STM JOURNALS

Journal of
VLSI Design Tools & Technology

ISSN: 2249-474X (Online)
ISSN: 2321-6492 (Print)
Volume 14, Issue 3, 2024
September–December
DOI (Journal): 10.37591/JoVDTT

https://journals.stmjournals.com/jovdtt

Research JoVDTT

RTL-to-GDS Implementation of a High-Speed On-Chip 32:1 Serializer Using Open-Source Tools

Meet B. Sangani[1,*], Pallavi G. Darji[2]

Abstract

Exploring the RTL-to-GDS implementation of a high-speed on-chip 32:1 serializer, this study investigates the integration of open-source tools within VLSI design, emphasizing sustainable practices in the semiconductor industry while operating at the nanometer scale. Addressing methodologies for optimizing power consumption and chip area utilization, particularly focusing on efficient use of non-renewable resources. The study is set against the backdrop of advanced 7nm FinFET technology, critical for enabling efficient data transmission in modern System-on-Chip (SoC) architectures. The paper intricately delineates the journey from RTL description to physical layout, harnessing the capabilities of Yosys and Open ROAD, prominent open-source tools, in conjunction with the ASAP7 Process Design Kit (PDK). The design achieves optimization in key performance metrics such as power dissipation and chip area utilization, while achieving a remarkable data transmission rate of 17 Gbps. Through meticulous experimentation and analysis, this implementation not only showcases the effectiveness and adaptability of open-source tools in contemporary semiconductor design but also underscores their potential to democratize access and drive innovation in IC development workflows.

Keywords: VLSI design flow, Serializer, Open-source tools, RTL-to-GDS flow, 7 nm technology

INTRODUCTION

In today's interconnected world, digital communication is permeating across every aspect of life, driving an ever-increasing demand for high-speed data transmission. Efficient data transfer is at the heart of modern electronic systems, whether streaming videos, downloading files, or conducting real-time transactions. The core of this quest for seamless connectivity is the field of Very Large-Scale Integration (VLSI), where complex functionalities are condensed into tiny silicon chips.

Central to the efficiency of these systems are serializers, which are humble yet indispensable components serving as a conduit for data exchange within electronic devices. The process of converting a stream of parallel data into a serial format can be likened to compressing a crowd of people into a single-file line, a function performed by serializers, enabling efficient communication over bandwidth-constrained channels. However, designing high-speed serializers presents significant challenges, such as minimizing power consumption, reducing the area footprint, and ensuring timing closure while maintaining robust performance and signal integrity.

The journey from an idea to a tangible chip involves a series of meticulously orchestrated steps akin to symphonies of creativity and precision. This journey, often referred to as the Register Transfer Level (RTL) to Graphic Design System (GDS)

***Author for Correspondence**
Meet B. Sangani
E-mail: meet.sangani@outlook.com

[1]Student, Department of Electronics and Communication Engineering, Dharmsinh Desai University, Nadiad, Gujarat, India
[2]Associate Professor, Department of Electronics and Communication Engineering, Dharmsinh Desai University, Nadiad, Gujarat, India

Received Date: September 18, 2024
Accepted Date: September 23, 2024
Published Date: September 30, 2024

Citation: Meet B. Sangani, Pallavi G. Darji. RTL-to-GDS Implementation of a High-Speed On-Chip 32:1 Serializer Using Open-Source Tools. Journal of VLSI Design Tools & Technology. 2024; 14(3): 1–10p.

flow, encompasses the translation of a design expressed in a hardware description language (HDL) into a physical layout that can be fabricated onto a semiconductor substrate [1].

At the heart of this intricate process lies a plethora of tools and methodologies, each contributing a unique essence to the alchemy of chip design. Among these tools, OpenROAD stands out as a beacon of innovation, a testament to the collaborative spirit of the open-source community. With its arsenal of advanced algorithms for placement, routing, optimization, and timing closure, OpenROAD empowers designers to navigate the labyrinthine complexities of the RTL-to-GDS transformation with unparalleled ease and efficiency [1].

In the crucible of the VLSI design, the significance of an efficient serializer design cannot be overstated. It serves as the linchpin upon which the performance, power consumption, and area utilization of the electronic system hinges. This research endeavors to explore the symbiotic relationship between serializer design and the OpenROAD tool, unravel the intricacies of efficient design methodologies, and pave the way for a new era of sustainable semiconductor innovation. Specifically, the aim was to investigate the challenges of designing a high-speed on-chip 32:1 serializer and leverage the capabilities of OpenROAD to optimize its performance, power, and area while ensuring timing closure.

In this paper, a journey is embarked that transcends the boundaries of conventional wisdom and ventures into uncharted realms of possibility. Through meticulous experimentation and analysis, light is sought to be shed on the transformative potential of open-source tools in the realm of digital integrated circuit design.

