Holistic Physical Design for Silicon Atomic Logic

Cramming More Components onto Clock Zones

Willem Lambooy*, Jan Drewniok†, Marcel Walter*‡, and Robert Wille*‡§

*Chair for Design Automation, Technical University of Munich, Germany
†Chair for AI Processor Design, Technical University of Munich, Germany
‡Munich Quantum Software Company GmbH, Garching near Munich, Germany
§Software Competence Center Hagenberg GmbH, Austria
Email: {willem.lambooy, jan.drewniok, marcel.walter, robert.wille}@tum.de
https://www.cda.cit.tum.de/research/nanotech/

***Abstract*—*Field-Coupled Nanocomputing* (FCN) presents an exciting foundation for post-CMOS computing, and among its variants, *Silicon Dangling Bond* (SiDB) logic shows particular promise. Building on this technology, recent work has established an end-to-end design flow that synthesizes SiDB circuits directly from logic-level specifications. These circuits rely on an FCN clocking scheme to ensure directional and stable computation across sequences of logic gates. This clocking mechanism is indispensable: it actively nullifies parasitic interactions—or *crosstalk*—that arise when multiple gates are placed in close proximity, a fundamental concern in FCN-based circuits. However, a curious mismatch arises: while logic gates in SiDB can be implemented at the atomic scale, the metal pitches used to generate the clock signal remain relatively coarse. As a result, the nanoscopic SiDB logic gates must be artificially enlarged to fit one gate per clock zone, simply to interface with the clocking infrastructure. This approach, though functional, squanders the incredible density potential of atom-scale SiDB logic. To unlock that potential, we advocate for a holistic gate design methodology: crafting entire multi-gate assemblies that are small enough to fit into a single clock zone, while simultaneously mastering the delicate crosstalk interactions that such compactness entails. Yet, this compact design challenge gives rise to a monstrous design space. The number of possible combinations grows exponentially in the number of gates, with the number of distinct gate implementations as base—of which there are many thousands. Each candidate solution must be evaluated using physical simulation across all input combinations, which itself scales exponentially with the number of SiDBs in the design. As a result, design automation efforts to date have favored simpler, more sparse circuit topologies to preserve tractability—at the expense of logic density. In this work, we confront the challenge head-on. With an experimental evaluation, it is shown that the proposed methodology enables cramming significantly more logic into each clock zone of an SiDB circuit—advancing toward a truly holistic physical design flow for atomic-scale logic.**

I. Introduction

The atomic precision achievable in *Silicon Dangling Bond* (SiDB) logic offers a tantalizing opportunity to realize ultra-dense, energy-efficient computation at the limits of physical scaling [1]–[6]. As an instantiation of *Field-Coupled Nanocomputing* (FCN), SiDB circuits operate through the Coulombic repulsion of electrons, rather than through electric current [7], [8]. This alternative computing paradigm eliminates resistive losses and enables the use of metastable charge configurations for logic encoding [7].

Recent progress has laid the groundwork for synthesizing such circuits end-to-end. From a high-level Boolean logic specification, automated tools now exist to generate SiDB layouts that function correctly under the FCN computing model [9]–[16]. A key component of this model is *clocking*: a global, zone-based electrical signal that sequentially activates circuit regions, enforcing directed signal propagation and suppressing unintended interactions between nearby logic gates [17], [18].

However, while the gates themselves can be constructed using just a few atoms, the metal interconnects that drive the clock signal are at least an order of magnitude larger [1], [11], [19]. As a result, the state-of-the-art scales up individual gates in the layout through elongating their I/O wires, so that each occupies an entire clock zone [11]. This ensures correct operation but comes at the cost of wasting space that could otherwise accommodate additional logic.

This compromise is unfortunate, as the ability to place gates densely is a major appeal of SiDB logic [1]. To fully exploit this, we propose a holistic approach: designing not just isolated gates but small gate assemblies, placed tightly together within a single clock zone. Achieving this demands a fine-grained understanding and control of the crosstalk that such compactness naturally introduces.

Designing such compact, multi-gate assemblies is fundamentally challenging. The design space grows exponentially with the number of gates and candidate implementations considered. Even when severely restricting gate design parameters to limit the number of possible implementations, the combinatorial growth is rapid. What's more, rigorously verifying the functional correctness of any given assembly requires exhaustive physical simulation across all input combinations, which itself scales exponentially in the number of SiDBs in base 3 [20], [21].

These scaling challenges have led most existing approaches to prioritize design tractability over spatial density, resulting in sparse and conservative circuit implementations. In this work, we take a *holistic* approach to physical design, addressing the challenge of compact multi-gate assemblies head-on through a sophisticated gate *co*-designing process. We introduce a methodology that achieves dense, functionally correct gate groupings within a single clock zone, substantially improving both spatial efficiency and logical throughput. Our results show that holistic physical design enables significantly tighter packing of logic gates without compromising correctness, paving the way for more practical and scalable silicon atomic logic.

First, the obligatory background on SiDBs, and FCN clocking is given in Section II, in which the concept of gate assemblies is also introduced. Then, in Section III, a discussion of the considered problem of holistic synthesis follows, illustrating how massive the search space for this problem really is. The main contribution of this work is given in Section IV, which details the quintessential techniques for achieving tractable holistic synthesis. These ideas were incorporated into an implementation with which the proposed methodology is experimentally evaluation, presented in Section V, which shows that practical runtimes were achieved for various NPN classes—empirically proving that already up to 5 gates can co-exists in a single clock zone without complications. Finally, Section VI concludes the paper.

II. Background

In this section, we dive into the logical platform that underlies this work, outlining the principles that give rise to stable binary computation in Section II-A and Section II-B. Following this, we discuss the relevant state-of-the-art in design automation for this technology in Section II-C, and touch on physical simulation in Section II-D, which is of paramount importance to this work.

