Mastering the Exponential Complexity of Exact Physical Simulation of Silicon Dangling Bonds

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

*Chair for Design Automation, 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*—*Silicon Dangling Bond* (SiDB) logic is a promising technology for energy-efficient computation, supported by significant advancements in manufacturing and design automation. However, physical simulation, essential for accurately predicting the behavior of SiDB logic prior to costly manufacturing, lags behind these developments. In particular, exact physical simulation, which scales exponentially with base 3, remains infeasible for larger SiDB assemblies, limiting its utility to small structures such as single gates. This computational bottleneck slows progress in SiDB technology and hinders the establishment of reliable ground truths for heuristic approaches. To address the challenge, this work presents a novel methodology for exact SiDB simulation that restructures the exponential search space according to a *hierarchical clustering*. The hierarchy structure enables a systematic pruning of the search space at its different levels: it provides an ordering of interactions between clusters of SiDBs to facilitate *efficacious* exploitation of dynamically-inferred problem-specific constraints—like solving a Sudoku. Experimental results demonstrate that the effective exponential base can be lowered to approximately 1.3, enabling, for the first time, the exact physical simulation of entire multi-gate SiDB circuits in minutes that would take the state of the art millions of years to compute. This breakthrough establishes a robust ground truth for SiDB logic validation, marking a pivotal step toward scalable, energy-efficient, and atomic-scale computing.**

I. Introduction

The rapid expansion of AI and digital transformation across industries is dramatically reshaping the demand for computing power. This surge in demand has significantly increased the need for data centers, which are set to consume three times the energy they use today by 2030 [1]. Traditional computing architectures, however, are facing physical and economic limitations, and the rising energy needs have spurred interest in more energy-efficient computing technologies.

One promising technology for energy-efficient computation is based on *Silicon Dangling Bonds* (SiDBs) to enable computing at the atomic level [2]–[5]. In this approach, precise control of charge states enables logic and memory functionality, overcoming the limitations of traditional transistor-based designs. SiDB logic has the potential to revolutionize computing by significantly reducing power consumption beyond the capabilities of conventional CMOS [6].

This growing interest has already led to the development of gate and circuit libraries, as well as design automation solutions aimed at enabling large-scale device design using the SiDB technology [7]–[17]. Moreover, with the establishment of the first enterprise adopters like *Quantum Silicon Inc.*, SiDB technology is making its way from academic research to industrial applications [4], [5], [18]–[20].

But while significant progress has been made in manufacturing and design automation tools for SiDB technology, *physical simulation*, essential for predicting the behavior of SiDB logic prior to costly manufacturing, is still lagging behind [2], [3], [7], [8], [21], [22]. In particular, *exact* physical simulation, which is essential for 1) establishing a reliable ground truth to evaluate the accuracy of heuristic approaches, and 2) supporting applications that require not only the ground state but all possible solutions, such as temperature-dependent simulations [23], is far behind manufacturing capabilities. More precisely, the state of the art in exact physical simulation is limited to simulating small SiDB assemblies such as single gates [22], rendering exact multi-gate simulation infeasible until now.

This backlog in exact physical simulation, compared to design automation and manufacturing capabilities, is due to its highly complex nature, requiring the solution of quantum mechanical equations. Even with semi-classical models, simulations require exhaustive consideration of a search space that scales exponentially with 3^n, where n is the number of SiDBs in the layout, leading to runtime demands that quickly become prohibitive.

To master this exponential complexity, this paper presents a new methodology for the physical simulation of SiDBs, capable of consistently and significantly reducing the exponential base. Our approach uses *hierarchical clustering*, where SiDBs are repeatedly grouped according to their electrostatic interactions to enable a pioneering command of the simulation search space. This structures the interaction network among all considered SiDBs, providing critical information for targeted search space reduction to *efficaciously* exploit problem-specific constraints. The hierarchically restructured search space is pruned systematically at different levels, iteratively leveraging constraints accumulated from previous search space reductions to produce further constraints—much like the human method to Sudoku solving.

Experimental evaluations show that a substantial reduction in the exponential base is achieved, bringing it down to around 1.3 for SiDB logic implementations. This advancement enables the simulation of SiDB circuits for the first time, establishing a ground truth for the physical simulation of SiDB circuits. This highlights the profound impact of *search space pruning in a cluster hierarchy* on mastering the complexity of exact physical simulation of SiDBs, paving the way for energy-efficient and atomic-scale computing.

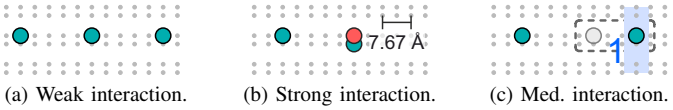
The remainder of this paper is structured as follows: in order to make this work self-contained, Section II reviews SiDBs, how they are used for logic, and the physical model to describe their electrostatic interactions, which is essential to understand the working principle in their application to logic realization. Section III explains what exact physical simulation means in the context of SiDB logic and reviews its state of the art. By outlining the current limitations, we underscore the urgent need for an efficient, exact physical simulator capable of handling multi-gate SiDB layouts, establishing a ground truth, and enabling temperature-dependent simulation. Section IV is the main contribution of this paper as it provides a detailed introduction to the proposed methodology. Section V presents an exhaustive experimental evaluation demonstrating the newfound mastery over exponential complexity, showcasing the ability to simulate multi-gate SiDB circuits. Finally, Section VI concludes the paper.

II. Preliminaries

This section covers the main concepts necessary for understanding the remainder of this work. First, Section II-A reviews SiDBs, explaining how they can be used to conduct computation at the atomic scale. Second, Section II-B reviews the physical model of SiDBs, which is crucial for understanding their underlying working principles and provides the basis for simulation methods.

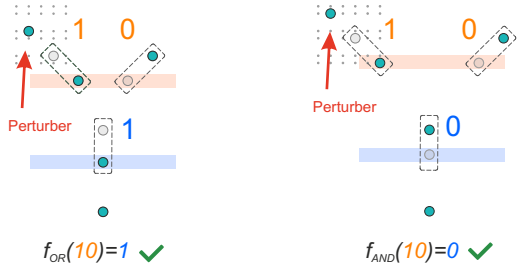
A. The SiDB Logic Platform

The fabrication of *Silicon Dangling Bonds* (SiDBs) begins with a hydrogen-passivated silicon crystal surface (H-Si(100)-2×1). Using a *Scanning Tunneling Microscope* (STM), hydrogen atoms are removed with atomic precision to create SiDBs, enabling their controlled arrangement at the atomic level [3], [4], [24]. An undisturbed SiDB hosts two electrons, charging it negatively (−). However, depending on surrounding electrostatic interactions, the charge state can change: moderately strong electrostatic interactions ($> -0.9\,\mathrm{V}; < -0.3\,\mathrm{V}$) may



(a) Weak interaction. (b) Strong interaction. (c) Med. interaction.

