[1] Sanjana K

[2] Dr. Uma B V

Digital Implementation of High-Speed PCIe Switch

Abstract: - The exponential increase in capacity and performance of Application Specific Integrated Circuits (ASICs) has made area, timing, and power consumption crucial considerations in Very Large-Scale Integration (VLSI) IC design. Logic synthesis, which translates RTL (Register-Transfer Level) designs into gate-level netlists, plays a key role in achieving the required objectives w.r.t timing and area. A bottom-up synthesis approach is employed to translate RTL descriptions into gate-level netlists, effectively bridging high-level synthesis and automated physical design. Logic Equivalency Checks (LEC) with the Cadence Conformal tool ensure functional consistency between the netlist and RTL. Evaluations at a 1 GHz operating frequency reveal that low optimization efforts lead to increased gate counts, area, and power consumption, resulting in synthesis failures. In contrast, high optimization reduces instance counts from 1.71M to 1.65M, area from 2.6M to 2.53M gates, and power from 761.51mW to 702.77mW, while effectively mitigating setup and hold violations through buffer management. The design successfully passes LEC, highlighting the importance of optimization in achieving efficient ASIC design.

Keywords: Logical Equivalence checks PCIe, Worst Negative Slack, Total Negative Slack, Compare points, Key points.

I. INTRODUCTION

With rapid advancements in VLSI design integration over the past few decades, EDA tools have become crucial for the design, verification, and debugging of larger digital circuits. These tools provide numerous opportunities to design a variety of electronic chips containing integrated circuits. Meeting functional and performance goals within a design time schedule and associated costs underscores the importance of EDA tools[1]. They minimize the time required to design complex ICs, eliminate manufacturing errors, reduce costs, and optimize IC design. Achieving better performance and high-quality results by meeting timing and area goals presents a challenge in the logic synthesis of hierarchical modular designs, standard-cell library and design constraints [2].

Logic synthesis is the conversion of a high-level description of a design into an optimized gate-level representation based on a standard cell library and design constraints. This highly automated procedure bridges the gap between advanced synthesis and physical design automation [3]. The Logical Equivalence Checks (LEC) method is used to verify the logical equivalence between the RTL and the netlist It has become an integral part of the design flow, especially for later manual changes such as engineering change orders (ECOs). LEC is a comprehensive verification method that provides full design functionality without test vectors[4].

PCIe (Peripheral Component Interconnect Express) is a high-speed connection standard primarily utilized for data transfer between electronic components. It is commonly found in desktop and mobile computers, server systems, and also in devices like set-top boxes and gaming consoles. At its core, PCIe establishes a direct point-to-point connection between two compatible devices, typically linking a motherboard with an expansion card or storage device, such as an SSD or hard drive. The standard employs differential signaling to transmit data via separate pairs of copper wires, enabling data rates of up to 16 GT/s. To optimize performance and ensure compatibility among various devices, the PCIe standard supports multiple lane configurations, allowing connections between components based on their bandwidth needs. For example, wider lanes like x16 are usually allocated for graphics cards that demand significant bandwidth for high-resolution content, whereas narrower lanes like x1 are designated for lower-speed peripherals such as USB or SATA ports[7].

II. METHODOLOGY

Logic synthesis is an essential process that transforms a high-level description of a digital design into an optimized gate-level representation, acting as a bridge between high-level synthesis and physical design automation. This

[1] M.Tech in VLSI & Embedded Systems, Department of ECE, RVCE Bengaluru, Karnataka, India. sanjanak.lvs22@rvce.edu.in

[2] Department of ECE, RVCE, Bengaluru, Karnataka, India. umabv@rvce.edu.in

process is crucial for creating digital systems, particularly synchronous circuits. The level of design automation and logic synthesis is influenced by market-driven decisions that dictate the preferred design styles.

Initially, logic synthesis converts RTL (Register Transfer Level) descriptions written in Verilog into a technology-independent netlist based on Generic TECHnology (GTECH) libraries. Following this, it utilizes various optimization rules and algorithms to generate a technology-specific netlist. Optimization can be carried out at three levels: architectural, logic, and gate levels, each addressing specific design aspects to enhance performance and efficiency.

Architectural optimization focuses on high-level synthesis tasks that work directly with the Hardware Description Language (HDL). This involves activities such as sharing common sub-expressions, selecting resources, reordering operators, and recognizing arithmetic expressions for data path synthesis. These tasks are performed during the optimization of unmapped designs and are guided by user-defined constraints and coding styles. By improving the high-level structure, architectural optimization seeks to create a more effective overall design before lower-level optimizations are implemented.

