Operational Domain Sketching for Silicon Dangling Bond Logic

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

***Abstract*— *Silicon Dangling Bond* (SiDB) logic is an innovative post-CMOS computing technology that operates at the atomic scale, delivering unmatched energy efficiency. However, despite its transformative potential, SiDB logic encounters a significant challenge: its sensitivity to material imperfections and variations in the physical properties of the H-Si(100)-2×1 surface on which it is fabricated, which can render gates non-operational. To address this sensitivity, the concept of the *Operational Domain* has gained traction in the literature—a range of physical parameter combinations that ensure reliable gate operation. Physical simulations are a standard approach to determining whether a gate is operational for given physical parameters. However, evaluating numerous parameter points can result in exponential scaling. While existing methods aim to reduce the number of simulator calls, they still rely on physical simulations, leading to inconvenient runtime. Motivated by that, we propose an algorithm that efficiently identifies non-operational parameter points without relying on physical simulations, while assuming operability for the remaining points. Since this approach might yield false positives, we call the resulting plot the *Operational Domain Sketch*, an approximation of the operational domain. Experimental analysis shows that the sketch can be obtained with runtime improvements of up to 365x compared to the state of the art, providing a fast method for determining the robustness of SiDB logic. This, in turn, facilitates the design of robust gates and enables reliable SiDB circuits.**

I. Introduction & Motivation

AI is currently reshaping various industries and attracting significant attention. As more people integrate AI tools into their daily lives, this trend is accompanied by increased energy consumption in data centers. This growing demand for energy could become a bottleneck that hinders the widespread adoption of AI tools [1]. To address this challenge, two approaches can be considered: improving software efficiency to achieve the same results with less training of the AI models, or developing new hardware architectures that are more energy-efficient [2], [3]. One promising candidate for the latter is *Silicon Dangling Bond* (SiDB) logic, which leverages the electrostatic interactions of SiDBs on the H-Si(100)-2×1 surface. Operating at the atomic scale and offering significantly higher energy efficiency than CMOS, this technology has the potential to revolutionize computing and establish a new paradigm. Consequently, substantial efforts are being made to develop dedicated design automation solutions and physical simulators to advance this technology [4]–[8]. Moreover, the SiDB technology has attracted attention not only in academia but also in industry, where companies such as *Quantum Silicon Inc.* are working to commercialize SiDB logic [9]–[12].

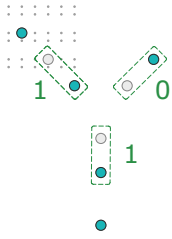
Jan Drewniok is with the Chair for AI Processor Design, Technical University of Munich, Munich, Germany. Marcel Walter and Robert Wille are with the Chair for Design Automation, Technical University of Munich, Munich, Germany, and also with the Munich Quantum Software Company GmbH, Garching near Munich, Germany. Robert Wille is also with the Software Competence Center Hagenberg GmbH, Hagenberg im Mühlkreis, Austria. E-mail: {jan.drewniok, marcel.walter, robert.wille}@tum.de

Despite significant advances in the SiDB technology, variations in material properties on the H-Si(100)-2×1 surface during material preparation, as well as atomic defects, remain significant challenges [13], [14]. Even minor deviations from the design parameters can render a previously functional gate non-operational [7], [15]. This issue becomes particularly problematic when the *Operational Domain*—a range of physical parameter combinations that ensure reliable gate functionality—is small. Therefore, determining this domain is crucial for assessing gate robustness and serves as an essential metric in gate design. To compute the operational domain, exact and approximate methods have been proposed in the literature [7], [15]: 1) *grid search* offers an exact method for mapping the entire operational domain. For every parameter point, physical simulations verify whether the gate fulfills the desired logic for all 2^n input combinations, where n is the number of inputs. While n is typically smaller than 3, this does not pose significant challenges [15], [16]. However, a major drawback of grid search is its quadratic dependence on the number of physical simulator calls in two-dimensional operational domains. Each of these simulator calls can scale exponentially in the worst case with 3^d, where d is the number of SiDBs [8]. 2) Approximate methods, such as *random sampling*, *flood fill*, and *contour tracing*, aim at enhancing efficiency by reducing the number of physical simulator calls [15]. In fact, it has been shown that the number of simulator calls is reduced by $95\,\%$ for contour tracing compared to grid search. However, this approach still requires physical simulations, and in general, these methods face significant challenges when the operational islands are small, difficult to identify, or fragmented across multiple islands.

