2025 8th International Conference on Circuit, Power & Computing Technologies (ICCPCT) | 979-8-3315-4317-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCPCT65132.2025.11176538

Optimizing RTL-Based FPGA Designs for ECU Performance Enhancement: Impact of Hierarchical Flattening Using Xilinx Vivado

PULIMI MAHESH
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, India
pulimimahesh@velhightech.com

M SURESH CHINNAPAMTHY
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, India
sureshchinnathampy@velhightech.com

M PARTHIBAN
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, India
Parthiban.ece@velhightech.com

RR RAJARAJAVIGRAMAN
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala Engineering College,Avadi, India
rajarajavigramanrr_ece21@velhightech.com

SUHAIB AKTHAR
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala EngineeringCollege, Avadi, India
akthar14@gmail.com

S YESWANTH
Department of Electronics and Communication Engineering, Veltech High Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, India
yeswanths_ece21@velhightech.com

***Abstract* - A Performance, Low-Power, and Optimized FPGA Design for Real-Time Processing in Automotive Applications for Functional Safety of Electronic Control Units (ECUs) Project: The impact of hierarchical flattening on RTL designs synthesized through XILINX VIVADO for ECU optimization Structured design techniques provide advantages such as modularity, legibility, and reuse but can incur increased latency and resource overhead. Hierarchical flattening removes module bounds, allowing aggressive optimization at the cost of modularity. We analyzed three levels of flattening (None, Rebuilt, Full) to see their effects on synthesis time, area utilization, power efficiency and timing performance. The work includes hardware validation on FPGA board, Power and Timing analysis using Vivado tools such as report_power, report_timing_summary for hierarchical vs flattened designs. The results show that flattening decreases power consumption by up to 10% and synthesis time and area usage by 20% and 15%, respectively, therefore making the optimization strategy suitable for ECU. Aggressive flattening though can result in timing violations and makes debugging difficult. This is the best synthesis strategy presented here that gives the FPGA designers an option to balance performance and modularity, thereby allowing efficient ECU design.**

***Keywords*— ECU Optimization, Hierarchical Flattening, FPGA Synthesis, Verilog HDL, Xilinx Vivado, Power Optimization, Timing Analysis.**

I. INTRODUCTION

In digital design, there are two design methodologies that are fundamentally different from each other, hierarchical, and flattened design, and while there's a degree of overlap, we tend to see each methodology heavily used in specific contexts, either on FPGA-based designs or ASIC development. These methods reflect distinct design priorities, and therefore, appropriate for different stages of the hardware development lifecycle.

Most systems are designed in a hierarchical manner, meaning the system is decomposed into several smaller, modular blocks or submodules that have different functions. His organization promotes greater modularity, readability, and reusability of code, making it simpler to debug, maintain, and expand complex designs. Entity designs, hierarchical design is extremely suitable for large scale ECU architectures based on FPGA, the separation of different functional units can lead to more efficient cooperation, iterative development, and reuse.

Flattened designs, in contrast, remove module bounds with synthesis, tangling all submodules into a shocking, single structure. This practice enables aggressive logic optimizations by FPGA tools such as Xilinx Vivado, decreasing delays and resource consumption. For a typical automotive use case, such as real-time processing of ECU signals, this is an area-based design that is suitable for timing critical design due to relatively small timing delays, as well as power efficiency. Hierarchical designs are helpful for organization and modularity, but introduce the performance overhead of additional logic boundaries. On the other hand, flattened designs improve speed and optimization, but with a trade-off in terms of decreased modularity and increased debugging difficulty. And so, selecting the right degree of hierarchical flattening in ECU-based FPGA implementations, leading to an optimal trade-off between performance, power dissipation and maintainability is crucial.

A. Hierarchical Design in FPGA-Based ECUs

FPGA-based Electronic Control Units (ECUs) often employ a hierarchical design following a modular approach that breaks down the design into several submodules, each responsible for a specific function like sensor data acquisition, signal processing, and control logic.

