Toward Compact Planar Circuits for Atomic-Scale Computing: A Hybrid Crossing Elimination Flow

Benjamin Hien, Marcel Walter, and Robert Wille
https://www.cda.cit.tum.de/research/nanotech/

***Abstract*— The growing computational demands of artificial intelligence challenge the continued scaling of conventional CMOS technology, motivating the exploration of alternative paradigms such as *Atomic-Scale Computing*. Among potential implementations, *Field-Coupled Nanocomputing* (FCN) is a promising candidate that transmits information through electric or magnetic field interactions rather than current flow, enabling extremely dense circuits with potentially terahertz operating frequencies. However, FCN circuits are implemented on a single physical layer where logic gates and interconnects share the same structures. Consequently, signal crossings pose a major challenge, since proposed wire-crossing structures are not sufficiently reliable for physically realizable layouts. Previous work addresses this by performing planarization—i.e., eliminating crossings—during logic synthesis via node duplication to provide planar networks for placement and routing. However, this approach suffers from poor scalability and neglects necessary preprocessing steps required for valid FCN logic networks. In this work, we propose a planar FCN synthesis flow that improves the interaction between preprocessing steps for FCN logic networks and planarization. Furthermore, we introduce a hybrid crossing elimination strategy that resolves intersections through either node duplication or equivalent logical structures that emulate signal crossings, thereby bypassing the need for physical crossing gates. Experimental results demonstrate significant reductions in network size, achieving an average reduction of $34.65\,\%$, and up to $97.71\,\%$ for circuits where node duplication exhibits particularly poor scalability, enabling more compact planar circuits suitable for subsequent placement and routing.**

I. Introduction & Motivation

The rapidly growing computational demands of artificial intelligence challenge the continued scaling of conventional CMOS technology. As physical limits increasingly hinder further device miniaturization, alternative paradigms such as atomic-scale computing are being explored [1], [2].

Field-Coupled Nanocomputing (FCN) is a class of candidates in which information is transmitted through electric or magnetic field interactions rather than current flow [3]. Its nanoscale computational elements enable extremely dense circuits and potentially terahertz operating frequencies [4]. Among FCN technologies, *Silicon Dangling Bonds* (SiDBs) have recently emerged as a particularly promising platform, enabling ultra-dense circuits through atomically precise fabrication on silicon surfaces [5]–[7].

Benjamin Hien, Marcel Walter, and Robert Wille are with the Chair for Design Automation, Technical University of Munich, Germany. Marcel Walter, and Robert Wille are also with the Munich Quantum Software Company (MQSC) GmbH, Garching near Munich, Germany. E-mail: {benjamin.hien, marcel.walter, robert.wille}@tum.de

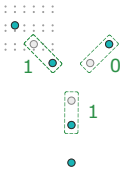
In contrast to CMOS, logic gates and interconnects in FCN circuits are realized by the same elementary structures on a single physical layer [3], [8]. Consequently, wires incur costs comparable to logic gates and cannot be routed across multiple metal layers as in conventional CMOS technologies. This constraint significantly alters the trade-offs in circuit synthesis and physical design [8], [9]. In particular, any signals that are supposed to cross each other on the final chip layout, require the insertion of dedicated crossing "gates" [10], [11].

However, not only do those structures add to the overall area cost of the layout, existing implementations for FCN technologies have been shown to be unreliable as they lead to signal instabilities that disrupt computation [12], [13]. As a result, physically realizable FCN circuits must avoid wire crossings entirely and adhere to strict planarity constraints. Recent works introduced planar placement and routing (P&R) algorithms capable of generating such layouts [14], [15]. These approaches, however, require planar logic networks as input. Consequently, wire crossings must be eliminated already during logic synthesis.

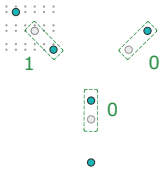
Prior work proposed a layer by layer node duplication strategy that guarantees planarity by duplicating nodes toward the primary inputs [10]. While effective, this approach suffers from poor scalability and can lead to exponential growth in network size and, as a direct consequence, in chip area. Furthermore, FCN logic networks require preprocessing steps, such as fanout substitution, which interacts unfavorably with node duplication and further amplifies structural overhead [15].