A. SiDB Circuits and Field-Coupled Nanocomputing

Silicon Dangling Bond (SiDB) logic is a promising technology within the broader paradigm of *Field-Coupled Nanocomputing* (FCN) [2], [7]. In FCN, information is encoded in the spatial arrangement of electrons rather than in charge transport [22]. Specifically for SiDBs, unpassivated silicon atoms on a hydrogen-terminated Si(100) surface exhibit one of three charge states associated with the number of electrons trapped at the dangling bond [19], [23]. Placing two SiDBs in certain proximity forms a *Binary-dot Logic* (BDL) pair, which tends to exchange one electron shared between the two, such that the associated two charge configurations of the pair may be regarded as respective binary logic states. A BDL *wire* emerges when multiple such pairs are chained with directional consistency, making the individual BDL pairs orient synchronously due to field-coupling, as seen in Fig. 1 [1].

Computation proceeds by Coulomb interaction between these localized charges [8]. Due to the strength and range of these interactions, individual logic gates may be constructed using only a handful of SiDBs [1]. Layouts are realized with atomic precision via hydrogen lithography techniques [6], [24]. However, this same coupling strength gives rise to parasitic interference (or *crosstalk*) between spatially proximate gates, posing a challenge to compositional logic design [20], [21].

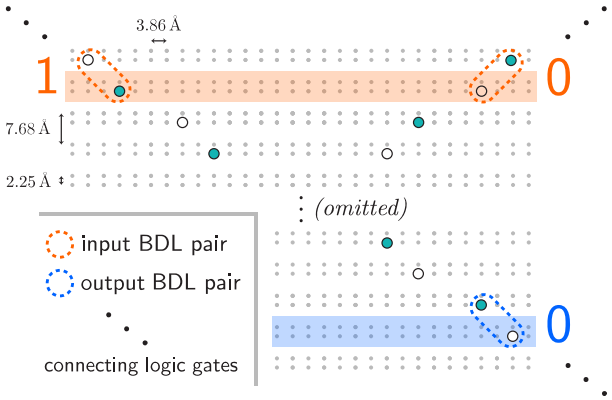

Fig. 1: Computation on the SiDB logic platform using BDL wires. This characteristic 2-input-1-output gate shape is tiled hexagonally, and used to implement Boolean functions throughout this work.

B. FCN Clocking Scheme

To mitigate crosstalk and enforce signal directionality, SiDB circuits rely on a spatially phased four-level clocking scheme [11]. The surface is partitioned into sequential *clock zones*, each controlled by an underlying metal line that modulates the electrostatic potential locally.

Each clock cycle advances the activation of these zones in a pipelined fashion. When a zone is in the *active* phase, its SiDBs can switch states in response to neighboring zones; when in the *locked* or *relaxed* phase, charge transitions are suppressed, preserving signal stability and directionality [11]. Correct computation across multi-gate circuits depends on carefully aligning gate placement and signal flow with this clocking infrastructure.

C. SiDB Gate Assemblies

In the *Bestagon* framework [11], which standardized the placement of SiDB gates in the plane to follow a hexagonal tiling, each logic gate is extended (e.g., via long input/output wires) to fully occupy a dedicated clock zone: a *supertile*. In Fig. 2, it is shown how a half-adder can be implemented in this framework. While this technique for reliable circuit synthesis simplifies crosstalk management and ensures compatibility with the clocking scheme, it sacrifices area efficiency—especially given that the actual gate logic may only span across a few atoms—let alone throughput: clock zones are consumed eagerly.

To overcome this, we consider *gate assemblies*: collections of gates placed densely within such a single clock zone supertile. This allows for much more compact logic, but demands holistic design to account for all inter-gate interactions. The design and validation of such assemblies form the central focus of this work.

D. Simulation-Based Validation

The physical correctness of a SiDB layout is determined by its response to all signal-representing charge configurations of the input BDL pairs [25]. This is evaluated using a physics-based simulator that computes the electrostatic energy landscape of the system and determines the ground-state configuration for each input [8], [20], [21], [26]. This reveals charge configurations for BDL pairs marked as respective outputs, which are then interpreted as binary values.

The simulation accounts for all pairwise Coulomb interactions between SiDBs, adjusted for screening and substrate effects. Given the exponential number of input combinations and—more drastically even—the exponential number of possible SiDB charge distributions in the number of SiDBs considered, simulation becomes a computational bottleneck—especially as assemblies grow in size and complexity [20], [21].

III. The Daunting Problem of Holistic Synthesis

Having outlined the physical and logical foundations of SiDB-based computation, as well as the constraints imposed by FCN clocking and validation, we now turn to the core challenge addressed in this work: synthesizing entire gate assemblies within a single clock zone. In this section, we examine why this task—unlike traditional gate-level design—demands a fundamentally different, holistic approach. We begin by contrasting isolated gate synthesis with the assembly of multiple gates in Section III-A, explore the explosive combinatorial complexity that emerges in Section III-B, formally define the synthesis problem in Section III-C, and conclude with a summary of key design objectives and constraints in Section III-D.

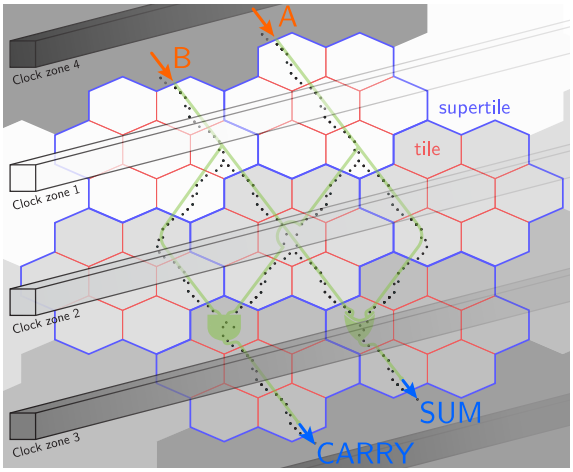

Fig. 2: A half-adder implemented by the state of the art [11], separating the critical logic to different supertiles. Extending the wiring and employing multiple clock zones avoids crosstalk where sensitive computations are carried out—sacrificing throughput and layout footprint.

A. From Individual Gates to Holistic Assemblies

Existing SiDB logic design methodologies have primarily focused on the synthesis and simulation of individual gates in isolation [10], [13], [16], [27]. In these approaches, each logic gate is treated as an atomic unit, extended spatially to fill an entire clock zone. This ensures minimal crosstalk and predictable behavior under the field-coupled nanocomputing clocking scheme. While this simplifies validation and supports modular composition, it fails to exploit the ultra-high density potential of atom-scale logic.

