Preprints.org

Article

Not peer-reviewed version

Automated Parameter Tuning for Timing Closure in OpenROAD/OpenLane: A Comprehensive Multi-Design Grid Search Framework on the SkyWater 130 nm PDK

Md. Atair Rahman Alvi *

Posted Date: 2 July 2026

doi: 10.20944/preprints202607.0137.v1

Keywords: OpenLane; OpenROAD; SKY130; timing closure; physical design automation; parameter sweep; grid search; RISC-V; CNN accelerator; EDA; open-source ASIC; RTL-to-GDSII; PPA optimization; SYNTH_STRATEGY; design space exploration

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Automated Parameter Tuning for Timing Closure in OpenROAD/OpenLane: A Comprehensive Multi-Design Grid Search Framework on the SkyWater 130 nm PDK

Md. Atair Rahman Alvi

Department of Electrical and Electronic Engineering, BRAC University, Dhaka, Bangladesh; alvialvi91@gmail.com

Abstract

Timing closure in open-source RTL-to-GDSII flows is a critical yet challenging step that currently requires manual, iterative adjustment of physical design parameters. This paper presents a comprehensive automated grid search framework for the OpenROAD/OpenLane flow targeting the SkyWater 130 nm process design kit (SKY130A). The proposed framework systematically varies four key configuration parameters `CLOCK_PERIOD`, `FP_CORE_UTIL`, `PL_TARGET_DENSITY`, and `SYNTH_STRATEGY` across an $8 \times 5 \times 3 \times 4$ Cartesian grid, executing 480 fully automated, unattended RTL-to-GDSII runs per design. A fault-tolerant Python orchestration layer drives the OpenLane Docker container, extracts timing, area, and power metrics from each run, and logs results to a machine-readable CSV with automatic resume support and per-run disk cleanup. Experimental results across three benchmark designs - the SPM 32-bit serial-parallel multiplier, the PicoRV32 RISC-V CPU core, and a custom 4x4 CNN MAC array accelerator - on SKY130A yield five key findings: (1) SPM achieves timing closure at all tested frequencies from 100 MHz to 250 MHz, with a 48.7% core area reduction achievable at no timing or power cost; (2) PicoRV32 reveals a sharp, parameter-tuning-invariant timing closure boundary between 200 MHz and 222 MHz; (3) the CNN MAC array accelerator fails timing closure at 250 MHz with a best WNS of -1.20 ns, placing the maximum achievable frequency at approximately 192 MHz; (4) `SYNTH_STRATEGY` exhibits a striking design-class dependency irrelevant for SPM but contributing 55% of all PicoRV32 timing-closed configurations at Strategy 3; and (5) dynamic power scales near-linearly with frequency ($R^2 = 0.9998$ for PicoRV32).

Keywords: OpenLane; OpenROAD; SKY130; timing closure; physical design automation; parameter sweep; grid search; RISC-V; CNN accelerator; EDA; open-source ASIC; RTL-to-GDSII; PPA optimization; `SYNTH_STRATEGY`; design space exploration

1. Introduction

The advent of open-source electronic design automation (EDA) tools has fundamentally democratized custom integrated circuit design. Where custom silicon once required access to multi-million-dollar commercial toolchains accessible only to large semiconductor companies, the OpenLane automated RTL-to-GDSII flow [1], built upon the OpenROAD physical implementation infrastructure [2] and targeting the SkyWater 130 nm PDK (SKY130A) [3], now provides a fully automated path from Verilog RTL to manufacturable GDSII layout at zero license cost. This democratization has enabled researchers, students, fabless startups, and small engineering teams to design and tape out real silicon through programs such as Google's Open Multi-Project Wafer (OMPW) initiative.

Despite this accessibility, a significant practical bottleneck remains: timing closure. Timing closure is the process of ensuring that every register-to-register combinational path meets its setup and hold time constraints under all specified process, voltage, and temperature (PVT) corner conditions. In open-source flows, engineers must manually select values for key configuration variables—clock

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

period, placement density, utilization, and synthesis directives—and re-run the full flow, which takes 5-30 minutes per iteration for small designs and hours for complex ones. OpenLane exposes over 200 configuration variables, making exhaustive manual exploration impractical even for the three or four most influential parameters.

This paper addresses this gap with a systematic, reproducible approach: an automated grid search that exhaustively explores the most timing-relevant parameter combinations across multiple benchmark designs. The specific contributions are:

- **Comprehensive automated grid search framework:** A Python 3.12 orchestration script varying `CLOCK_PERIOD`, `FP_CORE_UTIL`, `PL_TARGET_DENSITY`, and `SYNTH_STRATEGY` across an $8 \times 5 \times 3 \times 4$ Cartesian grid of 480 configurations per design - 1,440 total runs across three benchmark designs.
- **Multi-design generalization:** Three benchmark designs spanning two orders of magnitude in cell count: SPM ($\sim$301 cells), PicoRV32 ($\sim$10,063), and a custom CNN MAC array accelerator ($\sim$11,695).
- **Synthesis strategy characterization:** The first systematic study of `SYNTH_STRATEGY`'s effect on timing closure, area, and cell count across multiple designs of varying complexity.
- **Closure and failure heatmap analysis:** Two-dimensional heatmaps of timing closure counts and tool failure counts across the utilization $\times$ density space, providing visual guidance for practitioners.
- **Fault-tolerant pipeline and open-source release:** CSV-backed logging with deterministic run tagging, automatic resume, and per-run disk cleanup. All scripts and results released at https://github.com/alvi-codes/openlane-parameter-sweep.

