

CSCO: A Backside-PDN-Aware Clock-Signal Co-Optimization Framework for Improved PPA

Zixiao Wang¹, Leilei Jin^{1,†}, Zhen Zhuang¹, Rongmei Chen², Bei Yu^{1,†}
¹ The Chinese University of Hong Kong ² Peking University

Abstract

Backside power delivery networks (BSPDN) have emerged as a promising technology for advanced logic nodes to address IR-drop and PPA challenges. While BSPDN introduces additional routing resources on the backside, these resources are limited and must be carefully partitioned between clock and signal nets, creating a critical resource allocation tradeoff. Prior work either moves only the clock network or assumes a fixed clock tree and optimizes only signal nets, failing to explore the tradeoff space of backside resource allocation. Moreover, lacking frontside power-ground shielding, BSPDN introduces severe signal integrity (SI) degradation. We propose CSCO, a data-driven BSPDN-aware co-optimization framework that jointly allocates limited backside resources between clock and signal nets across frontside/backside layers. CSCO employs efficient search strategies to identify critical nets for backside routing without repeated evaluation, navigating the clock-signal allocation tradeoff to balance IR-drop, routing congestion, and PPA. The framework also leverages backside routing to mitigate coupling noise and crosstalk-induced SI issues. Experiments demonstrate improved WNS/TNS, frequency, and SI robustness without additional shielding overhead.

1 Introduction

As technology scaling advances into the sub-3nm era, physical design faces increasingly severe back-end-of-line (BEOL) challenges including routing congestion, IR-drop, and reliability degradation [1–4]. Metal resources traditionally allocated for signal routing and power delivery have become insufficient to sustain aggressive power, performance, and area (PPA) targets, making BEOL architecture and power delivery the dominant bottlenecks in system-level scaling. A promising solution has emerged through backside power delivery networks (BSPDN) [5–7]. By relocating the power grid to the wafer backside via nano-through-silicon-via (nTSV) connections, BSPDN alleviates IR-drop while liberating frontside metal layers for denser signal routing and higher operating frequencies as a critical enabler for next-generation design-technology co-optimization (DTCO) and 3D IC integration [8–13].

[†]Corresponding authors.

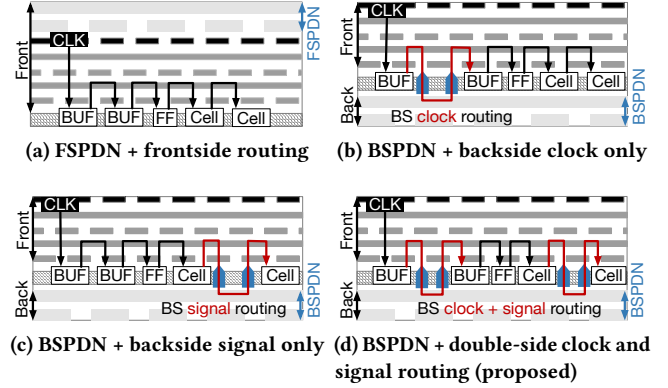


Figure 1: Backside design for clock-signal co-optimization with limited backside resources.

However, this architectural shift breaks the conventional separation between signal, clock, and power networks, demanding new cross-layer co-design methodologies [14–17]. Current research has explored backside routing in progressively broader yet still isolated ways, as illustrated in Fig. 1. Early BSPDN implementations focus exclusively on power delivery without exploiting residual backside routing resources [15]. Subsequent clock-specific approaches (Fig. 1(b)) relocate clock networks using specialized backside buffer cells, demonstrating improvements in clock skew and latency [2, 16, 18]. Other methods [6, 19–21] have explored the relocation of signal nets according to their geometry features (Fig. 1(c)), with an assumption that the clock net is pre-constructed.

Collectively, these BSPDN approaches suffer from three critical limitations that fundamentally constrain achievable PPA gains. First, clock and signal optimization remain decoupled, so methods targeting only clocks neglect timing-critical signal paths, while geometric heuristics for signals often miss compact yet performance-critical nets due to weak correlation between timing criticality and geometric properties. Second, existing approaches usually presuppose fixed power grid configurations (e.g., predetermined metal stripe widths and pitches) prior to net selection. However, the backside net routing and PDN configuration are inherently dependent. PDN synthesis proceeds without knowledge of routing

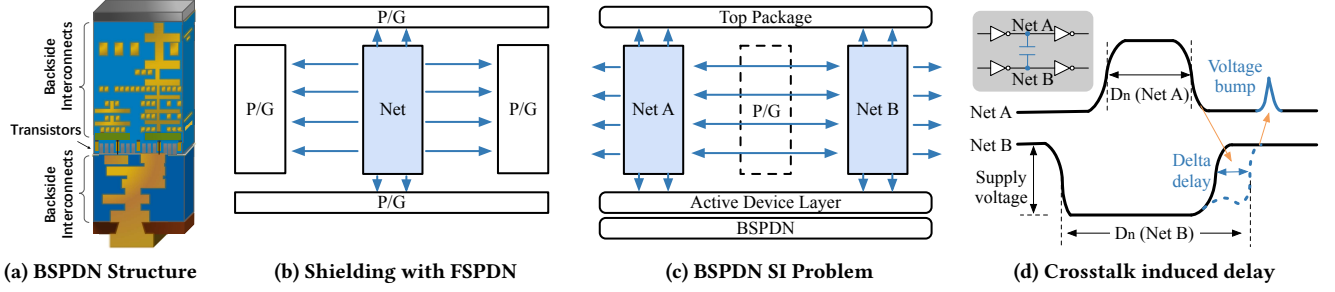


Figure 2: SI problem on BSPDN design.

demands, while routing engines ignore PDN-induced blockages, fragmenting the design space and preventing discovery of design-specific optimal solutions that balance power integrity, routing congestion, and timing closure. Third, existing works overlook a newly emergent physical effect: removing frontside power grids eliminates electromagnetic shielding between signal nets, creating a "gravity-less" coupling environment where crosstalk increases 2–4 $\times$ [22]. Without the confinement of frontside power straps, signal metals experience amplified noise propagation that can offset or even negate BSPDN’s timing benefits [23–25]. This insight motivates a co-optimization framework that exploits backside power layers as shields for SI-critical paths while jointly allocating backside resources across signal and clock networks.