In this work, we propose a planar FCN synthesis flow with two key contributions. 1) We improve the interaction between necessary preprocessing steps, in particular fanout substitution, for valid FCN logic networks and node duplication. 2) We introduce a hybrid crossing elimination strategy that decides on a layer by layer basis whether crossings are resolved through node duplication or logical structures, which replace crossing dependencies by equivalent logic without requiring physical wire crossings, thereby avoiding costly duplication cascades when they become structurally inefficient. Experimental results demonstrate an average reduction in network size of $34.65\,\%$, and up to $97.71\,\%$ for circuits where node duplication exhibits particularly poor scalability, enabling the generation of significantly more compact planar circuits.

(a) An SiDB wire consisting of three BDL pairs and one single perturber on the left to propagate the binary `1`.



(b) OR gate for the input pattern `10` [5], [16].



(c) AND gate for the input pattern `10` [16].

Fig. 1: The SiDB logic platform.

The paper is structured as follows: Section II overviews FCN and the SiDB technology together with the graph-theoretic concepts relevant for planar circuits. Section III reviews related work on planar networks in the FCN domain. Section IV introduces the proposed design flow and hybrid crossing elimination strategy. Section V presents experimental evaluations demonstrating improvements in logic networks and resulting planar layouts. Finally, Section VI concludes the paper.

II. Background

This section reviews the technological foundations and structural constraints underlying planar FCN circuits. We first describe the SiDB platform, then discuss the physical limitations of wire crossing implementations, and finally introduce the graph-theoretic concepts required for planar circuit realization.

A. Silicon Dangling Bonds (SiDBs)

SiDBs can be fabricated on hydrogen-passivated silicon surfaces by selectively removing hydrogen atoms using the tip of a scanning tunneling microscope [6], [17]. Each SiDB can exhibit discrete charge configurations–negative, neutral, or positive–depending on bulk doping and electrostatic influences [4].

When placed in close proximity, SiDBs interact electrostatically [18], [19]. Pairs of closely spaced SiDBs can share a charge, where the charge position encodes binary information. These *binary-dot logic* (BDL) pairs therefore represent binary logic states [5], [16]. BDL pairs can propagate signals by arranging them in chains, where electrostatic interactions between neighboring pairs transmit information. Such wires can be initialized using a single dangling bond acting as a *perturber*, defining the input state (Fig. 1a) [5], [16].

Beyond wires, carefully arranged SiDB pairs can implement Boolean logic gates such as OR and AND through electrostatic interactions between inputs [5], [16]. To support automated design, these structures can be organized into standardized gate implementations with predefined input and output locations, forming standard-tile libraries for SiDB circuits [7], [20].

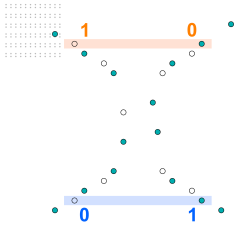


Fig. 2: Implementaiton of an SiDB crossing gate, which breaks at 21.78K.

B. Wire Crossings in FCN

Wire crossings in FCN technologies are fundamentally challenging due to their single-layer physical realization, preventing signals from crossing using vertical routing layers as in CMOS technologies. Although dedicated crossing structures have been proposed for SiDBs, they exhibit severe physical limitations. In particular, simulations show strong temperature sensitivity, with stable operation only up to $21.78\,\mathrm{K}$ [12], far below practical cooling methods such as liquid nitrogen ($77\,\mathrm{K}$). Consequently, such crossing structures are currently not suitable for practical computing applications.

Similar challenges have been reported for other FCN technologies such as *Quantum-dot Cellular Automata* (QCA), where both three-dimensional and coplanar crossing schemes have been explored [11]. However, fabrication constraints and severe signal degradation limit their practicality [13], [21].

Consequently, reliable wire crossings are currently not feasible in SiDB or other FCN technologies. To ensure physically realizable circuits, crossings must therefore be avoided entirely, requiring strictly planar physical designs without signal intersections.

C. Logic Networks and Planarity Constraints

Logic networks are an abstraction of circuits used in automated design, particularly during logic synthesis. A logic network is typically represented as a *Directed Acyclic Graph* (DAG), where each node corresponds to a Boolean function and edges represent signal dependencies. The sources of the graph are the *primary inputs* (PIs), while the sinks correspond to the *primary outputs* (POs). Nodes may have multiple *fanins* (nodes driving them) and *fanouts* (nodes they drive).

