Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 9
september 2025

Implementation of High Speed Routing using Buffer Optimization

Keerthana V, Srividya P

Abstract

Introduction: High-speed on-chip communication is critical in modern VLSI design to meet stringent performance requirements. As technology scales down and circuit complexity grows, interconnect delay has emerged as a dominant factor affecting overall chip performance. Efficient routing plays a key role in ensuring fast and reliable signal distribution across the chip's metal layers. Among the techniques to enhance routing performance, buffer optimization stands out, this involves the strategic insertion and sizing of buffers along long interconnects to minimize delay, prevent signal degradation and balance load capacitance. Proper buffer placement not only improves signal integrity and timing but also mitigates challenges such as excessive loading, noise and power inefficiency

Objectives: To model different design cases by modifying design files to reflect various scenarios like input-register, register-register, register-output and input-output; To analyze critical timing paths between macros; To optimize the design to meet timing requirements; To perform timing-driven placement.

Methods: The project aims to improve the timing performance in VLSI systems by analyzing and optimizing key timing paths across different scenarios, including IP-to-boundary, register-to-register and standard cell connections, within both single and multi-clock domains. Various design cases are developed to reflect realistic System On Chip (SOC) architectures and static timing analysis is conducted to identify the most critical paths. Buffer insertion is then strategically optimized to reduce delays and enhance signal quality, with attention to placement and sizing to maintain efficiency in area and power usage. A timing-driven placement approach is implemented to ensure that high-priority nets are optimally positioned during layout. The overall workflow is compatible with the standard Register Transfer Level (RTL)-to-GDSII design flow, ensuring relevance and applicability in real-world chip design processes.

Results: A detailed analysis aimed to improve timing efficiency in VLSI systems by optimizing buffer insertion between macros, thereby enabling timing-driven placement. Multiple timing scenarios were examined, including interactions between fixed blocks and standard cells, as well as standard cell-to-cell and combinational logic paths. These case studies were used to model and analyze different signal transitions, such as input-to-register, register-to-register, register-to-output and direct input-to-output flows.

Conclusions: With the growing complexity of system-on-chip designs and the demand for faster on-chip communication, maintaining timing accuracy across long interconnect paths has become a significant concern. To address this, various timing scenarios were modeled including fixed block to standard cell, standard cell to standard cell and interface-based paths. The impact of buffer placement on timing performance was evaluated using detailed timing reports from different placement stages. The findings revealed notable improvements in slack, reduced HPWL and optimized routing conditions. Implementing these strategies within the OpenROAD design flow confirmed that buffer insertion effectively aids in timing closure, signal integrity and overall layout quality.

Keywords: Buffer Optimization, Timing-Driven Placement, Static Timing Analysis, OpenROAD.

1. Introduction

High-speed on-chip communication is critical in modern VLSI design to meet stringent performance requirements. As technology scales down and circuit complexity grows, interconnect delay has emerged as a dominant factor affecting overall chip performance. Efficient routing plays a key role in ensuring fast and reliable signal distribution across the chip's metal layers. Among the techniques to enhance routing performance, buffer optimization stands out, this involves the strategic insertion and sizing of buffers along long inter- connects to minimize delay, prevent signal degradation and balance load capacitance. Proper buffer placement not only improves signal integrity

and timing but also mitigates chal- lenges such as excessive loading, noise and power inefficiency. By integrating buffer optimization into the routing stage, de- signers can achieve faster signal transmission, improved timing closure and better utilization of routing resources making it an important step in high-performance digital design.

The relentless scaling of semiconductor technologies has enabled the integration of billions of transistors on a sin- gle chip, leading to increasingly complex System-on-Chip (SoC) designs. While this progression enhances computational capability, it also amplifies design complexity and timing challenges. In advanced technology nodes, interconnect delays have overtaken logic delays as the dominant factor affecting overall performance, making timing closure a key concern in physical design stages[1].

Achieving timing closure in such dense designs requires precise control over net delays and signal integrity. Buffer insertion has emerged as a critical optimization technique to mitigate long-path delays, improve slew rates and satisfy setup and hold constraints[2]. However, effective buffer planning must be performed early in the design cycle to minimize costly iterations. This approach advocates for early-stage analysis and optimization, shifting traditionally back-end tasks to earlier phases to accelerate convergence and improve design quality [6].