Our objective is to design *gate assemblies*—compact, multi-gate logic blocks that fit entirely within a single clock zone. This allows us to bypass the need for artificial spacing and wiring overhead, paving the way toward significantly higher logic densities. However, doing so demands a departure from traditional gate-level abstraction. Instead, we must treat assemblies as holistic physical entities, where computation emerges from carefully orchestrated inter-gate coupling.

Since the design task we consider in this work is scoped exclusively at the *co*-designing of gates in a supertile, for convenience of communication, we henceforth use the broader term *"circuit"* to refer to a gate assembly in a single clock zone (unless explicitly stated). Furthermore, in Section IV, we often make use of the term *sub*-circuit where a circuit is considered: this intends a gate assembly that is contained within the circuit.

B. Combinatorial Design Space Explosion

The move from individual gates to assemblies introduces a daunting design challenge. Even if each gate can be implemented in a few thousand distinct, validated SiDB layouts, the number of possible combinations grows exponentially with the number of gates to be assembled. Moreover, the nonlocal nature of Coulomb interactions makes it impossible to reason about assemblies compositionally: interactions between gates can disrupt or enhance behavior in non-obvious ways.

Validation of each candidate assembly requires exact physical simulation across all 2^n input combinations (for n primary inputs), and for each such input, finding the charge ground state among exponentially many SiDB charge configurations. Consequently, brute-force enumeration and simulation of all layout combinations becomes infeasible, even for modestly sized assemblies.

Example 1. *To exemplify how the design space explodes—even under several restrictions to attenuate it—consider the task of designing the circuit laid out in Fig. 3, involving* 7 *gates connected through predefined wiring. When limited to* 4 *SiDBs placed per gate—a solid choice, according to [27]—together with the wiring as seen in Fig. 3, there are* $7 \cdot (6 + 4) = 70$ *SiDBs to simulate for each of the* $2^2 = 4$ *input combinations that is to be evaluated. With the possible SiDB placements*

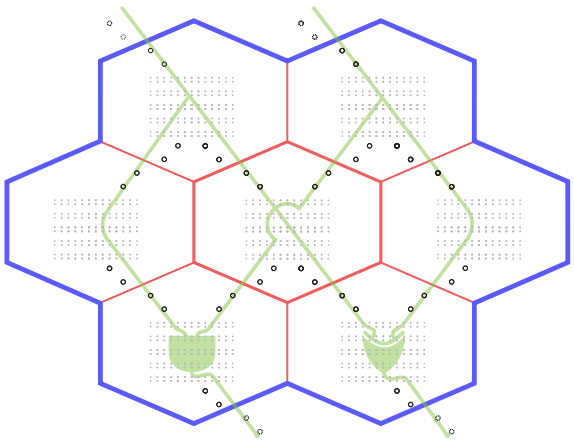

Fig. 3: Contrasting the inefficient half-adder implementation seen in Fig. 2, the same circuit could theoretically fit in a single clock zone.

for the gate design limited to the 130 *lattice positions shown at each tile, the number of considered gate implementations is* $\binom{130}{4}$*, i. e., over* 11 *million. Thus, evaluating all combinations of gate implementations for all input combinations amounts to a whopping* $\binom{130}{4}^7 \cdot 4 \approx 9.76 \cdot 10^{49}$ *exact simulation instances that each need to consider* $3^{70} \approx 2.50 \cdot 10^{23}$ *possible charge distributions.*

Alternatively, one may first reduce the number gate implementations for each individual gate through validating each candidate in separation, which requires simulation tasks that each consider 3^{6+4} *possible charge distributions—modern exact simulators do so in a whim [20], [21]. Nevertheless, depending on physical parameters passed to the simulation engine, there easily remain many thousands of validated candidates, and thus, still unspeakably many expensive exact simulation instances are in order for exhaustive enumeration of the remaining combinations—until one is found that implements the intended logic successfully under all input combinations.*

C. Formal Problem Definition

Let $\mathcal{C}$ denote a logic circuit specified as a directed acyclic graph (DAG), where nodes represent logic functions—i. e., the gates to implement—and edges represent signal flow. Each gate $g \in \mathcal{C}$ is associated with a Boolean function of arity n, and a set of physically realizable gate implementations $\mathcal{G}_g$ that may implement this function under certain electrostatic conditions.

We aim to determine a mapping $\phi : \prod_{g \in \mathcal{C}} \mathcal{G}_g$ assigning each gate a specific implementation. Together with a pre-defined structure of BDL wires that from the edges between the gates, this must render an implementation of the composite circuit $\mathcal{C}$, which operates under a shared electric field, and faithfully implements the specified logic behavior across all input combinations.

Crucially, ϕ must be chosen such that the electrostatic interactions among the selected implementations do not disrupt logical functionality—a property which cannot be determined in isolation but only by simulation of the resulting configuration.

D. Design Objectives and Challenges

Given the formulation above, the following objectives guide our design approach:

- **Correctness**: All ground-state charge configurations under all valid input combinations must match the target logic behavior at every gate output.
- **Scalability**: The number of full-circuit physical simulations—i. e., simulations that consider all SiDBs in a fully implemented circuit as determined by some $\phi : \prod_{g \in \mathcal{C}} \mathcal{G}_g$—must be minimized to maintain tractability, making the method applicable beyond toy examples, enabling the synthesis of useful logic structures spanning multiple gates.

To address these challenges, we propose a context-aware, simulation-driven design methodology described in Section IV. This approach avoids exhaustive enumeration until the design space is reduced sufficiently, using localized physical simulations with informed environmental contexts to guide a stochastic pruning of the design space.

IV. Novel Methodologies for Design of SiDB Circuits

With the problem of holistic circuit design described in its different facets in the previous section, this section lays out the proposed methodology to achieve said goals—as demonstrated via an experimental evaluation in Section V. To support the ideas conveyed in this section through illustration, Fig. 4 provides a visual overview. Following a description of the general idea of the proposed work in Section IV-A, the remainder of this section is divided into two parts. First, in Section IV-B, we detail the measures taken to improve the accuracy of physical simulations of components in a larger system controlled by a single clock zone. Then, in Section IV-C, an overview is given of the procedure in which the design space is iteratively reduced until a solution—i. e., a mapping ϕ, see Section III-C—can be found, while requiring a minimal amount of time-intensive simulations.