If there exists a path from node a to node b, then a belongs to the *transitive fanin* (TFI) of b. The transitive fanin cone of a node v, denoted $\mathrm{TFI}(v)$, contains v and all nodes on paths from the PIs to v, capturing the portion of the network that influences the value of v.

To obtain a structured representation of the DAG, nodes are organized according to their *depth*, defined as the length of the longest path from any PI to the node. Nodes with identical depth values form a *level*. Within each level, nodes are

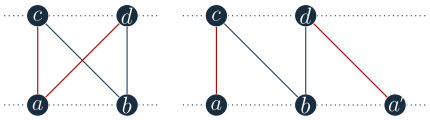


Fig. 3: The graph $K_{2,2}$ in a layered embedding. Left: layered embedding with $lvl_1 = \{a, b\}$ and $lvl_2 = \{c, d\}$ inducing a crossing. Right: layered planar embedding obtained by duplicating node a'.

assigned a *rank* that determines their horizontal placement. Together, levels and ranks define a *layered embedding* of the DAG, where edges connect nodes from lower to higher levels to ensure monotonic signal propagation. If such a layered embedding contains no edge crossings, it is referred to as a *planar embedding*.

Crossings in layered graphs are commonly reduced by reordering nodes within each level while keeping level assignments fixed. Tools such as *Graphviz* [22] employ heuristic methods for this purpose. However, minimizing crossings in layered graphs is NP-complete [22], and reordering alone cannot eliminate crossings if a crossing-free layered embedding does not exist.

Example 1. *As illustrated in Fig. 3, the graph $K_{2,2}$ admits a crossing-free embedding when nodes and edges can be placed freely in the plane. However, when nodes are restricted to fixed levels in a layered DAG, edges cannot be routed arbitrarily. The only remaining freedom is reordering nodes within each level. In this example, the only possible swaps are a with b and c with d, neither of which resolves the crossing.*

In FCN circuits, crossings in the layered graph correspond directly to wire crossings in the physical layout. Therefore, a *planar embedding* must be achieved during logic synthesis. Crossing minimization alone is insufficient; instead, the logic network must be structurally transformed until a crossing-free layered embedding exists, motivating dedicated planarization strategies.

III. Related Work

To address the need for planar logic networks, prior work proposed planarization strategies that modify the network structure to obtain level-planar embeddings. A representative approach for FCN logic networks was introduced in [10], which first applies path balancing and then eliminates crossings via layer-by-layer node duplication, propagating duplicates backward toward the primary inputs until all crossings are removed. While this guarantees planarity, it can introduce significant structural overhead, as duplicating a node typically requires replicating its entire TFI cone to preserve functional dependencies. Since duplications within the TFI cone must also be replicated, this process can cascade across multiple levels and lead to exponential growth in the number of nodes.

Example 2. *The duplication-based planarization principle is illustrated in Fig. 3. In the layered DAG representation of the graph $K_{2,2}$, a crossing occurs (bottom left). This crossing can be eliminated by duplicating node a, resulting in the planar embedding shown in the bottom right of Fig. 3.*

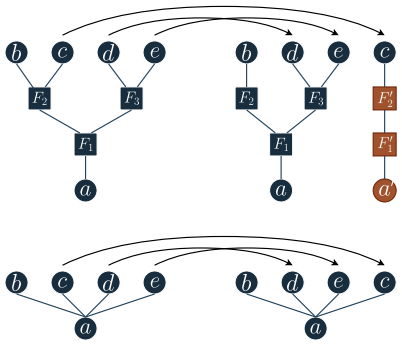


Fig. 4: Effect of node reordering on node duplication. Top Left: Fanout-substituted network. Top Right: Reordering the upper nodes during node duplication breaks the fanout tree structure, leading to unnecessary duplications. Bottom Left: Network without fanout substitution. Bottom Right: Reordering the upper nodes preserves structural dependencies and avoids redundant duplication.

However, networks obtained through duplication alone do not necessarily satisfy additional structural constraints required by FCN design flows. In particular, P&R algorithms typically require preprocessed networks, most notably *fanout-substituted* ones, where signal distribution is implemented through dedicated fanout structures rather than directly by logic nodes.