Fig. 1: SiDB Charge State Variability. Charge states $-$, $\mathbf{0}$ and $+$ are respectively indicated by turquoise, gray, and red.



(a) OR gate from [2] with input pattern `10` and output `1`.

(b) AND gate from [8] with input pattern `10` and output `0`.

Fig. 2: Operational Essentials of the SiDB Binary Logic Platform.

leave the SiDB neutral ($\mathbf{0}$), while strong interactions ($< -0.9\,\mathrm{V}$) can render it positively charged ($+$) [8], [21].

Example 1. *Consider Fig. 1, which sketches three SiDBs in three different arrangements. In Fig. 1a, the SiDBs are spaced four positions apart, resulting in weak electrostatic interactions that do not affect the charge states. In contrast, Fig. 1b shows an arrangement in which two SiDBs are positioned adjacent to each other, leading to strong electrostatic interactions ($< -0.9\,\mathrm{V}$) and causing the upper SiDB to become positively charged. In Fig. 1c, the SiDBs are placed at a medium distance, so that one SiDB becomes neutral while the other two remain negative. This case is particularly interesting as it may encode binary information: bit `1` is encoded by the location of the negative charge state in the pair highlighted by the dashed box, which spans less than* $1\,\mathrm{nm}$.

This pair of two SiDBs is referred to as a *Binary-dot Logic* (BDL) pair. Since SiDBs form an interwoven system with variable charge states, logic gates can be realized that operate solely by electrostatic repulsion. A single SiDB, known as a *perturber*, can be used to apply electrostatic pressure to set adjacent SiDB BDL pairs into a desired binary state. In fact, gates with a footprint of less than $30\,\mathrm{nm}^2$ have been fabricated this way [24].

Example 2. *Consider the placement of six SiDBs in two distinct arrangements, as illustrated in Fig. 2. Fig. 2a depicts an OR gate (experimentally verified in [24]) with the input* `10` *and an output of* `1`*. This information is encoded in the charge states of the BDL pairs. The input* `1` *is enforced by the perturber, which exerts electrostatic pressure on the left input BDL pair. Adjusting the distance between the BDL pairs reduces the electrostatic repulsion, resulting in altered behavior. Consequently, Fig. 2b depicts an AND gate with the input pattern* `10` *applied, yielding the expected output* `0`*.*

After introducing the SiDB technology and providing an understanding of how SiDBs can be used to construct logic gates based on the variability of charge states, the next section reviews the physical model used to qualitatively describe and simulate their behavior. This model serves as the basis for simulating the charge states of SiDBs.

B. Physical Model of SiDBs

As mentioned, SiDBs interact electrostatically with each other, which is described by the screened Coulomb potential as follows: let $\mathcal{L}$ denote a layout of SiDBs, such that any charge distribution for $\mathcal{L}$ is represented by a function of the type $\mathcal{D} := \{-1, 0, 1\}^{\mathcal{L}}$. We denote the set of charge states as $\{-, \mathbf{0}, +\}$ in the remainder. For any two SiDBs $i, j \in \mathcal{L}$, the electrostatic potential $V_{i,j}$ for a charge distribution $D \in \mathcal{D}$ is given by

$$V_{i,j}(D) = -\frac{q_e}{4\pi\epsilon_0\epsilon_r} \cdot \frac{e^{-\frac{d_{i,j}}{\lambda_{tf}}}}{d_{i,j}} \cdot D(j), \tag{1}$$

where λ_{tf} defines the *Thomas-Fermi screening length* and ϵ_r the *dielectric constant*, which were experimentally extracted to be $5\,\mathrm{nm}$ and 5.6, respectively [21], [24]. In addition, ϵ_0, q_e, and $d_{i,j}$ are the *vacuum permittivity*, the *electron charge* ($q_e = -e$; e: *elementary charge*), and the *Euclidean distance* between SiDBs i and j, respectively. These electrostatic interactions are locally accumulated at each SiDB i. Thus, its local potential for the given charge distribution is calculated as

$$V_{local,i}(D) = \sum_{j \in \mathcal{L},\, j \neq i} V_{i,j}(D). \tag{2}$$

By the physical model of SiDBs and their interactions, the local electrostatic potential at SiDB i dictates the charge state it attains. This means that not all $D \in \mathcal{D}$ adhere to the physical model. Instead, for such adherence, charge distributions need to obey the so-called *Population Stability* and *Configuration Stability*, which can be expressed by the respective following conditions.

1) For each SiDB $i \in \mathcal{L}$, $D(i) = -$ when $\mu_- + V_{local,i}(D) \cdot q_e < 0$, $D(i) = +$ when $\mu_+ + V_{local,i}(D) \cdot q_e > 0$, and $D(i) = \mathbf{0}$ otherwise. Thus, μ_- and μ_+ represent the thresholds that bound the energetic ranges associated with the respective charge states.
2) There exists no $D' \neq D$ that is reachable from D with a single-electron hop between SiDBs such that the system's total electrostatic potential energy reduces. Otherwise, the system will spontaneously transition from D to D'.

If both conditions are fulfilled for D, this state is said to be *metastable*, or, synonymously, *physically valid*, which we denote by means of a predicate $\mathcal{V}$ for which $\mathcal{V}(D)$ then evaluates to *true*. Therefore, the primary objective of any physical simulator for SiDB systems is to identify metastable charge distributions, with the lowest-energy configuration referred to as the *ground state*. The size of the simulation search space presents a significant computational challenge, since $|\mathcal{D}| = 3^n$, where $n = |\mathcal{L}|$. Here, n represents the number of SiDBs, each of which can exist in one of the three aforementioned charge states. The task of determining *all* solutions, i. e., all possible charge distributions that satisfy $\mathcal{V}$, is referred to as *exact* physical simulation, which is considered in this work.

III. Exact Physical Simulation & Related Work

This section discusses the exact physical simulation of SiDB layouts and reviews the status quo. First, in Section III-A, the considered problem is motivated and described. Second, in Section III-B, the state of the art in exact physical simulation is reviewed. Underlining current limitations, we highlight the need for a more advanced and efficient approach.

