Master's thesis
Computing Science

Radboud University Nijmegen

Hierarchically Bounded Composable Interactions

Applications and Extensions to Atom-Scale Logic Design

Author:
Willem Lambooy
willem.lambooy@gmail.com

Internal supervisor/assessor:
dr. Wieb Bosma
bosma@math.ru.nl

External supervisor/assessor:
dr. Marcel Walter
marcel.walter@tum.de

May, 2024

Abstract

In this thesis, we consider a system of variables to which local states may be assigned. Interactions are determined for each variable locally by respective evaluations of any given state assignment, such that, importantly, we may compose any two evaluations of partial state assignments to disjoint sets of variables into a definitionally equal evaluation of the joined partial state assignment. We focus on a problem in which we look for the exhaustive selection of state assignments for which all local evaluations satisfy a local requirement that is specific to the assigned state. Thereby, the number of variables in the system determines the exponential complexity in considering the complete function space. With *ClusterComplete*, an exact physical simulator of Silicon Dangling Bonds, it is demonstrated that the *specifics* of a given problem can be exploited to a significant degree, taming the exponential through a real-valued base reduction. One benchmark shows a base reduction of around 1.62 with respect to $\mathcal{O}\left(3^n\right)$ exhaustive iteration.

By operating from the assumption of the system, i.e., all possibilities in the system, we can obtain associated bound information on each pairwise local state-dependent interaction through enumerating all local state assignments to the respective interactors. This initial information structure becomes the axiom from which we construct a proof tree of pruned partial global state assignments: the bound information is used to monotonically produce new bound information; it can only become stricter. Pruning *partial* state assignments thus allows us to consider all of the exponentially many assignments in the function space efficiently, though, with the proposed, this is taken a step further in order to prune *efficaciously*.

A cluster hierarchy may be formed by a distance metric derived from the bound differences for respective pairwise interactions, which provides a heuristic for which interactions are most crucially dependent. Starting with all singleton clusters, the hierarchy determines which clusters should be *merged*, i.e., taken into an exhaustive consideration where we enumerate all remaining possibilities of the subsystem, pruning combinations of partial state assignments where we assume respective bounds for all projected interactions from all other clusters. Since the pairwise interactions within the merged cluster are most crucially dependent, we are able to *selectively* consider the subproblems that offer the highest yield of new information. Simultaneously, the other interactions are relatively less dependent on state information, further motivating the efficacious nature of the construction, as we now *flatten* bound information associated with partial state assignments: bound information of multiple *permutations* is unified to that of a single associated *combination*, allowing further pruning to be *efficient*.

Acknowledgments

Gratitude towards my parents may be quantified by $\aleph_\omega$, although they would not know what that means. In the last months of this process, they supported me to the point of maintaining while I resided with them. A great many thanks go out to Marcel Walter for introducing me to this world and supervising this year-long process. I thank Wieb Bosma for sticking with me and providing confidence for the generalisation, and Jan Drewniok for his amazing ideas (among which those that gave rise to this work) and ever-present physics assistance. Finally, to Lea, Lucas and Maarten: there are no words for that which is unconditional. You form a holy trinity to me.

Preface

How should this thesis be read?

The discussion herein takes place at the intersection of three overarching fields of study: computing science, mathematics, and physics. In particular, the former two merge into theoretical computing science which has been the Master's specialisation offered at Radboud University that provided the theoretical framework that laid the foundations for this work entirely—logical reasoning and thinking in abstractions are products of a higher level view that underlies courses like 'Category Theory & Coalgebra', 'Semantics & Domain Theory' and 'Complexity Theory', and these higher ideas have largely shaped me, academically, to the point where I am able to come up with and present (formalise) my findings in a complete thesis.

That being said, the application domain that dominates the content herein is founded in physics, yet, as a non-physicist myself, my background in physics that has largely been refined for the particular application domain merely supports an understanding of the context that gives rise to the physical *model* that I felt drawn to explore as I have done over various projects. Putting it more directly, the investigations were performed out of a complete disregard for the physical context, as it has no contribution to the problems that may be formed *within* the model, and thus only distracts the problem-solver from solving the problems at hand.

It is precisely this perspective that I want to induce upon the reader, regardless of their background. The prominent exhibition herein is one of *reasoning*, which appears prohibited in a purely physical context due to all the uncertainties that arise, starting with what is and is not possible in (current!) fabrication technology, what has and has not been observed experimentally (so far!), and so on. Fortunately, a physical model offers a framework in which we can deal with absolutes, allowing us to regard problems in physics as problems in computing science. Thereby, I want to stress an idea to those that wish to read this work, avoiding an easily misconceived interpretation of this work's focus. It is presented in bold on the next page.

Finally, to those solely interested in the *generic* methods and ideas presented in this thesis, Chapter 3 may be read without reading Chapter 2, as the former does not depend on any particulars of the application domain.

This is a thesis in the field of computing science, not physics, yet targeting a broader audience. The physical context is introduced in order to motivate the interest in the associated physical model. Some particulars of this physical context may be mentioned without further elaboration to present more detailed information to those interested. These facts are not essential to grasping the principal ideas presented in this thesis, nor should they hinder further reading.

How this thesis came to be

A trend of exhibiting the power of abstraction within a physical context started with my Bachelor's thesis. [16] There, I worked on quantum-dot cellular automata (QCA), which relates strongly to the technology that is of interest in Chapter 2. Particularly, I found that QCA, which is not particularly synonymous with fabrication, could in theory be *stacked*, disregarding fabrication challenges. [17] Moreover, I felt compelled to QCA simulation—trying a *discrete* approach—which, paired with the fact that dr. Marcel Walter ended up finding my Bachelor's thesis, eventually determined that I would work on the SiDB simulation problem that is the topic of Section 2.5.

Without further introducing the SiDB topic as that will be done in Chapter 1 and Section 2.1, I initially started to experiment with a radical approach to the *heuristic* simulation problem in September 2022 throughout an internship at the Technische Universität München, with which dr. Walter is affiliated, trying one-dimensional simulation of symmetric input using wave plots. My intuition for the SiDB dynamics started to develop in this time through many experiments, becoming countless when the Master's thesis project started out in March 2023.

A gut feeling for the potential of clustered simulation started around this time with a successful project that, roughly put, extended a heuristic SiDB simulator by permuting identified *groups* in its results in order to produce more results.[1] After a tool for finding kinks was developed that proved that it is too early for large-scale SiDB logic design assessing with current SiDB logic design, a heuristic simulator was created after the summer in partial fulfilment of dr. Wieb Bosma's course that used a recursive *quadtree* structure. Around this time, blurry visions emerged of a hierarchical information structure that would revolve around a sequential processing of projected interactions from each node onto all sibling nodes—already quite akin to what I ended up with for this thesis, though the information did not pertain to bounds yet, as the switch to the more difficult *exact* simulation was to be made in December. The algorithm design was very challenging, later generalised in early April.

[1] *CompoSim* can be built for SiQAD: https://github.com/wlambooy/fiction/tree/CompoSim.

Contents

1 Introduction **7**

2 SiDB and Logic **9**

2.1 SiDB Physics . . . 9
2.1.1 Fabrication . . . 10
2.1.2 Charge States . . . 11
2.1.3 Band Bending . . . 12
2.2 The Electrostatic Game of Points in a Plane . . . 15
2.2.1 The Rules of the Game . . . 16
2.2.2 Logic Design Challenge: a Two-State BDL Wire . . . 17
2.2.3 The Ground State Model . . . 19
2.3 Ternary Atomic Formulations . . . 20
2.3.1 Charge State Configurations in a Hierarchy . . . 20
2.3.2 Properties of Atomic Identity Logic . . . 23
2.3.3 First Order SiDB Predicates . . . 26
2.3.4 Atomic Forced Combinational Logic . . . 29
2.4 Ground State Logic . . . 34
2.4.1 Ordered Global States . . . 36
2.4.2 Preordered Judgements . . . 40
2.4.3 Towards Generalised State Space Ordering . . . 44
2.5 SiDB Simulation . . . 45
2.5.1 Exact Physical Simulation . . . 46
2.5.2 *ClusterComplete* . . . 49
2.5.3 Simulation with Uncertainty Maps . . . 56
2.5.4 Bound-Optimised Logic Design . . . 59
2.6 Notational Conventions and Lattice Theory . . . 60

3 State Space Pruning in a Cluster Hierarchy **63**

3.1 Clusterings and Morphisms . . . 65
3.1.1 A Category of Clusterings . . . 65
3.1.2 Operations on Clusterings . . . 66
3.2 Minimised State Space Construction . . . 67
3.3 Active State Space Destruction . . . 75

4 Applications and Future Improvements **77**
4.1 The Exact Physical SiDB Simulation Problem 77
4.2 Instantiation to Boolean Satisfiability 78
4.3 Corecursive Hierarchy Construction 80
4.4 State Relating and Global Evaluation 81
4.5 Summary and Concluding Ideas 82

A Properties of the Functor $\mathcal{D}$ **86**
A.1 $\mathcal{D}$ is a Functor 86
A.2 Proof of the $\mathcal{D}$-Iteration Invariant 88
A.2.1 Loop Invariant Properties 88
A.2.2 Loop Invariant Theorem 91

B *ClusterComplete* Simulations of 83-DB Crossover Layouts **93**

C Validity Witness Partitioning **98**

Chapter 1

Introduction

As proven by the first Time Hierarchy Theorem in 1965, exponential time problems belong amongst the most difficult problems in the world. [12] But what does it mean for a problem to be difficult? Intuitively, for the case of an exponential time problem it means that we require an amount of time that is exponential in the size of the input in order to terminate with an answer. Yet there is another facet of a problem that belongs to a class of superpolynomial complexity, which precisely connects to the degree of entanglement that may be associated with a measure of complexity. When a problem, formulated by a set of assumptions and a goal for a solution to attain, involves a high degree of interactivity between the input variables in relation to the given goal, then a proposed solution must capture the totality of the problem's relations. In particular, this implies that a solution to a problem with exponentially many relations may be applied to *any problem up to* that level of intraconnectivity that includes the assumptions of the former problem.

This sets the tone for this thesis, in which we tackle a problem which only assumes staticity and fragmentable local interaction evaluation. The goal for our problem will be to obtain the exhaustive selection of elements for which a polynomial predicate holds in a set that grows exponentially with the size of the input. In this work, we look at the direct application of a proposed contribution to the study of exact physical simulation of *Silicon Dangling Bonds*, to be discussed henceforth. In line with the paragraph above, the claim is that our method is applicable beyond the scope of this particular domain. To support this, the method referred to, presented in Chapter 3, is generalised away from the aforementioned application domain that inquired the creation originally. This allows for the generic patterns to be observable by researchers who may match a possibly highly intraconnected problem in their own field to our problem statement. In addition, the generalisation may serve as a basis to those who would take the challenge of lifting the proposed method to be compatible with weakened assumptions.

SiDB: Promising an Ultra-Green Computing Future

When one introduces the topic of a technology that contends with modern-day processing realisations, it is inevitable to bring up statements about the current state of affairs with regard to efficiency of development and the accompanying outlook on the future. In particular, the advancements in the field have long followed the prediction of growth by Gordon Moore, recently deceased at the time of writing, who formulated this prediction around the time Complementary Metal-Oxide Semiconductor (CMOS) technology was invented. [21] Yet, nearly half a *century* later, this exact computing paradigm is still the ubiquitous technology that has all of industry's focus, throughout a remarkable battle against the quantum-tunnelling effect that poses an increasing problem as the quest for 'cramming more components onto integrated circuits' continues. [2] In special, the Landauer limit places a hard asymptote on the current CMOS endeavours, which is apparent considering how Moore's Law has failed to hold up in the last decade. [38]

Forth come the class of *Field-Coupled Nanotechnologies* (FCN), to which Silicon Dangling Bonds (SiDBs) belong, the prominent topic in this thesis. This class of emerging nanotechnologies promise computation *below* the Landauer limit, precisely by utilising the quantum tunnelling effect as the means of information propagation. [19] Nemesis to CMOS, dearest friend to FCN. However, SiDB technology further distinguishes itself as the prominent candidate to come out on top over its FCN siblings that are mainly stuck in the world of theory: advancements in fabrication are led by the investments of Robert Wolkow's Quantum Silicon Inc., whose group pioneered in observing the application in logic processing with a proposed sub-30 nm^2 OR gate. [14]

Some interesting remarks about this technology follow. Tunnelling rates of $\sim 100\,\mathrm{THz}$ have been observed in SiDB systems, indicating that Terahertz clock speeds are well within reach; a great leap in computation throughput. [20] Perhaps a more important fact relates to the critical quest for green computing solutions. Consider the intrinsic energy difference between the **0** and **1** signal in CMOS, represented by a low voltage and a high voltage respectively. This means that an idle state consumes power consistently in order to be maintained. In comparison, SiDB only requires power for dynamic operation in the form of clocking pulses, where, in addition, *adiabatic* clocking is a solution to heat dissipation with information erasure. [18, 19, 27] Together, the promise is a new *order* of power efficiency.

Yet the problem of physically simulating these SiDBs is prerequisite to the development in the field due to costly fabrication. Especially *exact* physical simulation rapidly becomes computationally intractable for just a few dozen SiDBs. [11, 24] Recently, we have seen the exceptional use cases of exact simulators in, e.g., the creation of minimal logic gates. [10] Yet the state-of-the-art does not, for instance, allow testing gate robustness under the presence of nearby gates. In this thesis we intend to fill this gap.

Chapter 2

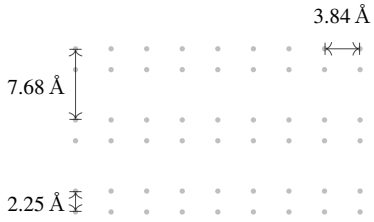
SiDB and Logic

Previously, we mentioned the SiDB technology along with some major takeaways for their application. In particular, we mentioned the capability of this technology to synthesise logic circuits at the *atom*-scale, crossing the miniaturisation asymptote that contemporary CMOS technology skims towards. In this chapter, we start off with a review on properties of the technology that are fundamental to this expansive chapter.

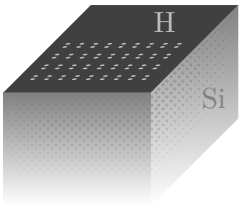
With Section 2.1, we zoom in on the technology and related preliminaries in physics, after which we zoom out in Section 2.2 to understand the problem of exact simulation in its minimal form. This forms an intuition that builds towards the proposed solution presented in Section 2.5. Extending beyond ideas that put forward state-of-the-art solutions, the reduction of the problem to the bare essentials allows us to focus on *constructive reasoning* in the SiDB domain, exemplified in Section 2.3. In addition, Section 2.4 features ideas and demonstrations that aim to inspire optimisations in the process of logic design on the platform. Concepts associated with *global state relevancy*, introduced there, make an interesting candidate for a future fusion—as discussed in theory in Chapter 4—with the primary contribution in this thesis. The focal idea of this work is demonstrated in the SiDB application domain in Section 2.5.2, and presented in the generalised form in Chapter 3. Facilitating this presentation, we will build a machinery with our notation, for which some preliminary mathematical conventions and structures are reviewed in Section 2.6, leaving the SiDB topic behind from then on.

2.1 SiDB Physics

Prerequisite is knowledge of relevant properties of the medium from which problems of interest for this thesis emerge. However, this being a thesis in *computing science*, we touch on the physical properties with limited elaboration on the physical environment, as to not distract from our focus on the associated physical model that is of focal interest in this chapter.



(a) The H−Si(100)-2×1 surface lattice, showing where DBs may be placed.



(b) A sliced silicon bulk of which the H-passivated surface may host SiDBs.

Figure 2.1: A minimal schematic of the physical environment of SiDBs.

2.1.1 Fabrication

When considering SiDB technology, we think of a planar technology. The plane that gives us a world of possibilities in both logic realisation and storage is the hydrogen-passivated surface of a silicon bulk. [1, 13] Think of this as follows: a big chunk of silicon atoms that are connected in a crystal structure is given a clean cut, creating the surface plane illustrated in Figure 2.1. This, however, leaves the silicon orbitals at the surface open to host up to two electrons. Each of these free sp^3 orbitals is satiated through *H-passivation*, leaving a surface of hydrogen atoms that neutralise them. [28, 29]

Depending on the orientation of the Si crystal with respect to the cut direction, different lattice structures are observed at the surface, specific to the associated orientation. [37] These effectively shape the 'grid' as identified by the regular pattern of H atoms visible at the surface, thereby forming the playing field when we want to create gates on these surfaces. Commonly, the (100)-2 × 1 orientation is considered, characterised by a grid of dimer rows, though recently the (111)-1 × 1 orientation, characterised by an all-even spacing, has been gaining interest as well.[1] [25]

We used the word 'visible' in the previous paragraph, but this is of course very far from visibility by the naked eye. Fortunately, advances in microscope technology have offered us an artificial eye, with for instance Scanning Tunnelling Microscopy (STM). [3] These treasures of recent technological advancements are able to move an atomically sharp tip with atomic precision. Thus, for the case of STM, this 'eye' may not only sense, but interact as well, which leads to understanding why Silicon Dangling Bonds are named the way they are. Specifically, with *hydrogen lithography*, the STM tip may send tiny voltage pulses when placed directly above one of the H atoms on the silicon bulk surface. [1] The result of this, after perhaps a few such pulses, is a removal of the H atom, as it detaches from its silicon friend. This forms a

[1]While studies on logic applications using the 111 lattice are not seen in the literature at the time of writing, the interest is confirmed with support for this lattice in an open-source platform used in SiDB research efforts: https://github.com/cda-tum/fiction/pull/380.

dangling bond in the void of the late H atom, which we abbreviate to DB.

Together with *hydrogen repassivation*, the feat of automated creation and erasure of DBs already has phenomenal implications. For instance, the aforementioned STM lithographic technique may be used to print binary information onto the surface, where a DB would be a **1**-bit and no DB a **0**-bit. [25] Wolkow's group stated a few years back that the then 45 million songs on iTunes could in theory be stored on the surface of a 0.25$ coin, giving an idea for how small *atom*-scale really is.[2] Notice that, by the STM-enabled reversibility of the DB creation in a similar process as described previously, where now the tip carries an H atom to dispose, an application in *rewritable* memory is offered. The SiDB technology could even find itself amongst consumer products in the future, as the erasable DB nanostructures remain intact around room temperature. Indeed, there is quite some wiggle room here: temperatures may go up to 200 °C without affecting the structural properties of a DB.[2]

2.1.2 Charge States

While the static DBs may already find applications in storage or memory, in this work, we are more interested in logic, which relates to SiDB technology being a *field-coupled* nanotechnology. Recall that this means that the logic operation is inherent to the phenomenon of quantum effects playing at this scale, which emerges from electrostatic interactions in which DBs play a crucial role. In particular, the DB plays the role of a *quantum-dot* here: a site that may be occupied by some amount of electrons. [36] Without any, the silicon orbital is short of an electron, resulting in a positive charge. With one electron, the charge state is said to be neutral, and the DB is energetically inactive. The final possibility is that there are two electrons at the site of the DB; this is the maximal amount of electrons the silicon atom can have in the sp^3 orbital that was opened by formation of the DB. Such energetically active states project accordingly-charged Coulombic potential in all directions. [26] We unfold the topic of charge states and their interactions in DB ensembles in the next section, after furthering the intuition of the dynamics at play.

The silicon bulk is typically considered to be n-doped, implying the existence of an electron surplus. [25] Consequently, a preference of electronegativity is induced onto the surface, and, as a result, DBs in such an environment will tend towards a negative charge state. Positive charges do exist, but are rare.[3] [24] The interesting property, however, is observed when two DBs charged in the same polarisation are placed in close enough proximity for the charges to clash energetically, i.e., *interact*. In such a

[2] 'Music stored in smallest stable rewritable atomic memory', uploaded by the first author of [1] (https://youtu.be/CIUxkceinXk).

[3] Positive charges play quite an important role in this work compared to other work on SiDB simulation, as we will be able to consider these *efficiently* in Section 2.5.2.

contained system of SiDBs, the measure of such a clash is *energy*. By the second law of thermodynamics, it follows directly that the universe tends any such contained system to a state of *minimum energy* at all times. [5] This naturally drives the system to stabilise in an energetically lower state when the opportunity is there; when the DBs are tightly spaced, they may share one electron that tunnels back and forth between them, avoiding the earlier potential clash in a non-equilibrium, called a *degeneracy*. [26]

2.1.3 Band Bending

As hinted at previously, the SiDB charge states, and in particular the interaction between charge states in nearby DBs, are the key to logic operation on this platform. Moving towards an understanding of the SiDB simulation problem, we will need to zoom in a bit further on the precise interaction that plays at this scale; this interaction is captured by the elaborate topic of *band bending*, of which we will now review concepts most relevant to our domain.

Conduction Band and Valence Band

Preliminarily, we must consider what the bands in question are that are being bent. By the *energy band theory of solids*, these are energetic ranges, the conduction band (CB) and the valence band (VB), dictate possible energetic states of electrons in the bulk, and in the free sp^3 orbitals of the SiDBs in our case. These ranges may vary from place to place in the solid when electrostatic forces cause a deviation from the energetic level of the bulk as a complete electron ensemble; CB and VB are bent locally in synchrony.

To give a rough intuition, the CB and the VB may be seen as a source and a sink of electrons respectively. Different types of materials, e.g., metals, semimetals, semiconductors and insulators, distinguish themselves in their electrical conductivity precisely as a consequence of the relation between the two bands for the respective material. For instance, metals are well known to be good electrical conductors, which is enabled by their intrinsic property of having the CB and VB overlap, such that electrons may travel freely from the former to the latter. Contrarily, insulators are materials that do not conduct well, which is explained by the relation between the bands for this type of material: the CB and VB lie far apart from one another, implying a high energy threshold for an electron to jump between the bands. This energetic vacuum between the CB and the VB is also present for semiconductors, to which silicon belongs, to a lesser extent. This is known as the *band gap*, which thus determines a range of excluded energetic states of bulk electrons.

Charge Transition Levels and Trap States

This band gap, and the electron capacity at an SiDB, together define the three SiDB charge states. A free sp^3 orbital yields energetic states that, in

contrast with the quasi-continuum of bulk electron energies, exist in *isolation* within the band gap, which thus enforces energy thresholds to charge state transitions. [14] Specifically, these are the $(\mathbf{+/0})$ and the $(\mathbf{0/-})$ charge transition levels, separated by a constant potential denoted by $\mathcal{E}$ and valued at 0.59 eV for our consideration. [22] Essentially, the existence of a dangling bond creates two *trap states*, referring to the electron that may occupy such a state which is then energetically trapped. This corresponds to the SiDB's quality of a quantum dot, since either none, one, or both of the trap states may be occupied, which we associate with SiDB**+**, SiDB**0**, and SiDB**−** respectively. The actual occupation of these trap states, and therefore the charge state of an SiDB, is decided by the relation of the aforementioned charge transition levels with respect to the *Fermi energy level*, denoted by E_F. This energy level constant is determined for the type of bulk, and refers to the aforementioned energy level of its complete electron ensemble. It may be seen as an energetic 'sea level': the occupation of either trap state is determined by the position of the associated charge transition level in relation to E_F; those 'submerged' are occupied.

We previously mentioned that the silicon bulk is commonly considered to be n-doped, which implied a surplus of electronegativity, i.e. free electrons. The doping polarity and magnitude is decided by the doping concentration and the type of doping in the bulk, which then decides the position of the energy levels associated with the CB and the VB in relation to E_F. In our domain, E_F is actually put very close to the CB, of which the associated energy level is always strictly above that of the VB for the semiconductor matter in consideration. This means that the charge transition levels $(\mathbf{+/0})$ and $(\mathbf{0/-})$, that are found in the band gap, are both below E_F in the most simple case of a lonesome DB. This now clarifies the earlier statement that SiDBs tend towards a negative charge state in our environment.

The precise difference in energy levels between E_F and $(\mathbf{0/-})$ is initially determined by μ_-, which depends on properties of the bulk. This will be a quintessential parameter to our simulation problem later. Similarly, μ_+ describes the initial difference between E_F and $(\mathbf{+/0})$, which can be inferred from the former by the constant potential $\mathcal{E}$ between the two charge transition levels. In terms of absolute values, these are approximately 0.3 eV and 0.9 eV below E_F respectively. [11] Relative to E_F, therefore, μ_- and μ_+ are strictly negative values, with the latter more so by an amount of $\mathcal{E}$. Note that we used the phrase 'initial' previously, since the aforementioned is valid for an undisturbed DB. Next, we consider the disturbed case, i.e. under the influence of other DBs and external potential sources.

Band Bending and Potential

We now arrive at the point where we may elaborate in detail on the interaction between SiDBs in close proximity of one another. Recall that we stated

earlier that both charge transition levels are below E_F when considering an SiDB that is not influenced externally. Now, when such a DB *is* influenced externally, we say that band bending occurs with a magnitude in proportion to the received potential that is local to that DB. This is called the *local electrostatic potential* at that DB. A positive local potential then 'bends' both charge transition levels upwards, towards the conduction band. As a consequence, one, or both of the charge transition levels may cross E_F, resulting in SiDB**0** or SiDB**+** respectively. Electrostatic potential, here, should be regarded as a measure of electro*negativity*. This is reflected by the negation in the formula for the Coulombic potential between two SiDBs i and j separated by Euclidean distance of $d_{i,j}$:

$$V_{i,j} = -\frac{q_e}{4\pi\epsilon_0\epsilon_r} \cdot \frac{1}{d_{i,j}} \cdot e^{-\frac{d_{i,j}}{\lambda_{TF}}}, \tag{2.1}$$

where q_e and ϵ_0 are constants: the former denotes the *electron charge*, while the latter is the *vacuum permittivity*. Furthermore, ϵ_r denotes the *relative permittivity* and λ_{TF} is the *Thomas-Fermi screening length*; these two, like μ_-, depend on properties of the material and are thus included in the set of physical parameters for our simulation efforts later. The main takeaway in this formulation is the following: electrostatic potential declines *exponentially* over distance, implying powerful interaction at close proximity, which rapidly weakens into negligibility with increased spacing. Moreover, we define $V_{i,i} \coloneqq 0$, simplifying much to come.

The local potential for some SiDB i may be computed by summing the potential projected onto it by all other SiDBs, along with a summed local potential of external sources of disturbance:

$$V_i = -V_i^{\mathsf{ext}} + \sum_j V_{i,j} \cdot \alpha(j), \tag{2.2}$$

where $\alpha(j) \in \{-1, 0, +1\}$ corresponds to the associated charge state of SiDB j, which may be assumed as a given for now in this passive context, while later, in the active context of physical simulation, we will consider these to be *assignments*. This casts a light on the exponential problem that is now slowly uncovered, as the local potential at some DB is dependent on the charge states of *all* other DBs. Indeed, the physical world is *fully connected*, making it non-trivial to combine physical simulations of interacting subsystems of disjoint collections of SiDBs into a simulation of the combined system.

Completing the system's dynamics, recall that the band bending that occurs with non-zero local potential has a direct consequence for the charge state of the recipient SiDB. Composing this with the energy level differences between the charge transition states and E_F in the undisturbed case, we obtain a set of conditions that is critical to the problem of physical SiDB simulation, and a major focus for Chapter 3, albeit considered in a generalised

form there. This is the *population stability criterion*, which holds universally in the physical world, and therefore contributes to the judgement of *physical validity*, also known as *metastability*. It holds *if and only if*, for all SiDBs i:

$$\alpha(i) = -1 \iff V_i < -\mu_- \tag{2.3}$$
$$\alpha(i) = +1 \iff V_i > -\mu_+ \tag{2.4}$$
$$\alpha(i) = 0 \iff V_i > -\mu_- \wedge V_i < -\mu_+.^{4} \tag{2.5}$$

This concludes a comprehensive capture of SiDB dynamics relevant herein, and a light introduction to the associated physical simulation problem, in which we look for metastable charge state assignments to SiDBs in a system.

2.2 The Electrostatic Game of Points in a Plane

From the SiDB's interactability that is enabled by the local band bending effect emerges a property that is critical to the dynamic operation of SiDB technology, spawning a logic realisation medium at the atom-scale. By the great range of freedom that the atomically precise placement and erasure of DBs offers, the medium puts forward a puzzle for researchers to delve into: *what DB ensembles compute logic*? More so, this becomes an optimisation problem, as we will see in Section 2.4, where we go in depth into the associated dimension of thermodynamics. Fortunately, the existing physical simulators allow us to play around with the modelled dynamics that have been described for the most part in the previous section, such that expensive fabrication is largely avoided in experimentation. [8, 11, 24] Such simulators, for a system with n SiDBs, explore the function space containing 3^n *global* states to the system that the model offers, thus growing *exponentially* in n.[5] Whereas heuristic methods exist, and may produce certificates that are useful in SiDB logic design, in Section 2.5 we will mainly look into considering this search space *exhaustively*—the primary problem of interest in this thesis.

With the problem-solving intention set for attacking the complete space of charge state assignments, it will be beneficial for us to steer away from the low-level dynamics at play, such that we may take a step in abstraction towards a view of a system of SiDBs as a 'points in a plane' game for which we may determine a objective. In particular, besides the aforementioned game of (optimal) logic realisation, the model yields a game of *pruning* by the population stability criterion defined above: given a system, we may infer by (2.1) through (2.5) whether certain charge states may never be

[4] We use strict comparisons here, intentionally keeping the on-threshold cases undefined. Considering that local band bending is valued in the uncountable domain of real numbers that is appropriate to the uncountable infinity at each point in our physical universe, equating such samples of the infinite is not particularly senseful.

[5] Any such global state may be regarded as a configuration, or distribution of charge states when examined passively.

assigned to some SiDBs, or only in combination with specific charge states assigned to nearby ones, i.e., whether certain *partial* global states may be pruned. We introduce the primary associated intuitions as 'rules' for our game in the following. Advancements in this line of thinking will enable an impactful contribution to the exhaustive physical SiDB simulation problem, as demonstrated in Section 2.5.2.

2.2.1 The Rules of the Game

The rules of the game can be captured simply as follows, where our consideration is appropriate to the physical setting in the absence of potential sources other than the SiDBs themselves:

Rule 1: Any energetically isolated SiDB rests in SiDB**−**.

Rule 2: Any SiDB placed in close proximity of another SiDB *may be* SiDB**+**.

To connect back to the physical context, **Rule 2** holds for a pair of SiDBs i and j if (not only if) the Euclidean distance $d_{i,j}$ that separates them is small enough for $V_{i,j} < \mu_+$ to hold. Since, by (2.2), local potential is accumulated, this rule may be extended to the following:

Rule 2a: Any SiDB i may be SiDB**+** if *and only if* its maximal local band bending in the negative direction is such that $\sum_j V_{i,j} < \mu_+$.

This assertion was first presented in [11] with the advent of an efficient exact physical SiDB simulator, i.e., one that considers the set of global states exhaustively without naming each element. Hence, these formalisations pave an intuition towards exact SiDB simulation.

There are two things to note here. First, since μ_- is a weaker threshold with regard to band bending in the negative direction, **Rule 2** and **Rule 2a** hold for SiDB**0** with the μ_- threshold in the respective way. Secondly, note that **Rule 2a** is easily extensible to incorporate V_i^{ext} as well, yet, since current efforts in the field are focused on analysis under static such external potential sources, they do not elevate the complexity of the problem, and we may leave them out for now. They will play a more involved role when the dynamics in consideration are extended with the concept of *clocking*, although still, at any instant, these forces may be regarded as static due to the one-sided interaction with such external electrostatic landscape perturbers.