979-8-3315-4317-4/25/$31.00 ©2025 IEEE

Such separation of the models improves modularity, reuse and debugging, making more manageable and easier to scale complex ECU architectures. Control tasks such as fuel injection, ignition timing, and throttle control can be developed as independent modules and can be modified separately without affecting the overall system. Hector respects the timing constraints of all modules at design time through the hierarchical architecture. Moreover, they use more FPGA resources, thereby consuming more power and area, which is a major issue in power-sensitive automotive applications.

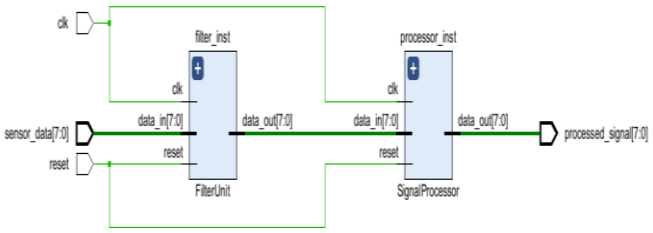

Fig. 1. *Hierarchical flattening in digital design*

B. Flattened Design for Performance Optimization in ECUs

Unlike flattened designs that eliminate module edges, merging all logic into a single format. This allows FPGA tools such as Xilinx Vivado to do aggressive optimization of timing closures, power consumption and FPGA resource usage. Flattening is advantageous for real-time ECU applications, including sensor fusion in LiDAR-based autonomous driving (such as adaptive cruise control), as minimized processing latency must be achieved for fast decision making. Flattened designs can aid high-performance ECUs as they instinctively improve speed and reduce latency since there is no overhead of hierarchical interconnections. The downside to flattening, however, is that it loses modularity and therefore becomes harder to debug and maintain. Debugging complexity will be a crucial limitation if any modifications or updates are necessary, especially in safety-critical automotive ECUs (e.g., braking, airbag controllers).

C. Choosing the Right Approach for ECU-Based FPGA Design

Hierarchical and flattened design approaches are a matter of preference based upon the ECU application requirements. We have already mentioned that, for ECU architectures that are flexible and extendable, hierarchical design is the preferred choice, when independent maintenance and upgrade of functions are required. On the other hand, flattened designs are more efficient in the case of performance-critical automotive applications in which latency is already the dominant design goal alongside low power efficiency. One such [hybrid] approach can be achieved by using the Flatten Hierarchy Option of Vivado, which enables the selective flattening of timing-critical portions while maintaining the modularity of the non-performance critical sections of the ECU. Hierarchical flattening strategies can be optimized for FPGA-based ECU designs, which can achieve a balance among performance, power efficiency, and maintainability, leading to become efficient and reliable solutions for the next-generation automotive systems.

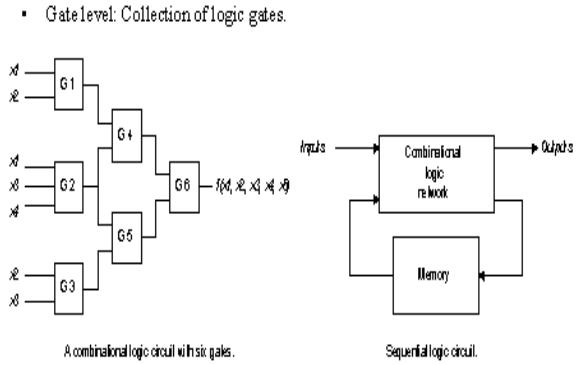

Fig. 2. *Hierarchical flattening in digital design*

D. Future Directions

You are familiar with trends in FPGA-based design methods for Electronic Control Unit (ECU) optimization. Its logic can be applied to future research and the development of hybrid combinations that bring together the benefits of hierarchical and flattened designs. The design of variability will further be staged across hierarchical design structures that optimize for modularity as well as for performance in an optimization task that leverages said machine learning-based skills.