A. Exact Physical Simulation of SiDB Layouts

Exact physical simulation of SiDB logic is a crucial step in its development as a post-CMOS technology, as it 1) establishes a reliable ground truth to evaluate the accuracy of heuristic approaches, 2) supports applications such as temperature-dependent simulations that depend on all possible solutions to the system that the physical model offers [23], and, 3) enables the exhaustive search for logic gate implementations, which requires certificates of gate operation for each candidate implementation [25]. Accordingly, the objective of simulating SiDBs exactly is to identify all possible charge distributions that meet the condition of *physical validity*, meaning they align with the underlying physical model. Thus, the objective of this task is to compute the set $\mathcal{D}_\mathcal{V} := \{D \in \mathcal{D} \mid \mathcal{V}(D)\}$. Since the number of possible charge configurations is exponential with base 3, finding the exhaustive subset of physically valid ones is a complex task. An initial approach forms $\mathcal{D}_\mathcal{V}$ with a runtime complexity of $\Theta(3^n)$ through enumerating all of $\mathcal{D}$, though the state of the art manages to achieve $\mathcal{O}(3^n)$, as discussed in more detail hereafter [22], [26].

B. State of the Art and its Limitations

Simulating SiDBs in an exact fashion is particularly challenging due to the base-3 exponentially large search space. The state-of-the-art physical simulator, *QuickExact*, employs a technique to address this challenge: *search space pruning*. Although this simulator may still enumerate all 3^n possible charge distributions, it is able to determine the charge state of d SiDBs based on their specific locations and corresponding electrostatic interactions before the enumeration process begins. This reduces the search space by a factor of 3^d, which significantly improves simulation runtimes for $d > 0$. As a result, *QuickExact* made it feasible to exactly simulate single gates with up to around 30 SiDBs in reasonable runtime [22].

Specifically, the state-of-the-art search space pruning can statically track down SiDBs in sparsely populated areas of the given layout, pre-determining that they must be negatively charged in all possible physically valid charge distributions. Hereby, *certain* specifics of the input layout are exploited for an exponential benefit in computing $\mathcal{D}_\mathcal{V}$, yet, the state of the art fails to do so *generically*: it cannot make inferences about groups of SiDBs to further reduce the search space. This is illustrated with the following example.

Example 3. *Consider the layout seen in Fig. 1b as an exact physical simulation problem. When a pruning process only assesses each SiDB separately, it would determine the separated SiDB to be negatively charged, while leaving e. g., the all-negative charge distribution to be validated in the enumeration of the remaining search space. However, with the intuition given in Example 1, one may infer that the strong electrostatic interaction between the two closely-spaced SiDBs in this layout—say, i and j—causes them to assume oppositely polarized charge states in any metastable charge distribution. Thus, 6 out of the $3^2 = 9$ possible charge state assignments to SiDBs i and j, can be pruned, leaving only $[i \mapsto \mathbf{0}, j \mapsto \mathbf{0}]$, $[i \mapsto -, j \mapsto +]$, and $[i \mapsto +, j \mapsto -]$.*

To summarize, the technique of capitalizing on specifics of the given layout of SiDBs is a potent seed, having opened the door to single-gate exact simulation. However, while the state-of-the-art adoption of the search space pruning technique allows it to infer restrictions on the charge state of single SiDBs, its pruning assessments remain limited to evaluating possibilities for atomic entities. As a result, the benefit is occasional, causing it to succumb to the base-3 exponential search space that remains when the static analysis leaves more than $\approx$ 30 SiDBs without a fixed charge state.

With these limitations of the state of the art in mind, the following section proposes a methodology to fully seize the potency of search space pruning, allowing the established technique to come to fruition with the exact simulation of larger circuit layouts.

IV. Proposed Methodology

Motivated by the observations from Section III-B, in this section, we propose a methodology that leverages the search space pruning technique to a far greater extent than previously achieved. In fact, the technique is generalized and extended for higher-order reasoning, enabling pruning assessments for non-atomic *clusters* of SiDBs. The generic effectiveness of this approach, as investigated later in Section V, directly shifts the time-complexity of the exact simulation of SiDB logic layouts with n SiDBs from $\mathcal{O}(3^n)$ to approximately $\mathcal{O}(1.3^n)$.

This section starts with Section IV-A outlining the proposed idea. Afterward, Section IV-B provides a substantive, yet high-level view of the proposed methodology's course, accentuating the divergence from existing solutions to exact SiDB simulation with its unique, *constructive* approach. Accordingly, the remaining subsections, with in particular Section IV-C, build toward an understanding of the proposed work through an elaborated series of definitions.

A. An Intuition for Search Space Pruning in a Cluster Hierarchy

In Section III-B, it was reviewed how the state of the art can infer limitations on the charge state of single SiDBs. With a much abstracted view, the analogy with the human method of solving a Sudoku is found: given some initial state of the Sudoku, one looks for squares that can be filled in directly using the available constraints. The analogy with search space pruning, as the state of the art adopts it, ends here, though for the proposed extension and generalization thereof, it only begins.

After one has filled in a Sudoku square through direct inference on the initial state, the constraints that emerge with this result may in turn be exploited to make basic inferences on this new state. Precisely this intuition translates to the proposed extension to search space pruning: employing *iteration*. After some iterations, the same trick might not yield anything further. The common strategy is then to find squares for which the possible value assignments are mutually dependent. Upon making such a grouping, one treats the group as a single entity with a bundled outgoing effect, allowing further deductions to be made.

The above is directly analogous to the proposed approach to exact simulation in which the objective of search space pruning is inherent to every algorithmic step. Adopting the elevated perspective that was sketched, state-of-the-art single SiDB pruning assessments can now be seen as inferences at the lowest level, since they can be inferred directly. Going beyond, and in line with the movements in the Sudoku-solving analogy, we may advance to higher-level reasoning through merging clusters of SiDBs into one. With such successive merges, a *cluster hierarchy* is established.

Fig. 3 may be glanced at for a visualization of the process described above, although definitions that allow for rigid interpretation are still to come with the following subsections.

B. Synopsis: Obtaining all Metastable States, Semi-Constructively

The proposed methodology consists of two consecutive stages that are elaborated on throughout the remainder of this section. In the first stage, which we refer to as the *construction stage* and discuss in detail in Section IV-C, a recursive data structure is built up that represents a minimized search space. The *type* of this structure is in essence the prominent contribution of this work, since this information-rich yet efficient reinterpretation of the exponential simulation search space naturally gives rise to an implementation that capitalizes on the pioneering command thereof: this proposed exact SiDB simulation algorithm is presented in Section V.

In the second stage of our approach, dually referred to as the *destruction stage* and described in Section IV-D, the data structure that was constructed in the preceding stage is broken down throughout a recursive unfolding procedure that efficiently extracts all physically valid charge distributions. Rather than just enumerating all possibilities in a (reduced) search space—i. e., *destructive* methodology, which foregoing exact simulators adopt—the proposed destructive stage enumerates possibilities in a bespoke arboreal representation of the search space in which it can prune branches, making it vastly more efficient.