LITERATURE REVIEW

In high-speed communication systems, serializers play a crucial role in enabling efficient data transfer by converting parallel data streams into serial formats and vice versa. The design and implementation of these critical components have traditionally been dominated by proprietary electronic design automation (EDA) tools and methodologies offered by industry giants, such as Cadence and Synopsys.

The papers provided exemplify the reliance on licensed Cadence and Synopsys tools for SerDes (Serializer/Deserializer) design and implementation [2]. While these tools offer robust and comprehensive solutions, they come with significant financial barriers that limit their accessibility to students, researchers, and smaller organizations with limited budgets.

To address this challenge, the open-source community has been actively developing a range of EDA tools and methodologies to provide cost-effective alternatives while fostering collaborative development and knowledge sharing. One notable example is the OpenROAD project, which provides an open-source framework for digital circuit design and optimization, including placement, routing, and timing closure algorithms [3]. Open ROAD and similar open-source tools have the potential to be extended and adapted for serializer implementation, thus offering a more accessible path for students, researchers, and independent designers. Despite promising efforts in the open-source EDA domain, several challenges remain. Most open-source tools are still in the early stages of development and may lack the robustness and comprehensive feature sets of their proprietary counterparts. Furthermore, the integration and interoperability of different open-source tools across the design flow can be a significant challenge, requiring concerted efforts from the community to establish a seamless workflow.

APPLICATIONS OF SERIALIZERS

Serializers, one of the major blocks of Serializer-Reserializes (SerDes), are fundamental components of high-speed data communication within and between integrated circuits. They addressed the limitations of parallel data buses by converting parallel data streams into a single serial stream (serialization) and vice

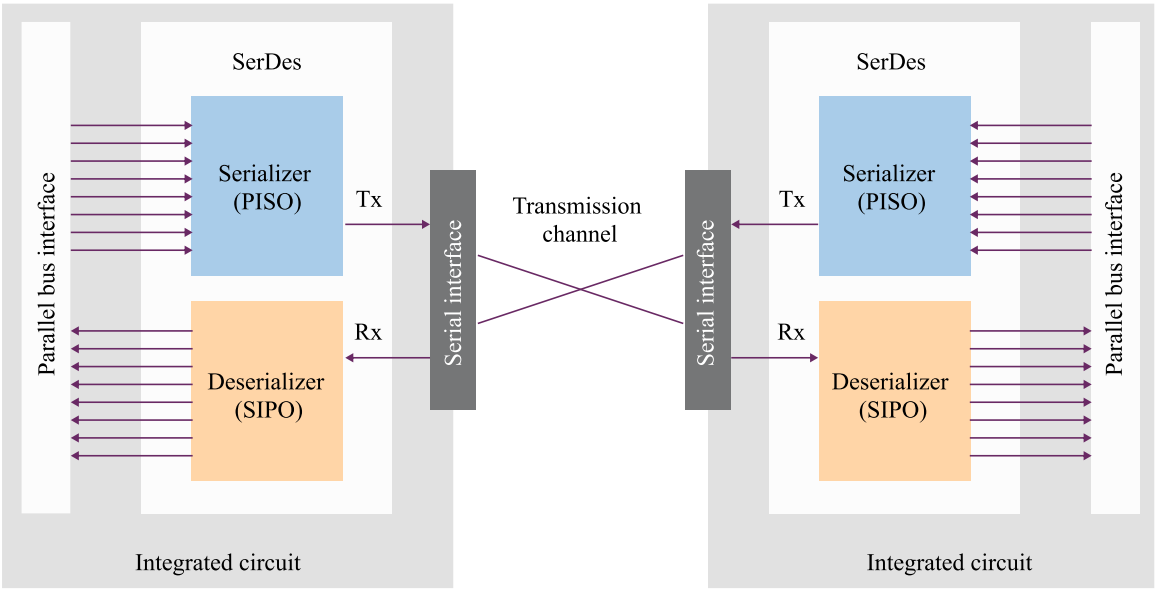

Figure 1. A pair of SerDes chips exchanging serial data over a serial transmission channel [4].

versa (deserialization) [5]. A pair of SerDes chips exchanges serial data over a serial transmission channel shown in Figure 1. This approach offers several advantages that are particularly valuable in VLSI design.

