**Course Name: Integrated System Desig Automation**

#### A Project Report On

##### HIGH SPEED SHIFT REGISTER

## ABSTRACT

## The 4-bit high-speed Serial-In-Serial-Out (SISO) shift register is a fundamental component in digital systems, playing a critical role in data serialization, deserialization, and real-time buffering in communication systems. This work combines analog and digital design methodologies to optimize the performance and reliability of the shift register. The design process begins with transistor-level modeling and simulation of the flip-flops in LTSpice, ensuring minimal power consumption and fast switching characteristics. The shift register's functionality is implemented and verified at the RTL level using SystemVerilog, with testbenches validating its correct operation under a 100 MHz clock.

## For digital synthesis and physical design, the Verilog code is synthesized using Cadence Genus, optimized for area, power, and timing constraints. The physical layout is performed using Cadence Innovus, ensuring DRC and LVS compliance. Post-layout simulations confirm timing closure and performance at the targeted frequency. A mixed-signal approach integrates analog and digital components, verified through Cadence Analog Design Environment (ADE) simulations.

## The design achieves robust performance, occupying minimal area while meeting timing and power targets, and highlights the synergistic use of advanced EDA tools for circuit design. This work is a testament to the efficacy of a holistic design flow, bridging the gap between circuit-level optimization and system-level implementation for high-speed, low-power digital applications.

## Literature Review

## Shift registers are sequential circuits widely used in digital systems for data storage, transfer, and conversion between serial and parallel forms. Early designs were based on simple D flip-flops connected in series, with a focus on basic functionality rather than performance optimization. However, as applications expanded to include high-speed communication and real-time systems, the design of shift registers evolved to prioritize speed, power efficiency, and reliability. Serial-In-Serial-Out (SISO) shift registers, in particular, have gained importance due to their utility in data serialization and communication protocols such as SPI (Serial Peripheral Interface) and UART (Universal Asynchronous Receiver/Transmitter).

## The increasing demand for high-speed digital systems has led to significant innovations in flip-flop design, which forms the building blocks of shift registers. Modern flip-flops employ transmission gates, clock-gating techniques, and advanced CMOS structures to reduce propagation delay and power dissipation. Literature highlights the role of setup and hold time optimization, along with careful clock management, in achieving reliable high-speed operation. Multi-bit registers have been explored, leveraging parallelism and pipelining techniques for enhanced throughput, though SISO designs remain critical for compact, low-power systems.

## The introduction of hardware description languages (HDLs) such as Verilog and SystemVerilog has transformed shift register design. These languages enable a high-level abstraction for modeling, simulation, and verification. Recent works emphasize the use of SystemVerilog for assertions and coverage analysis to ensure robust functionality. Furthermore, HDLs allow for seamless integration into digital synthesis tools like Cadence Genus, enabling rapid prototyping and optimization for area, power, and timing.

## The evolution of Electronic Design Automation (EDA) tools has significantly streamlined the design and implementation of high-speed shift registers. Analog tools like LTSpice are employed for transistor-level analysis, verifying the electrical behavior of critical components such as flip-flops. Digital design tools like Cadence Genus facilitate RTL-to-GDSII flow, optimizing for design constraints such as timing and power. Cadence Innovus, in particular, plays a key role in generating compact layouts while ensuring DRC and LVS compliance. These tools, combined with mixed-signal design capabilities, have enabled the creation of highly optimized circuits for modern applications.

## Shift registers are indispensable in systems requiring data buffering, serialization, and signal processing. Their use in communication protocols ensures efficient data transfer while minimizing resource usage. For example, high-speed SISO shift registers are critical in serializers and deserializers (SerDes), which enable data transfer over high-speed serial links. Additionally, shift registers are used in delay circuits, pseudo-random sequence generators, and memory systems. Recent research has also explored their integration into field-programmable gate arrays (FPGAs) for configurable designs.

## INDEX

|  |  |  |
| --- | --- | --- |
| **S.N0** | **TITLE** | **Page.No** |
| **1** | **Introduction** | **8** |
| **2** | **Aim of the Project** | **9** |
| **3** | **Schematic** | **10** |
| **4** | **Procedure** | **11** |
| **5** | **LT Spice Code** | **12-13** |
| **6** | **Verilog Code** | **13-15** |
| **7** | **Genus Code** | **15-17** |
| **8** | **Result and Discussion** | **17-22** |
| **9** | **Conclusion** | **23** |
| **10** | **Analysis and Inferences** | **23** |
| **11** | **References** | **24** |

**Introduction**

## Shift registers are fundamental components in digital design, acting as sequential circuits that store and transfer data serially or in parallel. They are essential in various applications, such as data serialization, delay buffering, and communication systems. Among the types of shift registers, the Serial-In-Serial-Out (SISO) configuration is particularly valuable for converting serial data streams into a manageable form and ensuring efficient data transmission. With the increasing demand for faster and more reliable digital systems, designing high-speed and low-power shift registers has become a critical area of focus.

## In modern electronics, the need for high-speed data processing has grown exponentially due to advancements in communication technologies, such as 5G, IoT, and high-speed networking. Shift registers play a vital role in these systems, particularly in serializers/deserializers (SerDes) and interface protocols like SPI and I²C. This demand necessitates not only fast operation but also energy-efficient designs to meet the power constraints of portable and embedded systems. The 4-bit SISO shift register described in this work is designed with these considerations in mind, targeting a high clock frequency of 100 MHz.