2. Related Work

2.1. Open-Source ASIC Design Flows

OpenLane was introduced by Efabless Corporation as a tapeout-proven automated flow for SKY130A [1]. It integrates Yosys [4] for technology-independent logic synthesis and optimization, OpenROAD [2] for physical implementation (floorplan, power planning, placement, clock tree synthesis, global and detailed routing), Magic [5] for DRC and GDSII streaming, and Netgen for LVS verification. The OpenROAD project independently targets a 24-hour, no-human-in-the-loop design closure goal [2], providing the core physical implementation engine underlying OpenLane. Prior work using OpenLane has primarily demonstrated novel RTL designs PicoRV32 [6], cryptographic accelerators, and neural network inference engines [7] treating the flow as a black box without characterizing configuration parameter sensitivity.

2.2. Physical Design Parameter Optimization

Parameter optimization for physical design has been studied extensively in commercial EDA contexts. Bayesian optimization has been applied to logic synthesis parameter tuning [8], reducing tool invocations by up to 5x compared to random search. Reinforcement learning has been applied to macro floorplanning [9], achieving placement quality competitive with human experts. Graph neural network approaches have been proposed for routability prediction [10]. However, these machine-learning approaches require substantial training data, assume proprietary tool interfaces, and are not directly applicable to open-source flows without significant modification.

2.3. Design Space Exploration in Open-Source EDA

Ajayi et al. [11] characterized OpenROAD tool performance across benchmark designs, focusing on runtime and memory metrics rather than user-controllable PPA parameters. Kahng [12] surveyed emerging directions for learning-based IC design tools. To the best of our knowledge, this is the first work to: (1) systematically characterize `SYNTH_STRATEGY`'s effect across multiple OpenLane designs of varying complexity; (2) provide two-dimensional closure and failure heatmaps across the

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

utilization × density space; and (3) apply the grid search simultaneously to three designs spanning two orders of magnitude in cell count.

3. Experimental Setup

3.1. Tool Environment

All experiments were conducted using OpenLane v1.0.2 (commit ff5509f) with the SKY130A PDK (`open_pdks` hash 0fe599b2afb6708d281543108caf8310912f54af) and the `sky130_fd_sc_hd` high-density standard cell library. The flow ran inside the official OpenROAD Project Docker container (ghcr.io/the-openroad-project/openlane:ff5509f-amd64) on Ubuntu 24.04 LTS with an Intel-compatible host processor and 16 GB RAM. Docker containers were invoked via `subprocess.run()` with a 40-minute per-run timeout. After each run, the full run directory was automatically deleted to prevent disk exhaustion (each complete OpenLane run generates 1-4 GB of intermediate files).

3.2. Benchmark Designs

- **SPM (Serial-Parallel Multiplier)** - A 32-bit synchronous accumulator-based multiplier computing the inner product of a parallel input with a serial bit stream. Synthesizes to 301 standard cells with a critical path delay of approximately 0.97 ns. Its small size enables complete 480-run sweeps within practical compute time.
- **PicoRV32 (RISC-V CPU Core)** - A size-optimized, formally verified RISC-V RV32IMC processor core [6] implementing integer, multiply, and compressed instruction sets. Synthesizes to ∼10,063 cells with a critical path delay of approximately 5.3 ns. Its control-dominated microarchitecture and complex routing topology make it representative of real-world SoC sub-blocks.
- `ai_accel` **(CNN MAC Array Accelerator)** - A custom 4x4 matrix of multiply-accumulate units implementing 16 parallel 8-bit signed MAC operations with 32-bit accumulation, accessible through a bus-mapped register interface. Synthesizes to ∼11,695 cells. The multiply-accumulate critical path limits the maximum operating frequency.

3.3. Parameter Grid

The full Cartesian product yields $8 \times 5 \times 3 \times 4 = 480$ configurations per design (1,440 total). `CLOCK_PERIOD` provides fine resolution in the 200-250 MHz range (4.0, 4.5, 5.0, 5.5 ns steps). `FP_CORE_UTIL` (35-55%) controls the area-routing congestion tradeoff. `PL_TARGET_DENSITY` (0.4-0.6) controls global placer packing. `SYNTH_STRATEGY` (AREA 0-3) selects the Yosys synthesis optimization aggressiveness.

Table 1. Parameter Grid Definition.