The last rule of this series to be presented connects directly to a concept that arose with the original research efforts into logic realisation on the SiDB platform: *binary dot logic*. Abbreviated to BDL, this term was coined by Huff *et al.*, who demonstrated the application of the SiDB technology to logic for the first time in [14] with their $5\,\mathrm{nm} \times 6\,\mathrm{nm}$ OR gate, in which pairs of SiDBs—i.e., *BDL pairs*—together represent a binary state that may be observed. The following rule captures the dynamic that enables binary logic on this atomic platform:

Rule 3: Any pair of SiDBs i and j for which $d_{i,j}$ is small enough such that $V_{i,j} < \mu_-$ *cannot* have a charge configuration of *only* SiDB−.

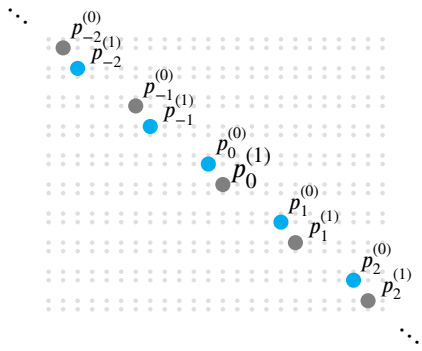
Tuning a distance between SiDBs in a pair to the thresholds determined by **Rule 2** and **Rule 3**, and moreover, for each SiDB in the pair, letting the spacing to other SiDBs in the environment be large enough such that that **Rule 2a** eliminates the possibility of SiDB+, we may consider a series of regularly placed SiDB pairs to be a *two-state* wire. Next, in Section 2.2.2, it is shown in a simple experimental demonstration that a BDL wire pattern exists in which no BDL pair may assume the (SiDB**0**, SiDB**0**) configuration under a standard set of physical parameters in ideal conditions. [34]

2.2.2 Logic Design Challenge: a Two-State BDL Wire

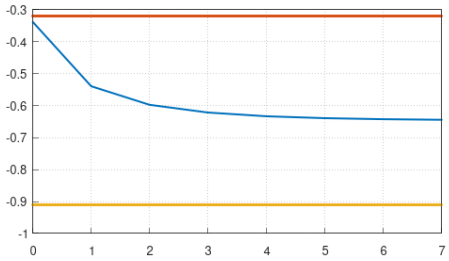
As a simple 'proof of existence', in here, we show an example of a BDL wire pattern for which we can analytically motivate the strictly two-state character in adherence with **Rule 2a** and **Rule 3**, for the case of such a BDL wire in isolation. Specifically, we consider an *infinite wire model* within the physical model of a system of SiDBs, in order to simplistically model the behaviour of such a wire as a contained system of arbitrary size. This is an instantiation of a model that yields an effectively limited logic design search space, since, depending on assumed physical properties of the medium, logic realising SiDB placements are constrained to effective interaction ranges determined by the *screened* Coulombic potential. In Section 2.3.2, we will consider the case of modelled identity logic—i.e., wires—more carefully, after which we look at a slightly more involved case for two-input combinational logic in Section 2.3.4. Preceding those sections in which *synthetic* conditions are given, however, we now motivate the logic design model approach with an *analytic* demonstration.

For all $k \in \mathbb{Z}$, we have that $V_{p_k^{(0)},p_k^{(1)}} \approx -0.34\,\text{eV} < \mu_-$ for the physical parameters specified in Figure 2.2, hence **Rule 3** says we cannot have only SiDB− in any of these pairs. To calculate the influence of neighbouring wire pairs, we consider the BDL wire to stretch out in both directions. Then, since **Rule 3** holds for all pairs, the potential that SiDB $p_0^{(1)}$ can receive from other SiDBs in the BDL wire is $\sum_{k=0}^{n} \left(V_{p_k^{(0)},p_0^{(1)}} + V_{p_{-k}^{(1)},p_0^{(1)}} \right)$, which is the curve shown in Figure 2.2b for an increasing number n of pairs connecting to either side. Note that, in order to propose a limit under various circumstances in our BDL wire environment, we assume a kink in the wire, as in Figure 2.2a.

It can be seen that the rapid exponential decline leaves a good amount of headroom for additional potential sources before SiDB+ can be attained by SiDB $p_0^{(0)}$, this would happen when the threshold of $\mu_+ = \mu_- - \mathcal{E} = -0.91\,\text{eV}$ is crossed. Now, in order to assert no other than the two-state configuration (SiDB−, SiDB**0**) is possible for any such BDL pair in the given environment, it suffices to show the state (SiDB**0**, SiDB**0**) will not occur. W.l.o.g., it can

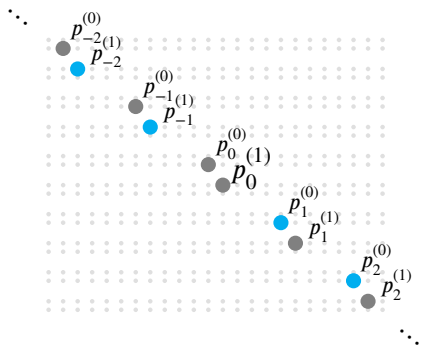


(a) A kinked BDL wire. Each pair contains SiDB− (blue) and SiDB**0**.

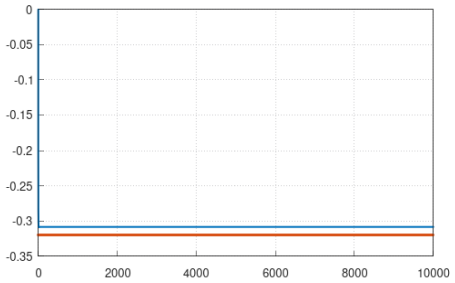


(b) The accumulated chargeless potential at SiDB $p_0^{(1)}$ for an increasing amount of BDL pairs added on either side. The charge transition thresholds are shown.

Figure 2.2: A BDL wire pattern that respects the two-state criterion given by **Rule 2a** and **Rule 3**, and an additional property that rules out (SiDB**0**, SiDB**0**) (see Figure 2.3). The physical parameters used follow a set standard: $\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$. [34]



(a) An unstable state of a BDL wire.



(b) The accumulated chargeless potential at SiDB $p_0^{(1)}$ for an increasing amount of BDL pairs added on either side. The **(0/−)** charge transition level is shown.

Figure 2.3: Figure 2.3b shows a clearly asymptotic behaviour, implying that no amount of BDL pairs connected in this fashion can have a pair with a (SiDB**0**, SiDB**0**) configuration—in ideal circumstances that is. Therefore, the configuration in Figure 2.3a is not metastable. The physical parameters are as in Figure 2.2.

only occur if $\sum_{k=0}^{n} \left(V_{p_k^{(0)}, p_0^{(1)}} + V_{p_{-k}^{(1)}, p_0^{(1)}} \right) < \mu_-$, which Figure 2.3 suggests is not the case for our assumed set of parameters.

Although this shows that each pair independently conforms to two-state behaviour, we have not considered the two-state behaviour of the *wire*. In Section 2.3.3, this issue of kinks is described in detail.

2.2.3 The Ground State Model

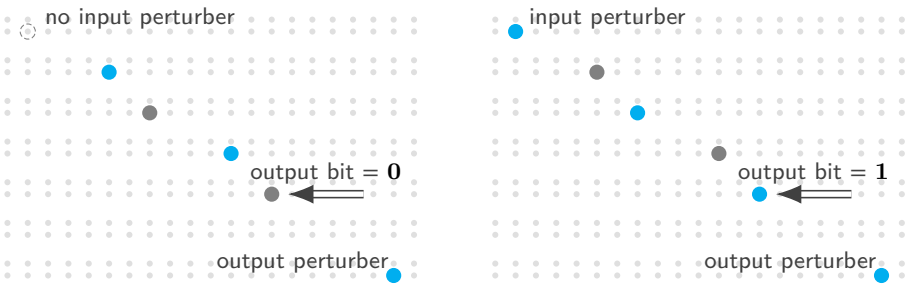
In Section 2.2.1, we discussed universal rules about the system that restricts the variety of charge state configurations that can manifest. Yet, when reasoning from the *ground state model*, in any instance of an observation, only the metastable configuration with the lowest system energy is expected to be observed. [24] This global energy level, also measured in eV, is computed as follows:

$$E = -\sum_i \frac{V_i \cdot \alpha(i)}{2}, \tag{2.6}$$

where the division by 2 ensures the symmetry of the Coulombic potential relation from (2.1) is respected, counting each just once, as (2.6) precisely quantifies the 'potential clash' that was mentioned in Section 2.1.2. Also mentioned there was a case of a non-equilibrium minimal energy configuration; this is an instance of a non-singleton set of charge state configurations with 'equal' system energy.[6] It may be possible that multiple charge state configurations result in equal system energy, as is the case for a non-equilibrium minimal energy configuration, as touched on at the end of Section 2.1.2. Any such equivalence class of charge configurations is a set of *degenerate* states. Any element of the equivalence class of minimal system energy is called the *ground state*, and if this title is shared, they are thus called degenerate ground states.

For our two-state wire, the ground state model offers us a framework for the propagation of binary information, encoded by the configuration of each two-state pair independently. When formed in sequence, these pairs find themselves aligned to reduce the energy between, thus representing the same information, while traversing space; this is exactly why we consider a structure as shown in Figure 2.4 to be a BDL wire. Thereby, the ground state model presents a thermodynamic dimension to the electrostatic game of points in plane, which, as we will discuss in Section 4.4, generalises to a *relative global state evaluation*, as opposed to universally required local state acceptance by respective local evaluations, i.e., population stability. Later, in Section 2.4, we go further into the implications of the ground state model on SiDB logic and its design with associated quality assessment methods, where global state relating will be a major topic. Preceding, however, we expand our discussion on logic design models, simultaneously introducing concepts that build towards Section 2.5.2 and ultimately Chapter 3.

[6] Referring back to [4], equality only makes sense within an inherently finite model. However, 'approximately equal' may be sufficient here, since electron hopping has been observed for a degeneracy where the assumed influence of, e.g., [STM] *tip-induced band bending* would always cause *some* energetic difference in global states. [24]



(a) The ground state of a BDL wire structure carrying the **0** signal.

(b) The ground state of a BDL wire structure carrying the **1** signal.

Figure 2.4: The only metastable states, and therefore ground states, of a small BDL wire segment with input **0** and **1** respectively. The output perturber ensures a 'default operation', while, in this case, input is applied to the wire by either the absence (**0**), or presence (**1**) of an input perturber. I/O perturbers are commonly placed with respect to the DB placement regularity of the wire pattern, as to simulate the presence of connecting wire segments.

2.3 Ternary Atomic Formulations

Having understood the physical intuition of the information propagation dynamic, we take the opportunity to describe modelled SiDB systems *synthetically*, i.e., in a non-numerical manner. This facilitates a discourse on optimisation heuristics relevant to the SiDB medium manufacturing process, which is discussed at the end of Section 2.3.3. In short, it is proposed that certain medium properties may host SiDB logic design with *physically forced* logic behaviour, as determined by analysis using a logic design model. Thereby, *optimal* medium properties may be found, which could be sought to be obtained with advancements in fabrication.

Following sections thus exemplify logic design models, furthering the idea that supported the analysis in Section 2.2.2. First, however, in Section 2.3.1 we introduce a view of SiDB systems that will enable a framework of reasoning that is a principal take-away in this chapter on SiDB and logic. To exemplify *constructivity* in this context, we may build on analytic results from Section 2.2.2, which assumed $\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$.

2.3.1 Charge State Configurations in a Hierarchy

In Section 2.2.2, we used the notation (SiDB**0**, SiDB**0**) to describe some charge configuration to an arbitrary pair of SiDBs. We now refine this notation by introducing *multisets*; this gives $\{\!\{\text{SiDB}\mathbf{0}, \text{SiDB}\mathbf{0}\}\!\}$, or more conveniently, $\{\!\{\mathbf{0}, \mathbf{0}\}\!\}$. Now, when we consider $\{\!\{\mathbf{0}, -\}\!\}$, this *entails* (SiDB**−**, SiDB**0**) *and* (SiDB**0**, SiDB**−**), i.e. in a *constructive* sense—a paramount concept towards Section 2.5.2 and later Chapter 3. To foreshadow, when considering

metastability as the criterion for the existence of charge configurations in consideration, multiset configurations that overrepresent the space of metastable subconfigurations may be discarded in construction of a minimised *hierarchical* space.[7]

For charge configurations in Figure 2.4, we can make the following observations. Building further on the case study from Section 2.2.2—which assumes an identical wire pattern—with respect to numerical properties of the silicon medium, the system $\boldsymbol{S}$ shown in Figure 2.4 contains two pairs that can each independently only attain $\{\!\{\mathbf{0}, -\}\!\}$. Furthermore, comparing the position of the output perturber to that of $p_0^{(1)}$ in Figure 2.2a, it is easy to see that the output perturber will always be SiDB−: Figure 2.2b shows that two connecting BDL pairs on either side maximally yield a chargeless potential of $\approx -0.6\,\mathrm{eV}$ local to $p_0^{(1)}$, whereas this is *at least* halved for that of the output perturber by the unidirectional wire influence local to it. Thus, the chargeless potential local to the output perturber end up above the **(0/−)** threshold visualised in Figure 2.2b. With a similar argument, we obtain a space of metastable configurations of both I/O SiDBs individually: $\{\{\!\{-\}\!\}\}$, i.e., the singleton configuration $\{\!\{-\}\!\}$ is the only one that either I/O SiDB may attain, thereby forming a singleton space of subconfigurations.

Hierarchical Charge Spaces

With the collected information on metastable subconfigurations of the wire segment in Figure 2.4, we may move towards a *hierarchical* view of the exhaustive stable state selection problem. Summarising, both pairs may only assume $\{\!\{\mathbf{0}, -\}\!\}$, and $\{\!\{-\}\!\}$ describes the subconfiguration for either I/O SiDB. For the case of Figure 2.4b, then, we may concatenate all these multisets, giving $m = \{\!\{\mathbf{0}, \mathbf{0}, -, -, -, -\}\!\}$. The all-singleton *charge spaces* of the two pairs and the perturber are *joined* into the charge space of $\boldsymbol{S}$: $\{m\}$, thus inheriting the property of being singleton. This is shown in Figure 2.5.

Inspecting the metastable multiset charge configurations of $\boldsymbol{S}$, we may state that $\{m\}$ is actually a *least upper bound* in the following sense: it describes an 'upper bound' to the charge space of $\boldsymbol{S}$, where we consider some set of multisets in this context an upper bound precisely when *all* metastable charge configurations can be extracted from it by assigning the elements to separate SiDBs. Since $\{m\}$ was constructed from respective upper bounds, i.e., singleton sets of multisets that represented all metastable charge *sub*configurations independently, the *join* that is $\{m\}$ should trivially be an upper bound. Finally, it is also the *least* such: we know that any σ that is an upper bound to the charge space of $\boldsymbol{S}$ must at least contain the elements that formed m. Therefore, σ is at least the singleton $\{m\}$.

[7] When we prefix 'sub' to a term $\boldsymbol{A}$, we often consider $\boldsymbol{A}$ while implying the (possible) existence of a larger term of the same or similar structure that is constituent of *at least* $\boldsymbol{A}$.

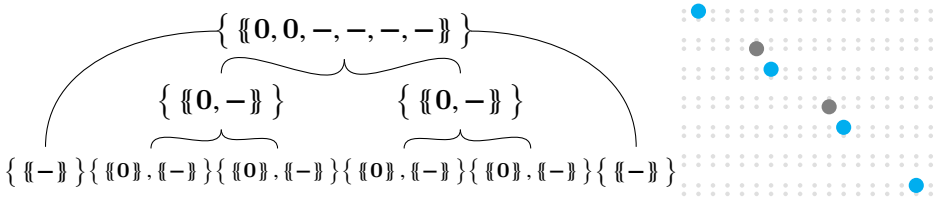


Figure 2.5: A representation of the *charge space* of a BDL wire segment carrying the **1** signal. This example builds on the analysis in Section 2.2.2, where the $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ and $\{\!\{-,-\}\!\}$ states were closed out for all pairs in our context. Therefore, the charge spaces for the pairs only contain $\{\!\{\mathbf{0},-\}\!\}$.

Compositionality and Bounds

It might seem unnecessary to reason about upper bounds on sets when only singletons come forth. Yet, zooming in to the charge space of any of the pairs, and in particular what charge spaces *composed* it, the mathematical machinery of sets and multisets in interleaved fashion appears more useful. For any of the pairs, we know by Figure 2.2 that the SiDBs contained therein may only assume SiDB**0** or SiDB**−**, which is captured by the charge space $\{\{\!\{\mathbf{0}\}\!\},\{\!\{-\}\!\}\}$. This metastability domain for the SiDBs themselves directly closes out any two-element multiset with a SiDB**+** to represent a metastable charge configuration of the pair, since it cannot be constructed from the charge spaces that represent minimal upper bounds for the pair constituents independently. However, we may construct the charge space of the pair *initially* as the complete combination the the sub-charge spaces: $\{\{\!\{\mathbf{0},\mathbf{0}\}\!\},\{\!\{\mathbf{0},-\}\!\},\{\!\{-,-\}\!\}\}$. Yet, we know from considerations in Section 2.2.2 that this upper bound on the metastable charge configurations of the subsystem—i.e., the pair—may be reduced to $\{\{\!\{\mathbf{0},-\}\!\}\}$.

This process of this specific charge space reduction can be described by the following steps. First, since positive charges cannot occur in $\boldsymbol{S}$, we are able to discard $\{\!\{-,-\}\!\}$ from the charge space of any pair in $\boldsymbol{S}$, precisely as in Section 2.2.2. Thereby, the *bound* on the *electrostatic potential* that each such subsystem can *project* onto all other subsystems in consideration may be adjusted accordingly. Compared to assuming the possibility of $\{\!\{-,-\}\!\}$, a projected bound would become *tighter* with the new information that $\{\!\{-,-\}\!\}$ is excluded.[8] Then, since the bounds are *relative* to a recipient subsystem, the actual charge configurations of a projecting subsystem of which the remaining charge space is now $\{\{\!\{\mathbf{0},\mathbf{0}\}\!\},\{\!\{\mathbf{0},-\}\!\}\}$, is now chosen such that it

[8] We say 'tighter' to hint at a more general bound update procedure that implements respective lower and upper bound updates. By (2.1), then, the exclusion of positive charges gave a lower bound update, while here we focus on upper bound updates through *pruning* negative charge states. This paragraph aims to give a rough intuition for the crucial 'prune→update→prune→update→ … ' procedure that is part of the algorithm presented in Section 2.5.2.

represents a *maximal* perturbance, which, in turn, was used to show that $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ is impossible in the perfect world that we assumed with the logic design wire model.

It is important to understand that $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ represents two underlying charge configurations. The notation facilitates a perspective on the state of the subsystem that is the pair: the interaction between *it* and every other subsystem independently does not vary greatly between the two. To connect back to the physical intuition, the pairs were formed with the intention of a balanced system energy over the alterations by having each BDL pair share an additional electron whose position represents a binary signal. Therefore switching to an approach of 'occurrence counting', we are allowed to take a *minimal* higher level view for which charge occurrence is respected with an according multiset representation. Since we consider the charge states of hierarchical groups, the appropriate concept of *compositionality* provides a means to capture that which is not represented in the condensed information form of a multiset.

2.3.2 Properties of Atomic Identity Logic

With these tools for synthetic description of charge configurations in a system, we may formalise what it means for a wire segment to represent a *signal*. Let $\vec{p}$ be a sequence of BDL pairs that constitute a proposed wire.[9] We now propose limits on inter-DB band bending effects, in order to consider an *idealistic* wire. This is a wire that, given any pair in the middle of an arbitrary long wire segment terminated by an output perturber, operates according to the presence, or absence of the input perturber, aligning to its neighbours (see Figure 2.4).

Let p_k for $k \in \mathbb{Z}$ represent a pair in an arbitrary long wire segment, and p_0 will our point of reference. In each pair p_k, there are two SiDBs that we can label $p_k^{(0)}$ and $p_k^{(1)}$ such that for all $b \in \{0,1\}$,

$$\left(\forall k' \in \mathbb{Z}_{>0} \,.\, d_{p_k^{(b)},p_{k'}^{(0)}} < d_{p_k^{(b)},p_{k'}^{(1)}}\right) \wedge \left(\forall k' \in \mathbb{Z}_{<0} \,.\, d_{p_k^{(b)},p_{k'}^{(0)}} > d_{p_k^{(b)},p_{k'}^{(1)}}\right). \tag{2.7}$$

This asserts that the wire is *efficient*, i.e., it is not longer than it needs to be to cross a certain distance. A wire of this form would now represent a non-kinked signal if and only if we have:

$$\forall k, k' \in \mathbb{Z} \,.\, \alpha\left(p_k^{(0)}\right) = \alpha\left(p_{k'}^{(0)}\right) \wedge \alpha\left(p_k^{(1)}\right) = \alpha\left(p_{k'}^{(1)}\right) \tag{2.8}$$

Without I/O perturbances, the signal we are looking to analyse is already undefined on the level of logic; not even the identity function is being

[9] The wire shown in Figure 2.2a may be taken as a visual aid, however note that we define properties of what makes a wire in this section, thus not assuming any given wire pattern this time.

computed. Therefore, our arbitrarily long wire is considered to be terminated with an output perturber as seen previously. We work towards defining a relation between the presence of an input perturber and the signal that the state of the wire encodes.

Now take $\bar{k} \in \mathbb{Z}_{>0}$ to be the last pair in the wire, such that this pair is terminated with an output perturber that is SiDB o. In order for this perturber to be effective, it is placed such that its charge state is always SiDB−—in Section 2.3.1, it is implied that there are many output perturber placements for which this is the case. We need $V_{o,p_{\bar{k}}^{(0)}} > V_{o,p_{\bar{k}}^{(1)}}$ so that we provide the influence needed to enforce the **0** signal in operation without an input perturber. This signal may be observed at SiDB $p_{\bar{k}}^{(1)}$, evaluated by $\alpha\left(p_{\bar{k}}^{(1)}\right) \stackrel{?}{=} -1$. Promptly, we can form a first bound on the chargeless potential local to $p_{\bar{k}}^{(1)}$, since it is prerequisite for the binary character of the wire that $p_{\bar{k}}^{(1)}$ captures a two-state phenomenon. Therefore: $V_{o,p_{\bar{k}}^{(1)}} > \mu_-$, such that (2.3) states that **1** is an observable output.

Furthermore, for our wire to be *consistent*, i.e., without kinks, we will place bounds on the relation between $p_{\bar{k}}^{(0)}$ and $p_{\bar{k}}^{(1)}$, and between those and the rest of the SiDBs in consideration. If, as considered in Section 2.2.2, the intra-pair distance is small enough such that **Rule 3** applies, while positive charges are avoided, then we only need to focus on the invalidity of $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ for any pair in the wire. For the moment, this will make the reasoning simpler, so we adopt this principle. First, this allows us to extend our previously formulated bound:

$$V_{o,p_{\bar{k}}^{(1)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(1)},p_{\bar{k}}^{(1)}} > \mu_-. \tag{2.9}$$

This further ensures that **1** is an observable output under the conditions of the rest of the wire behaving according to the **1** signal. Ideally, we would like that in the case of the rest of the wire behaving according to the **0** signal, **0** is the observable output and it is found in the only metastable subconfiguration of $p_{\bar{k}}$ that is $\{\!\{\mathbf{0},-\}\!\}$. This is to say two things independently. Primarily, we must exclude $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ from being attainable by $p_{\bar{k}}$ under this assumption of the alignment of the rest of the wire. This is asserted by the following:

$$\bigwedge_{b \in \{0,1\}} \left(V_{o,p_{\bar{k}}^{(b)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(0)},p_k^{(b)}} > \mu_- \right), \tag{2.10}$$

which translates to the intuitive question: can the local band bending effects at both elements of $p_{\bar{k}}$ independently of each other push the $(\mathbf{0}/-)$ charge transition threshold past E_F, given the output perturber and the rest of the wire representing the **0** signal? If the answer is 'yes', the assertion fails, implying that the two-state property is violated at $p_{\bar{k}}$, as $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ may describe its state.

Secondly, carrying now the assumption that $\{\!\{\mathbf{0}, -\}\!\}$ describes an upper bound on the charge configurations of $p_{\bar{k}}$, we formulate the requirement for the correct output to be observed at $p_{\bar{k}}^{(1)}$ as its only metastable configuration. That is:

$$V_{o,p_{\bar{k}}^{(1)}} + \sum_{k \in \mathbb{Z}_{\leq \bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(1)}} < \mu_-, \tag{2.11}$$

ensuring, in combination with (2.10), that the negative charge is placed at $p_{\bar{k}}^{(0)}$, and thus, $\mathbf{0}$ is observed. Note that when we assume the state $\{\!\{-, -\}\!\}$ to be invalid for $p_{\bar{k}}$ due to a intra-pair distance tuned to **Rule 3**, then (2.11) holds already from $V_{p_{\bar{k}}^{(0)},p_{\bar{k}}^{(1)}} < \mu_-$ alone. Alternatively, we may imply the consequence from stating that $p_{\bar{k}}^{(0)}$ must be SiDB$-$:

$$V_{o,p_{\bar{k}}^{(0)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(0)}} > \mu_-, \tag{2.12}$$

Finally, via unification of (2.11) and (2.12), we may state a property for the output pair under the effect of the output perturber and the rest of the wire that represents the $\mathbf{0}$ signal:

$$\left(V_{o,p_{\bar{k}}^{(0)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(0)}} > \mu_-\right) \wedge \left(V_{o,p_{\bar{k}}^{(1)}} + \sum_{k \in \mathbb{Z}_{\leq \bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(1)}} < \mu_-\right), \tag{2.13}$$

which tells us what is needed for $p_{\bar{k}}$ to be in accordance with the rest of the wire. To rule out the remaining manifestation of a $\{\!\{\mathbf{0}, -\}\!\}$ configuration, we state what is needed to deny metastability of the alternative BDL state. It suffices to show this by showing that for either $p_{\bar{k}}^{(0)}$ *or* $p_{\bar{k}}^{(1)}$, the respective charge state associated with a kink in the wire at $p_{\bar{k}}$ is not metastable:

$$\left(V_{o,p_{\bar{k}}^{(0)}} + V_{p_{\bar{k}}^{(1)},p_{\bar{k}}^{(0)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(0)}} > \mu_-\right) \vee \left(V_{o,p_{\bar{k}}^{(1)}} + \sum_{k \in \mathbb{Z}_{<\bar{k}}} V_{p_k^{(0)},p_{\bar{k}}^{(1)}} < \mu_-\right), \tag{2.14}$$

where we disregard whether there could be enough energy to cause SiDB$+$ for now. By focusing, e.g., on the first part of (2.14) and getting this in accordance with (2.13), we learn that $V_{p_{\bar{k}}^{(1)},p_{\bar{k}}^{(0)}} > \mu_-$ if we want these to be compatible. This, however, denies the shortcut for showing non-metastability of $\{\!\{-, -\}\!\}$, for which the intuition was given by **Rule 3**. We now lift the above inequalities to more intuitive predicates, in order to continue the discussion on the different states attainable by $p_{\bar{k}}$ in a more expressive way.

In particular, it is obvious that the expressions in (2.9) through (2.14) share a similar structure, and thus convey similar relations internally. While the 'direct' notation used here presents a non-obscured view of the expression's internals, the verbosity simultaneously obscures a higher perspective on the ideas that are being conveyed—the intention of the next section is to resolve this.

2.3.3 First Order SiDB Predicates

We continue on the arbitrarily long wire case $\vec{p}$, this time incorporating predicates to capture static mathematical ideas throughout a consideration, in order to arrive at a definition of an idealistic wire at the end of this section. These definitions exemplify the use of a logic design model, which is shown to synergise well with predicate logic as we will see now. Thereby, the remainder of Section 2.3 sketches a constraint satisfaction approach to physically forced SiDB logic design, as elaborated at the end of this section. First, this is demonstrated for wires in this section, then for simple combinational logic in Section 2.3.4.

To begin, we define a predicate $\mathsf{Witness}$ on an SiDB index b in a pair p_k in a BDL wire of arbitrary length ($k \in \mathbb{Z}$). Hereby, an interface for defining assertions within the previously introduced logic design model is offered, thus generalising the structure in the expressions of the previous section. The truth value of the evaluated predicate expresses $p_k^{(b)}$'s relation to the various charge transition thresholds in relation to the signals $s_1, s_2 \in \{0, 1\}$ represented on either side of p_k on the wire, assuming these signals are independently consistent.[10] To facilitate dynamic application of the predicate with respect to internally induced band bending effects, we abstract over an arbitrary real value representing this potential.

$$(s_1, s_2) \vdash \mathsf{Witness}^{-}_{k,b}(v) \coloneqq \left(\sum_{k' \in \mathbb{Z}_{<k}} V_{p_k^{(b)}, p_{k'}^{(s_1)}} \right) + \left(\sum_{k' \in \mathbb{Z}_{>k}} V_{p_k^{(b)}, p_{k'}^{(s_2)}} \right) > \mu_- - v \tag{2.15}$$

As the name suggests, the above predicate asserts that $p_k^{(b)}$ is a *witness* with respect to the SiDB$-$ charge state, i.e., the SiDB locally satisfies metastability for this charge state (see (2.3)). This is meant in the following sense: in our environment, its local potential with respect to the wire segments that cast a potential sum associated with the respective signal, is above μ_- relative to a dynamic potential v. Thus, it should be noted that, *universally*,

$$\forall \mathsf{CS}, \mathsf{CS}' \in \{-, \mathbf{0}, +\} \,.\, \mathsf{CS} \neq \mathsf{CS}' \wedge \mathsf{Witness}^{\mathsf{CS}} \implies \neg \mathsf{Witness}^{\mathsf{CS}'}, \tag{2.16}$$

where the versions of (2.15) for other CS are derived in accordance with (2.4) and (2.5).

Moving back to the examination in Section 2.3.2 of a wire terminated by an output perturber, we may reconnect to the discussion on $p_{\bar{k}}$, the last pair in the wire, by defining a special case for all s_2 and b:

$$\sum_{k \in \mathbb{Z}_{>\bar{k}}} V_{p_{\bar{k}}^{(b)}, p_k^{(s_2)}} \coloneqq V_{p_{\bar{k}}^{(b)}, o}. \tag{2.17}$$

[10] Signals will be given in an explicit context, which precedes $\vdash$ in our expressions.

We previously made an assessment of the metastable states of subsystem $p_{\bar{k}}$, influenced by the rest of the wire in a consistent $\mathbf{0}$ state. Now, we may write:

$$(0,0) \vdash \mathsf{Valid}_{\bar{k}}(\{\!\{\mathbf{0},\mathbf{0}\}\!\}) \implies \mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}(0) \wedge \mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}(0), \tag{2.18}$$

$$(0,0) \vdash \mathsf{Valid}_{\bar{k}}(\{\!\{-,-\}\!\}) \implies \mathsf{Witness}^{-}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \wedge \mathsf{Witness}^{-}_{\bar{k},1}\left(V^{\mathsf{int}}\right), \tag{2.19}$$

which appears in a much more intuitive form than considered previously. Here, $V^{\mathsf{int}} \coloneqq V_{p^{(0)}_{\bar{k}},p^{(1)}_{\bar{k}}}$, and Valid for a subsystem $p_{\bar{k}}$ describes whether a multiset $\boldsymbol{m}$ is a *lower bound* to the metastable states of the SiDBs in $p_{\bar{k}}$, in a dual sense compared to how upper bounds were formulated in Section 2.3.1, though targeting specific multisets instead of sets of those in a charge space. Specifically, we have that $\boldsymbol{m}$ is a lower bound on the multiset configurations of a pair $\boldsymbol{p}$ if and only if, in the considered context, there is a *composition* of $\boldsymbol{m}$ (see Figure 2.5) that is part of *a* metastable charge configuration for the system that $\boldsymbol{p}$ is contained in.