C. Construction of the Ground State Space

To motivate the constructive approach to exact physical simulation of SiDBs, it can be considered that for logic applications, $|\mathcal{D}_\mathcal{V}| \lll |\mathcal{D}|$; i. e., all metastable charge distributions form a minute portion of the search space. Since, as outlined in Section III, computing the former set is the objective of this exact simulation task, it would be tremendously more efficient to *construct* $\mathcal{D}_\mathcal{V}$ rather than filter out all $D \in \mathcal{D}$ for which $\neg\mathcal{V}(D)$—as implemented by current destructive methodologies.

While conceptually ideal, such methods have remained elusive due to the global interdependence of the SiDBs' charge states, as outlined in the following example.

Example 4. *We consider the task of obtaining $\mathcal{D}_\mathcal{V}$ for an SiDB layout of two gates G_1, G_2 that are connected in sequence. Since the gates are separated with the intent to allow for independent functioning, the Coulombic potential between SiDBs in separate gates is weak due to the exponential interaction decay. Yet, an accumulation of many of these inter-gate interactions at an SiDB in G_1 can easily cause it to cross a charge state transition threshold which it would otherwise not without G_2's presence. This affects the possible charge distributions in G_1, in turn influencing G_2 to complete the circle. Thus, the fully connected interaction network is not easily divided such that $\mathcal{D}_\mathcal{V}$ can be determined by first computing it for either gate separately.*

Thus, it becomes difficult to form solutions to bigger exact physical simulation problems by combining results from separated sub-problems. Difficult, but not impossible, we demonstrate in Section IV-D, which

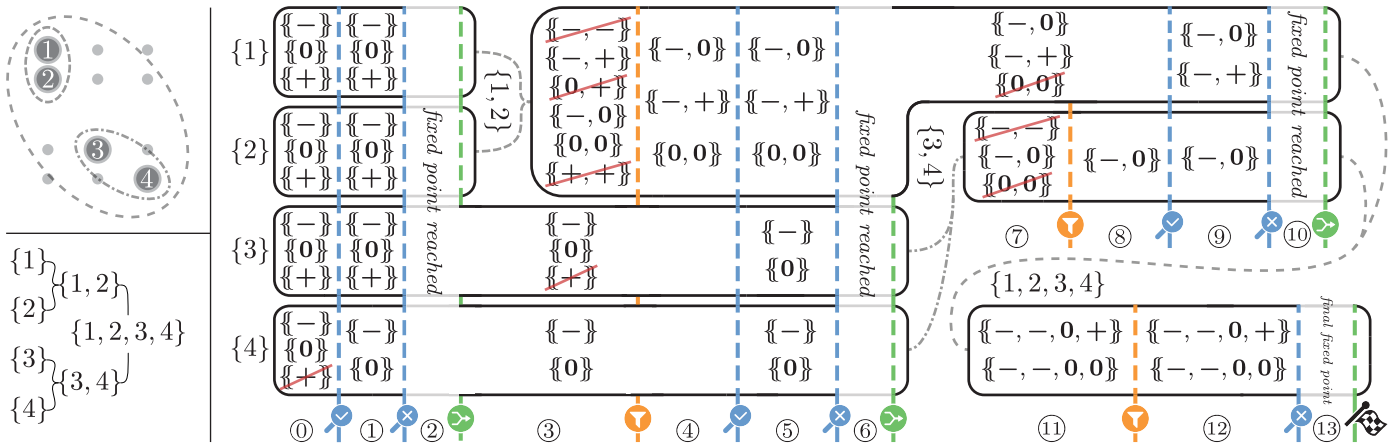


Fig. 3: Demonstration of a ground state space construction for a layout with four SiDBs under $\mu_- = -0.32\,\mathrm{V}$, $\lambda_{tf} = 5\,\mathrm{nm}$, and $\epsilon_r = 5.6$. The information state transitions that are indicated by the vertical lines only overlap clusters that are considered in the corresponding procedure.

critically depends on a restructured search space construct in which a destructive filtering methodology can be applied.

The construction stage, described hereafter, produces a restructured and minimized exact simulation search space that we call the *ground state space*. In short, the construction proceeds by strategically merging highly interactive clusters of SiDBs whilst pruning sections of the search space by analyzing the associated bounds on local electrostatic potentials. Through prioritizing only these most important interactions, the search space is pruned effectively and efficaciously.

In the following paragraphs, sequentially dependent definitions are given in order to express the aforementioned quintessential structure in its most important concepts. To enable an intuitive understanding of the construction stage, an exemplifying visualization is given in Fig. 3, where the ground state space construction of a 4-SiDB layout is shown. Consecutive states throughout this construction are displayed as labeled by ⓪ up to ⑬, with state transitions indicated by vertical separators that we may refer to by, e. g., ⓪ → ①. As definitions roll out throughout the following, relevant references are made to Fig. 3, for which interpretation thusly becomes apparent over this course. Alongside, Algorithm 1 provides a high-level description of this stage.

1) SiDB Layouts as Cluster Hierarchies: A prerequisite for the ability to solve bigger exact simulation problems from solutions to smaller ones is a recursive subdivision of the given layout of SiDBs. A *hierarchical clustering* naturally emerges, forming the root structure that the entire proposed simulation approach depends on.[1] The recursive structure is bounded on the top by the *top cluster*, i. e., the cluster containing all SiDBs in $\mathcal{L}$ that are each independently represented by singleton clusters at the bottom of the hierarchy. Any set of non-overlapping clusters that unify to $\mathcal{L}$ is called a *clustering*.

In Fig. 3, a layout with four SiDBs is shown, in which groupings of nearby SiDBs are made that are themselves, in turn, grouped to form the top cluster. Now recall from Eq. (1) that the inter-SiDB Coulombic potential decays exponentially over distance $d_{i,j}$, and further recall that the charge state of any SiDB depends on the accumulation of electrostatic potential with each other SiDB in the layout. Through forming clusters based on a proximity heuristic, the resulting hierarchy highlights which interactions are most critically interdependent. This property is exploited throughout the proposed methodology's course, whose efficacy thus depends on the quality of the hierarchical clustering by the aforementioned heuristic.

2) A Complete Information Structure of Potential Bounds: As we aim to enable the application of the search space pruning technique to any *clustering* of $\mathcal{L}$ in an iterated process, the technique is adopted through a generic conjoining of exclusions, rather than the specialized approach that the state of the art uses.[2] Now, in order to achieve an information structure from any given clustering χ in which such exclusions are to be made, it is imperative that the structure is complete in the representation of an information state in which *bounds* on the electrostatic potential local to each SiDB separately may be inferred. In accordance with the fully connected nature of the interactions within a system of SiDBs, any cluster $C \in \chi$ is annotated with such *potential bounds* relative to each SiDB $i \in \mathcal{L}$. Such information represents the bounds on the partial sum of the electrostatic potential local to i, where only the interactions from SiDBs in C are considered.