To address these issues, we propose CSCO, a BSPDN-aware co-optimization framework that jointly determines the BSPDN topology and the allocation of signal and clock nets across frontside and backside layers (Fig. 1(d)). Unlike prior approaches that fix PDN configurations before net selection, CSCO recognizes that optimal power grid structures depend on design-specific routing demands and explores both dimensions simultaneously. It employs fast surrogate modeling to navigate the coupled design space without costly full-routing iterations and exploits BSPDN shielding to protect SI-critical nets. This approach enables discovery of the better solutions that balance power integrity, signal integrity, and PPA metrics. Our key contributions are summarized as follows:

- **Unified Co-Optimization Framework:** The first methodology that concurrently synthesizes dual-side signal and clock networks under BSPDN constraints, achieving cross-layer integration of routing and power delivery planning.
- **Data-Driven Surrogate Modeling:** A scalable optimization scheme that employs low-rank statistical modeling to capture timing–resource interactions without costly full-routing evaluations.

- **Signal Integrity Mitigation via BSPDN Shielding:** Exploiting BSPDN’s electromagnetic properties to improve SI robustness.
- **Demonstrated Effectiveness:** On realistic CPU benchmarks, CSCO achieves significant improvement in PPA metrics while satisfying power and signal integrity constraints.

2 Preliminaries

Before presenting our method, we briefly introduce necessary background.

2.1 Backside PDN

BSPDNs use thick, low-resistivity backside metals connected to frontside buried power rails by sub-100-nm nTSVs [7, 26–29]. This separation reduces power-distribution resistance by up to 30% and frees frontside layers for signal routing [10, 12, 13, 17], as shown in Fig. 2(a).

Clock-Centric Backside Routing. Existing approaches [2, 3, 15, 18] relocate entire clock networks using backside buffer cells and nTSVs, improving skew and latency through GNN-based endpoint classification [16] or analytical optimization [18]. However, they ignore timing-critical signal routes and may migrate complete trees when only critical branches need optimization, wasting backside resources and nTSVs.

Signal-Centric Backside Routing. Signal-centric methods [6, 19–21] select nets using bounding-box size, pin count, or routing congestion. These geometric proxies can miss compact critical nets and favor large noncritical nets, producing suboptimal allocations. Net-level migration can also conflict with backside PDN structures, causing DRC violations that require post-routing fixes [15].

Lack of Clock-Signal Co-optimization. Existing works optimize clock and signal routing sequentially [2, 16, 18, 20, 21], despite their interdependence and competition for limited backside resources. Allocating more resources to clocks necessarily leaves fewer for signals, and vice versa. Clock tree synthesis determines signal-path timing budgets, whereas subsequent signal migration cannot influence clock

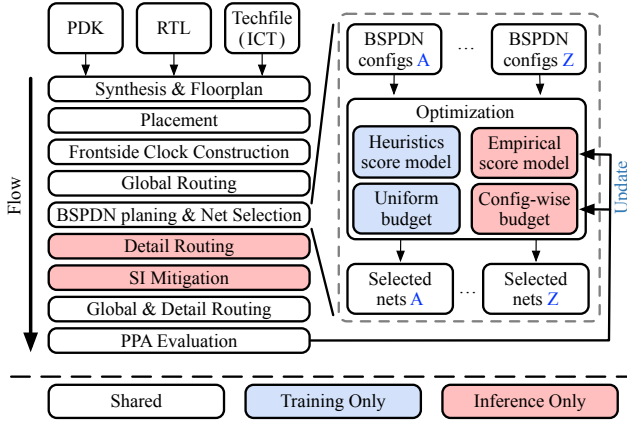


Figure 3: Overall design flow of CSCO.

routing. This separation prevents exploration of clock-signal resource-allocation tradeoffs.

SI Degradation Induced by BSPDN. Relocating the PDN to the backside removes frontside P/G grids that normally shield signal nets [22]. As shown in Figs. 2(b) and 2(c), FSPDN strips maintain favorable self-to-coupling capacitance ratios, whereas BSPDN eliminates this shielding and increases direct coupling between adjacent nets. The resulting aggressor-victim interference causes delta delay ΔD and voltage bumps on timing-critical nets (Fig. 2(d)). Backside signal routing can therefore exacerbate SI issues, requiring joint optimization of timing, routability, and signal integrity under limited backside resources.

2.2 Efficient Surrogate Model

Machine-learning surrogates replace costly physical-design evaluations in EDA. Neural models predict routability [30], pre-routing timing [31], and slack [32], but typically require thousands of full-routing and signoff labels. Gaussian-process Bayesian optimization [33] is more data-efficient and used in analog synthesis [34], but cubic scaling and continuous, low-dimensional assumptions limit high-dimensional binary allocation. Factorization machines (FMs) [35], originally developed for recommendation systems, capture pairwise interactions through low-rank embeddings with $\mathcal{O}(nl)$ complexity, require little data, and naturally handle binary inputs. This compact parameterization suits costly signoff labels. Their success in materials and drug-discovery optimization [36], together with limited use in EDA, motivates our adoption.

2.3 Evaluation Metrics

We evaluate the practical impact of backside allocation on overall design performance.

WNS and TNS. Worst Negative Slack (T_{wns}) and Total Negative Slack (T_{tns}) are standard timing indicators; larger values imply better performance, enabling higher effective frequency.

Delta-Delay and Bump Ratios. Signal integrity is characterized using the delta-delay ratio and bump ratio:

$$r_{delay} = \frac{\Delta D}{D_d + D_n + \Delta D}, \quad r_{bump} = \frac{V_{bump}}{V_{DD}}. \quad (1)$$

Here, D_d and D_n are the intrinsic driver and net delays, and V_{DD} is the supply voltage. Higher ratios indicate stronger SI-induced delay or voltage disturbance, and we additionally count how many nets exceed predefined thresholds for these two metrics.

3 Methods

We now introduce the proposed BSPDN-aware design framework, CSCO, which operates within the backside-aware design flow detailed in Section 4.1. As illustrated in Fig. 3, CSCO augments the conventional 3D-IC implementation process by introducing data-driven optimization stages that jointly consider net partitioning, BSPDN planning, and timing robustness.

Framework Overview. CSCO operates in two major phases, training and inference, to transfer knowledge from empirical exploration to efficient design-space prediction. The training phase focuses on data collection and surrogate model learning, while the inference phase exploits the trained model for BSPDN-aware optimization and timing estimation.