Example 3. *Consider again the right graph in Fig. 3. If node b represents a logic gate, the resulting network after duplication is not fanout substituted, since node b directly drives multiple successors. In practical FCN design flows, such signal distribution must instead be implemented using dedicated fanout nodes.*

More recently, the first scalable planar P&R algorithm for FCN circuits was introduced in [15]. In that flow, fanout substitution is first applied, followed by path balancing and node-duplication planarization, yielding logic networks that are compatible with planar P&R. This ordering is straightforward since fanout substitution requires subsequent path balancing, and node duplication itself requires a path-balanced network as input. However, performing fanout substitution before duplication can introduce additional structural overhead.

In particular, nodes belonging to the same fanout tree may be treated as independent nodes during node duplication. Consequently, structural dependencies within the fanout tree are not preserved, which can lead to redundant duplications.

Example 4. *This effect is illustrated in Fig. 4. In the fanout-substituted network (top left), signals are distributed through a fanout tree. When nodes are reordered during node duplication, the structural relationship between nodes in the fanout tree is lost (top right). As a result, nodes of the fanout*

tree may be duplicated independently, leading to unnecessary structural overhead.

Overall, existing planarization approaches rely exclusively on node duplication to eliminate crossings. The interaction and ordering of preprocessing steps—such as path balancing and fanout substitution—with duplication have not been systematically optimized, and alternative crossing elimination mechanisms have not been explored. As a result, scalable synthesis flows that produce compact planar and P&R-compatible FCN circuits remain an open problem.

IV. Proposed Approach

To address these limitations, we propose a synthesis flow for generating compact planar FCN circuits that is compatible with physical design constraints. The proposed approach ensures path balancing and fanout substitution while structurally eliminating all crossings. Furthermore, we introduce a hybrid crossing elimination strategy that selectively applies node duplication or dedicated logical crossing structures based on a cost-driven decision process.

A. Planar FCN Synthesis Flow

Based on the observations from the previous section, we modify the synthesis flow by postponing fanout substitution until after planarization. First, path balancing is performed to obtain the layered representation required for crossing elimination. Next, crossings are eliminated. Finally, fanout substitution is applied to obtain a network compatible with the P&R stage.

Although this requires an additional path balancing step after fanout substitution, this operation is computationally inexpensive. Since planarization is performed before these transformations, both fanout substitution and the subsequent balancing step must preserve the established planar embedding. Therefore, these procedures were adapted to maintain the planar embedding produced during planarization.

Example 5. *The benefit of postponing fanout substitution is illustrated in Fig. 4 (bottom). Without fanout substitution (bottom left), the structural dependency between nodes remains visible to the duplication algorithm. Consequently, node reordering preserves these dependencies and avoids redundant duplication (bottom right).*

By delaying fanout substitution, redundant duplication of fanout trees is avoided, resulting in significantly smaller planar networks.

B. Hybrid Crossing Elimination Strategy

Crossings are processed layer by layer in a top-down manner, starting from the primary outputs (POs) toward the primary inputs (PIs). For each processed layer, two alternative strategies are evaluated based on their estimated cost, namely the number of additional nodes that must be introduced into the network:

1) node duplication or 2) insertion of a dedicated logical crossing structure.

The strategy resulting in the smaller structural overhead is selected. Both approaches are discussed in the following.

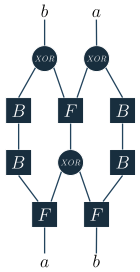


Fig. 5: Logical crossing structure used for local crossing resolution. The structure consists of three XOR gates together with fanout nodes (F) and buffers (B) to maintain signal propagation and fanout constraints.

1) Node Duplication: In the duplication strategy, the upper layer remains fixed while nodes in the lower layer are duplicated in order to remove crossings. When a node is duplicated, its fanin connections must also be replicated to preserve the functional dependencies of the circuit. As a result, duplication propagates toward lower layers and may continue until the primary inputs are reached.

To estimate this overhead, we approximate the duplication cost as

$$C_{\text{dup}} = \sum_{v \in D} (m_v - 1) \sum_{u \in \text{TFI}(v)} w(u),$$

where D denotes the set of nodes that must be duplicated, m_v is the number of copies created for node v, and $w(u)$ represents the structural weight of node u within the TFI cone.

