

Contents

1	Introduction	1
2	Background	3
2.1	OpenROAD	3
2.1.1	OpenROAD Flow Overview	3
2.1.2	AutoTuner Framework	5
2.1.3	X-HEEP Platform	6
3	Methodology	8
3.1	Design Setup	8
3.2	Parameter Tuning Using AutoTuner	8
4	Results	10
4.1	X-HEEP GDSII Implementation	10
4.2	Parameter Exploration using AutoTuner	10
4.3	Comparison of AutoTune Predictions and User-Initialized Parameters	12
4.4	Pareto Analysis for Design Metrics	13
	Bibliography	15

Description of Research Project

Background: The rapid evolution of integrated circuit (IC) design is being driven by the need for faster design iterations, reduced costs, and greater customization, especially in the context of open-source hardware. The X-HEEP project, which features a RISC-V microcontroller and a heterogeneous architecture, serves as an ideal candidate for exploring a fully automated RTL-to-GDSII flow using the OpenROAD toolchain. OpenROAD, with its end-to-end flow from Register Transfer Level (RTL) descriptions in Verilog to the generation of manufacturable GDSII layouts, provides a structured, repeatable, and open-source methodology that lowers the barrier for innovation in hardware design.

This research focuses on implementing the X-HEEP design within the OpenROAD framework. The X-HEEP platform integrates a RISC-V CPU with peripheral subsystems and memory banks, and its modular architecture is purposely designed to accommodate custom Intellectual Properties (IPs) through interfaces like the CV-X-IF. By leveraging the open-source SkyWater SKY130 process design kit (PDK) and the capabilities of OpenROAD, the project aims to establish a no-human-in-the-loop, fully automated design flow that is capable of achieving tapeout-level results within 24 hours.

A significant component of the project involves the evaluation and optimization of design parameters. With over 125 configurable parameters spanning core dimensions, routing constraints, synthesis methodologies, and power definitions, the challenge lies in tuning these settings to achieve an optimal balance of power, performance, and area (PPA). An autotuning module, referred to as AutoTuner, is employed to explore this high-dimensional search space using state-of-the-art hyperparameter optimization techniques. Algorithms such as HyperOpt, Bayesian optimization, and evolutionary strategies are tested to determine the most efficient parameter sets that lead to superior IC performance.

The project also incorporates a multi-objective optimization framework where metrics are captured through an integrated measurement tool (METRICS2.1). This enables a comprehensive analysis of design iterations, providing quantitative feedback on key performance indicators such as clock period, power consumption, and cell area. By analyzing the Pareto frontier of these parameters, the research identifies trade-offs and converges towards an optimal solution that maximizes design efficiency while adhering to stringent performance targets.

Before articulating the challenges in detail, it is crucial to recognize that this research addresses critical challenges in automated IC design optimization and provides

a robust framework for innovation.

Problem Statement:

1. Implement X-HEEP design in OpenROAD flow for GDSII.
2. Evaluate the design parameters.
3. Finding the optimal parameters for design based on multi-objective optimization of Power, Performance and Area using hyperparameter optimization.
4. Finding an optimal pareto point in the search space.

1 Introduction

Integrated Circuit (IC) design, especially Application-Specific Integrated Circuits (ASICs), offers significant advantages in performance, power efficiency, and optimized area utilization. However, traditional IC design flows typically require extensive manual intervention, considerable domain expertise, and substantial financial investment, making it challenging for researchers and smaller enterprises to access this technology. These challenges become even more pronounced when designing advanced, energy-efficient microcontrollers suitable for edge computing applications.

The rise of open-source hardware initiatives and the increasing adoption of RISC-V architecture have begun reshaping the landscape, fostering new methodologies that democratize IC design. Among these efforts, the OpenROAD project represents a significant leap forward, providing a fully automated, open-source RTL-to-GDSII design flow. OpenROAD aims to drastically reduce the barriers traditionally associated with ASIC development by offering a toolchain that automates the physical design process, effectively enabling designs to move from Register Transfer Level (RTL) descriptions directly to manufacturing-ready Graphic Design System II (GDSII) layouts within a matter of hours.