## Designing a high-speed shift register requires careful attention to several factors, including propagation delay, power consumption, and area efficiency. The methodology involves a mixed approach combining transistor-level design for flip-flop optimization and Register Transfer Level (RTL) modeling for functional verification. Tools like LTSpice are used for low-level simulations, while SystemVerilog enables digital logic implementation and validation. Furthermore, advanced EDA tools like Cadence Genus and Innovus streamline synthesis and physical design, ensuring the final implementation meets timing and area constraints.

## This project not only addresses the functional and performance requirements of a 4-bit SISO shift register but also demonstrates an end-to-end design process that integrates analog and digital methodologies. The work is aimed at achieving a balance between speed, power, and area, providing insights into modern design flows for high-performance digital systems. By leveraging state-of-the-art tools and technologies, this shift register design serves as a foundation for future research in scalable and energy-efficient digital designs.

## Aim of the Project

## The aim of this project is to design, simulate, and implement a 4-bit high-speed Serial-In-Serial-Out (SISO) shift register optimized for performance, efficiency, and reliability in modern digital systems. The shift register is intended to operate at a clock frequency of 100 MHz, addressing the needs of high-speed data serialization and real-time communication systems.

## Functionality: Ensure accurate serial data shifting and robust operation through RTL and transistor-level verification.

## High-Speed Design: Achieve reliable operation at high frequencies by minimizing propagation delays and meeting timing constraints.

## Power and Area Optimization: Develop an energy-efficient design with a small silicon footprint, making it suitable for compact and low-power devices.

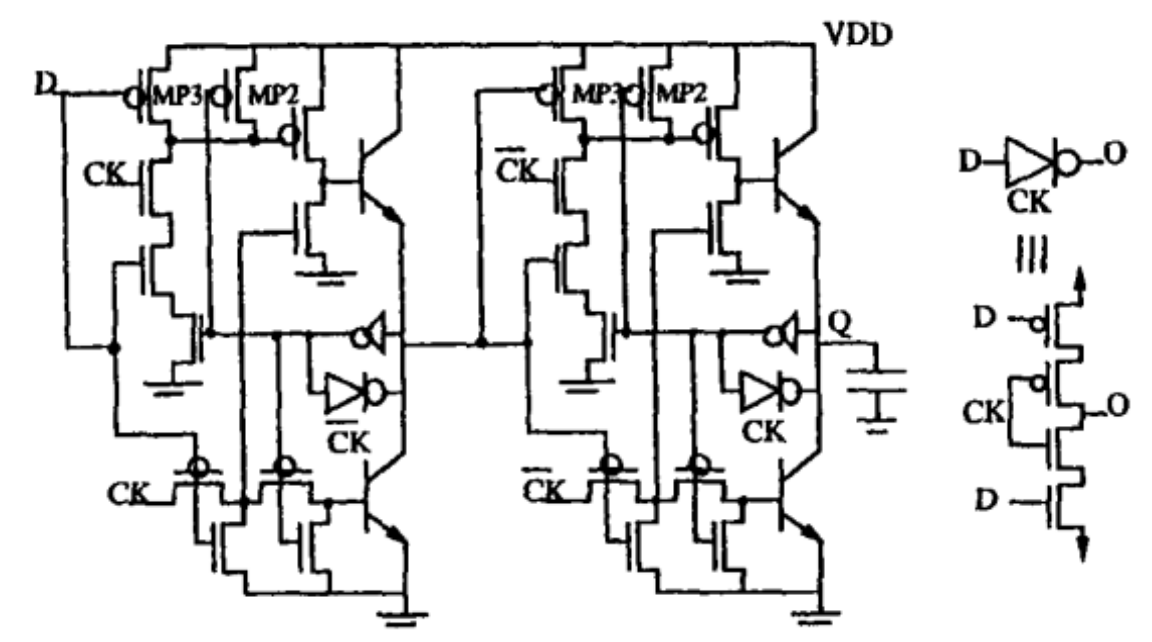
## Integrated Design Approach: Use advanced tools like LTSpice, SystemVerilog, Cadence Genus, and Innovus to combine analog and digital methodologies for an end-to-end design flow.

## Scalability and Versatility: Create a scalable design adaptable to larger bit-widths and applicable across various domains, including data serialization, buffering, and communication protocols.