2. Objectives

The objectives of this work are to model different design cases by modifying design files to reflect realistic SoC scenarios such as input-to-register, register-to-register, register-to-output and input-to-output. Critical timing paths between macros are analyzed to identify delay bottlenecks and performance-limiting interconnects. The design is then optimized through buffer insertion and placement refinement to meet timing constraints Finally, a timing-driven placement approach is implemented to ensure that high-priority nets are optimally positioned, enabling effective timing closure and reduced routing congestion.

3. Methods

The methodology focuses on enhancing timing performance in VLSI systems by targeting critical timing paths across multiple design scenarios. These scenarios include IP-to-boundary, register-to-register and standard cell connections, evaluated within both single and multi-clock domain environments to ensure robustness under realistic operational conditions.

To reflect practical System-on-Chip (SoC) architectures, representative design cases are developed that incorporate hierarchical modules, macro blocks and long interconnects. Static Timing Analysis (STA) is performed at an early stage to identify the most timing-critical nets and paths across multiple corners and modes. Once identified, strategic buffer insertion is applied to reduce propagation delays, minimize slew degradation and improve overall signal integrity. Buffer placement and sizing are carefully optimized to achieve timing closure while maintaining efficiency in area, power and routing resources.

A timing-driven placement strategy is adopted to ensure that high priority nets are optimally positioned during layout. By integrating timing constraints directly into the placement stage, the methodology reduces the reliance on late-stage Engineering Change Orders (ECOs) and post-route fixes, thereby shortening design turnaround time. The approach also incorporates congestion awareness to prevent buffer-induced routing bottlenecks.

Pin Placement

Pin placement is the process of assigning physical locations to the input/output (I/O) pins of a chip or block, determining where signals enter and exit the design. In this project's flow, pin placement occurs after floorplanning and before standard cell placement, ensuring optimal connectivity for critical paths such as IP-to-boundary, register-to-register and standard cell connections. Correct pin placement in this flow directly impacts the effectiveness of the subsequent buffer optimization stage by enabling shorter, less congested critical nets, ultimately improving overall timing closure in the shift- left design approach.

Timing Driven Placement

Timing-driven placement (TDP) is a physical design technique that optimizes cell locations based on timing constraints rather than solely minimizing

wirelength, aiming to reduce delays along critical paths identified through Static Timing Analysis. In this project, TDP was implemented after global placement using OpenROAD, focusing on timing critical paths such as input-to-register and register-to-register connections. Cells along these paths were repositioned and complemented with strategically placed buffers to mitigate RC delays, im- prove slew rates and enhance slack. By prioritizing these nets during placement, the flow ensured that setup and hold requirements were consistently met across both single and multi-clock domains. This integrated approach reduced Half-Perimeter Wire Length (HPWL), improved timing margins and enabled high-speed routing while maintaining acceptable congestion levels. Additionally, the iterative STA feedback during placement allowed early identification and resolution of timing bottlenecks, resulting in a more robust and performance oriented final layout. The key differences between timing-driven and wirelength-driven placement strategies in chip design is that the timing-driven placement is all about ensuring the design meets timing requirements, especially for critical paths, but it tends to be more demanding in terms of resources. On the other hand, wirelength-driven placement focuses on keeping the total wirelength short, which helps save area and makes the process less complex.

Buffer Insertion

The process of buffer insertion is a key part of timing optimization during the physical design stage. Design tools analyze signal paths to detect nets with excessive delay, poor slew rates or heavy fanout. Buffers are then strategically inserted along these paths as shown in figure 1, thereby im- proving signal propagation and ensuring reliable timing. This is especially important for paths that lie on the critical timing path of the design. Buffer insertion is commonly integrated with timing-driven placement and optimization techniques, where Electronic Design Automation (EDA) tools continuously evaluate and improve path delays while maintaining functional correctness. It is also a central component of clock tree synthesis (CTS), where buffers are used to balance clock skew and ensure uniform clock arrival times across the chip. Overall, buffer insertion enhances signal integrity, improves timing closure and contributes to achieving overall design performance targets.

