

QuickSim: Efficient and Accurate Physical Simulation of Silicon Dangling Bond Logic

Jan Drewniok¹, Marcel Walter², Samuel Sze Hang Ng³, Konrad Walus⁴, and Robert Wille⁵

Abstract—Silicon Dangling Bonds have established themselves as a promising competitor in the field of beyond-CMOS technologies. Their integration density and potential for energy dissipation advantages of several orders of magnitude over conventional circuit technologies sparked the interest of academia and industry alike. While fabrication capabilities advance rapidly and first design automation methodologies have been proposed, physical simulation effectiveness has yet to keep pace. Established algorithms in this domain suffer either from exponential runtime behavior or subpar accuracy levels. In this work, we propose a novel algorithm for the physical simulation of Silicon Dangling Bond systems based on statistical methods that offers both a time-to-solution and an accuracy advantage over the state of the art by more than one order of magnitude and a factor of more than three, respectively, as demonstrated by an exhaustive experimental evaluation.

I. INTRODUCTION & MOTIVATION

Recent years have seen an increase in scientific interest in the *Silicon Dangling Bond* (SiDB) logic platform—an emerging computational beyond-CMOS nanotechnology [1]–[5]. Of particular significance are its sub-nanometer elementary devices that allow for an integration density improvement of several orders of magnitude over current CMOS fabrication nodes [1], [2], [4]–[7], [9]; and its properties to compute Boolean logic without the flow of electrostatic current but, instead, via the Coulombic repulsion of charges [2], [6]. By this, SiDB logic promises ultra-low energy dissipation and establishes itself as a highly-anticipated green competitor in the beyond-CMOS domain [6], [9]–[11]. Moreover, it has been proposed as a candidate for the integration of quantum computers [12], and—since SiDBs are fabricated on silicon as well—with conventional CMOS circuitry [5].

Motivated by this, the scientific community has already proposed gate and circuit libraries [13]–[19] as well as design automation solutions [13], [15], [20], [21] for SiDB logic. But also commercially, this technology gains more and more interest as confirmed, e.g., by the recently founded *Quantum Silicon Inc.* which aspires to be among the first industry adopters of SiDBs and already secured multi-million dollar investments for this purpose [22], [23]. However,

the rapid advancement of SiDB fabrication capabilities that goes along with these developments [1]–[4] puts pressure on both design and simulation tools to keep pace. Particularly physical simulation, the foundation of system validation, poses a challenge. To reliably predict the behavior of any given system of n SiDBs, a high-dimensional optimization problem must be solved to capture all physically relevant effects. In its essence, 3^n charge configurations have to be enumerated, checked for physical feasibility, and their individual energies simulated such that the lowest of those can be picked—constituting a highly non-trivial (in fact, exponential) problem.

Thus far, only two approaches addressing this problem have been proposed in the literature: *ExhaustiveGS* (ExGS) [14], [17], which performs an exhaustive search of the entire (exponential) search space, and *SimAnneal* [13], which employs probabilistic sampling and, hence, offers a heuristic approach that trades runtime for completeness. Accordingly, ExGS is severely limited in its runtime efficiency, while SimAnneal only provides an approximation and, hence, is not guaranteed to determine an optimal (i.e., perfectly accurate) solution.

In this paper, we propose a novel algorithm (*QuickSim*) that aims to be both efficient *and* accurate. To this end, we are making use of effective search space pruning by incorporating ideas from statistical methods like *max-min diversity distributions* [24]. Exhaustive experimental evaluations (covering the simulation of common logic operations, established gate libraries, as well as random instances to demonstrate the performance on future layouts) confirm that, the resulting approach leads to a time-to-solution advantage of more than one order of magnitude. At the same time, it outperforms SimAnneal in terms of solution accuracy by more than a factor of four on randomly-generated SiDB layouts and by roughly a factor of three on established gate layouts.

The remainder of this work is structured as follows: In an effort to establish this paper as a self-contained article, Section II reviews preliminaries on SiDB systems. Section III discusses existing simulation approaches. Section IV introduces *QuickSim* in detail. Section V comprises of a case study that highlights the benefits of *QuickSim* over the state of the art. Finally, Section VI concludes the paper.

II. PRELIMINARIES

This section first provides an overview of the SiDB logic platform. Afterward, the physical simulation of SiDBs is elaborated on.