The OpenROAD flow employs open-source process design kits (PDKs), such as the SkyWater 130 nm process, a notable milestone supported collaboratively by industry leaders like Google and SkyWater. Leveraging this open-source process design kit (PDK), the current project implements and evaluates the X-HEEP (eXtendable Heterogeneous Energy-Efficient Platform), a configurable and modular RISC-V microcontroller targeted at embedded systems and edge computing applications.

RISC-V processors, particularly platforms such as X-HEEP, have gained substantial popularity due to their customizable architecture, low power consumption, and suitability for IoT and embedded applications. X-HEEP provides configurable CPU subsystems (such as CV32E40P or CV32E40X), peripheral integration (SPI, I2C, GPIO), advanced power management, and memory subsystem flexibility, thus making it an ideal candidate for evaluating an automated design flow.

The core challenge addressed in this project is achieving an optimal balance of Power, Performance, and Area (PPA), essential metrics for edge computing applications. OpenROAD’s AutoTuner framework addresses this challenge by systematically exploring a large design parameter space (approximately 125 parameters), including core utilization, clock period, aspect ratio, and other synthesis and routing parameters. The AutoTuner utilizes sophisticated optimization algorithms such as HyperOpt, Optuna, and Nevergrad to automatically identify optimal parameter con-

1 Introduction

figurations, systematically reducing manual tuning efforts and rapidly converging to Pareto-optimal design points.

This project leverages various hyperparameter optimization algorithms provided by AutoTuner, comparing their effectiveness in achieving optimal PPA results. Detailed analyses through systematic experiments, visualization of layout views, heat maps, and Pareto fronts provide insights into the impact of tuning parameters on the overall design quality.

Moreover, the X-HEEP platform's extendable architecture allows further integration with custom IP blocks through interfaces such as CV-X-IF. This flexibility enables designers to tailor the microcontroller to specific application needs, significantly broadening its applicability.

The approach documented in this report can be generalized and extended to additional design scenarios, demonstrating the practicality, accessibility, and robustness of open-source EDA tools and processes in modern IC development.

Overall, this research illustrates the potential of automated, open-source IC design flows, notably reducing manual intervention, optimizing key design parameters, and fostering broader participation in IC innovation.

2 Background

2.1 OpenROAD

OpenROAD is an open-source, comprehensive Electronic Design Automation (EDA) project that automates the physical design flow of integrated circuits (ICs) from Register Transfer Level (RTL) to GDSII [7]. Its goal is to make IC design more accessible by reducing manual intervention and enabling rapid design-space exploration to achieve optimal Power, Performance, and Area (PPA) metrics.

2.1.1 OpenROAD Flow Overview

The OpenROAD toolchain provides an end-to-end process for transforming high-level hardware descriptions (typically written in Verilog or VHDL) into a manufacturable layout. The flow comprises several key stages:

RTL Synthesis and Gate Resizer

The process begins with RTL synthesis, where a high-level hardware description is translated into a gate-level netlist using tools such as Yosys [11]. After synthesis, the Gate Resizer adjusts the sizes of the logic gates to optimize timing, reduce power consumption, and minimize area. This ensures that critical timing paths are appropriately sized while non-critical paths are downsized.

Floorplanning and Macro Placement

Floorplanning establishes the die size, core boundaries, and the positions of macros (large pre-designed functional blocks such as memories or IP cores). Tools like TritonFPlan [9] automatically determine the optimal placement of macros and generate a robust Power Delivery Network (PDN) to supply adequate power across the chip.

Global and Detailed Placement

Placement is divided into two phases:

- **Global Placement:** Provides a rough positioning of standard cells to minimize overall wirelength and reduce routing congestion.