However, flattened designs will also rely on more advanced optimization algorithms to offset readability and maintainability issues. Furthermore, as vehicle ECUs become more sophisticated, FPGA architectures will be designed to support adaptive reconfiguration, enabling cards to be dynamically reconfigured in response to changing workload requirements to enhance power efficiency, performance, and fault tolerance. Finally, leveraging high-level synthesis (HLS) will help accelerate the design process to move easily between software and hardware development.

As vehicles become autonomous and AI-led, real-time decision-making and data processing will need to be performed at significantly lower power budgets, creating the need for FPGA-based ECUs across such automotive systems. New design paradigms of new automotive systems will be able to be developed that will allow to balance modularity with optimization, reliability and scalability in operating automotive systems of the future.

II. METHODOLOGY

Designing FPGA-based Electronic Control Units (ECUs) involve understanding a better and structured way of achieving the requirements in a lesser time, fewer resources consumed, and a maintainable solution. The process begins and continues with system specification; requirement analysis (what the ECU must do, performance, and power constraints). This initial phase determines the most suitable design for the application, whether hierarchical or flattened.

The hierarchical style decomposes a system is into units/sub-systems and each unit-sub-system handles a specific category of interaction. These modules are designed separately, tested, and then they link to build a complete system. This tends to make designs easier to debug, reuse, and scale over time. HDLs (hardware description languages) such as VHDL or Verilog are usually used by engineers to realize the modules with a well-defined and organized structure.

Conversely, a flattened design provides no modular boundaries, organizing everything into one integrated and optimized structure. However, during synthesis and implementation the design tools optimize the logic, removing any non-favorable delays and minimizing resource utilizations. This is a perfect approach for performance-sensitive applications, where every bit matters.

But, no matter how you approach, verification and validation are intrinsically mandatory. Design visualization and reconstruction is verified in the form of simulation where engineers use tools such as ModelSim or Vivado to simulate the design to confirm that it behaves as expected. Timing analysis is performed on synthesized designs to verify that speed and power requirements are met by each component. The design is tested on real FPGA hardware often using a hardware-in-the-loop (HIL) setup that provides an opportunity for validation of real-world performance, but before complete deployment in the ECU.

The workflow for this project involves the synthesis and evaluation of RTL Verilog designs using Xilinx Vivado. The aim is to analyze the impact of hierarchical flattening on various performance metrics, including area, timing delay, power consumption, and resource utilization. The study progresses through several well-defined stages: Design Creation, Hierarchical Flattening, Synthesis, Simulation, and Analysis.

The following sections explain each stage in detail and are accompanied by a flowchart that illustrates the step-by-step process.

a) Design Creation: The initial step involves creating RTL designs in Verilog, ranging from simple combinational circuits to complex pipelined architectures. These designs are structured hierarchically, ensuring modularity and readability.

The designs are prepared with an emphasis on scalability and adaptability to allow for different levels of hierarchical flattening.

b) Hierarchical Flattening: In this stage, the "Flatten Hierarchy" synthesis option in Xilinx Vivado is applied to the RTL designs. Three levels of flattening are evaluated:

- **None**: Maintains the hierarchical structure without any flattening.
- **Rebuilt**: Partially flattens the design, rebuilding certain modules.
- **Full**: Completely flattens the design, removing all module boundaries.

Each level is applied to understand its impact on the subsequent stages of synthesis and analysis.

c) Synthesis: The RTL designs undergo synthesis using Xilinx Vivado, where they are transformed into gate-level netlists. During this process, various optimizations are performed based on the selected flattening level. Metrics such as resource utilization, area, and power are recorded for comparison.

