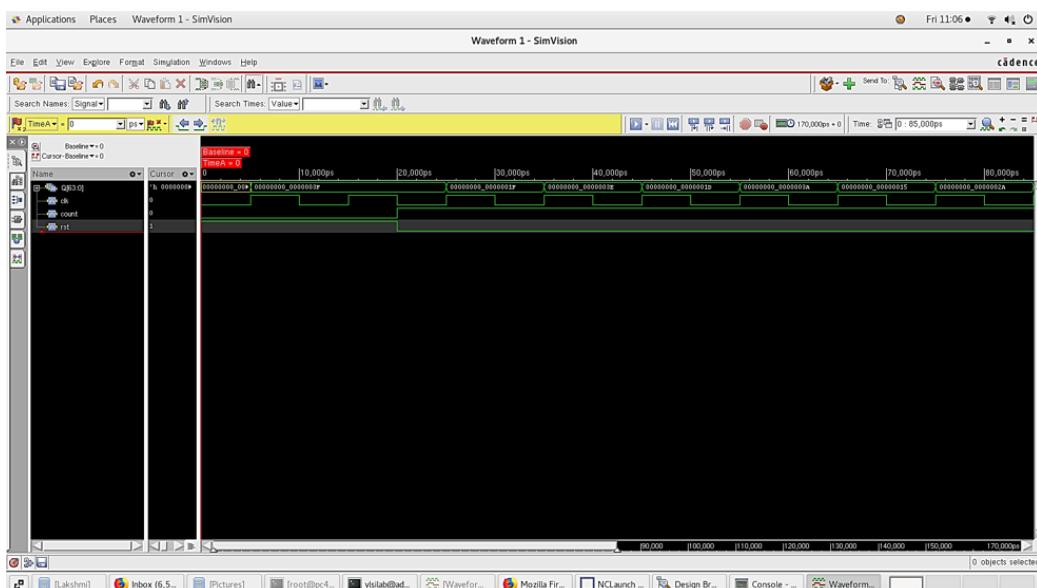


Figure 5.3: Simulated Output of 64 BIT Cascaded Counter in Cadence

## 5.2 Synthesis Report

In the proposed design of 64 BIT counter, the synthesis is done in Cadence Genus, the synthesized Netlist is observed along with Area, Power and Timing Reports.



Figure 5.4: Synthesized Netlist

```

root@adrecolab:/home/lakshmi/Desktop/test
File Edit View Search Terminal Help
INVX1          2    4.541   slow
INVXL          1    2.271   slow
MXI2XL          5    30.276  slow
NAND2BXL        6    27.248  slow
NAND2XL        53   160.463 slow
NAND3XL         3    13.624  slow
NAND4XL         2    10.597  slow
NOR2BX1         1    4.541   slow
NOR2BXL         4    18.166  slow
NOR2XL          11   33.304  slow
NOR3X1          1    4.541   slow
NOR4X1          4    24.221  slow
OA21X1          18   122.618 slow
OA21XL          3    20.436  slow
OAI21XL         1    5.298   slow
OAI21X1         10   45.414  slow
OAI21XL         11   49.955  slow
OR2X1           1    4.541   slow
OR4XL            1    6.812   slow
SDFFRHQX1       58   1448.707 slow
XNOR2XL         1    8.326   slow
-----
total           239  2319.898

      Type     Instances   Area   Area %
sequential        65 1559.971  67.2
inverter          13  29.519   1.3
logic             161 730.409  31.5
physical_cells     0   0.000   0.0
-----
total           239  2319.898 100.0

@genus:root: 10>