To illustrate, for D_- and D_+ respectively denoting the all-negative and all-positive charge distributions, a pair of potential bounds in an information structure in which no pruning has yet been performed can be formulated as follows: $\sum_{j \in C} V_{i,j}(D_z)$ with $z = -$ and $z = +$ for the lower and upper bound respectively.

3) Charge Multisets; Counting Charge States in Clusters: While, as described in Section III-B, state-of-the-art search space pruning is limited to excluding specific charge states assigned to SiDBs, we demonstrate that this is only the first step of a much more powerful pruning procedure. Enabling such extension is non-trivial, however: whereas pruning charge states for each SiDB separately requires constant time, i. e., $\forall i \in \mathcal{L}\,.\,\mathcal{O}(|\{i\} \to \{-, \mathbf{0}, +\}|) = \mathcal{O}(1)$, a naive adaptation of this process that operates on separate clusters $C \subseteq \mathcal{L}$ in a clustering χ (i. e., for which $\bigcup \chi = \mathcal{L}$) is quickly met with an exponential blow-up in $|C|$, i. e., $\forall C \in \chi\,.\,\mathcal{O}(|C \to \{-, \mathbf{0}, +\}|) = \mathcal{O}(3^{|C|})$. Consequently, this calls for an alternative pruning approach in which certain data are conscientiously unified.

This is where it becomes important that the clustering of $\mathcal{L}$ distinguishes clusters of SiDBs appropriately: stronger (intra-cluster) interactions are separated from the weaker (inter-cluster) ones. The proposed methodology takes advantage of the fact that, e. g., charge state assignments $[1 \mapsto -, 0 \mapsto \mathbf{0}]$ and $[0 \mapsto -, 1 \mapsto \mathbf{0}]$ influence SiDB 4 in Fig. 3 with a similarly-valued electrostatic interaction, since $d_{0,4} \approx d_{1,4}$. In fact, both charge state assignments may be unified to an assignment from the cluster $\{1, 2\}$ to the *multiset* $\{\!\{-, \mathbf{0}\}\!\}$, i. e., a set-like structure that keeps occurrence counts of its elements such that, e. g., $\{\!\{-, -\}\!\} \neq \{\!\{-\}\!\}$. Now, denoting the set of all *charge multisets* with k (not necessarily unique) charge states as $\mathcal{M}_k(\{-, \mathbf{0}, +\})$, a procedure that prunes charge multisets assigned to separate clusters C scales quadratically in $|C|$: $\mathcal{O}(|\{C\} \to \mathcal{M}_{|C|}(\{-, \mathbf{0}, +\})|) = \mathcal{O}(|C|^2)$.[3]

The effectiveness of pruning charge multisets has already been illustrated with Example 3, in which the multisets $\{\!\{-, -\}\!\}$, $\{\!\{-, \mathbf{0}\}\!\}$, $\{\!\{\mathbf{0}, +\}\!\}$, and $\{\!\{+, +\}\!\}$ can be pruned, leaving $\{\!\{-, +\}\!\}$ and $\{\!\{\mathbf{0}, \mathbf{0}\}\!\}$.

4) Refining Charge Spaces through Fixed Point Iteration: As suggested in Section IV-C2, potential bounds for an unpruned search space are determined from ranging over $\{-, \mathbf{0}, +\}$, i. e., the set of charge states that we *initially* assume to be attainable. Initially, that is, since a search

[1] Such a structure may be obtained through established methods [27].

[2] *QuickExact* looks for SiDBs that *must* be negatively charged in a separate procedure from finding those that *may* be positively charged.

[3] Note that for all sets A and $k \in \mathbb{N}$, $|\mathcal{M}_k(A)| = \binom{k+|A|-1}{k}$.

Algorithm 1: Ground State Space Construction

Input: An SiDB layout $\mathcal{L}$
Output: A minimized hierarchical ground state search space
1 $InitialClustering \leftarrow \{\{i\} \mid i \in \mathcal{L}\}$
2 $S \leftarrow \{(C, \{\{\!\{-\}\!\}, \{\!\{\mathbf{0}\}\!\}, \{\!\{+\}\!\}\}) \mid C \in InitialClustering\}$
3 **while** $S \neq \{(\mathcal{L}, \{\dots\})\}$ **do**
4 $\quad S' \leftarrow S$
5 $\quad$ **repeat** $S \leftarrow S'$; $S' \leftarrow$ RefineAllChargeSpaces(S)
6 $\quad$ **until** $S = S'$
7 $\quad S \leftarrow$ MergeAndFilter(S)
8 **return** S

space pruning procedure expressly aims to reduce the set of candidate charge states for all SiDBs through examining each candidate separately. With this perspective on the search space minimization that the state of the art employs, one realizes that it falls short of *iteration*; upon a reduction of the set of attainable charge states, the derived potential bounds are updated accordingly, which may enable further minimization of the simulation search space.

With the perceptual elevation from charge states assignments to SiDBs to charge multiset assignments to clusters, we arrive at *charge spaces*, which we define to be sets containing charge multisets that are attainable by some cluster. The construction stage starts with the initial clustering of only singleton clusters (Line 1 of Algorithm 1), and, premised on the lack of assumptions on $\mathcal{D}_\mathcal{V}$, assigns to each cluster the initial charge space $\{\{\!\{-\}\!\}, \{\!\{\mathbf{0}\}\!\}, \{\!\{+\}\!\}\}$ (Line 2 of Algorithm 1). This restructures the simulation search space $\mathcal{D}$, since each $D \in \mathcal{D}$ is represented in this initial state of the propose0d information structure: each charge state assignment is represented by the existence of a corresponding bijection between a cluster and an element of its charge space. I. e., for all $i \in \mathcal{L}$, $\{\!\{D(i)\}\!\}$ is an element of the initial charge space of singleton cluster $\{i\}$, raising the assignment $[i \mapsto \{\!\{D(i)\}\!\}]$. With this restructured search space, pruning to the extent that the state of the art adopts it is expressed through an initial refining of all charge spaces, adjusting potential bounds to any new constraints accordingly.