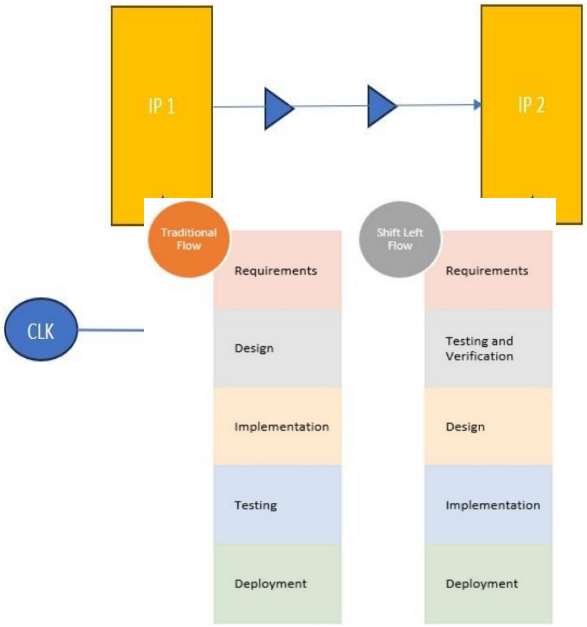

Fig. 1. Buffer Insertion

Shit Left Methodology

The shift left methodology is a software development approach that emphasizes the importance of integrating security, testing and quality assurance earlier in the development process, rather than later. The term ”shift left” refers to the idea of moving these activities to the left on the traditional software development timeline, which typically includes phases such as requirements gathering, design, implementation, testing, and deployment. In traditional software development, security, testing and quality assurance are often performed towards the end of the development cycle, just before deployment as shown in figure 2. However, this approach can lead to several issues like high cost of fixes, late discovery of bugs, delay in time to market etc.

Fig. 2. Traditional Flow vs Shift Left Flow

Design Specification

Improving the timing performance in VLSI systems by analyzing and optimizing key timing paths across different scenarios, including IP-to-boundary, register-to-register and standard cell connections, within both single and multi-clock domains. Various design cases are developed to reflect

realistic System On Chip (SOC) architectures and static timing analysis is conducted to identify the most critical paths. Buffer insertion is then strategically optimized to reduce delays and enhance signal quality, with attention to placement and sizing to maintain efficiency in area and power usage. A timing-driven placement approach is implemented to ensure that high-priority nets are optimally positioned during layout. The overall workflow is compatible with the standard Register Transfer Level (RTL)-to-GDSII design flow, ensuring relevance and applicability in real-world chip design processes. The different types of design cases that have been modeled are:

A. Timing Modeling for Fixed Blocks, Standard Cells and Combinational Logic: In physical design, it is essential to model various timing interactions between different elements, such as fixed-function blocks, standard cells and purely combinational paths. Timing between a fixed block and a standard cell must consider the rigid placement constraints of the fixed block, ensuring proper signal alignment and synchronization. Similarly, timing between standard cells must account for delay variations caused by placement and routing. Combinational connections, which lack clocked elements, must be carefully modeled to prevent excessive logic depth and meet path delay requirements.

B. Capturing Timing Characteristics of Different IPs: Each IP block has its own timing behavior, influenced by its internal architecture, performance targets and integration method. Accurate timing modeling of IPs involves specifying parameters like setup and hold times, internal delays, clocking schemes and latency. These timing attributes are critical for seamless integration and verification, ensuring the IP functions correctly within the overall design and interacts reliably with other system components.

C. Timing Paths Between IPs and Their Boundaries: When integrating IP blocks, it is necessary to define and model timing paths between internal logic and boundary pins. These paths include signals traveling from an IPs internal register to an output pin or from an input pin to internal logic. Accurate modeling of these paths ensures that the IP interfaces correctly with external modules and adheres to timing expectations at the system level. It also enables comprehensive static timing analysis during SoC integration.

D. Modeling Timing for Key Signal Transitions: Timing must be captured for various signal transitions within a design, including input to register, register to register, register to output and input to output. For input to register paths, it is important to ensure the data arrives before the capturing edge of the clock. Register to register timing defines sequential logic paths and ensures data is transferred correctly between flip-flops. Register to output paths model how data exits the module from a storage element, while input to output paths represent direct combinational transfers. Proper modeling of each case ensures the design meets setup, hold and propagation delay requirements.

4. Results

A detailed analysis aimed to improve timing efficiency in VLSI systems by optimizing buffer insertion between macros, thereby enabling timing-driven and performance-oriented routing. Multiple timing scenarios were examined, including interactions between fixed blocks and standard cells, as well as standard cell-to-cell and combinational logic paths. These case studies were used to model and analyze different signal transitions, such as input-to-register, register-to-register, register-to-output and direct input-to-output flows. The results highlighted that strategic buffer placement not only minimized propagation delay but also improved slew quality and reduced timing violations across critical paths. Furthermore, routing congestion was effectively managed, ensuring stable design performance under varying clock domains. This validates buffer optimization as a key methodology for achieving reliable timing closure, enhanced signal integrity, and scalable SoC implementations.