```

Figure 5.5: Area Report

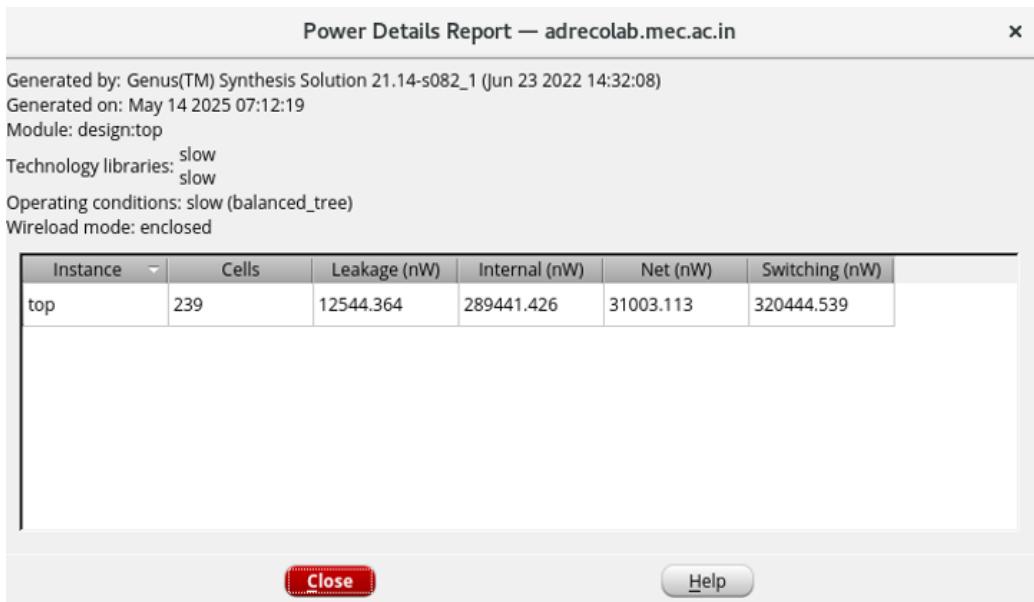


Figure 5.6: Power Report

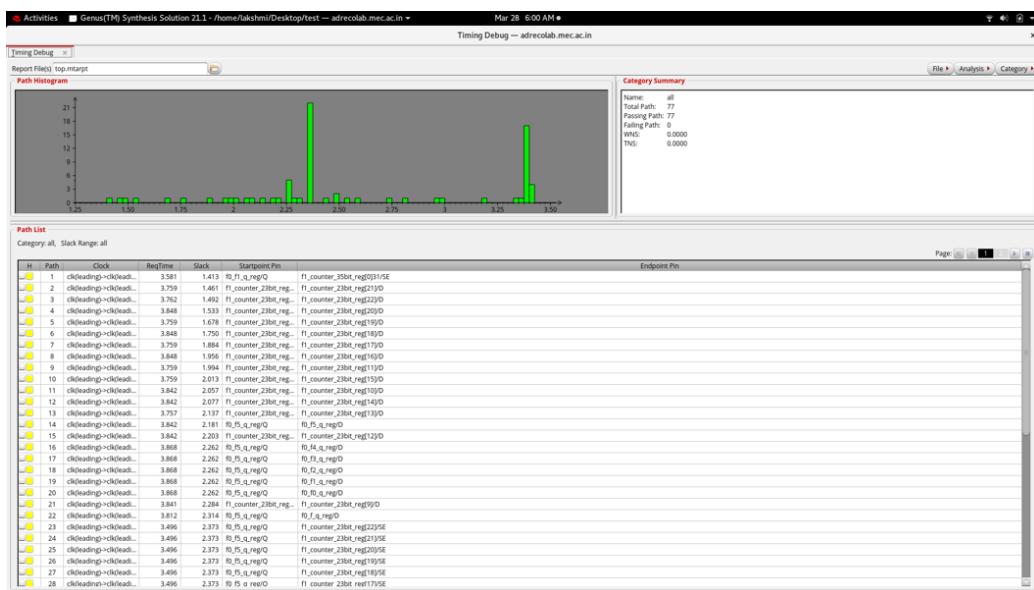


Figure 5.7: Timing Report

### 5.3 Inference

The design analysis indicates strong performance and optimization across key metrics. The maximum required time is 3.868 ns, and a positive slack of 2.262 ns confirms that the timing requirements are not only met but exceeded, ensuring reliable high-speed operation. The area utilization is 2319.898  $\mu\text{m}^2$ , reflecting an efficient layout with minimal silicon footprint. Additionally, the total power consumption is 653,433.442 nW, highlighting the design's suitability for low-power applications. Overall, the results demonstrate that the design is well-balanced in terms of timing, area, and power efficiency.

# Chapter 6

## RTL to GDSII Implementation of Proposed 8 BIT Counter

In the proposed design of 8 BIT counter, the Verilog code is written and simulated in Xilinx Vivado, Cadence Incisive, Synthesis is performed using Cadence Genus, Analysed the Power, Area and Timing Reports. Physical Design is done with Cadence Innovus.

### 6.1 Simulation Report

Simulation is performed in Xilinx Vivado and Cadence Incisive.

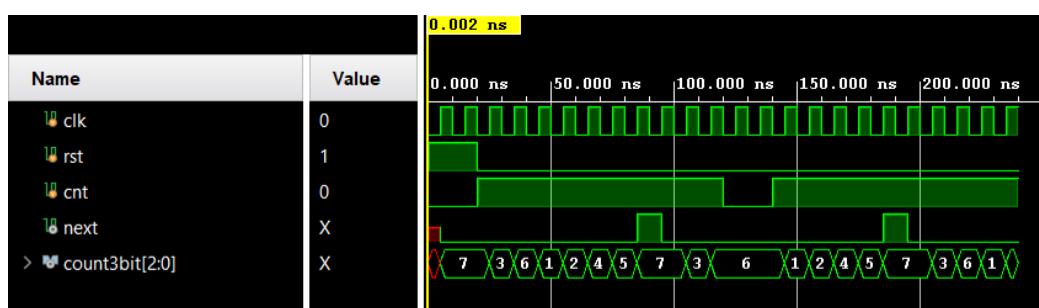


Figure 6.1: Simulated Output of LFSR

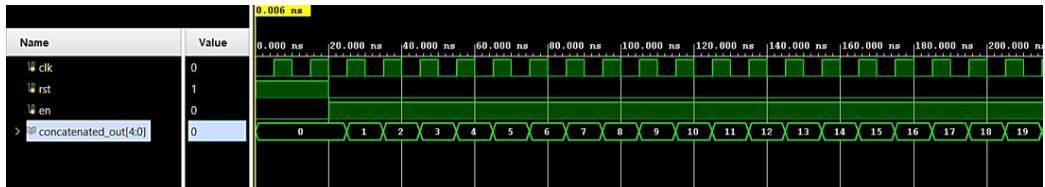


Figure 6.2: Simulated Output of 8 BIT Cascaded Counter

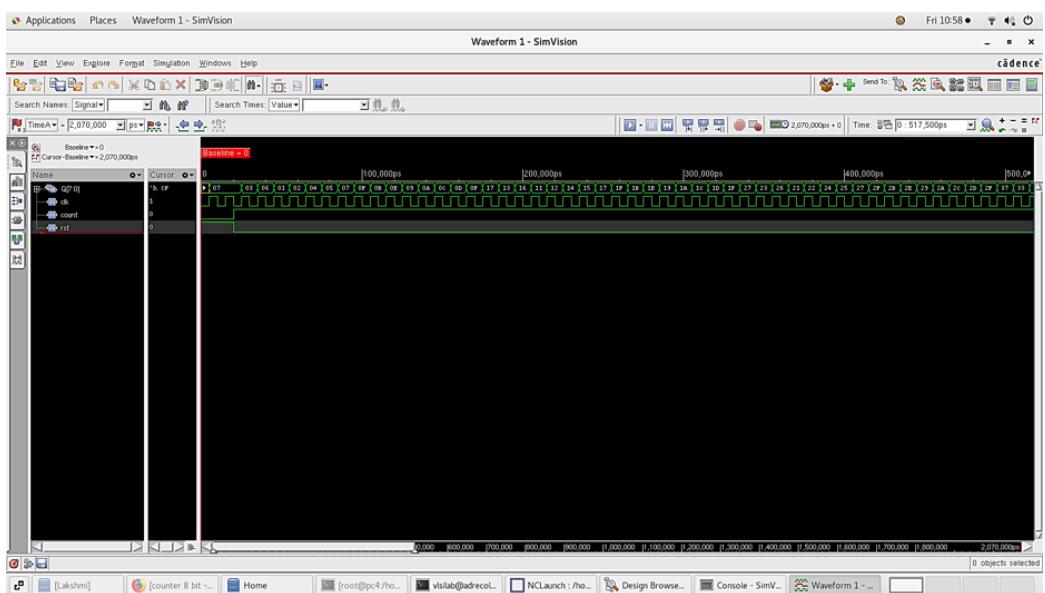


Figure 6.3: Simulated Output of 8 BIT Cascaded Counter in Cadence

## 6.2 Synthesis Report

In the proposed design of 8 BIT counter, the synthesis is done in Cadence Genus, the synthesized Netlist is observed along with Area, Power and Timing Reports.