The node weight is defined as

$$w(u) = \begin{cases} \frac{1}{2}, & \text{if } u \text{ is a buffer node,} \\ \alpha + \beta\, r^{\ell(u)}, & \text{otherwise,} \end{cases}$$

where $\ell(u)$ denotes the level of node u and $r > 1$ models the exponential growth of duplication cost with increasing depth. The constants α, β, and r are fixed parameters that were determined empirically through preliminary experiments. The constants α and β capture the base structural cost of duplicating logic nodes. Buffer nodes are assigned a lower weight since many buffers introduced during duplication are later removed when path balancing is applied again after fanout substitution.

2) Crossing Structure Insertion: Alternatively, crossings can be resolved locally by inserting a dedicated logical crossing structure. This structure does not represent a physical wire crossing; instead, it replaces the crossing dependency by an equivalent logic transformation. Each such structure consists of three XOR gates together with dedicated fanout nodes and buffering elements, effectively emulating the behavior of a signal crossing by appropriately recombining the input signals at the outputs. The resulting structure is illustrated in Fig. 5. In addition, signals that do not participate in the crossing may require additional buffering to preserve path balancing.

Example 6. *Considering the left graph shown in Fig. 3, the crossing can be resolved by inserting a crossing structure for the edges* (a, d) *and* (b, c)*. The remaining edges* (a, c) *and* (b, d) *then require additional buffering to maintain balanced signal paths.*

Before evaluating the number of crossings to be replaced, a lightweight min-cross heuristic [22] performs adjacent swaps within the lower layer in order to reduce the number of crossings.

The cost of resolving crossings through logical crossing structures insertion is estimated as

$$C_{\text{cross}} = N_{\text{cross}} \cdot C_{\text{structure}} + N_{\text{buffer}},$$

where N_{cross} denotes the number of remaining crossings after reordering, $C_{\text{structure}}$ represents the cost of one crossing structure, and N_{buffer} accounts for additional buffers required for path balancing.

For each layer, both strategies are evaluated and the option with the lower estimated cost is selected. The effectiveness of the resulting synthesis flow is evaluated experimentally in the following section.

V. Experiments

This section presents the experimental evaluation of the proposed planar synthesis flow on logic networks from established benchmark suites [23], [24]. The benchmarks cover circuits of varying sizes and structural characteristics. Experiments were conducted on an AMD Ryzen 7 PRO 6850U with 32 GB DDR5 RAM. To support reproducibility and open research, the algorithm is made publicly available as part of the *Munich Nanotech Toolkit* (MNT) at `https://github.com/cda-tum/fiction`.

In the following, we compare three variants of the planar synthesis flow. The first corresponds to the state-of-the-art approach (SoTA). The remaining two variants represent the contributions of this work, which are evaluated separately to analyze their individual impact: a reordered synthesis flow that postpones fanout substitution until after node duplication (FO), and the proposed hybrid crossing elimination strategy (Hybrid). This separation allows us to quantify the effect of both contributions independently. The comparison evaluates the resulting *network size* and *runtime*. The results are summarized in Table I.

The table also reports the preprocessing stage of each flow. In particular, path balancing and fanout substitution introduce additional nodes, while fanout substitution also affects the depth of the resulting network. The path-balanced networks serve as input for the flow proposed in this work, whereas the fanout-substituted and balanced networks correspond to the input required by the SoTA.

Across all benchmarks, postponing fanout substitution consistently reduces the number of nodes compared to the SoTA, as reflected by the comparison of FO and SoTA. On average, the reordered flow achieves a reduction of $20.80\,\%$ in network size. This improvement highlights that performing fanout substitution *after* planarization avoids unnecessary duplication of fanout structures and therefore leads to *substantially smaller networks*. The runtime for smaller benchmarks remains below $0.05\,\text{s}$, while for larger benchmarks the runtime is significantly reduced, ranging from $13.39\,\%$ to $50.00\,\%$ compared to the SoTA.

The benefits of the hybrid crossing elimination strategy depend on the circuit size and crossing density. For smaller benchmarks, duplication alone is often sufficient to eliminate crossings, and inserting additional crossing structures introduces unnecessary overhead. Consequently, the hybrid strategy does not provide additional benefits over the reordered flow for these small circuits. However, this does not pose a practical limitation, since the resulting planar networks are already compact.

As the benchmark size grows, the hybrid strategy becomes increasingly beneficial. The isolated effect of the hybrid crossing insertion can be observed in the comparison of Hybrid and FO. For example, in *xor5Maj* and *parity*, the hybrid approach achieves additional reductions of $31.42\,\%$ and $34.55\,\%$ on top of the reductions obtained by the reordered flow. This results in total reductions of $58.81\,\%$ and $39.22\,\%$ compared to the SoTA approach.