Benefits of Serialization

- *Reduced pin count:* Serial communication significantly reduces the number of I/O pins required compared to a parallel bus with multiple data lines. This is crucial for chip size optimization and cost reduction in VLSI design. Fewer pins translate into smaller chip footprints and potentially lower fabrication costs. Figure 2 illustrates the process of analog signal digitization and serialization before transmission to a System-on-Chip (SoC). This digitization and serialization process significantly reduces the number of pins required [4].
- *Simplified routing:* Using a single or differential pair of pins for data transmission simplifies the routing of signals on the chip. Parallel buses have become increasingly challenging to route at high frequencies because of issues such as signal skew (misalignment) and crosstalk (interference between adjacent lines). Serializers alleviate these complexities by concentrating data on fewer lines, leading to cleaner and more reliable signal routing.
- *Higher data rates:* Serial communication enables higher data transmission rates compared to parallel buses. By concentrating the data onto a single lane, serializers can operate at higher frequencies owing to improved signal integrity. This allows for faster data exchange between components within a VLSI system.

Common Applications for SerDes

SerDes play a vital role in various high-speed communication scenarios within integrated circuits.

- *High-speed chip-to-chip communication:* SerDes are essential for inter-die communication within multicore processors, network-on-chip (NoC) architectures, and other high-performance VLSI systems. They enable data exchange between the processing cores, memory controllers, and other on-chip components at very high speeds, facilitating efficient operation within the chip.

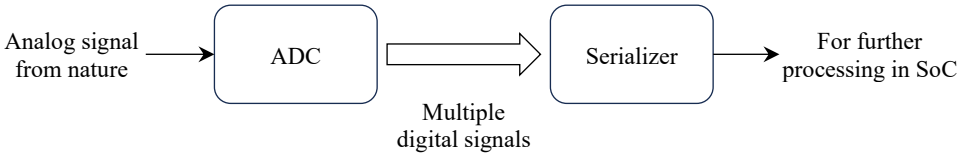

Figure 2. Use of serializer in SoC.

- *Memory interfaces:* Modern memory interfaces like DDR (Double Data Rate (DDR)) memory and Graphics DDR (GDDR) rely heavily on SerDes for high-bandwidth communication between processors and memory. SerDes enables fast data transfer between these components, ensuring that processors have quick access to the required data [6].
- *Off-chip communication:* Industry standards such as PCI Express (PCIe), USB, and Ethernet use SerDes for high-speed data transfer between chips on printed circuit boards (PCB) or across longer distances using cables or optical links. SerDes enable efficient communication between chips on a board or across systems, thereby facilitating data exchange between various components.

METHODOLOGY

The methodology adopted in this study presents a systematic and comprehensive approach to the design and implementation of a serializer that ensures compliance with the designated design specifications and requirements. This section provides an insightful overview of the essential methodologies and techniques employed in RTL-to-GDS flow utilizing the OpenROAD framework, as illustrated in Figure 3.

Experimental Setup

For the experimental setup, OpenROAD flow scripts (ORFS), a framework within the OpenROAD project, were utilized [7] ORFS offers a fully autonomous RTL-GDSII flow for rapid architecture, design space exploration, and detailed physical design implementation. In addition, ORFS allows for manual intervention through Tcl commands and Python APIs, enabling finer user control of individual flow stages.

Design Specifications

Figure 4 illustrates the functional components and their interconnections in the serializer circuit. Initially, a 32-bit parallel data stream is fed into a latch to temporarily hold the data. Upon the rising edge of the clock signal, the data are transferred to a 32:1 multiplexer, where the selection of the input lines is controlled by a 5-bit counter. With each rising edge of the clock signal, the counter increments and sequentially selects each input line of the multiplexer. The selected data are then routed through a tri-state buffer, allowing it to reach the final output when the enable signal is active. Once the 5-bit counter reaches its maximum value (11111), indicating the completion of serialization, the load signal resets, and the latch re-latches the input data.

PDK Selection and SDC File

The choice of the Process Design Kit (PDK) plays a pivotal role in the successful realization of serializer design. For this study, the ASAP7 PDK, renowned for its compatibility with advanced nanometer-scale designs, emerged as the optimal selection among available alternatives, such as SkyWater130 and Nangate45. The decision to opt for ASAP7 was based on its extensive feature set tailored for high-performance applications [8].

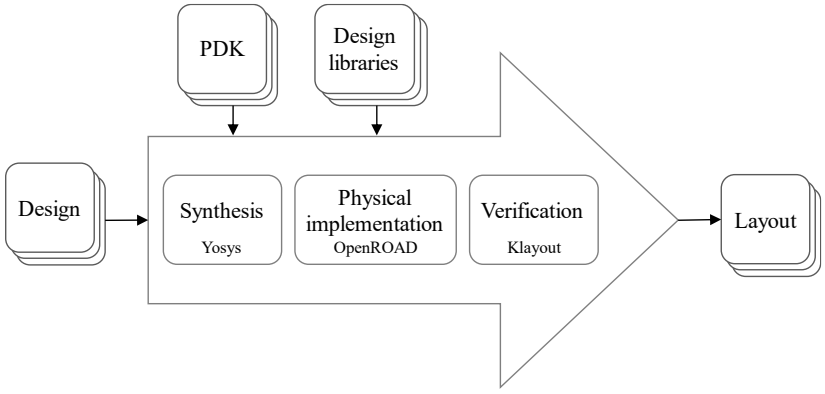

Figure 3. Overview of the RTL-to-GDS flow using OpenROAD.