Figure 6.4: Synthesized Netlist

**Gate Count Report — adrecolab.mec.ac.in**

---

Generated by: Genus(TM) Synthesis Solution 21.14-s082\_1 (Jun 23 2022 14:32:08)  
 Generated on: Mar 17 2025 07:42:29  
 Module: design:top  
 Technology library: slow  
 Operating conditions: slow (balanced\_tree)  
 Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload
top	32	283.837	0.000	283.837	

Filtering

**Close**      **Help**

Figure 6.5: Area Report

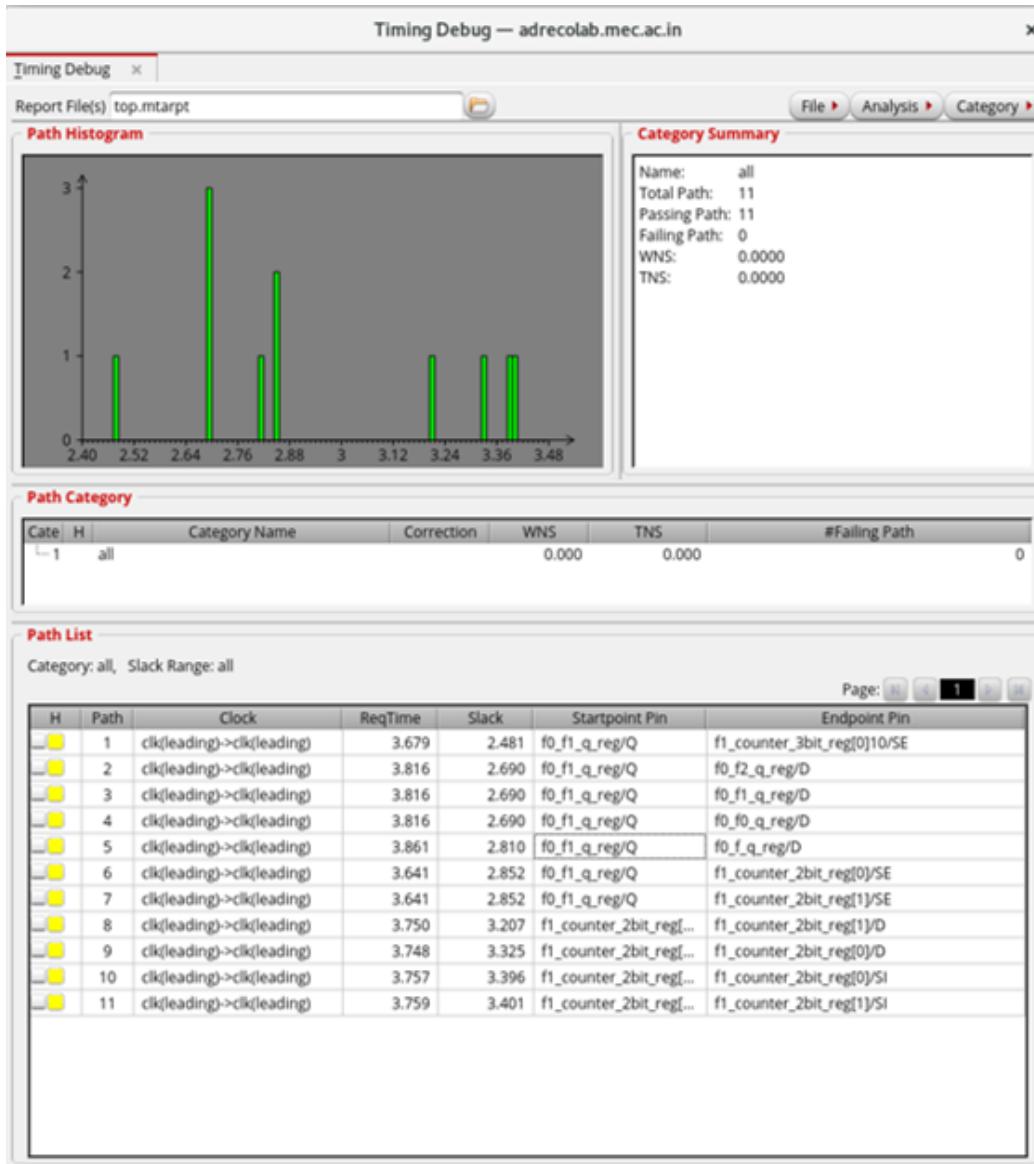


Figure 6.6: Timing Report

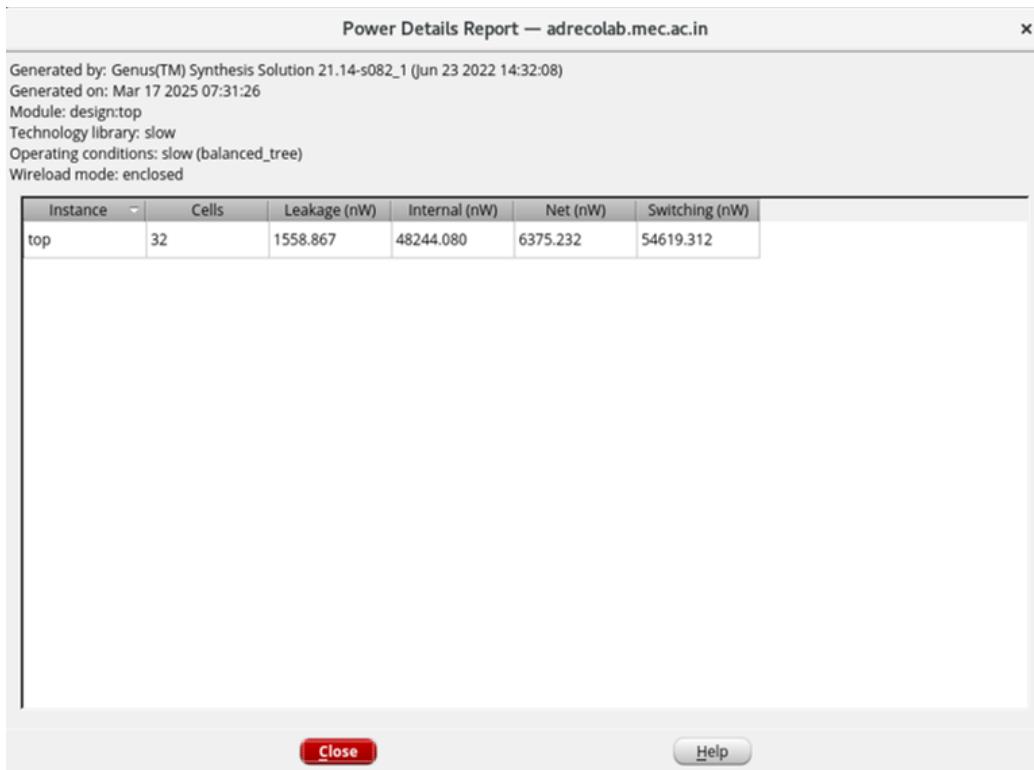


Figure 6.7: Power Report

### 6.3 Inference

The synthesis and timing analysis of the 8-bit counter indicate a highly efficient design. The total power consumption is 110,797.491 nW, showcasing low power usage suitable for power-sensitive applications. The design occupies a small area of 283.837  $\mu\text{m}^2$ , demonstrating excellent resource efficiency. The maximum required time is 3.816 ns, and a positive slack of 2.861 ns confirms that the design comfortably meets all timing constraints. These results suggest that the 8-bit counter is optimized for compact, low-power, and high-speed performance, making it ideal for integration into larger digital systems.