Design for Testability (DFT) is a vital component of VLSI ASIC design, allowing for the integration of testing features during the design phase to enhance fault detection. Structured DFT techniques, like SCAN and Built-In Self-Test (BIST), improve the controllability and observability of sequential elements. The SCAN method simplifies test pattern generation by transforming sequential elements into combinational logic. This is achieved by connecting flip-flops in series to create scan chains, enabling arbitrary values to be shifted into and out of the circuit's internal state. Various styles of scan flip-flops, especially multiplexed scan cells, are used for their compatibility with modern design approaches and support from automation tools.

Logical Equivalence Checking (LEC) is a critical verification method for comparing RTL designs with their corresponding netlists, serving as an integral part of the design workflow. This comprehensive verification process assesses design functionality without relying on test vectors, facilitating quick validation of design changes. LEC consists of two phases: setup and verification. During the setup phase, RTL constraints are established, while the verification phase involves mapping, comparison, and diagnosis of the designs. Key components, such as primary inputs and outputs, are mapped between the reference model (RTL) and the updated model (netlist) for comparison. The diagnosis phase identifies non-equivalent points and potential error candidates, using visual aids to enhance analysis and debugging.

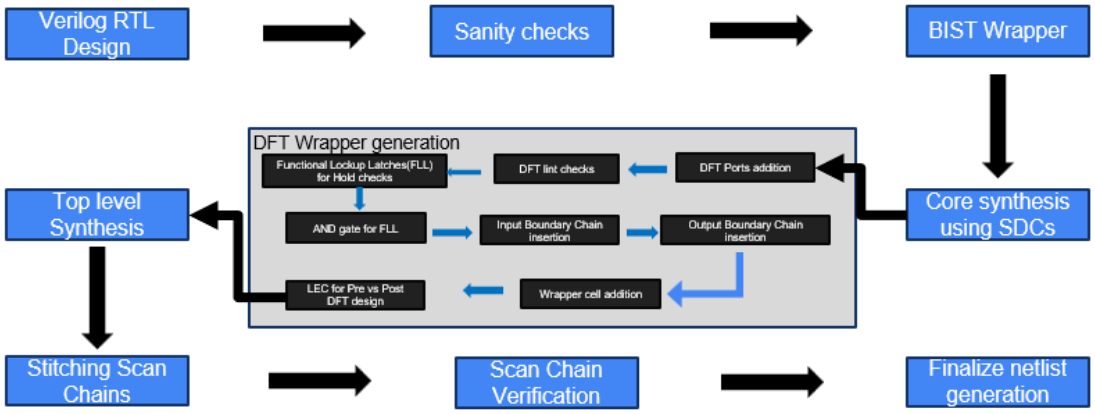

Figure 1: PCIe synthesis flow

Synthesis is executed twice during the design flow: initially after Test Built-In Self-Test (TBIST) insertion, which focuses on the block core, and subsequently after Design for Test (DFT) insertion, yielding the final netlist that is provided to the layout team. The corresponding flowchart is illustrated in Fig 1.

The synthesis process for a PCIe (Peripheral Component Interconnect Express) block encompasses several essential steps tailored to align with PCIe standards. It begins with the input of the RTL design, represented in a hardware description language (HDL) such as Verilog or VHDL, accurately capturing the functionality of the PCIe protocol, which includes transaction, data link, and physical layers. Next, the synthesis tool checks for syntax errors and conducts basic functional verification to ensure the RTL code represents a coherent design. The tool then optimizes

the RTL code using techniques like logic minimization, pipelining, and retiming to improve performance and throughput.

Following optimization, the RTL code undergoes technology mapping, where it is linked to a technology library containing standard cells optimized for high-speed interfaces. This step produces a gate-level netlist that meets PCIe signal integrity and timing specifications. Constraints related to timing, area, and power are applied to ensure adherence to PCIe protocols, followed by static timing analysis (STA) to assess timing characteristics both before and after synthesis.

A synthesis report is generated, offering insights into area, timing, and power characteristics, which helps designers refine the RTL code or constraints as needed. Finally, post-synthesis verification includes functional checks and equivalence verification to confirm that the synthesized netlist upholds the protocol's integrity, ensuring the reliable operation of the PCIe block.