¹Jan Drewniok is with the Chair for Design Automation, Technical University of Munich, Germany. Email: jan.drewniok@tum.de

²Marcel Walter is with the Chair for Design Automation, Technical University of Munich, Germany. Email: marcel.walter@tum.de

³Samuel Sze Hang Ng is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada. Email: samueln@ece.ubc.ca

⁴Konrad Walus is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada. Email: konradw@ece.ubc.ca

⁵Robert Wille is with the Chair for Design Automation, Technical University of Munich, Germany, and Software Competence Center Hagenberg GmbH, Austria. Email: robert.wille@tum.de

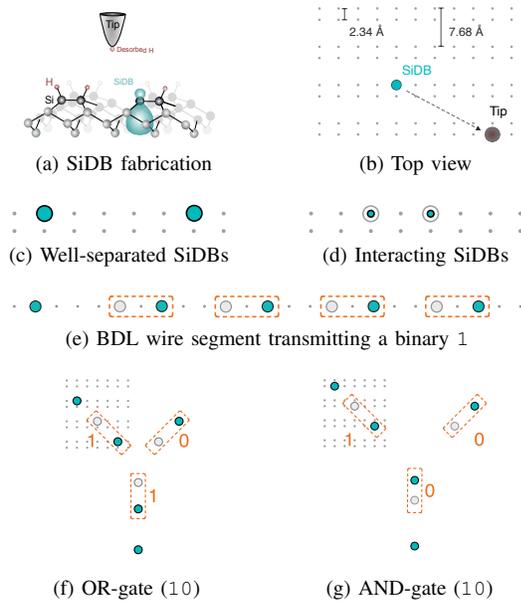


Fig. 1: Elemental SiDB logic components

A. The SiDB Logic Platform

SiDBs are fabricated on a *Hydrogen-passivated Silicon* (H-Si(100)-2×1) surface. Using the atomically-sharp probe of a *Scanning Tunneling Microscope* (STM), a voltage can be applied that breaks the bond between silicon and hydrogen with atomic precision [2]–[4], [25]–[27]. The split-off hydrogen atom is then desorbed to the probe and leaves behind an open, i.e., dangling, valence bond; an SiDB as illustrated in Fig. 1a. A top view on Fig. 1a is schematically shown in Fig. 1b, where the teal-colored dot represents the SiDB. It also illustrates the basic principle of the fabrication process of an SiDB layout where, after the first SiDB is generated, the tip moves to a new Si-dimer to desorb the next hydrogen atom to generate a second SiDB. Each SiDB created this way acts as a chemically-identical quantum dot that can confine a maximum of two electrons due to its sp^3 -orbital [28]. Thus, each SiDB is either negatively, neutrally, or positively charged, i.e., in precisely one of three different states.

Moreover, the SiDBs' charge transition energy levels lie within the bandgap: intuitively, this means that their states are stable unless disturbed by the presence of other charges [8], [29]. In n-doped systems with a near-surface depletion region, SiDBs are able to maintain their quantum dot property but tend to be negatively charged due to the raised Fermi energy. This precise property is exploited in order to use SiDBs in the creation of logic elements such as gates [6]. In fact, SiDBs can be created in pairs of metastable configurations such that the state of one SiDB depends on and influences the other's via band bending; a concept coined *Binary-Dot Logic* (BDL) [6].

Fig. 1b to Fig. 1g depict top-down views of SiDBs drawn as teal halos on an H-Si surface, where hydrogen atoms are visualized by gray dots. In Fig. 1c, two SiDBs are created with a spacing of ≈ 2 nm such that their interaction is

limited and both assume a negative charge state. In contrast, SiDBs that are placed with just a single dimer in between, as illustrated in Fig. 1d, strongly interact and thus share a single electron in superposition due to their symmetry if not externally excited. However, in the presence of an external charge, the electron will assume the position with maximum distance to said charge to minimize the system's total electrostatic potential energy. This concept is illustrated in Fig. 1e, where an additional SiDB is placed on the left and acts as a *perturber* that inflicts Coulombic pressure on the first BDL pair, which then does the same to the next BDL pair. Thus, a sequence of BDL pairs acts as a binary wire that transmits information exclusively via the repulsion of electrical fields [6]. The same concept can be utilized to realize, e.g., an OR-gate as shown in Fig. 1f (here, illustrated using the binary input 10; the repulsion leads to output 1). Both, an SiDB wire consisting of eight SiDBs as well as an OR-gate (total size smaller than 30 nm^2) were already experimentally realized in a lab by Huff *et al.* [6]. Additionally, further gates can be realized using this concept as, e.g., the AND-gate shown in Fig. 1g.

B. Physical Simulation

To explore this new emerging technology and to validate designs of novel circuits, the physical simulation of SiDB layouts is of immense interest. To predict the state that will be assumed by an arbitrary arrangement, i.e., a layout, of multiple SiDBs, an electrostatic potential simulation must be conducted. The electrostatic potential $V_{i,j}$ at position i generated by an SiDB in state $n_j \in \{-1, 0, 1\}$ at position j is given by [6], [26]