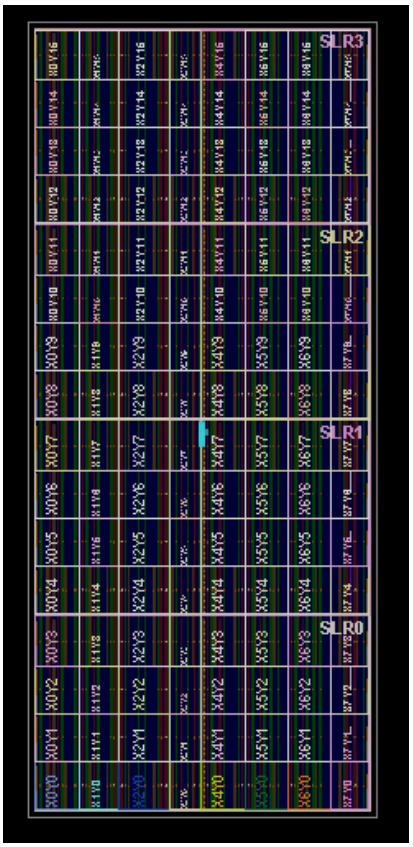

Fig. 3. *resource mapping image.*

d) Simulation: The synthesized designs are simulated to verify their functional correctness. This step ensures that the flattening process does not introduce errors or compromise the intended behavior of the designs.

e) Analysis: The final stage involves analyzing the results of synthesis and simulation. Key metrics such as timing delay, power consumption, and resource utilization are compared across the three levels of flattening. The trade-offs are evaluated to identify scenarios where hierarchical flattening provides significant benefits or drawbacks.

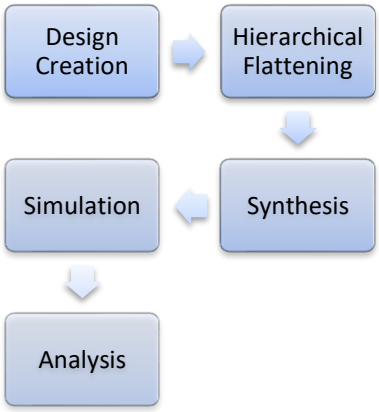

Fig. 4. *Workflow Chart*

Fig. 4. This workflow chart shows the complete process of the implementation of the hierarchical flattening in the FPGA board using the Xilinx Vivado.

III. Stimulation and Result

A. *Simulation Output:*

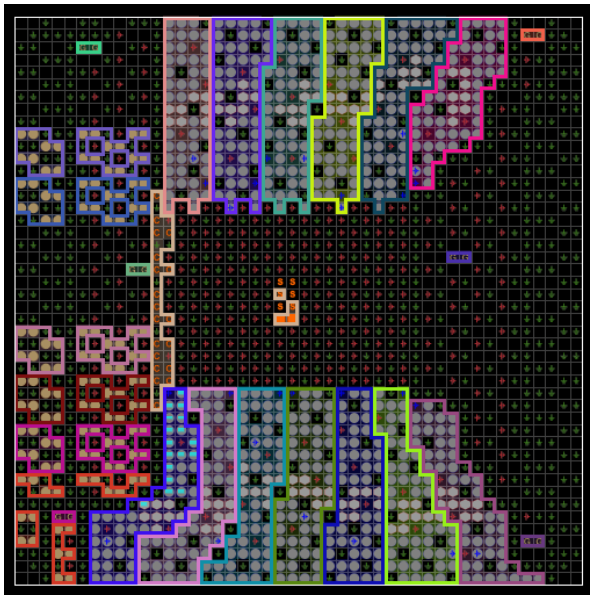

Fig. 5. *Input/Output Design*

Fig. 5. Is the design of the hierarchical fattening on the RTL design in the FPGA using the Xilinx Vivado.