It is important to see the distinction between unidirectionality versus bidirectionality in the above implications; $\mathsf{Witness}$ asserts a part of what Valid entails. This means that by, e.g., showing $(0,0) \vdash \neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}(0)$, then this is a *proof* of $(0,0) \vdash \neg\mathsf{Valid}_{\bar{k}}(\{\!\{\mathbf{0},\mathbf{0}\}\!\})$. That is, under the assumption of the electrostatic environment associated with the wire state $(0,0)$, $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ is excluded from the charge space of $p_{\bar{k}}$.

Compositionality in Application

Going further with our synthetic description of identity logic, we may apply the defined predicates to the situation of a $\{\!\{\mathbf{0},-\}\!\}$ case. This is more involved, since we find by Figure 2.5 that the charge space of a BDL pair is not singleton, and thus we have to deal with multiple compositions. We can formulate the 'witnesses' that $\mathsf{Valid}_{\bar{k}}(\{\!\{\mathbf{0},-\}\!\})$ entails as the following, this time introducing disjunctions:

$$\begin{aligned}(0,0) \vdash \mathsf{Valid}_{\bar{k}}(\{\!\{\mathbf{0},-\}\!\}) \implies &\left(\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \wedge \mathsf{Witness}^{-}_{\bar{k},1}(0)\right) \\ &\vee \left(\mathsf{Witness}^{-}_{\bar{k},0}(0) \wedge \mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}\left(V^{\mathsf{int}}\right)\right).\end{aligned} \tag{2.20}$$

This in turn implies that under $(0,0)$, it suffices to show $\neg\mathsf{Valid}_{\bar{k}}(\{\!\{\mathbf{0},-\}\!\})$ by showing $\neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}\left(V^{\mathsf{int}}\right)$ and $\neg\mathsf{Witness}^{-}_{\bar{k},0}(0)$, *or* by showing $\neg\mathsf{Witness}^{-}_{\bar{k},1}(0)$ and $\neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}\left(V^{\mathsf{int}}\right)$. Then, for either SiDB in $p_{\bar{k}}$, replacing the summed potential from sources outside the pair to V^{ext} for the moment, we get:

$$\left(V^{\mathsf{ext}} + V^{\mathsf{int}} > \mu_- \vee V^{\mathsf{ext}} + V^{\mathsf{int}} < \mu_+\right) \wedge V^{\mathsf{ext}} < \mu_-. \tag{2.21}$$

However, assuming that $V^{\mathsf{int}}+V^{\mathsf{ext}}$ is not strong enough to cause SiDB$\mathbf{+}$, from the fact that V^{int} is a negative number, we get that this never holds, indicating with this that $p^{(0)}_{\bar{k}}$ and $p^{(1)}_{\bar{k}}$ both accept SiDB$\mathbf{0}$ and SiDB$\mathbf{-}$ under $p_{\bar{k}} \mapsto \{\!\{\mathbf{0},-\}\!\}$.

More interestingly, we want to make statements that attribute identity-computing behaviour to $p_{\bar{k}}$. That is to say:

$$(0,0) \vdash \mathsf{Witness}^{-}_{\bar{k},0}(0) \wedge \mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}\left(V^{\mathsf{int}}\right) \tag{2.22}$$

$$(1,0) \vdash \mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \wedge \mathsf{Witness}^{-}_{\bar{k},1}(0), \tag{2.23}$$

to specify that the orientation that follows the rest of the wire—i.e., the particular composition of $\{\!\{\mathbf{0},-\}\!\}$ that aligns with the BDL pairs in the state associated with the signal 0 and 1 for (2.22) and (2.23) respectively—is locally metastable at $p_{\bar{k}}$, and:

$$(0,0) \vdash \neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \vee \neg\mathsf{Witness}^{-}_{\bar{k},1}(0) \tag{2.24}$$

$$(1,0) \vdash \neg\mathsf{Witness}^{-}_{\bar{k},0}(0) \vee \neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}\left(V^{\mathsf{int}}\right), \tag{2.25}$$

to deny the physical validity of $p_{\bar{k}}$ *not* aligning with the rest of the wire.

Conditions to Physically Forced BDL Wire Alignment

After having reviewed the different conditions for our BDL pair $p_{\bar{k}}$ to orient in a $\{\!\{\mathbf{0},-\}\!\}$ charge configuration in accordance with the rest of the BDL wire, we may now compose requirements that force the aligned state by the metastability condition. That is:

$$\begin{aligned}(0,0) \vdash\ & \mathsf{Witness}^{-}_{\bar{k},0}(0) \wedge \mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}\left(V^{\mathsf{int}}\right) \\ & \wedge \left(\neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \vee \neg\mathsf{Witness}^{-}_{\bar{k},1}(0)\right) \\ & \wedge \left(\neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},0}(0) \vee \neg\mathsf{Witness}^{\mathbf{0}}_{\bar{k},1}(0)\right) \\ & \wedge \left(\neg\mathsf{Witness}^{-}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \vee \neg\mathsf{Witness}^{-}_{\bar{k},1}\left(V^{\mathsf{int}}\right)\right) \\ & \wedge \left(\neg\mathsf{Witness}^{+}_{\bar{k},0}\left(V^{\mathsf{int}}\right) \vee \neg\mathsf{Witness}^{+}_{\bar{k},1}\left(V^{\mathsf{int}}\right)\right),\end{aligned} \tag{2.26}$$

where the condition for $(1,0)$ is similar, with the first two lines adjusted according to (2.23) and (2.25). The lower three lines of the assertion stay the same, where the last line asserts SiDB+ may not occur, and the two lines above it are directly derived from (2.18) and (2.19). For the last line, by ensuring $\neg\mathsf{Valid}_{p_{\bar{k}}}(\{\!\{-,+\}\!\})$, this implies $\neg\mathsf{Valid}_{p_{\bar{k}}}(\{\!\{\mathbf{0},+\}\!\})$, and therefore also $\neg\mathsf{Valid}_{p_{\bar{k}}}(\{\!\{+,+\}\!\})$ by the way the charge transition threshold comparisons are defined. Intuitively, if $V^{\mathsf{ext}} + V^{\mathsf{int}} > \mu_+$, then $V^{\mathsf{ext}} > \mu_+$ and $V^{\mathsf{ext}} - V^{\mathsf{int}} > \mu_+$, since V^{ext} and V^{int} are strictly negative numbers in our consideration.

Observe now that we may swap out $\bar{k}$ for any $k \leq \bar{k}$ and make statements for different (s_1, s_2). For instance, we may want to create a requirement for the physical invalidity of kinks for a given BDL wire pattern. For $s \in \{0,1\}$:

$$\begin{aligned}(s,\neg s) \vdash\ & \neg\mathsf{Valid}_k(\{\!\{\mathbf{0},-\}\!\}) \wedge \neg\mathsf{Valid}_k(\{\!\{\mathbf{0},\mathbf{0}\}\!\}) \\ & \wedge\neg\mathsf{Valid}_k(\{\!\{-,-\}\!\}) \wedge \neg\mathsf{Valid}_k(\{\!\{-,+\}\!\}),\end{aligned} \tag{2.27}$$

where each invalidity proof may be given by showing the invalidity of one associated witness, as in, e.g., (2.18) and (2.19), or multiple in a disjunctive case, like in (2.20).

To conclude this part, the method of formalisation of silicon atomic logic properties in this section presents a means for precise analysis for a non-inverting logic computation of single arity fan-in and fan-out; i.e., the identity function, for which we gave an exemplifying logic design model as an arbitrarily long wire segment of BDL pairs.[11] By analysing the asymptotic behaviour of big sums of potential sources from increasing distances (see Section 2.2.2), it is here proposed that SMT solvers or linear programs may confirm *physically forced* behaviour of logic implementations for a given wire pattern in an otherwise undisturbed environment for a set of physical parameters.[12] Since the inter-DB potential effects are characterised by properties of the medium, it is possible that there exists an *optimal* medium that passes a given set of properties with the most headroom for external perturbances. In future work, it could be studied whether, in practice, such optimal physical properties of the medium exist that could be influenced by the fabrication process. [33]

2.3.4 Atomic Forced Combinational Logic

Having discussed the most simple case of identity binary dot logic on the silicon platform in a new level of detail, we may proceed to a discussion on combinational logic, where we take the simplistic OR gate shown in Figure 2.6 as a running example. We will first direct our efforts at formalising the required properties for physically forced consistent behaviour, after which we rejoin the status-quo of the SiDB field which relies on *ground state logic*, discussed in Section 2.4.

The OR gate shown in Figure 2.6 resembles the canonical one with which the potential of the SiDB platform for logic implementations was first shown in a fabrication experiment. [14] However, it should be observed that the method of applying binary inputs differs from Figure 2.4; the method there is more akin to the just mentioned OR gate from [14], though here we adopt a new standard set by [34]. It mimics the presence of connecting BDL pairs to the input pairs in BDL wire fashion, though for each of these 'ghost pairs', only the SiDB is placed that would be SiDB− if the pair were to represent the respective input signal.

Now, rather than directing our attention at *robust* gates, i.e., ones that

[11] It should be noticed that the straightforward formalisation of BDL principles is merely an example of a logic design model for wires. The inherent ternary character of the SiDB charge states motivates experimental efforts to construct a logic design model that may yield physically forced ternary SiDB logic—if it exists—at a revolutionary logic density.

[12] The property to a logic implementation that we call 'physically forced' behaviour here is called *temperature robustness* in the literature. [9]

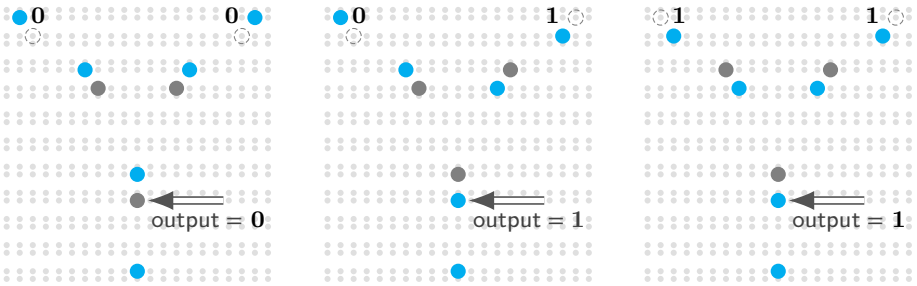


Figure 2.6: A symmetric BDL OR gate under the **00**, **01** and **11** inputs. Note that here, the *position* of the input perturber determines the input signal. The states shown are the only metastable charge configuration for our standard set of physical parameters: $\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$.

operate under varying circumstances, we consider this much simpler gate design for which we make no claims on its robustness.[13] As we intend to capture combinational logic principles in a style similar to Section 2.3.3, taking the straightforward logic gate design shown in Figure 2.6 will help us place our focus.

Physically Forced BDL OR Operation

In order to facilitate a reuse of the predicates defined in Section 2.3.3, we continue with the arbitrary long wire $\vec{p}$ framework. Yet, this time, we consider p_{-1} and p_1 to be the pairs next to the left and right input respectively, terminating respective wire segments $(p_k)_{k\in\mathbb{Z}_{<0}}$ and $(p_k)_{k\in\mathbb{Z}_{>0}}$ of arbitrary length. Contrary to our prior consideration of a pair in a wire, which has one end oriented towards it and the other away from it, both input wires are oriented towards p_0. Fortunately, our definition of Witness associates some $p_k^{(s)}$ with the SiDB in p_k that assumes SiDB− when p_k represents the s signal, so this is appropriately relative to p_0.

We will assume that each of these input wires represents a consistent signal, and we use (s_1, s_2) as a signal context much like in Section 2.3.3, referring to the signal in the left and right input respectively. Our pair of interest is p_0, with $p_0^{(1)}$ representing the OR gate output, as in Figure 2.6. With this, we may give a set of requirements that, if satisfied, ensures the logical OR operation is implemented with a proposed gate that fits our simple model. For all $s_1, s_2 \in \{0, 1\}$:

$$\begin{aligned}(s_1, s_2) \vdash\ & \neg\mathsf{Valid}_0(\{\!\{\mathbf{0}, \mathbf{0}\}\!\}) \wedge \neg\mathsf{Valid}_0(\{\!\{-, -\}\!\}) \wedge \neg\mathsf{Valid}_0(\{\!\{-, +\}\!\}) \\ & \wedge \mathsf{Valid}_0(\{\!\{\mathbf{0}, -\}\!\}),\end{aligned} \tag{2.28}$$

[13] Assessing gate operation under varying circumstances is called the *operational domain computation* [23, 24, 33]. We touch on this assessment method in Section 2.4.

and:

$$(0,0) \vdash \mathsf{Witness}^{\mathbf{0}}_{0,1}\left(V^{\mathsf{pert}} + V^{\mathsf{int}}\right), \tag{2.29}$$

$$\forall s \in \{0,1\}\,.\,(s, \neg s) \vdash \mathsf{Witness}^{-}_{0,1}\left(V^{\mathsf{pert}}\right), \text{ and} \tag{2.30}$$

$$(1,1) \vdash \mathsf{Witness}^{-}_{0,1}\left(V^{\mathsf{pert}}\right), \tag{2.31}$$

where now $V^{\mathsf{int}} \coloneqq V_{p_0^{(0)}, p_0^{(1)}}$ and $V^{\mathsf{pert}} \coloneqq V_{p_0^{(1)}, o}$, with o being the output perturber SiDB. Here, we have that (2.28) limits the charge space of p_0 to contain only $\{\!\{\mathbf{0}, -\}\!\}$—recall that the latter part of (2.28) implies $\neg\mathsf{Valid}_0(\{\!\{\mathbf{0}, +\}\!\}) \wedge \neg\mathsf{Valid}_0(\{\!\{+, +\}\!\})$, as touched on in Section 2.3.3. With that, we only need to specify the behaviour of, e.g., $p_0^{(1)}$; the other SiDB will be forced to take the other element of $\{\!\{\mathbf{0}, -\}\!\}$ by the associated requirement in (2.28).

Assertions for Gate Operation

In much the same way, properties can be synthesised that assert the physically forced operation of implementation of other logical operations like AND, XOR, NAND, etc. Often, however, more DBs are needed besides the input wires, perturbers and an output pair. Any addition here adds up exponentially, but we may extend our model to capture this nonetheless. We continue our assumption of two-state input wires to make the situation approachable, since the interaction between input wires in practice poses a threat to the logical functioning of a gate. In order to take the SiDBs that are placed in the *logic canvas* into account, we now write $\mathsf{Valid}_k(m)(v_0, v_1)$ to enable a dynamic operation of this predicate also.[14] This means that the v's are now added to the dynamic argument of the respective witness it entails, e.g.: $\mathsf{Valid}_0(\{\!\{\mathbf{0}, \mathbf{0}\}\!\})(v_0, v_1) \implies \mathsf{Witness}^{\mathbf{0}}_{0,0}(v_0) \wedge \mathsf{Witness}^{\mathbf{0}}_{0,1}(v_1)$.

Furthermore, we write $\boldsymbol{L}$ for the set of SiDBs in the logic canvas, distinct from the SiDBs in consideration before. Our goal will be to formulate the properties required for gate operation, observable like before in p_0, where this time we account for an arbitrary amount of SiDBs in $\boldsymbol{L}$. This is already a tricky situation, since we do not put requirements on how the SiDBs in $\boldsymbol{L}$ are placed in relation to each other, hence positive charges may occur. The intent of this demonstration is thus to illustrate how an exhaustive analysis of an arbitrary gate rapidly grows in complexity. This implies that more clever techniques may be required that are more considerate of $\boldsymbol{L}$'s specifics, for instance by analysing $\boldsymbol{L}$ for groups of SiDBs that show synchronised behaviour.

[14] The logic canvas of a gate is where the additional SiDBs are placed. It is usually considered to be bordered by a frame determined by the placements of the input and output wires. [34] Note that for simplicity we only consider single output perturbers here, not multiple, nor wires.

Lastly, in order to use our Witness predicate on SiDBs in $\boldsymbol{L}$, we may subscript an element of $\boldsymbol{L}$, which then replaces $p_k^{(b)}$ in the definition (see (2.15)). Being loose with the notation, we will accept superscripts to Witness from $\{-1, 0, +1\}$ because the intention behind writing this is obvious. Now we may present our statement for all $\alpha \in \{-1, 0, +1\}^{\boldsymbol{L}}$, thus requiring exponential scaling in $|\boldsymbol{L}|$. Let $\forall b \in \{0,1\} \,.\, v_b \coloneqq -\sum_{l \in \boldsymbol{L}} V_{p_0^{(b)},l} \cdot \alpha(l)$. We assert:

$$\begin{aligned}(s_1, s_2) \vdash \neg\mathsf{Valid}_0\left(\{\!\{\mathbf{0}, \mathbf{0}\}\!\}\right)\left(v_0, v_1\right) \wedge \neg\mathsf{Valid}_0\left(\{\!\{-, -\}\!\}\right)\left(v_0, v_1\right) \\ \wedge \neg\mathsf{Valid}_0\left(\{\!\{-, +\}\!\}\right)\left(v_0, v_1\right) \wedge \mathsf{Valid}_0\left(\{\!\{\mathbf{0}, -\}\!\}\right)\left(v_0, v_1\right),\end{aligned} \tag{2.32}$$

ensuring like in (2.28) that only $\{\!\{\mathbf{0}, -\}\!\}$ can be attained by p_0. For the logic verification part, we let $f : \{0,1\}^2 \to \{0,1\}$ describe our 2-ary Boolean function. We define a functor lifting binary values to associated BDL charge states: $0 \overset{\mathsf{BDL}}{\longmapsto} \mathbf{0}$, and $1 \overset{\mathsf{BDL}}{\longmapsto} -$. For all $l \in \boldsymbol{L}$, let $v_l \coloneqq -\sum_{l' \in \boldsymbol{L}} V_{l,l'} \cdot \alpha\left(l'\right)$. We use this to verify local metastability at l, alongside verification of the logic operation. For all $s_1, s_2 \in \{0,1\}$:

$$\begin{aligned}(s_1, s_2) \vdash \mathsf{Witness}_{0,1}^{\mathsf{BDL}(f(s_1,s_2))}\left(V^{\mathsf{pert}} + \left(1 - f\left(s_1, s_2\right)\right) \cdot V^{\mathsf{int}} + v_1\right) \\ \wedge \forall l \in \boldsymbol{L} \,.\, \mathsf{Witness}_l^{\alpha(l)}\left(V_{l,o} + v_l + V_{l,p_0^{(f(s_1,s_2))}}\right).\end{aligned} \tag{2.33}$$

Let us elaborate on the above formula, in particular clarifying the generalisation over the logic function to verify the implementation of. From the more trivial example in (2.29) through (2.31), we observe that if the witness assertion is made for $\mathsf{BDL}\left(f\left(s_1, s_2\right)\right)$ in all cases. Moreover, when $f\left(s_1, s_2\right) = 0$, then the SiDB$-$ is at $p_0^{(0)}$ and vice versa, thereby expressible by $p_0^{(f(s_1,s_2))}$. Lastly, if and only if a witness for $\mathbf{0}$ is required, the internal potential for the pair should be considered, hence the inversion $\left(1 - f\left(s_1, s_2\right)\right)$.

The second conjunct in (2.33) ensures that all SiDBs in the logic canvas adhere to local metastability. The fraction of the local potential received by other SiDBs in the logic canvas is negated to v_l such that its polarity is consistent with our other definitions that operate on the non-positive range of the screened Coulomb potential from (2.1). Both the effect of the always-SiDB$-$ output perturber, and the SiDB$-$ in the output pair in the SiDB index corresponding to $f\left(s_1, s_2\right)$ are accounted for.

Gate and Wire Design

It should be noted that the OR gate in Figure 2.6, while showing physically forced behaviour for the tiny input wires shown, fails to do so when the input wires get longer. With each additional BDL pair added to an input wire, the global energy level rises, which makes room for a greater variety of physically valid states. Local band bending effects that only just fell short

of a threshold may now cross it, setting in effect a chain of variations, as this crossing may enable another SiDB to experience the same, and so forth. Hence, the discussion here is primarily of theoretic interest; discovering what can be stated in this domain.

Albeit not performed with this work, automated methods built on the ideas presented herein may be able to show whether physically forced logic gates even exist for a given model like any of the above. By considering the asymptotic behaviour of the received effect of a BDL wire of increasing length, the problem could be made scalable using fair approximations, or—even better—*solutions* to the limits like the one seen in Figure 2.3b. Posing a sequel to [10], in which an exact SiDB simulator (discussed in Section 2.5) was used to fill a logic canvas with a minimal amount of SiDBs that enables a desired ground state logic computation, the question of existence of physically forced logic gates could be entertained using a similar methodology. Exhaustive searches for candidate wire patterns and/or logic gates could, depending on the model, rely on exact SiDB simulator queries that quickly surpass the current limitations in the field. *However*, the exact SiDB simulator to be presented in this thesis bumps the threshold of computational intractability by a considerable amount.

It should be stated that although we previously considered environments that were electrostatically undisturbed outside of the SiDBs in consideration, this disregards the imminent presence of $\mathrm{H-Si(100)\text{-}2\times1}$ surface defects. [7] In particular, *charged defects* pose a threat to our assumptions, since, as the name suggests, these defects interact in the potential landscape, and their effect can be simulated. [23] Fortunately, there have been advancements in *design automation* of SiDB technology, with *defect-aware physical design* putting forward a methodology powered by convolutional neural networks to circumvent placement of logic implementations on or near (charged) defects. [32] This motivates the sensibility of a discussion under perfect conditions, as this allows us to focus on particular problems where surface defects play no integral role. The aforementioned methodology puts forward a remedial solution, compensating for (a part of) the shortcomings of the considered model in practical use.

In the context of ground state logic, discussed next, a gate is operational if and only if the ground state computes the desired function under all inputs. This leaves a great number of candidates that fulfil this role, of which an optimum must be picked. This is an ongoing study, with the current standard set by the reinforcement learning-powered *Bestagon* gate library, which also standardised a wire pattern and logic canvas size. [34] With the novelty of the field, however, there is still much gain with the advent of more powerful exact SiDB simulators that enable gate creation like in [10], as they provide precise means for verification needed for true *optimality* of a proposed design within a search domain.

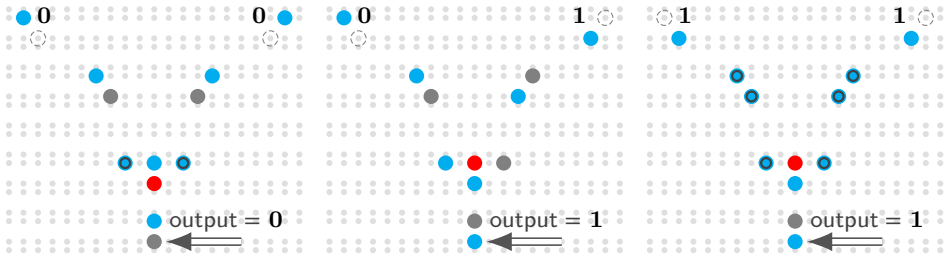


Figure 2.7: A symmetric OR gate under the **00**, **01** and **11** inputs that uses a positive charge (marked red) instead of an output perturber. The states shown are the metastable charge configurations that are minimal in system energy; they are thus *ground states*. Note that we are seeing *degeneracies* in these ground states: multiple pairs of SiDBs can share a second electron in the above layout, as marked by the dark circles superimposed on the **−** charge state markings. For the **11** input, the two input BDL pairs are connected such that when the rightmost pair has the SiDB**−** at the bottom DB, the leftmost pair will have it at the top DB, and vice versa. Simulated with our standard set of physical parameters: $\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$.

Hence, for the next section, we aim to formalise an *ordering* of metastable global states, for which the ground state will then be the bottom element. By providing a framework of reasoning for any given ground state-oriented problem like optimal gate design, we intend to present inequality relations that may serve guidant to a formulation of optimisation heuristics for the given problem.

2.4 Ground State Logic

As promised, we will now move away from the binary theoretical considerations of physically forced logic, and instead judge a logic implementation on the SiDB platform to be operational if and only if the correct logic computation can be observed in the ground state, i.e. the state of minimal energy. If a system has multiple metastable charge configurations, we refer to the ordered equivalence classes of system energy—where each equivalence class contains either a single state of a set of degenerate states—as follows: primarily, there is the ground state, then the *first excited* state, then the *second excited* state, and so on. Hence, e.g., the first excited state could be a set of degenerate states. In smaller symmetric systems these may occur with some regularity, though for larger systems that are typically non-symmetric, containing multiple gates for instance, degenerate states are very rare.

In Figure 2.7 and Figure 2.8, an example of ground state logic can be seen, as the presented OR gate has multiple metastable states, of which the

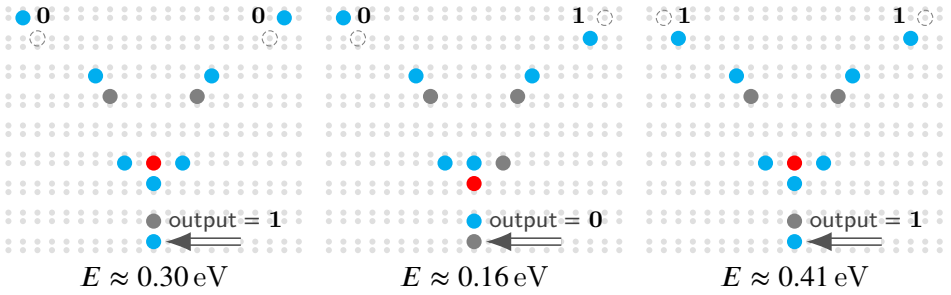


Figure 2.8: The same SiDB logic gate as in Figure 2.7, but this time the *first excited states* are shown; these are the metastable states that are lowest in energy from all metastable states, apart from the ground state. For a certain rise in system temperature, these states are more likely to be seen than the ground state, yet they do not conform to the behaviour of an OR gate. [9] Simulated with the same physical parameters as in Figure 2.7.

respective ground states compute the OR function. Simultaneously, this tiny gate acts as a proof of concept for silicon logic without output perturbers, instead using the attractive force of a positive charge that is induced due to a local dense SiDB population. Actually, when utilising the input application method used for the sub-30 nm^2 OR gate from [14], where no input perturber indicates the **0** signal as in Figure 2.4, this layout may form a new record for *smallest OR implementation* at sub-22 nm^2.[15]

This, however, does not mean that this gate is useful in a practical sense, since these dimensions are too small for individual addressability of such a gate by current-day expectations of clocking electrodes that must be placed in relation to a network of gates in order to ensure a directed signal flow. [34] Instead, this proof of concept should inspire research like [10] in which minimal logic gates were sought for exhaustively for a given I/O environment, this time considering the means of positive charges as replacements for output perturbers.

When assessing the operation of a given logic implementation, the *temperature* in the system becomes an important consideration. In particular, the temperature is directly connected to which of the ordered metastable states is to be observed. [9] At 0 K, the ground state is always occupied by the system (theoretically), then, after a temperature threshold is crossed, the probability of the system occupying the first excited state increases. Crossing another threshold, the second excited state may occur, etc. This introduces

[15] The input perturber for the first input would then be placed one dimer row up and one column to the left from the top DB in the leftmost input BDL pair, and symmetrically for the other input. By moving the input perturbers each one column inwards, a sub-18 nm^2 AND gate is formed. Furthermore, this claim of a 'record' should be taken with a grain of salt, since assumed properties of the medium differ in μ_-: here we used $\mu_- = -0.32$ eV, $\epsilon_r = 5.6$, $\lambda_{TF} = 5$ nm, but in [14], their OR gate was shown operational for $\mu_- = -0.23$.

the concept of *temperature robustness* of a logic implementation, where the notion that we previously considered as 'physically forced behaviour' would now be the property of being *temperature robust*; i.e., if there is only one metastable state, the system temperature cannot affect this. Furthermore, the concept of being *operational* can be stretched to accept correct logic behaviour in *transparent* excited states, with which the degree of robustness would then correspond with the lowest temperature at which the logic function to implement is defied; i.e., the first *erroneous* state. [9] This may therefore cross a number of excited states that adhere to the specification.

2.4.1 Ordered Global States

Previously we elaborated on the robustness of a logic implementation with regards to temperature. In that scope, we consider a comparison of system energy levels associated with the metastable states only. However, the concept of robustness in this context goes beyond temperature robustness only: efforts have been made to assess resilience under variations in the electrostatic landscape or properties of the medium. [23, 33] The methodology here comes down to the application of appropriate SiDB simulator engines, using their judgements in a binary fashion; i.e., 'for the given parameters and environment, the logic specification is/is not adhered to'. Yet, while such simulators are incredibly useful and adept in making such judgements, they give us no *direct* indication of the quality of, e.g., a logic computing ground state. Instead, some space needs to be searched in order to make *indirect* quality assessments through relating a region of said space through binary judgements. For instance, [33] searches a physical parameter space, efficiently mapping an operational domain. The quality assessment can now be made through analysis of the obtained operational domain, which, by now, is a canonical form of assessment. [22, 24, 30]

In this, we intend to formalise a framework from which quality assessments may be made *directly*, though this implies an exhaustive consideration of all exponentially many states. However, through relating these states by an ordering, an efficient selection could be made of states that require further inspection. This being a development for the ground state logic framework, the system energy of a state will have a prominent role. Moreover, the metastability of a state is considered, though through a novel lens: instead of a binary assessment of physical validity, with in particular an assessment of the population stability criterion (see (2.3) through (2.5)), we now aim to capture a *degree* of population stability for a state, ultimately coupling this to the associated system energy.

We set ourselves up in a most general fashion. Given is a system $\boldsymbol{S}$ of SiDBs. Recall that for each $i \in \boldsymbol{S}$, V_i defines the local potential at i, which includes the effect of external potential sources such as charged defects (see (2.2)). Now, we want to relate the global states of $\boldsymbol{S}$ exhaustively, but this

means relating $3^{|S|}$ objects. Perhaps many of those, however, are not relevant for the goal of attributing a certain quality to $\boldsymbol{S}$, so we will need a clever way to know which states we may discard. In our context in which we create an ordering, 'discarding' may mean: 'throw this in the top bin', implying that we are interested in what ends up below it. Note our use of the word 'bin'; with it we introduce the resulting structure to be an ordered collection of bins, i.e. equivalence classes. This synergises with the concept of a *preorder*, which will provide us a means to deal with structure. The ordering relation in a preorder is transitive and reflexive, but not antisymmetric as in a *partial order*, allowing us to form equivalence classes of global states that are mutually related, whilst keeping those in the same bin as separate objects.[16]

Relating States by Relevancy

Take $\mathscr{G}_S \coloneqq \{-1, 0, +1\}^S$ to be the set of global states over $\boldsymbol{S}$. Before defining an ordering over $\mathscr{G}_S$ directly, however, let us first build an intuition for what this relation should entail by considering how either furthest end of the relation spectrum should look like. We previously discussed that states that are not relevant to a quality assessment of $\boldsymbol{S}$ may move to the top, i.e. every element in $\mathscr{G}_S$ will be below it. A higher ordered element in this context should receive a semantic interpretation of having a higher level of confidence attributed that we may omit the associated global state from consideration; i.e., a higher *irrelevancy*. Intuitively, such states could have a system energy in the highest ranges that $\boldsymbol{S}$ can deliver, or would be *grossly* defying the population stability criterion, by, e.g., defying it for each SiDB individually by a considerable amount.