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

where λ_{TF} defines the *Thomas-Fermi screening length* and ϵ_r the *dielectric constant* which were experimentally extracted to be 5 nm and 5.6, respectively [6]. Moreover, 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 position i and j , respectively. As can be seen from Eq. (1), all SiDB states influence each other and, by this, form an interwoven system. A layout's total electrostatic potential energy E is then calculated by using the superposition principle of the electrostatic potential, i.e.:

$$E = -\sum_{i < j} V_{i,j} \cdot n_i \cdot q_e \quad (2)$$

The expected state of an SiDB layout is given by the metastable charge configuration with the lowest energy. Therefore, in order to determine the said state, Eq. (2) has to be minimized—yielding a high-dimensional optimization problem. On top of that and in order to guarantee a valid simulation result, the physical constraint of *metastability* must be met. Metastability can be described by the combination of two criteria, namely *population stability* and *configuration stability*. Both must be obeyed to guarantee simulation result validity.

a) *Population Stability*: The charge state of each SiDB must be consistent with the energetic charge transition levels μ_- and μ_+ relative to the Fermi energy. The energy of the SiDB's charge transition level at position i depends on the local electrostatic potential $V_{local,i}$ at that position defined by:

$$V_{local,i} = \sum_{j,j \neq i} V_{i,j} \quad (3)$$

Intuitively, this means that an SiDB is preferably neutrally charged when adjacent SiDBs are negatively charged and vice versa. This relation is formally expressed by the following conditions: SiDB- when $\mu_- + V_{local,i} \cdot q_e < 0$, SiDB+ when $\mu_+ + V_{local,i} \cdot q_e > 0$, and SiDB0 otherwise.

b) *Configuration Stability*: If there is no feasible single-electron hop event between arbitrary SiDBs leading to a lowering of the system's total electrostatic potential energy, the charge state fulfills the configuration stability. In other words, if this constraint is not satisfied, there would exist a state of lower electrostatic potential energy that the system would spontaneously converge to.

Overall, the charge configuration with the lowest electrostatic potential system energy that is also physically valid, i. e., satisfies metastability, is called the system's *ground state*. It represents the charge configuration of a given SiDB system at low temperature and, by this, its physical behavior. Hence, the goal of any physical simulation technique is to determine the said ground state for a given SiDB layout.

III. MOTIVATION

In this section, SiDB simulation approaches from the literature are reviewed first. Afterward, the obstacles that arise when attempting to balance the runtime and accuracy of these algorithms are discussed.

A. Previously Proposed Simulation Approaches

As outlined above, the physical simulation of SiDB logic poses a high-dimensional optimization problem that, due to its exponential complexity, is computationally intractable. Since simulation constitutes a core capability of any design automation flow for SiDB-based logic, managing this complexity is a key priority. However, only a few automatic solutions for the physical simulation of SiDBs exist so far, and they are either of high runtime/poor scalability (due to the exponential search space) or of a rather poor accuracy (due to aggressive approximations). Two methods represent the current state of the art: *ExhaustiveGS* (ExGS) [14], [17] and *SimAnneal* [13].

1) *ExGS*: Since every SiDB can either be negatively, neutrally, or positively charged, a layout consisting of n SiDBs can exhibit 3^n unique charge configurations. ExGS enumerates all of them, determines if they are physically valid, and, if this is the case, computes the electrostatic potential energy of the layout via Eq. (2). This eventually allows obtaining the charge configuration with minimal energy.

Since ExGS covers the entire search space exhaustively, it is guaranteed to always determine the system's ground state. At the same time, however, this requires the traversal of the entire exponential search space and, hence, even in the best case takes an exponential runtime to do so. Thus, ExGS is practically applicable to small layouts only.

2) *SimAnneal*: In contrast to ExGS, SimAnneal is a heuristic algorithm that generates approximate results. As the name suggests, it employs the strategy of *simulated annealing* [30], which is a probabilistic technique for approximating the global optimum of a given objective function which, in this case, is the electrostatic potential energy of closely-placed SiDBs. SimAnneal is a powerful algorithm that is capable of correctly determining the SiDBs' charge distribution of many layouts. Moreover, since SimAnneal is a physically-inspired algorithm, it inherently captures the charge configuration of SiDBs. On the downside, it is a probabilistic heuristic that requires many simulation steps to result in the ground state. Especially, when many charge configurations lead to a similar electrostatic potential system energy close to the ground state, long runtimes are required.

B. Trade-off between Efficiency and Accuracy

As elaborated on above, the goal of physical simulation algorithms for SiDB layouts is to determine the system's ground state. Probabilistic heuristics like SimAnneal usually run a number of times on the same input hoping that the ground state is determined at least once. To evaluate their performance in doing so, their required runtime could be used as a measure. However, in probabilistic algorithms, *single runtime* t_s and *result accuracy* $p_s(t_s)$ are strongly correlated, i. e., more single runs of, e. g., SimAnneal, increase the likelihood of determining a layout's ground state and vice versa. Hence, single runtime alone is not a meaningful metric as accuracy should also be considered. This trade-off is captured by the *Time-To-Solution* (TTS) metric that provides the *total* runtime needed to derive a solution of a certain confidence level η [31]. More precisely, TTS is defined by:

$$TTS(t_s) = t_s \cdot R(t_s), \text{ with} \quad (4)$$

$$R(t_s) = \begin{cases} \frac{\ln(1-\eta)}{\ln(1-p_s(t_s))} & p_s(t_s) \in (0, \eta) \\ 1 & p_s(t_s) \in [\eta, 1] \end{cases} \quad (5)$$