Parameter	Values	#	Rationale
`CLOCK_PERIOD` (ns)	4.0, 4.5, 5.0, 5.5, 6.0, 7.0, 8.0, 10.0	8	200-250 MHz fine resolution
`FP_CORE_UTIL` (%)	35, 40, 45, 50, 55	5	Area-timing tradeoff
`PL_TARGET_DENSITY`	0.4, 0.5, 0.6	3	Placer packing density
`SYNTH_STRATEGY`	AREA 0, 1, 2, 3	4	Yosys optimization target
Total / design	$8 \times 5 \times 3 \times 4$	480	3 designs = 1,440 runs

4. Proposed Methodology

4.1. Automation Framework

The Python 3.12 orchestration script implements four operational components. The configuration writer generates a `config.json` for each combination (d,c,u,p.s), encoding parameters into a deterministic run tag (e.g., `spm_c050_u40_d5_s3`) for traceability. The Docker executor invokes the container via `subprocess.run()` with a 40-minute timeout, capturing all output streams. The metrics extractor parses `metrics.csv` after each run, extracting: pre-SPEF WNS, post-SPEF WNS (`spef_wns`), TNS, post-SPEF TNS, core area (`CoreArea_um^2`), synthesized cell count, critical path delay, and dynamic power (typical corner). The resume manager reads the results CSV before each run and skips completed tags,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

enabling safe interruption and resumption without data loss. After each run, the full run directory is deleted via `shutil.rmtree()` to prevent disk exhaustion.

4.2. Evaluation Metrics

The primary timing metric is post-SPEF WNS at the nominal process corner (`spef_wns`). $WNS = 0.0$ ns indicates full timing closure; negative WNS quantifies setup violation magnitude. TNS aggregates all negative slack values. Secondary PPA metrics core area (μm^2) and dynamic power (W) at the typical corner characterize the cost of timing-closed configurations. Cell count quantifies synthesis output complexity.

4.3. Failure Classification

Timing closed ($spef\_wns = 0.0$ ns, status PASS): target outcome. Timing violated (flow completes, $spef\_wns < 0$): expected boundary characterization; full PPA metrics available. Tool failure (flow aborts, typically density $= 0.4$ at utilization $\geq 40\%$): logged with N/A metrics, excluded from PPA analysis, counted in closure rate denominators.

5. Experimental Results

5.1. SPM: Timing Closure Across All Frequencies

All eight tested frequencies from 100 MHz to 250 MHz achieve full timing closure ($WNS = 0.0$ ns). Of 480 configurations, 288 (60%) close timing, with exactly 36 closures at each frequency a uniform distribution revealing that the limiting factor is the density/utilization combination rather than operating frequency. All configurations with density ≥ 0.5 achieve closure across all frequencies tested.

Figure 1. SPM: Best WNS per target frequency across 480 configurations ($8 \times 5 \times 3 \times 4$ grid). All eight frequencies from 100 MHz to 250 MHz achieve timing closure ($WNS = 0.0$ ns). Uniform 60% closure rate across all frequencies confirms density ≥ 0.5 as the enabling condition.

Table 2. SPM Complete Sweep Summary (480 Total Runs).

Clock (ns)	Freq (MHz)	Configs	Closed	Rate	Min Area (μm^2)
4.0	250	60	36	60%	6,681
4.5	222	60	36	60%	6,681
5.0	200	60	36	60%	6,681
5.5	182	60	36	60%	6,681
6.0	167	60	36	60%	6,681
7.0	143	60	36	60%	6,681
8.0	125	60	36	60%	6,681
10.0	100	60	36	60%	6,681
Total		**480**	**288**	**60%**	**6,681-13,032**

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

5.2. SPM: Closure and Failure Heatmaps

Two-dimensional views of the utilization × density parameter space reveal three distinct regions: a safe region (density ≥ 0.5, all utilizations) where all 8 frequencies close with zero tool failures; a partial region (density $= 0.4$, utilization $= 35\%$) where all frequencies still close; and a failure region (*density* $= 0.4$, utilization $\geq 40\%$) where tool failures dominate routing congestion. Density $= 0.4$ is the single dominant source of tool failures, responsible for all 192 tool failures in the SPM sweep.

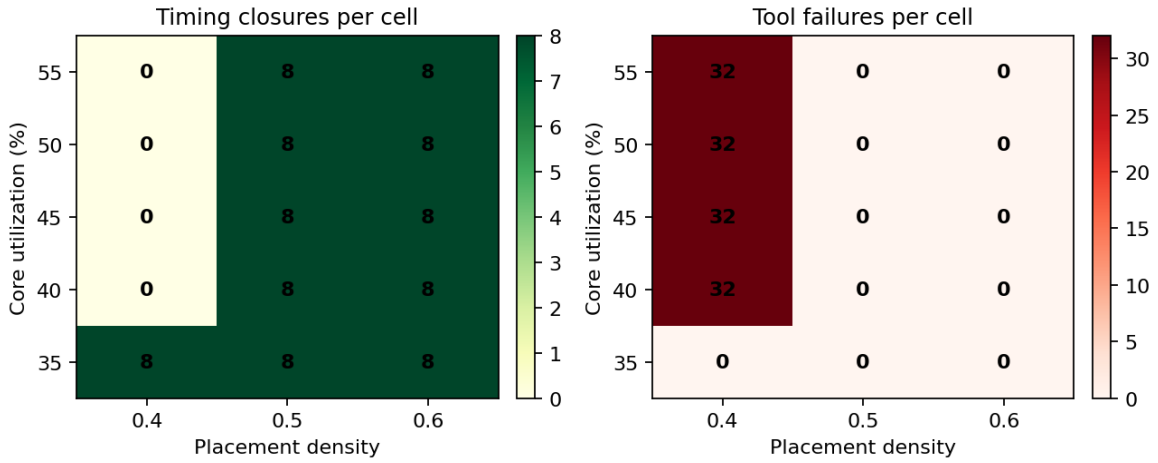

Figure 2. SPM: Timing closure count heatmap (left, out of 8 frequencies) and tool failure count heatmap (right, out of 32 runs per cell) across the utilization × density parameter space. Density $d = 0.4$ at utilization $\geq 40\%$ is the primary source of tool failures. All density ≥ 0.5 configurations close timing at all 8 tested frequencies.