In our n-doped context, the latter will *always* be the case for the all-SiDB**+** state. Note that for very small or very sparse systems, the ranges in system energy might be thin, so the former property does not put forward an obvious way to judge some $\boldsymbol{\alpha}$ to join the top bin. Inherently to ground state logic, which relies on principles from thermodynamics, the global states should be assessed *in relation to each other*: if, in a purely theoretic sense, we have a million physically valid states out of a million, with a massive difference in energy between the ground state and the last excited state, we would be able to 'discard' the latter. Yet if we have, e.g., 3 physically valid states out of a million with similarly a massive energy difference, the last excited state remains an object of interest due to its rarity—unless we find that the temperature will need to rise beyond what will be considered a reasonable level for a given context before occupation of this state would start to become probabilistically reasonable. This therefore introduces two

[16] In loose terms: reflexivity means $x \leq x$, transitivity means $x \leq y \wedge y \leq z \implies x \leq z$, antisymmetry means $x \leq y \wedge y \leq x \implies x = y$.

parameters: a maximum system temperature to consider T^{max}, and the minimum probability P^{min} of occupation of a state before we account for it.

On the other side of the spectrum, we would like to find the states that *are* relevant to a quality assessment of S. The most prominent candidate to step forward is of course the ground state, and perhaps some excited states. Furthermore, there might be states much below the ground state in system energy that require just a tiny bit of potential of some polarity before a threshold is crossed at some SiDB that hindered the metastability of the entire layout. It is precisely the existence of these 'bottlenecked' states that motivate the entirety of this novel approach. Invalidity may depend on a fraction of potential energy local to a single SiDB, yet contemporary SiDB simulators either return only metastable states, or do not attribute non-metastable states with a *metastability proximity* heuristic.

More specifically, our interpretation of the relation of a state with metastability will be expressible with negative numbers also, thereby distinguishing the 'proximity' of the positive fragment from the negative *population rigidity*. This is exactly what is defined in the following adaptation of the population stability criterion (see (2.3) through (2.5)), which now takes an error offset $e \in \mathbb{R}$:

$$\begin{aligned} \mathsf{Stable}_\alpha(e) \coloneqq \bigwedge_{i \in S} \Big(& \big(\alpha(i) = -1 \wedge V_i < -\mu_- + e\big) \\ & \vee \big(\alpha(i) = +1 \wedge V_i > -\mu_+ - e\big) \\ & \vee \big(\alpha(i) = 0 \wedge V_i > -\mu_- - e \wedge V_i < -\mu_+ + e\big) \Big). \end{aligned} \tag{2.34}$$

Note that a positive error 'widens' the fractions of the bandgap to overlap, thereby being more allowing, as opposed to being more restrictive for negative error value. The content of this equation is not anything novel, as implementations of prominent SiDB simulation algorithms like presented in [8, 11, 24]. However, the perspective shift lies in seeing the error as *dynamic*, where *it* defines the interest relation directly. This therefore contrasts contemporary simulation-powered space searches with *static* error, while at the same time posing compatibility as a new dimension to explore. Also space searches that involve a multitude of local analyses as in [23] are compatible with the proposed, as static local external perturbations are captured in V_i. However, in the context of robustness assessments in a conceptual sense, our approach opposes contemporary methods by being well fit for a direct, single-simulation type of assessment that captures *generic robustness*, whilst leaving an interface for specialised instances with local or global variations to the SiDBs or properties of the medium.

The above Stable predicate allows us to form a requirement for a *minimal stability error* $\mathfrak{e}_\alpha$:

$$\mathfrak{e}_\alpha = \min\{e \in \mathbb{R} \mid \mathsf{Stable}_\alpha(e)\}. \tag{2.35}$$

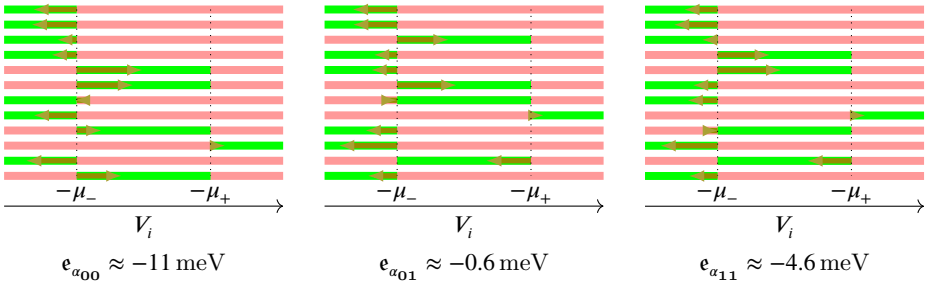


Figure 2.9: From left to right: the stability errors for the ground state of the OR gate shown in Figure 2.7 under inputs **00**, **01** and **11** respectively. Of the degenerate ground states only a single one is displayed. Each of the horizontal bars is associated with an SiDB; from top to bottom, the order of the SiDBs corresponds with traversing the respective OR gate state in Figure 2.7 top left to bottom right, as you would read a book. The charge states for each SiDB for the given states are shown by an associated fraction of the band gap highlighted. The arrows point to the value of the local potential at that SiDB, and their lengths are associated with the stability error for that SiDB. Note that the arrow's origin is always the nearest charge transition threshold. Since only metastable states are shown, the arrows all point somewhere within the highlighted ranges, and thus represent *negative stability error*. Thereby, taking the maximum of the stability errors shown here for each state individually corresponds with taking the length of the respectively shortest arrow; these are the decimal values shown.

We propose that this value can be obtained directly by considering only a finite set that is our system $\boldsymbol{S}$:

$$\mathfrak{e}_\alpha \coloneqq \max_{i \in S} \begin{cases} V_i + \mu_- & \text{if } \alpha(i) = -1, \\ -\mu_+ - V_i & \text{if } \alpha(i) = +1, \\ -\mu_- - V_i & \text{if } \alpha(i) = 0 \text{ and } V_i < -\mu_- + \varepsilon/2, \text{ and} \\ V_i + \mu_+ & \text{otherwise.} \end{cases} \tag{2.36}$$

Note that with the notation that was set up with (2.2), the dependency of V_i on α is implicit. The correctness of the above approach is motivated by an example in Figure 2.9, which was formed using the above. From this result we can now assess *directly* that the ground states in Figure 2.7 are all not 'very stable', since, for each of the states shown, only a tiny amount of potential of the right polarisation is needed at the respective *critical* SiDB to create a bottleneck for the population stability: this tiny amount is the distance between the minimal error associated with the critical SiDB and and 0.

Nonetheless, typically most of the $3^{|S|}$ states would not adhere to the population stability criterion, and thus have a *positive minimum stability*

error associated. However, as discussed before, these states could still be interesting if they could tip over into stability easily; i.e. their minimum stability error is small. Especially those with an associated system energy that is below the ground state's energy should stay relatively close to the ground state in our information structure.

2.4.2 Preordered Judgements

Having introduced the key components to our ordering of global states, i.e. attributed system energy and minimal stability error, we now define a preorder over $\mathscr{G}_S$ that defines a judgement structure $\mathfrak{J}_S \coloneqq \left(\mathscr{G}_S, \precsim\right)$, of which the primary objects are $\alpha \in \mathscr{G}_S$. Relevant to making a judgement of α, dependent on it we have the system energy E_α associated with it (see (2.6)), and simultaneously the minimum stability error $\mathfrak{e}_\alpha$ that was defined in (2.36). We now give the primary ordering structure, which simply orders elements by the minimum stability error:

$$\forall \alpha, \alpha' \in \mathscr{G}_S \,.\, \mathfrak{e}_\alpha \leq \mathfrak{e}_{\alpha'} \overset{\text{def}}{\iff} \alpha \precsim \alpha' \tag{2.37}$$

With this definition, the ground state is not necessarily all the way on the bottom. If it were, then that would indicate already that aside from being the metastable state with lowest system energy, it would also be the most stable state, as determined by $\mathfrak{e}_\alpha$. However, the system energy must attribute too, and this ties in with the parameters T^{max} and P^{min} we introduced.

According to Drewniok *et al.* in [9], we can calculate the occupation of a state in thermal equilibrium at a fixed temperature T. Let $\mathscr{M}_S \subseteq \mathscr{G}_S$ be the set of metastable global states; therefore $\forall \alpha \in \mathscr{M}_S \left(\mathfrak{e}_\alpha \leq 0\right)$, along with the property of *configuration stability* that we are not considering explicitly here. [24] For any $\alpha \in \mathscr{M}_S$, we can obtain the probability of this state being occupied by S at $T = T^{\mathsf{max}}$:

$$\frac{1}{\mathsf{prob}_\alpha} = e^{\frac{E_\alpha}{k_B T}} \cdot \sum_{\alpha' \in \mathscr{M}_S} e^{-\frac{E_{\alpha'}}{k_B T}}, \tag{2.38}$$

where k_B is the Boltzmann constant. Now, when we write $\mathsf{prob}_\alpha(\mathscr{S}_S)$ for some $\mathscr{S}_S \subseteq \mathscr{G}_S$ with $\alpha \in \mathscr{S}_S$, we mean to replace $\mathscr{M}_S$ in the above equation with $\mathscr{S}_S$. Indeed, the subset of global states to take into thermodynamic consideration assumes a dynamic role in the following.

In order to introduce refinement to our ordering, we define a neutral basis to enhance iteratively:

$$\mathfrak{J}_S^{\gamma_0} \coloneqq \left(\mathscr{G}_S \cup \{\top\}, \precsim_0\right) \tag{2.39}$$

with $\forall \alpha \in \mathfrak{J}_S^{\gamma_0} \left(\alpha \precsim_0 \alpha \wedge \alpha \precsim_0 \top\right)$ to trivially satisfy reflexivity and transitivity of the ordering (see [16]), and to introduce the relation to the top element $\top$.[17]

[17] In our bespoke notation, all set notation will be compatible with preordered sets. This means that the set notation targets the underlying set of the preorder.

We will use this element to unify that which we discard to be ordered above everything all the time, and may therefore have the semantic judgement $E_\top \coloneqq \infty$, $\mathfrak{e}_\top \coloneqq \infty$ attributed.[18] The destructive operation here will be the process of the functor $\mathcal{D} : \mathfrak{F}_S^{\gamma_n} \to \mathfrak{F}_S^{\gamma_{n+1}}$ that we now raise inductively with $\gamma_0 \coloneqq \{\top\}$ and $\gamma_n \subseteq \gamma_{n+1}$ and $\gamma_n \subseteq \mathfrak{F}_S^{\gamma_n}$ for all $n \in \mathbb{N}$:

$$\mathcal{D}\left(\alpha \in \mathfrak{F}_S^{\gamma_n}\right) \coloneqq \begin{cases} \alpha & \text{if } \alpha \neq \hat{\mathfrak{e}}_{\gamma_n} \text{ or } \mathsf{Relevant}_{\gamma_n}(\alpha), \text{ and} \\ \top & \text{otherwise,} \end{cases} \tag{2.40}$$

where we write $\hat{\mathfrak{e}}_\gamma$ for $\hat{\mathfrak{e}}_\gamma(\varnothing)$, which is defined by the following *maximal error rank* function:

$$\hat{\mathfrak{e}}_{\gamma_n}\left(\mathcal{S}_S \subseteq \mathfrak{F}_S^{\gamma_n}\right) \coloneqq \begin{cases} \operatorname{argmax}\left\{\mathfrak{e}_\alpha \;\middle|\; \alpha \in \mathfrak{F}_S^{\gamma_n} \setminus \left(\gamma_n \cup \mathcal{S}_S\right)\right\} & \text{if } \gamma_n \cup \mathcal{S}_S \subset \mathfrak{F}_S^{\gamma_n}, \\ \top & \text{otherwise.} \end{cases} \tag{2.41}$$

To complete the definition of (2.40), we present a predicate that may be used to show *irrelevancy* of some α when *un*satisfied:

$$\mathsf{Relevant}_{\gamma_n}(\alpha) \coloneqq \mathsf{prob}_\alpha\left\{\alpha' \in \mathfrak{F}_S^{\gamma_n} \;\middle|\; \alpha' \precsim_n \alpha \vee \alpha' = \hat{\mathfrak{e}}_{\gamma_n}(\{\alpha\})\right\} \geq P^{\mathsf{min}}. \tag{2.42}$$

Now, in order to satisfy $\forall n \in \mathbb{N}\left(\gamma_n \subseteq \mathfrak{F}_S^{\gamma_n} \wedge \gamma_n \subseteq \gamma_{n+1}\right)$, we define the γ update as follows:

$$\gamma_{n+1} \coloneqq \gamma_n \cup \left\{\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_n}\right)\right\}, \tag{2.43}$$

i.e.: by expanding this γ set that is used to obtain the state of maximal error rank, we move down the error rank with larger n, thus requiring each next state to have a smaller minimal stability error. If the currently considered state of maximal error rank (as determined by γ_n) is considered irrelevant, the unification with $\top$ essentially discards it such that we get $\mathfrak{F}_S^{\gamma_n} \supseteq \mathfrak{F}_S^{\gamma_{n+1}}$ for all $n \in \mathbb{N}$, which is shown by **Lemma 2** in Appendix A.2.

The role of γ here is to keep track of the states that we want to exclude in a search for the state of maximal error rank. This is done through accumulating maximal error rank states in γ that *are* found to be relevant. The states in γ that are not $\top$ are 'put aside' in a sense: we have looked at them and found them to be relevant in the context in which they were assessed. Whether they will be relevant when taken together with states with a much lower error rank is a different matter, which will be resolved later.

Furthermore, for $\mathcal{D}$ to be a *functor*, it must also operate on the ordering relation $\mathcal{D}\left(\precsim_n\right) \coloneqq \precsim_{n+1}$, which it thus defines *constructively*; for all $\alpha \in \mathfrak{F}_S^{\gamma_n}$:

$$\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_n}\right) = \hat{\mathfrak{e}}_{\gamma_n} \wedge \alpha \precsim_n \hat{\mathfrak{e}}_{\gamma_n} \overset{\text{def}}{\iff} \mathcal{D}(\alpha) \precsim_{n+1} \hat{\mathfrak{e}}_{\gamma_{n+1}}, \tag{2.44}$$

[18] Therefore, we say that for all $\mathcal{S}_S$, $\alpha \in \mathcal{S}_S \subseteq \mathcal{G}_S$, $\mathsf{prob}_\alpha\left(\mathcal{S}_S \cup \{\top\}\right) = \mathsf{prob}_\alpha\left(\mathcal{S}_S\right)$.

and for all $\alpha' \in \mathfrak{F}_S^{\gamma_n}$, $\alpha' \neq \hat{\mathbf{e}}_{\gamma_{n+1}}$ or $\alpha = \alpha'$:

$$\alpha \precsim_n \alpha' \overset{\text{def}}{\iff} \boldsymbol{\mathcal{D}}(\alpha) \precsim_{n+1} \boldsymbol{\mathcal{D}}\left(\alpha'\right). \tag{2.45}$$

Before elaborating on the process that is the iteration of $\boldsymbol{\mathcal{D}}$, we now state a *loop invariant* throughout this process. Let $\vec{\mathbf{e}}$ be the sequence of all $\alpha \in \mathscr{G}_S$ ordered by $\mathbf{e}_\alpha$. We will say that $\vec{\mathbf{e}}_0$ denotes the state of highest error rank, meaning $\vec{\mathbf{e}}_{|\mathscr{G}_S|-1}$ is the lowest in this regard. Moreover, for all $i \in \mathbb{N}$ with $i \geq \left|\mathscr{G}_S\right|$, we define $\vec{\mathbf{e}}_i \coloneqq \top$, and in order to enforce determinism, we make sure that for all $\alpha, \alpha' \in \mathscr{G}_S$, $\alpha \neq \alpha' \implies \mathbf{e}_\alpha \neq \mathbf{e}'_\alpha$. This is trivial to ensure in $\mathbb{R}$ for our finite application domain, and as a consequence, the $\vec{\mathbf{e}}$ ordering as well as the order of the successive argmax picks in (2.41) are fixed, which aids the proofs in Appendix A.2.

$$\begin{aligned} &\forall n \in \mathbb{N} \,.\, \top \in \gamma_n \wedge \forall \alpha \in \gamma_n \setminus \{\top\} \,.\, \exists i \in \mathbb{N} \,.\, i < n \wedge \alpha = \vec{\mathbf{e}}_i \wedge \\ &\mathsf{prob}_\alpha \left\{ \vec{\mathbf{e}}_{i'} \mid i' \in \mathbb{N}, i' = i+1 \vee \left(i' \leq i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n \right) \right\} \geq \boldsymbol{P}^{\mathsf{min}} \end{aligned} \tag{2.46}$$

In particular, this invariant tells us that for every state α that is collected in γ, we know that it was judged to be relevant in context with the state one error rank below, and further only the states that are in *consecutive* error ranks above α that are also collected in γ. In Appendix A.2, this invariant is proven.

Now observe what happens when we iterate $\boldsymbol{\mathcal{D}}$: we start at the state with maximal stability error required for it to be metastable. We examine the thermodynamic probability for this state to be occupied when this, $\hat{\mathbf{e}}_{\gamma_0}$, and the state that is next up for maximal stability error, $\hat{\mathbf{e}}_{\gamma_1}$, are considered together. Notice that other than the state next up, from now referred to as 'one rank down', the only state further taken into account are the states that $\precsim_0$ orders below α, which is just α by the definition below (2.39).

This introduces two cases: either (1) the energy difference between the two is so big that at $\boldsymbol{T}^{\mathsf{max}}$ the probability that $\hat{\mathbf{e}}_{\gamma_0}$ is occupied in favour of $\hat{\mathbf{e}}_{\gamma_1}$ is less than $\boldsymbol{P}^{\mathsf{min}}$, or (2) it equal or greater than it:

$< \mathbf{P}^{\mathsf{min}}$: This means that not only $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_1}} \leq \mathbf{e}_{\hat{\mathbf{e}}_{\gamma_0}}$, i.e., the error bounds need to be lower for the former to be population stable, but also $E_{\hat{\mathbf{e}}_{\gamma_1}} \lll E_{\hat{\mathbf{e}}_{\gamma_0}}$. Whether $<$, $\ll$, or $\lll$ would be appropriate to write here is directly determined by $0 < \boldsymbol{P}^{\mathsf{min}} < 1$ and $\boldsymbol{T}^{\mathsf{max}}$: a smaller probability at a higher temperature enforces a bigger gap. These judgements together tell us that, in relation to $\hat{\mathbf{e}}_{\gamma_1}$, $\hat{\mathbf{e}}_{\gamma_0}$ is *dwarfed* in relevance. This leads us to discard it in unification with $\top$, leading to $\mathfrak{F}_S^{\gamma_0} \supset \mathfrak{F}_S^{\gamma_1}$. The ordering $\precsim_0$ goes through no inherent change when transformed to $\precsim_1$.

$\geq \mathbf{P}^{\mathsf{min}}$ In this case, the gap is not large enough, as decided by $\boldsymbol{P}^{\mathsf{min}}$, to consider $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_0}}$ discarded. It could be that its associated system energy is

very low, making it a ground state candidate under the right conditions. Since $\hat{\mathbf{e}}_{\gamma_0}$ is the least metastable state, however, these conditions must be favouring a very improbable state, so this must be taken into account as we move forward—we will resolve this issue at the end of this section. When we now move forward, we have $\mathfrak{F}_S^{\gamma_0} = \mathfrak{F}_S^{\gamma_1}$, but our ordering relation updates: $\hat{\mathbf{e}}_{\gamma_0} \precsim_1 \hat{\mathbf{e}}_{\gamma_1}$, at least somewhat contrasting their minimal stability error ordering.

For the latter case something interesting happens to the procedure, since the change of our ordering changes the next iteration. In particular, when $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_2}}$ is now considered in a thermodynamic analysis, it is taken in relation to both $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_1}}$ and $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_0}}$. The branching occurs once more.

$< \mathbf{P}^{\min}$**:** We have found that $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_1}}$ dwarfed in relevance when taken with the two states with neighbouring minimal stability error. Then, with $\hat{\mathbf{e}}_{\gamma_1} \neq \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_1}\right) = \top$, this breaks a chain of states of neighbouring error rank in which each consecutive element was judged to be relevant with our occupation heuristic at $T^{\max}$. This chain thus provides a context of states that are further from metastability than the current one, yet each element in the chain was found to be energetically favoured enough when taken with the chain as context, plus the state one error rank down. Observe now that our ordering changes with $\hat{\mathbf{e}}_{\gamma_1}$ being discarded to $\top$: $\mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_0}\right) = \hat{\mathbf{e}}_{\gamma_0} \precsim_2 \top = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_1}\right)$, leaving $\hat{\mathbf{e}}_{\gamma_0}$ isolated in the first chain that was cut off—it will not be related to anything other than it or taken into account in further iteration of $\mathcal{D}$.

$\geq \mathbf{P}^{\min}$ This judgement tells us that the chance that $\hat{\mathbf{e}}_{\gamma_1}$ is occupied in favour of $\hat{\mathbf{e}}_{\gamma_0}$ and $\hat{\mathbf{e}}_{\gamma_2}$ is sufficient to consider this state relevant in this context where states are likely still far from being metastable. Notice that this heuristic is *local* in the sense that states are assessed within a range of states of similar error rank. Due to the variation in the $3^{|S|}$ states in terms of system energy primarily, we will never end up with a chain that ends up spanning the whole range, thus keeping the assessments local. The ordering now changes to $\hat{\mathbf{e}}_{\gamma_0} \precsim_2 \hat{\mathbf{e}}_{\gamma_1} \precsim_2 \hat{\mathbf{e}}_{\gamma_2}$, enabling the transitive property.

This process thus continues, creating such chains that connect contextually relevant states, where the context is a slice of the error ranking. When all states have either been discarded or put in γ, the iteration is said to have reached a *fixed point* at $n \geq \left|\mathcal{G}_S\right|$: neither of the sets γ and $\mathfrak{F}_S^{\gamma}$ will undergo a change from this point; we can thus regard this as the point at which we terminate the iteration. This leaves us with separated chains in which states of consecutive error rank are found, with the ordering defying their minimal stability error ordering for each pair of non-$\top$ states. This is exactly

what we will account for now, by defining an operation that will create the promised equivalence classes out of each chain that are themselves ordered by error rank, simply by the following. For all $\alpha \in \mathfrak{F}_S^{\gamma_{|\mathscr{G}_S|}}$: $\mathcal{D}(\alpha) \coloneqq \alpha$ and $\mathcal{D}\left(\precsim_{|\mathscr{G}_S|}\right) \coloneqq \precsim$ with also for all $\alpha' \in \mathfrak{F}_S^{\gamma_{|\mathscr{G}_S|}}$, $\alpha \precsim_{|\mathscr{G}_S|} \alpha' \vee \mathfrak{e}_\alpha \leq \mathfrak{e}_{\alpha'} \overset{\text{def}}{\iff} \alpha \precsim \alpha'$.

2.4.3 Towards Generalised State Space Ordering

In the previous, we looked at a way to connect information on population stability and system energy, which are the prominent features that decide occupation of state. It should noticed, however, that the method of doing this that was described in Section 2.4.2 is merely of an idea that extends to novel *direction* in research efforts on the simulation of SiDBs, moving towards *dynamic error simulation* and *bound-optimised logic design*, discussed in sections Section 2.5.3 and Section 2.5.4. Before examining these ideas further, let us consider some extensions to the methods presented in the previous section.

With the definition of Relevant in (2.42), the $\mathcal{D}$-iteration machine could be said to have a 'lookahead' of 1. This may naturally be adapted to work with any natural number, which impacts the rate at which states are discarded positively; as the number of states considered together increases, the occupation probability will spread out. Notice also that the extended range of states consecutively following up any current state in the error rank contains only states that are more population stable than it, making it only reasonable to take these states into account when considering the current one that is then a lower bound in this regard. Setting it to $\left|\mathscr{G}_S\right|$, we obtain a complete lookahead which the definitions support by [18], thus considering the complete set of states to which any current state forms a lower bound.

For another extension to the method, we may choose some real number err^{max} to shrink the error rank, discarding every state above the given limit. The lower bound will be found by some $i \in \mathbb{N}$ such that $\mathfrak{e}_{\vec{\mathfrak{e}}_{i+1}} \leq err^{\mathsf{max}} < \mathfrak{e}_{\vec{\mathfrak{e}}_i}$ and so we set $\precsim_{i+1} \coloneqq \precsim_0$ and $\mathfrak{F}_S^{\gamma_{i+1}} \coloneqq \mathfrak{F}_S^{\gamma_0} \setminus \left\{\vec{\mathfrak{e}}_j \mid 0 \leq j \leq i\right\}$, $\gamma_{i+1} \coloneqq \gamma_0$, to thus commence the procedure at $i+1$ if this non-$\vec{\mathfrak{e}}_0$ lower bound exists. Additionally, it is possible to incorporate a minimal error for the *configuration* stability criterion, which requires that no single charge hop event may lead to a lower system energy. A minimal error for this condition thus determines the accepted undershoot of the system energy resulting from a swap of two different charge states. Any state that exceeds this bound may then be filtered out beforehand.

To foreshadow what will be discussed in Section 2.5.4, we may use the method described to construct clusters that perform robust ground state logic as assessed by our judgements, with which we intend to maximise the gap in both stability error and energy between the ground state and the first erroneous state. Preceding this, we now enter the topic of SiDB simulation,

to which we have referred many times already because of its fundamental role in computational SiDB analysis; the major topic of this thesis, *and* essential to SiDB logic design. However, in the previous sections we proposed a lateral way to look at this very problem, by considering that we may assess states directly, thus enabling the creation of *robust* SiDB logic components *constructively*. The key insight here lies in seeing the logic components as compositions of smaller components that interact, which could be just a string of SiDB-bits for the case of a wire. Therefore this shifts our perspective from ordering *global* states, to ordering *state spaces* generically, of which the associated system is to be placed in a *context*, i.e., a layout with some holes—a more general version of a logic design canvas.

2.5 SiDB Simulation

In the field of SiDB research, and in particular in the study that moves towards enabling the platform for atom-scale logic implementation, computer-aided efforts in the field all rely on *physical simulation*, herein simply referred to as *simulation*, as the essential tool.[19] As input to an SiDB simulator, a layout is given with some DBs placed, to which local and global external potential can be applied that remains static throughout a simulation run. In addition, along with lattice-defining constants, there are three material-dependent parameters that may be configured to suit the properties of the material that hosts the DBs that are simulated, together constituting the *physical parameters*: ϵ_r and λ_{TF} determine the screened Coulomb potential for a given inter-DB distance (see (2.1)), and μ_-, which has received many mentions already in foregoing sections, determines the **(0/−)** charge transition threshold. It should be noted that static system temperature can also be taken into account as we have seen in Section 2.4.2, yet state-of-the-art SiDB simulation methods consider this environmental factor extraneous to the core functionality of a simulator, which discussed hereafter.

Depending on the type of simulator that is invoked, results are returned with the highest accuracy that the simulation model can offer, though at the cost of exponential runtime, or a polynomial-time heuristic, i.e. non-exhaustive search is performed in search for physically valid states with in particular a *proposed* ground state. The latter type is useful for its scalability, and may be used to, e.g. deny proposed ground states by finding a different state of lower system energy in the exponential search space, hence proposing a new state of lowest system energy. Furthermore, any physically valid layouts that can be found for a given layout contribute to the knowledge on a given configuration of SiDBs under given conditions, since analysis of the

[19] Other classes of simulators for the SiDB technology exist, with charge hopping simulation and electrostatic landscape solving in particular. [6, 24] When referring to simulators in this thesis, we never consider other classes than the one described in this section.

energy of excited states is relevant to temperature analysis. [9] Note then, that such analysis is performed by invocation of a simulator, whose results may then be used.

As hereby implied, the objective in SiDB simulation is to assess the exponentially many states that the physical model supports, with in particular deciding metastability and computing the system energy of any state that is chosen to be assessed. For heuristic simulation, the effort is typically directed in search of the ground state, used to implement logic as discussed in Section 2.4, since the observed logic behaviour in the ground state is the primary indicator to assess correctness of a proposed logic implementation. The main contributions in the field for this class of simulator are *SimAnneal*, scaling to virtually unlimited problem sizes for the objective of finding metastable states, due to its highly efficient stochastic method that mimics the physical annealing process that takes place, and *QuickSim*. [22] The latter simulator also incorporates random processes in order to cover interesting regions of the search space, yet its method involves search space pruning through ideas from statistical physics, in avoidance of the search getting stuck in a local energetic minimum—SiDB simulators are thus also referred to as *ground state finders*.

2.5.1 Exact Physical Simulation

The above concludes a review on heuristic simulation, since our focus in this field is on the exhaustive counterpart. While the former type may be used to *deny* ground states found by a heuristic simulation run, or even to prove a metastable state exists for a given input, it cannot be used to generate a *proof* that a state is the ground state, as this, per definition, requires exhaustive consideration of the search space that the physical model offers. In this process, *all* physically valid states are encountered, thus this type of *exact* physical simulation simultaneously finds all excited states along with it, which may thus be forwarded to a temperature analysis as mentioned previously.

The inherent lossless nature of exact simulation ensures that a single expensive simulation run returns the limit of the information that can be obtained for a given input. That means that, for the context of ground state finding, its *time to solution* (TTS) is equal to the runtime of this one invocation, while heuristic simulators need to account for a *confidence factor* that is determined through invoking the simulator multiple times, and having a reference ground state to use as objective. [24] For purposes of logic design on the SiDB platform, having to account for any confidence factor at all is very limiting, and thus the determinism of exact, single-run simulation is preferred in, for example, the creation of minimal logic gates. [10] In this effort, an exact simulator was applied to assess the logic-correctness of exhaustive placement of a minimal amount of SiDBs in a logic canvas.

This thus introduces a placement complexity of n *choose* k, where n is the number of lattice sites in the logic canvas where a DB could be placed, and k is the number of DBs to place. To assess each such placement, it would be impractical to rely on heuristic simulation, as that would defy the claim of a minimal logic gate obtained in this way to be truly minimal: exhaustive searches for logic implementations *require* exhaustive simulation.

However, the exponential run-time complexity of exhaustive state analysis presents a fierce opponent; even exact simulation of SiDB logic gates with over 20 DBs used to already thread the line of computational intractability, and that is excluding consideration of positive charges altogether. The first solution to the exhaustive problem has been *Exhaustive Ground State Simulation* (ExGS) which adopted a brute-force method, iterating through the possible states one by one to assess, primarily, their metastability. [24] This exact simulator proposed the first way of *verification* through such exhaustive consideration, albeit limited to fairly small problem sizes. Thus, efforts like minimal logic gate design were constrained under the *raw* exponential runtime complexity.