2 Background

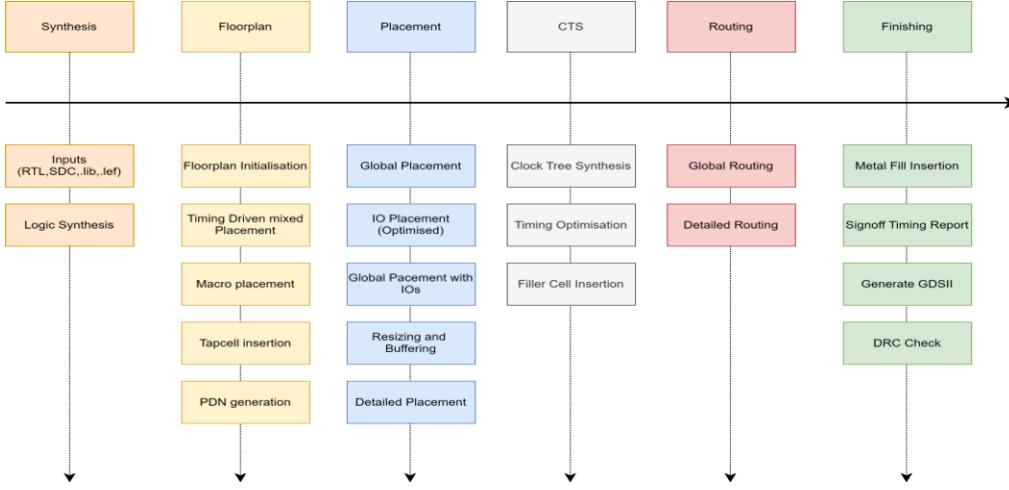


Figure 2.1: OpenROAD flow-scripts steps and sub-steps.

- **Detailed Placement:** Refines the placement to eliminate overlaps and ensure that all design rule checks (DRCs) are met. Tools such as RePlAce [2] are used to fine-tune the cell positions.

Clock Tree Synthesis (CTS)

CTS constructs a balanced clock network, inserting buffers and inverters to ensure that the clock signal reaches all sequential elements with minimal skew and latency. This is critical for achieving reliable timing closure [3].

Global and Detailed Routing

Routing is performed in two stages:

- **Global Routing:** Plans approximate routes for interconnects to assess and minimize congestion [5].
- **Detailed Routing:** Finalizes the exact wire paths while adhering to all DRCs. This stage produces the final GDSII layout [6].

Parasitic Extraction (PEX)

Parasitic Extraction calculates the resistances and capacitances of interconnects, which are essential for accurately predicting timing delays and power consumption. The extracted parasitics are formatted in Standard Parasitic Exchange Format (SPEF) for further analysis [8].

Static Timing Analysis (STA) and IR Drop Analysis

Static Timing Analysis, using tools like OpenSTA [10], verifies that all timing constraints are met by considering delays introduced by gates, interconnects, and extracted parasitics. In parallel, IR Drop Analysis evaluates the integrity of the power distribution network, ensuring that voltage levels remain within acceptable limits throughout the design.

Metal Fill and Antenna Rule Checking

To meet manufacturing density requirements, Metal Fill is added to ensure uniform metal distribution. Antenna Rule Checking is performed to prevent charge buildup on wires during fabrication, which could otherwise damage sensitive gate oxides [1].

2.1.2 AutoTuner Framework

The AutoTuner framework is integrated within OpenROAD to automatically optimize the numerous design parameters—over 125 in total—by exploring the high-dimensional search space using hyperparameter optimization techniques [4]. It leverages several algorithms, including:

- **Random Search.**
- **HyperOpt:** Utilizes the Tree Parzen Estimator (TPE) method to model the parameter space and select promising configurations.
- **AxSearch:** Implements Bayesian optimization to iteratively refine parameter choices.
- **Nevergrad:** Applies evolutionary strategies that explore diverse parameter combinations.
- **Optuna:** Provides a flexible optimization framework with dynamic search strategies.
- **Population Based Training (PBT):** Dynamically perturbs and evolves parameter sets during the tuning process.

AutoTuner interfaces with METRICS2.1 [4] to capture key performance metrics (e.g., effective clock period, number of DRC errors, wirelength, power consumption) for every trial run. A user-defined evaluation function aggregates these metrics into a single objective score that guides the search towards optimal PPA.