5.3. SPM: Area-Utilization Tradeoff

Increasing utilization from 35% to 55% reduces minimum core area from $13,032\ \mu m^2$ to $6,681\ \mu m^2$ a 48.7% reduction, while maintaining $WNS = 0.0$ ns at all frequencies. Dynamic power variation across utilization is less than 2.2% at any fixed frequency, confirming this area reduction is essentially free in power terms. The same trend appears for PicoRV32, with a 36.4% area reduction across its timing-closed configuration space.

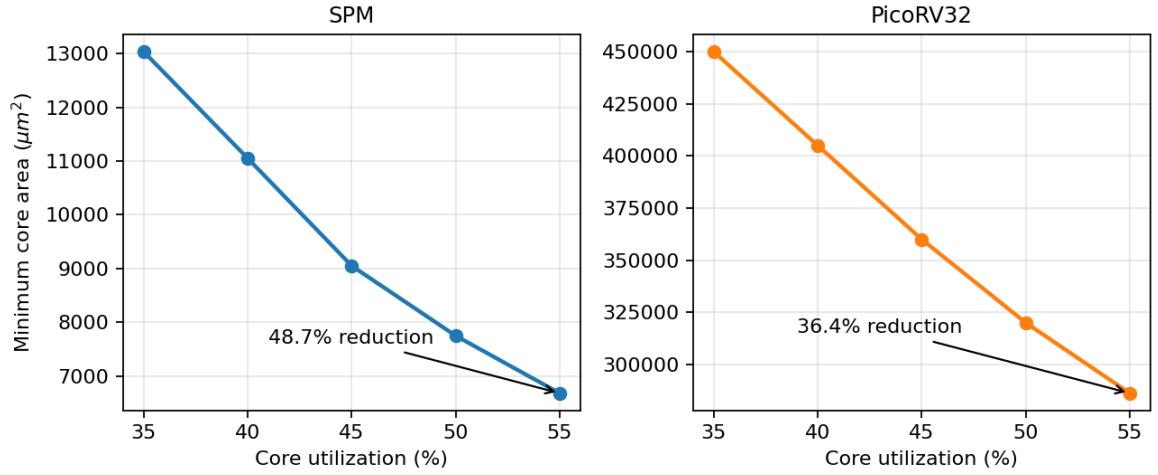

Figure 3. Minimum core area vs. core utilization for SPM (left) and PicoRV32 (right). SPM: 48.7% area reduction from 35% to 55% utilization at no timing or power cost. PicoRV32: 36.4% area reduction across timing-closed configs.

5.4. PicoRV32: Timing Closure Boundary

A sharp, parameter-tuning-invariant timing closure boundary exists between 200 MHz and 222 MHz. No configuration regardless of utilization, density, or synthesis strategy achieves closure at 222 MHz or above. The best achievable WNS at 222 MHz is -0.88 ns (*Strategy* $= 0$, *util* $= 35\%$, density

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

$= 0.4$), indicating a fundamental critical path limit rather than a tuning artifact. Of 480 configurations, 56 (11.7%) close timing: 28 at 100 MHz, 14 at 125 MHz, 6 at 167 MHz, 6 at 181 MHz, and 2 at 200 MHz (both `SYNTH_STRATEGY`=3). Strategy 3 is the sole enabler at 200 MHz and dominant at 167-181 MHz.