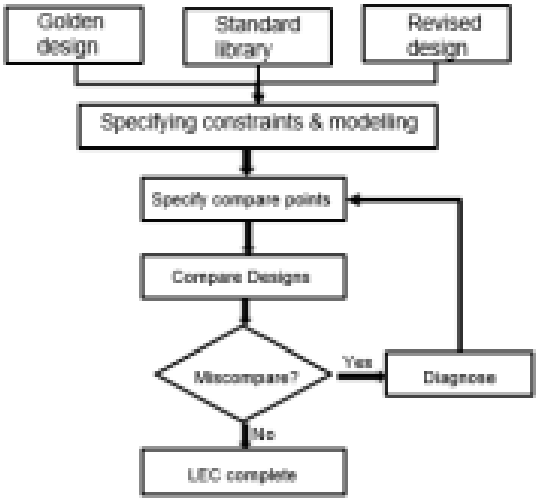

Figure 2: LEC design flow

Logical Equivalence Checking (LEC) is a crucial verification step in the design flow of a PCIe block, as illustrated in Fig 2. This process ensures that the synthesized design (netlist) functions exactly as specified in the original RTL design. The procedure begins with preparing the RTL design, where the PCIe block is described using a hardware description language (HDL) such as Verilog or VHDL, followed by an initial functional verification to confirm correct operation under various conditions.

Once the RTL code is synthesized into a gate-level netlist using the appropriate synthesis tools, synthesis reports are generated, detailing the area, timing, and power characteristics. The next step involves setting up the equivalence checking environment, which includes defining design constraints and creating test benches to validate functional correctness.

LEC tools, such as Cadence Conformal or Synopsys Formality, are then utilized to load both the original RTL design and the synthesized netlist for equivalence checking. This process involves logic simplification and gate-level simulation. The results are carefully analyzed to verify if the designs are equivalent, and any discrepancies are addressed using the tool’s debugging features, with necessary modifications made to the RTL or constraints.

Finally, the LEC process is repeated to ensure all discrepancies have been resolved, followed by functional validation through simulations to confirm that the behavior matches, particularly in corner cases relevant to PCIe functionality. This systematic approach is vital for verifying the reliability and correctness of the PCIe implementation before proceeding further in the design process.

III. Results & Discussion

The synthesis reports generated by the Genus tool are carefully analyzed and organized into tables. Timing adjustments are made to refine the synthesis process. Table 3.1 presents a comparison of timing, area, and power results for the case study, showcasing the differences between synthesis conducted with and without constraints, as well as the results from scan synthesis. The data indicates that the total number of instances, total area, and total

power with a low optimization effort level are approximately 24%, 18%, and 26% higher, respectively, compared to those achieved with a high optimization effort level.

Table 3.1: Synthesis result for different optimization effort level

Parameter	Effort level in core synthesis		Effort level final synthesis	
	Low	High	Low	High
Total Instance Count (Million instances)	2.04	1.717	2.47	1.708
Total Area (Million gates)	2.86	2.64	2.99	2.43
Total Power(mW)	799.69	761.51	702.77	668.4

When the mapping effort level is set to low, optimization is not conducted effectively, leading to an increase in the number of gates, area, and consequently, power consumption. From the results of this experiment, it can be concluded that achieving optimal performance in terms of timing, area, and power requires setting a high mapping effort level. The netlist produced during the synthesis phase for the core is compared against the golden netlist (RTL design) to verify logical equivalence. Key points in the revised netlist are evaluated against their corresponding counterparts in the RTL design for various effort levels. The summary report is presented in Table 3.2 below.

Table 3.2: LEC result for different optimization effort level

Key Points	Low Effort Level		High Effort Level	
	Golden	Revised	Golden	Revised
PI	2532	2532	2526	2526
PO	2635	2635	2635	2635
BBOX	38	38	38	38
NEQs	312		0	
State key points	195024		195024	
Status	FAIL		PASS	

When mapping and optimization are not accurately applied due to a low mapping effort level, non-equivalent points can arise. This misalignment means that key points are not mapped correctly, leading to a failure in the equivalence check. If the compared key points are equivalent at the end of the report, the comparison result is marked as PASS.

Static Timing Analysis (STA) is conducted for a single mode and corner. The STA summary reports indicate the total number of paths that were violated during setup and hold checks. By utilizing various switches, the number of violating paths and the negative slack for both setup and hold timing checks was significantly reduced. The results, including the initial and final counts of violating paths and the negative slack for setup and hold, are summarized in Tables 3.3 and 3.4, respectively.