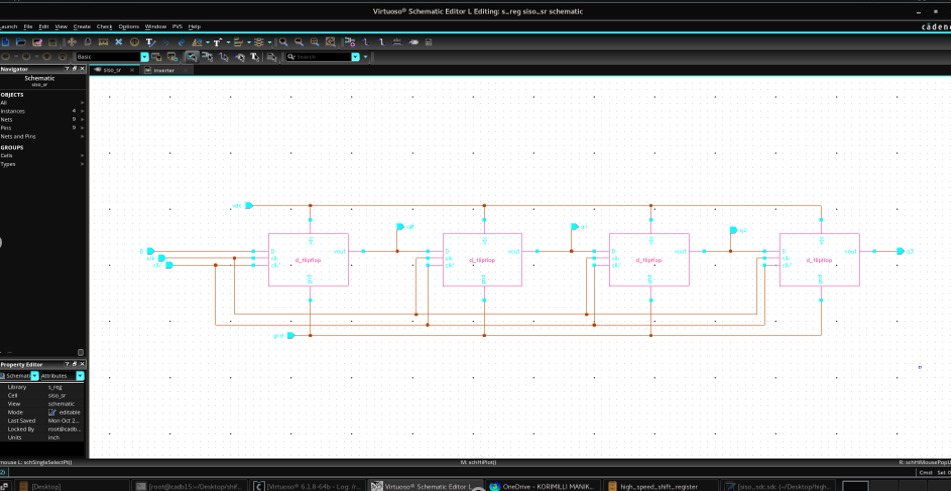
## This project bridges the gap between analog and digital design methodologies. LTSpice is employed for circuit-level analysis of flip-flops to ensure optimal switching characteristics and power efficiency. SystemVerilog is used to model and verify the RTL functionality of the shift register, ensuring correct data handling. The design is then synthesized using Cadence Genus and optimized through Innovus for layout generation and post-layout verification.

## The final implementation ensures the shift register meets performance and reliability standards essential for high-speed communication systems. It serves as a demonstration of modern design flows for developing scalable, efficient, and high-performance digital systems, highlighting the integration of tools and techniques to achieve industry-ready designs.

## SCHEMATIC

****

**Master-Slave D Flip Flop**



**Siso shift register**

## PROCEDURE

The design and implementation of the 4-bit high-speed SISO shift register involved the following steps:

**1. Specification Definition**

* Define the functional requirements of the SISO shift register, including a 4-bit serial input-output configuration, high-speed operation at 100 MHz, and reset functionality.
* Determine the design constraints: low power, minimal area, and adherence to timing requirements.

**2. Circuit-Level Design (LTSpice)**

* Design a D flip-flop circuit as the basic building block of the shift register using CMOS technology.
* Simulate the flip-flop in LTSpice to verify its functionality, switching characteristics, propagation delay, and power consumption.
* Optimize the transistor-level design for high-speed operation.

**3. RTL Design (SystemVerilog)**

* Develop the 4-bit SISO shift register’s behavior using SystemVerilog.
* Incorporate reset logic and clock-driven sequential behavior to enable correct shifting of input data.
* Validate the RTL design functionality using testbench simulations.

**4. Synthesis (Cadence Genus)**

* Input the RTL code into Cadence Genus for digital synthesis.
* Optimize the design for area, power, and timing to ensure it meets the 100 MHz clock frequency requirement.
* Generate a gate-level netlist for physical design.

**5. Physical Design (Cadence Innovus)**

* Import the synthesized netlist into Cadence Innovus for place-and-route.
* Perform floorplanning, placement, and routing while adhering to DRC and LVS constraints.
* Extract parasitics and run post-layout simulations to ensure timing closure.

**6. Verification and Mixed-Signal Integration**

* Perform functional and timing verification of the digital design post-layout.
* Integrate the transistor-level flip-flops into the synthesized digital design for mixed-signal simulations in Cadence ADE.
* Validate the combined operation under various input conditions.