The starting point for iterative charge space refining is depicted in state ⓪ for the example layout in Fig. 3, and is regarded as the *initial information state*. For this example layout along with specified physical simulation parameters, it can be determined in the first refining step that cluster $\{4\}$ may not attain $\{\!\{+\}\!\}$, i. e., SiDB 4 cannot be positively charged in any charge distribution in $\mathcal{D}_\mathcal{V}$. This progresses the execution to information state ①, in which the potential bounds are refined accordingly. Precisely the *monotonicity* of the refining procedure gives rise to the constructive component of the proposed methodology: starting with no information, the information structure that ultimately becomes the minimized *ground state space* is built up dynamically from constraints on $\mathcal{D}_\mathcal{V}$ through successive pruning passes. When such a pruning pass fails to find a charge space element to exclude, the information state is said to have reached a *fixed point* (e. g., ② = ③ in Fig. 3), signaling the termination of the iteration. Lines 5–6 of Algorithm 1 describe this fixed point iteration.

5) Merging Charge Spaces Enables Efficacious Filtering: To enable charge multiset pruning for multi-SiDB clusters, the proposed methodology relies on a cluster hierarchy that reveals an ordering of the most strongly interacting (groups of) SiDBs. For the context of an algorithmic execution, this hierarchy can be thought of as the product of a successive transmutation of the initially all-singleton clustering of $\mathcal{L}$ through *merge* actions, until only the top cluster is left. Each merge operation takes the strongest interacting clusters of SiDBs together and merges them into a new cluster.[4]

In Fig. 3, these operations are performed at the information state transitions with the green merge icon (e. g., ② → ③). Here, it can be seen that the charge spaces—shown in their entirety for every cluster in every respective clustering associated with the successive information states—of the merged clusters $\{1\}$ and $\{2\}$ are merged through taking their "cross summation", forming the charge space of the merged cluster $\{1, 2\}$.[5] Note here that the constructive character of this stage of the proposed methodology makes a strong appearance: when merging charge spaces, the new charge space is directly constructed from the ones to merge; e. g., $\{\!\{+, +\}\!\}$ is never considered to be a charge space element of cluster $\{3, 4\}$ (⑥ → ⑦).

It is important to observe here that multisets in constructed charge spaces are *composed* out of multisets from respective charge spaces from the clusters that were merged, and, in particular, multiple multiset compositions may compose the same multiset.[6] Thus, charge multisets inherently have a compositional structure that is consistent with the *sub*hierarchy that is bounded by the associated cluster—this structure is unfolded in the destruction that is described in Section IV-D.[7]

Finally, the last procedure of the construction to describe is the *filter* step, which directly follows a merge step each time. Under fixed potential bounds derived from the respective charge spaces of clusters that were not merged with the previous merge, the new charge space is refined *in detail*: by checking each *composition* of each charge multiset that is being assessed, the proposed approach to exact simulation *selectively* taps into the exponential complexity of the simulation problem, making assessments based on detailed potential bounds information that is specific to the composition. After the charge space is filtered such that the elements for which all compositions were pruned are removed, the potential bounds for each element are *flattened* in order to mitigate the exponential information blow-up: the resulting bounds are determined through an inclusive unification of the respectively associated bounds for each composition of the charge multiset, thus compiling multiple potential bounds into one whilst remaining exact.

Line 7 of Algorithm 1 describes the entire process that transmutes a clustering through merging the most strongly interacting clusters, constructing a new charge space that is refined with the use of exponential information as described. In fact, the detailed filtering that happens here is *efficacious*: the strongest interactions yield the most variation in the potential bounds over different compositions of the charge multiset of which compositions are filtered, in turn hosting optimal likelihood of successful pruning and thus maximizing the use of the exponential information.[8]

D. Destruction: Extracting All Valid Charge Distributions

After the construction phase outlined in the previous section terminates when the top cluster charge space reaches a fixed point, a recursive information structure of hierarchically related charge spaces of decomposable charge multisets is returned: the *ground state space*. The goal of the destruction phase is to unfold this structure in all the ways in order to extract all physically valid charge distributions.

Starting with the compositions of the charge space elements of the top cluster, the destruction creates branches through recursive calls for each composition of a charge multiset that is being unfolded into the charge multisets that formed it, as seen in Lines 7–9 of Algorithm 2 which describes the recursive unfolding process described here. Thus, each branch *specializes* the information state of potential bounds, which is updated in such a way that the flattened potential bounds are now specialized to the specific composition of the destructed charge multiset (Line 8 of Algorithm 2).

Paramount to the efficiency of the proposed exact simulation method, however, are intermediary validity checks at each level of the recursion throughout this unfolding process. Using the information state that is specialized to the current *clustering state*—i. e., a clustering that associates one charge space element for each cluster in it—the algorithm assesses whether the clustering state may unfold into a physically valid charge distribution, terminating this branch of the recursion if the contrary is found to be true (Lines 1–2 of Algorithm 2). Alternatively, the recursion finds a base case when the clustering is the singleton clustering, at which

[4]The interaction strength between two clusters can be interpreted in various ways—Ward's method from [28] was used in the evaluation in Section V.

[5]To perform a cross summation of two sets of multisets A and B means to produce the set that contains the multisets that are formed by summing the respective occurrence counts of a and b, where $(a, b) \in A \times B$.

[6]E. g.: $(\{1\}, \{\!\{-\}\!\})$ merged with $(\{2\}, \{\!\{+\}\!\})$ becomes $(\{1, 2\}, \{\!\{-, +\}\!\})$, same as $(\{1\}, \{\!\{+\}\!\})$ merged with $(\{2\}, \{\!\{-\}\!\})$.

[7]E. g., $\{1\} \subset \{1, 2\} \supset \{2\}$ is a subhierarchy of the complete one in Fig. 3.

[8]When a charge multiset contains a charge state that cannot be assigned to any SiDB in the associated cluster under considered potential bounds, it is pruned. This is described formally in chapter 3 of [29].

Algorithm 2: Extract Metastable Charge Distributions

Input: A clustering state X, i.e., $X = \{(C_1, m_1), \ldots, (C_k, m_k)\}$ for some $1 \leq k \leq n$ where $\big(\forall (C, m) \in X \,.\, C \subseteq \mathcal{L} \wedge m \in \mathcal{M}_{|C|}(\{-, \mathbf{0}, +\})\big)$ and $\{C_i\}_{1 \leq i \leq k}$ is a clustering of $\mathcal{L}$

Output: The set of all physically valid charge distributions that can be extracted from X

1 **if** *potential bounds analysis for X yields that there are no physically valid charge distributions that can be extracted* **then**
2 **return** $\varnothing$
3 **if** $|X| = n$ **then** `// X is an all-singleton clustering state`
4 **return** ExtractChargeDistribution(X)
5 $(\bar{C}, \bar{m}) \leftarrow$ ChooseClusterToUnfold(X) `// e.g., largest`
6 $V \leftarrow \varnothing$ `// physically valid charge distributions`
7 **foreach** *Composition* $\in$ GetCompositions($\bar{C}, \bar{m}$) **do**
8 $X' \leftarrow$ replace $(\bar{C}, \bar{m})$ in X with *Composition* and specialize potential bounds accordingly
9 $V \leftarrow V \cup$ ExtractMetastableChargeDistributions(X')
10 **return** V

point all potential bounds correspond precisely to the local potentials for the charge distribution that is trivially extracted from the clustering state containing only singleton charge multisets (Lines 3–4 of Algorithm 2).