Within the framework, a *key novelty* of CSCO lies in its BSPDN-aware formulation, which links PDN planning, net partitioning, and timing through a unified surrogate-driven optimization flow.

3.1 Problem Formulation

The goal of this stage is to determine the optimal allocation of all nets $\mathcal{N}$ in a given design under a specific BSPDN configuration. Let $\mathcal{N} = \{s_1, \dots, s_m, c_1, \dots, c_n\}$, where s_i and c_j denote signal and clock nets, respectively. With above symbols, we formulate backside resource allocation as an optimization problem under BSPDN-dependent constraints:

$$\text{Eval}(\mathbf{x}) = T_{wns}(\mathbf{x}) + \lambda T_{tns}(\mathbf{x}), \quad (2)$$

where T_{wns} and T_{tns} denote the worst and total negative slack, respectively, and λ controls their relative importance. The design variable is a binary vector

$$x_i \in \{0, 1\}, \quad \forall i \in [1, m + n], \quad (3)$$

where $x_i=1$ indicates that net i is assigned to the backside, otherwise the net is assigned to the frontside.

This optimization problem is challenging for the following three reasons. First, evaluating any candidate $\mathbf{x}$ requires full

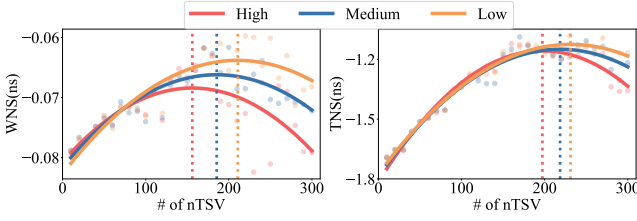


Figure 4: Timing performance (WNS and TNS) as a function of nTSVs count under different BSPDN densities.

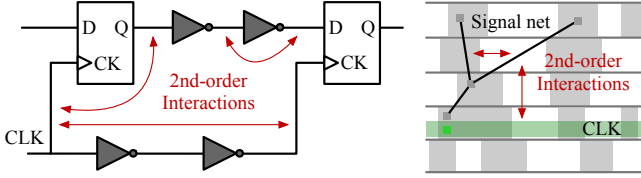


Figure 5: The second-order relationship between nets.

routing and timing analysis, making each trial extremely expensive. Second, the search space is a high-dimensional 0–1 domain with hundreds of thousands of variables, rendering brute-force or classical combinatorial methods infeasible. Third, extending this basic formulation to be explicitly BSPDN-aware is nontrivial: the BSPDN configuration simultaneously impacts IR-drop, routing resources, and timing, and must be incorporated through a compact yet physically meaningful constraint, which we introduce next via a configuration-dependent nTSV budget model.

BSPDN-Dependent Constraints and Budget Determination. Unlike prior works that treat nTSV resources as fixed, we explicitly model BSPDN dependence, allowing power-network density to directly affect routing capacity and timing. A denser BSPDN enhances IR-drop robustness but limits routing resources, so we capture this trade-off through a global nTSV budget constraint:

$$\sum_{i=1}^{m+n} t_i x_i \leq B, \quad (4)$$

where t_i is the number of nTSVs required by net i (equal to its pin count). Timing improves as B increases until a configuration-dependent threshold, beyond which excessive nTSVs degrade routability, as shown in Fig. 4.

The budget B is determined empirically by sweeping nTSV allocations and recording corresponding PPA outcomes. The inflection point of the performance–nTSV curve is selected as the final budget, which remains fixed during inference for consistent evaluation across configurations. This BSPDN-aware formulation links power-delivery planning with timing optimization, bridging two stages traditionally treated independently in 3D-IC design flows.

However, a direct optimization on Eqs. (2) to (4) is computationally prohibitive due to expensive routing and timing analysis. To address this challenge, we employ a data-efficient surrogate model to approximate $\text{Eval}(\mathbf{x})$ and enable scalable optimization.

3.2 Surrogate Model

The true evaluation function $\text{Eval}(\mathbf{x})$ requires full routing and timing analysis, making data collection extremely expensive. Such scarcity renders deep-learning surrogates impractical, motivating the use of a lightweight and interpretable model that can learn from limited samples.

Low-Rank Interaction Modeling. To efficiently approximate the objective function Eq. (2) with limited data, we employ a second-order FM model [35]. Given a binary partition vector $\mathbf{x} \in \{0, 1\}^{m+n}$ indicating backside assignment, the k -order FM prediction is defined as:

$$\hat{y}(\mathbf{x}) = w_0 + \mathbf{w}^\top \mathbf{x} + \sum_{d=2}^k \sum_{i_1 < \dots < i_d} \langle \mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_d} \rangle \prod_{j=1}^d x_{i_j}, \quad (5)$$

where $w_0 \in \mathbb{R}$ is a global bias, $\mathbf{w} \in \mathbb{R}^{m+n}$ models first-order (individual-net) effects, $\mathbf{v}_i \in \mathbb{R}^l$ is a latent embedding vector representing the hidden timing characteristics of net i and the operator $\langle \mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_d} \rangle = \sum_{f=1}^l \prod_{j=1}^d v_{i_j, f}$ captures d -way interactions. All $\mathbf{v}_i$ are assembled into the latent matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+n}]^\top$, so that $\{w_0, \mathbf{w}, \mathbf{V}\}$ constitute all learnable parameters of the model.

In this work, we adopt the second-order case ($k=2$) for several circuit-level considerations: 1) most timing dependencies arise from *pairwise* interactions, including between signal nets connected to the same cell, and between clock and signal nets along critical paths, as illustrated in Fig. 5. Higher-order correlations contribute marginally yet increase data complexity. Thus, a second-order FM effectively captures the dominant physical dependencies while remaining data efficient. 2) Compared with neural surrogates, FM offers greater interpretability and stability under limited training data, making it particularly suitable for BSPDN-aware optimization. 3) Furthermore, once trained, the FM model generalizes across different BSPDN configurations without retraining, since it encodes intrinsic inter-net relationships rather than configuration-specific parameters. This cross-configuration consistency allows the same surrogate to guide optimization under varying PDN densities.