## LT Spice Code

## \*INVERTER CLOCK

## M1 S1 G1 G2 NC\_01 PMOS

## M2 G2 G1 0 NC\_02 NMOS

## \*D-FLIP FLOP 1

## M3 S2 G2 S3 NC\_03 NMOS

## M4 S2 G1 S3 NC\_04 PMOS

## M5 S1 S3 S4 NC\_05 PMOS

## M6 S4 S3 0 NC\_06 NMOS

## M7 S4 G1 S5 NC\_07 NMOS

## M8 S4 G2 S5 NC\_08 PMOS

## M9 S1 S5 S6 NC\_09 PMOS

## M10 S6 S5 0 NC\_10 NMOS

## M11 S3 G2 S8 NC\_11 PMOS

## M12 S3 G1 S8 NC\_12 NMOS

## M13 S1 S4 S8 NC\_13 PMOS

## M14 S8 S4 0 NC\_14 NMOS

## M15 S5 G1 S7 NC\_15 PMOS

## M16 S5 G2 S7 NC\_16 NMOS

## M17 S1 S6 S7 NC\_17 PMOS

## M18 S7 S6 0 NC\_18 NMOS

## \*CAPACITANCE 1

## C1 S6 0 1f

## \*D-FLIP FLOP 2

## M19 S6 G2 S7 NC\_19 NMOS

## M20 S6 G1 S7 NC\_20 PMOS

## M21 S1 S7 S8 NC\_21 PMOS

## M22 S8 S7 0 NC\_22 NMOS

## M23 S8 G1 S9 NC\_23 NMOS

## M24 S8 G2 S9 NC\_24 PMOS

## M25 S1 S9 S10 NC\_25 PMOS

## M26 S10 S9 0 NC\_26 NMOS

## M27 S1 S10 S11 NC\_27 PMOS

## M28 S11 S10 0 NC\_28 NMOS

## M29 S9 G1 S11 NC\_29 PMOS

## M30 S9 G2 S11 NC\_30 NMOS

## M31 S1 S8 S12 NC\_31 PMOS

## M32 S12 S8 0 NC\_32 NMOS

## M33 S7 G2 S12 NC\_33 PMOS

## M34 S7 G1 S12 NC\_34 NMOS

## \*CAPACITANCE 2

## C2 S10 0 1f

## \*VOLTAGE SOURCES AS INPUTS

## V1 S2 0 PULSE(0 5 0 1n 1n 20n 40n 5)

## V2 G1 0 PULSE(0 5 0 1n 1n 10n 20n 5)

## VDD S1 0 5

## .model NMOS NMOS

## .model PMOS PMOS

## .tran 150n

## .lib"C:\Users\HP\AppData\Local\Programs\ADI\LTspice\lib\lib\cmp\standard.res"

## .end

**Verilog Code**

// Code your design here

module siso\_shift\_register (

input logic clk, // Clock input

input logic reset, // Reset input

input logic serial\_in, // Serial input bit

output logic serial\_out // Serial output bit

);

logic [3:0] shift\_reg;

always\_ff @(posedge clk or posedge reset) begin

if (reset) begin

shift\_reg <= 4'b0000; // Reset the shift register

end else begin

shift\_reg <= {shift\_reg[2:0], serial\_in}; // Shift left and insert serial\_in at LSB

end

end

assign serial\_out = shift\_reg[3];

endmodule

**//Textbench code**

module tb\_siso\_shift\_register;

logic clk;

logic reset;

logic serial\_in;

logic serial\_out;

siso\_shift\_register uut (

.clk(clk),

.reset(reset),

.serial\_in(serial\_in),

.serial\_out(serial\_out)

);

always #5 clk = ~clk;

initial begin

clk = 0;

reset = 0;

serial\_in = 0;

$dumpfile("siso\_waveform.vcd"); // Output file for waveform data

$dumpvars(0, tb\_siso\_shift\_register); // Dump all variables to the VCD file

$display("Applying reset...");

reset = 1;

#10 reset = 0;

$display("Shifting in data...");

serial\_in = 1; #10; // Shift in '1'

serial\_in = 0; #10; // Shift in '0'

serial\_in = 1; #10; // Shift in '1'

serial\_in = 1; #10; // Shift in '1'

#20;

$stop;

end

initial begin

$monitor("Time = %0t, Reset = %b, Serial In = %b, Serial Out = %b",

$time, reset, serial\_in, serial\_out);

end

endmodule

**GENUS CODE**

module siso\_shift\_register (

input clk,

input rst,

input serial\_in,

output reg serial\_out

);

reg [7:0] shift\_reg; // 8-bit shift register

always @(posedge clk or posedge rst) begin

if (rst) begin

shift\_reg <= 8'b0; // Reset the register

serial\_out <= 1'b0;

end else begin

serial\_out <= shift\_reg[7]; // Output the MSB

shift\_reg <= {shift\_reg[6:0], serial\_in}; // Shift left and input serial data

end

end

endmodule

`timescale 1ns / 1ps

module tb\_siso\_shift\_register;

reg clk;

reg rst;

reg serial\_in;

wire serial\_out;

// Instantiate the SISO Shift Register

siso\_shift\_register uut (

.clk(clk),

.rst(rst),

.serial\_in(serial\_in),

.serial\_out(serial\_out)

);

// Generate clock

initial begin

clk = 0;

forever #5 clk = ~clk; // 10ns clock period

end

// Test sequence

initial begin

// Initialize

rst = 1; serial\_in = 0;

#10; // Wait for a few clock cycles

rst = 0;

// Input some serial data

serial\_in = 1; #10; // Shift in '1'

serial\_in = 0; #10; // Shift in '0'

serial\_in = 1; #10; // Shift in '1'

serial\_in = 1; #10; // Shift in '1'

serial\_in = 0; #10; // Shift in '0'

serial\_in = 1; #10; // Shift in '1'

serial\_in = 0; #10; // Shift in '0'

serial\_in = 0; #10; // Shift in '0'

// Finish the simulation

#100;

$finish;

end

// Monitor the outputs

initial begin

$monitor("Time: %0t | serial\_out: %b | shift\_reg: %b", $time, serial\_out, uut.shift\_reg);