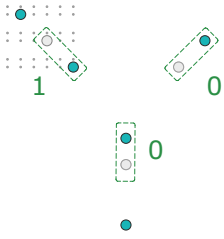
In scenarios where such challenges arise, and an exact computation of the operational domain is required, grid search remains the de facto method. However, given its limitations, there is an urgent need for a radical new approach. Motivated by that, we propose an algorithm that efficiently identifies non-operational parameter points without relying on physical simulations, while assuming operability for the remaining points. Since this approach might yield false positives, we call the resulting plot the *Operational Domain Sketch*, an approximation of the operational domain. Experimental analysis shows that the sketch can be obtained with runtime improvements of up to 365x compared to the state of the art, providing a fast method for determining the robustness of SiDB logic. This, in turn, facilitates the design of robust gates and enables reliable SiDB circuits.

To make this paper self-contained, Section II reviews SiDBs, explains their application in constructing binary

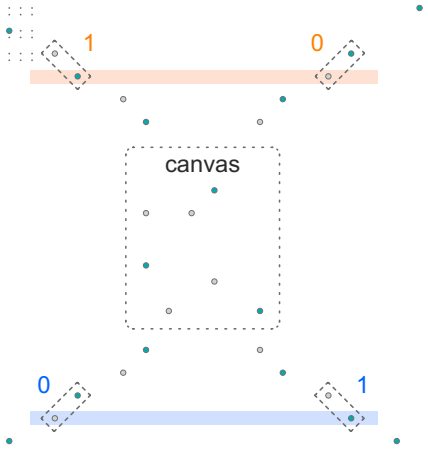
(a) An SiDB wire consisting of three BDL pairs and one single one (perturber) to propagate the binary `1`.



(b) OR gate for the input pattern `10` [14], [17].



(c) AND gate for the input pattern `10` [17].



(d) CX gate for input pattern `10` from the *Bestagon* SiDB gate library [18].

Fig. 1: The SiDB logic platform.

logic, and provides an overview of the operational domain, a fundamental concept for the remainder of this paper. Afterwards, Section III reviews existing approaches from the literature for computing the operational domain. It highlights that all these approaches depend on physical simulation and identifies their shortcomings. In response, Section IV introduces an algorithm that efficiently generates the operational domain sketch—an approximation of the operational domain—without relying on physical simulation. Section V presents an experimental evaluation demonstrating the efficiency of the proposed algorithm. Finally, the paper concludes with a summary in Section VI.

II. Preliminaries

This section lays the foundation for the remainder of this paper. First, Section II-A introduces SiDBs, beginning with their fabrication process and concluding with their application in constructing logic. Second, Section II-B reviews and explains the operational domain, a key concept in the SiDB technology.

A. The SiDB Logic Platform

While the natural formation of SiDBs is often random and difficult to control, they can be intentionally created with atomic precision on the H-Si(100)-2×1 surface using an atomically sharp tip of a scanning tunneling microscope [14], [19]. Positioning the tip near the surface and applying a voltage pulse can break the bond between the silicon atom and its hydrogen atom, causing the hydrogen atom to desorb and adhere to the tip, leaving behind a precisely positioned SiDB.

When examining the charge states of SiDBs, it has been observed that they can exist in three distinct states, depending on the number of electrons they host [14], [19]. This, in turn, is influenced by the local electrostatic potential at the SiDB: moderately strong electrostatic interactions ($> -0.9\,\mathrm{V}; < -0.3\,\mathrm{V}$) may leave the SiDB neutral, while strong interactions ($< -0.9\,\mathrm{V}$) can render it positively charged [17], [20]. This unique property of variable charge states is leveraged to construct a *Binary-dot Logic* (BDL) pair, consisting of two SiDBs that share one additional electron. In this configuration, one SiDB is neutral, while the other is negatively charged. The BDL pair effectively represents a binary digit smaller than $1.0\,\mathrm{nm}$, with the bit state encoded in the location of the additional electron. By arranging multiple BDL pairs close together, where electric field interactions remain significant, a BDL wire can be created to propagate information [14]. When a single SiDB, which is negatively charged unless placed too close to the wire, is positioned near the wire, it generates Coulombic pressure, resulting in a binary `1`. For this reason, such an SiDB is referred to as a *perturber*.