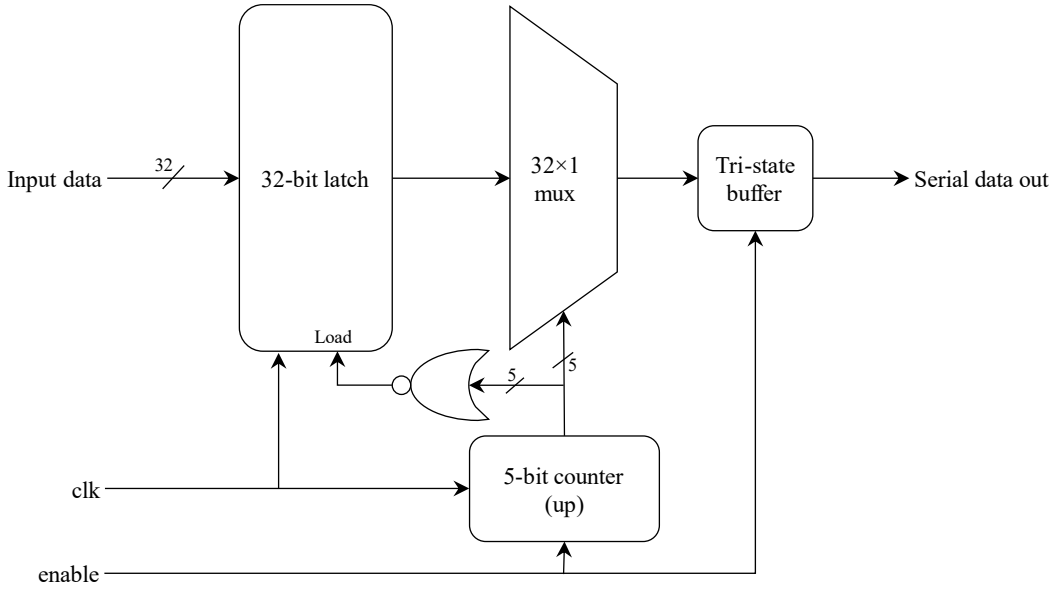

Figure 4. Logic diagram of 32:1 serializer.

The Synopsys Design Constraint (SDC) file is of significant importance throughout the design flow of integrated circuits. It serves as a comprehensive guide, specifying crucial timing constraints and design parameters essential for an accurate Static Timing Analysis (STA). Within the SDC file, various parameters are specified, including clock periods, input/output delays, timing exceptions, and synchronization requirements [9]. These specifications are indispensable for ensuring proper functionality and achieving timing closure of the serializer circuit. By defining detailed timing specifications, the SDC file helps in enforcing setup and hold times for flip-flops, thereby preventing timing violations and ensuring the reliable operation of the design. In the context of the current design, the SDC file specifies a 10 GHz clock with a 50% duty cycle. This parameter is essential for further STA analysis, as it accurately defines the timing characteristics of the design and facilitates the thorough verification and validation of timing constraints.

RTL-to-GDS Flow

The RTL-to-GDS flow encompasses a series of intricate steps aimed at transforming the RTL description of the serializer design into a fully realized physical layout. This process involves synthesis, floor planning, placement, clock tree synthesis (CTS), routing, and finishing stages, each contributing to the overall success of the design implementation.

The synthesis serves as the initial stage of the RTL-to-GDS flow, wherein the Verilog HDL description of the serializer is transformed into a gate-level netlist. This netlist, comprising interconnected standard cells from the selected PDK, forms the foundation for the subsequent stages of the design flow [10].

In the floor planning stage (Figure 5(a)), the layout after floor planning is depicted, showing the initial placement of the VCC and GND lines. This foundational step in the RTL-to-GDS flow involves allocating physical space on the chip to different functional blocks of the serializer. The placement of power lines is critical to ensure proper power distribution and signal integrity throughout the design.

Following floor planning, the placement stage (Figure 5(b)) determines the precise locations of individual cells within the designated chip area. After placement, all cells synthesized from the ASAP7 PDK library were positioned optimally. This optimization process considers factors such as wire length, signal integrity, and power distribution to achieve an efficient layout design.