The emphasis on 'raw' is with a good reason: while, in the current state of the field, the problem of exact simulation puts forward a direct requirement for exponential runtime as the number of possible states grows exponentially with the number of DBs, we may apply search space pruning to drastically improve runtimes, though not escaping the complexity class. In particular, this involves a slight restatement of the problem of exact SiDB simulation: we want to find *all* metastable states. This directly enables search space pruning, through finding metastability 'blockages' that entail a whole region of the search space. Referring back to Section 2.2 where we presented intuitions for what constitutes a physically valid state, we had, e.g., **Rule 1** stating that any energetically isolated SiDB rests in SiDB**−** on our n-doped bulk surface, and **Rule 2a** stating that SiDB**+** may only occur in a DB-dense region. Thereby, any arbitrary global state that fails under the properly formulated inequalities associated with these respective conditions will thus defy it for one or more SiDB, and hence any other global state that shares the assignment to one of these SiDBs may be pruned along with it.

These intuitions empowered the creation of *QuickExact*, with the aforementioned presented along with it as the technique of *physically informed space pruning*. [11] Specifically, this involves the process of detecting SiDBs that *must be* SiDB**−**, as well as detecting SiDBs that *cannot* be SiDB**+**. With *QuickExact*, this computationally inexpensive analysis is performed to reduce the simulation search space by some exponential factors before the remaining space is iterated through efficiently, though in non-parallelised fashion. Indeed, as can be seen in Figure 2.10, anything found in this process shrinks the search space drastically, which thereby transfers the problem complexity from a measure on the input size, i.e., the number of SiDBs in the input layout, to a measure on input *density*: the sparser the input,

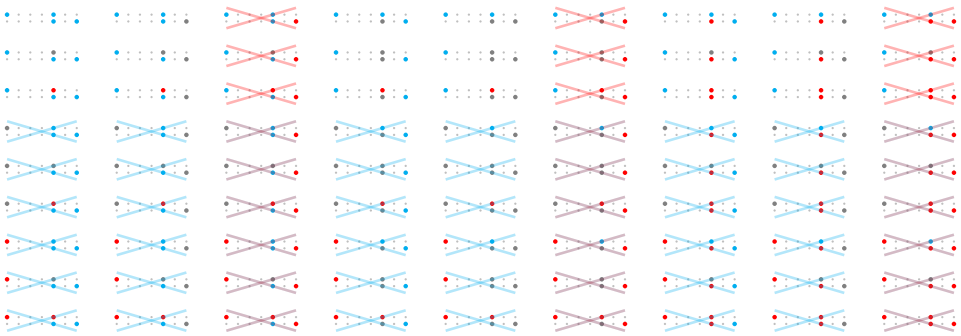


Figure 2.10: The global state space of a 4-DB layout with 63 of $3^4 = 81$ states pruned. The blue crosses correspond to the leftmost DB's non-accordance with **Rule 1**, and overlapping, the red crosses come from the rightmost DB going against **Rule 2a**. The physically performed space pruning shown here was performed using the parameters: $\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$.

the more SiDBs will not be able to attain SiDB+, or even SiDB−, and thus the smaller the remaining search space. This way of determining SiDB simulation problem complexity has not yet been considered in the literature thus far, but we will see in Section 2.5.2 that this is an appropriate way to gauge anticipated exact simulation runtimes, especially under the *complete* adoption of the physically informed space pruning concept—discussed after the following.

Before moving on to introduce the proposed addition to the field of exact SiDB simulation, we should first consider the impact of *QuickExact*. This simulator raised the bar to allow simulation of single logic gates some three orders of magnitude faster than *ExGS*, thus opening new doors for layout verification by making SiDB logic simulation scalable up to around 35-40 DBs. [11] In particular, when assessing the operational domain of a given logic gate, as touched on in Section 2.3.4 and Section 2.4.1, many simulation runs need to be performed to map operationality in the physical parameter space, which thus had to be done heuristically before *QuickExact* came to be. This shows that a raise of the bar of at most two dozen is already enough to make the exact method available to a new range of applications. For the case of *QuickExact*, many simulations of larger logic gates were unlocked.

The specific application taken as example in the previous paragraph does not suffer greatly from resorting to heuristic simulation, since an approximation of an operational domain can be considered sufficient information to judge a logic implementation on, especially for smaller layouts for which the heuristic simulators are more accurate. [8] However, the field of exact simulation is only recently coming to fruition, with the proposed intending to launch to an entirely new level as we will see next. Then also, in Section 2.5.4, we will discuss not *an* application of SiDB physical simulation, but,

proposedly, *the* application thereof. This statement will be clarified there, although it may already be said that the ideas presented there find their roots in exact simulation entirely, therewith, considering the weight of the latter statement, demonstrating the essentiality of delicate exact methods.

2.5.2 *ClusterComplete*

Having reviewed the state-of-the-art of SiDB simulation as well as the general objective of the *exact* flavour, the stage is set to present an overview of the proposed addition to the field that introduces a new dimension to layout verification and automated logic design: the proposed exact simulator is able to simulate *multiple* gates in connection, with a demonstration shown of a 2-input 2-output gate with all gate connections satiated, totalling to over 80 DBs. Moreover, whereas physical simulations are typically performed without consideration of positive charges to reduce the search space from base 3 to base 2 exponential, the proposed method operates unconstrained in this regard. For simulations of BDL layouts that were designed to operate only with SiDB**0** and SiDB**−** like the aforementioned demonstration, it even shows no increase in runtime for the higher base number: the *generalised* physically informed space pruning algorithm that gives rise to the proposed simulator emerges is able to eliminate the existence of positive charges in metastable charge configurations through iterative reasoning in a cluster hierarchy. This therefore presents a stark contrast to state-of-the-art simulators—especially heuristic ones—that do not offer base 3 simulation, or do offer it at a heavy cost of simulation accuracy and/or speed. Thereby, positive charges have been largely avoided since their behaviour, especially in larger context, could not be simulated well, while Figure 2.7 shows an instance of their use in SiDB logic, demonstrating that the third charge state presents a unique design component as a potential sink.

In this part of the thesis, we will give a high-level overview of the algorithm that extends and generalises the concept with which *QuickExact* raised the bar of tractability of exhaustive consideration of physically valid states. The idea will be presented here with regard to the application for which the method was developed, i.e. exact SiDB simulation, yet the exponential problem that is attacked with this is of a highly simple form when stripped down to the minimum. Thus, considering the effectiveness of the proposed as discussed in this section, it yearns for a generalised presentation of the proposed solution that may be instantiated in completely different fields of study. This will be the topic of Chapter 3, in which the SiDB domain will be left behind in favour of generality. In here, the reader interested in the SiDB application will be primed to recognise the different aspects in the generic setting of that chapter, so that the applied algorithm can be extrapolated autonomously.[20]

[20] In Section 4.1, which may be read after the generalised problem statement is given at

Higher-Order Physically Informed Space Pruning

We have mentioned that the proposed adopts a *generalised* version of pruning methods introduced with *QuickExact*, yet this header states a *higher-order* character too. This precisely originates from the algorithm's dependency on a given cluster hierarchy, without which the proposed could have never been more efficient than *QuickExact*. We will not go into formalities on cluster hierarchies as that will be the topic of Section 3.1, yet we have already lightly introduced the topic in Section 2.3.1, in which hierarchical charge spaces were introduced that lie fundamental to what is discussed here. Now 'higher-order' becomes a relevant term, since we consider a finite hierarchy of clusters, i.e.: clusters composed of clusters, charge spaces composed of charge spaces—all built from individual elements (the SiDBs) that are lifted to singleton cluster hierarchies to compose with others. In this line of thinking, the 'classical' physically informed space pruning quantifies over atomic objects, i.e. the non-decomposable SiDBs, thereby constituting first-order reasoning.

From the inherently recursive nature of the structure we are working with emerge recursive operations that shape the simulation algorithm in its *constructive*, and *destructive* facets: the construction of a minimal hierarchical charge space that forms a minimised upper bound in the sense discussed in Section 2.3.1, and through *dual* operation, an efficient dissection of this charge space produces the simulation result.[21] The constructive algorithm that is the former part, which, by the duality, thus shapes *both* sides of the simulation procedure is *Ground State Space*—their whole is $\mathcal{C}luster\mathcal{C}omplete$.

Characteristics of the former, which we abbreviate to *GSS*, have already had display in this thesis: throughout Section 2.3 we discussed charge spaces and frequently zoomed in on SiDBs in a pair while in particular separating the intra-pair electrostatic interaction from the interactions external to the pair. Just so, the ideas of *GSS* allow for *constructive reasoning* as we have seen in that section, where we, e.g., looked at temperature-robust BDL wire alignment in Section 2.3.3 by assuming bounds on the received potential from neighbouring pairs as in Section 2.3.2 and (2.15).

This reasoning becomes constructive when the obtained information of an instance, e.g. a reduced charge space of a BDL pair, is absorbed by the *active context*, e.g. the other BDL pairs: this information of charge space reduction may be implemented as an update of the received potential bounds for each other BDL pair.[22] Thus, the stepwise dynamic becomes a construction: we start with no information, assuming all-unpruned charge spaces, then

the start of Chapter 3, the instantiation for the problem of exact physical SiDB simulation is given.

[21] Recall that we defined an upper bound to be a (hierarchical) charge space from which all metastable global states may be extracted. As there could always be room for improvement, we will not state that the *least* upper bound is constructed.

[22] Intuitively, the *passive context* captures the static electrostatic interactions that are typically considered unidirectional for the scope of SiDB simulation.

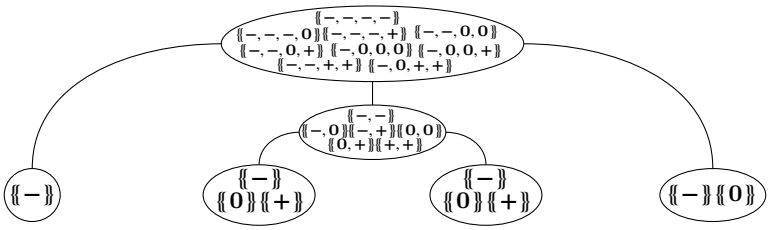


Figure 2.11: The hierarchical charge space of the 4-DB layout from Figure 2.10 is shown for which only the middle DBs are clustered. The charge spaces shown correspond to the state after physically informed space pruning was performed: the leftmost DB *must be* SiDB**−** and the rightmost DB *may not* be SiDB**+**. For each non-singleton cluster, its charge space is constructed through taking every combination of one charge space element from each direct child. Hence, while pruning is generally a destructive operation, here it becomes part of a constructive dynamic: only singletons are initialised with a complete charge space, while the rest is built up in this fashion. Thereby, pruning charge space elements low in the hierarchy has tremendous effect on the charge spaces above it; fewer combinations can be made, which thus carries through with increasing effect as we go further up the hierarchy. Note, however, that the charge space of the pair has not undergone pruning yet.

we gather information through making successive assessments, searching for witnesses to defy the validity of a charge space element—which then constitutes *information*. Such unspecific language becomes useful when we abstract over this process, allowing us to focus on the *flow* of this information while we take the construction to higher-order, now reasoning over clusters.

In much of Section 2.3 we simply considered pairs to keep it practical, sticking to a well-overseeable second-order view: the system had subsystems, i.e. the pairs, but we kept it quite limited from there. However, the multiset notation and notions of compositionality in Section 2.3.1 directly set us up for the now effortless generalisation to a finite-order cluster hierarchy. With that idea, we can regard the space pruning demonstrated by Figure 2.10 as the first round of a larger process that is the hierarchical construction, keeping it at only an analysis of singletons in this analogy. Moreover, the physically informed space pruning as introduced with *QuickExact* looked specifically for SiDBs that *must be* negative (**Rule 1**), and those that *may not* be positive (¬ **Rule 2a**). We generalise this here to the latter form of query entirely, such that the former information is constructed by two exclusions. These exclusions correspond to the proofs of ¬Valid predicates much like we have seen in Section 2.3.3 and Section 2.3.4. Indeed, there we *instantiated* the methodology of *Ground State Space*.

Since visual examples of hierarchical charge spaces grow very rapidly in size for less trivial layouts, we stick to a continuation of Figure 2.10 in

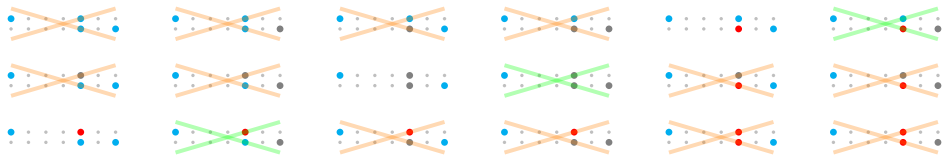


Figure 2.12: Going beyond the classical level of pruning shown in Figure 2.10, the charge spaces of the DBs in the middle pair are merged and checked. This results in the orange crosses, as only $\{\!\{\mathbf{0},\mathbf{0}\}\!\}$ and $\{\!\{-,+\}\!\}$ are accepted. Now the bound on the projected potential from the pair-cluster to the rightmost DB can be updated: we find that $\{\!\{+\}\!\}$ can be excluded for this singleton cluster, as indicated by the green crosses. The remaining states are the only metastable states for the parameters specified in Figure 2.10.

Figure 2.11. However, although the charge space of the *top cluster*—i.e., container of the whole system, and thus the node shown at the top—is presented in Figure 2.11, it would not occur like that at any point if *GSS* would be invoked on the layout in question. In particular, the charge space of the pair is not evaluated, since that is already beyond the first-order reach of the classical space pruning techniques applied in Figure 2.11. Instead, *GSS* looks at each of these charge space elements that are constructed through *merging* the charge spaces of the middle DBs in the way explained in Figure 2.11. Each charge space element of the pair-cluster is then checked by determining the bounds on the received potential at either DB given the charge space element itself, and the further electrostatic context from other DBs and other sources that are *all* considered static throughout such a checking procedure. The resulting information from the merge-and-check procedure is sent to the other DBs, resulting, in this case, in a *Ground State Space* from which only all metastable global states can be extracted, as seen in Figure 2.12.

Evaluation and Results

With this, we have sketched a very rough idea of the *Ground State Space* construction that predominantly determines the process of producing the exact simulation result: the exhaustive set of physically valid states. The algorithmic details become apparent in the more generic setting in the next chapter. Hence, we move on to the evaluation of the novel addition to the field, with in particular a discussion on the complexity *in application*, i.e., the characteristics of scaling for input layouts that implement SiDB logic.

Primarily, however, we look at the performance on unique layouts with completely arbitrary placement in a confined space. The latter aspect is of utmost importance here when it comes to algorithms that implement any form of physically informed space pruning, as discussed in Section 2.5.1; a billion-DB random layout is almost surely a trivial problem when confined to a square kilometre—or even a metre for that matter.

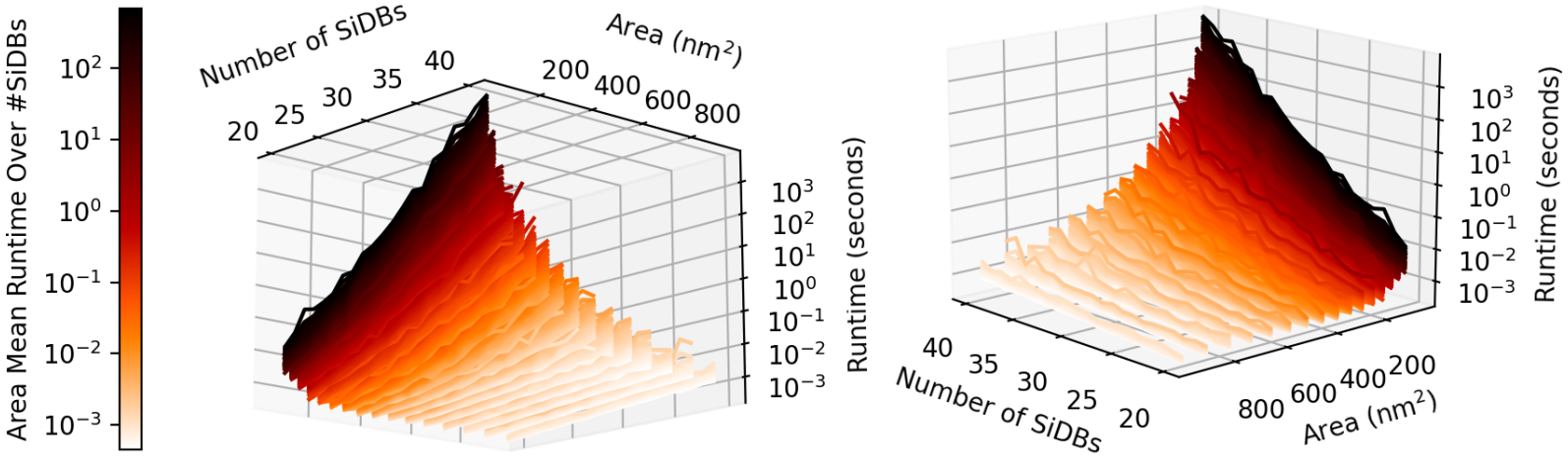


Figure 2.13: *ClusterComplete* simulation benchmark of 50 unique layouts of between 20 and 40 SiDBs placed arbitrarily in a square of different sizes. The same three-dimensional figure is shown from different angles. It is especially interesting to note that the variety between the runtimes for layouts with the same number of DBs and layout constraints shows that the proposed physical simulator is able to *consistently* take advantage of the layout's specifics. This opposes physically informed space pruning that performs just one pruning pass, resulting in a high variety depending on the layout: for some it may reduce the problem difficulty massively, while, for the same random layout parameters, it could leave a sublayout to be solved exhaustively, requiring several orders of magnitude more time to terminate. This is especially the case for base 3 simulation, which is the simulation base for the benchmark shown.

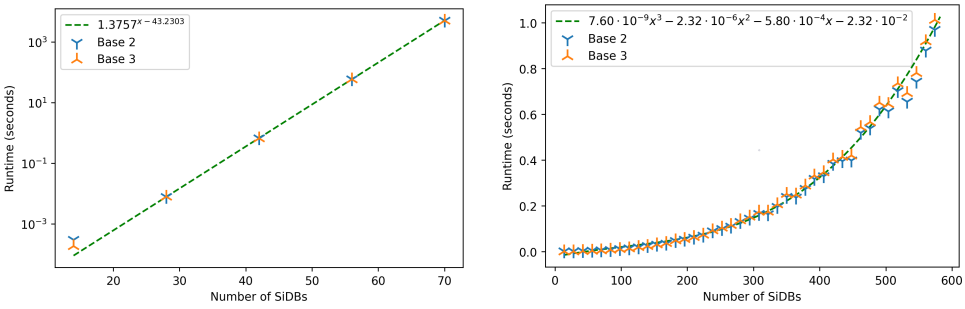
Figure 2.13 displays the scaling characteristic of *ClusterComplete* when invoked on random layouts. We may observe that the average *density* with respect to the various possible placements—i.e., the number of SiDBs / area—is directly proportional to the plotted runtimes. The implementation that was used for this benchmark uses a form of parallelisation in the destructive stage of the simulation process. However, due to the recursive nature of the implementation, the effect of the multithreading is limited. Recall that a disjunction emerges when decomposing a multiset as in Section 2.3.3, where we looked at a multiset charge configuration that had multiple compositions. This introduces branching in the destructive stage of the algorithm in which some branches may be much pruned earlier than others throughout the dual pruning process. The constructed *Ground State Space* is divided in separate sections of the top cluster charge space for each thread to unfold separately in search for metastable states, and thus some threads finish their work earlier than others. A single thread often ends up bottlenecking the runtimes, so by this imbalance, the runtimes are not scaled linearly with the number of available processor cores.[23]

Upon experimental evaluation of layouts implementing SiDB logic, it was found that area and the number of SiDBs are not the sole contributors to the complexity of the exact simulation problem, when *ClusterComplete* defines it. SiDB logic as implemented by BDL wires and logic gates from the *Bestagon* gate library poses little to no threat with regard to consideration of SiDB+, i.e. base 3 simulation, since the BDL layout style, with its typical inter-DB spacing, generally prevents this charge state from occurring in physically valid states. Also by its binary logic-enabling design, *Ground State Space* would easily find, e.g., $\{\!\{-,-,-,-\}\!\}$ to be omitted from the charge space of neighbouring BDL pairs that form a cluster. Thereby, the base 3 state space is greatly pruned, and the base 2 subspace also, enabling much larger exhaustive SiDB logic simulations than possible before.

However, taking the density as computed by the density in a rectangular *bounding box* of such layouts to give the estimate for the complexity fails, as depicted in Figure 2.14. Here, a *Bestagon* diagonal BDL wire was simulated for increasing sizes, which shows exponential growth in exact simulation time, while the bounding box density decreases. Rather, this example indicates a scaling with *effective density*, which would thus remain constant for increasing wire lengths. This would fit in Figure 2.13, where we see exponential scaling for any constant density, with the base of the exponential determined by it. The effective density would need to be computed with account for the physical parameters; it has not been investigated further for this thesis, yet it is suspected that the effective area that is used to compute the associated

[23] All benchmarks in this thesis were run on 8 cores of a 4th generation Intel i7 CPU.

[24] Curiously, it was found that $2^{x-\frac{x}{14}} \,/\, 1.38^x \approx 1.38^x$ for very high x even, where 14 is the number of DBs in a *Bestagon* wire tile.



(a) Exponentially scaling runtimes of one invocation of *ClusterComplete* with a single thread displayed on log-scale.

(b) *Ground State Space* runtimes averaged over a 1000 runs. This implementation does not support multi-threading.

Figure 2.14: Plots for an exact simulation complexity analysis of a BDL wire of increasing *Bestagon* wire segments, each consisting of 14 DBs. The x-axes show the number of SiDBs in the connected wire segments; the y-axes show runtimes in seconds. The slope in Figure 2.14a was roughly matched to 1.38^x, thus, for this case study, suggesting a complexity improvement in the order of approximately 1.45^x and 2.18^x over exhaustive consideration of the base 2 and base 3 state spaces respectively (like *ExGS* does). At the same time, this also directs to the improvement over *QuickExact* for the simulation parameters that were used ($\mu_- = -0.32\,\text{eV}$, $\epsilon_r = 5.6$, $\lambda_{TF} = 5\,\text{nm}$): the SiDB placement in these BDL wires produces only an amount of negative SiDBs per wire segment as detected by *QuickExact* that is at most constant (rapidly declining to 1 for these parameters), and thus the effect of the pruning is weakly linear.[24] In Figure 2.14b, the runtimes of the construction phase for the diagonal wire problem are approximated by a polynomial of low degree.

density may be obtained by first deriving a certain radius from the physical parameters, then using this radius from each SiDB to cover a section of the layout which then determines the effective area. The former step, finding a map from the parameters to this radius, remains the challenge here.

For exact simulation of combinational logic, the 2-input 2-output *Bestagon* crossover (CX) gate was chosen as its logic design complexity is highest of all the gates, as it is the only 2-output gate for which positive charges cannot be trivially closed out, albeit one SiDB short of the double wire gate. Rather than simulating the CX gate singled out as done in recent publications, we extend the layout to simulate with trivial gate connections on all interfaces. [8, 11] These are simply four straight wire gates connecting to the gate that serve as input and output ports on the *gate-level layout* within the open-source framework *fiction* that is specifically designed for FCN. [35] With *ClusterComplete*, the movement is set for exact simulation of gate-level designed layouts.

Initially, the simulation was carried out on the 'raw' layout, i.e., without

input and output perturbers applied. This saves 4 DBs in our case, but no gate operation can be expected.[25] The base 2 and base 3 simulation both terminated approximately in the same time for the 79-DB layout, a little over 4 and a half hours, both finding 1935 metastable states. Contrary to the individual simulations for Figure 2.14, which always gave the same top cluster charge space size for the different simulation bases, here we have a difference of 24 versus 68 states, confirming that SiDB+-containing states can indeed not be pruned trivially for the extended CX gate.

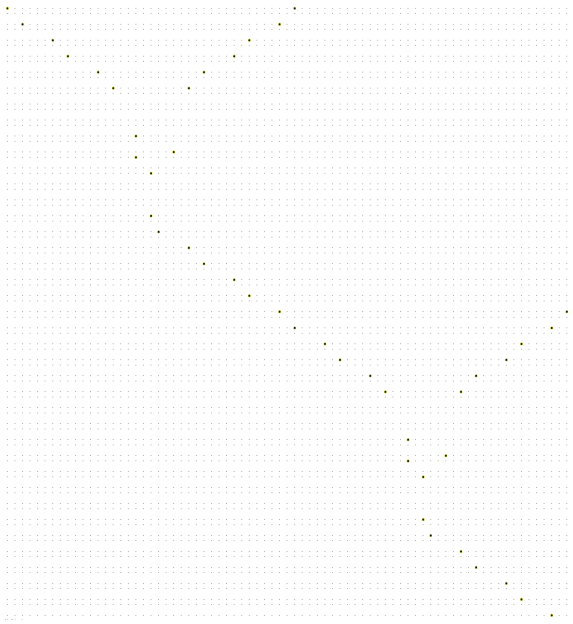
When input and output perturbers were applied, totalling the extended CX layout to 83 DBs, the physical simulation took much longer to terminate. This is in line with the expectation for exact SiDB logic simulation scaling that was set with Figure 2.14, where exponential growth was seen for constant effective density. With the current step from 79 to 83 DBs, the effective density was kept approximately the same by the inherent placement of input and output perturbers; the BDL wire placements patterns are respected, thus the perturbers extend the layout regularly. In Appendix B, the simulation results for the performed base 3 *ClusterComplete* simulations are shown.

As a final display of the newly available SiDB physical simulation force, the operational domain was computed *exactly* for another SiDB logic layout, though approximately half the size of the previous. At 42 DBs for each input combination, computing the operational domain for a layout of this size would have previously been infeasible. While simulating the ground state for this layout may be feasible with *QuickExact* depending on the time allotted, its scaling becomes prohibitive for a full operational domain calculation as shown in Figure 2.15; a great quantity of simulator calls are necessary, as pointed out in the figure's description. In contrast, one such call to *ClusterComplete* will take about a minute to terminate, thus enabling affordable larger scale operational domain calculation.

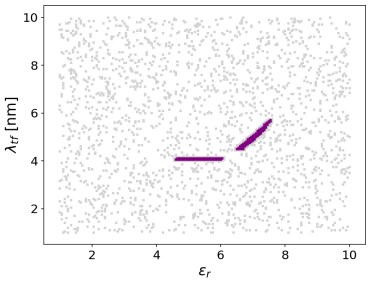
2.5.3 Simulation with Uncertainty Maps

When considering a *layout* to simulate, we typically consider a collection of SiDBs that, like a logic gate, would be placed in a larger context. To attribute any value to a simulation result, we commonly assume that the bounding box that encloses the layout would not contain more or less SiDBs—other than I/O perturbers, perhaps—when the layout is placed in a context, as the simulation result is critically dependent on any SiDBs present, *especially* when the effective density, as introduced in the previous section, would be

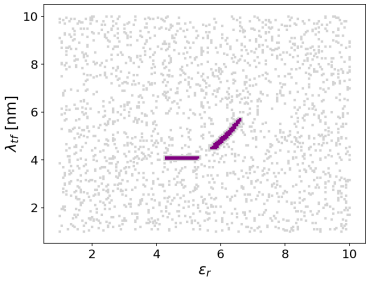
[25] It should be noted, also, that the *Bestagon* gate library, according to one of its authors who supervises this thesis, although designed for the same physical parameters as used here, was not tested with gate connections, and thus does not guarantee that gate operation is preserved for these more challenging contexts. The solution to this logic design challenge that previously approached with reinforcement learning and heuristic simulation is proposed in theory in Section 2.5.4.



(a) Two *Bestagon* OR gates in sequence.



(b) Op.dom. for $\mu_- = -0.28$



(c) Op.dom. for $\mu_- = -0.32$

Figure 2.15: The operational domain of an SiDB logic layout with 42 SiDBs for each input state. For each point in the parameter spaces shown, up to 2^3 input combinations were assessed with individual *ClusterComplete* simulations; the layout is operational precisely when a disjunction over the three inputs is computed in the ground state for each of the inputs. Rather than a *grid search*, the operational domain was computed using *flood fill*, which avoids mapping the entire non-operational space. [33]

shifted by the change, thus affecting the SiDB interaction statics to some significance. Thus, we may conclude for a fixed layout for which we have simulation results, that there is a relation between the difference in effective density within the layout's bounding box when the layout is placed in a context, and the value of the simulation results for the layout in isolation.

Inspecting this relation more carefully, we may observe that a measure of the change in effective density as only considered within the layout's bounding box does not capture contextual perturbations of, for instance, connecting wires or nearby gates. However, considering these sources to be outside the bounding box, we do obtain an entry point to distinguishing the *significance* of a simulation result *locally*: that is, we may be more certain about our judgement of metastability when concerning with sublayouts that are centred with respect to the bigger layout's bounding box. By the exponential interaction decay (see (2.1)), we find that, intuitively, a context has a greater effect on the extremities of a layout than a collection of SiDBs that are contained in the centre. Naturally, as is to be expected of SiDB

logic gate operation, the extremities will have an appropriate effect on the centred sublayout, which could be taken to be (a part of) the logic canvas. Yet, beyond concerning just with the ground state, the principal information that is obtained with an exhaustive ground state search is the set of *all* metastable configurations for the (sub)layout; the part of this information that specifies the metastable states of the sublayout under the presence of the SiDBs placed in the layout that are not in the sublayout, may thus be attributed a higher significance.

This is especially the case for the regularly placed BDL implementations that are configured with the intention of maintaining a *balanced* electrostatic landscape by sharing negative charges. Through these design characteristics, it will be unlikely that a context changes the metastability behaviour of the sublayout, i.e., the set of all physically valid states of the sublayout; the extremities, which could be the wires, act as a *buffer* to the electrostatic landscape as perceived by the sublayout. This buffer, that is created by configuring the extremities such that electrostatically active charges are shared locally, should be sufficiently strong to resist crossing a certain context-influence threshold that would impact the electrostatic landscape within the layout majorly. Thereby, the metastability results for the sublayout are consistent with respect to contextual influence, up to the resilience of the electrostatic potential buffers in the layout's extremities, along with the dimensions of the perturbance vacuum surrounding the sublayout.

Dynamic Error Simulation

In order to introduce consideration of significance to the physical SiDB simulation process, we propose to proceed the field to simulating with *dynamic* metastability error. For this approach, the usual ≈ 0 metastability error is used for what may be considered normal significance, as a reference, while higher error values may be used for a layout's extremities, for which we previously concluded that the typical simulation approach would yield results of low significance. Note that, while the notion of significance fits naturally with the intention of exact simulation, heuristic simulation may adopt the dynamic error simulation just the same; either kind of simulator may operate using an *uncertainty map*. Such a mapping from positions on the layout surface to an error value that would be used to test the metastability criteria may be derived from a collection of possible contexts—think of a gate library that defines a number of possible adjacent *tiles*—or, additionally, a *defect map* that stores charged defects in particular, i.e., sources of electrostatic potential associated with a surface defect type with charged features. [7, 23]

Through such dynamic consideration of metastability, layouts may be judged from a single simulator invocation directly, as discussed in Section 2.4.1. Operational domain mapping, which does this indirectly, may offer additional ranges of judgement by considering dynamism in the physical

parameters that would simultaneously parametrise the uncertainty map creation. [33] While such uncertainty maps would, in the proposed application, simply map to positive error offsets only, we may consider the negative kind too, and when it would be useful. 'Negative error' may sound unintuitive, coming from certain fields of study, though in the application of *robust* logic design, it becomes an asset: states that are metastable under negative metastability error have, so to speak, a *local buffer* before their stability is affected. Naturally, components that *operate*, i.e., by ground state logic, would preferably do so under a local buffer for electrostatic potential; this, along with a thermodynamic consideration as Section 2.4.2, makes the component robust, *definitionally*. This is discussed next, in which we merge simulation of SiDBs with logic design.