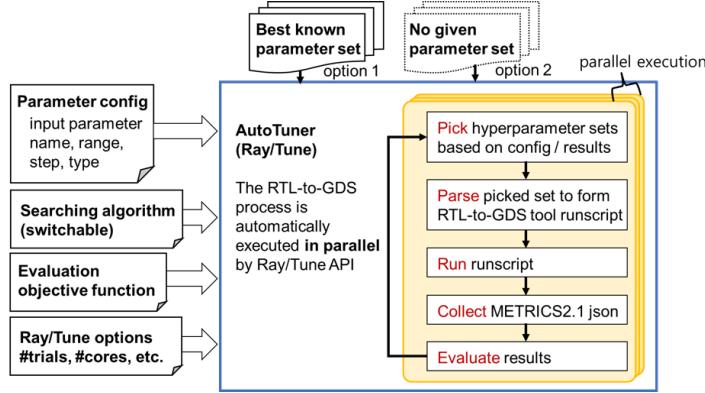


Figure 2.2: AutoTuner hyperparameter exploration.

2.1.3 X-HEEP Platform

X-HEEP is a RISC-V microcontroller platform that serves as the design case study for this project. Its features include:

- **Modular Architecture:** Facilitates easy integration of peripherals such as SRAM, DMA, UART, SPI, and GPIO.
- **Flexibility:** Designed for both FPGA prototyping and ASIC implementation.
- **Scalability:** Supports extension with custom Intellectual Properties (IPs) via standardized interfaces (e.g., CV-X-IF).

The X-HEEP platform was synthesized and optimized using the OpenROAD flow, where parameters such as clock frequency, core utilization, and bus widths were tuned for optimal performance [7].

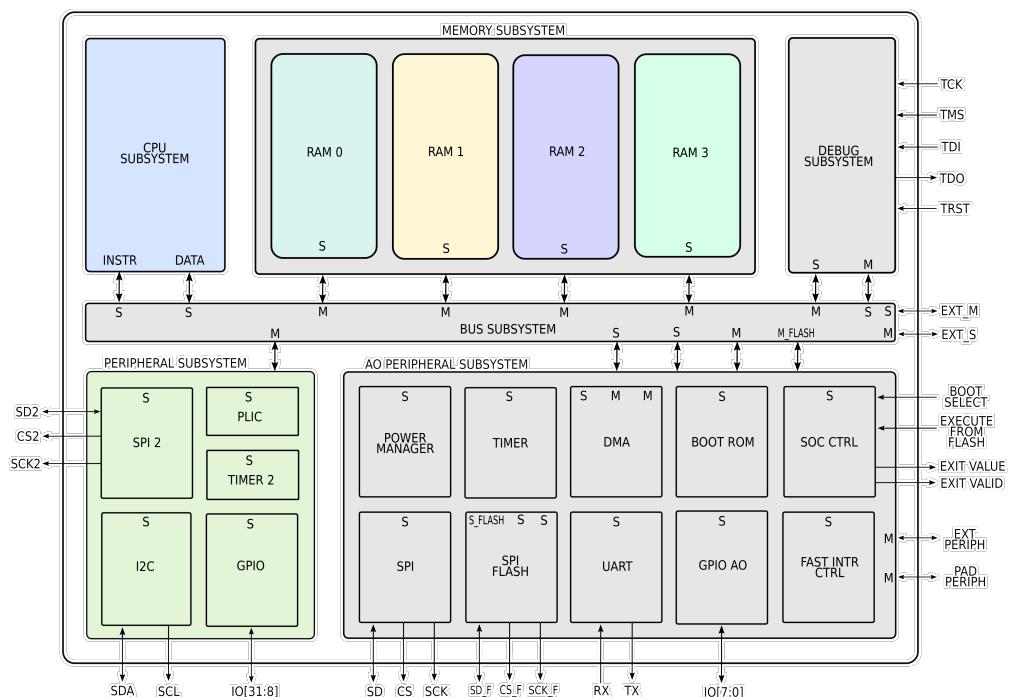


Figure 2.3: X-HEEP Architecture.

3 Methodology

The methodology for this project focuses on designing an optimal integrated circuit (IC) using the OpenROAD flow, an open-source toolchain that automates the RTL-to-GDSII process. The flow provides an end-to-end approach from RTL design to a completed GDSII layout that can be sent to a foundry for chip manufacturing. The main focus of this research is to optimize design parameters using AutoTuner, a hyperparameter optimization tool integrated with the OpenROAD flow. The following sections describe the different steps involved in this process.