Concluding the description of the proposed methodology's course, the set of all valid charge distributions $\mathcal{D}_\mathcal{V}$ for a given SiDB layout $\mathcal{L}$ can now be computed: first the ground state space $S = \{(\mathcal{L}, M)\}$ for some top cluster charge space $M \in \mathcal{M}_n(\{-, \mathbf{0}, +\})$ is constructed using Algorithm 1, after which it is destructed into $\mathcal{D}_\mathcal{V}$ through $\mathcal{D}_\mathcal{V} \leftarrow \bigcup_{m \in M}$ ExtractMetastableChargeDistributions($\{(\mathcal{L}, m)\}$).

V. Experimental Evaluations

This section presents the results of an experimental evaluation of the methodology that was proposed in Section IV. First, Section V-A discusses the applied setup. Afterwards, two sets of results for different experiments are presented: 1) an experiment in Section V-B that quantifies the exponential base reduction when adopting the proposed methodology, and 2) an experiment in Section V-C that demonstrates the successful simulation of benchmark SiDB circuits from the literature. These layouts were previously far beyond the boundary of computational tractability for exact simulation; thus, the proposed work shifts this boundary from single-gates to full circuits—a crucial step in advancing SiDB technology.

A. Experimental Setup

The proposed methodology discussed in Section IV has been implemented in C++17 under the name *ClusterComplete*, and is built on top of the *fiction* [30] framework, which is part of the *Munich Nanotech Toolkit* (MNT).[9] The first experiment, which analytically investigates time complexity in the application to logic implementations on the SiDB platform, is structured as follows: a homogeneous chain of gates is incrementally extended, from a single gate up to a final chain of seven gates connected in series. For each chain length, an exact physical simulation of the layout is performed. This approach analyzes how runtime changes with the length of the chain, revealing how the runtime depends on the number of SiDBs in the layout and enabling extraction of the exponential base. This process is repeated for implementations of all 10 true 2-input Boolean functions plus wires and inverters. For the second experiment, SiDB circuits from the literature [9] comprising 6–7 gates are simulated using *ClusterComplete*. To enable a comparative perspective, the extrapolation results from the first experiment are applied to project runtimes of the state-of-the-art exact simulator *QuickExact*.

All evaluations were run on an Ubuntu 20.04 system equipped with an AMD Ryzen Threadripper PRO 5955WX processor (16 processor cores @ 4.00 GHz to 4.50 GHz) and 128 GB DDR4 RAM.

B. Extrapolation Base a in a^n for Exact SiDB Logic Simulations

All results of the first experiment are summarized in Table I. Two rows are displayed for each exact simulation algorithm that was tested,

[9] *ClusterComplete* is publicly available at https://github.com/cda-tum/fiction

TABLE I: Base a in a^n: *ClusterComplete* (CC) vs. SOTA (QE).

		Scaling Results Extrapolated from Homogeneous Chains of SiDB Gates												Mean
		WIRE	INV	AND	OR	NAND	NOR	XOR	XNOR	LT	LE	GT	GE	
QE	Base	1.705	1.822	1.941	1.949	2.333	2.327	1.935	2.323	1.867	1.898	1.943	1.876	**1.993**
	σ	0.158	0.155	0.044	0.041	0.330	0.334	0.047	0.339	0.090	0.074	0.044	0.085	0.141
CC	Base	1.276	1.296	1.326	1.318	1.310	1.317	1.335	1.361	1.332	1.285	1.354	1.297	**1.317**
	σ	0.013	0.002	0.000	0.002	0.002	0.003	0.002	0.002	0.004	0.003	0.001	0.007	0.003

TABLE II: Exact simulation of SiDB circuits using *ClusterComplete*.

Benchmark [9]			Physical Simulation Runtimes	
	Name	#SiDBs	*QuickExact* [22] (extrapolated)	*ClusterComplete*
[31]	xor5_majority	70	$\approx$ 2 million years	886 s
	xor5_r1	70	$\approx$ 2 million years	1985 s
[32]	HA	74	$\approx$ 31 million years	5833 s
	mux21	76	$\approx$ 127 million years	8434 s

providing the extrapolated base a in the exponential a^n—where n is the number of SiDBs—that was obtained through a linear regression on a log-scale. While up to seven gates were used to extrapolate the base for *ClusterComplete*, it was not feasible to simulate chains of more than three gates with *QuickExact*. As a result, the base extrapolation for the latter suffers from a higher standard error σ.

This experiment clearly demonstrates that the exponential base remains below 1.4 in all evaluated cases when using *ClusterComplete*, with some achieving a base as low as 1.276 and 1.285 in the case of the chain of WIRE gates and LE gates respectively—all whilst boasting a minuscule standard error. Meanwhile, the state of the art was only able to determine the absence of positive charges for these BDL gate implementations, thus yielding a scaling of 2^n.

The observed exponential base reduction—going from 2^n in state-of-the-art methods to around 1.3^n—represents a significant decrease in computational complexity. The groundbreaking approach masters the exponential nature of exact physical simulation of SiDBs and sets a new standard for computational feasibility in the SiDB technology.

C. SiDB Circuit Simulation

To show the significant impact of the reduced exponential base of the proposed methodology as quantified previously, the second experiment—outlined above in Section V-A—focuses on the simulation of realistic SiDB circuits from the literature. The simulation runtimes are summarized in Table II, where also approximated runtimes for *QuickExact* are provided through extrapolation using the data from the previous experiment.

Given the *magnitudinal* difference between the runtimes of the respective exact simulators for such larger problems, the benefit of the proposed methodology is obvious. This demonstrates its breakthrough power, overcoming the long-standing stagnation at single-gate simulation, and enabling the exact simulation of full circuits.

VI. Conclusion

As the field of *Silicon Dangling Bond* (SiDB) logic continues to advance, garnering increased attention as a promising beyond-CMOS technology, physical simulation remains a critical bottleneck—particularly for exact simulation. The exponential growth of the search space, scaling with base 3, has confined the application of exact physical simulation to single-gate designs, stalling progress toward larger, more complex layouts. To address this limitation, this work introduces a novel methodology that restructures the exponential search space according to a *hierarchical clustering*. The hierarchy structure enables a systematic pruning of the search space at its different levels, exploiting dynamically-inferred problem-specific constraints to reduce the effective exponential base from 3 to approximately 1.3.