A. General Idea

A high-level description of the ideas presented in this work is given here, capturing the essence of the two parts of this section that follow. In the previous sections, it was laid out that field-coupled technologies like SiDB exhibit subtle, yet functionally significant crosstalk between gates in a shared clock zone. Instead of neglecting this effect upon which any FCN implementation is built, in the objective to form logic gate assemblies that operate according to a specification, we propose a way of designing gates using rapid, small-scale physical simulations, while incorporating known information of the electrostatic context induced by surrounding gates that share the electric field.

To enable these small-scale simulations to be used toward the larger goal of circuit design, a procedure is proposed in which many smaller simulations are used to discriminate candidate gate implementations for each gate in the circuit. For the respective gates, this enables the set of candidate gate implemenations to be pruned, removing those that are deemed *least* compatible fits in the circuit. To determine such compatibility, we propose an empirical strategy that ranks candidates through a number of randomly sampled evaluations, in which a candidate for the one gate is simulated together with various remaining candidate implementations for another gate.

Iterating on this pruning process yields a significant reduction of the design space, arriving at sets of candidate gate implementations for the respective gates that are *mutually compatible*—as judged by simulated evaluations of gate pairs—i. e., each such simulation finds that the associated logic function of the respective pair was perfectly realized. The sets of candidates are minimized further through repeating the process with increasingly more extensive simulations: considering gate triplets, and so forth.

Finally, the point is reached where all gates in the circuit are considered for the physical simulations. It is at this point that it becomes possible to encounter an assignment of gate implementations to the different gates in the circuit that is operational, completing the circuit design task successfully. All the while, the pruning up to this point ensures *efficacious* consideration of the humongous design space: minimizing the number of time-intensive physical simulations that are required to validate any possible solution.

B. Context-Aware Simulation and Logic Assessment of Sub-Circuits

In this part, we go in depth on the techniques devised to perform simulations of gate assemblies (sub-circuits) that from a part of a larger logic gate assembly: i. e., the circuit we wish to design. We enrich physical simulation of sub-circuits with static potential offsets that amount from the electrostatic influence accumulated from supposedly functioning BDL wires in the surrounding circuit context.[1]

For the purpose of circuit generation, this work relies on exact physical simulation, since only this type of simulation supports applications that require *all* solutions, i. e., both the ground state and all excited states—which our application requires, as will soon come to light. However, since exact simulations have an exponential time-complexity in the number of SiDBs, one ought to be very careful when to instantiate them. Regarding the immensity of the design space, as outlined in Section III, it already becomes infeasible to enumerate all possibilities with less than a handful of gates. Moreover, the innate sensitivity to crosstalk of the field-coupling-driven logic platform—as touched on in Section II-A—entails that connecting operational gates generally fails to yield an operational assembly. The solution we propose hereafter

[1] In addition, charged defects can be taken into account in a straightforward manner as mere static influences in the electrostatic landscape—just as in [16].

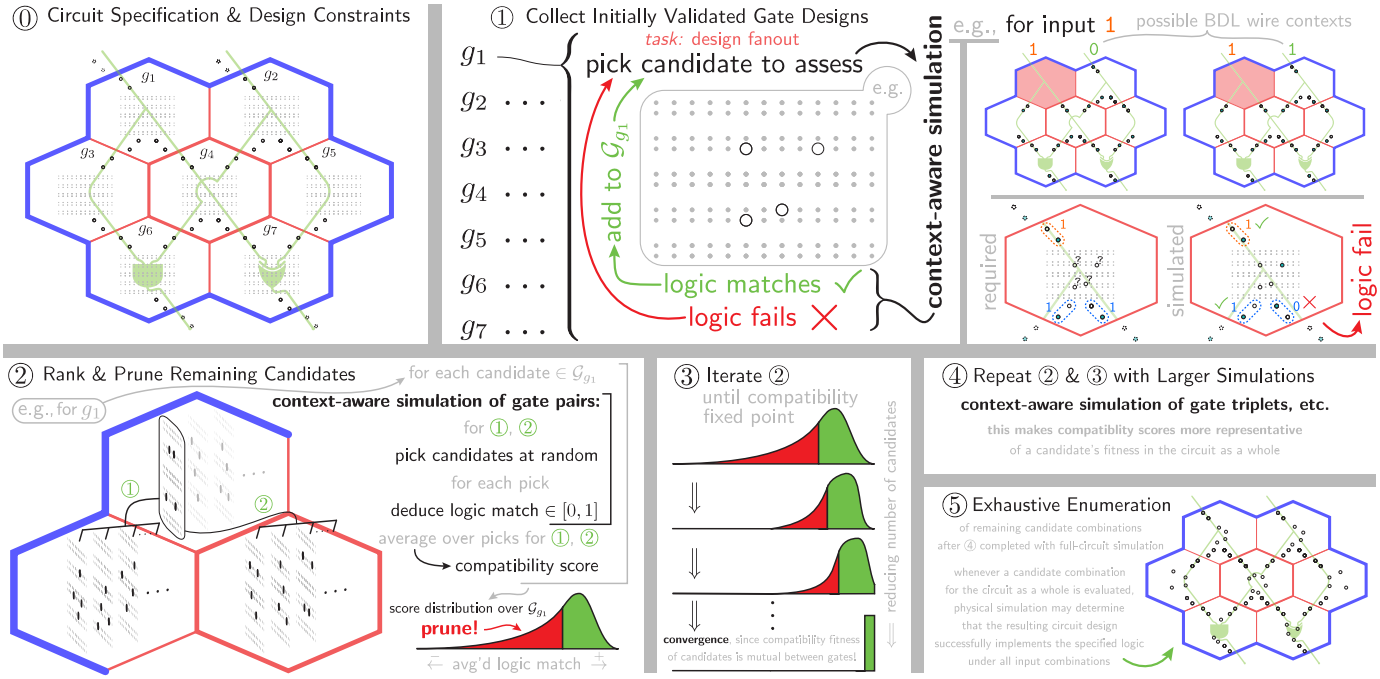

Fig. 4: Sketch of the critical components of the proposed methodology to design of gate assemblies in a shared clock zone.