Example 1. *In Fig. 1a, a BDL wire is depicted, consisting of three BDL pairs and one perturber on the left, causing all pairs to encode a* `1`*, with the bit information determined by the charge distribution. Similarly, the process can be reversed by placing a perturber SiDB on the right side, which applies Coulombic pressure to set all pairs to encode a* `0`*.*

When combining BDL pairs in a Y-shape, gates can be realized as demonstrated in the literature via physical simulation and experimental measurements [17], [19].

Example 2. *Fig. 1d illustrates the crossing (CX) gate (crosses two signals) from the state-of-the-art* Bestagon *SiDB gate library [18] for the input pattern* `10`*. The gate consists of two inputs and two outputs, each comprising two BDL pairs. The area between the pins, enclosed by dashed lines and referred to as the* canvas *in the literature, contains 7 SiDBs. As intended, the* `0` *and* `1` *are directed to the diagonally opposite outputs, respectively. SiDB gates are typically designed, as in this example, using standardized I/O pins and placing the SiDBs in specific arrangements within the canvas to achieve the desired logic.*

Despite significant advancements in manufacturing capabilities and the development of dedicated physical design and simulation tools such as *fiction* [21] and *SiQAD* [17], SiDB logic remains highly sensitive to electrostatic interactions. Minor variations in the material properties governing these interactions can render a gate non-operational [15]. Therefore, evaluating the resilience of a gate to such variations is essential for its design and fabrication. To this end, the concept of the operational domain has been introduced in the literature [7], which is described next.

B. Operational Domain

The operational domain encompasses all physical parameter combinations for which an SiDB layout L implements a given Boolean function f. Specifically, let $\mathcal{P}$ represent the physical parameters under investigation. Additionally, $v_{L,f} : \mathcal{P} \rightarrow \{\mathsf{operational}, \mathsf{non\text{-}operational}\}$ is a function that indicates whether a given SiDB layout L for a given parameter point $p \in \mathcal{P}$ is a valid implementation of f ($v_{L,f} = \mathsf{operational}$) or not ($v_{L,f} = \mathsf{non\text{-}operational}$). To evaluate $v_{L,f}$, typically, a physical simulation is conducted to determine the layout's charge distribution under the given parameters, and therefore, the encoded logic [15]. Using these definitions, the operational domain can be defined as follows:

$$\mathit{OpDom} \coloneqq \{p \in \mathcal{P} \mid v_{L,f}(p) = \mathsf{operational}\}.$$

Several algorithms have been proposed in the literature to determine *OpDom*, which are briefly reviewed in the next section.

III. Related Work

As previously discussed, the operational domain is critical for evaluating the robustness of SiDB logic against variations of physical parameters before fabrication. In [7], the concept of the operational domain was first introduced and analyzed for various SiDB gates. The grid-search-based method determines the operational domain by evaluating $v_{L,f}(p)$ using physical simulations for each parameter point in the parameter space $\mathcal{P}$. While this approach is accurate, a major drawback of grid search is its reliance on a quadratic number of physical simulator calls in two-dimensional operational domains. Each of these simulator calls can scale exponentially with 3^d in the worst case, where d represents the number of SiDBs. To reduce the number of simulator calls, various approximate methods have been proposed in the literature [15]: random sampling, flood fill, and contour tracing.

a) Random Sampling: This approach selects random samples from the parameter space and runs a physical simulation for each one.

b) Flood Fill: The approach begins with random sampling to identify parameter points where the layout is operational. Once such a point is found, a flood fill algorithm is applied. This involves running simulations for all adjacent parameter points, gradually expanding outward. The expansion continues iteratively until the edge of the non-operational domain is reached.

c) Contour Tracing: This approach starts similarly to flood fill, by sampling random parameter points within the parameter space. For each operational point, the operationality of parameter points along a line to the left is examined sequentially until a non-operational point is encountered. The contour is then traced using a Moore neighborhood search, a well-established method that examines up to 8 neighboring points (the Moore neighborhood) for each contour point.