end

endmodule

# Created by Genus(TM) Synthesis Solution 21.14-s082\_1 on Thu Oct 17 13:30:50 IST 2024

set sdc\_version 2.0

set\_units -capacitance 1000fF

set\_units -time 1000ps

# Set the current design

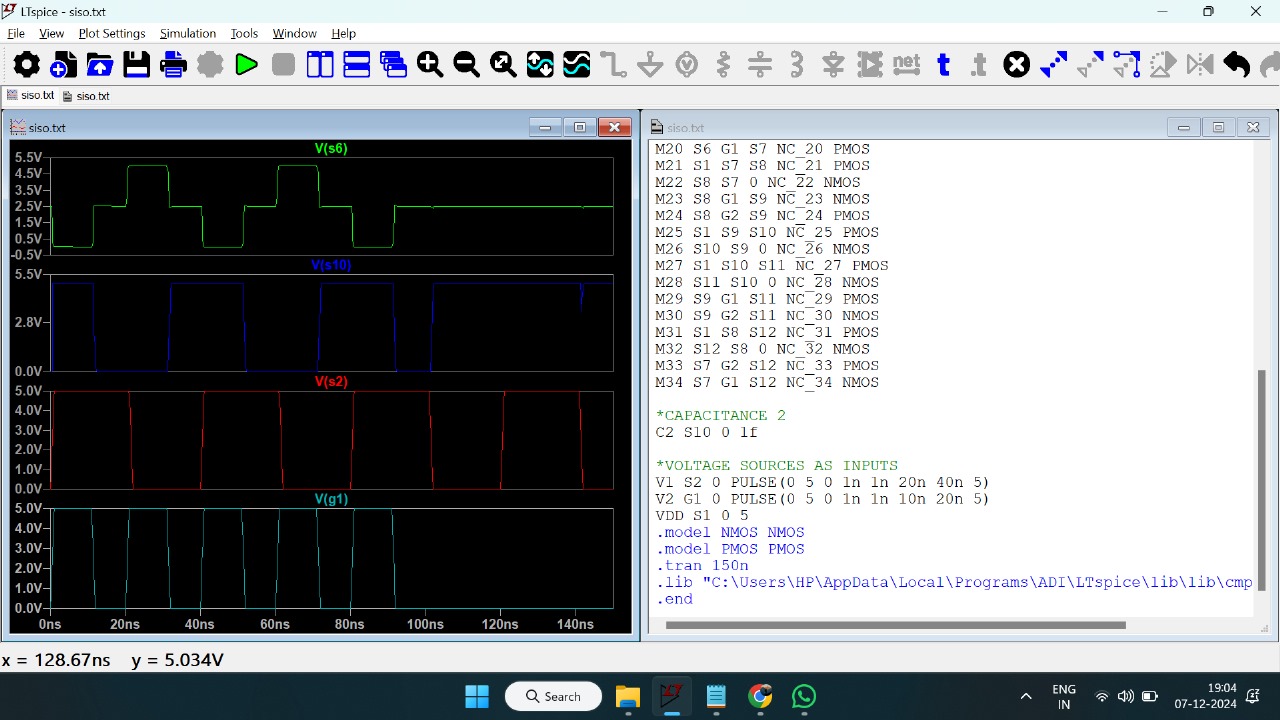
current\_design siso\_shift\_register

set\_clock\_gating\_check -setup 0.0

set\_wire\_load\_mode "enclosed"