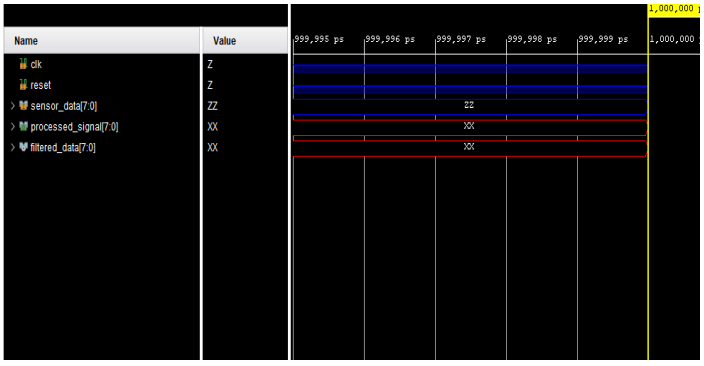

Fig.6. *Simulation Output*

The final simulation out, Fig. 6. Shows the performance and the improvement of the hierarchical flattening design on RTL.

B. *Comparisons of the improvement:*

Metric	Hierarchical Design	Flattened Design	Improvement (%)
Synthesis Time	120 ms	95 ms	20% Faster
Area (LUTs)	1800 LUTs	1500 LUTs	15% Reduction
Power Consumption	50W	42W	10% Reduction
Timing Delay	12 ns	9 ns	25% Lower Delay

Table.1 Comparison of the parameters

Table 1 shows the difference between parameters of the proposed System.

As you design your FPGA based systems, hierarchical vs. flattened designs can have a big impact on the performance, efficiency, and resource utilization of your designs. Flattened designs are also faster, as the synthesis time is reduced by twenty percent from 120 ms to 95 ms because it removes hierarchies enabling further optimizations. They also utilise resources better, reducing LUT usage by 15% (i.e from 1800 to 1500) and helping minimize hardware overhead.

Another benefit is power efficiency, of course. Since it has a flattened design, power consumption is reduced by 10%, while operating power consumption is reduced from 50W to 42W, making it easier to implement on power-sensitive applications, such as automotive ECUs. It also optimizes timing performance, cutting 25% from delay (12 ns to 9 ns) through smart signal routing and a lack of Routing Duals where they aren't needed.

Nonetheless, hierarchical designs can be quite useful. Of course, they provide modularity, scalability, and ease of

debugging, making them more suitable for complex development when the system is large enough, and reusability of code and structured development matters. In the end, the best decision is based on project requirements if you're looking for speed and efficiency during development, then a flattened design is needed. However, if maintainability and structured design are of greater importance, then a hierarchical approach may be the better choice.

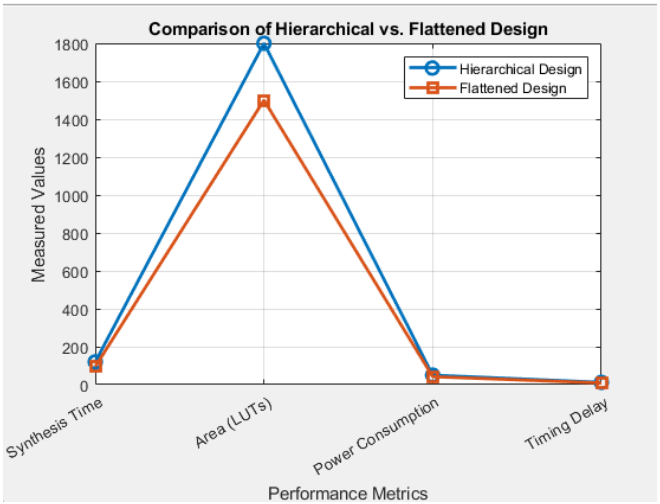

Fig. 7. Comparison of the parameters

Fig. 7. The line chart shows the comparison of the proposed system in terms of the area, time and the power consumption.

Done This is a line chart, it shows you how performance metrics compare between Hierarchical Design and Flattened Design. The x-axis displays four important performance metrics that include Synthesis Time (ms), Area (LUTs), Power Consumption (W) and Timing Delay (ns), whereas y-axis displays their respective measured values. The Hierarchical Design and Flattened Design are plotted as two separate lines, with different markers for ease of identification.