Algorithm 1: Context-Aware Sub-Circuit Simulation

Input: A candidate design $\mathcal{L}$ for a sub-circuit $\hat{\mathcal{C}}$ with n inputs relative to a circuit $\mathcal{C}$ with $m \geq n$ inputs, along with an input combination $\vec{I}$ to $\hat{\mathcal{C}}$ (i. e., $|\vec{I}| = n$).

Output: The simulation results for $\hat{\mathcal{C}}$ under $\vec{I}$ with regard to $\mathcal{C}$, along with associated fractional relevance scores for logic behavior assessment.

1 $V^{\mathsf{ext}} \leftarrow [db \mapsto [\mathsf{LB} \mapsto \infty, \mathsf{UB} \mapsto -\infty]]_{db \in \mathcal{L}}$

2 $\vec{I'}^{\hat{\mathcal{C}},\vec{I}} \leftarrow \{\vec{I'} \mid$ input combination $\vec{I'}$ to $\mathcal{C}$ (i. e., $|\vec{I'}| = m$), $\vec{I'}$ applied to $\mathcal{C}$ is consistent with $\vec{I}$ applied to $\hat{\mathcal{C}}\}$

3 **if** $\left|\vec{I'}^{\hat{\mathcal{C}},\vec{I}}\right| = 0$ **then**

4 **return** $(\varnothing, \perp)$ `// $\vec{I}$ applied to $\hat{\mathcal{C}}$ cannot occur in $\mathcal{C}$`

5 **foreach** *Input combination* $\vec{I'} \in \vec{I'}^{\hat{\mathcal{C}},\vec{I}}$ **do**

6 $D \leftarrow$ the charge distribution of the BDL pairs in the skeleton of $\mathcal{C} \setminus \hat{\mathcal{C}}$ according to $\vec{I'}$

7 $D \leftarrow D \cup \mathcal{L}$ `// chargeless SiDBs are added`

8 **foreach** *SiDB* $db \in \mathcal{L}$ **do**

9 $v \leftarrow V_{db}(D)$ `// compute local potential`

10 $V^{\mathsf{ext}}(db)(\mathsf{LB}) \leftarrow \min(V^{\mathsf{ext}}(db)(\mathsf{LB}), v)$

11 $V^{\mathsf{ext}}(db)(\mathsf{UB}) \leftarrow \max(V^{\mathsf{ext}}(db)(\mathsf{UB}), v)$

12 $\vec{D_{\mathcal{L}}} \leftarrow \mathsf{ExactSimulation}(\mathcal{L}, V^{\mathsf{ext}})$

13 $R \leftarrow [D_{\mathcal{L}} \mapsto 0]_{D_{\mathcal{L}} \in \vec{D_{\mathcal{L}}}}$

14 **foreach** *Input combination* $\vec{I'} \in \vec{I'}^{\hat{\mathcal{C}},\vec{I}}$ **do**

15 $O \leftarrow \varnothing$ `// for sorting simulation results by energy`

16 **foreach** $D_{\mathcal{L}} \in \vec{D_{\mathcal{L}}}$ **do**

17 $D \leftarrow$ the charge distribution of the BDL pairs in the skeleton of $\mathcal{C} \setminus \hat{\mathcal{C}}$ according to $\vec{I'}$

18 $D \leftarrow D \cup D_{\mathcal{L}}$ `// charged SiDBs are added`

19 $O \leftarrow O \cup (D_{\mathcal{L}}, E(D))$ `// compute energy`

20 $(D^*_{\mathcal{L}}, e^*) \sim O$, s.t. $e^* = \min(\{e \mid (\_, e) \in O\})$

21 $R(D^*_{\mathcal{L}}) \leftarrow R(D^*_{\mathcal{L}}) + 1/\left|\vec{I'}^{\hat{\mathcal{C}},\vec{I}}\right|$

22 **return** $(\vec{D_{\mathcal{L}}}, R)$

contributes a solid framework for making a sub-circuit simulation *context-aware* with respect to the larger circuit.

1) Context-Aware Sub-Circuit Simulation: In order to enrich simulations of sub-circuits with awareness of neighboring gates in the circuit not taken into the simulation directly, it is possible to incorporate known information of this environment, encoding it as V^{ext}: i. e., external potential local to all SiDBs to be simulated. Since our circuit design approach uses a pre-set skeleton carrying wire signals, we may infer the possible charge distributions of the BDL pairs within the circuit that fall outside of the region to simulate, accumulating their potential effect onto each SiDB in the region to simulate to V^{ext}.

For some input combination given to the sub-circuit to simulate, there exist a select number of logic states for the BDL wires in the surrounding circuit that respect the specification of the circuit. Since the goal is to design gates that remain operational when connected with other operational gates, the principle of mathematical induction may be applied: the surrounding BDL wires are assumed to represent logic that is consistent with the current input to the sub-circuit.

However, as there can be multiple solutions here, this needs to be accounted for: V^{ext} is collected for each SiDB to simulate as a *finite range*, bounded by local potential values that are derived from assigning physical signals to the BDL wires in the circuit. These bounds are derived from the two *consistent* input combinations to the whole circuit—i. e., w.r.t. the input combination to the sub-circuit to simulate—that influence the considered SiDB with minimal and maximal electrostatic potential respectively. In Algorithm 1, the process described above is captured in the lines up to Line 12.

2) Logic Match Assessment of Simulation Results: For the pruning procedure that is outlined in Section IV-C, it is important that candidate gate implementations can be sufficiently distinguished in their simulated performance in the circuit context. To this end, it is proposed to form a *logic match*: i. e., some $r \in [0, 1]$ such that $r = 0$ when the intended logic is not at all recognizable in the simulation results, and, opposing, $r = 1$ when the specification is implemented without flaw.