For large benchmarks, the poor scalability of duplication alone becomes increasingly apparent, making the hybrid strategy particularly beneficial. In these situations, selectively inserting crossing structures can significantly reduce the number of required duplications. A particularly extreme example is benchmark *c432*, where the node count is reduced from $802\,828$ to $18\,353$, corresponding to a reduction of $97.71\,\%$ when comparing Hybrid to the SoTA approach. Since far fewer nodes need to be duplicated, the algorithm terminates much faster, and the runtime decreases by $99.61\,\%$. Other large circuits also benefit significantly from the hybrid strategy, with reductions of up to approximately $80\,\%$, demonstrating that the approach scales substantially better once duplication alone becomes inefficient. For these benchmarks, some outliers appear in the relative runtime measurements. However, in absolute terms the additional overhead introduced by the cost evaluation remains negligible.

At the same time, the proposed hybrid strategy is effective only within a certain range of circuit sizes and crossing densities. When the number of crossings becomes extremely large already during the first elimination steps, both duplication and crossing-structure insertion scale poorly. In such cases, the hybrid strategy cannot provide additional benefits. This behavior can be observed for *c880*, where no improvement over the reordered flow (FO) is obtained.

Moreover, for the five largest benchmarks of the ISCAS85 suite—*c2670*, *c3540*, *c5315*, *c6288*, and *c7552*—neither SoTA, FO, nor Hybrid can produce planar networks, as all approaches eventually run out of memory due to excessive structural growth. These benchmarks are therefore omitted from Table I. This behavior highlights a limitation of the current planarization approach and may indicate a more fundamental scalability boundary for strictly planar FCN logic networks.

TABLE I: Comparison of the baseline SOTA, the reordered fanout-substitution flow (FO), and the proposed hybrid approach (HYBRID) in terms of node count and runtime. Percentage values denote relative differences. Negative values (green) indicate improvements, i.e., reductions in node count or runtime, while positive values (red) indicate degradations.