Additionally, a standard simplification [35] for $k=2$ reduces computational complexity from $\mathcal{O}((m+n)^2 l)$ in Eq. (5) to $\mathcal{O}((m+n)l)$:

$$\hat{y}(\mathbf{x}) = w_0 + \mathbf{w}^\top \mathbf{x} + \frac{1}{2} (\|\mathbf{V}^\top \mathbf{x}\|_2^2 - \|\text{Diag}(\mathbf{x}) \mathbf{V}\|_F^2), \quad (6)$$

which significantly improves efficiency and is well-suited for GPU parallelization.

Table 1: Normalized post-global-routing metrics used for cold-start net selection.

Symbol (Unit)	Definition	Range	Net Type
$\tilde{t}_{\text{tran}}$ (ps)	Transition delay	≥ 0	Both
$\tilde{r}_{\text{first}}$ (N/A)	Rank of first critical path	$\{1, \dots, m+n\}$	Both
$\tilde{n}_{\text{load}}$ (N/A)	Number of loads	$\mathbb{Z}^+$	Both
$\tilde{C}_{\text{est}}$ (fF)	Estimated net capacitance	≥ 0	Signal
$\tilde{N}_{\text{path}}$ (N/A)	Occurrence count in all paths	$\{1, \dots, m+n\}$	Signal
$\tilde{t}_{\text{arr}}$ (ns)	Arrival time	≥ 0	Clock
$\tilde{N}_{\text{crit}}$ (N/A)	Occurrences in top-200 critical paths	$\{0, \dots, 200\}$	Clock

Cold-Start Data Construction. Before training the surrogate model, we require an initial dataset for cold-start learning. We propose an empirical selection criterion to identify nets most likely to impact timing when migrated to the backside. For each net type (signal or clock), we extract five post-global-routing metrics that are lightweight to compute and available without detailed routing, as summarized in Table 1.

After normalizing each metric to $[0, 1]$, we compute a composite score quantifying the timing improvement potential of each net:

$$\text{Score}_{\text{sig}} = \alpha_1 \tilde{t}_{\text{tran}} + \alpha_2 \tilde{n}_{\text{load}} + \alpha_3 \tilde{C}_{\text{est}} + \alpha_4 \tilde{N}_{\text{path}} - \alpha_5 \tilde{r}_{\text{first}}, \quad (7)$$

$$\text{Score}_{\text{clk}} = \beta_1 \tilde{t}_{\text{tran}} + \beta_2 \tilde{n}_{\text{load}} + \beta_3 \tilde{N}_{\text{crit}} - \beta_4 \tilde{t}_{\text{arr}} - \beta_5 \tilde{r}_{\text{first}}, \quad (8)$$

where $(\tilde{\cdot})$ denote normalized values and α_i, β_i are weighting coefficients (set equally in this work). A higher score indicates a stronger likelihood that the corresponding net will contribute positively to timing performance if allocated to the backside.

This criterion efficiently constructs the training dataset by focusing computation on timing-critical nets, thereby reducing routing iterations required for model initialization.

3.3 Optimization and Inference

With the surrogate model established, optimization proceeds in two phases: the training phase for generating diverse samples, and the inference phase for obtaining the BSPDN-aware partitioning.

Training Phase: Greedy + Exploration. During training, we aim to collect high-quality yet diverse partitioning samples. Nets are first ranked by their normalized utility, defined as the empirical score (Eqs. (7) and (8)) divided by the required number of nTSVs t_i . We then perform greedy selection under a preset clock/signal ratio until the total nTSV budget is reached. To enhance diversity, a fixed fraction of selected nets is randomly replaced in each run, enabling broader exploration and improving surrogate coverage.

Inference Phase: FM-Guided Greedy + Simulated Annealing. In inference, the trained FM model provides a fast

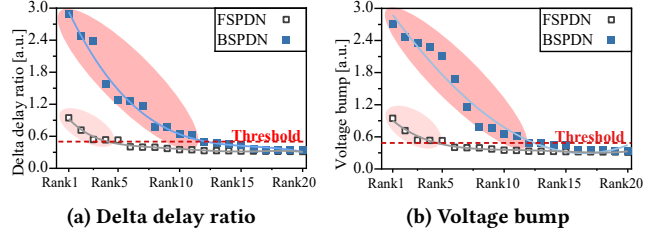


Figure 6: Crosstalk impacts on nets: voltage bump and delta-delay at 4GHz operating frequency.

estimation of timing performance. For each net, we compute its importance by combining the first-order term and its pairwise interactions in the FM output, then divide by t_i to reflect its benefit per backside cost:

$$I_i = \frac{w_i + \sum_{j \neq i} \langle v_i, v_j \rangle x_j}{t_i}, \quad (9)$$

where I_i denotes the importance of net i , w_i is its first-order coefficient, $\langle v_i, v_j \rangle$ represents the pairwise interaction from the FM model, and t_i is the number of nTSVs required by the net. We again perform greedy selection under the clock/signal ratio until the total budget is filled, yielding an initial feasible solution. To further refine this solution, we adopt simulated annealing (SA) directly on the surrogate. At each iteration, a few nets are swapped between dies to obtain a new candidate $\mathbf{x}'$, and the update is accepted with probability:

$$P_{\text{accept}} = \min \left(1, \exp \left(\frac{\hat{y}(\mathbf{x}') - \hat{y}(\mathbf{x})}{T} \right) \right), \quad (10)$$

where $\hat{y}(\cdot)$ is the FM-predicted performance and T is the temperature that gradually cools down (e.g., $T_{t+1} = \eta T_t$, $\eta \in (0, 1)$). The optimization stops when no further improvement is observed or a preset iteration limit is reached. The final partitioning $\mathbf{x}^*$ is used for full routing and PPA evaluation.

This two-phase design balances exploration and exploitation: the training phase diversifies data for surrogate learning, while the inference phase leverages the surrogate to efficiently refine the final BSPDN-aware allocation.

3.4 Signal Integrity Mitigation

Based on post-detailed-routing analysis of our initial BSPDN and FSPDN designs, we observe notable differences in signal integrity behavior. As shown in Fig. 6, BSPDN exhibits 2–4× higher voltage bumps and delay variations than FSPDN. The exponential decay trend in both metrics indicates that only a subset of nets is inherently more vulnerable to SI degradation. The red-shaded regions in Fig. 6 mark the performance gap between the two PDN schemes, with the red dashed lines denoting the 0.5 ratio safety threshold. In our methodology,

Table 2: Performance metrics comparison across all benchmarks at 2GHz target frequency. Bold values indicate the best results. FS denotes frontside PDN with frontside routing. BSPDN+FR denotes BSPDN with frontside routing.