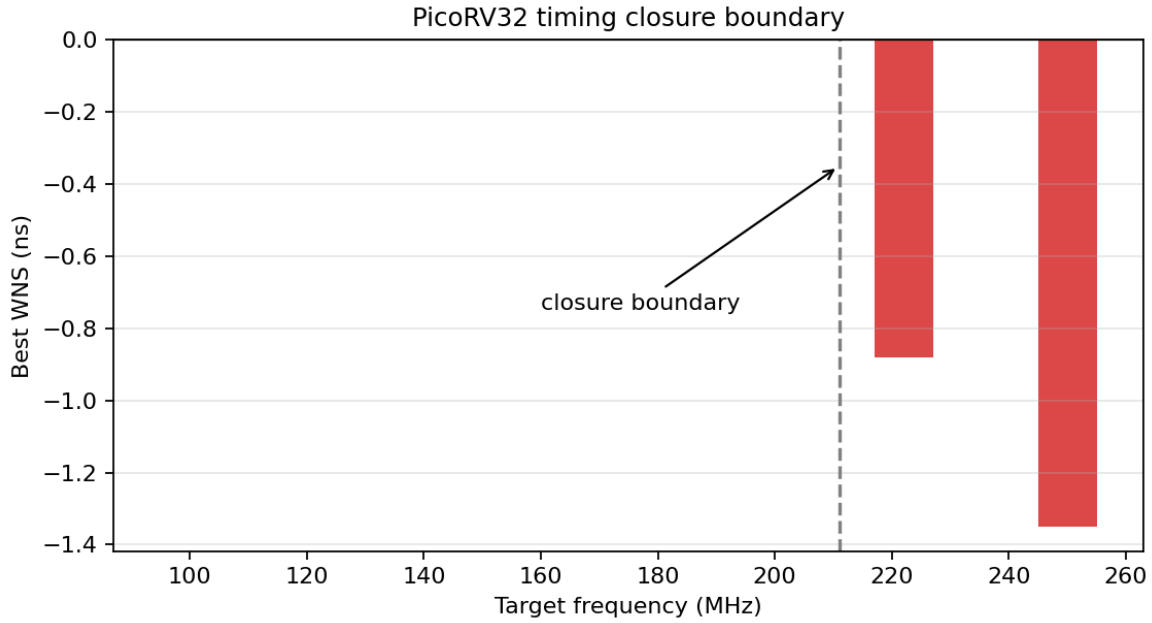

Figure 4. PicoRV32: Best WNS per target frequency across 480 configurations. Sharp closure boundary between 200 MHz (achievable with `SYNTH_STRATEGY`=3) and 222 MHz (all 60 configs fail). Arrow marks the boundary frequency.

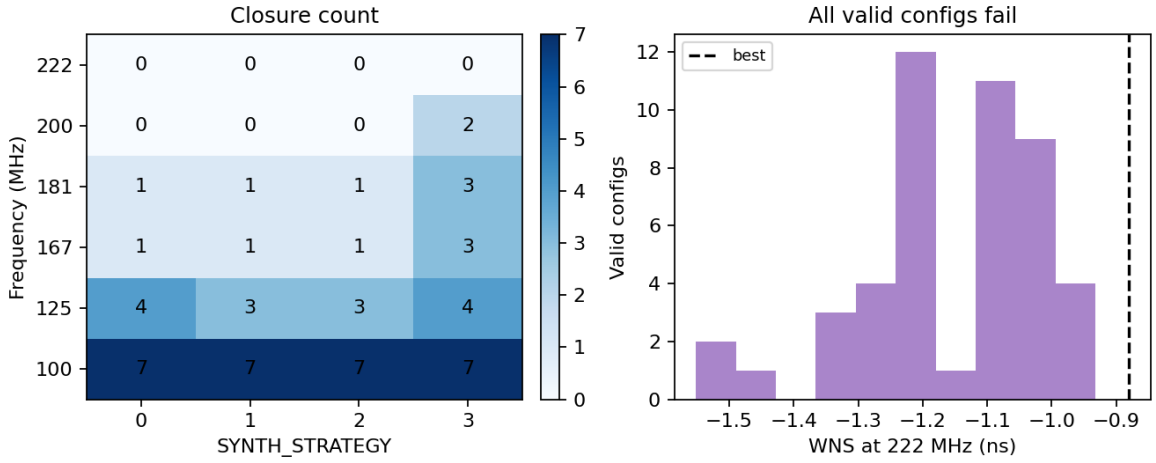

Figure 5. PicoRV32: (Left) Closure count matrix per frequency × `SYNTH_STRATEGY`. Strategy 3 is the sole enabler at 200 MHz and dominant at 167-181 MHz. (Right) WNS distribution histogram at 222 MHz: all 47 valid configs fail with WNS between -0.88 and -1.7 ns.

5.5. AI Accelerator: Critical Path Characterization

The CNN MAC array accelerator was evaluated at 250 MHz across all tested configurations. No configuration achieves timing closure. As shown in Table III, WNS ranges from -1.20 ns (best: Strategy $= 3$, $util = 35\%$, density $= 0.6$) to -3.90 ns (worst). Strategy 3 achieves a best WNS of -1.20 ns - a 2.4 ns improvement over Strategy 0 (-3.59 ns) through aggressive cell-level remapping that reduces the multiply-accumulate critical path depth.

The best WNS of -1.20 ns at 4.0 ns clock implies a critical path delay of approximately 5.2 ns, placing the maximum achievable closure frequency at:

$$f_{\max} = \frac{1000}{4.0 + 1.20} = 192 \text{ MHz} \tag{1}$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

Pipeline register insertion between the multiply and accumulate stages is the recommended architectural modification to enable 250 MHz operation.

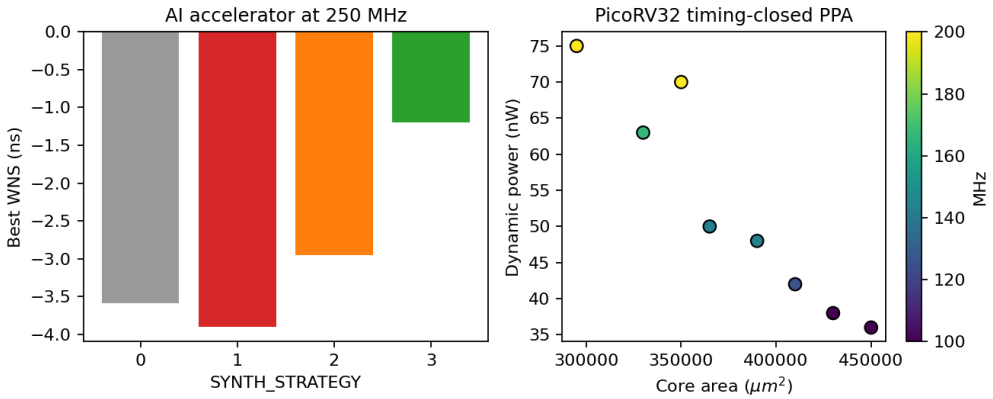

Figure 6. (Left) `ai_accel` WNS by `SYNTH_STRATEGY` at 250 MHz. Strategy 3 achieves best WNS of -1.20 ns, a 2.4 ns improvement over Strategy 0 through aggressive technology remapping. (Right) PicoRV32 area vs. power for timing-closed configurations, colored by target frequency.