## 6.4 Physical Design

Physical design in Cadence Innovus is the process of converting a synthesized digital circuit into a physical layout that can be fabricated on silicon. It involves several key steps such as floorplanning, placement, clock tree synthesis (CTS), routing, and physical verification.

### 6.4.1 Floor Plan

Floorplanning is a crucial step in the physical design process where the overall structure and layout of the chip are defined. It involves arranging major components such as macros, standard cell areas, I/O pads, and power grids within the chip's core area. The goal is to optimize placement for area efficiency, performance, and routing ease. Good floorplanning ensures minimal congestion, balanced utilization, and effective clock and power distribution, which directly impacts the overall timing, power, and area of the final design.

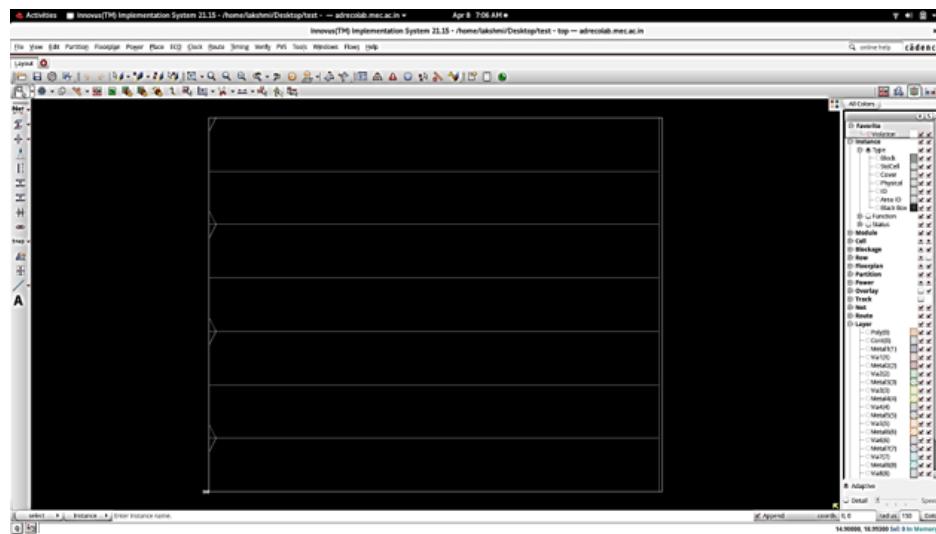


Figure 6.8: Floor Plan (a)

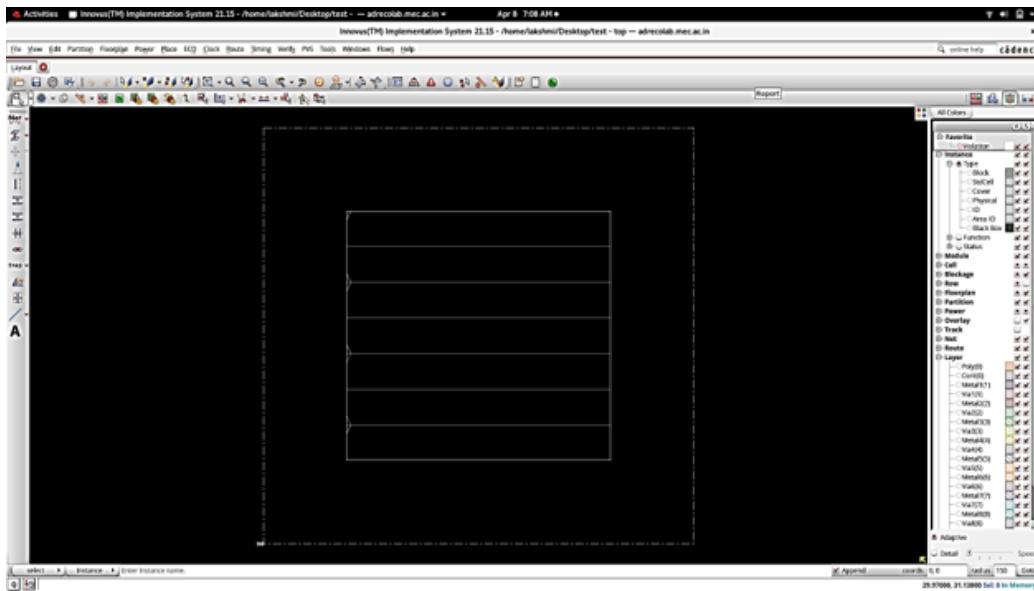


Figure 6.9: Floor Plan (b)

#### 6.4.2 Power Plan

Power planning is a critical step in the physical design flow that ensures reliable and efficient power delivery to all parts of the chip. During this stage, power and ground (P/G) networks are created, including power rings, straps, and vias, to distribute power from the input pads to the internal logic blocks. The objective is to minimize IR drop, electromigration, and voltage fluctuations, which can affect circuit performance and reliability. A well-designed power plan supports current demands of the chip while maintaining signal integrity and ensuring the design meets power-related constraints.