	SHA256 (#Cell = 7.9k)					JPEG (#Cell = 182k)					ARM Cortex-A7 (#Cell = 108k)				
	FS	BSPDN+FR	DAC24[16]	DAC25[18]	CSCO	FS	BSPDN+FR	DAC24[16]	DAC25[18]	CSCO	FS	BSPDN+FR	DAC24[16]	DAC25[18]	CSCO
Utilization (%)	82.91	82.36	82.36	82.36	82.36	78.24	75.50	75.50	75.50	75.50	78.16	74.15	74.15	74.15	74.15
Area (μm^2)	1319	1311	1311	1311	1311	31285	32280	32280	32280	32280	14267	13491	13491	13491	13491
#Metal (BS+FS)	0+6	3+6	3+6	3+6	3+6	0+7	3+7	3+7	3+7	3+7	0+8	3+8	3+8	3+8	3+8
#nTSV	N/A	N/A	270	194	230	N/A	N/A	396	768	800	N/A	N/A	316	634	600
Clock Skew (ps)	24.17	27.94	27.02	19.72	32.09	112.16	35.37	31.93	24.86	34.17	217.67	95.06	66.79	52.97	43.49
Clock Latency (ps)	83.29	75.31	64.63	51.47	75.13	226.26	149.31	112.11	79.31	124.46	456.98	206.51	147.67	123.87	135.80
Wirelength (m)	0.05	0.04	0.04	0.04	0.04	0.81	0.70	0.69	0.68	0.69	1.47	1.37	1.38	1.38	1.39
Total Power (mW)	7.41	7.17	6.77	7.09	7.05	152.92	143.42	149.01	137.44	140.21	140.97	131.93	122.67	120.01	142.68
IR-drop (mV)	62.54	14.13	14.32	14.25	14.35	55.95	23.79	24.09	24.02	25.99	170.02	17.88	18.93	18.03	18.58
T_{wns} (ns)	-0.196	-0.111	-0.094	-0.106	-0.061	-0.150	-0.126	-0.122	-0.125	-0.077	-0.194	-0.081	-0.092	-0.106	-0.011
T_{tms} (ns)	-4.06	-1.96	-2.04	-2.14	-1.09	-28.53	-25.25	-24.44	-25.06	-7.66	-1.756	-1.637	-1.579	-1.499	-1.139
Eff. Freq (GHz)	1.50	1.64	1.70	1.65	1.78	1.54	1.61	1.63	1.60	1.73	1.44	1.72	1.69	1.65	1.96
# $r_{\text{delta}} > 40\%$	3	8	8	8	2	3	16	16	16	4	7	17	17	17	2
# $r_{\text{bump}} > 30\%$	1	3	3	3	0	2	12	12	12	2	5	27	26	27	3

Table 3: Technology specifications and design rules.

BS	Width/Spacing(μm)	Pitch(μm)	FS	Width/Spacing(μm)	Pitch(μm)
BSM0	0.053 / 0.028	0.04	M0	0.020 / 0.020	0.04
BSM1	0.125 / 0.375	0.5	M1	0.037 / 0.020	0.06
BSM2	0.5 / 1.5	2	M2/3	0.020 / 0.020	0.04
BSM3	0.5 / 1.5	2	M4-M8	0.038 / 0.038	0.08

nets with voltage bump or delta-delay ratios exceeding 40% are flagged as SI-critical. These nets are ranked by severity and explicitly reassigned to backside routing tracks, where stronger electromagnetic shielding can effectively suppress excessive interference, while preserving global routing balance and timing closure.

This SI optimization stage is designed to operate independently from the primary net selection in our flow, since SI degradation is driven more by routing congestion and net density than by timing metrics like WNS or TNS. Accordingly, we reserve backside routing resources for SI-critical nets, which proves more resource-efficient than extending BSPDN power rails into frontside metal layers.

4 Experimental Results

This section presents experiments on representative designs to validate the effectiveness of the proposed CSCO method. Both quantitative and qualitative results are reported.

4.1 Experiment Settings

Technology Specifications. Our BSPDN uses nTSVs to connect buried power rails (BPRs) to backside metal layers [27]. BPRs reduce standard-cell height by eliminating conventional VDD/VSS BEOL layers above the FEOL. Table 3 summarizes frontside and backside metal-stack parameters.

Backside-Aware Design Flow. Since commercial implementation and signoff tools do not natively support double-sided flows with explicit backside metal layers, we developed a custom Interconnect Technology file (ICT) defining a backside-optimized BSM1–BSM3 BEOL stack and generated corresponding QRC files for parasitic extraction. TCAD simulations were translated into SPICE-compatible nTSV models with $C_{\text{nTSV}} = 0.444 \text{ fF}$ and $R_{\text{nTSV}} = 23 \Omega$.

The flow iterates frontside and backside implementation with PDN-aware backside routing, parasitic extraction and merge, and signoff timing. It provides parasitic and timing feedback while exploiting backside routing resources without major modifications to established flows.

Benchmark Designs. We evaluate SHA256 (172 clock nets and 7.8k signal nets), JPEG (3.5k clock nets and 180k signal nets) [37], and ARM Cortex-A7 (1.5k clock nets and 116k signal nets) [38], a power-efficient CPU for mobile and embedded systems. All designs are implemented in 7 nm technology ($V_{\text{DD}} = 0.75 \text{ V}$) using Cadence Genus [39] and Innovus [40]. Power integrity is evaluated with an in-house IR-drop tool [1], and timing is signed off using Synopsys PrimeTime SI [41].

Implementation. CSCO is implemented in Python 3.8 and evaluated on a Rocky Linux 8.10 cluster with two Xeon Platinum 8480+ CPUs (112 cores) and two NVIDIA L40S GPUs.

Hyper-parameters. We set the loss weight $\lambda = 0.1$ and latent embedding dimension $l = 12$. The FM is trained using MSE and SGD (batch size 32, learning rate 10^{-4} , and 5×10^3 update steps), with 20% of greedily selected nets randomly replaced for exploration. We collect 200 samples for SHA256 and 500 for both JPEG and ARM. During inference, simulated annealing starts at $T = 10^{-5}$, cools with $\eta = 0.995$, and stops at 10^{-6} . Clock and signal net budgets are fixed at 30% and 70%, respectively.