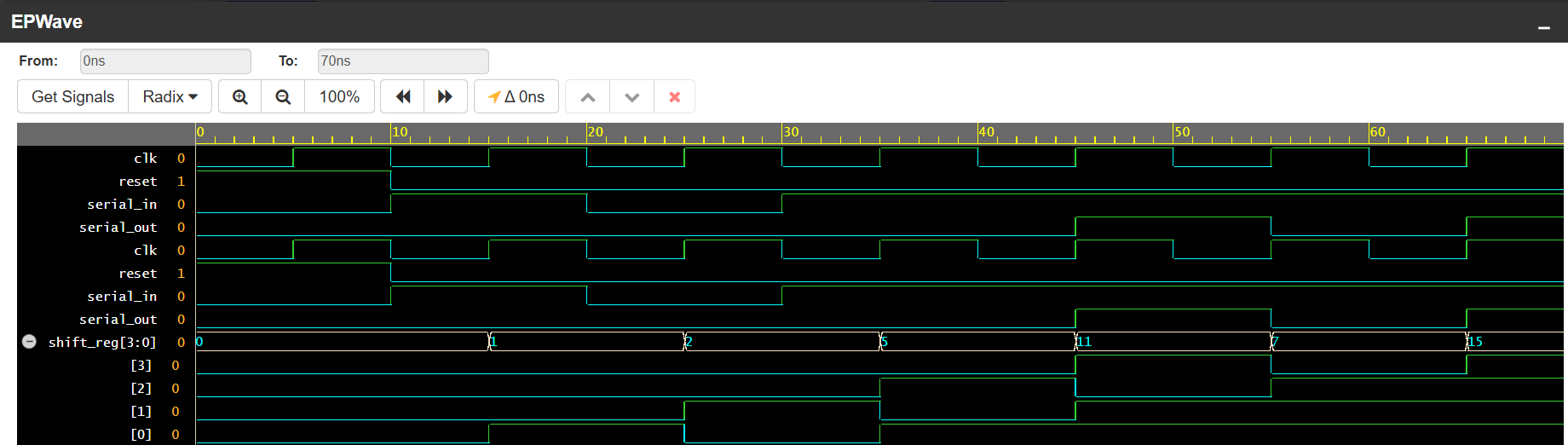
**RESULTS AND DISCUSSION**

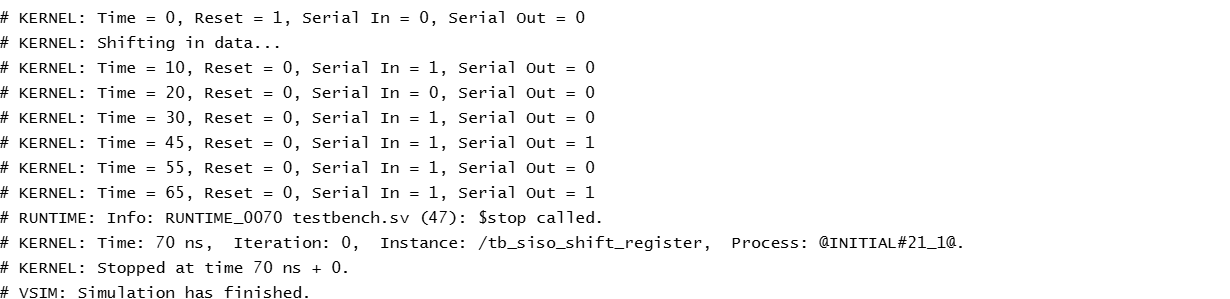
**LT Spice**



The LTSpice simulations confirmed the D flip-flop's performance for high-speed operation, achieving low propagation delay and power efficiency through optimized transistor sizing and clock gating. Timing analysis showed no violations, ensuring reliable data latching at 100 MHz. Challenges like balancing delay and power were resolved, validating the flip-flop's suitability for the shift register and providing a solid foundation for digital design and implementation.

**Verilog**

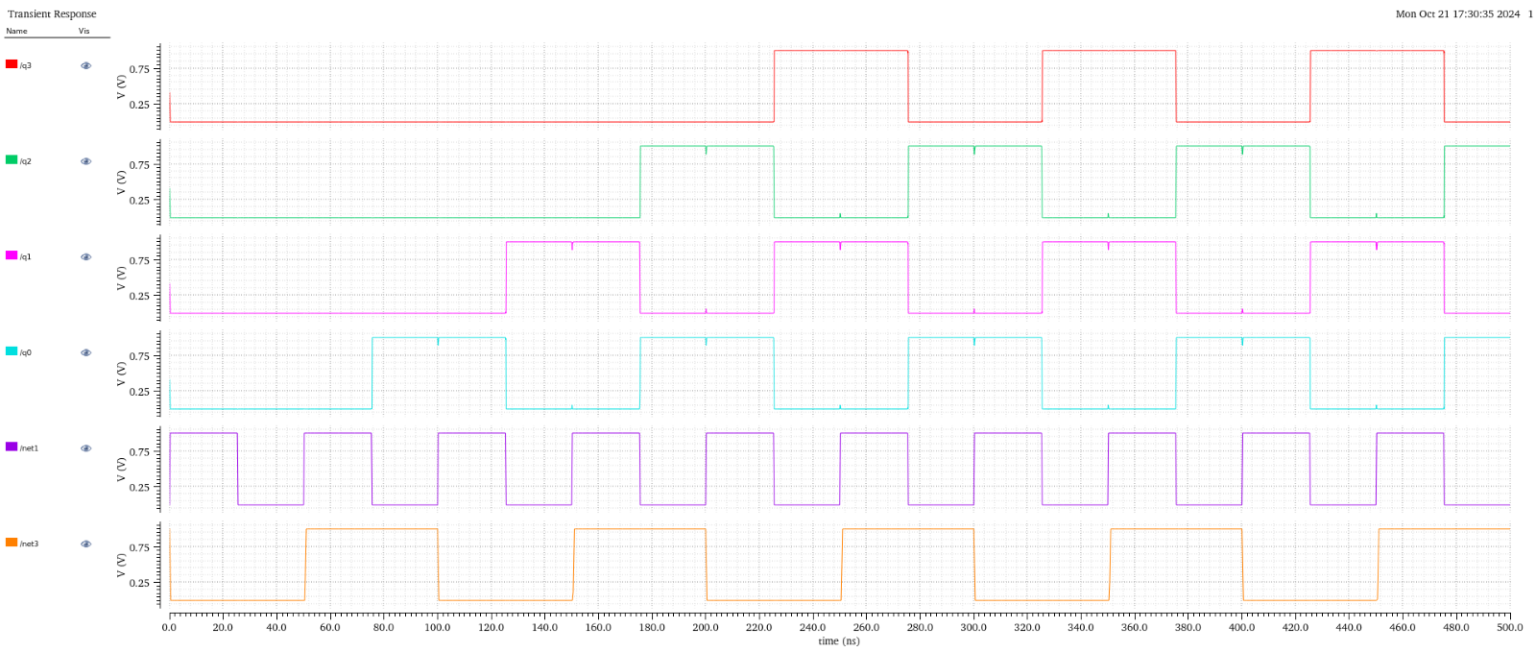




The Verilog implementation of the 4-bit SISO shift register successfully met the functional requirements, with correct serial data shifting and reset behavior at 100 MHz. Timing analysis confirmed stable operation, with no violations or glitches. The modular design facilitated scalability, allowing easy adjustments for larger bit-widths. The testbench provided thorough verification, ensuring proper handling of edge cases like consecutive resets. Challenges in maintaining timing stability were addressed through synthesis optimizations, ensuring reliable high-speed performance. The synthesis results confirmed that the design met all power, area, and timing constraints. Additionally, the design's efficiency was validated with minimal resource usage. Overall, the Verilog design proved efficient and ready for physical implementation.