ExGS constitutes the extreme case of this trade-off: The single runtime is maximal (as the *entire* search space is traversed), but the accuracy obtained is 100% (since all possibilities are considered). Thus, ExGS' TTS is large compared to SimAnneal's. In contrast, SimAnneal's accuracy is lower than ExGS', whereas its single runtime is smaller by several orders of magnitude. Since the single runtime influences the accuracy, small runtimes lead to a small accuracy and vice versa. Therefore, an ideal trade-off between these two attributes is desired to yield an efficient simulation. Overall, neither of the existing solutions—ExGS or SimAnneal—provide such a satisfactory trade-off between efficiency and accuracy. In this work, we address this shortcoming.

IV. PROPOSED ALGORITHM: *QuickSim*

In this section, we propose a novel algorithm for physical simulation of SiDB layouts. We start by sketching the general idea on a more abstract level. Afterward, we go into more detail and present the algorithm implementing the proposed idea.

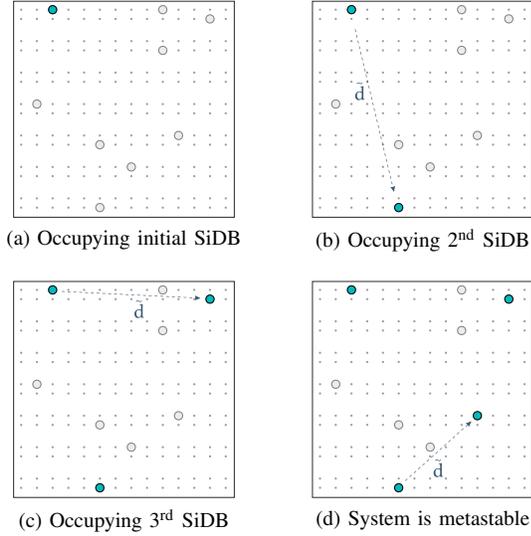


Fig. 2: Illustration of the general idea ($\mu_- = -0.15$ eV)

A. General Idea

The proposed algorithm *QuickSim* for physical SiDB simulation aims to provide a significantly improved trade-off between single runtime and result accuracy. To this end, we are making use of effective search space pruning by incorporating ideas from statistical methods like *max-min diversity distributions* [24]. This leads to a new simulation algorithm that works fundamentally differently than the state of the art. More precisely, while previously proposed approaches alter the given system’s electron number several times during one simulation run, considering *all* possible charge configurations in the case of ExGS or effectively randomly guessing configurations in the case of SimAnneal, we propose to iteratively and purposefully distribute electrons on the basis of insights provided by physical principles. Given the electrons’ tendency to always occupy SiDBs that are placed the farthest distance away from one another (to minimize their electrostatic potential energy), we can infer that any given number of electrons must settle in an equally-spaced allocation across the SiDBs. In other words, the electrostatic potential energy of a system is *minimized* when the electron distribution’s diversity is *maximized*. We can create such max-min diversity distributions by iteratively filling up placed SiDBs with electrons starting from the neutral state such that each newly introduced charge has a maximal distance from all others. Conducting this iteration once with every SiDB as the starting point and, along the way, computing both the system’s electrostatic potential energy and metastability, we obtain an algorithm that is less probabilistic and less sensitive to hyperparameters than SimAnneal. Additionally, it requires fewer simulation steps to achieve the same accuracy, which considerably reduces its TTS.

B. Implementation Details

The details of *QuickSim* are exemplified given the pseudocode from Alg. 1 and 2 and visualized in Fig. 2. The algorithm starts with an empty electron distribution D , i. e.,

Algorithm 1: Physical Simulation

Input: SiDB Layout L comprised of n dangling bonds
Input: Physical parameters $P = \{\mu_-, \mu_+, \lambda_{tf}, \epsilon_r\}$
Input: Number of cycles c
Output: Electron distribution D and system energy E

```

1  $E^* \leftarrow \infty$ 
2  $D^* \leftarrow [D_0 = 0, \dots, D_n = 0]$ 
3 for  $cycle \leftarrow 1$  to  $c$  do
4   foreach  $db \in L$  do
5      $D \leftarrow [db = -1]$ 
6     for  $i \leftarrow 1$  to  $n - 1$  do
7        $D \leftarrow \text{ADJACENT-SEARCH}(L, D)$ 
8        $E \leftarrow \text{energy of } L \text{ given } D \text{ and } P$  // Eq. (2)
9       if  $D$  is phys. valid given  $P$  and  $E < E^*$  then
10         $D^* \leftarrow D$ 
11         $E^* \leftarrow E$ 
12        break
13      end if
14    end for
15  end foreach
16 end for
17 return  $(D^*, E^*)$ 