While these methods can identify operational domains with up to $95\,\%$ fewer simulator calls [15], they have a significant limitation: for layouts with small or disconnected operational domains, they are at risk of not reconstructing the domain in its entirety due to their reliance on an initial random sampling process.

Example 3. *In Fig. 2, the operational domain in the $(\epsilon_r, \lambda_{tf})$-space is shown for $\epsilon_r \in [1, 10]$ and $\lambda_{tf} \in [1, 10]$, with a resolution of 0.01 in each direction. This domain is computed for the CX gate from the state-of-the-art* Bestagon *SiDB gate library using grid search, flood fill, and contour tracing described before [15], [18]. In Fig. 2a, the operational domain is computed via grid search with a total number of $811\,801$ samples. This reveals two operational islands: one near the center and another near the top-right around $(\epsilon_r = 8, \lambda_{tf} = 10\,\mathrm{nm})$. In Fig. 2b, flood fill is used with up to 2000 initial random samples. Despite the large sample size, it only detects the top-right operational island. Finally, in Fig. 2c, contour tracing is employed, again with up to 2000 random samples. Although it identifies the central operational island, the second operational island remains undetected. This highlights the risks and limitations of algorithms that attempt to identify operational parameter points through initial random sampling, such as flood fill and contour tracing.*

This paper proposes an approach that, for the first time, efficiently determines an approximation of the operational domain (operational domain sketch) without relying on physical simulation. The details of this approach are provided in the following section.

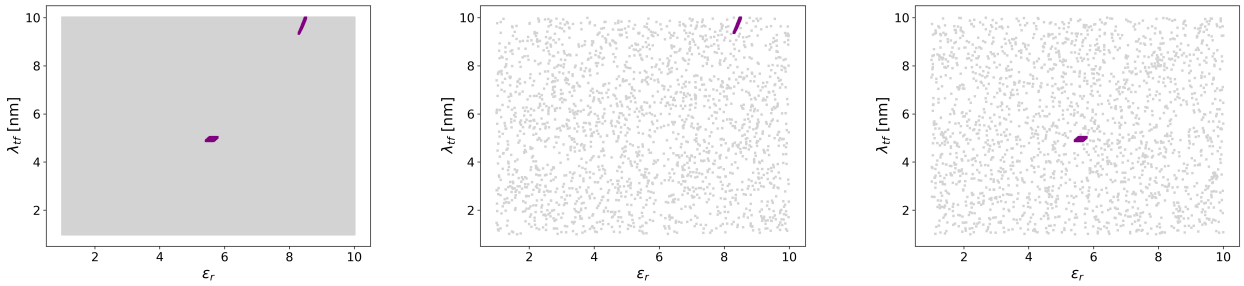
IV. The Proposed Algorithm

This section constitutes the main contribution of this paper by presenting the general idea and the details of the proposed algorithm. First, the general idea of how to determine the operational domain sketch without relying on physical simulator calls is outlined in Section IV-A. Second, the implementation details are described in Section IV-B.

A. General Idea

We employ a series of filtering techniques that can identify non-operational parameter points without simulation, while ensuring no false negatives. In other words, every parameter point deemed non-operational by the filtering methods is indeed non-operational. All parameter points that are not detected as non-operational by the filtering strategies are considered as operational. Since this approach might produce some false positives, we call the resulting plot an *Operational Domain Sketch*, an approximation of the operational domain. In the following, the general idea of the three strategies is explained:

a) Positive Charge Detection: Implementing SiDB logic using positively charged SiDBs has never been experimentally demonstrated, making such configurations undesirable. Positively charged SiDBs typically arise in environments with strong electrostatic interactions [17]. These conditions are often linked to regions characterized by low ϵ_r values, high λ_{tf} values, or when SiDBs are positioned in close proximity to one another. To evaluate whether positively charged SiDBs can occur, the local electrostatic potential of each SiDB is examined under the assumption that all SiDBs are in a negative charge state. This assumption maximizes the local electrostatic potential for each SiDB



(a) Grid search with 811 801 samples (step size 0.01).