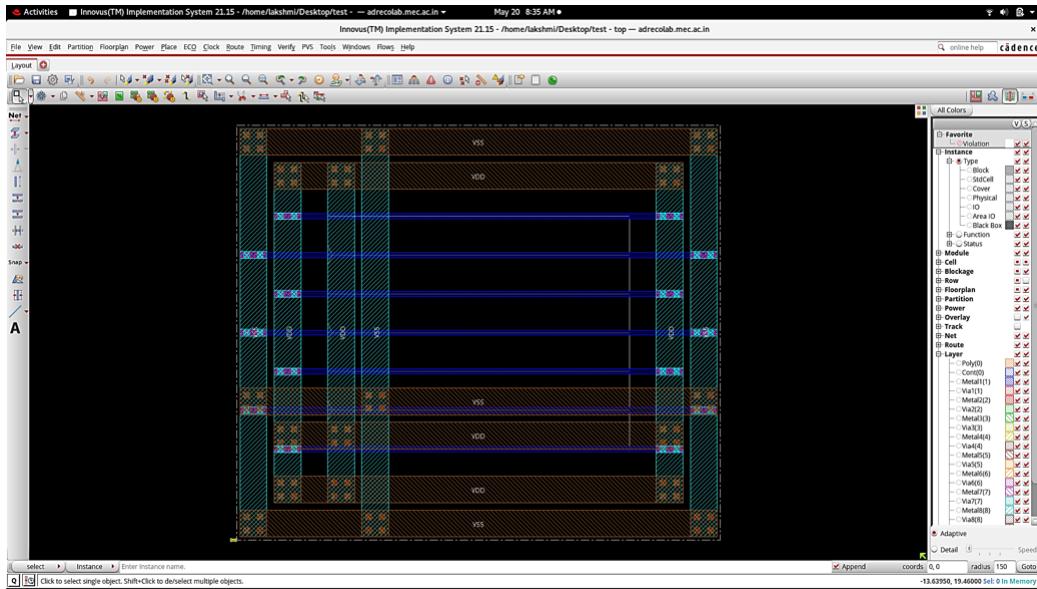


Figure 6.10: Power Plan

### 6.4.3 Placement

Standard cell placement in Cadence Innovus is the process of arranging the synthesized logic gates (standard cells) within the defined core area of the chip. This step comes after floorplanning and power planning. The goal of placement is to position the cells in a way that minimizes wirelength, timing delays, and congestion, while ensuring that the design meets area and power constraints. Innovus uses advanced optimization algorithms to achieve efficient placement, taking into account factors like timing critical paths, clock domains, and cell utilization. A good standard cell placement forms the foundation for successful clock tree synthesis and routing, leading to better overall performance and manufacturability.

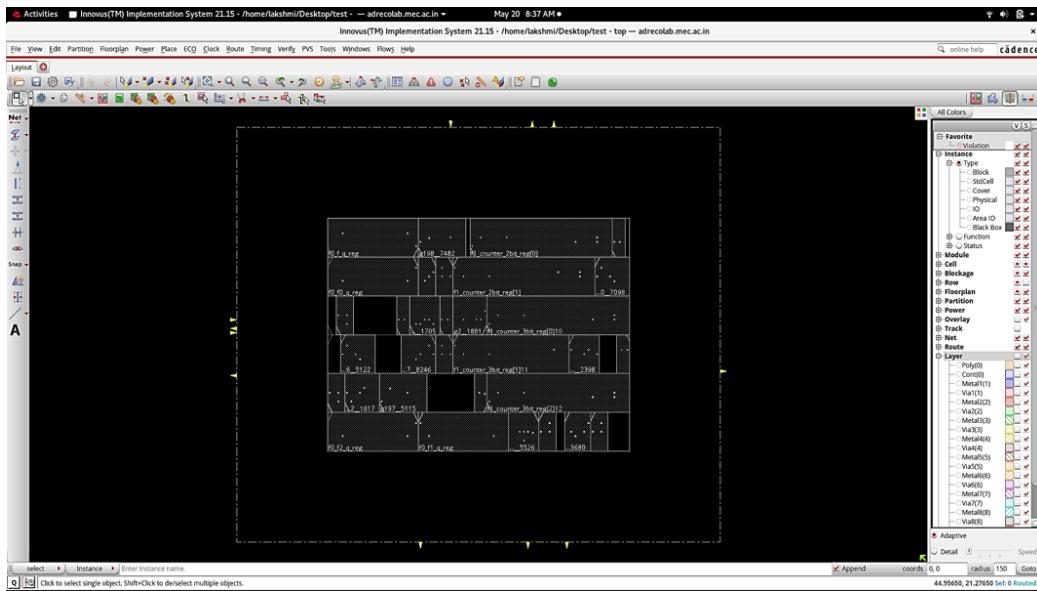


Figure 6.11: Standard Cell Placement

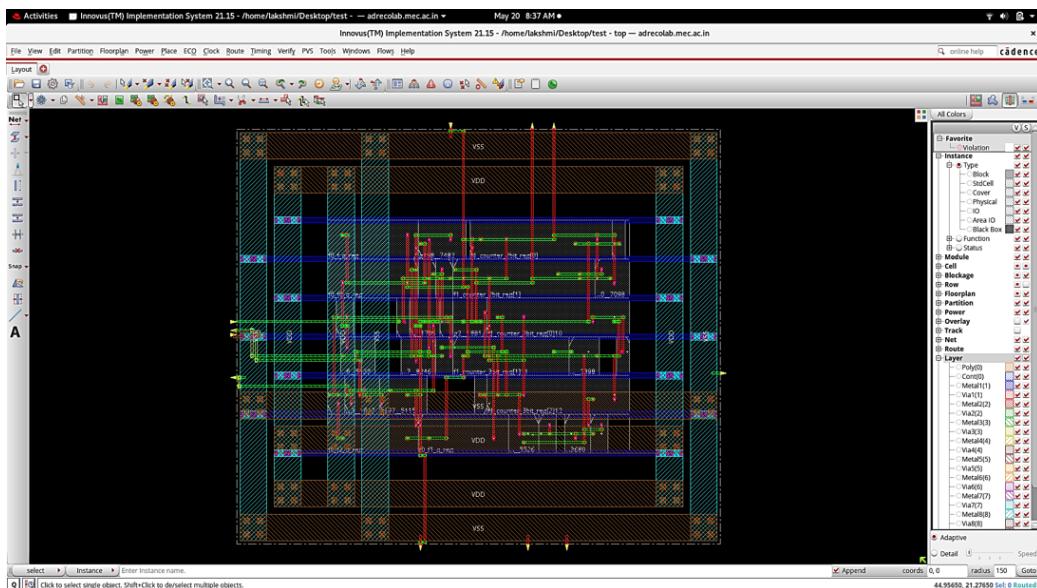


Figure 6.12: Layout after Standard Cell Placement

#### 6.4.4 Clock Tree Synthesis

Clock Tree Synthesis (CTS) is a key step in the physical design flow where the clock signal is distributed evenly to all sequential elements, such as flip-flops and latches, in the design. The main goal of CTS is to minimize clock skew and clock latency while ensuring proper timing synchronization across the chip.