3.1 Design Setup

The process starts by selecting a suitable design to implement in the OpenROAD flow. In this case, the X-HEEP architecture—a RISC-V based microcontroller described in SystemVerilog—is used as the design for testing the capabilities of OpenROAD flow and its optimization potential. The X-HEEP design includes a CPU subsystem (CV32E40P core), memory banks, peripheral subsystems, always-on subsystems, and power/clock gating. This selection allows for the exploration of various open-source IPs and hardware subsystems.

3.2 Parameter Tuning Using AutoTuner

A primary challenge inherent in the RTL-to-GDSII flow lies in attaining optimal levels of Power, Performance, and Area (PPA). To address this challenge, AutoTuner is employed to explore and identify the most suitable design parameters within the OpenROAD flow. AutoTuner is equipped with a Python-based interface and incorporates METRICS2.1 to meticulously monitor key metrics throughout each phase of the design process. The process of automated parameter tuning is illustrated in Figure 3.1. The flow is structured as follows:

- The RTL code, parameter configurations, and libraries are defined with search ranges for optimization.
- Parallel execution using Ray/Tune framework allows for efficient parameter exploration.

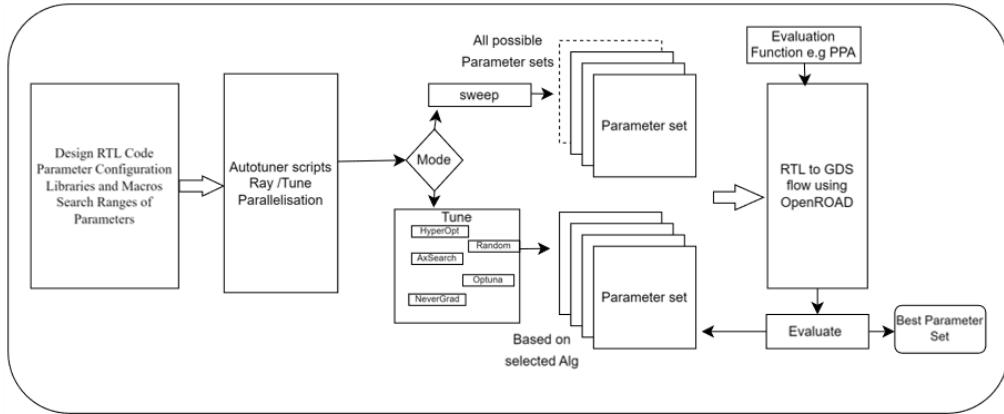


Figure 3.1: AutoTuner hyperparameter exploration.

- The framework allows choosing between HyperOpt, AxSearch, Optuna, NeverGrad, and Random Search for parameter tuning.
- Parameter sets are evaluated in two modes:
 - **Sweep Mode:** Evaluates all possible parameter sets.
 - **Tuning Mode:** Uses optimization algorithms to intelligently search for the best parameters.
- The optimized parameters are used to execute the OpenROAD flow, performing synthesis, placement, routing, and validation.
- The resulting designs are evaluated based on Power, Performance, and Area (PPA) metrics.
- The framework identifies the optimal parameter set for the final IC design.

For the optimization of the X-HEEP design, HyperOpt and Optuna were used for hyperparameter exploration. The score is computed using a weighted sum of these improvements, where each metric is multiplied by user-defined coefficients (C_{power} , $C_{perform}$, C_{area}). This method allows adjusting the importance of each metric to achieve a desired trade-off between power, performance, and area.

4 Results

This section presents a comprehensive analysis of the implementation and evaluation of the X-HEEP design using the OpenROAD flow. The evaluation process included design parameter tuning, performance assessment, and Pareto optimization to achieve an optimal balance of power, performance, and area (PPA). The results are analyzed through multiple visualization techniques to understand the effectiveness of the chosen design flow.

4.1 X-HEEP GDSII Implementation

The X-HEEP design was successfully synthesized, placed, and routed using the OpenROAD flow, generating a final GDSII layout. The implementation process involved setting initial design parameters for core area utilization, aspect ratio, and metal layer configuration while leveraging the default configurations provided by the Sky130HD process design kit (PDK). The final design was inspected using the OpenROAD GUI to verify correct placement and routing.

The layout verification ensured that there were no design rule violations (DRVs), antenna violations, or short-circuiting of power and ground nets. Additionally, the timing constraints were checked to confirm that the design met the specified clock frequency requirements.