```

Algorithm 2: Adjacent-Search

Input: Partial electron distribution D
Output: Extended electron distribution D'

```

1  $d^* \leftarrow \{\min_{p \in D} \text{dist}(u, p) \mid u \in L \setminus D\}$ 
2  $\tilde{d} \leftarrow \max_{d \in d^*} d$ 
3  $S \leftarrow \{u \mid u \in L \setminus D, d_u^* \geq \tilde{d} \cdot \alpha\}$ 
4 Randomly select one  $db' \in S$ 
5 return  $D \cup [db' = -1]$ 

```

all SiDBs in the given layout L are set to be neutrally charged (Alg. 1, Line 2). For each SiDB in the layout as the starting point (Alg. 1, Line 4), electrons are successively distributed while maximizing diversity. To this end, the initial SiDB db is assigned an electron, i. e., flipped to a negative charge (Alg. 1, Line 5). From there, Alg. 2 is iteratively called (Alg. 1, Line 7) to determine other SiDBs with maximum distance to all negatively charged ones (Alg. 2, Lines 1–2). These are enumerated within a threshold given by the value α (Alg. 2, Line 3). While α is represented as a variable factor between 0 and 1, the algorithm was found to perform best when picking $\alpha = 0.7$. Out of the possible candidates, one SiDB is randomly selected (Alg. 2, Line 4) and assigned an electron (Alg. 2, Line 5). The newly resulting electron distribution D in the layout L is utilized to compute its electrostatic potential energy E under the physical parameters P , i. e., $\mu_-, \mu_+, \lambda_{tf}$ and ϵ_r (Alg. 1, Line 8). If D is physically valid under P (cf. Section II-B) and if E is smaller than the best energy value found so far (Alg. 1, Line 9), this constitutes a better candidate for the overall simulation result, thus, both D and E are kept as current best (Alg. 1, Lines 10–11). This entire procedure is repeated c times (Alg. 1, Line 3) and the best result found is returned (Alg. 1, Line 17).

V. EXPERIMENTAL EVALUATIONS

To demonstrate the applicability and performance of *QuickSim* in comparison to the state of the art, we implemented *QuickSim* and conducted exhaustive experimental evaluations. This section summarizes and analyzes our findings. To this end, we first describe the experimental setup. Afterward, we provide the respectively obtained results

from experimental simulations considering three different classes of SiDB layouts, namely 1) randomly-generated, 2) machine learning (ML)-generated, and 3) established ones from the literature. Finally, the obtained results are discussed.

A. Experimental Setup

In order to properly evaluate the trade-off between efficiency and accuracy as well as to showcase how *QuickSim* overcomes this, we performed *Time-To-Solution* (TTS, as defined in Section III-B) evaluations. To this end, we considered the two state-of-the-art physical simulation algorithms (ExGS and SimAnneal as reviewed in Section III-A) as well as *QuickSim*. As benchmarks, we considered a variety of SiDB layout classes whose details are reviewed in the following section. To determine $p_s(t_s)$ as introduced in Section III-B, the simulations have been conducted 10 000 times and the success number (of correct simulation results) was collected. Thus, $p_s(t_s) = \#success/10\,000$. For the whole evaluation, the confidence level has been set to $\eta = 99.7\%$.

For comparisons, the SimAnneal implementation as provided by *SiQAD* [13] was utilized, whereas ExGS and *QuickSim* are custom implementations. All code has been implemented in C++ on top of the *fiction*¹ framework [32] which is part of the *Munich Nanotech Toolkit* (MNT) and compiled with AppleClang 14.0.0. For convenient applicability, we made *QuickSim* available on GitHub as an open-source plugin for SiQAD.² Additionally, all obtained experimental data is also publicly accessible.³ All experiments were performed on a macOS 12.6 machine with an Apple Silicon M1 Pro SoC with 32 GB of integrated main memory.

Since SimAnneal offers a range of hyperparameters that can be tuned to a problem instance at hand, we considered the TTS data for SimAnneal in two configurations: *default*, i. e., as set out-of-the-box, and *optimized*, i. e., manually tuned settings for improved TTS on the considered benchmark set. Thus, *QuickSim*'s performance can be compared against a setting that provides an even better simulation performance than supplied by [13]. It can be concluded that reducing the number of annealing cycles by one order of magnitude from 10 000 (def.) to 1 000 (opt.), results in optimized TTS values—providing a highly optimized baseline against which we compare *QuickSim* in this work. For both ExGS and *QuickSim* we apply default parameters, for ExGS that entails using two-state simulation and for *QuickSim* $\alpha = 0.7$ and $c = 80$.