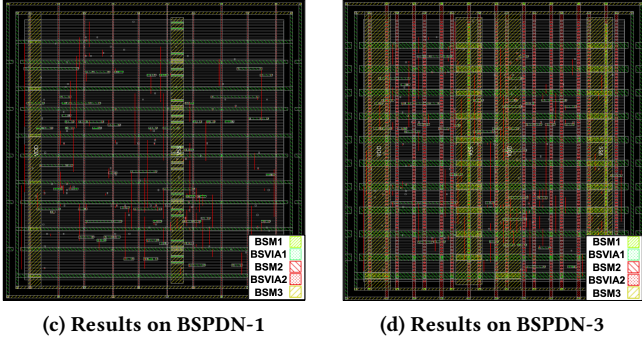
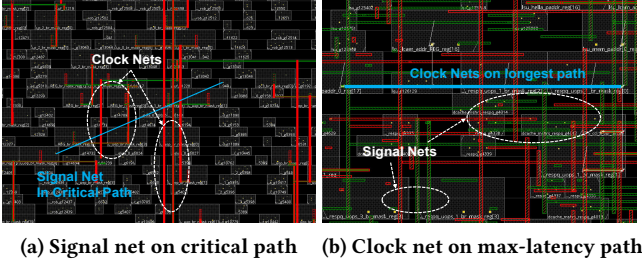


Figure 7: Routing critical nets through backside and CSCO results under various BSPDN configurations.

4.2 PPA Evaluations

In addition to the metrics from Section 2.3, we report nTSV count, clock skew, wire length, total power, effective frequency, and IR-drop. Table 2 compares CSCO across three benchmarks with baselines and two competitors, DAC24 [16] and DAC25 [18]. FSPDN+FR and BSPDN+FR perform frontside-only routing with FSPDN and BSPDN, respectively; DAC24 applies GNN-based backside net assignment, whereas DAC25 routes critical-path clock nets on the backside.

CSCO achieves the best timing across all methods, improving WNS by up to 85% on ARM Cortex-A7 and by over 60% on average, and TNS by 35–70%. These gains increase the effective frequency to 1.78GHz for SHA256 (20% over FSPDN) and 1.96GHz for ARM (36.1% over the baseline). CSCO also uses fewer nTSVs than DAC25 with negligible area overhead (e.g., 74.15% utilization for ARM), demonstrating an effective balance between performance and resource efficiency.

Visualization. Figs. 7(a) and 7(b) visualizes critical-path signal and clock nets. Frontside congestion limits their routing efficiency, whereas backside routing expands the solution space for timing optimization.

Figs. 7(c) and 7(d) shows CSCO’s backside routing results on SHA256 under two BSPDN configurations, illustrating how available backside resources affect routing patterns and utilization. Fig. 9(a) compares heuristic and empirical-model

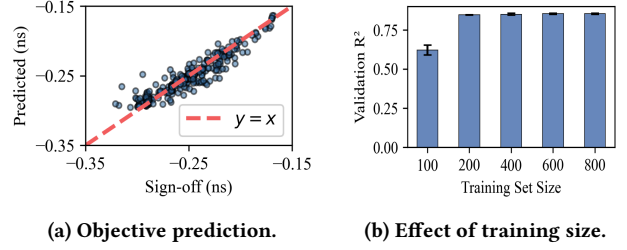


Figure 8: Surrogate model evaluation.

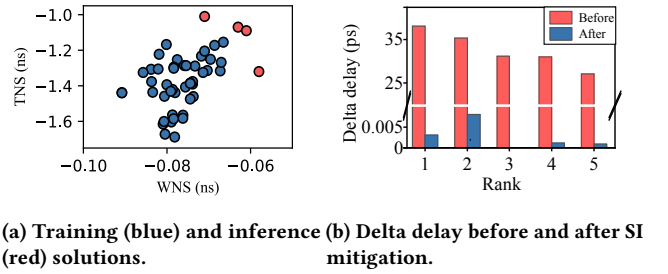


Figure 9: Solution quality and SI analysis.

solutions, showing that CSCO extends the Pareto frontier toward better solutions.

4.3 Surrogate Model

To understand how CSCO works, we evaluate the proposed surrogate FM model across multiple dimensions here.

Accuracy and Consistency. The surrogate FM in Section 3.2 serves as a fast proxy for sign-off timing. On 337 held-out backside routing allocations from SHA256, its predictions correlate closely with the sign-off objective, achieving an R^2 of 0.858 and an RMSE of 0.0141 ns; the latter is small relative to the approximately 0.17 ns baseline objective. The correlation is particularly strong in the upper-right region of Fig. 8(a), where better-timing solutions lie, making the model suitable for search guidance. It also preserves candidate rankings, with top- K overlaps of 100%, 90%, and 85% at $K = 5, 10$, and 20, respectively. **Training Robustness.** We evaluate validation R^2 on SHA256 with different training-set sizes, reporting the mean and standard deviation over three runs in Fig. 8(b). At 200 samples, the consistently small variation indicates robustness to data sampling and optimization randomness. The model converges rapidly and generalizes well with only a few hundred samples, reducing calls to the expensive sign-off engine and accelerating co-optimization. **Efficiency.** Training completes within 15 minutes on a single GPU, and each backside routing assignment is evaluated in

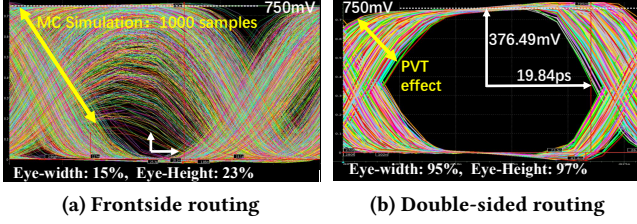


Figure 10: SI mitigation on the worst-case net: eye diagrams with (a) frontside and (b) double-sided routing.

under 30 ms, compared with approximately 323.4 seconds per iteration for PrimeTime sign-off. Replacing sign-off analysis with the surrogate thus provides over 10,000 $\times$ faster objective evaluation. Moreover, one model applies to all BSPDN configurations of a given design without retraining, further reducing computational cost.