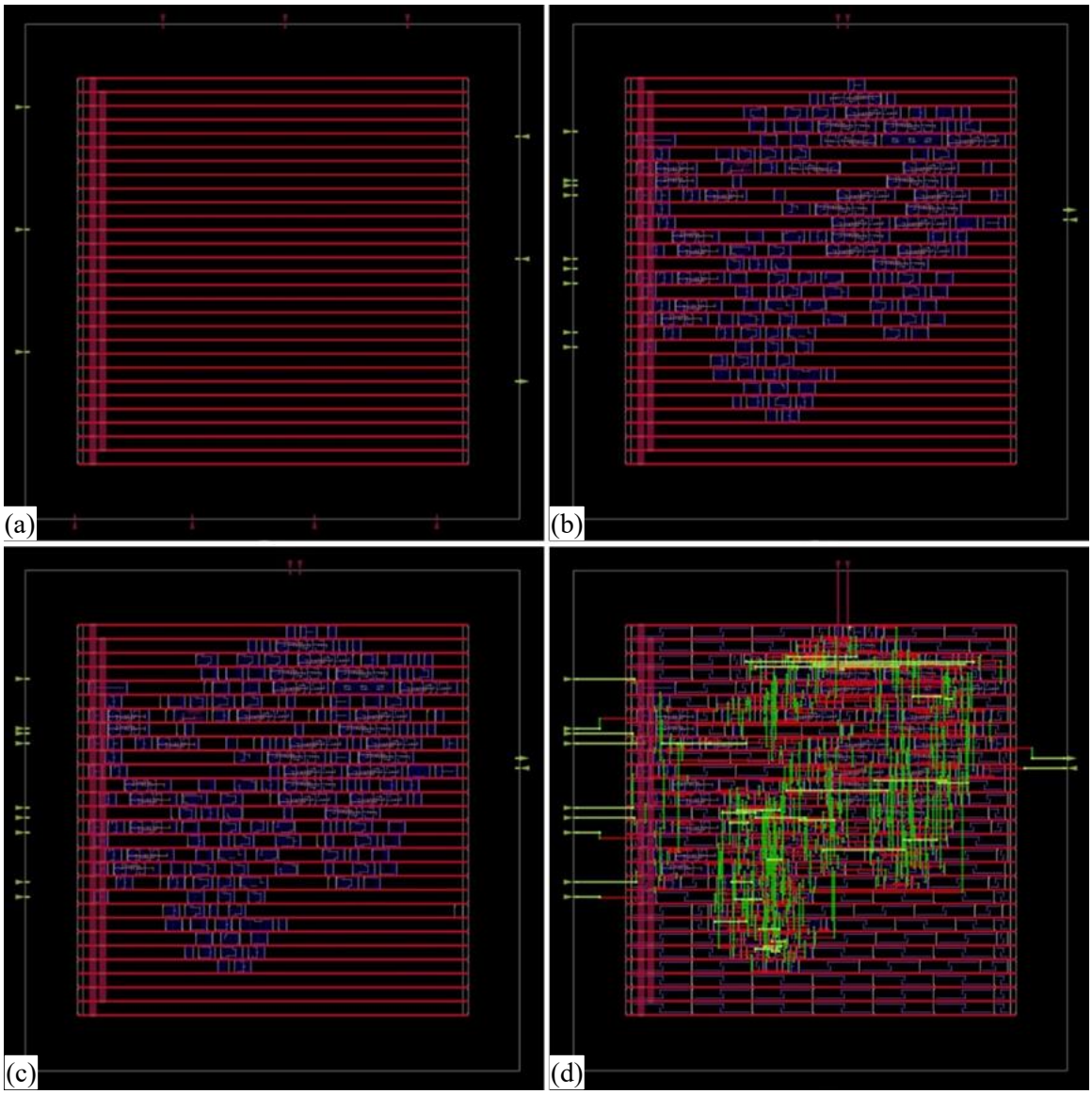

Figure 5. Layouts after (a) floor planning, (b) placement, (c) clock tree synthesis, and (d) routing.

In the CTS stage depicted in Figure 5(c), new buffers are added for clock distribution. This layout after CTS illustrates the optimization of clock tree structures to minimize skew and ensure the synchronous operation of all elements. The insertion of buffers is crucial for enhancing clock distribution and synchronization and ensuring accurate timing performance.

Finally, in the routing stage, as shown in Figure 5(d), the final layout after routing was presented. This layout demonstrates the establishment of physical connections between cells, with all the cells connected to the power supply. Routing involves the generation of metal layers and routing tracks to accommodate interconnections while adhering to design rules and constraints. The completion of interconnections ensures proper signal propagation and meets the design constraints, culminating in a finalized layout ready for fabrication or integration into other systems.

Finally, the finishing stage involves various post-processing steps, such as design rule checking (DRC), layout versus schematic (LVS) verification, and timing analysis to validate the physical layout against the original design intent. This ensures compliance with fabrication requirements and guarantees the functionality and reliability of the serializer design in real-world applications.