2.5.4 Bound-Optimised Logic Design

In the previous discussion, we left off with a notion of operationality of a component, along with definitional robustness assessments. Notice that a 'component' that operates, may really be anything that fulfils the job, and thus we may think of *logic computing clusters*. The idea of *bound-optimised logic design* is to produce definitionally robust logic components, i.e., **BOLD** gates. Rather than using SiDB simulators in order to assess creations that ought to compute logic in the ground state, this proposed direction of development shoots for the end goal directly, spawning *automation* of SiDB logic design that previously included many manual tasks, with in particular the creation of a *skeleton*, which includes the logic canvas that determines the search space for logic designs. [10] Such skeletons are made such that they fit a tile of some geometry that may be connected to other tiles, as seen in Figure 2.15 and Appendix B, which show applications of the hexagonal gate library proposed in [34]. Depending on the chosen tile size, geometry, and lattice orientation (e.g., (100)-2×1, (111)-1×1), there are numerous possibilities for such skeletons alone, that, by empirical experience with manual gate operation assessing, determine the logic consistency of a synthesised layout predominantly. Thereby, the generalisation to logic computing clusters presents the way out of the chains of manual configuration for gate library design.

The optimisation heuristic that is introduced will enable the process of finding optimal logic computing structures to be automated fully, stretching beyond the limitations of human imagination and manual effort by considering the *entire* bounded search space that contains a structure that is optimal for a given task. **BOLD** is a combination of the idea of definitional robustness through local resilience discussed in Section 2.5.3, which creates a heuristic to optimise that could also incorporate temperature considerations from Section 2.4.2, with methods of electrostatic potential bound analysis in a cluster hierarchy that gave rise to *ClusterComplete*. They unify into

something that is neither an SiDB simulator, nor a metric to assess the quality of a logic implementation on the platform: instead, rather than having these two components as separate stages requiring different tools to achieve the goal of creating rigid SiDB logic structures, **BOLD** aims for this objective *directly*. Thereby, this omits the need for a stand-alone simulator of which the use is embedded in manual configuration efforts, while ultimately such a simulation tool finds a—or perhaps even *the*—predominant use in a design flow towards achieving the aforementioned goal. Thus, the future of machine-synthesised logic could yield surprising results, taking maximal advantage of the immense creative freedom for generating logic structures that the SiDB platform offers—separating the technology from the constraints on human imagination that stem from decades of CMOS dependency.

BOLD is a future continuation of the ideas that founded *ClusterComplete*, advancing SiDB logic through omitting a human-operated simulator interface.

2.6 Notational Conventions and Lattice Theory

A selection of mathematical utility that is represented through notation in this thesis may not be known to some readers, and, by the finite nature of legible symbols, non-unique definitions are imminent. Thereby, this section aims to clear the path towards Chapter 3, such that no confusions may arise. Consequently, the format of this section likens to a glossary of sorts.

Functions and Domains

In this thesis, functions are commonly assumed to be defined on all inputs, i.e.: *totally defined*. We then say that such an $f : A \to B$ is *total*, for arbitrary sets A and B. Function types may also be given as *exponentials* $f : B^A$. When considering some $f : B^A$ that is *partial*, we implicitly assume some unique element of B to represent the undefined value, typically written as $\bot_B$. For notational convenience, and in order to direct attention elsewhere in a definition, we will introduce sets without specifying this special element to be contained explicitly; $\bot_B$ will not be counted with $|B|$ as its semantics are separated from B entirely. Since we assumed $\bot_B$ to be unique, we may later on imply that this element is also present in B.

In particular, also, partiality allows us to order *functions* of a related type:

$$\forall f : B^C, g : B^A \,.\, f \sqsubseteq g \overset{\text{def}}{\iff} \operatorname{dom}(f) \subseteq \operatorname{dom}(g) \land \forall a \in \operatorname{dom}(f) \,.\, f(a) = g(a), \tag{2.47}$$

where $C \subseteq A$ and $\operatorname{dom}(f) \coloneqq \left\{a \in A \mid f(a) \neq \bot_B\right\}$. Furthermore, for such a $C \subseteq A$, the domain of $f : B^A$ may be *restricted* through $f\big|_C : B^C$, such that

$f\big|_C \coloneqq \bigsqcup \{g \in B^C \mid g \sqsubseteq f\}$. We may instantiate an anonymous function that may be partial, defined on $k \in \mathbb{N}$ inputs, with the notation:

$$f\left[a_1 \mapsto b_1, \ldots, a_k \mapsto b_k\right] \coloneqq \bigsqcup \left\{g \in B^A \mid f\big|_C \sqsubseteq g,\ \forall 1 \leq i \leq k \,.\, g\left(a_i\right) = b_i\right\}, \tag{2.48}$$

where $a_i \in A$ and $b_i \in B$ for all $1 \leq i \leq k$, and $C = A \setminus \left\{a_i \mid 1 \leq i \leq k\right\} \subseteq A$. Thus, f is *overloaded* in a sense. When no function is prefixed before the bracket notation, $\bot_{B^A}$ should be assumed; this is the case in all of Chapter 3.

If $X \subseteq B^A$, then *least upper bound* of X, i.e., $\bigsqcup X \in B^A$, has the following properties: it is an *upper bound* of X by $\forall x \in X \left(x \sqsubseteq \bigsqcup X\right)$, and it is the *least* such: $\forall f \in B^A \left((\forall x \in X \,.\, x \sqsubseteq f) \implies \bigsqcup X \sqsubseteq f\right)$. The *greatest lower bound* is defined dually. For the scope of this thesis, we will be able to translate these operations on sets to respective binary *join* and *meet* operations that that are written with infix notation, e.g.: $f \sqcup g \coloneqq \bigsqcup \{f, g\}$. Then, $[a \mapsto b] \sqcup \left[a' \mapsto b'\right] = \left[a \mapsto b, a' \mapsto b'\right]$ and $[a \mapsto b] \sqcap \left[a' \mapsto b'\right] = \bot_{B^A}$, where $a, a' \in A$ with $a \neq a'$, $b, b' \in B$, and $\forall a \in A \left(\bot_{B^A}(a) = \bot_B\right)$.

Notions in the above that do not rely on underlying set theory may extend in compatibility to *morphims*, which generalise the notion of a function. Any *endomorphism*, i.e., some $f : A \to A$, may be iterated $n \in \mathbb{N}$ times on some $a \in A$, written as $f^n(a)$, where $f^0(a) = a$, $f^2(a) = f(f(a))$, etc. Moreover, a is a *fixed point* of f when $f(a) = a$.

Functors and Predicates

One functor with an important role in this chapter that was taken implicitly is the *multiset functor* $\mathcal{M}$, such that, for any set A, $\mathcal{M}(A)$ is the set of multisets on A such that for $m \in \mathcal{M}(A)$, $m(a)$ denotes the number of occurrences of $a \in A$ in m. Thus, we may $a \in m$ and $a \in \operatorname{dom}(m)$ interchangeably. Furthermore, we will use summation, i.e., *concatenation* of multisets, which may be interpreted as a pointwise summation of the occurrence counts. We sometimes use functors, in particular the powerset functor $\mathcal{P}$, on elements $a \in A$ that are not sets. This lifts the element to the right type in an isomorphic way, simply creating a singleton. Other properties of functors are not important for us in Chapter 3, though an in-depth analysis of functor properties is performed in Appendix A.1.

A *predicate* P over some set A may be seen as a subset of A for which the predicate holds, and thus $P \in \mathcal{P}(A)$. We may test any $a \in A$ in two equivalent ways of writing: $P(a)$, or $a \in P$, both yielding a truth value. We occasionally use the notation $a \sim A$, which should be read as 'a take from A', where $a \in A$. When we have $f : A \to \mathcal{P}(B)$, then $a \sim f$ denotes some $a \in A$ for which $f(a) \neq \varnothing$, i.e.: a applied to f does not yield the trivially false predicate.

Lattices and Groups

The order theory that has made much appearance in this section already with definition of $\sqsubseteq$ in (2.48), but also previously in Section 2.4.2 where we considered partial orders (see [16]), is foundational to *lattice theory*.[26] A lattice is a structure relative to some partial order relation, say $\leq$, though contrary to a partially ordered set, a lattice is required to have unique top $\top$ and bottom $\bot$ elements that bound the underlying set in respective ways, whereby all binary joins and meets exist such that $\top$ and $\bot$ are their respective identities. Thus, a lattice defines all finite joins and meets. The earlier definitions of least upper bounds and greatest lower bounds of function domains extend to lattice theoretic notions. In a *complete* lattice, joins and meets exist for *all* subsets of the underlying set. The powerset lattice $(\mathcal{P}(A), \subseteq)$ is an example of a complete lattice.

In Chapter 3, a *lattice-ordered group* will play a fundamental role. A *group* is a tuple $\langle A, \oplus, {}^{-1}, 0 \rangle$ such that A is a set, $\oplus : A \times A \to A$ and ${}^{-1} : A \to A$ are defined on all (pairs of) elements in A, and $0 \in A$. Furthermore, we have the property of *associativity*, i.e., $\forall a, a', a'' \in A \left(a \oplus \left(a' \oplus a''\right) = \left(a \oplus a'\right) \oplus a''\right)$, and we have some properties with regard to the group inversion operation and the identity element 0: $\forall a \in A \left(a \oplus a^{-1} = 0 = a^{-1} \oplus a \wedge 0 \oplus a = a = a \oplus 0\right)$. As a general fact, it follows that $\forall a, a' \in A \left(\left(a \oplus a'\right)^{-1} = a'^{-1} \oplus a^{-1}\right)$. The property of *commutativity* may hold, making it an *abelian* group, requiring that $\forall a, a' \in A \left(a \oplus a' = a' \oplus a\right)$. While common binary operations like $+$ and $\cdot$ defined on number sets are commutative, matrix multiplication puts forward an instance of a group that is not commutative. Any group is also a *semigroup*, also called *monoid*, which omits the inverse operation.

Combining the two above notions, a lattice-ordered group is a pair $\left(\left\langle A, \oplus, {}^{-1}, 0, \vee, \wedge \right\rangle, \leq\right)$ such that $\langle A, \vee, \wedge \rangle$ is a lattice and $\left\langle A, \oplus, {}^{-1}, 0 \right\rangle$ is a group, such that $\forall a, a', x, y \in A \left(a \leq a' \implies x \oplus a \oplus y \leq x \oplus a' \oplus y\right)$. With that, we get a number of inferred properties and definitions, with in particular $|a| \coloneqq a \vee a^{-1}$, i.e., the *magnitude* or *absolute value*. Additionally, the inverse operation reverses the order: $\forall a, a' \in A \left(a \leq a' \iff a^{-1} \geq a'^{-1}\right)$. Unfortunately, the standard notation for binary logical operations (which also form a Boolean lattice) clash with the symbols we use for the group joins and meets. The context in which the operations are used will ensure that there may not be any confusion; any element of the group to which the joins and meets relative to $\leq$ apply will be introduced with its type. An example of a commutative lattice ordered group is $(\langle \mathbb{R}, +, -, 0, \max, \min \rangle, \leq)$, where $\leq$ is just the regular ordering on (real) numbers. When reading Chapter 3 for discovering how *ClusterComplete* was created, this structure should be kept in mind for exact SiDB simulation instantiation of the generic formulation, presented next, and instantiated in Section 4.1.

[26] Order theoretic lattices should not be confused with crystallographic lattice models.

Chapter 3

State Space Pruning in a Cluster Hierarchy

Many systems can be captured as a graph in which the weighted edges describe interactions. In this work, a method is described to efficiently compute a subset of interest of a set that grows exponentially in the number of vertices, where the selected elements are state assignments to the vertices for which, for each vertex, the local interaction value for this state as evaluated by the vertex satisfies a respective requirement. Here, our values will live in a lattice-ordered abelian group $\langle W, \oplus, {}^{-1}, 0, \vee, \wedge \rangle$ relative to the partial order relation $\leq$, such that the ordering attributed to the interaction values interfaces semantic information that defines a means to *join* and *meet* arbitrary subsets of W.[1]

Instead of talking about vertices, however, we define $\mathcal{V}$ to be the finite set of variables of interest. The finite set of states is denoted $\mathcal{Z}$ such that the exhaustive set of which a particular selection is to be computed is the set of all state assignments $\mathcal{G}_{\mathcal{V}} \coloneqq \mathcal{Z}^{\mathcal{V}}$, which thus grows exponentially in $|\mathcal{V}|$ with $|\mathcal{Z}|$ as the base. For any $v \in \mathcal{V}$, we have a *state acceptance* map $\tau_v : \mathcal{Z} \to \mathcal{P}(W)$ that becomes a predicate for any $z \in \mathcal{Z}$ applied. Further, the evaluation function $\psi_v : \mathcal{G}_{\mathcal{V}} \to W$ maps a state assignment to a local interaction value such that, for any $\alpha \in \mathcal{G}_{\mathcal{V}}$, $\psi_v(\alpha) \in \tau_v(\alpha(v))$ becomes the requirement for v accepting α. Hence, we may write the set that we wish to compute as the following:

$$\mathsf{ValidAssignments} \leftarrow \left\{ \alpha \in \mathcal{G}_{\mathcal{V}} \mid \forall v \in \mathcal{V}.\, \tau_v(\alpha(v))\left(\psi_v(\alpha)\right) \right\}. \tag{3.1}$$

[1] The commutative property is required such that, together with the inverse operation, we can write, for all $u \subseteq W$, $w \in u$: $\left(\bigoplus u\right) \oplus w^{-1} = \bigoplus (u \setminus \{w\})$. This property is useful in the proposed solution when the inverse operation is sufficiently efficient to compute such that the dynamic operation on the aggregate is preferred over re-aggregation. When this does not apply, the proposed remains applicable under re-aggregations as described above. If the given lattice-ordered monoid structure is *dually residuated*, then magnitudes as used in (3.2) may be determined. [15]

Note here that α need not be total; behaviour of τ_v can be defined on the element $\bot_{\mathcal{Z}}$ that represents the undefined state. Then, for any *disjoint* $\alpha_1, \alpha_2 \in \mathcal{G}_{\mathcal{V}}$, i.e., $\alpha_1 \sqcap \alpha_2 = \bot_{\mathcal{G}_{\mathcal{V}}}$, for which $\alpha = \alpha_1 \sqcup \alpha_2$, we require that the *partial evaluations* may be combined as follows: $\psi_v(\alpha_1) \oplus \psi_v(\alpha_2) = \psi_v(\alpha)$. This property allows us to compute $\psi_v(\alpha)$ incrementally, and in particular it presents the opportunity to compute the aggregate ψ_v in a sectioned manner. Our approach to compute ValidAssignments relies on a cluster hierarchy of the variables that determines this sectioning for any instance in which ψ_v is to be evaluated, with one section corresponding to interactions local to v from within the cluster, while the remainder captures those external to it.

The proposed exploits this choice of aggregation order along with an assumed variation in the interaction weights, such that ψ_v may be determined by a few interactions predominantly, as determined by the lattice structure, where it becomes interesting if the *bounds* on these interactions, as determined by different state assignments, vary. The *interaction span* thus gives the heuristic for the creation of the cluster hierarchy. More specifically, in order to create a cluster hierarchy from $\mathcal{V}$ we require a distance metric that can either be given directly, or derived as follows for all $v' \in \mathcal{V}$, $v' \neq v$:

$$\frac{1}{d_{v,v'}} \coloneqq f \sum_{(v_1,v_2)\in\{(v,v'),(v',v)\}} \left| \mathsf{UB}_{v_1,v_2} \oplus {\mathsf{LB}_{v_1,v_2}}^{-1} \right|, \tag{3.2}$$

where

$$\mathsf{UB}_{v_1,v_2} \coloneqq \bigvee_{z\sim\tau_{v_1}} \bigvee_{z'\sim\tau_{v_2}} \psi_{v_1}\left(\left[\begin{array}{l} v_1 \mapsto z, \\ v_2 \mapsto z' \end{array}\right]\right) \text{(and dually)}, \tag{3.3}$$

and $f : W \to \mathbb{R}$ is an arbitrary monotone function with respect to $\leq$ on $\mathbb{R}$. A directly given distance metric must be compatible with the above, i.e., an appropriate morphisms must exist that proves the monotone inverse correspondence between the bidirectionally accumulated interaction spans and a given distance. If (3.2) yields 0, then the resulting distance is ∞.

Such a distance metric facilitates the creation of a cluster hierarchy through an agglomerative procedure in which clusters are formed with minimised inter-variable distances within each cluster; these variables may thus interact with a high by (3.2). Hereby, the resulting clusters allow us to consider interactions of higher variation separately from those of lower variation. In particular, by flattening possibly many interactions of lower variation to bounds on the local interaction value at some variable, the herein proposed dynamic that builds on this idea offers an improvement over computing ValidAssignments through exhaustive consideration of $\mathcal{G}_{\mathcal{V}}$, reducing the complexity of this operation by shrinking the base of the exponential that is bounded by $|\mathcal{Z}|$. The extent of this improvement depends on all of aspects of given problem since the procedure exploits the particulars to gain the advantage, as demonstrated in Section 2.5.2.

In the following, we first form a basis of notation with regard to cluster hierarchies, clusterings, and morphisms between clusterings in Section 3.1, such that the dual phases of *construction* and *destruction* may be presented in Section 3.2 and Section 3.3 respectively, where the latter phase unfolds the structure that is formed in the former phase to ValidAssignments. Both phases find their efficiency through continued pruning opportunities that emerge selectively, as guided by the given cluster hierarchy that highlights relations of major interest. Thereby, a minimised number of pruning tests of maximised efficacy dynamically operate on the exponential search space determined by $\mathscr{G}_{\mathcal{V}}$ to develop the exhaustive selection of interest, raising the threshold of computational intractability with respect to $|\mathcal{V}|$ for many problem instances that align with our description.

3.1 Clusterings and Morphisms

With (3.2), we determined the relation that is taken as the heuristic throughout the process of forming a cluster hierarchy. The details of this procedure involve decisions that depend on the metric space, with in particular the way that distances between clusters or even cluster hierarchies are derived in the metric. Depending on the problem instance, the performance of the proposed solution to computing ValidAssignments may vary with different methods adopted, and thus studies could be performed for an instance in search of an optimally synergetic approach for building a cluster hierarchy. However, by the dependency on the instance, this issue becomes non-focal to this thesis, so we may regard a cluster hierarchy of all elements in $\mathcal{V}$ as given.

3.1.1 A Category of Clusterings

Let us define precisely what it means for something to be a *clustering*. Intuitively, a clustering of $\mathcal{V}$ must be a set of non-overlapping nodes which together contain all elements of $\mathcal{V}$, representing a slice in the cluster hierarchy of $\mathcal{V}$. Formally, we define an $\mathcal{H}$-clustering χ to live in a category $\mathcal{X}_{\mathcal{H}}$ of sets of disjoint cluster hierarchy nodes, where we typically write $C \in \chi$ for such a node. The cluster hierarchy nodes are elements of the given partial order structure $\mathcal{H}$ in which objects represent different nodes in the hierarchy over the finite set $\mathcal{V}$. Each node must parent a non-empty subset of $\mathcal{V}$, and thus we have precisely $|\mathcal{V}|$ unique leaf nodes each corresponding to a single unique element. We may write them as singletons $\{v\}$ where we assume $v \in \mathcal{V}$.

The relation between elements of $\mathcal{H}$ is analogous to regular subset inclusion $x \subseteq y$ and thus overloads the notation so. It may receive the interpretation 'y *parents* x', which then encodes the hierarchy structure entirely with the usual properties of a (non-total) partial ordering.[2] In particular, we

[2]We have non-totality for $|\mathcal{V}| > 1$, since $\{v\}, \{v'\}$ with $v \neq v'$ are not related at all.

have that y *parents* x holds *if and only if* the set of variables 'held by' x is a subset of that of y. Then, for $\top_{\mathcal{H}} \coloneqq \bigcup \mathcal{H}$, it holds that $\forall C \in \mathcal{H}\left(C \subseteq \top_{\mathcal{H}}\right)$, and, importantly, we have that $\top_{\mathcal{H}}^{\dagger} = \mathcal{V}$, where $\dagger : \mathcal{H} \to \mathcal{P}(\mathcal{V})$ unfolds the hierarchy to the set of variables in that branch in the hierarchy; $C^{\dagger}$ may be read as: 'the set of variables held by C'.[3] Slightly abusing notation, we write '$c \in C$' for $c, C \in \mathcal{H}$ with $c \subset C$ and c is contained in the set of *direct* children of C, i.e. $\nexists c' \in \mathcal{H}\left(c \subset c' \subset C\right)$.

Moving back up to the level of clusterings, we may now compose requirements for what precisely defines a clustering. In particular, for all $\chi \in \mathcal{X}_{\mathcal{H}}$, it holds that $\mathcal{V} = \bigcup_{C \in \chi} C^{\dagger}$, and for all $C \in \mathcal{H}$,

$$C \in \chi \implies \left(\forall C' \in \chi \,.\, C^{\dagger} = C'^{\dagger} \vee C^{\dagger} \cap C'^{\dagger} = \varnothing\right). \tag{3.4}$$

This says: distinct elements of a clustering cannot be related, i.e. they are disjoint. Hereby, a clustering is a complete representation of a slice in the cluster hierarchy, and we would thus interpret $\chi^{\dagger}$ as $\mathcal{V}$ for all $\chi \in \mathcal{X}_{\mathcal{H}}$.

Such $\chi \in \mathcal{X}_{\mathcal{H}}$, i.e. objects in the category of $\mathcal{H}$-clusterings, are related by another partial order structure that relates $\chi_1, \chi_2 \in \mathcal{X}_{\mathcal{H}}$ *if and only if* χ_1 may be transformed into χ_2 in zero or more steps by taking a set $X \subseteq \chi_1$ and replacing it in χ_1 by the join $\bigcup X \in \mathcal{H}$, i.e., the least node parenting each $C \in X$.[4] This definition thus trivially satisfies the requirement of transitivity for this relation. The relation between elements of $\mathcal{X}_{\mathcal{H}}$ thus allows us to write chains $\chi_1 \precsim \chi_2 \precsim \ldots$ in $\mathcal{X}_{\mathcal{H}}$, enabling (least) fixed point iteration.

3.1.2 Operations on Clusterings

We now describe dual ways to traverse the partial order relation in $\mathcal{X}_{\mathcal{H}}$, both of which lay the foundations for the two phases of the algorithm that is presented in this work. In particular, the constructive phase described in Section 3.2, we start at $\bot_{\mathcal{X}_{\mathcal{H}}} \coloneqq \{\{v\} \mid v \in \mathcal{V}\}$ and end at $\top_{\mathcal{X}_{\mathcal{H}}} \coloneqq \left\{\top_{\mathcal{H}}\right\}$, while the destructive phase that unfolds the construct to ValidAssignments goes from $\top_{\mathcal{X}_{\mathcal{H}}}$ to $\bot_{\mathcal{X}_{\mathcal{H}}}$, as elaborated in Section 3.3.

Respectively, we have an algebra and an coalgebra both for the identity functor $\mathsf{Id}_{\mathcal{X}_{\mathcal{H}}}$ given by the respective maps $\mathfrak{C}$, $\mathfrak{D}$ that both take and return an object in $\mathcal{X}_{\mathcal{H}}$. Note that we could simply be talking about arbitrary maps on clusterings, yet the duality in these notions aligns well with the duality of the two phases of the proposed solution process.[5]

[3] Note that we may take unions here since $\mathcal{H}$ is a finite partial order structure that is simultaneously a directed set; i.e. for all $c, c' \in \mathcal{H}$, there is a $C \in \mathcal{H}$ such that C parents both c and c' (note that the 'parents' relation is reflexive). The join of some set X of nodes in $\mathcal{H}$ is then the node $C \in \mathcal{H}$ for which $C^{\dagger} = \bigcup_{c \in X} c^{\dagger}$.

[4] We then get a complete lattice with the bottom and top elements $\{\{v\} \mid v \in \mathcal{V}\}$ and the singleton $\left\{\top_{\mathcal{H}}\right\}$ respectively.

[5] Furthermore, the dummy $\mathsf{Id}_{\mathcal{X}_{\mathcal{H}}}$ functor makes semantically clear that these operations on clusterings are entirely self-contained.

We have a straightforward requirement for our constructor and destructor operations, enabling us to obtain respective fixed points. For all $\chi \in \mathcal{X}_{\mathcal{H}}$:

$$\left(\chi = \top_{\mathcal{X}_{\mathcal{H}}} = \mathfrak{C}\chi \vee |\mathfrak{C}\chi \setminus \chi| = 1\right) \wedge \left(\chi = \bot_{\mathcal{X}_{\mathcal{H}}} = \mathfrak{D}\chi \vee |\chi \setminus \mathfrak{D}\chi| = 1\right). \quad (3.5)$$

In practice, these may be implemented respectively by, e.g., choosing a $C \in \chi$ that is maximal (resp. minimal) in $\left|C^{\dagger}\right|$, though more elaborate heuristics may present a benefit that may depend on the problem instance. Abstracting from their implementation, we may direct our attention to the dynamics of construction and destruction discussed in the next sections, solely relying on the semantics of these operations to be given.

3.2 Minimised State Space Construction

Initialisation of the structure of the construct is step zero in the construction. The structure that is built is a hierarchical state space that corresponds with the given cluster hierarchy $\mathcal{H}$ in its hierarchical structure, representing, at each node, a minimised collection of distinct combinations of state assignments to variables held by the node. Here, the size of this collection is directly determined by the efficacy of the pruning operation with respect to the state acceptance map τ that was invoked for the node and its children. Indeed, then, the initialisation is given for each of the leaves, which each obtain a state space $\mathcal{S}$ containing the states that the associated variable may assume:

$$\forall v \in \mathcal{V}.\, \mathcal{S}_{\{v\}} \coloneqq \left\{\left(\{\!\{z\}\!\}, \left\{\left(\left\{p_{v,z}\right\}, \xi_{v,z}\right)\right\}\right) \;\middle|\; z \sim \tau_v\right\}, \quad (3.6)$$

where $\left\{p_{v,z}\right\} = \{(\{v\}, \{\!\{z\}\!\})\}$ and $\xi_{v,z} = \left[v \mapsto \left\{\psi_v([v \mapsto z])\right\}\right]$ together make a *state space composition* of a singleton cluster $\{v\}$, where $\left\{p_{v,z}\right\} \subset \mathcal{P}$ and $\xi_{v,z} : \{v\}^{\dagger} \to \mathcal{P}(W)$, to be refined for the higher compositions later. It pairs a set of *projector states* $\mathcal{P} \coloneqq \left\{(C, m) \mid C \in \mathcal{H},\ m \in \mathcal{S}_C^*\right\}$ of which the former and the latter part of some $p \in \mathcal{P}$ is indexed through $p_{\mathfrak{c}}$ and $p_{\mathfrak{m}}$ respectively, together with a map from each variable in the relevant cluster to a set that will be used to give a *finite range* in $\mathcal{P}(W)$, as determined by an upper and a lower bound.[6] State spaces thus contain pairs of state multisets along with a set of *compositions* of this multiset that is purely self-referential and singleton for the leaf nodes, i.e., the elements of the $\bot_{\mathcal{X}_{\mathcal{H}}}$ clustering, which define the initialisation. Hence, each state space element will be unique in the associated multiset, since the different ways to algebraically compose one multiset will be represented by the set of compositions. Disregarding this information, we may retrieve the set of multisets in some $\mathcal{S}_C$ by $\mathcal{S}_C^* \coloneqq \left\{m \mid (m, \_) \in \mathcal{S}_C\right\}$.

Ultimately, $\mathcal{S}'_{\top_{\mathcal{H}}}$ is constructed, marking the end of the construction phase. This state space will contain multisets that are composed of multisets of

[6] From now on, the powerset functor $\mathcal{P}$ is considered to be the *finite* powerset functor.

direct children of $\top_{\mathcal{H}}$, where we assume $|\mathcal{V}| > 1$ since the proposed method for computing ValidAssignments offers no benefit to such trivial problem sizes. In particular, since pruning operations are performed for each clustering on the path between $\bot_{\mathcal{X}_{\mathcal{H}}}$ and $\top_{\mathcal{X}_{\mathcal{H}}}$ as determined by $\mathfrak{C}$, by the algebraic property of compositionality we get that $\mathcal{S}_{\top_{\mathcal{H}}}$ may omit many combinations of state assignments flattened into a multisets, with respect to the exhaustive set that contains every multiset that contains one element of $\left\{z \in \mathcal{Z} \mid z \sim \tau_v\right\}$ for every $v \in \mathcal{V}$. We say 'flattened' here, since the multiset representation allows us to consider state information without directly considering where the states came from, i.e., to which variables they were assigned. The composition structure remembers this for us, allowing us to use multisets to represent multiple state assignments at once. By the distance heuristic that formed $\mathcal{H}$, the interaction from some *projecting cluster* to a variable, that is determined by a multiset state assignment to the former, is comparatively *strongly* bounded if the latter is not contained in the former; it varies weakly between different compositions of this multiset. Thereby, we keep track of the *bounds* of such an interaction from composition to composition as is described in the following, which are then used to perform pruning operations.[7]

To facilitate traversal of $\mathcal{X}_{\mathcal{H}}$, we require additional structure that keeps track of bound information. These ranges $\left[\underset{\sim}{w}, \hat{w}\right] \subseteq W$ for some $\underset{\sim}{w}, \hat{w} \in W$ with $\underset{\sim}{w} \leq \hat{w}$ are the flattened information associated with a state assignment combination represented by a multiset. From (3.1) we may observe that this set constitutes state assignments that have been evaluated with ψ independently by all variables, resulting in an aggregation of locally *received* interaction values that is then tested with the respective τ. Thereby, when considering clusters and flattened information, each cluster C has a bespoke role of a *projector* of interaction values onto variables, internal or external to C, that independently determine this value associated with state assignments to variables in $C^{\dagger}$. This is captured by a map $\pi_C : \mathcal{S}_C^* \to \mathcal{V} \to \mathcal{P}(W)$, such that for any $v \in \mathcal{V}$, we may retrieve the interaction value range spanned by some $m \in \mathcal{S}_C^*$ by $\pi_C(m)(v)$. Then the entire range of what C can project onto v as determined by its state space that determines π_C, is given by the map $\mathcal{B}_{\pi_C} : \mathcal{V} \to \mathcal{P}(W)$, where $\mathcal{B}_{\pi_C}(v) \coloneqq \bigcup_{m \in \mathrm{dom}(\pi_C)} \pi_C(m)(v)$. Lastly, for any $p \in \mathscr{P}$, we overload π through $\pi_p : \mathcal{V} \to \mathcal{P}(W)$ with $\pi_p \coloneqq \pi_{p_{\mathfrak{c}}}\left(p_{\mathfrak{m}} \in \mathcal{S}_{p_{\mathfrak{c}}}^*\right)$.