In assessing a candidate design for a sub-circuit, the overall logic match is naturally averaged from the logic match attributed to the respective simulation results under different inputs to the sub-circuit that can exist in the circuit it is a part of (Lines 2–4 of Algorithm 1). However, since the exact simulation is conducted with *bounded* local external potential, the system energy computed for the returned charge distributions inherits this characteristic. This complicates the logic behavior assessment process—which otherwise regards only the ground state—as there may now be various "ground" states, relative to the various possibilities for the electrostatic environment derived from the BDL wire states in the surrounding parts of the circuit.

The proposed solution to this caveat is a fair weighing of the simulation results according to their respectively valued recognition as ground state. As described on Lines 13–22 of Algorithm 1, this is done through enumerating consistent input combinations to the whole circuit, collect-

ing actual (single-valued) system energy values under the respectively associated assumptions of the electrostatic environment. Each assumption yields one ground state, which is then credited accordingly (Line 21).[2] Now, with the simulation results and an associated weight distribution, the logic match assessment can be performed for all returned charge distributions with a non-zero weight attributed.[3]

C. Pruning Gate Implementations through Stochastic Trialing

Having outlined the methodologies involved the context-aware assessment of candidate designs of sub-circuits, the general principles conceived for employing these methods as means to achieve iterated reduction of the design space are now described. In essence, the idea is to keep the exponentially scaling simulation instances as small as possible in the number of SiDBs considered, processing polynomially many such instances in order to determine *well-founded comparative means* for distinguishing candidate gate implementations by their compatibility the current design space. Through pruning the most *in*compatible gate implementations for the respective gates in the circuit in successive pruning rounds, the design space is carefully shrunk, becoming increasingly intra-compatible until a combination of gate implementations is found that composes an operational circuit.

While the essence of such iterated pruning can be realized in various ways—ranging from heuristic to exact methods, determining the well-foundedness of the comparitive means—it is deemed out of scope for this work to evaluate the runtime and solution-finding performance of different methods for employing the novel concept of context-aware sub-circuit simulation and logic assessment presented in Section IV-B. Nonetheless, in order to expose the potential of this proposed methodological backbone for design of operational SiDB logic circuits—as showcased in Section V—a simple, yet demonstrably effective heuristic method was conceived: gate implementations are assessed through *stochastic trialing* of their compatibility with connecting gates, sampling the remaining solution space at random.

1) Proposed Algorithmic Flow: To aid in forming an intuitive understanding of this iterative pruning process using stochastic trialing, the reader is referred to Fig. 4, which captures its essence visually. To start, an initial collecting of gate implementations is performed for each gate in the circuit, using methods from [10] and [16] extended with context-aware simulation with respect to the circuit $\mathcal{C}$ to form. This yields a set G_g of gate implementations for each gate $g \in \mathcal{C}$, such that their product $\Pi_{g \in \mathcal{C}} G_g$ comprises the circuit design solution space after this initialization step.

While the initial gate design process relies on context-aware physical simulation of single gates—a scale at which even exact simulation thrives—this only accounts for the extra-simulatory SiDBs fixed by the skeleton, neglecting the imminent presence of those whose placements vary across different implementations of surrounding gates. To address this gap, we now turn to simulation of gate connections, yielding well-founded comparisons for pruning gate implementations.[4] This is applied in iteration until the discriminatory power is depleted at this one-step larger simulation scale, leaving a *significantly* reduced solution space—though, depending on the circuit design problem, further reduction may be required before enumerating the remaining combinations becomes feasible.

Before outlining how stochastic trialing yields discriminatory competence, a description of the overall algorithmic flow is completed first. Intuitively, simulating more gates at the same time yields a logic behavior assessment that is more representative of the simulated gate implementations' actual performance in the circuit. In this, a delicate balance emerges between the accuracy of the evaluated performance with respect to the actual performance, and the (exponentially scaling) simulation scale, which also scales with the number of input combinations to assess. In any case, false negatives—leading to pruning, e. g., gate implementations that would perform marvelously in the completed circuit—should be avoided. Yet at the double-gate simulation scale at which it is still feasible to carry out numerous simulations, depending on the total number of gates in the circuit, the aforementioned accuracy might be off considerably; the assessments made at this level may hold up only weakly in the full circuit scope.

Ultimately, the proposed method to circuit design relies critically on adequate consistency in the assessments as the simulation scale approaches consideration of all gates in the circuit. Hence, when tactfully pruning only the least promising candidates for each gate, the solution space is refined through careful iteration. When the candidate comparison methodology yields reasonably consistent assessments over successive stages, the solution space finally becomes intra-compatible, while sufficiently reduced in order to enumerate all remaining gate implementation combinations with relatively time-consuming full-circuit simulation. Once simulations consider all gates concurrently, it becomes possible—if not likely, depending on the circuit design problem and the precise manner of solution space refining—to find a combination that implements the circuit perfectly for all input combinations.

2) Discerning Gate Implementation Compatibility in the Circuit: For the final discussion on the proposed methodologies for design of SiDB logic circuits, a heuristic method is presented that enables candidate gate implementations to be ranked. Previously, in Section IV-B, the simulation and logic behavior assessment of sub-circuits was discussed, which offer the essential ingredients for such a ranking. These sub-circuits are formed by taking gates together in a sub-circuit simulation, throughout which also the presence of the other gates in the circuit are acknowledged. In the methodology for discerning gate implementations presented in this work, each sub-circuit $\hat{\mathcal{C}}$ is formed through taking some gate g in the circuit $\mathcal{C}$, and adding to it a set of gates not including g. Logic consistency evaluations are made for each remaining gate implementation candidate in $\mathcal{G}_g$ in the manner described next, attributing a compatibility score to each to discriminate between them.

To evaluate some candidate c in $\mathcal{G}_g$ using *stochastic trialing*, the spaces $\{\mathcal{G}_{g'} \mid g' \in \hat{\mathcal{C}},\, g' \neq g\}$ are sampled at random for a given number of trials, yielding an assignment $\phi_{\hat{\mathcal{C}}} : \forall g' \in \hat{\mathcal{C}} \,.\, \mathcal{G}_{g'}$ that instantiates the sub-circuit with respective implementations for the gates it contains. This way, the number of simulations required to assess a candidate implementation $c \in \mathcal{G}_g$ remains manageable, while still capturing the diversity of contexts in which it may operate. Each such instantiation is then assessed for logical correctness under all input combinations, proceeding as described in Section IV-B2. A score for c is determined by accumulating the real-valued logic match obtained in successive assessments, reflecting its observed *contextual compatibility*.