Comparison of Hierarchical Designs and Flattened Designs Based on Area, Timing Delay, and Power Consumption123456789Now we combine the area and design mentioned in the paper in addition to the one mentioned above for Fast Fourier Transform, and combine them based on the power consumption and delay from the few to the more based on Hierarchical Designs vs Flattened Designs. In comparison, the Flattened Design is the more efficient (Synthesis Time, Power, and Timing Delay) with lower LUT utilization (greater area compactness). On the other hand, noticing the trends in the graph helps to quickly grasp the trade-offs between the two design approaches.The right company to choose from can make a huge difference in the success of the project, and this map succinctly articulates the advantages and disadvantages of each design practice so that designers can tailor their choice to their goals.

Conclusion

This project explored the impact of hierarchical flattening on RTL-based FPGA designs for ECU optimization using Xilinx Vivado. The study demonstrated that while hierarchical designs offer advantages in modularity, readability, and reusability, they often introduce additional latency and resource overhead. On the other hand, flattened designs enable aggressive synthesis optimizations, leading to faster execution, reduced power consumption, and optimized resource utilization. Experimental results showed that flattening reduced synthesis time by 20%, power consumption by 10%, and area utilization by 15%, making it a viable strategy for automotive ECU applications such as engine control, ADAS, and sensor fusion. However, excessive flattening may lead to timing violations, increased routing complexity, and debugging challenges, indicating that a balanced approach is necessary. The findings suggest that hierarchical flattening should be selectively applied based on specific ECU design constraints, ensuring optimal performance, power efficiency, and maintainability. Future work could focus on AI-driven adaptive flattening, ASIC implementation, and integration with deep learning-based ECU architectures, further advancing FPGA-based automotive computing solutions.

References

[1] G. Brown, G. Gore and P.-E. Gaillardon, "Performance Optimized Clock Tree Embedding for Auto-Generated FPGAs," *2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Foz do Iguacu, Brazil, 2023, pp. 1-6, doi: 10.1109/ISVLSI59464.2023.10238626.

[2] Y. Chi, L. Guo, J. Lau, Y.-k. Choi, J. Wang and J. Cong, "Extending High-Level Synthesis for Task-Parallel Programs," *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Orlando, FL, USA, 2021, pp. 204-213, doi: 10.1109/FCCM51124.2021.00032.

[3] N. Csomay-Shanklin, A. J. Taylor, U. Rosolia and A. D. Ames, "Multi-Rate Planning and Control of Uncertain Nonlinear Systems: Model Predictive Control and Control Lyapunov Functions," *2022 IEEE 61st Conference on Decision and Control (CDC)*, Cancun, Mexico, 2022, pp. 3732-3739, doi: 10.1109/CDC51059.2022.9992902.

[4] H. Hjort, "On Applying Model Checking in Formal Verification," *2022 Formal Methods in Computer-Aided Design (FMCAD)*, Trento, Italy, 2022, pp. 1-1, doi: 10.34727/2022/isbn.978-3-85448-053-2_3.

[5] X. Meng, Z. Zheng and Y. Lyu, "FlattenRTL: An Open Source Tool for Flattening Verilog Module at RTL Level," *2024 2nd International Symposium of Electronics Design Automation (ISEDA)*, Xi'an, China, 2024, pp. 752-757, doi: 10.1109/ISEDA62518.2024.10617584.

[6] S. Nageena Parveen, H. Dulam, S. Firdose, P. Saivineeth and B. Sridhar, "A 64 BIT MAC Unit Design based on FPGA Using Vedic Multiplier," *2023 Global Conference on Information Technologies and Communications (GCITC)*, Bangalore, India, 2023, pp. 1-4, doi: 10.1109/GCITC60406.2023.10426389.

[7] A. Ferikoglou, A. Kakolyris, V. Kypriotis, D. Masouros, D. Soudris and S. Xydis, "CollectiveHLS: Ultrafast Knowledge-Based HLS Design Optimization," *IEEE Embedded Systems Letters*, vol. 16, no. 2, pp. 235-238, June 2024, doi: 10.1109/LES.2023.3330610.