Table 3.3 Setup Summary results

	WNS (Worst Negative Slack) (ns)	TNS (Total Negative Slack) (ns)	Violating Paths
Initial	-0.225	-220	1350
Final	-0.015	-0.045	10

Table 3.4 Hold Summary results

	WNS (Worst Negative Slack) (ns)	TNS (Total Negative Slack) (ns)	Violating Paths

Initial	-0.03	-3.417	15
Final	-0.01	-0.072	2

The violations in setup and hold were reduced by the tool using different approaches where totally, 60 repeaters were added, 1 buffer was deleted and approximately 600 cells were resized, or VT swapped for reducing or fixing setup violations and 1000 buffers were added and 10 cells were resized, or VT swapped for reducing hold violations.

IV. CONCLUSION

The synthesis and Logical Equivalence Checking (LEC) processes performed for the PCIe module revealed notable enhancements in instance count, area, and power consumption when a high optimization effort was applied. The synthesis results indicated that increased optimization during both the core and final synthesis stages led to a decrease in total instance count, reduced area, and lower power consumption, all contributing to a more efficient and compact design. This high-effort optimization strategy proved beneficial in achieving optimal design results.

The LEC findings confirmed the logical equivalence between the synthesized netlist and the golden RTL design. While the low optimization effort level resulted in some non-equivalent points, the high effort level successfully ensured that all key points were equivalent, yielding a pass status for the LEC.

Moreover, the Static Timing Analysis (STA) results showed that both setup and hold violations were effectively reduced by the tool, with significant improvements noted in the final analysis. Techniques such as buffer insertion, cell resizing, and voltage threshold swapping were instrumental in addressing these timing challenges, ultimately leading to a more robust design.

REFERENCES

[1] J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs A Practical Approach*, Springer Science Business Media, ISBN978-0-387- 93819-6,2009

[2] I.Pomeranz, "*Diagnostic Test Point Insertion and Test Compaction*," in IEEE Transactionson Very Large-Scale Integration (VLSI) Systems, vol.31, no. 2, pp.276-285, Feb.2023

[3] Y.Kwon, I.Hanand Y.Shin, "*Clock Gating Synthesis of Netlist with Cyclic Logic Paths*," 2019 IEEE / ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO,USA,2019

[4] Cadence Design Systems,"*Cadence User Guides*," Cadence Design Systems, 2018, [Online] Available: https://support. cadence.com

[5] T.Le and J.Di, "*Golden reference matching for gate- level netlist functionality identification,"* 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017

[6] E.M. Rudnick, V. Chickermane, P. BanerjeeandJ.H. Patel, "*Sequential circuit test ability enhancement using an on scan approach*," in IEEE Transactions on Very Large- Scale Integration (VLSI) Systems, vol.3 no.2, June 2015

[7] E. M. Rudnick, V. Chickermane and J. H. Patel, "*An observability enhancement approach for improved testability and at-speed test*," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 8, Aug. 2014

[8] Y.Higami, T.Inamoto, S.Wang, H.Takahashi and K. K. Saluja, "*Improving of Fault Diagnosis Ability by Test Point Insertion and Output Compaction*," 2023 International Technical Conference on Circuits / Systems, Computers, and Communications (ITC-SCC), Jeju, Korea, Republicof, 2023.

[9] Microchip Technology, "*Conformal LEC Basics*," Microchip Technology, 2018, [Online] Available: https://confluence. microchip.com/

[10] D. Bhattacharya and N.K. Jha, *"FinFETs: From Devices to Architectures*," Advances in Electronics}, vol. 14, July 2014. [Online] Available: https://doi.org/10.1155/2014

[11] M.D. Edwards, Automatic Logic *Synthesis Techniques for Digital Systems*, Macmillan New Electronics Series, ISBN 978-0-333-55569-9, 2022.

[12] N. Gustavo, W. Gilson, and R. Ricardo, *Protecting chips against hold time violations due to variability*, Springer Netherlands, ISBN 978-94-007-2427-3, 2019

[13] C. H. Gebotys and M. I. Elmasry, *Integration of algorithmic VLSI synthesis with testability in corporation*, Proceedings of the IEEE 1988 Custom Integrated Circuits Conference, Rochester, NY, 2019.

[14] S.S.K. Chiuand C.A. Papachristou, *Abuilt-inself- testing approach for minimizing hardware overhead*,[1991 Proceedings] IEEE International Conference on Computer Design:VLSI in Computers and Processors, Cambridge, MA, 2021.