B. Considered Layouts and Obtained Results

In the first series of evaluations, we considered randomly-generated gate layouts. While the simulation of readily available gate libraries has more practical value (and, hence, is considered as well below), this allows demonstrating the performance of the respective simulations for potential future gate libraries showcasing the broad applicability. To this end, we considered the 2-in-1-out and 2-in-2-out-templates that have recently been proposed in [15] and are considered promising SiDB candidates for fabrication. Our benchmark

TABLE I: Physical simulation of randomly-generated layouts

| BENCHMARK | | | PHYSICAL SIMULATION | | | | | | |
|--------------|---------|--------|---------------------|---------|----------------|-----------|-------|-------------|-----------------|
| | | | EXGS [17] | | SIMANNEAL [13] | | | | <i>QuickSim</i> |
| Name | #SiDBs | #inst. | TTS | default | | optimized | | default | |
| | | | | acc. | TTS | acc. | TTS | acc. | TTS |
| 2-1 | 20 – 23 | 114 | 172.31 | 40.6 | 29.59 | 25.8 | 11.43 | 82.6 | 1.18 |
| 2-2 | 21 – 25 | 114 | 452.09 | 16.1 | 363.52 | 11.9 | 28.72 | 79.7 | 1.73 |
| <i>Mean</i> | | | | 28.4 | | 18.9 | | 81.2 | |
| <i>Total</i> | | | 624.40 | | 393.11 | | 40.15 | | 2.91 |

TABLE II: Physical simulation of ML-generated layouts

| BENCHMARK | | | PHYSICAL SIMULATION | | | | | | |
|--------------|---------|--------|---------------------|---------|----------------|-----------|-------|-------------|-----------------|
| | | | EXGS [17] | | SIMANNEAL [13] | | | | <i>QuickSim</i> |
| Name | #SiDBs | #inst. | TTS | default | | optimized | | default | |
| | | | | acc. | TTS | acc. | TTS | acc. | TTS |
| OR | 19 – 23 | 68 | 2680.08 | 59.8 | 12.52 | 38.9 | 10.23 | 86.0 | 0.54 |
| AND | 25 | 25 | 2086.82 | 18.1 | 27.59 | 15.0 | 21.47 | 66.4 | 1.60 |
| NOR | 20 – 23 | 20 | 1312.83 | 26.0 | 10.55 | 15.4 | 2.72 | 99.6 | 0.05 |
| NAND | 20 – 23 | 16 | 1300.58 | 45.7 | 7.79 | 37.3 | 22.12 | 94.9 | 0.07 |
| CX | 29 | 4 | 577.00 | 21.5 | 21.45 | 13.2 | 12.52 | 29.0 | 0.56 |
| HA | 31 | 4 | 433.91 | 17.8 | 20.03 | 15.1 | 3.98 | 17.5 | 4.45 |
| FO2 | 27 | 2 | 65.42 | 21.4 | 0.78 | 22.8 | 0.12 | 23.3 | 0.38 |
| XOR | 21 – 23 | 5 | 4.49 | 30.0 | 2.09 | 32.4 | 0.28 | 95.0 | 0.02 |
| XNOR | 25 | 4 | 2.97 | 20.4 | 6.54 | 25.0 | 8.94 | 72.8 | 0.16 |
| <i>Mean</i> | | | | 29.0 | | 23.9 | | 65.0 | |
| <i>Total</i> | | | 8464.07 | | 109.34 | | 82.38 | | 7.83 |

set contains 114 random implementations for each template with a range of SiDBs per layout from 20 to 25. In a second series of evaluations, we considered SiDB layouts generated using the machine learning approach presented in [20]. In contrast to the randomly-generated instances, these implement common logic operations such as AND, OR, NAND, NOR, etc. Hence, these instances provide a representative use case for physical simulation applied in the evaluation of new gate libraries. This benchmark set contains a total of 148 novel SiDB layouts. As above, we simulated them with all three approaches and compared their TTS. To this end, each gate has been simulated in up to five configurations: with input perturbers specifying one of the possible 2^n input combinations as well as without any input perturbers at all. However, some ML-generated gates are very sensitive to a change in physical parameters and sometimes do not work correctly for all input combinations (00, 01, 10, 11). Hence, a further series of evaluations are conducted. In a third series of evaluations, all 12 gate layouts from [15] are simulated. They represent a library of established gates in the SiDB domain. The simulation is conducted identically to the one in the evaluation of the ML-generated layouts.