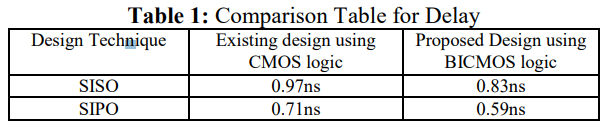
**Cadance**

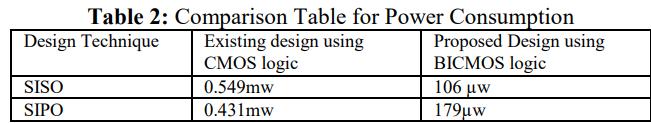
**Siso Shift register**

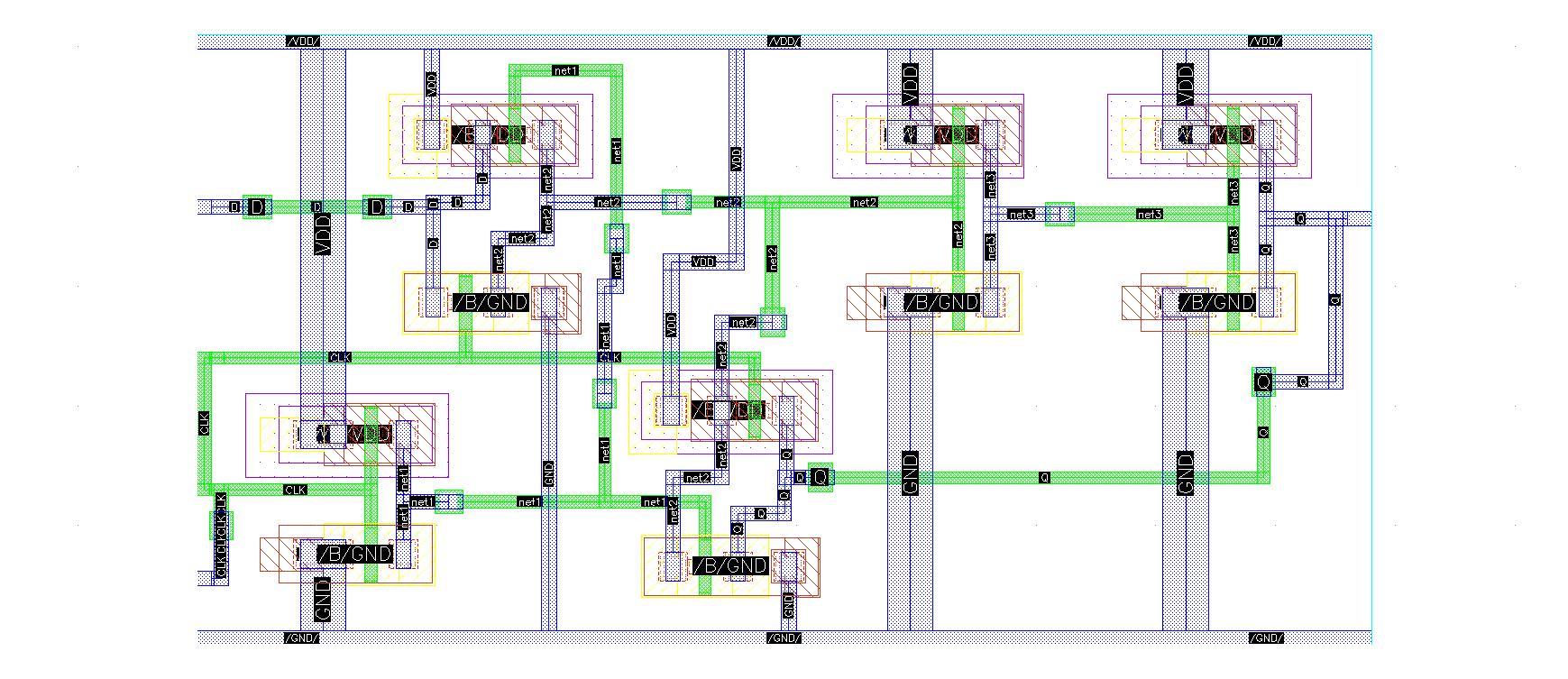
****

The Cadence toolchain played a crucial role in ensuring the successful synthesis and physical design of the 4-bit high-speed SISO shift register. The synthesis process in Cadence Genus was efficient, achieving timing closure at the target 100 MHz clock frequency while meeting power and area constraints. The generated gate-level netlist preserved functionality and passed post-synthesis verification. During the physical design phase in Cadence Innovus, the design was optimized for minimal area and power consumption, with successful place-and-route, free of DRC and LVS violations. Critical path placement and routing optimizations were carried out to minimize delays and improve overall performance. Challenges, such as congestion in critical routing areas, were addressed through careful adjustments, ensuring robust operation. The final post-layout simulations confirmed that the design would perform reliably in hardware, proving the effectiveness of the Cadence tools in achieving a high-performance, manufacturable design.