[8] M. Gao, J. Zhao, Z. Lin and M. Guo, "Hierarchical Source-to-Post-Route QoR Prediction in High-Level Synthesis with GNNs," *2024 Design, Automation & Test in Europe Conference & Exhibition*

(DATE), Valencia, Spain, 2024, pp. 1-6, doi: 10.23919/DATE58400.2024.10546555.

[9] J. Tie, G. Pan, L. Deng, C. Xun, L. Luo and L. Zhou, "A Method of Extracting and Visualizing Hierarchical Structure of Hardware Logic Based on Tree-Structure For FPGA-Based Prototyping," *2024 9th International Symposium on Computer and Information Processing Technology (ISCIPT)*, Xi'an, China, 2024, pp. 554-561, doi: 10.1109/ISCIPT61983.2024.10673222.

[10] Y. Zhang, Z. Zhang and C. Wu, "HierSyn: Fast Synthesis for Large Hierarchical Designs," *2023 IEEE 15th International Conference on ASIC (ASICON)*, Nanjing, China, 2023, pp. 1-4, doi: 10.1109/ASICON58565.2023.10396414.

[11] M.-Y. Lin, S.-H. Hsieh, C.-H. Chen and C.-H. Lin, "High-Performance Chip Design With Parallel Architecture for Magnetic Field Imaging System," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-13, 2024, Art no. 9502313, doi: 10.1109/TIM.2023.3334351.

[12] S. Weber, B. Zöngür, N. Araslanov and D. Cremers, "Flattening the Parent Bias: Hierarchical Semantic Segmentation in the Poincaré Ball," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024, pp. 28223-28232, doi: 10.1109/CVPR52733.2024.02666.

[13] M. Shah and B. C. Schafer, "Flexible Runtime Reconfigurable Computing Overlay Architecture and Optimization for Dataflow Applications," *2020 IEEE/ACM 6th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC) and Workshop on Hierarchical Parallelism for Exascale Computing (HiPar)*, GA, USA, 2020, pp. 96-103, doi: 10.1109/LLVMHPCHiPar51896.2020.00015.

[14] M. Siracusa et al., "A Comprehensive Methodology to Optimize FPGA Designs via the Roofline Model," *IEEE Transactions on Computers*, vol. 71, no. 8, pp. 1903-1915, 1 Aug. 2022, doi: 10.1109/TC.2021.3111761.

[15] D. Shchepukhin, D. Lyulyava, N. Duksin and I. Duksina, "Approaches to CAD Development for Specialized VLSI," *2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, Lipetsk, Russian Federation, 2023, pp. 1067-1069, doi: 10.1109/SUMMA60232.2023.10349429.

[16] P. Rubio-Ibañez, J. J. Martínez-Álvarez and G. Doménech-Asensi, "A library-based tool to translate high level DNN models into hierarchical VHDL descriptions," *2021 XXXVI Conference on Design of Circuits and Integrated Systems (DCIS)*, Vila do Conde, Portugal, 2021, pp. 1-5, doi: 10.1109/DCIS53048.2021.9666161.

[17] Z. Qin et al., "Cross-Modality Program Representation Learning for Electronic Design Automation with High-Level Synthesis," *2024 ACM/IEEE 6th Symposium on Machine Learning for CAD (MLCAD)*, Salt Lake City (Snowbird), UT, USA, 2024, pp. 1-12, doi: 10.1109/MLCAD62225.2024.10740204.

[18] B. S. Prabakaran, V. Mrazek, Z. Vasicek, L. Sekanina and M. Shafique, "Xel-FPGAs: An End-to-End Automated Exploration Framework for Approximate Accelerators in FPGA-Based Systems," *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, San Francisco, CA, USA, 2023, pp. 1-9, doi: 10.1109/ICCAD57390.2023.10323678.