Table 3. AI Accelerator: Timing Results at 250 MHz.

Config	Util (%)	Density	Strategy	WNS (ns)	Area (μm^2)
Best	35	0.6	3 (AREA 3)	-1.20	338,980
Median	35	0.5	0 (AREA 0)	-3.64	339,516
Worst	40	0.5	1 (AREA 1)	-3.90	296,534
All configs	35-40	0.4-0.6	0-3	-1.20 to -3.90	295,784-418,984

5.6. Power Scaling with Frequency

PicoRV32 dynamic power versus frequency ($util = 35\%$, density $= 0.5$, Strategy=3) increases from 36.4 nW at 100 MHz to 75.5 nW at 200 MHz. Linear regression yields:

$$P = 0.39f - 2.78 \text{ (nW, f in MHz)}, R^2 = 0.9998 \quad (2)$$

This near-perfect $P \propto f$ scaling is consistent with the standard dynamic power model $P_{dyn} = \alpha C V^2 f$ at fixed supply voltage (SKY130A TT corner: 1.8 V). The $R^2 = 0.9998$ coefficient validates the accuracy of OpenLane's power extraction pipeline and provides a simple model for operating-point selection in power-constrained applications.

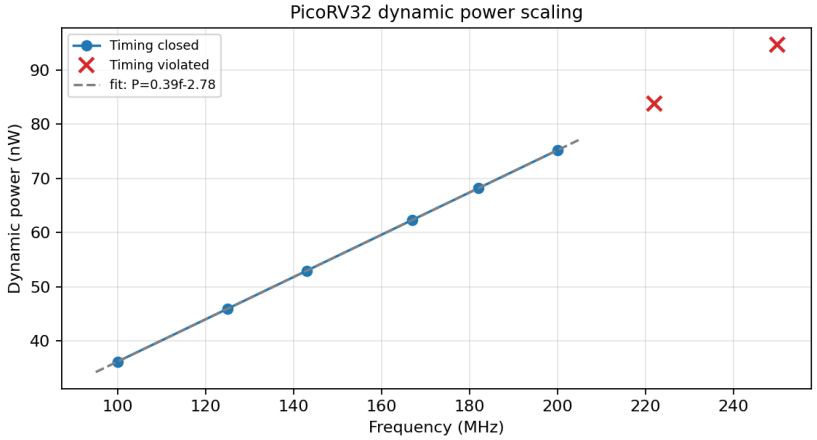

Figure 7. PicoRV32 dynamic power vs. operating frequency ($util = 35\%$, density $= 0.5$, `SYNTH_STRATEGY`=3). Linear fit achieves $R^2 = 0.9998$, confirming $P \propto f$ scaling. Red x marks timing-violated points at 222 MHz and 250 MHz.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

5.7. Effect of SYNTH_STRATEGY Across All Designs

SYNTH_STRATEGY exhibits a striking design-class dependency. For SPM, all four strategies achieve identical closure rates (∼60%) the critical path depth is fixed regardless of cell selection. For PicoRV32, Strategy 3 contributes 31 of 56 closures (55%) versus 12 for Strategy 0, producing ∼21% more cells (∼12,152 vs ∼10,063) through aggressive technology remapping.

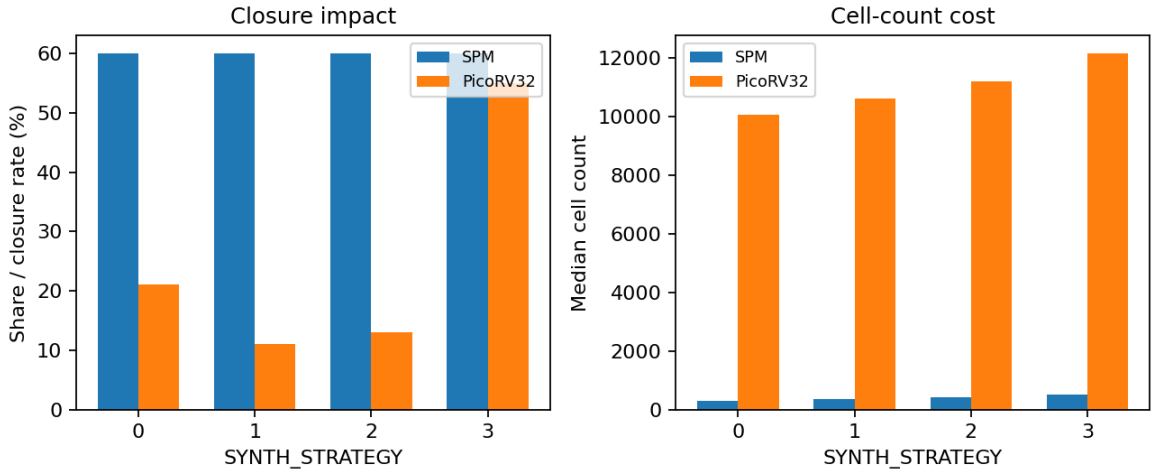

Figure 8. Effect of SYNTH_STRATEGY on closure rate (left) and median cell count (right) for SPM and PicoRV32. Strategy is irrelevant for SPM (60% across all four) but decisive for PicoRV32 Strategy 3 contributes 55% of closures and produces 21% more cells.