(b) Flood Fill with up to 2000 initial random samples.

(c) Contour Tracing with up to 2000 initial random samples.

Fig. 2: Operational Domain Computation for the *Bestagon* CX gate [18] with $\epsilon_r, \lambda_{tf} \in [1, 10]$, and $\mu_- = -0.32\,\text{eV}$ using grid search, flood fill, and contour tracing. Purple and gray present operational and non-operational parameter points, respectively.

and hence constitutes the "best case" for the occurrence of positive charges. If the potential at any SiDB exceeds approximately $-0.9\,\text{V}$—the threshold for positively charged SiDBs as described in Section II-A—then the occurrence of positively charged SiDBs is possible. Otherwise, positively charged SiDBs are deemed unlikely. Thus, this provides an efficient way to determine whether positively charged SiDBs can occur at all without conducting physical simulations, thereby indicating whether the layout is operational or non-operational at the given parameter point.

b) I/O Pins Failure Detection: When physical parameters deviate from their default values, I/O pins may no longer function as intended. This can happen if strong electrostatic interactions cause both SiDBs in a BDL pair to become neutrally charged, or if weak interactions cause both SiDBs to become negatively charged. To assess whether the electrostatic interaction at a given parameter point leads to a successful BDL pair function, the following procedure is used: For each input pattern, the charge states of the BDL pairs of the I/O pins are set according to the intended logic. The system is then checked to determine whether any charge distribution among the canvas SiDBs satisfies the underlying physical model. The I/O pins will not function as expected if no valid distribution is found, and thus, the gate is non-operational.

c) I/O-signals Instability Detection: Even if no I/O pin failures are detected in the previous step, it does not guarantee that the pins are carrying the desired signals. This is because an SiDB layout typically includes both ground and excited states, and the previous step cannot distinguish between them. As a result, the absence of detected failures does not ensure that the ground state encodes the correct signals. To address this, it is crucial to verify signal stability. This is done by flipping the signals at the I/O pins and checking whether it reduces the system's energy. If it does, the system would remain in the flipped state, resulting in a logic failure and indicating that the signals are unstable. However, not all signal instabilities can be captured by this check. For example, kinks inside the I/O pins stay undetected, leading to false positives in the operational domain. For this reason, we refer to the set of detected operational parameter points as the operational domain sketch.

If a layout for a given parameter point successfully passes all three filtering steps, it is deemed operational for that specific parameter point.

B. Implementation Details

The details of the proposed algorithm are outlined in Algorithm 1, using the $(\epsilon_r, \lambda_{tf})$-space as an example[1]. However, the approach works for arbitrary parameter spaces. The process begins with the Boolean function f that the given SiDB layout L is intended to implement as input. Additionally, the parameter ranges for ϵ_r and λ_{tf} are given by R_{ϵ_r} and $R_{\lambda_{tf}}$, respectively.

First, the operational domain sketch is initialized as an empty set (Line 1). Then, all possible parameter combinations within the $(\epsilon_r, \lambda_{tf})$-space, defined by $R_{\epsilon_r} \times R_{\lambda_{tf}}$, are enumerated. Next, it is determined for each parameter point, whether the SiDB layout L can be declared non-operational using the three proposed strategies outlined above (Line 2). Afterward, it is checked (Line 3) whether the layout for the given physical parameter p could result in positively charged SiDBs as explained in Section IV-A. If they can occur, p is not added to the operational domain sketch, and the layout is investigated for the next parameter point p (Line 5).

The second step evaluates whether the failure of I/O pins for a given p can be detected for any of the 2^n input combinations (Line 10), as explained in Section IV-A. If it turns out that the I/O pins do not work, the parameter point p is labeled as non-operational, and the layout is investigated for the next parameter point p (Line 11).

Finally, the algorithm verifies if the I/O-signals are instable (Line 13) as explained in Section IV-A. If the signals are unstable, p is considered as non-operational, and again, the layout is investigated for the next parameter point p (Line 14).