	BENCHMARK CIRCUIT				PREPROCESSING				SOTA		FO VS. SOTA				HYBRID VS. SOTA				HYBRID VS. FO	
	Benchmark	I	O	$\lvert N\rvert$	N_{blc}	D_{blc}	$N_{\text{bfo+blc}}$	$D_{\text{bfo+blc}}$	N_{sota}	t_{sota}	N_{fo}	t_{fo}	$\Delta N[\%]$	$\Delta t[\%]$	N_{hyb}	t_{hyb}	$\Delta N[\%]$	$\Delta t[\%]$	$\Delta N[\%]$	$\Delta t[\%]$
Fontes18 [23]	xor	2	1	9	11	3	13	4	15	0.00	14	0.00	-6.67	0.00	14	0.00	-6.67	0.00	0.00	0.00
	1bitAdderAOIG	3	2	20	36	8	42	10	52	0.00	51	0.00	-1.92	0.00	51	0.00	-1.92	0.00	0.00	0.00
	t	5	2	18	23	4	31	6	39	0.00	28	0.00	-28.21	0.00	28	0.00	-28.21	0.00	0.00	0.00
	t_5	5	2	18	23	4	31	6	39	0.00	28	0.00	-28.21	0.00	28	0.00	-28.21	0.00	0.00	0.00
	c17	5	2	15	21	4	33	7	35	0.00	29	0.00	-17.14	0.00	29	0.00	-17.14	0.00	0.00	0.00
	b1_r2	3	4	17	28	4	35	6	46	0.00	45	0.00	-2.17	0.00	45	0.00	-2.17	0.00	0.00	0.00
	majority	5	1	24	43	9	49	10	55	0.00	47	0.00	-14.55	0.00	47	0.00	-14.55	0.00	0.00	0.00
	majority_5_r1	5	1	24	31	6	41	8	53	0.00	37	0.00	-30.19	0.00	37	0.00	-30.19	0.00	0.00	0.00
	newtag	8	1	27	43	8	48	9	52	0.00	45	0.00	-13.46	0.00	45	0.00	-13.46	0.00	0.00	0.00
	clpl	11	5	23	88	10	118	14	216	0.00	165	0.00	-23.61	0.00	165	0.00	-23.61	0.00	0.00	0.00
	xor5_r1	5	1	33	55	11	67	14	89	0.00	86	0.00	-3.37	0.00	86	0.00	-3.37	0.00	0.00	0.00
	1bitAdderMaj	3	1	34	71	14	96	19	208	0.00	166	0.00	-20.19	0.00	137	0.00	-34.13	0.00	-17.47	0.00
	cm82a_5	5	3	49	84	9	111	13	190	0.00	140	0.00	-26.32	0.00	140	0.00	-26.32	0.00	0.00	0.00
	2bitAdderMaj	5	2	61	116	15	175	23	555	0.00	329	0.00	-40.72	0.00	323	0.00	-41.80	0.00	-1.82	0.00
	xor5Maj	5	1	77	165	20	227	28	1277	0.00	767	0.00	-39.94	0.00	526	0.00	-58.81	0.00	-31.42	0.00
	parity	16	1	121	151	15	181	19	561	0.00	521	0.00	-7.13	0.00	341	0.00	-39.22	0.00	-34.55	0.00
ISCAS85 [24]	c17	5	2	15	21	4	33	7	35	0.00	29	0.00	-17.14	0.00	29	0.00	-17.14	0.00	0.00	0.00
	c432	36	7	452	2934	58	3996	78	802 828	26.59	614 317	23.03	-23.48	-13.39	18 353	0.09	-97.71	-99.66	-97.01	-99.61
	c499	41	32	859	3589	33	5262	51	271 465	3.17	207 720	2.21	-23.48	-30.28	64 686	2.98	-76.17	-5.99	-68.86	34.84
	c880	60	26	701	3490	39	4907	55	42 442	0.10	24 221	0.05	-42.93	-50.00	24 221	0.38	-42.93	280.00	0.00	660.00
	c1355	41	32	1107	4829	43	7214	67	208 421	0.81	174 946	0.66	-16.06	-18.52	40 036	0.58	-80.79	-28.40	-77.12	-12.12
	c1908	33	25	848	5339	53	7634	78	488 732	7.23	338 969	4.32	-30.64	-40.25	109 066	5.26	-77.68	-27.25	-67.82	21.76
Average													**-20.80**				**-34.65**		**-18.00**	

For each benchmark, I and O denote the number of primary inputs and outputs. $|N|$ is the initial node count. N_{blc} and D_{blc} denote the node count and depth after balancing, while $N_{\text{bfo+blc}}$ and $D_{\text{bfo+blc}}$ correspond to buffer removal followed by balancing. For the evaluated flows, N denotes the final node count and t the runtime in seconds. All relative differences are computed as $\Delta x = (x_{\text{new}} - x_{\text{ref}})/x_{\text{ref}}$, such that negative values indicate improvements. Runtimes below $0.005\,\text{s}$ are reported as $0.00\,\text{s}$.

For all benchmarks where planar networks are obtained, HYBRID reduces network size by $34.65\,\%$ on average over SOTA and by an additional $18.00\,\%$ over FO.

VI. CONCLUSION AND FUTURE WORK

This work presented a planar synthesis flow that combines node duplication with selective insertion of logical crossing structures to eliminate wire crossings in logic networks. Compared to the state-of-the-art planarization approach (SOTA), the proposed method improves the interaction between preprocessing and planarization and introduces a hybrid crossing elimination strategy.

Experimental results show that postponing fanout substitution already reduces network size by an average of $20.80\,\%$ compared to SOTA. The hybrid strategy further improves scalability, reducing network size by $34.65\,\%$ on average and by an additional $18.00\,\%$ compared to the reordered flow, with reductions of up to $97.71\,\%$ for circuits where duplication scales poorly. Although extremely dense crossing patterns remain challenging, the approach enables substantially more compact planar networks across a wide range of benchmarks.

Future work will investigate additional planarization techniques, such as decomposition methods and precomputed planar structures, to further reduce duplication overhead and improve scalability for very large circuits.

REFERENCES

[1] H. Esmaeilzadeh, E. Blem, R. S. Amant *et al.*, "Dark Silicon and the End of Multicore Scaling," in *International Symposium on Computer Architecture (ISCA)*. IEEE, 2011, pp. 365–376.