A. Analysis of Timing Paths in a Single Clock Domain

TABLE I

SINGLE CLOCK TIMING PATH ANALYSIS

Each path shown in Table I is characterized by key timing parameters including required time, arrival time and slack. The required time indicates the latest moment a signal can arrive without violating setup constraints, while the arrival time reflects the actual timing of signal arrival at the capture

Paths	Required	Arrival	Slack
1	1.65	0.20	1.45 (MET)
2	2.17	0.72	1.45 (MET)
3	2.10	0.57	1.53 (MET)
4	2.06	0.50	1.56 (MET)
5	2.17	0.60	1.56 (MET)
6	1.96	0.20	1.76 (MET)
7	2.07	0.20	1.87 (MET)

point. The slack, which is the difference between these two values, is positive for all paths analyzed, demonstrating that the design operates safely within the time limits.

B. Analysis of Timing Report for Multiple Clock Domain

TABLE II

MULTI-CLOCK TIMING PATH ANALYSIS

The timing analysis results demonstrated in Table II is an example that shows two independent paths originating from the same source flip-flop but constrained by two different clock domains: clock1

Paths	Clocks	Required	Arrival	Slack
1	Clock2	4.50	0.54	3.96 (MET)
2	Clock2	4.48	0.50	3.98 (MET)
3	Clock2	4.42	0.20	4.22 (MET)
4	Clock1	1.76	0.20	1.56 (MET)
5	Clock1	2.17	0.61	1.56 (MET)
6	Clock1	2.17	0.55	1.61 (MET)

and clock2. This scenario represents a common use case in modern VLSI designs, where components operate at varying frequencies and require individual timing constraints to ensure proper synchronization and data transfer. This multi-clock analysis ensures accurate timing validation, helping to avoid false positives or missed violations during static timing checks. Both paths meet their respective timing requirements, validating the designs timing integrity across multiple clock domains.

C. Analysis of Timing Report after Buffer Insertion

Buffering plays a key role in managing signal delay, improving transition quality, and ensuring that timing paths meet their respective setup requirements.

TABLE III

TIMING PATH ANALYSIS BEFORE BUFFER INSERTION

Paths	Required	Arrival	Slack
1	4.50	0.54	3.96 (MET)
2	4.48	0.50	3.98 (MET)
3	4.42	0.20	4.22 (MET)
4	1.76	0.20	1.56 (MET)
5	2.17	0.61	1.56 (MET)
6	2.17	0.55	1.61 (MET)

TABLE IV

TIMING PATH ANALYSIS AFTER BUFFER INSERTION

Paths	Required	Arrival	Slack
1	4.50	2.10	2.40 (MET)
2	4.48	0.61	3.87 (MET)
3	4.42	0.29	4.13 (MET)
4	1.76	0.29	1.47 (MET)
5	2.17	2.05	0.11 (MET)
6	2.17	2.00	0.17 (MET)

Table III presents the timing summary before buffer insertion, where arrival times are low and slack values appear large due to ideal or underestimated net delays.

After buffer insertion, as shown in Table IV, arrival times increase and slack is reduced. This reflects a more realistic post-layout delay model. Despite tighter slack margins, all paths still meet timing requirements, indicating successful and accurate timing closure. In both scenarios, buffer insertion is clearly instrumental in achieving timing closure. By controlling path delays and regulating slew rates, buffers ensure that signals arrive at their destinations within the required timing window.

5. Discussion

This project on “High-Speed Routing through Buffer Optimization” effectively demonstrated how buffer insertion enhances timing performance, signal integrity, and routing efficiency in VLSI design. By analyzing different path types across placement stages using the OpenROAD flow, the implementation resulted in improved slack, reduced wirelength (HPWL) and better overall layout quality.

Future enhancements may include leveraging machine learning for automated buffer placement, adapting the method to support multi-clock and multi-power domain designs, and combining timing and power optimization. Improving integration

with commercial EDA tools, automating constraint generation, and extending the approach to advanced nodes like 5nm would further increase its scalability and practical application in industry.