If L successfully passes all three filtering steps for p, the layout is deemed operational for this parameter point and p is added to the operational domain sketch accordingly (Line 17). After checking all parameters, the operational domain sketch is returned (Line 19).

[1]The details of the working principle and implementation can be found at https://github.com/cda-tum/fiction.

Algorithm 1: Operational Domain Sketch (ϵ_r, λ_{tf})

Input: Boolean function $f : \mathbb{B}^n \to \mathbb{B}^m$ to implement
Input: SiDB layout L
Input: Physical simulation parameters P
Input: Parameter range $R_{\epsilon_r} = x_1, \ldots, x_k$
Input: Parameter range $R_{\lambda_{tf}} = y_1, \ldots, y_l$
Output: Operational Domain Sketch ODS

```
1  ODS ← ∅
2  foreach p ∈ R_εr × R_λtf do
3      foreach i = 0 ... 2^n − 1 do
4          if positive charges occur then
5              goto Line 2 and continue with next p
6          end if
7      end foreach
8      foreach i = 0 ... 2^n − 1 do
9          apply input pattern i to L
10         if I/O pin failure is detected then
11             goto Line 2 and continue with next p
12         end if
13         if I/O-signals are instable then
14             goto Line 2 and continue with next p
15         end if
16     end foreach
17     ODS ← ODS ∪ p
18 end foreach
19 return ODS
```

V. Experimental Evaluations

To demonstrate the efficiency of the algorithm proposed in this work, an experimental evaluation is conducted. First, the experimental setup is described in Section V-A. Second, in Section V-B, the results are presented, which are afterward discussed in Section V-C.

A. Experimental Setup

Algorithm 1 has been implemented in C++17 and integrated into the *fiction* framework [21], which is part of the *Munich Nanotech Toolkit* (MNT, [22]). All code was compiled with AppleClang 15.0.7, and the experiments were run on a macOS 15.2 machine with an Apple Silicon M1 Pro SoC with 32 GB of integrated main memory.

The established *Bestagon* SiDB gate library was used from the literature for the experimental evaluation [18]. For each gate, the operational domain in the $(\epsilon_r \in [1, 10], \lambda_{tf} \in [1, 10])$-space for $\mu_- = -0.32\,\text{eV}$ was computed with the proposed operational domain sketch and with grid search using a step size of 0.05 in each dimension. *QuickExact* was used as the physical simulator for grid search. The number of operational parameter points and the runtime were collected for both algorithms.

B. Obtained Results

The results of the experiment are summarized in Table I. The first and second columns, "Name" and "#SiDBs", specify the name of the layout and the number of SiDBs of the layout. Subsequently, the simulation results for the operational domain are presented, comparing the grid search approach with the proposed method.

The 3rd and 4th columns ("$\#op_{\text{sota}}$" and "$t_{\text{sota}}[\text{s}]$") indicate the number of detected operational parameter points and the corresponding runtime for the state-of-the-art (SOTA) approach. Similarly, the 5th and 6th columns ("$\#op_{\text{sketch}}$" and "$t_{\text{sketch}}[\text{s}]$") provide the same metrics for the proposed algorithm. Finally, the 7th and 8th columns ("$\#op_{\text{sota}}/\#op_{\text{sketch}}$" and "$t_{\text{sota}}/t_{\text{sketch}}$") describe the overlap between the operational domains of the grid search approach and the proposed method, along with the runtime ratio between the two methods. The table ends with a row that summarizes the average of "$\#op_{\text{sota}}/\#op_{\text{sketch}}$", and the sum of "$t_{\text{sota}}[\text{s}]$" and "$t_{\text{sketch}}[\text{s}]$" of all gates.