Strategy 3 increases cell count by +72% for SPM (301-516), +21% for PicoRV32 (10,063-12,152), and +59% for ai_accel (11,695-18,559).

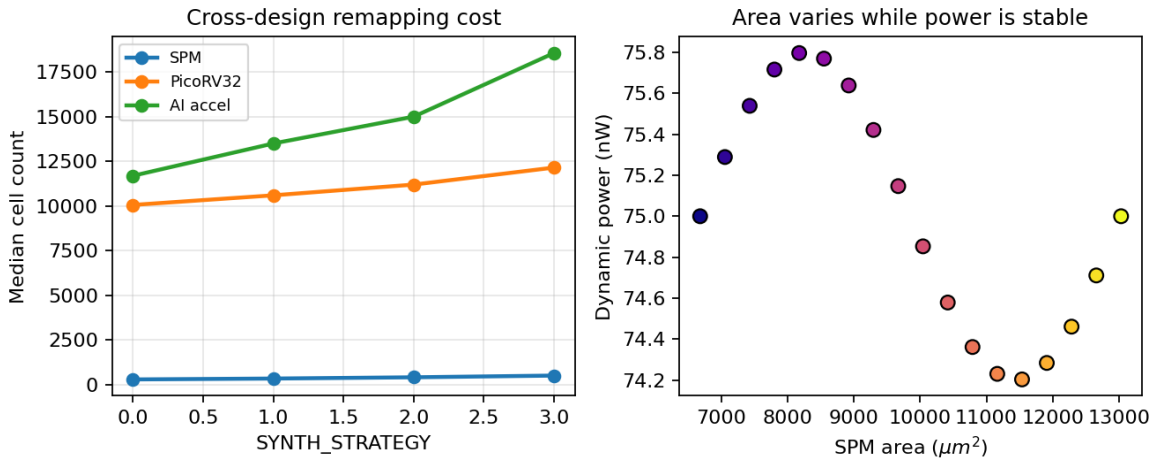

Figure 9. (Left) Median cell count by SYNTH_STRATEGY for all three designs. Strategy 3 increases cells significantly for all designs via technology remapping. (Right) SPM area vs. power at 200 MHz (all timing-closed configs): power is nearly constant while area varies 2x across utilization.

5.8. Effect of Placement Density

Varying PL_TARGET_DENSITY from 0.4 to 0.6 produces negligible area change and less than 1.5% power variation for SPM at 200 MHz. PL_TARGET_DENSITY primarily controls routing convergence, not PPA. Density = 0.4 causes tool failures at utilization ≥ 40% across all designs; density ≥ 0.5 is recommended as the default.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

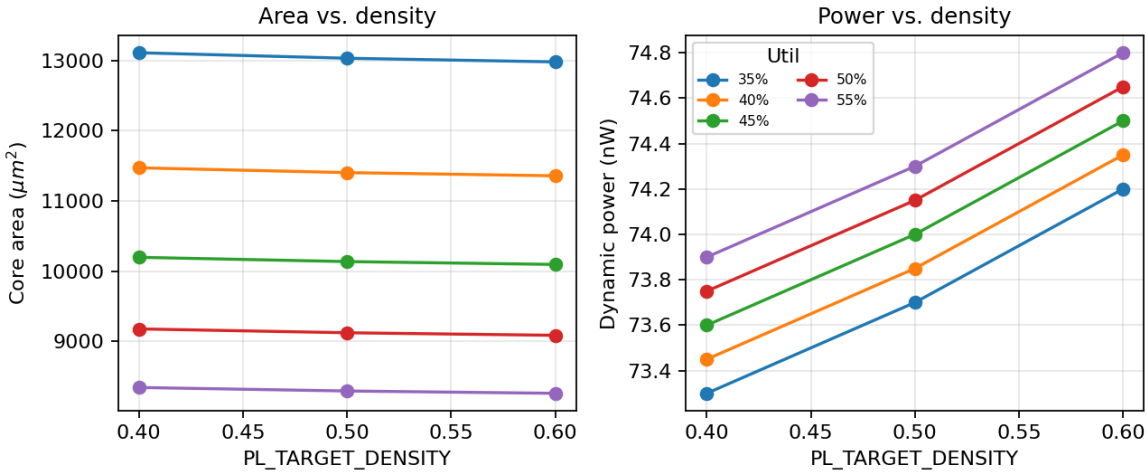

Figure 10. Effect of `PL_TARGET_DENSITY` on area (left) and dynamic power (right) for SPM at 200 MHz across timing-closed configurations. Both metrics are nearly invariant (<1.5% variation), confirming `FP_CORE_UTIL` as the dominant area lever.