The methodology achieves, for the first time, the exact physical simulation of multi-gate SiDB circuits. Experimental results using the proposed exact simulator *ClusterComplete* demonstrate its capability to handle complex layouts in minutes that would otherwise take the state of the art millions of years to simulate. Establishing a robust ground truth for the physical simulation of SiDB logic underscores the profound impact of *search space pruning in a cluster hierarchy* on overcoming the barrier imposed by the base 3 exponential complexity. This breakthrough paves the way for scalable, energy-efficient, and atomic-scale computing.

References

[1] A. Granskog, D. H. Diaz, J. Noffsinger *et al.*, "The Role of Power in Unlocking the European AI Revolution," *McKinsey & Company*, October 2024. [Online]. Available: https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/the-role-of-power-in-unlocking-the-european-ai-revolution

[2] T. Huff, H. Labidi, M. Rashidi *et al.*, "Atomic White-Out: Enabling Atomic Circuitry through Mechanically Induced Bonding of Single Hydrogen Atoms to a Silicon Surface," *ACS Nano*, 2017.

[3] J. Pitters, J. Croshaw, R. Achal *et al.*, "Atomically Precise Manufacturing of Silicon Electronics," *ACS nano*, vol. 18, pp. 6766–6816, 02 2024.

[4] R. Achal, R. A. Wolkow, J. Pitters *et al.*, "Lithography for editable atomic-scale devices and memories," US Patent 11,955,172, 2024.

[5] M. Rashidi, J. Croshaw, and R. A. Wolkow, "Automated Atomic Scale Fabrication," US Patent, 2022, 17/429,443.

[6] L. Livadaru, P. Xue, Z. Shaterzadeh-Yazdi *et al.*, "Dangling-bond Charge Qubit on a Silicon Surface," *New Journal of Physics*, 2010.

[7] S. Hofmann, M. Walter, and R. Wille, "Scalable Physical Design for Silicon Dangling Bond Logic: How a 45° Turn Prevents the Reinvention of the Wheel," in *2023 IEEE 23rd International Conference on Nanotechnology (NANO)*, 2023, pp. 872–877.

[8] S. S. H. Ng, J. Retallick, H. N. Chiu *et al.*, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *TNANO*, 2020.

[9] M. Walter, S. S. H. Ng, K. Walus *et al.*, "Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, 2022, pp. 739–744.

[10] M. Walter, J. Croshaw, S. S. H. Ng *et al.*, "Atomic Defect-Aware Physical Design of Silicon Dangling Bond Logic on the H-Si(100)-2×1 Surface," in *DATE*, 2024.

[11] S. Hofmann, J. Drewniok, M. Walter *et al.*, "MNT Designer: A Comprehensive Design Tool for Field-Coupled Nanocomputing," in *2025 IEEE 25th International Conference on Nanotechnology (NANO)*, 2025, pp. 40–45.

[12] B. Hien, M. Walter, S. Hofmann *et al.*, "A Fully Planar Approach to Field-Coupled Nanocomputing: Scalable Placement and Routing Without Wire Crossings," in *2025 IEEE 25th International Conference on Nanotechnology (NANO)*, 07 2025, pp. 649–654.

[13] M. Walter, J. Drewniok, and R. Wille, "The Operational Domain Explorer: A Comprehensive Framework to Unveil the Thermal Landscape of Silicon Dangling Bond Logic Beyond Conventional Operability," in *2025 IEEE 25th International Conference on Nanotechnology (NANO)*, 2025, pp. 28–33.

[14] J. Drewniok, M. Walter, and R. Wille, "QuickTrace: An Efficient Contour Tracing Algorithm for Defect Robustness Simulation of Silicon Dangling Bond Logic," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025, pp. 1–5.

[15] J. Drewniok, M. Walter, S. S. H. Ng *et al.*, "On-the-fly Defect-Aware Design of Circuits based on Silicon Dangling Bond Logic," in *2024 IEEE 24th International Conference on Nanotechnology (NANO)*, 2024, pp. 30–35.

[16] ——, "Unifying Figures of Merit: A Versatile Cost Function for Silicon Dangling Bond Logic," in *2024 IEEE 24th International Conference on Nanotechnology (NANO)*, 2024, pp. 91–96.

[17] S. Hofmann, M. Walter, and R. Wille, "Physical Design for Field-coupled Nanocomputing with Discretionary Cost Objectives," in *2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)*, vol. 1, 2025, pp. 1–5.

[18] R. A. Wolkow, R. Achal, T. Huff *et al.*, "Multiple Silicon Atom Quantum Dot and Devices inclusive thereof," US Patent 10,937,959, 2021.

[19] R. A. Wolkow, M. Rashidi, W. Vine *et al.*, "Initiating and Monitoring the Evolution of Single Electrons within Atom-defined Structures," US Patent 11,047,877, 2021.

[20] S. S. H. Ng, M. Walter, J. Drewniok *et al.*, "Building a Machine Learning Accelerator with Silicon Dangling Bonds: From Verilog to Quantum Dot Layout," in *2025 IEEE 25th International Conference on Nanotechnology (NANO)*, 2025, pp. 483–488.

[21] T. Huff, T. Dienel, M. Rashidi *et al.*, "Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot," *ACS Nano*, 2019.

[22] J. Drewniok, M. Walter, and R. Wille, "The Need for Speed: Efficient Exact Simulation of Silicon Dangling Bond Logic," in *ASP-DAC*, 2024.

[23] ——, "Temperature Behavior of Silicon Dangling Bond Logic," in *2023 IEEE 23th International Conference on Nanotechnology (NANO)*, 2023, pp. 925–930.

[24] T. Huff, H. Labidi, M. Rashidi *et al.*, "Binary Atomic Silicon Logic," *Nature Electronics*, 2018.

[25] J. Drewniok, M. Walter, and R. Wille, "Minimal Design of SiDB Gates: An Optimal Basis for Circuits Based on Silicon Dangling Bonds," in *NANOARCH*, 2023.

[26] S. S. H. Ng, "Computer-aided Design of Atomic Silicon Quantum Dots and Computational Applications," Master's thesis, University of British Columbia, 2020.

[27] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[28] J. H. Ward, "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[29] W. Lambooy, "Hierarchically Bounded Composable Interactions: Applications and Extensions to Atom-Scale Logic Design," Master's thesis, Radboud University Nijmegen, 2024.

[30] M. Walter, R. Wille, F. S. Torres *et al.*, "*fiction*: An Open Source Framework for the Design of Field-coupled Nanocomputing Circuits," 2019.

[31] A. Trindade, R. Ferreira, J. Nacif *et al.*, "A Placement and Routing Algorithm for Quantum-dot Cellular Automata," in *SBCCI*, 2016.

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