C. Discussion

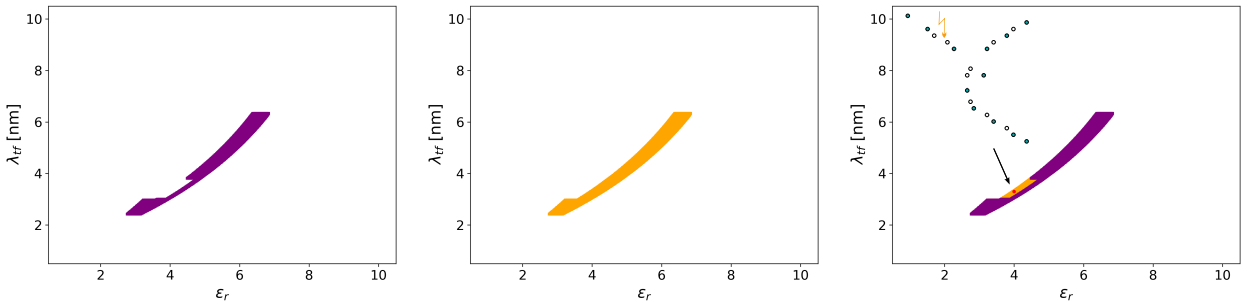
The evaluation reveals several points, which are discussed in the following: The proposed algorithm shows significantly improved runtimes. For example, while the state-of-the-art method requires $5.60\,\text{s}$ to reconstruct the operational domain of the *Bestagon* AND gate, the proposed sketching algorithm reduces this time to only $0.25\,\text{s}$, resulting in a runtime improvement by a factor of about ≈ 23. For more complex gates like CX, HA, and DOUBLE WIRE, the runtime improvement is even more pronounced. While the state-of-the-art method requires $733.18\,\text{s}$ for the DOUBLE WIRE, the proposed algorithm completes the task in only $2.00\,\text{s}$, yielding a remarkable runtime improvement of a factor of about 365 compared to grid search.

Since the proposed approach is an approximate method to determine the operational domain, the overlap ratio $\#op_{\text{sota}}/\#op_{\text{sketch}}$ is 1.19 on average, indicating that the sketch is $19\,\%$ larger than the domain. Notably, for the XOR gate, the sketch is twice as large as the domain, with a ratio of $\#op_{\text{sota}}/\#op_{\text{sketch}} = 2.03$. However, for most gates, the sketch aligns closely with the operational domain. Importantly, the sketch aligns perfectly with the actual domain for complex gates like the CX and HA gates.

Example 4. *Fig. 3 illustrates the operational domain and the sketch for the* Bestagon *AND gate. Fig. 3a shows the operational domain obtained via grid search, with the purple samples representing operational parameter points. Fig. 3b displays the operational domain sketch generated by the proposed approach, with orange samples representing the operational parameter points identified by the approach. The sketch closely resembles the operational domain. Finally, Fig. 3c presents both the operational domain and the sketch in a single plot. It can be observed that only one region of the parameter space yields different outcomes. While the sketch labels this region as operational, it is actually non-operational. A closer inspection of the charge distribution of the AND gate with the input pattern* `10` *reveals that, at this specific parameter point (red dot), the gate exhibits a kink in the left input pin. This is, as mentioned in Section IV-A, a false positive, since it is incorrectly labeled as operational by the proposed approach. This scenario highlights why the proposed approach's outcome is referred to as a "sketch".*

VI. Conclusion

Silicon Dangling Bond (SiDB) logic stands out as a promising post-CMOS candidate, offering atomic-level operation and exceptional energy efficiency. However, its sensitivity to material imperfections poses a critical barrier to reliable gate functionality. To tackle this issue, the concept of the *Operational Domain* was introduced, serving as a key metric for evaluating gate robustness. While existing methods aim to reduce the number of simulator calls, they still rely on physical simulations, leading to inconvenient



(a) Operational Domain. (b) Operational Domain Sketch. (c) Stacked Visualization.

Fig. 3: Comparison of the operational domain and sketch for the *Bestagon* AND gate [18], where purple and orange represent operational parameter points determined by the respective methods.

TABLE I: Comparison of the state-of-the-art grid search and the proposed operational domain sketch approach. $\#op$ denotes the number of identified operational parameter points, and $t[\mathrm{s}]$ represents the runtime in seconds for each method, respectively.