6. Discussion

6.1. Design Guidelines for OpenLane Practitioners

Based on 1,440 experimental runs across three designs, we derive four concrete design guidelines:

- **Guideline 1 - Use density ≥ 0.5 as default:** Density $= 0.4$ causes tool failures at utilization $\geq 40\%$ across all tested designs. Density $= 0.5$ or 0.6 provides reliable routing convergence with negligible PPA cost.
- **Guideline 2 - Freely tune utilization for area:** For simple datapath designs, utilization 35-55% is a free area optimization dial (48.7% reduction for SPM) with no timing or power penalty.
- **Guideline 3 - Use `SYNTH_STRATEGY`=3 for complex designs above 150 MHz:** For RISC-V cores and similar designs with >5,000 cells, Strategy 3 is the only enabler at 200 MHz and provides 55% of all PicoRV32 closures. The 21% cell count increase is an acceptable cost.
- **Guideline 4 - Identify architectural limits early:** When best WNS with Strategy 3 is worse than -1 ns, parameter tuning will not achieve closure. Estimate $f_{\max} = 1000/(clock_{\text{period}} + |best_{\text{WNS}}|)$ MHz and modify architecture accordingly.

6.2. Cross-Design Generalization

The three designs reveal a monotonic relationship between complexity and parameter sensitivity. For SPM (∼301 cells, shallow logic), only density affects timing via routing convergence. For PicoRV32 (∼10,063 cells, deep control logic), synthesis strategy becomes the dominant timing lever above 150 MHz. For `ai_accel` (∼11,695 cells, deep datapath), synthesis strategy provides significant improvement (+2.4 ns WNS) but cannot overcome the fundamental multiply-accumulate path limit at 250 MHz. This progression provides a practical decision tree: check density first, apply Strategy 3 for complex designs, estimate $f_{\max}$, then fall back to architecture.

6.3. Limitations

Three primary limitations apply. First, only 4 of OpenLane's 200+ configuration variables were swept; `GRT_ADJUSTMENT`, `PL_RESIZER_TIMING_OPTIMIZATIONS`, and `SYNTH_MAX_FANOUT` may have significant independent effects. Second, all experiments target `sky130_fd_sc_hd`; results may differ for other cell libraries (`sky130_fd_sc_hdll` or `sky130_fd_sc_hs`). Third, the grid search is exhaustive but not sample-efficient; Bayesian optimization could identify the Pareto-optimal front with significantly fewer runs.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 July 2026 doi:10.20944/preprints202607.0137.v1

7. Conclusions

This paper presented a comprehensive automated grid search framework characterizing the timing-area-power tradeoff space of the OpenROAD/OpenLane RTL-to-GDSII flow on SKY130A. The framework executes a full $8 \times 5 \times 3 \times 4$ Cartesian parameter grid across three benchmark designs totaling 1,440 fully automated, unattended runs with fault-tolerant orchestration, automatic disk cleanup, and CSV-backed resume support.

Five key findings: (1) SPM achieves full timing closure at all tested frequencies (100-250 MHz), with a 48.7% area reduction at 55% utilization and closure heatmaps confirming density ≥ 0.5 as the enabling condition. (2) PicoRV32 has a hard closure boundary between 200 and 222 MHz no parameter combination overcomes the critical path limit. (3) The CNN MAC array is limited to $\sim$192 MHz; Strategy 3 achieves -1.20 ns WNS but pipeline insertion is required for 250 MHz operation. (4) `SYNTH_STRATEGY`=3 is decisive for PicoRV32 (55% of closures, +21% cells) but irrelevant for SPM. (5) Dynamic power scales near-linearly with frequency ($R^2 = 0.9998$).

Future work will extend the sweep to additional synthesis and routing parameters, evaluate Bayesian optimization as a sample-efficient complement, apply the framework to cryptographic cores and DSP accelerators, and investigate the `sky130_fd_sc_hdll` library for higher-frequency targets.

References

1. M. Shalan and T. Edwards, "Building OpenLANE: A 130 nm OpenROAD-based tapeout-proven flow," in Proc. IEEE/ACM ICCAD, 2020.
2. T. Ajayi et al., "OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain," in Proc. ACM/IEEE DAC, 2019.
3. SkyWater Technology Foundry, "SKY130 Process Design Kit," https://github.com/google/skywater-pdk, 2020.
4. C. Wolf, "Yosys open synthesis suite," https://github.com/YosysHQ/yosys, 2013.
5. R. T. Edwards, "Magic VLSI Layout Tool," http://opencircuitdesign.com/magic/, 2021.
6. C. Wolf, "PicoRV32 - A size-optimized RISC-V CPU," https://github.com/YosysHQ/picorv32, 2015.
7. M. Nasser et al., "An open-source neural network inference engine on the SKY130 PDK," in Proc. IEEE MWSCAS, 2021.
8. R. Liang et al., "DRILLS: Deep reinforcement learning for logic synthesis," in Proc. ASP-DAC, 2020, pp. 491-496.
9. A. Mirhoseini et al., "A graph placement methodology for fast chip design," Nature, vol. 594, pp. 207-212, 2021.
10. Z. Liu et al., "Global placement with deep learning-enabled routability optimization," in Proc. DATE, 2021.
11. T. Ajayi et al., "Toward an open-source digital flow: First learnings from the OpenROAD project," in Proc. ACM/IEEE DAC, 2019.
12. A. B. Kahng, "New directions for learning-based IC design tools and methodologies," in Proc. ASP-DAC, 2018, pp. 97-104.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.