4.4 Signal Integrity Analysis

We count nets with $r_{delay} > 0.4$ or $r_{bump} > 0.3$ as SI violations. Table 2 shows that replacing FSPDN with naive BSPDN+FR increases both violations across all benchmarks. On ARM Cortex-A7, the delay- and bump-violation counts increase from 7 to 17 and from 5 to 27, respectively. CSCO reduces the totals across all three benchmarks from 41 to 8 (80.5%) and from 42 to 5 (88.1%). On ARM, these counts decrease to 2 and 3, demonstrating that CSCO suppresses aggressor-victim coupling while preserving timing closure.

Fig. 9(b) shows that the maximum delta delay among the top five critical nets decreases from approximately 30 ps to below 0.01 ps. In Figs. 10(a) and 10(b), CSCO-enabled double-sided routing improves the normalized eye width and height of the worst-case net from 15%/23% to 95%/97%, corresponding to an absolute eye width of 19.84 ps and eye height of 376.49 mV. These results demonstrate robust high-frequency operation without additional area overhead.

4.5 Joint BSPDN and Net Allocation

CSCO jointly optimizes BSPDN configurations and net allocation. As summarized in Table 4, progressively wider power stripes reduce line resistance at the cost of higher overall capacitance and less routing space. The static IR-drop maps in Fig. 11 illustrate the resulting trade-off between power integrity and routing congestion. Unlike approaches requiring optimization or retraining for each PDN setting, CSCO reuses one surrogate across heterogeneous BSPDN layouts to efficiently explore the joint PDN-routing space and identify Pareto-efficient net-to-layer assignments.

Table 5 shows a non-monotonic trade-off among the configurations. BSPDN-3 minimizes IR-drop but incurs slightly

Table 4: BSPDN power-rail configurations. Entries list BSM1/BSM2/BSM3 values; r and c denote resistance and capacitance per unit length.

BSPDN Config	BSPDN-1	BSPDN-2	BSPDN-3
Power Rail Width (μm)	0.10/0.40/1.0	0.20/0.60/2.0	0.35/0.80/3.0
Power Rail Pitch (μm)	2.1/1.8/10	2.1/1.8/10	2.1/1.8/10
Unit r ($\Omega/\mu\text{m}$)	4.20/0.21/0.016	1.86/0.14/0.008	1.10/0.10/0.005
Unit c (fF/ μm)	0.41/0.23/0.083	0.37/0.21/0.153	0.39/0.30/0.270

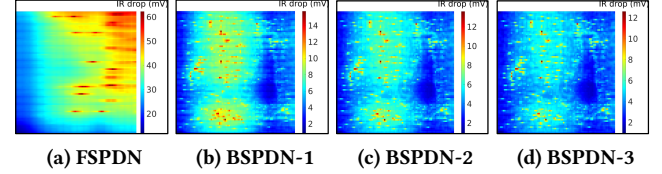


Figure 11: Static IR-drop maps for SHA256 under FSPDN and three BSPDN configurations (design area: $40 \times 40 \mu\text{m}^2$).

Table 5: The solutions given by CSCO on SHA256 with various BSPDN configurations.

Configs	IR-Drop (mV)	#nTSV	WNS (ns)	TNS (ns)	$r_{delay} > 0.4$	$r_{bump} > 0.3$
BSPDN-1	15.76	240	-0.060	-1.17	2	1
BSPDN-2	14.35	230	-0.061	-1.09	2	0
BSPDN-3	13.61	210	-0.067	-1.18	1	1

worse timing from higher parasitics; BSPDN-1 achieves the best WNS but higher IR-drop. BSPDN-2 provides the most balanced solution, combining low IR-drop with the best TNS and stable WNS. These results show that CSCO captures configuration-dependent trade-offs and guides topology selection according to design-specific PPA objectives.

5 Conclusion

This paper presents CSCO, a BSPDN-aware co-optimization framework that jointly determines PDN topology and dual-side clock/signal routing. By integrating PDN planning with backside-aware net allocation and leveraging data-driven surrogate modeling, CSCO efficiently explores the coupled design space without costly full-routing iterations. Unlike prior methods, CSCO accounts for amplified crosstalk introduced by removing frontside power shields and exploits BSPDN metals to protect SI-critical paths. Evaluations on realistic benchmark designs demonstrate that CSCO achieves significant PPA gains while satisfying power and signal integrity constraints, establishing a unified methodology for next-generation DTCO and 3D ICs.