Figure 6.13: CTS

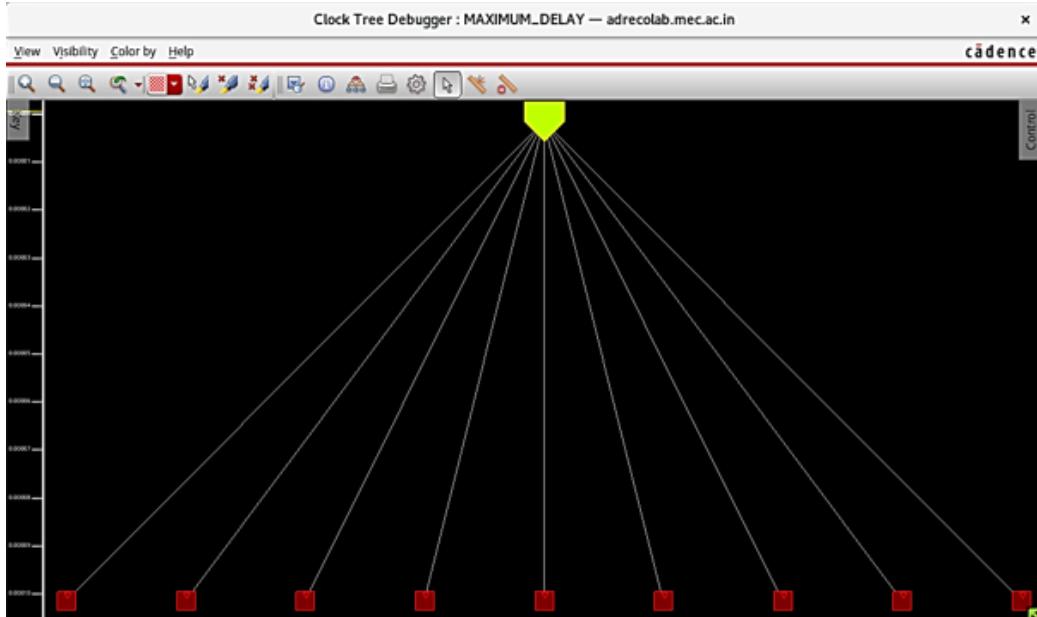


Figure 6.14: CTS Distribution

#### 6.4.5 Routing

Routing is a crucial stage in the physical design process where electrical connections are established between placed standard cells, macros, and I/O pins according to the netlist. It involves creating metal wires across multiple layers to complete the signal paths defined during logic synthesis. In Cadence Innovus, routing is performed in two phases: global routing, which plans approximate paths to minimize congestion, and detailed routing, which finalizes the exact wire paths while adhering to design rules such as spacing, width, and via usage. The objective of routing is to ensure signal integrity, minimize resistance and capacitance, and meet timing and reliability constraints. Efficient routing directly impacts the chip's performance, power consumption, and manufacturability, making it a vital step before physical verification and tape-out.

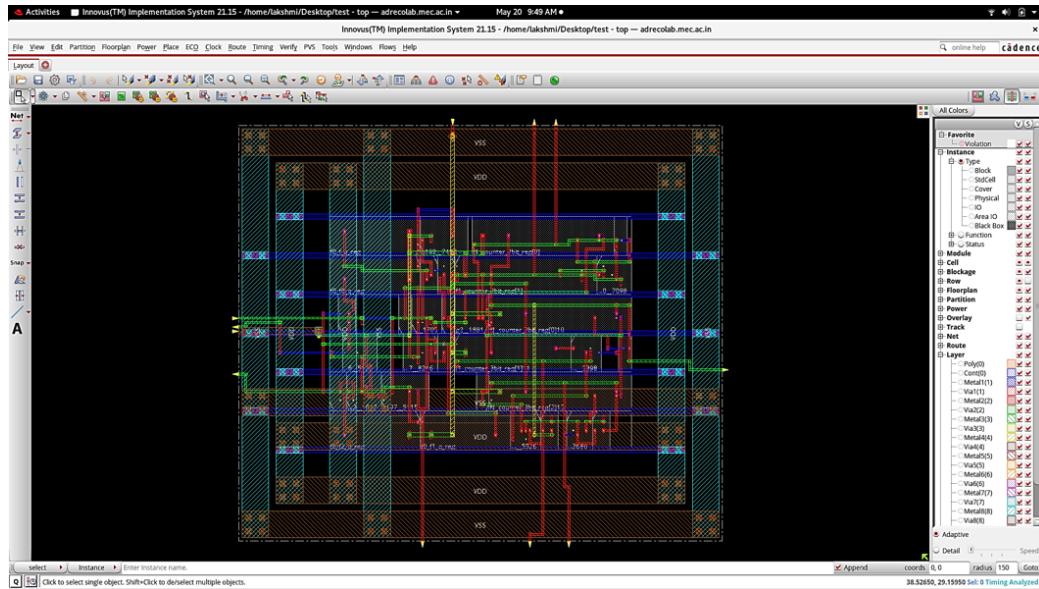


Figure 6.15: Routing

## STA

Timing reports are essential in the physical design process to ensure that a digital circuit meets all its timing requirements. Two key aspects of timing analysis are setup time and hold time, which ensure proper data transfer between sequential elements like flip-flops. Setup time is the minimum amount of time before the clock edge that the data input must be stable. A setup violation occurs when data arrives too late, potentially leading to incorrect operation. In a timing report, setup analysis checks if the data can propagate from one register to another within the clock period, considering delays through combinational logic and clock tree. Hold time is the minimum amount of time after the clock edge that the data must remain stable. A hold violation happens when data changes too soon, before being properly latched. Hold analysis ensures that data does not arrive too early at the destination flip-flop.