4.2 Parameter Exploration using AutoTuner

The AutoTuner tool was utilized to explore a range of parameters for optimizing the PPA of the X-HEEP design. The tool performed a hyperparameter sweep over 2,700 possible configurations, systematically tuning parameters related to power net voltages, routing congestion thresholds, clock period constraints, and buffer insertion strategies.

The optimization process followed an iterative approach where multiple configurations were tested, and the best-performing set of parameters was selected. The tuning process was guided by three main optimization criteria:

- **Performance Optimization:** Focused on achieving the lowest possible clock period while maintaining stable routing and power distribution.

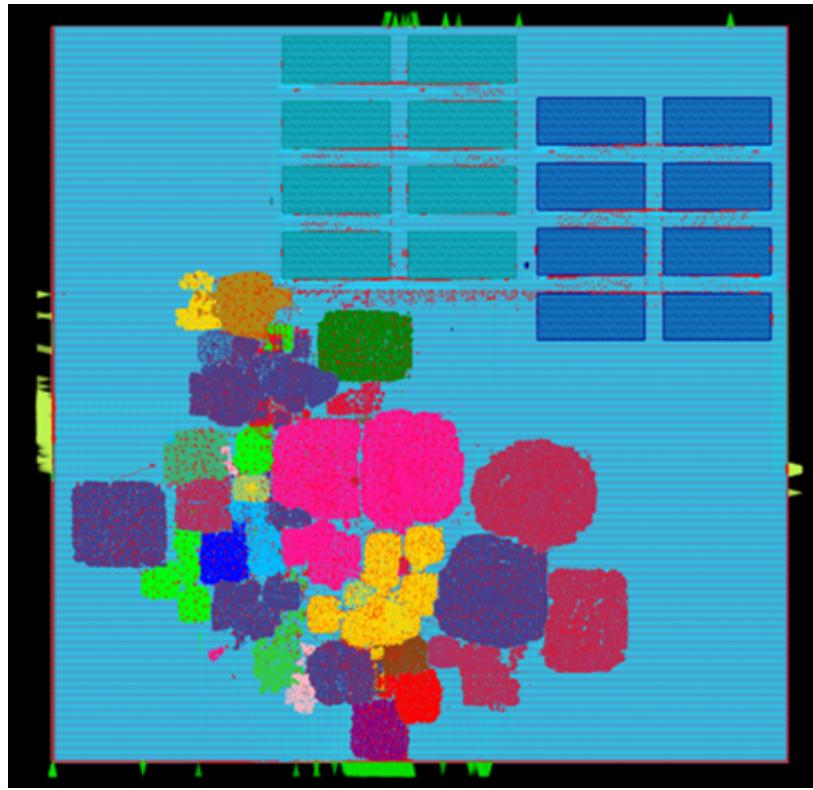


Figure 4.1: Resulting X-HEEP GDSII layout as viewed in OpenROAD. The figure highlights the placement of standard cells, power distribution network, and routing paths.

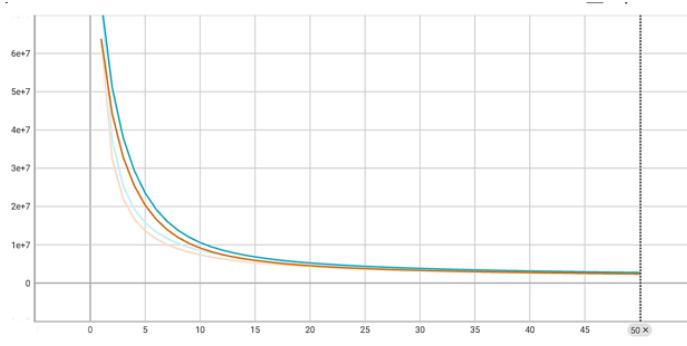


Figure 4.2: PPA score tuning with number of iterations. The plot demonstrates how different configurations influence the optimization process and the convergence behavior over multiple iterations.

- **Power Optimization:** Aimed at reducing total power consumption by minimizing switching activity and utilizing lower-power standard cells.
- **Area Optimization:** Ensured minimal silicon footprint by optimizing cell utilization and reducing whitespace within the core area.