All obtained results are summarized in Table I, Table II, and Table III for the randomly-generated, ML-generated, and established layouts, respectively. In each table, the first three columns provide the name of the benchmark, their SiDB range, and the number of simulated layouts (instance count). Afterward, the accuracy and the accumulated TTS values (in sec.) for all considered simulation approaches (and their configurations) are stated. Since ExGS is an exact algorithm and, thus, always returns the ground state, the accuracy is not listed in the table to avoid redundancy. Furthermore, to ensure a sophisticated evaluation of *QuickSim*, TTS gained with optimized settings is collected in addition to the TTS of SimAnneal in default settings as discussed before providing a highly optimized baseline against which we compare *QuickSim*. In the tables' last two rows, the mean value of the

¹<https://github.com/cda-tum/fiction>

²<https://github.com/cda-tum/mnt-siqad-plugins>

³<https://github.com/cda-tum/mnt-quick-sim-results>

TABLE III: Physical simulation of established gate layouts

| BENCHMARK [15] | | | PHYSICAL SIMULATION | | | | | | |
|----------------|---------|--------|---------------------|------|----------------|------|----------|-------------|-------------|
| Name | #SiDBs | #inst. | EXGS [17] | | SIMANNEAL [13] | | QuickSim | | |
| | | | TTS | acc. | TTS | acc. | TTS | acc. | |
| Double Wire | 28 – 30 | 5 | 1149.21 | 2.3 | 35.80 | 1.9 | 4.87 | 65.2 | 0.30 |
| CX | 27 – 29 | 5 | 618.88 | 17.5 | 21.00 | 14.5 | 13.07 | 64.0 | 0.25 |
| HA | 24 – 26 | 5 | 54.82 | 2.8 | 4.27 | 2.2 | 0.38 | 96.3 | 0.04 |
| AND | 21 – 25 | 5 | 24.31 | 31.1 | 2.33 | 20.2 | 1.49 | 90.5 | 0.03 |
| XOR | 21 – 23 | 5 | 5.65 | 38.0 | 2.13 | 24.2 | 0.42 | 96.0 | 0.02 |
| OR | 21 – 23 | 5 | 5.57 | 61.4 | 0.71 | 41.8 | 0.14 | 97.7 | 0.01 |
| XNOR | 21 – 23 | 5 | 5.56 | 36.0 | 1.99 | 34.3 | 0.13 | 90.7 | 0.03 |
| FO2 | 20 – 21 | 3 | 1.28 | 67.2 | 0.26 | 48.2 | 0.05 | 98.9 | 0.01 |
| NOR | 19 – 21 | 5 | 1.20 | 40.2 | 1.35 | 23.0 | 0.81 | 95.0 | 0.02 |
| NAND | 19 – 21 | 5 | 1.19 | 56.9 | 0.67 | 38.3 | 0.13 | 99.7 | 0.01 |
| INV | 18 – 19 | 6 | 0.53 | 78.1 | 0.35 | 52.7 | 0.09 | 83.6 | 0.01 |
| Wire | 15 – 17 | 6 | 0.08 | 84.0 | 0.24 | 39.0 | 0.13 | 86.9 | 0.00 |
| <i>Mean</i> | | | | | | | | 88.7 | |
| <i>Total</i> | | | 1868.28 | 43.0 | 71.10 | 28.4 | 21.71 | | 0.73 |

accuracies and the sums of the individual accumulated TTS are collected, respectively.

C. Discussion

All obtained results confirm the discussions from Section III-B: ExGS obtains the ground state simulation result in a single run, but also requires the largest runtime. SimAnneal is faster but, even after a tedious tuning of the number of annealing cycles for SimAnneal to minimize TTS, a substantial amount of iterations are still required and accuracy is far from satisfactory. Overall: ExGS has high TTS and SimAnneal low accuracy. In contrast, QuickSim achieves simulation performances that excel in both accuracy and TTS—outperforming ExGS and SimAnneal on all three evaluations. While, in comparison to SimAnneal, QuickSim’s simulation accuracy is increased by more than a factor of three, its TTS is improved by more than one order of magnitude at the same time because its increased accuracy does not negatively impact its runtime. Compared to ExGS, its TTS is reduced by several orders of magnitude, while the simulation accuracy is only slightly lower, again emphasizing the overcoming of the trade-off between efficiency and accuracy.

VI. CONCLUSIONS

The recent advancement in the design automation and fabrication capabilities of *Silicon Dangling Bonds* (SiDBs) as an emerging computational beyond-CMOS technology was fueled by scientific and commercial interest. In order to keep pace with the progress made in this area, physical simulation is a key tool as it poses the foundation of any design validation workflow.

In this work, the novel algorithm *QuickSim* for physical simulation of SiDB layouts was proposed that outperforms the current state-of-the-art algorithm SimAnneal by more than one order of magnitude and more than a factor of three in terms of time-to-solution (TTS) and accuracy, respectively. These results were revealed by an exhaustive experimental evaluation on three different sets of SiDB layouts: 1) randomly generated, 2) machine learning-generated, and 3) established ones from the literature, which constitutes a total of 436 test cases. In an effort to support open research and open data, we made the algorithm’s implementation, all evaluation data, and the test scripts publicly available. Future work aims to further improve the performance of the simulator,

as well as to add other features such as the consideration of temperature (which, as recently explored in [33], plays an important role in this technology).