After the RTL-to-GDS flow, the final GDS layout depicted in Figure 5(a)–(d) is ready to be sent to a fabrication facility that supports the ASAP7 PDK for manufacturing. Alternatively, they can be integrated seamlessly into other systems, ensuring compatibility and interoperability with diverse electronic designs. This finalized layout represents the culmination of meticulous design and optimization efforts, ensuring the functionality, reliability, and manufacturability of serializer design for real-world applications.

RESULT AND ANALYSIS

An optimal balance between Performance, Power, and Area (PPA) is deemed paramount in VLSI design, as it directly affects the efficiency and competitiveness of integrated circuits. Careful consideration of PPA metrics ensures that performance requirements are met, power consumption is minimized, and silicon area utilization is maximized. In the subsequent sections, a detailed analysis of timing performance, power consumption, and area utilization is undertaken to elucidate the effectiveness of the design strategies employed.

Timing Analysis

Timing analysis serves as a critical assessment of the temporal performance of serializer design, ensuring adherence to specified timing constraints. Examining timing infractions, such as setup and hold violations, is essential to ensuring that the circuit operates correctly under various circumstances.

In the analysis, shown in Figure 6, the presence of positive setup slack of 42.45ps and hold slack of 33.49ps indicates that the design meets or exceeds the required setup and hold times. This signifies that the data signals were stable and settled before the arrival of the clock edge, ensuring proper latch behavior. The absence of setup or hold violations confirms the robustness and reliability of the design within the timing constraints imposed by the clock frequency specified in the .sdc file.

Moreover, based on the detailed logs presented in Figure 6, the minimum clock period was calculated to be 57.55ps, corresponding to a frequency of 17.37 GHz. Consequently, the maximum achievable data rate for this design was 17.37 Gbps, providing valuable insights into the performance capabilities of the serializer circuit.

Power Consumption

Power consumption analysis is crucial for evaluating the energy efficiency of serializer design. By examining the power distribution across different components and circuits within the chip, insights into the power utilization patterns and potential areas for optimization can be gained.

```
Startpoint: _261_ (falling edge-triggered flip-flop clocked by clk')
Endpoint: _260_ (falling edge-triggered flip-flop clocked by clk')
Path Group: clk
Path Type: max

  Delay    Time   Description
---------------------------------------------------------
   0.00    0.00   clock clk' (fall edge)
  32.95   32.95   clock network delay (propagated)
   0.00   32.95 v _261_/CLK (DFFLQNx2_ASAP7_75t_SL)
  28.83   61.77 ^ _261_/QN (DFFLQNx2_ASAP7_75t_SL)
   3.62   65.39 v _173_/Y (NAND2x1_ASAP7_75t_SL)
  19.40   84.79 v _174_/Y (OA211x2_ASAP7_75t_SL)
   0.11   84.90 v _260_/D (DFFLQNx1_ASAP7_75t_SL)
          84.90   data arrival time

 100.00  100.00   clock clk' (fall edge)
  32.66  132.66   clock network delay (propagated)
   0.03  132.68   clock reconvergence pessimism
         132.68 v _260_/CLK (DFFLQNx1_ASAP7_75t_SL)
  -5.34  127.35   library setup time
         127.35   data required time
---------------------------------------------------------
         127.35   data required time
         -84.90   data arrival time
---------------------------------------------------------
          42.45   slack (MET)
```

```
Startpoint: _272_ (falling edge-triggered flip-flop clocked by clk')
Endpoint: _272_ (falling edge-triggered flip-flop clocked by clk')
Path Group: clk
Path Type: min

  Delay    Time   Description
---------------------------------------------------------
   0.00    0.00   clock clk' (fall edge)
  33.17   33.17   clock network delay (propagated)
   0.00   33.17 v _272_/CLK (DFFLQNx1_ASAP7_75t_SL)
  26.58   59.75 v _272_/QN (DFFLQNx1_ASAP7_75t_SL)
   9.87   69.62 v _151_/Y (OA21x2_ASAP7_75t_SL)
   4.68   74.30 ^ _152_/Y (NOR2x1_ASAP7_75t_SL)
   0.09   74.40 ^ _272_/D (DFFLQNx1_ASAP7_75t_SL)
          74.40   data arrival time

   0.00    0.00   clock clk' (fall edge)
  33.19   33.19   clock network delay (propagated)
  -0.03   33.17   clock reconvergence pessimism
         33.17 v _272_/CLK (DFFLQNx1_ASAP7_75t_SL)
   7.74   40.90   library hold time
          40.90   data required time
---------------------------------------------------------
          40.90   data required time
         -74.40   data arrival time
---------------------------------------------------------
          33.49   slack (MET)
```

Figure 6. Detailed timing logs of max and min path.

Table 1. Power consumption breakdown (in Watts) of the serializer design.