Benchmark			Operational Domain Simulation				Ratio	
	Name	#SiDBs	$\#op_{\text{sota}}$	$t_{\text{sota}}[\mathrm{s}]$	$\#op_{\text{sketch}}$	$t_{\text{sketch}}[\mathrm{s}]$	$\#op_{\text{sota}}/\#op_{\text{sketch}}$	$t_{\text{sota}}/t_{\text{sketch}}$
Bestagon [18]	WIRE straight	16	5449	0.52	7706	0.17	1.41	2.98
	WIRE diagonal	17	2486	0.92	2486	0.22	1.00	4.26
	INV straight	19	1634	1.29	1664	0.44	1.02	2.95
	INV diagonal	19	511	1.48	725	0.50	1.42	2.99
	AND	23	593	5.60	686	0.25	1.16	22.70
	NAND	21	505	1.76	531	0.18	1.05	9.54
	OR	23	743	4.77	743	0.22	1.00	21.82
	NOR	21	858	1.60	1368	0.17	1.59	9.24
	XOR	23	563	5.67	1144	0.25	2.03	22.77
	XNOR	23	529	6.33	529	0.24	1.00	25.99
	FAN-OUT	21	228	5.33	228	0.19	1.00	28.15
	CX	29	22	151.88	22	0.60	1.00	252.43
	HA	26	111	32.62	111	0.24	1.00	136.05
	DOUBLE WIRE	30	952	733.18	967	2.00	1.02	365.72
Total				952.95		5.68	1.19	

runtime. Motivated by that, we proposed an algorithm that efficiently identifies non-operational parameter points without relying on physical simulations, while assuming operability for the remaining points. Since this approach might yield false positives, we call the resulting plot the *Operational Domain Sketch*, an approximation of the operational domain. Experimental analysis showed that the sketch can be obtained with runtime improvements of up to 365x compared to the state of the art, providing a fast method for determining the robustness of SiDB logic. This, in turn, facilitates the design of robust gates and enables reliable SiDB circuits. To support open research and open data, the implementation and simulation results are publicly available on GitHub as part of the *Munich Nanotech Toolkit* (MNT).

References

[1] C. Debus *et al.*, "Reporting electricity consumption is essential for sustainable AI," *Nature Machine Intelligence*, 2023.

[2] X. Zhihao *et al.*, "Large-scale photonic chiplet Taichi empowers 160-TOPS/W artificial general intelligence," *Science*, 2024.

[3] X. Meng, "Optimization of algorithmic efficiency in AI: Addressing computational complexity and scalability challenges," *Applied and Computational Engineering*, 2024.

[4] J. Drewniok *et al.*, "*QuickSim*: Efficient *and* Accurate Physical Simulation of Silicon Dangling Bond Logic," in *IEEE-NANO*, 2023.

[5] S. Hofmann *et al.*, "Scalable Physical Design for Silicon Dangling Bond Logic: How a 45° Turn Prevents the Reinvention of the Wheel," in *IEEE-NANO*, 2023.

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

[7] S. S. H. Ng *et al.*, "Simulating Charged Defects in Silicon Dangling Bond Logic Systems to Evaluate Logic Robustness," *TNANO*, 2024.

[8] J. Drewniok *et al.*, "The Need for Speed: Efficient Exact Simulation of Silicon Dangling Bond Logic," in *ASP-DAC*, 2024.

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

[10] M. Rashidi *et al.*, "Automated Atomic Scale Fabrication," US Patent, 2022, 17/429,443.

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

[12] ——, "Multiple Silicon Atom Quantum Dot and Devices inclusive thereof," US Patent 10 937 959, 2021.

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

[14] ——, "Binary Atomic Silicon Logic," *Nature Electronics*, 2018.

[15] M. Walter *et al.*, "Reducing the Complexity of Operational Domain Computation in Silicon Dangling Bond Logic," in *NANOARCH*, 2023.

[16] J. Drewniok *et al.*, "Minimal Design of SiDB Gates: An Optimal Basis for Circuits Based on Silicon Dangling Bonds," in *NANOARCH*, 2023.

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

[18] M. Walter *et al.*, "Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, 2022.

[19] J. Pitters *et al.*, "Atomically Precise Manufacturing of Silicon Electronics," *ACS Nano*, 2024.

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

[21] M. Walter *et al.*, "*fiction*: An Open Source Framework for the Design of Field-coupled Nanocomputing Circuits," 2019, arXiv:1905.02477.

[22] ——, "The Munich Nanotech Toolkit (MNT)," in *IEEE-NANO*, 2024.