This map π provides a crucial extension to the state space structure,

[7] We defined the state acceptance map τ to be given as an arbitrary predicate, though the bounds we keep track of will span a range in $\mathcal{P}(W)$. It should be noted that some predicates, like ones that exclude only specific elements of W, might not synergise well with this approach for the efficacy of the pruning operations that depend on such a predicate. Specifically, by spanning a range we disregard which elements in this range may be formed in an *algebraic* sense. Future work may yield a variation of the proposed procedure to computing ValidAssignments in which the dynamic exclusion structure is refined to be synergetic with predicates for which the current solution is less useful, possibly at the cost of the complexity of the algebraic combinations that arise.

with which we can keep track of minimised, but exhaustive interaction value ranges associated with state space elements respective to each variable. Upon successful pruning, defined shortly, the pruned element is removed from its state space, introducing a dynamic trait to our structures that are then updated accordingly. Along with updates of projected interactions, the received interactions are updated also, which, for each cluster $\boldsymbol{C}$, are stored as a range spanned by bounds on the values aggregated from interactions *external* to the cluster. Similar to ξ, this map is defined on each $v \in \boldsymbol{C}^\dagger$: $\rho_{\boldsymbol{C}} : \boldsymbol{C}^\dagger \to \mathcal{P}(\mathcal{W})$, and we will define a *receptor state* as a pairing of a cluster and a variable contained by it: $\mathscr{R} \coloneqq \left\{(\boldsymbol{C}, v) \mid \boldsymbol{C} \in \mathcal{H},\ v \in \boldsymbol{C}^\dagger\right\}$. Similar to before, we may index the respective constituents of some $r \in \mathscr{R}$ by $r_{\mathfrak{c}}$ and $r_{\mathfrak{v}}$.

Having defined the dynamic structures that will allow us to progressively build a minimised state space, we continue the series of initialisations for every singleton cluster $\{v\} \in \bot_{\mathcal{X}_{\mathcal{H}}}$:

$$\forall z \sim \tau_v, v' \in \mathcal{V}.\, \pi_{\{v\}}\left(\{\!\{ z \}\!\}\right)\left(v'\right) \coloneqq \left\{\psi_{v'}\left([v \mapsto z]\right)\right\}, \text{ and} \tag{3.7}$$

$$\rho_{\{v\}}(v) \coloneqq \bigoverline{\bigoplus_{v' \in \mathcal{V},\ v' \neq v}} \mathscr{B}_{\pi_{\{v'\}}}(v), \tag{3.8}$$

where we define the powerset functor now specifically on the group operation $\overline{\underline{\oplus}} =: \mathcal{P}(\oplus) : \mathcal{P}(\mathcal{W}) \times \mathcal{P}(\mathcal{W}) \to \mathcal{P}(\mathcal{W})$ as follows, spanning a range that is determined by aggregated bounds of the arguments: for all $u, u' \in \mathcal{P}(\mathcal{W})$:

$$u \mathbin{\overline{\underline{\oplus}}} u' \coloneqq \left[\left(\bigwedge u\right) \oplus \left(\bigwedge u'\right), \left(\bigvee u\right) \oplus \left(\bigvee u'\right)\right] \in \mathcal{P}(\mathcal{W}).^{8} \tag{3.9}$$

This leads us to defining the pruning operation, defined as a predicate $\mathscr{T} : \mathcal{P}\left(\left\{(p, \xi) \in \mathscr{P} \times \left(\mathcal{P}(\mathcal{W})\right)^{\mathcal{V}} \;\middle|\; \operatorname{dom}(\xi) \supseteq p_{\mathfrak{c}}^\dagger\right\}\right)$, where the dependent function argument maps a variable contained by the cluster in the associated projector state to an interaction span, so that this predicate may be used in two separate ways: to test a *multiset* in a state space, and to test a *composition* of a multiset in a state space. The former kind may be performed on a static clustering, while the latter is used in the process of *merging*, which is elaborated on after the pruning procedure on static clusterings is presented.

The predicate $\mathscr{T}$ accepts the input based on *witness counting*. The multiset associated with the given projector state determines the demands in this regard; for some multiset $m \in \mathcal{M}(\mathcal{Z})$, a requirement emerges for any $z \in \mathcal{Z}$: the $m(z) \in \mathbb{N}$ occurrences of $z \in m$ demand that $m(z)$ variables must witness that z is accepted by the τ predicate respective to each variable, where the interaction value ranges determine the acceptance. That is, for all $p \in \mathscr{P}$ and $\xi : p_{\mathfrak{c}}^\dagger \to \mathcal{P}(\mathcal{W})$:

$$\mathscr{T}(p, \xi) \overset{\text{def}}{\iff} \forall z \in p_{\mathfrak{m}}.\, p_{\mathfrak{m}}(z) \leq \sum_{v \in p_{\mathfrak{c}}^\dagger} \operatorname{sign}\left|\left(\rho_{p_{\mathfrak{c}}}(v) \mathbin{\overline{\underline{\oplus}}} \xi(v)\right) \cap \tau_v(z)\right|. \tag{3.10}$$

[8] With this definition, the powerset functor maps a lattice-ordered abelian group into a lattice-ordered abelian monoid: $\left(\left\langle \mathcal{W}, \oplus, {}^{-1}, 0, \vee, \wedge \right\rangle, \leq\right) \overset{\mathcal{P}}{\longmapsto} \left(\left\langle \mathcal{P}(\mathcal{W}), \overline{\underline{\oplus}}, \mathcal{P}(0), \cup, \cap \right\rangle, \subseteq\right)$.

Notice that by disregarding state assignments to individual variables, we attempt to prune $p_{\mathfrak{m}} \in \mathcal{S}^*_{c_{\mathfrak{c}}}$ that represents a *combination* of state assignments to variables in $p^\dagger_{\mathfrak{c}}$. This 'lossy' approach tests whether a *weaker* property for non-acceptance is provable, which would then *imply* non-acceptance of $\tau_v(z)$ for *some* $v \in p^\dagger_{\mathfrak{c}}$ and $z \in p_{\mathfrak{m}}$, thereby proving that the state assignment combination represented by p *cannot* be part of an assignment in ValidAssignments; i.e:

$$\forall p \in \mathscr{P}.\, p \;\mathfrak{pruned} \overset{\text{def}}{\iff} \forall \alpha \in \mathsf{ValidAssignments}.\, p_{\mathfrak{m}} \neq \sum_{v \in p^\dagger_{\mathfrak{c}}} \{\!\{\alpha(v)\}\!\}\,, \quad (3.11)$$

$$\mathcal{S} \vdash \neg\mathscr{T}(p, \pi_p) \iff p \;\mathfrak{pruned}, \quad (3.12)$$

since $\pi_p\big|_{p^\dagger_{\mathfrak{c}}}, \rho_{p_{\mathfrak{c}}} : p^\dagger_{\mathfrak{c}} \to \mathcal{P}(\mathcal{W})$ together represent all information inferred by $\mathcal{S}$ that is local to the variables in cluster $p_{\mathfrak{c}}$, and $\mathcal{S}$ is complete with respect to ValidAssignments per hypothesis of the construction.

The lossiness, which simultaneously grants the $\mathcal{O}\left(\left|\mathrm{dom}\left(p_{\mathfrak{m}}\right)\right| \cdot \left|p^\dagger_{\mathfrak{c}}\right|\right)$ time complexity for evaluating $\mathscr{T}(p, \pi_p)$, can be attributed to both the disregard to specific compositions of $p_{\mathfrak{m}}$, and in particular the fact that any variable in $p^\dagger_{\mathfrak{c}}$ can be a witness for multiple states in this multiset, disregarding that any $\alpha \in \mathsf{ValidAssignments}$ may only assign one state to it. As a costly solution, *witness partitioning* could be performed, although the resulting subproblem may grow factorially in the number of overlapping witnesses, when solved as described in Appendix C.

Now let $\chi \in \mathcal{X}_{\mathcal{H}}$ and $\mathfrak{S}_\chi$ be the type of state spaces for this clustering such that $\mathfrak{S}_\chi : \forall C \in \chi\left(\mathcal{S}_C \times \pi_C \times \rho_C\right)$. We now define a procedure Update_χ of which we want to obtain the *least fixed point*; that is, this state space transformer is iterated on $\mathcal{S}$ until the identity is computed. To this end, we first define a non-commutative operation $\overline{\ominus} : \mathcal{P}(\mathcal{W}) \times \mathcal{P}(\mathcal{W}) \to \mathcal{P}(\mathcal{W})$ that may revert a range aggregation with respect to $\overline{\oplus}$ defined in (3.9); i.e., for all $u, u' \in \mathcal{P}(\mathcal{W})$, $\left(u \mathbin{\overline{\oplus}} u'\right) \mathbin{\overline{\ominus}} u' = u$:

$$u \mathbin{\overline{\ominus}} u' \coloneqq \left[\left(\bigwedge u\right) \oplus \left(\bigwedge u'\right)^{-1}, \left(\bigvee u\right) \oplus \left(\bigvee u'\right)^{-1}\right] \in \mathcal{P}(\mathcal{W}).^{9} \quad (3.13)$$

This operation allows us to write $\left(u \mathbin{\overline{\ominus}} u'\right) \mathbin{\overline{\oplus}} u'' = \left(u \mathbin{\overline{\oplus}} u''\right) \mathbin{\overline{\ominus}} u'$ for all $u, u', u'' \in \mathcal{P}(\mathcal{W})$ since the meets and joins of a finite non-empty interval

[9] This is defined for $\left(\bigwedge u\right) \oplus \left(\bigwedge u'\right)^{-1} \leq \left(\bigvee u\right) \oplus \left(\bigvee u'\right)^{-1}$. Further solidifying this definition and (3.9), we extend the notational convention with respect to the powerset functor—that was already assumed to yield only finite sets in [6]—now assuming that the empty set is excluded from the set of sets it returns. This avoids special attention for $\bigwedge \varnothing$ (and dually), since no infinity elements are assumed. [4] Moreover, the empty interval will never occur *actively* in the construction: it can occur *if and only if* $\mathcal{S}_C = \varnothing$ for some $C \in \chi$, which directly determines $\mathsf{ValidAssignments} = \varnothing$, thereby terminating the process.

are always the $\underset{\smile}{w}, \widehat{w} \in W$ with $\underset{\smile}{w} \leq \widehat{w}$ with which the interval was given. In particular, we may write $\left(u \mathbin{\underline{\ominus}} u'\right) \mathbin{\underline{\oplus}} u'' = u \mathbin{\underline{\ominus}} \left(u' \mathbin{\underline{\ominus}} u''\right)$ if $u' \mathbin{\underline{\ominus}} u''$ is defined (see [9]).[10] This facilitates an efficient update procedure when we consider $u = \rho_{C'}(v)$, $u' = \mathscr{B}_{\pi_C}(v)$ and $u'' = \mathscr{B}_{\pi'_C}(v)$ for some $C, C' \in \chi$ with $C \neq C'$, $v \in C'^{\dagger}$, $S'_C \subseteq S_C$ and $\pi'_C = \pi_C\big|_{S'^*_C}$; by (3.8) and hypothesis of the construction we have that $u = a \mathbin{\underline{\oplus}} u'$ for some aggregate $a \in \mathcal{P}(W)$ (possibly $\{0\}$, see [8]) and thus $u \mathbin{\underline{\ominus}} u'$ is defined, and in particular $u' \mathbin{\underline{\ominus}} u''$ is defined: recall from $S'_C \subseteq S_C$ that $u'' \subseteq u'$ by definition of $\mathscr{B}$, and that therefore $\bigwedge u'' \geq \bigwedge u'$ and $\left(\bigwedge u''\right)^{-1} \leq \left(\bigwedge u'\right)^{-1}$ (and dually), hence:

$$\begin{aligned}\left(\bigwedge u'\right) \oplus \left(\bigwedge u''\right)^{-1} &\leq \left(\bigwedge u'\right) \oplus \left(\bigwedge u'\right)^{-1} = 0 \\ &= \left(\bigvee u''\right) \oplus \left(\bigvee u''\right)^{-1} \leq \left(\bigvee u'\right) \oplus \left(\bigvee u''\right)^{-1}.\end{aligned} \tag{3.15}$$

Efficiency is found here by the fact that $u' \mathbin{\underline{\ominus}} u''$ may be derived without computing bounds on u' through $\mathscr{B}_{\pi_C}(v)$. Instead, for $R = S^*_C \setminus S'^*_C$:

$$\begin{aligned}&u' \mathbin{\underline{\ominus}} u'' \\ &= \left[\left(\bigwedge u'\right) \oplus \left(\bigwedge_{m \in S'^*_C} \bigwedge \pi_C(m)(v)\right)^{-1}, \left(\bigvee u'\right) \oplus \left(\bigvee_{m \in S'^*_C} \bigvee \pi_C(m)(v)\right)^{-1}\right] \\ &= \left[\left(\bigwedge_{m \in R} \bigwedge \pi_C(m)(v)\right) \oplus \left(\bigwedge u''\right)^{-1}, \left(\bigwedge_{m \in R} \bigwedge \pi_C(m)(v)\right) \oplus \left(\bigvee u''\right)^{-1}\right] \\ &=: \mathscr{B}_{\pi_C \setminus R}(v),\end{aligned} \tag{3.16}$$

since for all $m \in S^*_C \setminus R$, the elements of $\pi_C(m)(v) = \pi'_C(m)(v)$ cancel out. Algorithm 3.1 implements this dynamic updating of *received* interaction value ranges stored in ρ upon removal of state space elements, through computing differences in *projected* interaction value ranges: π versus π'.

Having reached a fixed point, we find that the information we can obtain from χ is exhausted. This is the point at which we may apply $\mathfrak{C}\chi = \chi' \in \mathcal{X}_{\mathcal{H}}$.

[10] In fact, the first part of the 'double negation' is always defined by commutativity of $\oplus$:

$$\begin{aligned}&\left(\bigwedge u\right) \oplus \left(\left(\bigwedge u'\right) \oplus \left(\bigwedge u''\right)^{-1}\right)^{-1} = \left(\bigwedge u\right) \oplus \left(\bigwedge u'\right)^{-1} \oplus \left(\bigwedge u''\right) = \left(\bigwedge u''\right) \\ &\leq \left(\bigvee u''\right) \leq \left(\bigvee u\right) \oplus \left(\bigvee u'\right)^{-1} \oplus \left(\bigvee u''\right) = \left(\bigvee u\right) \oplus \left(\left(\bigvee u'\right) \oplus \left(\bigvee u''\right)^{-1}\right)^{-1}.\end{aligned} \tag{3.14}$$

Algorithm 3.1: $\mathsf{Update}_\chi : \mathfrak{S}_\chi \to \mathfrak{S}_\chi$

Input: State space $\mathcal{S}_C$ and maps π_C and ρ_C for all $C \in \chi$
Output: For all $C \in \chi$, $\mathcal{S}'_C \subseteq \mathcal{S}_C$, along with $\pi'_C : \mathcal{S}'^*_C \to \mathcal{V} \to \mathcal{P}(W)$ and respectively updated $\rho_C : C^\dagger \to \mathcal{P}(W)$

foreach $C \in \chi$ **do**
  $R \leftarrow \left\{ m \in \mathcal{S}^*_C \mid p = (C, m) \in \mathscr{P},\ \neg\mathscr{T}\left(p, \pi_p\right) \right\}$;
  $\mathcal{S}'_C \leftarrow \left\{ (m, \Omega) \in \mathcal{S}_C \mid m \notin R \right\}$;
  $\pi'_C \leftarrow \pi_C\big|_{\mathcal{S}'^*_C}$;
  foreach $m \in R$ **do**
    foreach $C' \in \chi$ *with* $C' \neq C$ **do**
      foreach $v \in C'^\dagger$ **do**
        $\rho_C(v) \leftarrow \rho_C(v) \mathbin{\overline{\underline{\ominus}}} \mathscr{B}_{\pi_C \setminus R}(v)$
return $\left(\mathcal{S}', \pi', \rho\right)$

For the remainder, fix non-singleton $C \sim \chi' \setminus \chi$: the new parent. Primarily, we derive ρ_C from ρ_c for all $c \in C$ by subtracting projected interaction value ranges that are now *internal* to C, i.e., the interactions between newly-formed *siblings* that we will then consider in more detail. Recall that by our distance metric, the siblings were formed by their (mutual) comparatively high interaction bound difference. This update of received interaction value ranges is first applied to the children, after which it may be forwarded to ρ_C:

$$\forall c \in C, v \in c^\dagger \,.\, \rho_c(v) \leftarrow \rho_c(v) \mathbin{\overline{\underline{\ominus}}} \left(\overline{\underline{\bigoplus_{c' \in C,\, c' \neq c}}} \mathscr{B}_{\pi_{c'}}(v) \right), \text{ and} \tag{3.17}$$

$$\rho_C \coloneqq \bigsqcup_{c \in C} \rho_c. \tag{3.18}$$

Notice that the application of $\mathbin{\overline{\underline{\ominus}}}$ is defined here by similar reasoning to before when we found that $u \mathbin{\overline{\underline{\ominus}}} u'$ was defined.

This sets us up for the process of *state space merging*, which is sketched by considering the type of the merged state space below. Let $\mathcal{S}'$ be taken from $\mathfrak{S}_{\chi'}$; we require $\forall C' \in \chi\left(\mathcal{S}'_{C'} \coloneqq \mathcal{S}_{C'}\right)$, i.e., 'backwards compatibility', and we may specify the constraints for the largest state space for C as follows:

$$\mathcal{S}'_C \subseteq \left\{ (m, \Omega) \;\middle|\; \exists m \in \mathcal{M}(\mathcal{Z}),\ \Omega = \left\{ (\omega, \_) \in C_C \;\middle|\; m = \sum_{p \in \omega} p_{\mathfrak{m}} \right\} \neq \varnothing \right\}, \tag{3.19}$$

where, for any non-singleton $C' \in \mathcal{H} \setminus \bot_{\mathcal{X}_\mathcal{H}}$, the set of all *state space composi-*

tions of the children of C' is defined as follows:

$$\mathcal{C}_{C'} \coloneqq \left\{ \left(\omega \subset \mathscr{P}, \xi : C'^{\dagger} \to \mathcal{P}(W)\right) \;\middle|\; \forall c \in C' \,.\, \left|\left\{(p, \_) \in \omega \mid p_{\mathfrak{c}} = c\right\}\right| = 1, \right.$$
$$\left. \forall v \in C'^{\dagger} \,.\, \xi(v) = \overline{\bigoplus_{p' \in \omega}} \pi_{p'}(v) \right\}.^{11} \tag{3.20}$$

Hereby, this states in particular that a parent state space is *composed* of combinations of a single state space element of each of its children, enforced by $\exists p \in \mathscr{P}\Big(p_{\mathfrak{c}} = c \wedge p_{\mathfrak{m}} \in \mathcal{S}^{*}_{p_{\mathfrak{c}}} = \mathcal{S}'^{*}_{c}\Big)$ for each $c \in C$, where the different compositions of a multiset accumulate m are represented by Ω, which contains the distinct sets of projectors states ω that together constitute m.

Notice that this refers back to $\mathcal{S}$; in fact, similar to Update, the merging procedure we now describe defines a state space transformer too, now forming a fixed point for $\top = \mathfrak{C}\top$. The construction is thus described inductively, given the initial definitions for $\bot_{\mathcal{X}_{\mathcal{H}}}$, after which we abstracted to a generic $\chi \in \mathcal{X}_{\mathcal{H}}$. With our definitions, we ascertain that $\mathcal{S}'$ remains *complete* with respect to ValidAssignments, i.e., all $\alpha \in$ ValidAssignments may be obtained by forming assignments from $\mathcal{S}'$; the 'passive' variant of the state space destruction described in Section 3.3.

The type in (3.19) does not show any pruning—when all of $\mathcal{C}_C$ is found in $\mathcal{S}'_C$, completeness is intuitively satisfied by (3.20). However, our definitions now amass to a hierarchical structure that facilitates the quintessential operation of the proposed solution: inspecting a few sets of variables (i.e., $\{c \in C\}$) more elaborately, by now considering *all combinations* of state assignment combinations between the siblings (i.e., multisets from respective state spaces). Importantly, the given cluster hierarchy heuristically maximises the interaction variation, such that interaction value ranges, determined by respective bounds, are comparatively large, and thereby varied. By this variation heuristic, we *maximise* the chance that an arbitrary state acceptance map τ will be *sensitive* to different $(\omega, \xi) \in \mathcal{C}_C$, and thereby the proposed finds high pruning efficacy for an efficient number of queries during the construction.

We build $\mathcal{S}'_C$ by considering each $(\omega, \xi) \in \mathcal{C}_C$. In particular, we define $\mathcal{S}'_C$ to be the *smallest* set of its type given by (3.19), such that for such (ω, ξ):

$$(\omega, \xi) \in \bigcup_{(\_, \Omega) \in \mathcal{S}'_C} \Omega \overset{\text{def}}{\iff} \bigwedge_{p \in \omega} \mathscr{T}(p, \xi) \,.^{12} \tag{3.21}$$

[11] The ξ maps hereby give the bounds on the interaction sums *internal* to the parent cluster C for the paired ω. Thereby, these are bounds on *partial* sums, whereas in Section 3.3, these will be *complete* sums that capture information for which $\mathcal{V}$ is considered fully.

Now, by (3.19), (3.20) and (3.21), this determines $\mathcal{S}'_C$.

To now complete the construction with respect to $\mathfrak{S}_{\chi'}$, we must define the *projected* interactions of the parent cluster π_C, and make respective updates to all external *receivers* through ρ. We will use the notion of a receptor state $r = \left(r_{\mathfrak{c}}, r_{\mathfrak{v}}\right) \in \mathscr{R}$ to make a semantic separation that aids the higher-level view. Then, for any $r \in \mathscr{R}$ such that $r_{\mathfrak{c}} \in \chi' \setminus \{C\}$:

$$\forall\,(m,\Omega) \in \mathcal{S}'_C, \omega \in \Omega\,.\,\pi_C\,(m)\left(r_{\mathfrak{v}}\right) \coloneqq \overline{\bigoplus_{p \in \omega^*}}\, \pi_p\left(r_{\mathfrak{v}}\right), \tag{3.22}$$

after which we may update $\rho_{r_{\mathfrak{c}}}$ appropriately:

$$\rho_{r_{\mathfrak{c}}}\left(r_{\mathfrak{v}}\right) \leftarrow \left(\rho_{r_{\mathfrak{c}}}\left(r_{\mathfrak{v}}\right) \mathbin{\overline{\ominus}} \left(\overline{\bigoplus_{c \in C}}\, \mathscr{B}_{\pi_c}\left(r_{\mathfrak{v}}\right)\right)\right) \mathbin{\overline{\oplus}} \mathscr{B}_{\pi_C}\left(r_{\mathfrak{v}}\right), \tag{3.23}$$

where $\forall c \in C\left(c \neq r_{\mathfrak{c}}\right)$ and considerations for (3.17) ensure $\overline{\ominus}$ is defined here.

Lastly, now, we *flatten* the information in the different maps ξ indirectly contained in $\mathcal{S}'_C$, which, by (3.20) and [11], store sibling contributions to local interaction value ranges. This scope fills the space that was opened with (3.17), thus ensuring *all* interactions are accounted for when $\mathscr{T}$ is evaluated as in (3.10). After this step, when the merging process is done, $\mathscr{T}$ will be tried on *multisets* again with $\mathsf{Update}_{\chi'}$, rather than compositions. Then, by (3.12) and (3.10), we find that for all $(m,\Omega) \in \mathcal{S}'_C$ with $p = (C,m)$ a projector state, π_p must respect all $(\_,\xi) \in \omega \in \Omega$. This is done by taking respective meets and joins on either side (done implicitly in (3.24) below), ensuring that our interaction bounds are never spurious, thus ensuring completeness with respect to $\mathsf{ValidAssignments}$:

$$\forall\,(m,\Omega) \in \mathcal{S}'_C, v \in C^\dagger\,.\,\pi_C\,(m)\,(v) \coloneqq \bigcup_{(\_,\xi)\in\Omega} \xi\,(v), \tag{3.24}$$

thereby, together with (3.22), defining $\pi_{C'}$ for all $C' \in \chi$.

With these definitions, we obtain the second state space transformer $\mathsf{Merge}_\chi : \mathfrak{S}_\chi \to \mathfrak{S}_{\mathfrak{C}\chi}$, and thus also $\mathsf{Merge}_\chi \circ \bigsqcup_{k\in\mathbb{N}} \mathsf{Update}^k_\chi : \mathfrak{S}_\chi \to \mathfrak{S}_{\mathfrak{C}\chi}$ by Algorithm 3.1. The least fixed point of this operation applied to the initial $\left(\mathcal{S},\pi,\rho\right) \in \mathfrak{S}_{\bot_{\mathcal{X}_\mathcal{H}}}$ yields a *minimised* $\mathcal{S}'_{\top_\mathcal{H}}$, thus finishing the construction.

Algorithm 3.2: Minimised State Space Construction

1 $\bot_{\mathfrak{S}} = \left(\mathcal{S},\pi,\rho\right) \leftarrow \mathsf{Initialise};$

2 $\mathcal{S}_{\top_\mathcal{H}}\left(\mathcal{S}',\pi',\rho\right) \leftarrow \bigsqcup_{n\in\mathbb{N}} \left(\mathsf{Merge}_{\mathfrak{C}^n\bot_{\mathcal{X}_\mathcal{H}}} \circ \bigsqcup_{k\in\mathbb{N}} \mathsf{Update}^k_{\mathfrak{C}^n\bot_{\mathcal{X}_\mathcal{H}}}\right)^n \left(\bot_{\mathfrak{S}}\right)$

[12] Note here that, contrarily to the group lattice meet, '$\bigwedge$' performs a logical 'and'.

3.3 Active State Space Destruction

Now that we have obtained a minimal $\mathcal{S}_{\top_{\mathcal{H}}}$, we now unfold the information stored in it to ValidAssignments. Instead of extracting *all* state assignments that may be extracted from it, however, we destruct *actively* by taking pruning opportunities that $\mathcal{T}$ offers, however now by a different motive than (3.11): $\mathcal{T}$ is used on our stored state space compositions, though now in consideration with other cluster states that together form a clustering.

Recall by (3.20) that, for any $C \in \mathcal{H} \setminus \bot_{\mathcal{X}_{\mathcal{H}}}$, and for each of its state space compositions $(\omega, \xi) \in \mathcal{C}_C$ that is included in some Ω where $(m, \Omega) \in \mathcal{S}_C$, the map $\xi : C^\dagger \to \mathcal{P}(W)$ stores the hierarchical local interaction spans as aggregated with $\overline{\oplus}$ from only projectors in $\{c \subset C\}$. Yet, by (3.24), this interaction span is *flattened* with respect to the individual compositions of each multiset that is accumulated hierarchically: for all χ in a chain

$$\left\{\{v\} \;\middle|\; v \in C^\dagger\right\} = \bot_{\mathcal{X}_{\mathcal{H}_C}} \prec \mathfrak{C}\bot_{\mathcal{X}_{\mathcal{H}_C}} \precsim \cdots \precsim \chi \precsim \cdots \precsim \top_{\mathcal{X}_{\mathcal{H}_C}} = \{C\}\,, \tag{3.25}$$

such that there exists an $\omega \in \mathcal{X}_\chi$ for which $m = \sum_{p' \in \omega} p'_{\mathfrak{m}}$, where $\mathcal{X}_\chi$ is the set of *clustering states* for χ, i.e. the *complete* version of a state space composition (see [11]) by replacing C' with χ in the definition of $\mathcal{C}_{C'}$ in (3.20).

Note that the perspective shift between the complete hierarchy with $\mathcal{V} = \top^\dagger_{\mathcal{H}}$, and a *subhierarchy* of $\mathcal{H}$, i.e., $\mathcal{H}_C \coloneqq \{c \mid c \subseteq C\}$ for some $C \in \mathcal{H}$, allows us to see the ξ maps as interaction span representations relative to some clustering of the subhierarchy: $\mathcal{X}_{\{c \in C\}} = \mathcal{C}_C$, where $\{c \in C\} \in \mathcal{X}_{\mathcal{H}_C}$ is the $\mathcal{H}_C$-clustering of direct children of C. For this phase, however, we will always consider the complete hierarchy, which could be done by considering a set of subhierarchies H of which the set of respective $\top$ elements is a clustering for the complete hierarchy, i.e., $\left\{\top_h \mid h \in H\right\} \in \mathcal{X}_{\mathcal{H}}$. Thus, *aggregating* respective projections $\overline{\bigoplus}_{h \in H} \pi_{\top_h}$ precisely as in (3.20), we may prune in the active destruction process just as we did (3.21) with the ξ map built from internal projections, though this time taking the perspective of some $\top$ such that $\forall h \in H\,(h \subset \top)$, thus a more 'zoomed in' position; notice that previously, only $\top_{\mathcal{H}}$ considered all variables internal to it, yet this $\top$ may have direct children that are not direct children of $\top_{\mathcal{H}}$, and thus, at the end of a *branch*, we get $\forall v \in \mathcal{V}(\{v\} \in \top)$.

The character of branching is inherent to unfolding compositions that are constructed in a hierarchy. When *decomposing* some m from $\mathcal{S}_{\top_h}$ with $h \in H$ as before, a *particular* composition of projector states from $\mathcal{C}_{\top_h}$ makes a clustering state relative to h. Note that the interaction spans local to each variable in the subhierarchy were derived from flattened interaction spans of subhierarchies under h along with respective sibling projections (see (3.24), (3.20)), where the flattening enabled efficient pruning throughout the construction. Thus, we now need to *unflatten* during the process of unfolding that is characteristic of this phase, using the information stored in π.

By doing so, we may put the ξ maps to dynamic use in order to store interaction spans associated with different multiset unfoldings in it, i.e., clustering states, while keeping this contained to a single branch of the unfolding rather than relying on a globally available ρ. This map of the interaction spans as aggregated from projectors *external* to the given cluster was updated in particular with (3.17) and (3.18), such that, after the construction, $\rho_C = \mathcal{P}(\theta)$ for all $C \in \mathcal{H}$, since nothing is external to $\top_{\mathcal{H}}$. Hence, this *global* information structure is effectless in the definition of $\mathcal{T}$ in (3.10) (see [8]), consequently shifting all focus to *local* information, i.e., the ξ maps in the remaining compositions (ω, ξ) selected from some $\mathcal{C}$.

These state space compositions (ω, ξ), that are clustering states relative to some subhierarchy, will be combined to a clustering state of $\mathcal{H}$ through considering the compositions' projections onto *each* $v \in \mathcal{V}$, instead of onto only those v contained in the cluster that parents this composition, as in (3.20). To be precise, when Algorithm 3.1 reaches a fixed point, the merging process secures the elements of the state space of the children; they will be preserved in the construct $\mathcal{S}$. Thereby, we may prepare all compositions of these 'saved' state space elements to be used in the unfolding described with Algorithm 3.3, where we now overload $\overline{\oplus}$ and $\overline{\ominus}$ to be overloaded to operate on two functions $\xi, \xi' : \mathcal{V} \to \mathcal{P}(W)$, and return one by, e.g.: $\forall v \in \mathcal{V}\Big(\Big(\xi \,\underline{\overline{\oplus}}\, \xi'\Big)(v) \coloneqq \xi(v) \,\underline{\overline{\oplus}}\, \xi'(v)\Big)$. This preparation is simply to compute each ξ in one of these compositions by computing the aggregate in (3.20) for all remaining $v \in \mathcal{V}$ not in the respective child cluster.