Group	Internal power	Switching power	Leakage power	Total power
Sequential	3.85E-04	1.12E-05	4.05E-07	3.97E-04
Combinational	2.27E-04	1.97E-04	1.00E-06	4.25E-04
Macro	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Pad	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Total	6.12E-04	2.08E-04	1.41E-06	8.21E-04

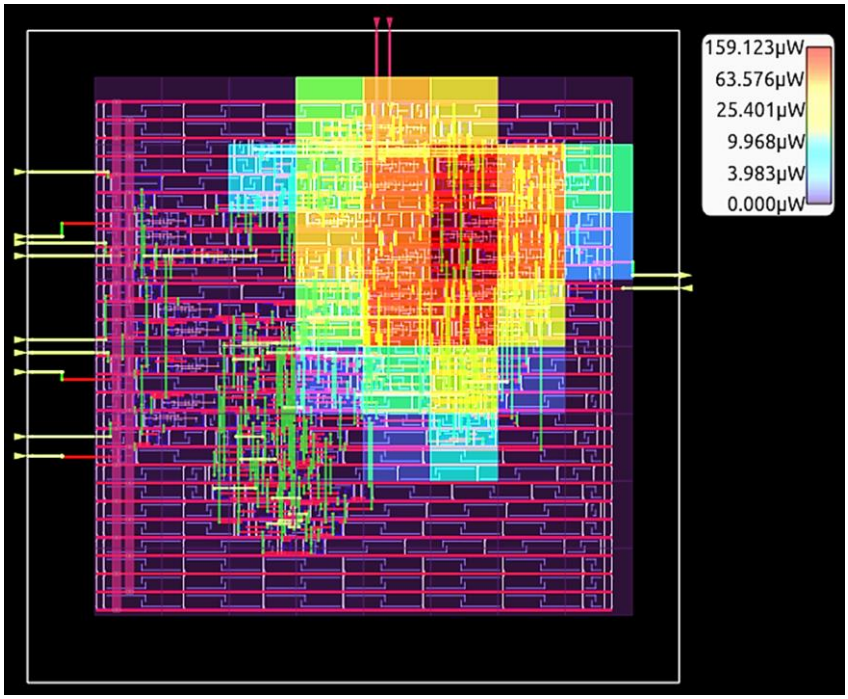

Figure 7. Power density distribution over the chip.

The power consumption breakdown reveals that combinational circuits account for 51.70% of the total power consumed, whereas sequential circuits contribute 48.30%. Further analysis indicates that internal power constitutes most of the power consumption at 74.5%, followed by switching power at 25.30%, and leakage power at 0.20%. This breakdown provides valuable insights into the dominant sources of power consumption in the serializer design (Table 1). The overall power requirement of the serializer design is measured to be 0.821 mW, as specified by the ASAP7 PDK with VDD set at 0.7V and VSS at 0V for the power supply voltage and ground, respectively.

Additionally, Figure 7 illustrates the power density distribution over the final chip layout, captured using OpenROAD GUI. This visualization offers a comprehensive view of the power distribution patterns across different regions of the chip, aiding in the identification of potential hotspots and areas requiring power optimization.

Area Utilization

Area utilization analysis provides insights into the efficient use of silicon real estate using serializer design. The serializer takes up 8.36 μm^2, whereas the chip takes up 22 μm^2, meaning that 38% of the available space is used. Figure 8 illustrates the placement density captured by OpenROAD GUI. This visualization offers a detailed representation of how the components and logic cells are distributed across the chip layout, providing insights into the spatial organization of the design. By analyzing the area utilization and placement density, designers can optimize the layout of the serializer design to maximize performance and minimize silicon waste.