After all trials are completed, the cumulative score of candidate c reflects its empirical robustness across randomly sampled local circuit environments. This scoring procedure is repeated for all candidates in $\mathcal{G}_g$, resulting in a relative ranking of gate implementations based on their likelihood of functioning correctly in the broader circuit design. By removing the lowest-scoring candidates—–those least likely to be contextually sound—–the set $\mathcal{G}_g$ is pruned. Through iterative application of this process across all gates in $\mathcal{C}$, the full design space is gradually reduced to an increasingly tractable subset comprising mutually compatible gate implementations. As mentioned previously, repeating this for larger simulation scales when discriminatory capability is exhausted for the current scale—i. e., when all candidates score equally—it eventually becomes feasible to exhaustively evaluate the remaining candidates at the full-circuit level.

V. Experimental Evaluation

To assess the feasibility and effectiveness of our proposed methodology for holistic SiDB gate assembly design, we conducted a series of experiments on logic functions drawn from different NPN equivalence classes with 2 to 4 inputs. The evaluation pipeline proceeded as follows:

1) We selected representative non-constant truth tables from a range of NPN classes with 2–4 inputs, defaulting to the smallest truth table that represents it for each NPN class.
2) Using the ABC logic synthesis tool [28], we generated high-level logic descriptions (Verilog files) for each target function.
3) Each Verilog description was processed by the SMT-based placement and routing (P&R) tool `exact` [29], modified to include a constraint that the resulting gate layout must fit entirely within a single supertile. The tool then determined which logic function would be hosted at each tile in the supertile.

[2]While in small, typically symmetric SiDB assemblies it is not uncommon to find multiple *degenerate* ground states, this becomes negligible in larger systems.

[3]A specific implementation for the logic match assessment for a simulated charge distribution of a sub-circuit is left out, as there exist a multitude of solutions for this—their assessment is out of scope for this work. For the experimental evaluation in Section V, a simple ratio was used: the number of BDL pairs exhibiting correct logic behavior relative to the total number of BDL pairs in the simulated sub-circuit.

[4]For our experimental evaluation of the proposed techniques—presented in Section V—runtime is saved by invoking simulations efficaciously; i. e., to roughly discern gate implementations with minimal resources. Here, only gate connections are considered—more accurate discrimination may be achieved through, e. g., considering all surrounding gates, or all gates in the circuit.

TABLE I: *Expanding SiDB Standard Cell Functionality from Single-Gate to Circuit:* Various NPN Classes Crammed onto a Supertile

#in	tt	$\lvert\mathcal{C}\rvert$	time	#in	tt	$\lvert\mathcal{C}\rvert$	time	#in	tt	$\lvert\mathcal{C}\rvert$	time	#in	tt	$\lvert\mathcal{C}\rvert$	time	#in	tt	$\lvert\mathcal{C}\rvert$	time
2	0x1	2	0.0137	3	0x7	4	0.920	4	0x7	5	196.430	4	0x7f	5	13.116	4	0x660	5	408.571
2	0x6	2	0.0218	3	0x1e	2	0.160	4	0x1e	5	14.889	4	0x1fe	5	156.671	4	0x6f9	5	110.857
3	0x1	2	1.073	3	0x69	2	0.129	4	0x1f	5	84.400	4	0x356	5	102.380	4	0x7f8	5	112.730
3	0x6	4	1.565	4	0x1	5	120.555	4	0x69	5	85.309	4	0x357	5	82.672	4	0x996	5	107.555

#in = number of inputs tt = truth table identifying NPN class (hex) $\lvert\mathcal{C}\rvert$ = number of gates time = runtime (min)

4) For each function for which the supertile P&R was successful, we applied our proposed co-design methodology for compact, context-aware gate assembly design in a shared clock zone.[5]
5) We recorded whether the co-design process was successful, the number of tiles involved in the final layout, and the total runtime of the full pipeline.

All experiments were conducted on a system equipped with an Intel Core i7-14900HX processor (24 cores, base/turbo: 2.2 GHz / 5.8 GHz, 36 MB L3 cache) and 32 GB DDR5 RAM. The code was written in C++17 on top of the *Munich Nanotech Toolkit* (MNT) [30], and is made publicly available at https://github.com/cda-tum/fiction.

Table I summarizes the results, reporting only functions for which placement and routing was successful. It shows that our method successfully managed the vast design space in several non-trivial cases, demonstrating the viability of our approach for dense atomic-scale logic design.

These results highlight the effectiveness of our simulation-driven pruning and context-aware design methodology. Despite the immense combinatorial design space, the approach consistently identified working implementations for diverse logic functions, within practical runtimes. This lays a solid foundation for further refinement and automation of ultra-dense atomic logic synthesis in shared clock zones. Critically, this work enables the automated design of SiDB circuits that operate without error *by design*—as assessed through exact validation at each clock zone. This represents a substantial departure from prior work, which validates only isolated logic gates and therefore disregards the changes in logic behavior that arise when these gates are expanded with additional SiDBs to span entire clock zones. Accordingly, our holistic methodology forms the missing piece of the puzzle in advancing reliable atom-scale logic design.

VI. Conclusion

Field-coupled nanocomputing (FCN) offers a compelling pathway beyond the limitations of traditional CMOS, and *silicon dangling bond* (SiDB) logic—thanks to its atomic precision and energy efficiency—stands at the forefront of this vision. While recent efforts have successfully established an end-to-end design flow that synthesizes SiDB circuits from logical specifications, they have done so under restrictive layout conditions that squander the very density advantages SiDB promises. By enforcing a one-gate-per-clock-zone paradigm to combat crosstalk via coarse metal-pitch clocking, the state-of-the-art effectively inflates atomic-scale logic to macroscopic proportions.