WNS (setup) = 2.350

WNS (hold) = 0.012

Violating path = 0

```

-----  

      timeDesign Summary  

-----  

Setup views included:  

WORSTCASE  

-----  

+-----+-----+-----+-----+
| Setup mode | all | reg2reg | default |
+-----+-----+-----+-----+
| WNS (ns): | 2.350 | 2.350 | 2.940 |
| TNS (ns): | 0.000 | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 | 0 |
| All Paths: | 14 | 11 | 7 |
+-----+-----+-----+-----+
  

+-----+-----+-----+
| | Real | | Total |
DRV | | | |
+-----+-----+-----+
| | Nr nets(terms) | Worst Vio | Nr nets(terms) |
+-----+-----+-----+
| max_cap | 0 (0) | 0.000 | 0 (0) |
| max_tran | 0 (0) | 0.000 | 0 (0) |
| max_fanout | 0 (0) | 0 | 0 (0) |
| max_length | 0 (0) | 0 | 0 (0) |
+-----+-----+-----+
  

Density: 90.714%  

-----  

Reported timing to dir timingReports  

Total CPU time: 0.67 sec  

Total Real time: 2.0 sec  

Total Memory Usage: 1902.148438 Mbytes

```

Figure 6.16: WNS (Setup)

```

-----  

      timeDesign Summary  

-----  

Hold views included:  

BESTCASE  

-----  

+-----+-----+-----+-----+
| Hold mode | all | reg2reg | default |
+-----+-----+-----+-----+
| WNS (ns): | 0.012 | 0.012 | 0.000 |
| TNS (ns): | 0.000 | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 | 0 |
| All Paths: | 11 | 11 | 0 |
+-----+-----+-----+
  

Density: 90.714%  

-----  

Reported timing to dir timingReports  

Total CPU time: 0.71 sec  

Total Real time: 1.0 sec  

Total Memory Usage: 1860.019531 Mbytes

```

Figure 6.17: WNS (Hold)

#### 6.4.6 DRC and LVS Check

DRC check, or Design Rule Check, is a critical step in the physical verification process to ensure that the layout of a chip adheres to the manufacturing rules defined by the semiconductor foundry. These rules include constraints on parameters such as minimum spacing between wires, minimum width of metal layers, via sizes, and enclosure rules. The purpose of DRC is to identify any violations that could cause fabrication issues or lead to chip failure. In Cadence Innovus, DRC is performed after the routing stage using built-in rule decks provided by the foundry. The tool highlights any violations so that the designer can correct them before finalizing the layout. A clean DRC report, with zero violations, is essential before generating the final GDSII file for tape-out, ensuring that the chip can be manufactured reliably and within the foundry's specifications.

LVS, or Layout Versus Schematic, is a key step in the physical verification process that ensures the physical layout of a chip matches its original schematic or netlist. It checks whether the connectivity and components in the layout correspond exactly to those defined in the logical design after synthesis. In Cadence tools, LVS is performed after routing using rule files provided by the foundry. The tool compares the layout's extracted netlist with the schematic netlist and reports any mismatches, such as missing or extra connections, incorrect device parameters, or differences in component count. A successful LVS check confirms that the layout accurately implements the intended design functionality. Passing LVS is essential before tape-out to guarantee the manufactured chip behaves as expected.

PVS 22.20-64b Reports: Done [DRC] DRC\_Las...

[LVS] LVS\_LAST [DRC] DRC\_Last

```

DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
DFM FILL: Cumulative Time CPU = 0(s) REAL = 0(s)

Total CPU Time : 1(s)
Total Real Time : 5(s)
Peak Memory Used : 27(M)
Total Original Geometry : 2718(6758)
Total DRC RuleChecks : 807
Total DRC Results : 0 (0)
Summary can be found in file top.sum
ASCII report database is /home/lakshmi/Desktop/test/DRC_Last/top.drc_errors.ascii
Checking in all SoftShare licenses.

Design Rule Check Finished Normally. Tue May 20 10:14:16 2025

```

Find  Prev Next  Match case  Whole word  Use RegExp  Highlight

No errors found	Warnings: 9	Info: 809
<b>⚠ [WARN]</b> Command CELL_CASE at line 53 in file /home/install/Physical_Verification/PVS_90nm/PVS_RULES/drc.pvl is obsolete. <b>⚠ [WARN]</b> VIRTUAL_CONNECT -name at line 57 in file /home/install/Physical_Verification/PVS_90nm/PVS_RULES/drc.pvl is skipped. It is set in control file.		

Find  Prev Next  Match case  Whole word  Use RegExp

**Help** Issues  Warnings  Info ReRun ReSubmit Close Report Help

/home/lakshmi/Desktop/test/DRC\_Last

Figure 6.18: DRC Check

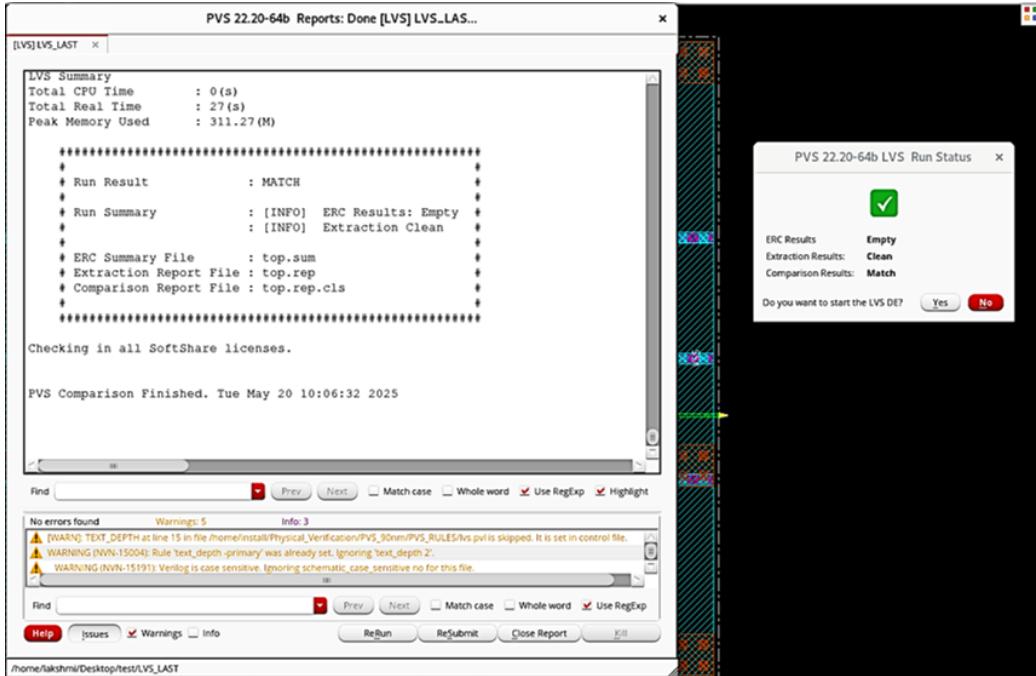


Figure 6.19: LVS Check

#### 6.4.7 GDSII

GDSII, which stands for Graphic Data System II, is the standard file format used to represent the physical layout of an integrated circuit for fabrication. It contains detailed information about the chip's geometry, including layers, polygons, vias, and placement of cells, as defined during the physical design process. In Cadence Innovus, once all verification steps like DRC and LVS are passed, the final layout is exported as a GDSII file. This file is then sent to the semiconductor foundry for mask generation and chip fabrication. GDSII acts as the bridge between design and manufacturing, ensuring that the layout data is accurately transferred to the fabrication process. The actual GDSII file can be visible from the fabrication, this image depicted is the format of GDSII file opened in text editor, Gedit.