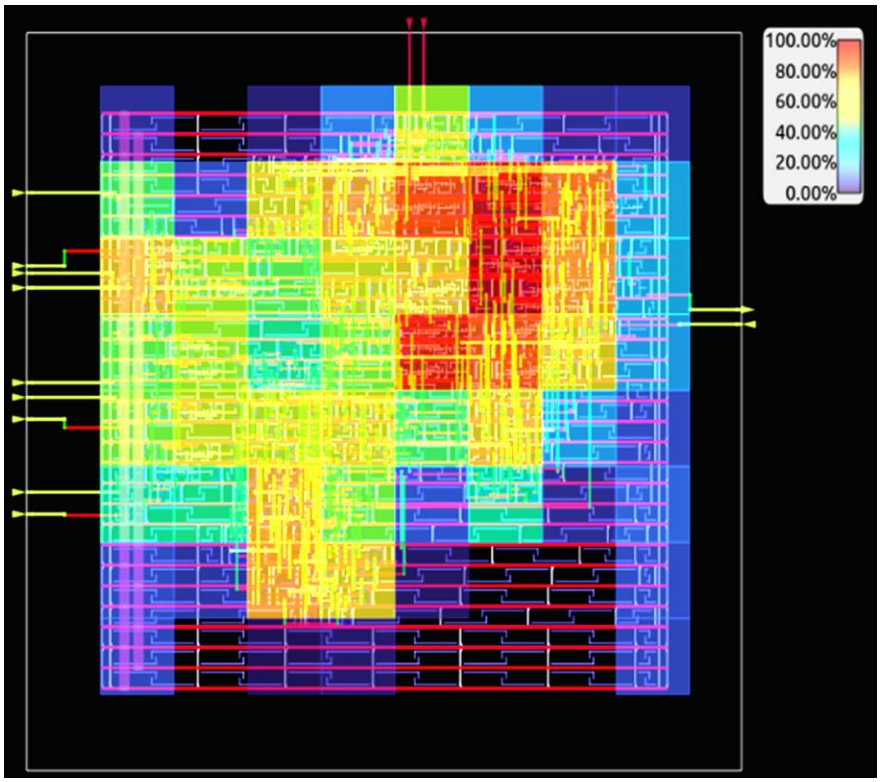

Figure 8. Placement density of cells over the chip.

CONCLUSION

This study successfully implemented a high-performance 32:1 serializer operating at an impressive 17.37 Gbps data rate using the open-source OpenROAD framework and 7 nm ASAP7 PDK. Through rigorous design efforts across the RTL-to-GDS flow, the serializer achieved remarkable efficiency, with a power consumption of 0.821 mW and an area utilization of 38 %, demonstrating the potential of open-source tools to enable sustainable semiconductor practices.

The comprehensive timing analysis validated operational reliability, while the power breakdown provided insights into optimization opportunities. The seamless navigation of the complex RTL-to-GDS transformation using OpenROAD underscores its robustness and adaptability, showcasing the capabilities of open-source tools in tackling intricate VLSI challenges. By leveraging collaborative efforts, this study took a significant step towards democratizing access to cutting-edge design methodologies.

Looking ahead, this research opens avenues for students and educators to explore serializer designs using open-source tools. By embracing the open-source community's collaborative spirit and leveraging freely available resources, students can gain practical exposure to industry-grade VLSI design flows, fostering hands-on learning and skill development in semiconductor design.

Acknowledgment

Sincere gratitude is expressed to the open-source community, particularly the developers of the OpenROAD framework and ASAP7 PDK, for their invaluable contributions and commitment to democratizing access to advanced semiconductor design tools and methodologies.

REFERENCES

1. Reda S. Overview of the OpenROAD digital design flow from RTL to GDS. In: 2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT); 2020; Hsinchu, Taiwan. IEEE; 2020. p. 1. DOI: 10.1109/VLSI-DAT49148.2020.9196319.

2. Jaiswal N, Gamad R. Design of a new serializer and deserializer architecture for On-Chip SerDes transceivers. Circ Syst. 2015;6:81-92. DOI: 10.4236/cs.2015.63009.
3. OpenROAD. (2024). The OpenROAD Project – Foundations and Realization of Open and Accessible Design. [online] Available from: https://theopenroadproject.org/
4. Sheldon R. (2023). SerDes (serializer/deserializer). [online] TechTarget. Available from: https://www.techtarget.com/searchstorage/definition/SerDes
5. Safwat S, Hussein EE-D, Ghoneima M, Ismail Y. A 12Gbps all digital low power SerDes transceiver for on-chip networking. 2011 IEEE International Symposium of Circuits and Systems (ISCAS); 2011; Rio de Janeiro, Brazil. 2011. p. 1419-22. DOI: 10.1109/ISCAS.2011.5937839.
6. Sharad M, Rao VSRP, Mandal P. A new double data rate (DDR) dual-mode duobinary transmitter architecture. 2011 24th International Conference on VLSI Design; 2011; Chennai, India. 2011. p. 12-7. DOI: 10.1109/VLSID.2011.52.
7. The-OpenROAD-Project (2020). GitHub - The-OpenROAD-Project/OpenROAD-flow-scripts: OpenROAD's scripts implementing an RTL-to-GDS Flow. [online] GitHub. Available from: https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts
8. Arizona State University. (2017). ASAP7 Calibre Decks. Predictive PDK (ASAP) – ASU Engineering. [online] Available from: https://asap.asu.edu/
9. Taraate V. Design constraints and SDC commands. In: ASIC Design and Synthesis. Singapore: Springer; 2021. Pages 139-151. DOI: 10.1007/978-981-33-4642-0_10.
10. Golshan K. Clock Tree synthesis. In: The Art of Timing Closure. Cham: Springer; 2020. DOI: 10.1007/978-3-030-49636-4_6.