[2] M. B. Taylor, "A Landscape of the New Dark Silicon Design Regime," *IEEE Micro*, vol. 33, no. 5, pp. 8–19, 2013.

[3] N. G. Anderson *et al.*, *Field-coupled Nanocomputing: Paradigms, Progress, and Perspectives*. New York: Springer, 2014.

[4] L. Livadaru, P. Xue, Z. Shaterzadeh-Yazdi *et al.*, "Dangling-bond Charge Qubit on a Silicon Surface," *New J. Phys.*, vol. 12, no. 8, p. 083018, 2010.

[5] T. Huff, H. Labidi, M. Rashidi *et al.*, "Binary Atomic Silicon Logic," *Nat. Electron.*, vol. 1, no. 12, pp. 636–643, 2018.

[6] J. Pitters, J. Croshaw, R. Achal *et al.*, "Atomically Precise Manufacturing of Silicon Electronics," *ACS Nano*, 2024.

[7] M. Walter, S. S. H. Ng, K. Walus *et al.*, "Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, 2022.

[8] D. A. Reis, C. A. T. Campos, T. R. B. S. Soares *et al.*, "A Methodology for Standard Cell Design for QCA," in *ISCAS*, 2016, pp. 2114–2117.

[9] S. Hofmann, M. Walter, and R. Wille, "Late Breaking Results: Wiring Reduction for Field-coupled Nanotechnologies," in *DAC*, 2024.

[10] A. Chaudhary, D. Z. Chen, X. S. Hu *et al.*, "Fabricatable Interconnect and Molecular QCA Circuits," *TCAD*, vol. 26, no. 11, pp. 1978–1991, 2007.

[11] P. D. Tougaw and C. S. Lent, "Logical devices implemented using Quantum Cellular Automata," *J. Appl. Phys.*, vol. 75, no. 3, pp. 1818–1825, 1994.

[12] J. Drewniok, M. Walter, and R. Wille, "Temperature Behavior of Silicon Dangling Bond Logic," in *IEEE-NANO*, 2023, pp. 925–930.

[13] J. Retallick, M. Babcock, M. Aroca-Ouellette *et al.*, "Algorithms for Embedding Quantum-Dot Cellular Automata Networks onto a Quantum Annealing Processor," 2017.

[14] M. Walter, R. Wille, F. Sill Torres *et al.*, "Scalable Design for Field-coupled Nanocomputing Circuits," in *ASP-DAC*. ACM New York, NY, USA, 2019, pp. 197–202.

[15] B. Hien, M. Walter, S. Hofmann *et al.*, "A fully planar approach to field-coupled nanocomputing: Scalable placement and routing without wire crossings," in *IEEE-NANO*, 2025.

[16] S. S. H. Ng, J. Retallick, H. N. Chiu *et al.*, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *TNANO*, vol. 19, pp. 137–146, 2020.

[17] M. B. Haider, J. L. Pitters, G. A. DiLabio *et al.*, "Controlled Coupling and Occupation of Silicon Atomic Quantum Dots at Room Temperature," *PRL*, vol. 102, no. 4, p. 046805, 2009.

[18] J. L. Pitters, L. Livadaru, M. B. Haider *et al.*, "Tunnel coupled dangling bond structures on hydrogen terminated silicon surfaces," *The Journal of Chemical Physics*, vol. 134, no. 6, p. 064712, 2011.

[19] T. R. Huff, T. Dienel, M. Rashidi *et al.*, "Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot," *ACS Nano*, vol. 13, no. 9, pp. 10 566–10 575, 2019.

[20] M. D. Vieira, S. S. H. Ng, M. Walter *et al.*, "Three-Input NPN Class Gate Library for Atomic Silicon Quantum Dots," *IEEE Design & Test*, 2022.

[21] R. A. Wolkow, L. Livadaru, J. Pitters *et al.*, *Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics*. Springer, 2014, pp. 33–58.

[22] E. R. Gansner, E. Koutsofios, S. C. North *et al.*, "A technique for drawing directed graphs," in *IEEE Symposium on Visual Languages*. IEEE, 1993, pp. 152–160.

[23] G. Fontes, P. A. R. L. Silva, J. A. M. Nacif *et al.*, "Placement and Routing by Overlapping and Merging QCA Gates," in *ISCAS*, 2018.

[24] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *International Symposium on Circuits and Systems (ISCAS)*. IEEE Press, 1985, pp. 677–692.