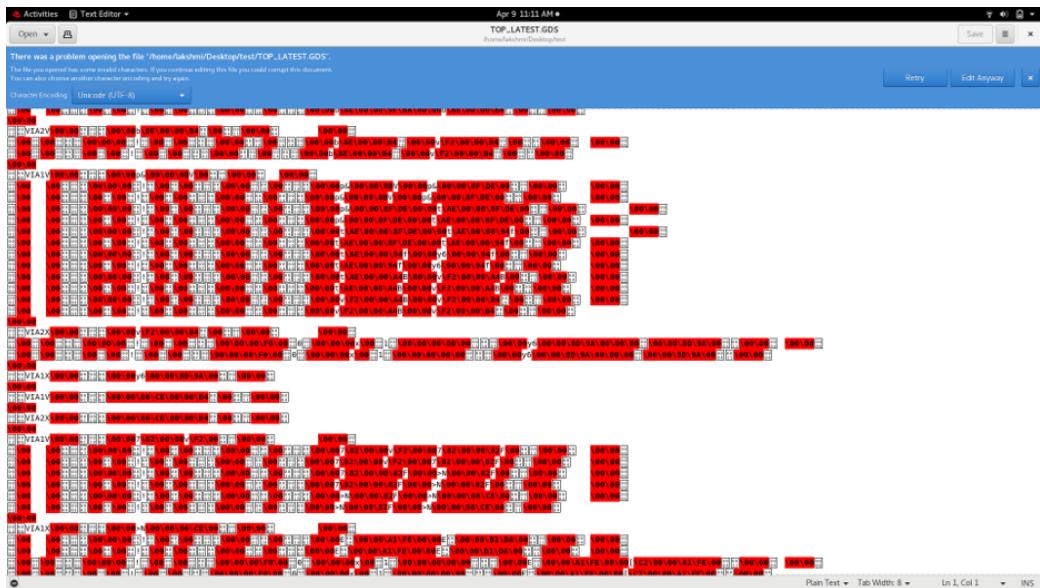


Figure 6.20: GDSII

# Chapter 7

## Results and Conclusion

The comparison between the existing and proposed 64-bit designs reveals significant improvements in the proposed architecture across key performance metrics. The proposed design demonstrates a notable reduction in power consumption, decreasing from 743,549.755 nW in the existing design to 653,433.442 nW, reflecting a savings of approximately 12.1 percent. Additionally, the silicon area utilization has been optimized, with the proposed design occupying 2319.898 cells compared to 2352.445 cells in the existing design, resulting in a reduction of around 1.38 percent. In terms of performance, the proposed design achieves a slight speed enhancement, reducing the maximum required time from 3.894 ns (256 MHz) to 3.868 ns (258.53 MHz). Most importantly, the Power-Area Product, a key indicator of overall design efficiency, has improved significantly—from 1,74,91,59,903.4 to 1,51,58,98,935.2—indicating a gain of approximately 13.33 percent. These improvements collectively highlight the proposed design's superiority in terms of power efficiency, area optimization, and overall performance.

Parameter	Existing Design (64-bit)	Proposed Design (64-bit)
Power (nW)	743549.755	653433.442
Area (cells)	2352.445	2319.898
Speed (Max Required Time)	3.894 ns (256 MHz)	3.868 ns (258.53 MHz)
Power-Area Product	1,74,91,59,903.4	1,51,58,98,935.2

Table 7.1: Performance Comparison of Existing[1] and Proposed 64-bit Designs

Parameter	Percentage Improvement
Power Reduction	12.1% ↓
Area Reduction	1.38% ↓
Speed Increase	0.67% ↑
Power-Area Product Reduction	13.33% ↓

Table 7.2: Percentage Improvement in Proposed 64-bit Design over Existing[1] Design

# Chapter 8

## Future Scope

The use of Galois Linear Feedback Shift Register (LFSR) counters presents several promising directions for future applications across diverse technological domains:

- *Ultra-low power IoT devices*: Galois LFSR counters can be optimized for minimal power consumption, making them highly suitable for battery-operated and energy-harvesting devices where energy efficiency is critical.
- *High-performance computing (HPC)*: These counters can be utilized in high-speed data processing systems, including network packet tracking, digital signal processing (DSP), and real-time computation environments that demand speed and efficiency.
- *Quantum and post-quantum cryptography*: LFSR-based counters play an important role in random number generation and cryptographic key management. Their relevance is expected to grow as cryptographic requirements evolve to address future security challenges.

- *AI and machine learning hardware accelerators:* In hardware designed to accelerate AI and ML algorithms, Galois LFSR counters offer fast and efficient state transitions that support rapid processing and low-latency operations.
- *Automotive and aerospace systems:* The integration of LFSR-based counters in safety-critical and fault-tolerant applications is projected to expand, driven by the increasing demand for reliable and power-efficient electronics in automated vehicles and space-grade systems.
- *Next-generation FPGA and ASIC implementations:* Future VLSI architectures are expected to increasingly incorporate LFSR-based counters for enhanced pipeline design, clock domain synchronization, and high-throughput data management, especially in advanced FPGA and ASIC platforms.

## REFERENCES

- [1] Rohith Bathula, Sunil Kumar Amgoth, “*64-bit High Speed Counter With Galois LFSR*”, 2024 International Conference on Circuit, Systems and Communication (ICCSC), 2024.
- [2] Hyungjoon Bae, Yujin Hyun, Suchang Kim, Sangsoo Park, “*High Speed Counter With Novel LFSR State Extension*”, IEEE Transactions on Computers, Vol. 72, No. 3, March 2023.
- [3] Jayasurya K, Ajith R, “*Area and Power Optimized RTL to GDS II Flow*”, 2024 5th International Conference on Circuits, Control, Communication and Computing, 2024.
- [4] Amit Shohal, Jasbir Kaur, “*Efficient RTL to GDS II Flow for Finite State Machine Integration: A Physical Design Approach*”, 2024 IEEE 5th India Council International Subsections Conference (INDISCON), 2024.