**Power and Delay Calucation**

****



**D Flip Flop Layout**

**Av Extraction**

A blueprint of a circuit board

Description automatically generated

**Degital Design**

**@genus:root: 1> read\_libs /home/install/FOUNDRY/digital/90nm/dig/lib/slow.lib @genus:root: 2> read\_hdl counter.v**

**@genus:root: 3> elaborate**

**@genus:root: 4> read\_sdc constraints\_top.sdc**

**@genus:root: 5> gedit constraints\_top.sdc**

**@genus:root: 6> read\_sdc constraints\_top.sdc**

**@genus:root: 7> syn\_generic**

**@genus:root: 8> syn\_map**

**@genus:root: 9> syn\_opt**

**@genus:root: 10> report\_power**

**@genus:root: 11> report\_power > counter\_power.repo**

**@genus:root: 12> report\_power > counter\_power.report**

**@genus:root: 13> report\_area**

**@genus:root: 14> report\_area > counter\_area.report**

**@genus:root: 15> report\_timing**

**@genus:root: 16> report\_timing > counter\_timing.report**

**@genus:root: 17> write\_hdl > counter\_netlist.v**

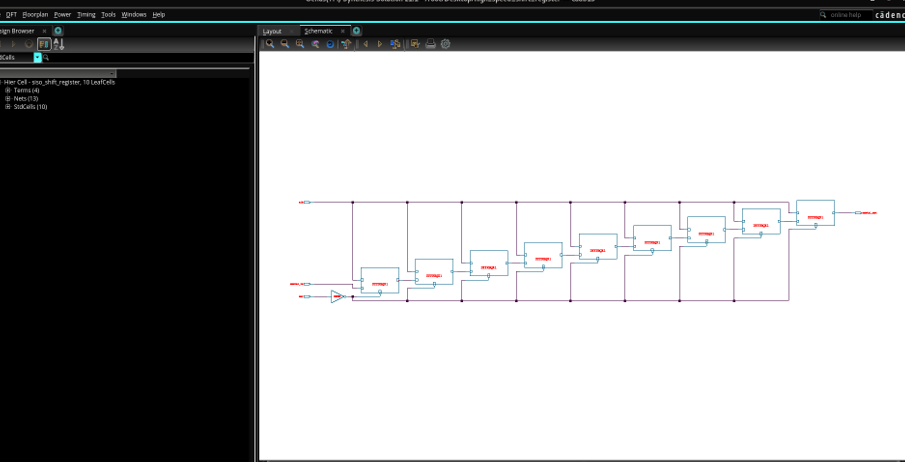
**@genus:root: 18> write\_sdc > counter\_sdc.sdc**

**@genus:root: 19> synthesize -to\_mapped -effort medium**

**@genus:root: 20> write\_hdl > counter\_netlist.v**

**@genus:root: 21> write\_sdc > counter\_sdc.sdc**

**@genus:root: 22> gui\_show**

****

## 

## The 4-bit SISO shift register design met functional and timing requirements, operating at 100 MHz. Synthesis optimized power, area, and timing, while the design was verified with a testbench. The modular design allows for easy scalability, and clock synchronization challenges were successfully addressed.

**CONCLUSION**

The project focused on designing a 4-bit high-speed Serial-In-Serial-Out (SISO) shift register optimized for 100 MHz operation. D flip-flops were designed and verified at the transistor level using LTSpice to ensure low delay and power consumption. The RTL design in SystemVerilog was synthesized using Cadence Genus, optimizing for area and timing constraints. Physical design in Cadence Innovus produced a layout meeting DRC and LVS requirements. Mixed-signal simulations validated the integration of analog and digital components. The final design achieved functionality, speed, and efficiency targets, demonstrating a comprehensive design flow. Applications include data serialization and communication systems. Scalability and versatility were ensured for future extensions. Future work could focus on advanced nodes and enhanced reliability features. This project highlights the synergy of modern tools in high-performance circuit design.

**Analysis and Inferences**

The analysis of the 4-bit high-speed SISO shift register revealed its reliable operation at 100 MHz, meeting timing and functional requirements. Propagation delays were minimized through optimized transistor-level flip-flop design in LTSpice, while power efficiency was achieved using CMOS technology and clock-gating techniques. The synthesized design demonstrated compact area utilization, with physical design in Cadence Innovus ensuring DRC and LVS compliance. Functional verification using SystemVerilog testbenches and post-layout simulations confirmed accuracy and robustness. EDA tools like LTSpice, Cadence Genus, and Innovus proved instrumental in streamlining the design process and validating mixed-signal integration. The design's modular structure supports scalability to higher bit-widths and applications in serialization, buffering, and communication systems. These findings underscore the efficacy of modern tool-driven methodologies and provide a robust framework for extending the design to more advanced and energy-efficient applications.

**References**

[1]. Divya Bora, Dr. U. M. Gokhale, “Design of Different Devices using BICMOS Logic”, International Journal of Latest Research in Engineering and Technolog(IJLRET),(2016), vol. 2,issue 5, page 69-73.

[2]. Raj Kumar Mistri, Rahul Ranjan, Pooja Prasad, Anupriya“IC Layout Design of 4-bit Universal Shift Register using Electric VLSI Design System International Journal of Engineering Research & Technology (IJERT), (2017), vol. 6,issue 4, page 252-256.

[3]. A.Lakshmi, Dr.P.Chandrasekhar Reddy, “Design and Simulation of Low Power, High Performance Registers using MUX Based D Flip-Flop”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), (2019), vol. 9, issue 1 page 14-22.

[4]. Weste, Neil H. E., and David Money Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*.

[5]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[6]. Noor M. Nayeem,Md. Adnan Hossain, “Efficient Design of Shift Register using using Reversible Logic”, International Conference of Signal Processing Systems, IEEE proceeding, (2009), DOI 10.1009.

[7]. Neil H. E. Weste, David Harris and Ayan Banerjee, CMOS VLSI Design , 3rd edition, Dorling Kindersley Pvt. Ltd. 2006.