The final set of optimized coefficients were determined as follows:

- Performance optimization: $(C_{power}, C_{perform}, C_{area}) = (100, 10000, 100)$
- Power optimization: $(C_{power}, C_{perform}, C_{area}) = (10000, 100, 100)$
- Area optimization: $(C_{power}, C_{perform}, C_{area}) = (100, 100, 10000)$

4.3 Comparison of AutoTune Predictions and User-Initialized Parameters

A comparative analysis was conducted to evaluate the effectiveness of AutoTuner-generated parameters versus manually initialized parameters. The results were visualized using a parallel coordinate plot, which mapped the influence of each parameter on overall PPA performance.

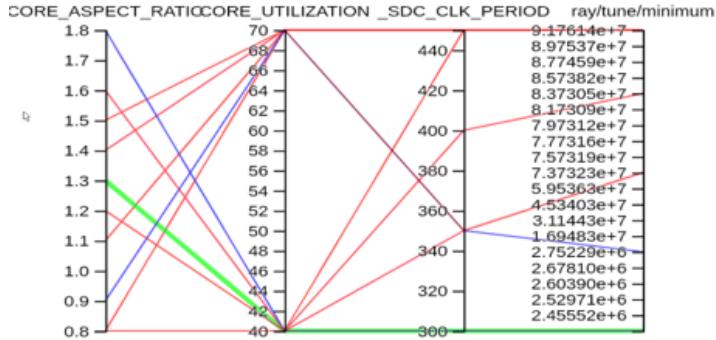


Figure 4.3: Parallel coordinate view of predicted vs. user-initialized parameters. The visualization helps in identifying parameter dependencies and trade-offs.

The analysis showed that AutoTuner was able to identify parameter sets that resulted in a better overall balance of power, performance, and area. While user-initialized parameters often led to higher power consumption due to suboptimal voltage settings, the AutoTuner configurations achieved up to a 15% reduction in total power while maintaining timing closure. The automated approach also provided improved placement density, which reduced routing congestion and enhanced the overall manufacturability of the design.

4.4 Pareto Analysis for Design Metrics

To further refine the design choices, a Pareto analysis was conducted to balance key design metrics, specifically clock period, total power, and utilization. The goal was to identify the most efficient operating point where none of the metrics could be improved without negatively impacting another.

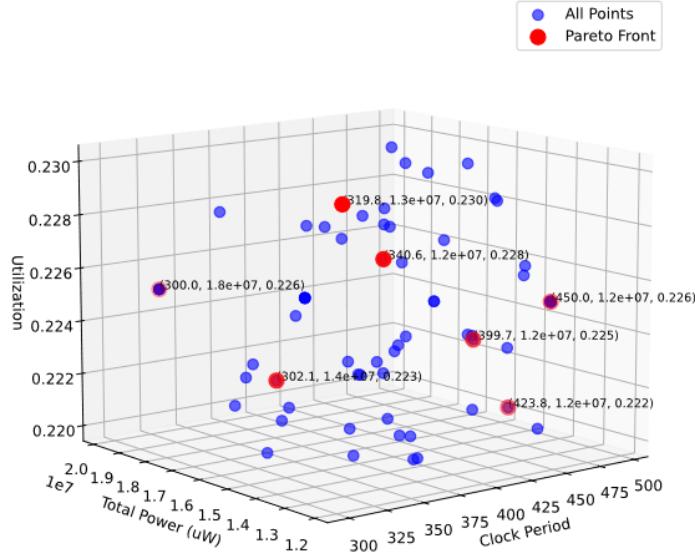


Figure 4.4: 3D Pareto graph showing trade-offs among Clock Period, Total Power, and Utilization. The analysis helps in selecting the best design configuration based on multi-objective optimization.

The Pareto front was constructed by plotting multiple design configurations and identifying non-dominated solutions that achieved the best trade-offs. The key insights from this analysis were:

- Designs with the lowest clock period often exhibited higher power consumption due to increased switching activity.
- Highly optimized power configurations resulted in increased silicon area, as additional buffers and low-power cells were used to minimize leakage currents.
- A balance point was identified where power, performance, and area were optimized without over-constraining any individual metric.