References

- [1] R. Chen, M. Lofrano, G. Mirabelli *et al.*, “Power, performance, area and thermal analysis of 2D and 3D ICs at A14 node designed with back-side power delivery network,” in *Proc. IEDM*. IEEE, 2022, pp. 23–4.
- [2] P. Vanna-iampikul, H. Yang, J. Kwak *et al.*, “Back-side design methodology for power delivery network and clock routing,” in *Proc. IEEE VLSI Symp.*, 2024, pp. 1–2.
- [3] K. Subramani, R. Sengupta, M. Loden *et al.*, “Backside routing enablement considerations for advanced node GAA devices,” in *Proc. IEEE VLSI Symp.* IEEE, 2025, pp. 1–3.
- [4] F. Xie and T. Wei, “Exploring efficient thermal management solutions for backside power delivery network (BSPDN) systems using multi-scale modeling,” *IEEE TCPMT*, 2025.
- [5] R. Chen, G. Sisto, M. Stucchi *et al.*, “Backside pdn and 2.5 d mimcap to double boost 2d and 3d ics ir-drop beyond 2nm node,” in *Proc. IEEE VLSI Symp.* IEEE, 2022, pp. 429–430.
- [6] J. Lee, S. Lee, Y. Ahn *et al.*, “A novel backside signal inter/intra-cell routing method beyond backside power for angstrom nodes,” in *Proc. IEEE VLSI Symp.* IEEE, 2025, pp. 1–3.
- [7] S. Yang, P. Schuddinck, M. Garcia-Bardon *et al.*, “ppa and scaling potential of backside power options in n2 and a14 nanosheet technology,” in *Proc. IEEE VLSI Symp.* IEEE, 2023, pp. 1–2.
- [8] S. S. K. Pentapati, K. Chang, V. Gerousis *et al.*, “Pin-3d: A physical synthesis and post-layout optimization flow for heterogeneous monolithic 3d ics,” in *Proc. ICCAD*, 2020, pp. 1–9.
- [9] S. T. Nibhanupudi, S. Oruganti, R. Mathur *et al.*, “Buried power rails and back-side power grids: Prospects and challenges,” in *Proc. DAC*. IEEE, 2023, pp. 1–4.
- [10] B.-S. Kim, S. Choi, J. H. Lee *et al.*, “Expanding design technology co-optimization potentials with back-side interconnect innovation,” in *Proc. IEEE VLSI Symp.* IEEE, 2024, pp. 1–2.
- [11] J.-S. Kim, J. Kim, D.-J. Yang *et al.*, “Addressing interconnect challenges for enhanced computing performance,” *Science*, vol. 386, no. 6727, p. eadk6189, 2024.
- [12] S. Choi, J. Jung, A. B. Kahng *et al.*, “PROBE3.0: a systematic framework for design-technology pathfinding with improved design enablement,” *IEEE TCAD*, vol. 43, no. 4, pp. 1218–1231, 2023.
- [13] A. Veloso, B. Vermeersch, R. Chen *et al.*, “Backside power delivery: game changer and key enabler of advanced logic scaling and new stco opportunities,” in *Proc. IEDM*. IEEE, 2023, pp. 1–4.
- [14] M. O. Hossen, B. Chava, G. Van der Plas *et al.*, “Power delivery network (pdn) modeling for backside-pdn configurations with buried power rails and *mu* tsvs,” *IEEE TED*, vol. 67, no. 1, pp. 11–17, 2019.
- [15] M. G. Park, A. Rahman, and S. K. Lim, “BS-PDN-Last: Towards optimal power delivery network design with multifunctional backside metal layers,” in *Proc. DAC*, 2025, pp. 1–6.
- [16] N. E. Bethur, P. Vanna-iampikul, O. Zografos *et al.*, “GNN-assisted back-side clock routing methodology for advance technologies,” in *Proc. DAC*, 2024.
- [17] S. Choi, C. Gilardi, P. Gutwin *et al.*, “Omni 3d: Beol-compatible 3-d logic with omnipresent power, signal, and clock,” *IEEE TED*, 2025.
- [18] X. Jiang, H. Lu, Y. Zhao *et al.*, “A systematic approach for multi-objective double-side clock tree synthesis,” in *Proc. DAC*. IEEE, 2025, pp. 1–7.
- [19] F.-Y. Hsu, T.-C. Lin, W.-K. Mak *et al.*, “A bounding box-based net partitioning method for double-sided routing,” in *Proc. GLSVLSI*, 2024, pp. 397–402.
- [20] T.-C. Lin, F.-Y. Hsu, W.-K. Mak *et al.*, “An effective netlist planning approach for double-sided signal routing,” in *Proc. ASPDAC*, 2024, pp. 288–293.
- [21] C.-P. Tsai, F.-Y. Hsu, W.-K. Mak *et al.*, “An effective ECO methodology for reducing back-side design rule violations in double-sided signal routing,” in *Proc. ICCAD*, 2024, pp. 1–9.
- [22] P. Dubey, O. Zografos, N. Pantano *et al.*, “Managing crosstalk in multi-ghz front side clock for back side power enabled sub-2nm 2d/3d ics,” in *Proc. ICICDT*. IEEE, 2024, pp. 1–4.
- [23] V. A. Chhabria, B. Keller, Y. Zhang, and *et al.*, “XT-PRAGGMA: Crosstalk pessimism reduction achieved with GPU gate-level simulations and machine learning,” in *Proc. MLCAD*, 2022, pp. 63–69.
- [24] R. Liang, Z. Xie, J. Jung, and *et al.*, “Routing-free crosstalk prediction,” in *Proc. ICCAD*, 2020, pp. 1–9.
- [25] F. Liu, G. Guo, Y. Ye *et al.*, “Graphcad: Leveraging graph neural networks for accuracy prediction handling crosstalk-affected delays,” in *Proc. ISPD*, 2025, pp. 125–133.
- [26] Y. Zhou, S. Song, H. Kükner *et al.*, “Backside power delivery in high density and high performance context: Ir-drop and block-level power-performance-area benefits,” in *Proc. IEEE VLSI Symp.* IEEE, 2024, pp. 1–2.
- [27] L. Wang, F. Xie, J. Liu *et al.*, “Power and thermal integrity analysis of high performance and low power cpus at sub-2nm node designed with various advanced backside pdns,” in *Proc. IEDM*, 2024, pp. 1–4.
- [28] A. Rahman, H. Yang, C. Hao *et al.*, “Modeling and design methodology for backside integration of voltage converters,” in *Proc. ISPD*, 2025, pp. 242–250.
- [29] H. Lu, Y. Ge, X. Jiang *et al.*, “First experimental demonstration of self-aligned flip fet (ffet): A breakthrough stacked transistor technology with 2.5 t design, dual-side active and interconnects,” in *Proc. IEEE VLSI Symp.* IEEE, 2024, pp. 1–2.
- [30] Z. Xie, Y.-H. Huang, G.-Q. Fang *et al.*, “Routenet: Routability prediction for mixed-size designs using convolutional neural network,” in *Proc. ICCAD*, 2018, pp. 1–8.
- [31] E. C. Barboza, N. Shukla, Y. Chen *et al.*, “Machine learning-based pre-routing timing prediction with reduced pessimism,” in *Proc. DAC*, 2019, pp. 1–6.
- [32] Z. Guo, M. Liu, J. Gu *et al.*, “A timing engine inspired graph neural network model for pre-routing slack prediction,” in *Proc. DAC*, 2022, pp. 1207–1212.
- [33] B. Shahriari, K. Swersky, Z. Wang *et al.*, “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [34] W. Lyu, F. Yang, C. Yan *et al.*, “Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design,” in *Proc. ICML*. PMLR, 2018, pp. 3306–3314.
- [35] S. Rendle, “Factorization machines,” in *Proc. ICDM*. IEEE, 2010, pp. 995–1000.
- [36] R. Tamura, Y. Seki, Y. Minamoto *et al.*, “Black-box optimization using factorization and ising machines,” 2025. [Online]. Available: <https://arxiv.org/abs/2507.18003>
- [37] OpenCores: The reference community for Free and Open Source gateware IP cores, <https://opencores.org/>.
- [38] ARM Cortex-A7 Processor, <https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a7>.
- [39] Cadence, “Genus’s user guide,” https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html.
- [40] Cadence, “Innovus Reference Manual,” https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/innovus-implementation-system.html.
- [41] Synopsys, “Primetime User Guide,” <https://www.synopsys.com/implementation-and-signoff/signoff/primetime.html>.