References

[1] J. Ahn, K. Chang, K.-M. Choi, H. Park, and T. Kim, "DTOC-P: Deep-learning-driven timing optimization using commercial EDA tool with practicality enhancement," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 43, no. 8, 2024.

[2] S. C. Lo, A. J. Barker, S. R. Whiteley, et al., "Simulation methodology for timing analysis and design optimization in digital superconducting electronics," in 2022 23rd International Symposium on Quality Electronic Design (ISQED), IEEE, 2022.

[3] C.-H. Chen, C.-N. J. Liu, W.-T. Tu, T.-C. Chen, and I. H.-R. Jiang, "Accurate estimation of buffered interconnect delay based on virtual buffering and multi-level cluster tree techniques," in 2024 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE, 2024.

[4] R. Linga, V. Vijayabhasker, and M. Sumalatha, "A low-power timing- error-tolerant circuit by controlling a clock," in 2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), 2024.

[5] W. Dghais, M. Souilem, and M. Alam, "Mixed-signal overclocked I/O buffers model abstraction for signal integrity assessment," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 3, 2019.

[6] X. Wang, Y. Sun, and M. Fang, "Reconfigurable clock tree design methodology for wide voltage scaling using custom buffer," in 2021 IEEE 14th International Conference on ASIC (ASICON), IEEE, 2021.

[7] S. N. Shahsavani and M. Pedram, "TDP-ADMM: A timing driven place- ment approach for superconductive electronic circuits using alternating direction method of multipliers," in Proceedings of the Conference, 2020.

[8] J. Monteiro, N. K. Darav, G. Flach, et al., "Routing-aware incremental timing-driven placement," in 2016 IEEE Computer Society Annual Symposium on VLSI, IEEE, 2016.

[9] N. Uysal and R. Ewetz, "An OCV-aware clock tree synthesis methodol- ogy," in 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), IEEE/ACM, 2021.

[10] P. Liao, D. Guo, Z. Guo, S. Liu, Y. Lin, and B. Yu, "Dreamplace 4.0: Timing-driven placement with momentum-based net weighting and lagrangian-based refinement," IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems, vol. 42, no. 10, pp. 3374–3387, 2023.

[11] J. Jung, G.-J. Nam, L. N. Reddy, I. H.-R. Jiang, and Y. Shin, "OWARU: Free space-aware timing-driven incremental placement with critical path smoothing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 9, 2018.

[12] L. Yuan, H. You, and G. Guo, "System static timing analysis method based on hierarchization," in 2023 International Symposium of Electron- ics Design Automation (ISEDA), 2023.

[13] S. Nikoli, G. Zgheib, and P. Ienne, "Timing-driven placement for FPGA architectures with dedicated routing paths," in 2020 30th International Conference on Field-Programmable Logic and Applications (FPL), IEEE, 2020.

[14] Y. Attaoui, M. Chentouf, Z. E. A. A. Ismaili, and A. E. Mourabit, "Clock gating efficiency and impact on power optimization during synthesis flow," in 2021 International Conference on Microelectronics (ICM), IEEE, 2021.

[15] X. Guo, X. Zhao, and L. Zhu, "Integration of shift-left updates into logic synthesis and macro placement," in 2024 Conference of

Science and Technology for Integrated Circuits (CSTIC), 2024.

[16] Z. Lin, M. Wei, Y. Chen, P. Zou, J. Chen, and Y.-W. Chang, “Electrostatics-based analytical global placement for timing optimiza- tion,” in 2024 Design, Automation and Test in Europe Conference (DATE), 2024.

[17] Srivatsa V. G., Arunkumar P. Chavan, and H. V. Ravish Aradhya, “De- sign of High Performance and Low-Power H-Tree Using MultiThreshold Split-IO Buffers,” Proc. of IEEE International Conference on Smart Electronics and Communication (ICOSEC), 2020.

[18] H. V. Ravish Aradhya et al., “Design and Implementation of Low Power Mitchell Algorithm Based Logarithmic Multiplier,” Proc. of IEEE RTEICT- Recent Trends in Electronics, Information and Communication Technology, 2017.

[19] Ravish Aradhya, Ranjit K. G., and Arunkumar P. Chavan, “Subthresh- old Adiabatic Logic (SAL) Based Building Blocks for Combinational System Design,” Proc. of IEEE RTEICT, 2017.