REFERENCES

- [1] M. B. Haider *et al.*, “Controlled Coupling and Occupation of Silicon Atomic Quantum Dots at Room Temperature,” *Physical Review Letters*, 2009.
- [2] 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.
- [3] N. Pavliček *et al.*, “Tip-Induced Passivation of Dangling Bonds on Hydrogenated Si(100)-2×1,” *Applied Physics Letters*, 2017.
- [4] R. Achal *et al.*, “Lithography for Robust and Editable Atomic-Scale Silicon Devices and Memories,” *Nature Communications*, 2018.
- [5] R. A. Wolkow *et al.*, “Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics,” 2014.
- [6] T. Huff *et al.*, “Binary Atomic Silicon Logic,” *Nature Electronics*, 2018.
- [7] X. Wang *et al.*, “Atomic-Scale Control of Tunneling in Donor-Based Devices,” *Communications Physics*, 2020.
- [8] J. L. Pitters *et al.*, “Charge Control of Surface Dangling Bonds Using Nanoscale Schottky Contacts,” *ACS Nano*, 2011.
- [9] M. Rashidi *et al.*, “Initiating and Monitoring the Evolution of Single Electrons within Atom-defined Structures,” *Physical Review Letters*, 2018.
- [10] C. S. Lent *et al.*, “Bennett Clocking of Quantum-dot Cellular Automata and the Limits to Binary Logic Scaling,” *Nanotechnology*, 2006.
- [11] G. Toth *et al.*, “Quasiadiabatic Switching for Metal-Island Quantum-Dot Cellular Automata,” *Journal of Applied Physics*, 1999.
- [12] A. S. Dzurak *et al.*, “Construction of a Silicon-Based Solid State Quantum Computer,” *Quantum Information & Computation*, 2001.
- [13] S. S. H. Ng *et al.*, “SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits,” *TNANO*, 2020.
- [14] M. D. Vieira *et al.*, “Three-Input NPN Class Gate Library for Atomic Silicon Quantum Dots,” *IEEE Design & Test*, 2022.
- [15] M. Walter *et al.*, “Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic,” in *DAC*, 2022.
- [16] H. N. Chiu *et al.*, “PoisSolver: A Tool for Modelling Silicon Dangling Bond Clocking Networks,” *IEEE-NANO*, 2020.
- [17] S. S. H. Ng, “Computer-Aided Design of Atomic Silicon Quantum Dots and Computational Applications,” Master’s thesis, University of British Columbia, 2020.
- [18] H. N. Chiu, “Simulation and Analysis of Clocking and Control for Field-coupled Quantum-dot Nanostructures,” Master’s thesis, University of British Columbia, 2020.
- [19] A. N. Bahar *et al.*, “Atomic Silicon Quantum Dot: A new Designing Paradigm of an Atomic Logic Circuit,” *TNANO*.
- [20] R. Lupoiu *et al.*, “Automated Atomic Silicon Quantum Dot Circuit Design via Deep Reinforcement Learning,” *arXiv*, 2022.
- [21] M. Walter *et al.*, “Scalable Design for Field-coupled Nanocomputing Circuits,” in *ASP-DAC*, 2019.
- [22] R. A. Wolkow *et al.*, “Multiple Silicon Atom Quantum Dot and Devices inclusive thereof,” US Patent 10937959, 2021.
- [23] R. Wolkow *et al.*, “Initiating and Monitoring the Evolution of Single Electrons Within Atom-Defined Structures,” US Patent 11047877, 2021.
- [24] M. G. Resende *et al.*, “GRASP and Path Relinking for the Max—Min Diversity Problem,” *Computers & Operations Research*, 2010.
- [25] M. Rashidi *et al.*, “Automated Atomic Scale Fabrication,” US Patent, 2022, 17/429,443.
- [26] T. R. Huff *et al.*, “Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot,” *ACS Nano*, 2019.
- [27] R. Achal *et al.*, “Detecting and Directing Single Molecule Binding Events on H-Si (100) with Application to Ultradense Data Storage,” *ACS Nano*, 2019.
- [28] M. Taucer, “Silicon Dangling Bonds Non-equilibrium Dynamics and Applications,” 2015.
- [29] M. Rashidi *et al.*, “Time-Resolved Single Dopant Charge Dynamics in Silicon,” *Nature Communications*, 2016.
- [30] S. Kirkpatrick *et al.*, “Optimization by Simulated Annealing,” *Science*, 1983.
- [31] T. Albash *et al.*, “Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing,” *Physical Review X*, 2018.
- [32] M. Walter *et al.*, “fiction: An Open Source Framework for the Design of Field-coupled Nanocomputing Circuits,” 2019.
- [33] J. Drewniak *et al.*, “Temperature Behavior of Silicon Dangling Bond Logic,” in *IEEE-NANO*, 2023.