In this work, we challenged that compromise. We embraced the unavoidable complexity that arises from designing multiple logic gates to co-exist and interact in a single clock zone—a necessity for unlocking true atomic-scale density. Despite exponential growth in *both* the design and validation space, our holistic gate design methodology provides a viable pathway through this labyrinth via efficacious assessment of candidate gate implementations, enabling quintessential pruning steps. Rather than avoiding the crosstalk that defines FCN behavior, we tamed it—co-designing gate assemblies that operate in harmony within the shared electrostatic landscape of a single clock field.

Our experimental results show that this methodology is not only conceptually sound but practically effective: high-level logic specifications are crammed into a single clock zone using exact placement and routing, after which the holistic gate design methodology—based on exact physical simulation—battles a gargantuan search space to create a layout of SiDBs that implements the desired logic. As the first work to present successful SiDB logic design—as validated through physical simulation—of *circuits* rather than singled-out gates, our pioneering work sets the stage in an advanced frontier towards large-scale SiDB design automation, now enabling dense logic design within a single clock zone using the proposed holistic design methodology. The fact that this is possible—let alone in mere seconds for small gate assemblies, and up to a few hours for large ones—proves that ultra-dense, holistic SiDB circuit design is not just a theoretical ambition but an achievable reality. With this, we take a decisive step toward the future of atom-scale computing: one in which every clock tick carries not just a gate, but a circuit.

[5]Since this work focusses on design of supertiles from [11] made up of hexagons, the P&R problem is significantly constrained, and, as a result, many NPN classes cannot be implemented on such a supertile. Since this work presents a methodology that is separate from this P&R and tiling problem, it is left to future work to explore how the proposed methodology can be capitalized further.

References

[1] T. Huff *et al.*, "Binary Atomic Silicon Logic," *Nature Electronics*, 2018.
[2] L. Livadaru *et al.*, "Dangling-bond Charge Qubit on a Silicon Surface," *New Journal of Physics*, 2010.
[3] M. Rashidi *et al.*, "Automated Atomic Scale Fabrication," US Patent, 2022, 17/429,443.
[4] R. A. Wolkow *et al.*, "Quantum random number generator," Patent US Patent App. 18/282,631, may 23, 2024.
[5] ——, "Initiating and Monitoring the Evolution of Single Electrons within Atom-defined Structures," US Patent 11 047 877, 2021.
[6] R. Achal *et al.*, "Lithography for editable atomic-scale devices and memories," US Patent 11,955,172, 2024.
[7] T. R. Huff *et al.*, "Atomic White-Out: Enabling Atomic Circuitry through Mechanically Induced Bonding of Single Hydrogen Atoms to a Silicon Surface," *ACS Nano*, 2017.
[8] S. S. H. Ng *et al.*, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *TNANO*, 2020.
[9] S. Hofmann *et al.*, "Scalable Physical Design for Silicon Dangling Bond Logic: How a 45° Turn Prevents the Reinvention of the Wheel," in *IEEE-NANO*, 2023.
[10] J. Drewniok *et al.*, "Minimal Design of SiDB Gates: An Optimal Basis for Circuits Based on Silicon Dangling Bonds," in *NANOARCH*, 2023.
[11] M. Walter *et al.*, "Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, 2022, pp. 739–744.
[12] ——, "Atomic Defect-Aware Physical Design of Silicon Dangling Bond Logic on the H-Si(100)-2×1 Surface," in *DATE*, 2024.
[13] R. Lupoiu *et al.*, "Automated Atomic Silicon Quantum Dot Circuit Design via Deep Reinforcement Learning," 2022. [Online]. Available: http://arxiv.org/abs/2204.06288
[14] S. Hofmann *et al.*, "Late Breaking Results: Wiring Reduction for Field-coupled Nanotechnologies," in *DAC*, 2024.
[15] ——, "A* is Born: Efficient and Scalable Physical Design for Field-coupled Nanocomputing," in *NANO*, 2024, pp. 80–85.
[16] J. Drewniok *et al.*, "On-the-fly Defect-Aware Design of Circuits based on Silicon Dangling Bond Logic," in *NANO*, 2024, pp. 30–35.
[17] V. Vankamamidi *et al.*, "Clocking and Cell Placement for QCA," in *IEEE-NANO*, vol. 1, 2006.
[18] M. Walter *et al.*, "Design Automation for Field-coupled Nanotechnologies," in *ISVLSI*, 2020.
[19] T. R. Huff *et al.*, "Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot," *ACS Nano*, 2019.
[20] J. Drewniok *et al.*, "The Need for Speed: Efficient Exact Simulation of Silicon Dangling Bond Logic," in *ASP-DAC*, 2024.
[21] W. Lambooy *et al.*, "Mastering the Exponential Complexity of Exact Physical Simulation of Silicon Dangling Bonds," in *ASP-DAC*, 2026.
[22] C. S. Lent *et al.*, "Molecular Quantum-Dot Cellular Automata," *Journal of the American Chemical Society*, vol. 125, no. 4, pp. 1056–1063, 2003.
[23] M. Ziatdinov *et al.*, "Robust multi-scale multi-feature deep learning for atomic and defect identification in scanning tunneling microscopy on h-si(100) 2x1 surface," *arXiv*, 2020.
[24] J. Pitters *et al.*, "Atomically Precise Manufacturing of Silicon Electronics," *ACS Nano*, 2024.
[25] M. Walter *et al.*, "Reducing the Complexity of Operational Domain Computation in Silicon Dangling Bond Logic," in *NANOARCH*, 2023.
[26] J. Drewniok *et al.*, "Temperature Behavior of Silicon Dangling Bond Logic," in *IEEE International Conference on Nanotechnology (IEEE NANO)*, 2023, pp. 925–930.
[27] ——, "Unifying Figures of Merit: A Versatile Cost Function for Silicon Dangling Bond Logic," in *IEEE International Conference on Nanotechnology (IEEE NANO)*, 2024, pp. 91–96.
[28] R. Brayton and A. Mishchenko, "ABC: An Academic Industrial-Strength Verification Tool," in *Computer Aided Verification*. Springer, 2010, pp. 24–40.
[29] M. Walter *et al.*, "An Exact Method for Design Exploration of Quantum-dot Cellular Automata," in *DATE*, 2018.
[30] ——, "The Munich Nanotech Toolkit (MNT)," in *IEEE-NANO*, 2024.