Now, $\mathsf{ValidAssignments} \leftarrow \bigcup_{(m,\Omega)\in\mathcal{S}_{\top_{\mathcal{H}}}} \bigcup_{(\omega,\xi)\in\Omega} \mathsf{Unfold}_{\mathfrak{D}\top_{\mathcal{X}_{\mathcal{H}}}}(\omega, \xi)$.

Algorithm 3.3: $\mathcal{S} \vdash \mathsf{Unfold}_{\chi} : \mathcal{X}_{\chi} \to \mathcal{P}(\mathsf{ValidAssignments})$

Input: A clustering state $(\omega, \xi) \in \mathcal{X}_{\chi}$
Output: A subset of ValidAssignments

if $\bigvee_{p\in\omega} \neg\mathcal{T}(p, \xi)$ **then**
  return $\varnothing$ // **pruned**
if $\chi = \bot_{\mathcal{X}_{\mathcal{H}}}$ **then**
  return $\left\{ \bigsqcup_{(\{v\},\{\!\{z\}\!\})\in\omega} [v \mapsto z] \right\} \subseteq \mathcal{G}_{\mathcal{V}}$ // *found* ValidAssignment
$G \leftarrow \varnothing$;
$\hat{p} \sim \left\{ p \in \omega \mid \{p_{\mathfrak{c}}\} = \chi \setminus \mathfrak{D}\chi \right\}$; // *unfold* $\hat{p}$
$\omega \leftarrow \omega \setminus \{\hat{p}\}$;
$\xi \leftarrow \xi \,\underline{\overline{\ominus}}\, \pi_{\hat{p}}$; // *allow specialisations of* $\hat{p}$
foreach (ω', ξ') *such that* $\exists\, (\hat{p}_{\mathfrak{m}}, \Omega) \in \mathcal{S}_{\hat{p}_{\mathfrak{c}}} .\, (\omega', \xi') \in \Omega$ **do**
  $G \leftarrow G \cup \mathsf{Unfold}_{\mathfrak{D}\chi}\left(\omega \cup \omega', \xi \,\underline{\overline{\oplus}}\, \xi'\right)$ // *unfold specialisation*
return G

Chapter 4

Applications and Future Improvements

4.1 The Exact Physical SiDB Simulation Problem

With Chapter 3, the algorithm that defines *ClusterComplete*, presented in Section 2.5.2, is given for a much broader problem than is to be solved for exact physical simulation of SiDBs. The following example shows how this problem is instantiated to fit the problem structure given in the previous chapter, where we showed how to solve it.

Example 1. First off, the set of variables $\mathcal{V}$ is given by the system S that contains all SiDBs. Depending on the simulation base, the set of states $\mathcal{Z}$ is either $\{-1, 0, +1\}$ for base 3 (full base), or $\{-1, 0\}$ for base 2. Our values will live in the lattice-ordered abelian group given in Section 2.6: $(\langle \mathbb{R}, +, -, 0, \max, \min \rangle, \leq)$. Then, given the set of physical parameters, with in particular the μ_- material property (see Section 2.1.3), we may infer the state acceptance map τ. Yet, we will allow this parameter to be specified for each SiDB individually, which we thus localise with a superscript in the following. Preceding, however, since exact SiDB simulation rejects validity for any partially defined state assignment, we specify that $\tau_i\left(\perp_{\mathcal{Z}}\right) \coloneqq \varnothing$ for all $i \in S$. Further, for every SiDB $i \in S$:

$$\tau_i(-1) \coloneqq \left(-\infty, -\mu_-^i\right] \qquad \tau_i(0) \coloneqq \left[-\mu_-^i, -\mu_+^i\right] \qquad \tau_i(+1) \coloneqq \left[-\mu_+^i, \infty\right), \tag{4.1}$$

where we omitted the static simulation error as given in (2.34). For the case of base 2 simulation, τ_i would not be defined on $+1$.

Lastly, the state evaluation function ψ may be determined for every $i \in S$:

$$\psi_i\left(\alpha \in \mathcal{G}_S\right) \coloneqq \sum_{j \in \mathrm{dom}(\alpha)} \begin{cases} -V_i^{\mathrm{ext}} & \text{if } j = i, \text{ and} \\ V_{ij} \cdot \alpha(j) & \text{otherwise.}^{1} \end{cases}, \tag{4.2}$$

4.2 Instantiation to Boolean Satisfiability

Besides the application to the exact simulation of SiDBs, described in Section 2.5.2, the generalisation proposed in Chapter 3 is now instantiated to the NP complete problem SAT-CNF as an example, where this does not necessarily define the *only* such instantiation.

Example 2. For $\mathbb{B} \coloneqq \{0,1\}$, let $\varphi : \mathbb{B}^{\mathcal{V}} \to \mathbb{B}$ be a Boolean formula in conjunctive normal form, determining a Boolean judgement to a totally defined Boolean state assignment to variables. We first consider φ to be free of quantifiers. Then, assuming a non-trivial case, φ is represented by a non-empty set of clauses $\Upsilon \in \mathcal{P}(\mathcal{V} \times \mathbb{B})$, in which each clause contains pairs that represent literals: a variable and either 0 if it is negated, and 1 otherwise; therefore these correspond to the state assignment that makes the literal true. Furthermore, we will require that φ is *minimal* with respect to contradicting literals: i.e., $\forall (v,b) \in Y \in \Upsilon ((v, 1-b) \notin Y)$.

For our lattice-ordered abelian group, we choose $\langle \mathbb{Z}, +, -, 0, \max, \min \rangle$ relative to $\leq$, i.e., ordered integers. Since SAT-CNF is characterised by a *global* requirement entirely, we define the following for all $v \in \mathcal{V}$ equally, and thus omit the subscripts. Now, for $\alpha \in \mathbb{B}^{\mathcal{V}}$, we define $\tau \coloneqq \text{const}\,\{|\Upsilon|\}$, and:

$$\psi(\alpha) \coloneqq \sum_{Y \in \Upsilon} \text{sign}\left|\left\{\left(v', \alpha\left(v'\right)\right) \in Y \mid v' \in \text{dom}\,(\alpha)\right\}\right|, \text{ and} \tag{4.3}$$

Thereby, α is satisfied for v *if and only if* α, considering its domain, makes *at least one* literal true in every clause. Notice that the summation in (4.3) fits the requirement for ψ that we may 'take it apart' with respect to partially defined α.

Such a formula is naturally quantified existentially for each variable as SAT-CNF $\varphi \iff$ ValidAssignments $= \varnothing$, but now, as a step towards *quantified Boolean formulas* (QBF, PSPACE complete), consider $v_\forall \in \mathcal{V}$ such that $\varphi' \coloneqq \forall v_\forall \varphi$; i.e., φ' is true only if φ is true for all Boolean assignments to $v_\forall$. We may capture this with new global requirements:

$$\psi'(\alpha) \leftarrow \sum_{b \in \mathbb{B}} \psi\left(\alpha\left[v_\forall \mapsto b\right]\right) \qquad \text{and} \qquad \tau' \leftarrow \text{const}\left\{\sum_{b \in \mathbb{B}} |\Upsilon|\right\}. \tag{4.4}$$

.

Lastly, we assume $v_\exists \in \mathcal{V}$ such that $v_\exists \neq v_\forall$ in order to introduce a *quantifier alteration*: $\varphi'' \coloneqq \forall v_\forall \exists v_\exists \varphi$. This time, we use *local* requirements;

[1] This instantiation defines the local electrostatic potential evaluation V_i as in (2.2).

first $\psi'' \leftarrow \psi$ and $\tau'' \leftarrow \tau$, after which:

$$\begin{aligned} \psi''_{v_\forall}(\alpha) &\leftarrow \sum_{b \in \mathbb{B}} \psi\left(\alpha\left[v_\forall \mapsto 0, v_\exists \mapsto b\right]\right), \\ \psi''_{v_\exists}(\alpha) &\leftarrow \sum_{b \in \mathbb{B}} \psi\left(\alpha\left[v_\forall \mapsto 1, v_\exists \mapsto b\right]\right), \text{ and} \end{aligned} \tag{4.5}$$

$$\tau''_{v_\forall}, \tau''_{v_\exists} \leftarrow \text{const}\left[|\Upsilon|, \sum_{b \in \mathbb{B}} |\Upsilon|\right]. \tag{4.6}$$

Notice that the above is just one way of representing φ''; e.g., (4.3) could also have been adjusted to allow the element of freedom with respect to $v_\exists$. Here, however, we took advantage of the more expansive restriction that local requirements offer, to dividing the $\forall$ requirement for $v_\forall$ into two ranges: they independently accept *if and only if* one of the assignments to $v_\exists$ satisfies all clauses.

If we are to consider, besides the essence of the fit, its *use*, we must consider the quality of the heuristic that the distance metric offers in terms of *variety*, as discussed in Section 3.2. Also, referring to [7], the definition of τ may not be fruitful for the pruning efficacy. In particular, τ synergises best with the proposed methods when it is defined in terms of ranges; a coordinate space lattice may not be efficacious in this way for accepting coordinates that are densely spaced, relative to the interaction spans determined by ψ. In the case of Example 2, the defined ranges are either singletons, or, in (4.6), a range that is scaled in proportion to the definition of ψ.

To inspect the relation between a Boolean formula and variable distance variety, let φ be an arbitrary SAT-CNF formula for which $\mathcal{V}$, $\mathcal{Z}$, τ and ψ are defined as described with (4.3). By (3.2) and (3.3), we get that, for any $v, v' \in \mathcal{V}$, $d_{v,v'}$ is inversely proportional to the difference between the minimum and the maximum number of clauses that v and v' can together satisfy. Therefore, by (4.3), and the assumption of φ being minimal, as discussed in Example 2, this difference depends on the number of clauses that contain v or v', thus attaining the smallest distance to the variable pair that together cover the most clauses. For the rather plain case of SAT-CNF, we have no variation in the number of satisfied clauses between a varying amount of clauses that v and v' share. However, when introducing a *dependency* through some $v \wedge v'$, this dependency between the state assignments to v and v' is represented in the distance metric as the new difference is bounded from above by the old.

When considering the objective of existential satisfiability, then by the inherent relativity of the distance heuristic, such an increase in the distance between variables becoming *dependent* highlights the other relations, i.e., pairs of variables, that may together contribute a greater interaction range to satisfiability of φ, with in particular one assignment combination that

maximises the number of satisfied clauses, thereby putting it forward as a probable *fragment* of a satisfying assignment. For a universal satisfiability query, testing if φ is a *tautology*, i.e., $\forall \alpha \in \mathcal{G}_V(\varphi(\alpha) = 1)$, the definitions in Example 2 may be adjusted to accommodate the destructive objective of *dis*proving as opposed to constructive proving of existence, which may be done by counting the number of *un*satisfied clauses in (4.3), and then leaving τ the way it is.[2] Then, the goal of obtaining a certificate for proving that φ is not a tautology becomes both a feat of construction and destruction with the proposed, as we may construct a hierarchical representation of what such a certificate may consist of, to then destructively search in it *actively*.

Fragments of assignments that, exactly as before, contribute a large *variety* to the satisfiability of φ, are highlighted by the distance metric, such that in particular the fragments that are likely to contribute to such a certificate may be retained in the construction of a minimised hierarchical state space of assignment fragments that *do not* satisfy φ, under bounds on the satisfiability coverage of assignments to remaining variables. Then, unfolding this state space, we assess actively what combinations of fragments for which we have specialised bounds we may discard in the search of a φ-tautology-disproving certificate. Thereby, the search for a satisfying assignment to φ is performed in precisely the same manner, modulo the inversion in the acceptance criterion.

4.3 Corecursive Hierarchy Construction

In Section 3.1 we assume a hierarchy $\mathcal{H}$ to be given, which was then used as a static object in the rest of Chapter 3. This decision in the presentation is rooted in the implementation of *Ground State Space* (see Section 2.5.2), which relies on ALGLIB functionality to produce a hierarchy through agglomerative clustering based on a Euclidean distance metric.[3] Traversal of this hierarchy was implemented through merging to (resp. splitting) the smallest (resp. largest) parent for the semantic operations $\mathfrak{C}$ and $\mathfrak{D}$ respectively. With the benefit of hindsight, it was observed that in particular this implementation for $\mathfrak{C}$ appears to be the only implementation that takes full advantage of the static hierarchy. This suggests that Chapter 3 overgeneralises in this regard.

It is proposed that *dynamic* hierarchy construction resolves this dissonance in the generalisation, although this was not implemented for any instantiation like Example 1. Recall that the static hierarchy was risen from bound information derived from the given system before any pruning was performed (see (3.2), (3.3)). Putting it this way, it is already implied that the refined bound information that is gained throughout the construction in Section 3.2

[2] Specifically, the number of dissatisfied clauses should only be counted for the clauses that contain a variable in the domain of the partial state assignment.

[3] `clusterizer` from ALGLIB Free Edition was used (https://www.alglib.net/).

may serve to determine a better merge operation than merging to the smallest parent in the static $\mathcal{H}$. The dependency of Section 3.2 on this staticity may be resolved through restructuring the construction to a *corecursive* procedure.

4.4 State Relating and Global Evaluation

In much throughout Section 2.4, we considered the concept of *relevancy*, which connected minimal validity error to state occupation probabilities that arise in a thermodynamic context. More generally, the latter can be thought of as an attribute to a state assignment as a whole, i.e., a *global* evaluation, in addition to the local ones. In the context of furthering Chapter 3, the minimal stability error from (2.36) is generalised to a minimal validity error, i.e., the minimal $w \in W$ relative to $\leq$ that describes the difference between any element of the acceptance region for a given state, and any element in the given local interaction value range, where $w \leq \theta$ precisely when the two regions share at least one element. However, the principle idea of *pruning* relies on setting a particular threshold—which would be θ minimal validity error mark, such that any computed difference w below it results in acceptance of the associated validity query. For the efficiency of the solving procedure, the pruning, i.e., the state space minimisation, is the key ingredient to tackling the exponential blowup in the number of states we need to consider for increasing $|\mathcal{V}|$.

Other than computing ValidAssignments (VA), the techniques presented in Chapter 3 may find generalisation to the feat of computing a preorder relation over $\mathcal{G}_{\mathcal{V}}$, *lazily*. That is, $\mathcal{G}_{\mathcal{V}}$ would be sorted such that we may efficiently yield the states that would be, say, ordered below all of $\mathcal{G}_{\mathcal{V}}$; the instantiation to computing ValidAssignments may correspond to a preorder yielding two equivalence classes that would precisely be VA and $\mathcal{G}_{\mathcal{V}} \setminus \mathsf{VA}$, with the former ordered below the latter. For proving this, one would connect the transitivity of the preorder to the transitivity in validity between multiset compositions of multisets, thus defining a preorder *totally* through specifying fewer relations on multiset state assignments. Thereby, for added equivalence classes, e.g., subdivisions of $\mathcal{G}_{\mathcal{V}} \setminus \mathsf{VA}$, the defining relations would specialise over the separation that was computed with VA, thus explaining the *lazy* aspect: the strictest selections are computed first, such that selections of states for which a weaker requirement holds may be computed through *back-tracking* through the hierarchical state space, looking the highest points in the constructed hierarchy at which the state spaces that were attuned to the prior criterion are *complete* with respect to the weaker one (Section 3.2, under (3.20)). These may be taken as the minimal distance starting point to refine the partial order structure through furthering the existing multiset composition hierarchy with new multisets and compositions.

Any preorder may thus be computed through iteration, akin to the

iteration in Section 2.4.2, though in this case starting with the strictest requirement, then weakening iteratively, each time obtaining an additional equivalence class for which a weaker requirements hold. However, if we are to consider globally evaluated attributes to state assignments that live in perhaps another lattice-ordered abelian group, these may be bounded similar to the locally evaluated ones, given that, like our requirement for all ψ_v, the global evaluation decomposes into aggregations of disjoint partial global evaluations. Using the minimised hierarchical state space, global evaluations may be bounded: for the SiDB domain, the system energy sums the local evaluations in multiplication with their assigned integer state (see (2.6)), which may thus by bounded through both bounds on the local evaluations, and knowledge of pruned state assignments as presented by the minimised state space structure. The precise intricacies of assigning bounds on the global evaluation value to partial state assignments—that were the projector states in Chapter 3—are left to future progression in the logical framework of *hierarchically bounded composable interactions*.

4.5 Summary and Concluding Ideas

In this thesis, we have considered the problem of finding the exhaustive global state selection of universally satisfied *fragmentable* local evaluations. The latter property was shown to be crucial for the proposed method for solving this problem, as it allows us to solve arbitrary subsystems, where we may then use those results to construct solutions for bigger systems consisting of disjoint subsystems that we solved independently. The process relies wholly on fixed point iteration in a *cluster hierarchy*, elevating the iterative character of state space pruning to proceed in a novel dimension. Throughout, the information that we have—i.e., pruned possibilities within the system—is utilised to yield *maximal* new information. Then, besides this efficacious aspect of the proposed method, we motivated that apt information compression facilitates efficiency: permutation information of a subsystem may be flattened to respective combination information with *minimal* information loss with respect to the interactions projected onto other subsystems.

However, the 'lost' information is set aside in a *composition* structure, rather, allowing the constructed minimised hierarchical state space to be destructed *actively* through uniting respective composition-specific information of disjoint subhierarchies into information of the complete hierarchy, generating pruning opportunities that may again be capitalised upon. Thereby, this thesis presents an algorithm for computing the desired subset of a set that grows exponentially in the number of variables that exhibits a real-valued reduction of the base of the exponent, refining the problem's static-base exponential time complexity to be determined precisely by the *specifics* of the given system of interactions, as demonstrated with *ClusterComplete*.

Bibliography

[1] R. Achal et al. 'Lithography for Robust and Editable Atomic-Scale Silicon Devices and Memories'. In: *Nature Communications* 9 (2018). DOI: 10.1038/s41467-018-05171-y.

[2] K. Ahmed and K. Schuegraf. 'Transistor wars'. In: *IEEE Spectrum* 48 (2011), pp. 50–66. DOI: 10.1109/MSPEC.2011.6056626.

[3] R. J. Behm and W. Hösler. 'Scanning Tunneling Microscopy'. In: ed. by R. Vanselow and R. Howe. 1986, pp. 361–411. ISBN: 9783642827273. DOI: 10.1007/978-3-642-82727-3_14.

[4] G. Birkhoff. *Lattice Theory*. 3rd. 1967. DOI: 10.2307/2268183.

[5] H. B. Callen. *Thermodynamics and an introduction to thermostatistics; 2nd ed.* 1985. DOI: 10.1119/1.19071.

[6] H. N. Chiu. 'Simulation and Analysis of Clocking and Control for Field-Coupled Quantum-Dot Nanostructures'. PhD thesis. University of British Columbia, 2020. DOI: 10.14288/1.0389918.

[7] J. Croshaw et al. 'Atomic Defect Classification of the H–Si(100) Surface through Multi-Mode Scanning Probe Microscopy'. In: *Beilstein Journal of Nanotechnology* 11 (2020), pp. 1346–1360. ISSN: 2190-4286. DOI: 10.3762/bjnano.11.119.

[8] J. Drewniok, M. Walter, S. S. Hang Ng et al. 'QuickSim: Efficient and Accurate Physical Simulation of Silicon Dangling Bond Logic'. In: 2023, pp. 817–822. DOI: 10.1109/NANO58406.2023.10231266.

[9] J. Drewniok, M. Walter and R. Wille. 'Temperature Behavior of Silicon Dangling Bond Logic'. In: 2023, pp. 925–930. DOI: 10.1109/NANO58406.2023.10231259.

[10] J. Drewniok, M. Walter and R. Wille. 'Minimal Design of SiDB Gates: An Optimal Basis for Circuits Based on Silicon Dangling Bonds'. In: International Symposium on Nanoscale Architectures '23. 2024. ISBN: 9798400703256. DOI: 10.1145/3611315.3633241.

[11] J. Drewniok, M. Walter and R. Wille. 'The Need for Speed: Efficient Exact Simulation of Silicon Dangling Bond Logic'. In: Asia and South-Pacific Design Automation Conference '24. 2024, pp. 576–581. ISBN: 9798350393545. DOI: 10.1109/ASP-DAC58780.2024.10473946.

[12] J. Hartmanis and R. E. Stearns. 'On the Computational Complexity of Algorithms'. In: *Transactions of the American Mathematical Society* 117 (1965), pp. 285–306. DOI: 10.1090/S0002-9947-1965-0170805-7.

[13] T. R. Huff, T. Dienel et al. 'Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot'. In: *ACS Nano* 13.9 (2019), pp. 10566–10575. ISSN: 1936-0851. DOI: 10.1021/acsnano.9b04653.

[14] T. R. Huff, H. Labidi et al. 'Binary Atomic Silicon Logic'. In: *Nature Electronics* 1.12 (2018), pp. 636–643. ISSN: 2520-1131. DOI: 10.1038/s41928-018-0180-3.

[15] J. Kühr. 'Ideals of Noncommutative DR -Monoids'. In: *Czechoslovak Mathematical Journal* 55.1 (2005), pp. 97–111. ISSN: 1572-9141. DOI: 10.1007/s10587-005-0006-0.

[16] W. Lambooy, C. Kop and S. Smetsers. *Linearly and Arboreally Stackable Quantum-dot Cellular Automata and Their Discrete Simulation*. 2021. URL: https://www.cs.ru.nl/bachelors-theses/2021/Willem_Lambooy___1009854___Linearly_and_arboreally_stackable_quantum-dot_cellular_automata_and_their_discrete_simulation.pdf.

[17] W. Lambooy, M. Walter and R. Wille. 'Exploiting the Third Dimension: Stackable Quantum-dot Cellular Automata'. In: International Symposium on Nanoscale Architectures '22. 2023. ISBN: 9781450399388. DOI: 10.1145/3565478.3572529.

[18] R. Landauer. 'Irreversibility and Heat Generation in the Computing Process'. In: *IBM Journal of Research and Development* 5.3 (1961), pp. 183–191. DOI: 10.1147/rd.53.0183.

[19] C. S. Lent, M. Liu and Y. Lu. 'Bennett Clocking of Quantum-Dot Cellular Automata and the Limits to Binary Logic Scaling'. In: *Nanotechnology* 17.16 (2006), p. 4240. DOI: 10.1088/0957-4484/17/16/040.

[20] L. Livadaru et al. 'Dangling-Bond Charge Qubit on a Silicon Surface'. In: *New Journal of Physics* 12.8 (2010), p. 083018. DOI: 10.1088/1367-2630/12/8/083018.

[21] G. E. Moore. 'Cramming More Components onto Integrated Circuits'. In: *Electronics* 38.8 (1965), pp. 114–117. DOI: 10.1109/N-SSC.2006.4785860.

[22] S. Ng et al. 'SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits'. In: *IEEE Transactions on Nanotechnology* (2020), pp. 1–1. DOI: 10.1109/TNANO.2020.2966162.

[23] S. S. H. Ng et al. 'Simulating Charged Defects in Silicon Dangling Bond Logic Systems to Evaluate Logic Robustness'. In: *IEEE Transactions on Nanotechnology* 23 (2024), pp. 231–237. DOI: 10.1109/TNANO.2024.3372946.

[24] S. S. H. Ng. 'Computer-Aided Design of Atomic Silicon Quantum Dots and Domputational Applications'. PhD thesis. University of British Columbia, 2020. DOI: 10.14288/1.0392909.

[25] J. Pitters et al. 'Atomically Precise Manufacturing of Silicon Electronics'. In: *ACS Nano* 18.9 (2024), pp. 6766–6816. ISSN: 1936-0851. DOI: 10.1021/acsnano.3c10412.

[26] M. Rashidi et al. 'Initiating and Monitoring the Evolution of Single Electrons Within Atom-Defined Structures'. In: *Phys. Rev. Lett.* 121 (16 2018), p. 166801. DOI: 10.1103/PhysRevLett.121.166801.

[27] J. Retallick and K. Walus. 'Limits of Adiabatic Clocking in Quantum-Dot Cellular Automata'. In: *Journal of Applied Physics* 127 (2020), p. 054502. DOI: 10.1063/1.5135308.

[28] D. Riedel. 'Surface Hydrogenation of the Si(100)-2×1 and Electronic Properties of Silicon Dangling Bonds on the Si(100):H Surfaces'. In: 2017, pp. 1–24. ISBN: 9783319518466. DOI: 10.1007/978-3-319-51847-3_1.

[29] M. Taucer. 'Silicon Dangling Bonds Non-equilibrium Dynamics and Applications'. PhD thesis. University of Alberta, 2015. DOI: 10.7939/R33B5WG81.

[30] M. D. Vieira et al. 'Three-Input NPN Class Gate Library for Atomic Silicon Quantum Dots'. In: *IEEE Design & Test* 39.6 (2022), pp. 147–155. DOI: 10.1109/MDAT.2022.3189814.

[31] M. M. Vopson. 'Estimation of the Information Contained in the Visible Matter of the Universe'. In: *AIP Advances* 11.10 (2021). DOI: 10.1063/5.0064475.

[32] M. Walter, J. Croshaw et al. 'Atomic Defect-Aware Physical Design of Silicon Dangling Bond Logic on the H-Si(100)2x1 Surface'. In: *arXiv* (2023). DOI: 10.48550/arXiv.2311.12042.

[33] M. Walter, J. Drewniok et al. 'Reducing the Complexity of Operational Domain Computation in Silicon Dangling Bond Logic'. In: International Symposium on Nanoscale Architectures '23. 2024. ISBN: 9798400703256. DOI: 10.1145/3611315.3633246.

[34] M. Walter, S. S. H. Ng et al. 'Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic'. In: Design Automation Conference '22. 2022, pp. 739–744. ISBN: 9781450391429. DOI: 10.1145/3489517.3530525.

[35] M. Walter, R. Wille et al. *Design Automation for Field-coupled Nanotechnologies*. 2022. ISBN: 9783030899516. DOI: 10.1007/978-3-030-89952-3.

[36] R. A. Wolkow et al. 'Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics'. In: ed. by N. G. Anderson and S. Bhanja. 2014, pp. 33–58. ISBN: 9783662437223. DOI: 10.1007/978-3-662-43722-3_3.

[37] D. Yang, X. Liang and X. Yu. 'Silicon Wafer Processing'. In: ed. by Y. Wang et al. 2024, pp. 1665–1676. ISBN: 9789819928361. DOI: 10.1007/978-981-99-2836-1_75.

[38] N. Zhang. *Moore's Law is Dead, Long Live Moore's Law!* 2022. DOI: 10.48550/arXiv.2205.15011. arXiv: 2205.15011 [cs.GL].

Appendix A

Properties of the Functor $\mathcal{D}$

A.1 $\mathcal{D}$ is a Functor

For $\mathcal{D}$ to be considered a functor, we need to check that the domains are respected, as well as the reflexivity and transitivity of the preorder.
Proof: Primarily, we know that $\mathcal{D}$ respects the domains by the definition of said domains. Now take some arbitrary system S, and begin with $n \in \mathbb{N}$, $n = 0$ and $\alpha \in \mathfrak{F}_S^{\gamma_n} = \mathcal{G}_S \cup \{\top\}$ by (2.39). Throughout this, we will simultaneously show the required properties for $\precsim_{n'}$ to be considered a preorder for all $n' \in \mathbb{N}$ using induction. In addition, we can claim inductively that for all $n'' \in \mathbb{N}$ with $n'' \geq n'$, we have $\alpha \neq \top \wedge \hat{\mathfrak{e}}_{\gamma_{n''}} \precsim_{n'} \alpha \implies \alpha = \hat{\mathfrak{e}}_{\gamma_{n''}}$. By definition of $\precsim_0$, we obtain that reflexivity and transitivity are respected trivially, and our claim holds trivially.

Now take these properties for our induction hypothesis (IH) for some $n' \in \mathbb{N}$, and let the claimed property be labelled $(*)$; we now consider $n' + 1$. For the reflexivity requirement, take some $\alpha \in \mathfrak{F}_S^{\gamma_{n'+1}}$, then, by definition, there exists a $\overset{\ddagger}{\alpha} \in \mathfrak{F}_S^{\gamma_{n'}}$ such that $\alpha = \mathcal{D}(\overset{\ddagger}{\alpha})$. By our IH, we have $\overset{\ddagger}{\alpha} \precsim_{n'} \overset{\ddagger}{\alpha}$, and thus (2.45) gives us $\alpha \precsim_{n'+1} \alpha$ directly. Hence we continue and prove $(*)$ for $n' + 1$, so assume $\alpha \neq \top$ and $\hat{\mathfrak{e}}_{\gamma_{n'+1}} \precsim_{n'+1} \alpha$. If $\alpha = \hat{\mathfrak{e}}_{\gamma_{n'+1}}$, we are done, so assume the converse. In this case, we get that only (2.45) applies, which tells us $\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'+1}} \in \mathfrak{F}_S^{\gamma_{n'}}\right) = \hat{\mathfrak{e}}_{\gamma_{n'+1}} \precsim_{n'} \overset{\ddagger}{\alpha} \neq \top$. Now the induction hypothesis for $(*)$ tells us that $\overset{\ddagger}{\alpha} = \hat{\mathfrak{e}}_{\gamma_{n'+1}}$, and by (2.40) we get that this is equal to α, so we get a contradiction, giving us our conclusion that $\alpha = \hat{\mathfrak{e}}_{\gamma_{n'+1}}$.

For transitivity, take some other $\alpha', \alpha'' \in \mathfrak{F}_S^{\gamma_{n'+1}}$ for which there exist $\overset{\ddagger}{\alpha'}, \overset{\ddagger}{\alpha''} \in \mathfrak{F}_S^{\gamma_{n'}}$ such that $\alpha' = \mathcal{D}\left(\overset{\ddagger}{\alpha'}\right)$ and $\alpha'' = \mathcal{D}\left(\overset{\ddagger}{\alpha''}\right)$, and also $\alpha \precsim_{n'+1} \alpha'$ and $\alpha' \precsim_{n'+1} \alpha''$. We can distinguish cases by (2.44) and (2.45) in order to arrive at the conclusion $\alpha \precsim_{n'+1} \alpha''$, where we assume that $\alpha \neq \alpha' \neq \alpha''$ since our assumptions otherwise trivially give us this conclusion. Furthermore, by $(*)$, we get that if $\overset{\ddagger}{\alpha'} = \hat{\mathfrak{e}}_{\gamma_{n'+1}} = \mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'+1}}\right) = \alpha'$, then $\alpha' = \alpha''$, thus trivially satisfying transitivity.

We can identify all combinations of the relations $\left\{ \overset{*}{\alpha} = \hat{\mathfrak{e}}_{\gamma_{n'}}, \overset{*}{\alpha} \neq \hat{\mathfrak{e}}_{\gamma_{n'}} \right\}$, $\left\{ \alpha^{*\prime} = \hat{\mathfrak{e}}_{\gamma_{n'}}, \alpha^{*\prime} \neq \hat{\mathfrak{e}}_{\gamma_{n'}} \right\}$, and $\left\{ \alpha^{*\prime\prime} = \hat{\mathfrak{e}}_{\gamma_{n'+1}}, \alpha^{*\prime\prime} \neq \hat{\mathfrak{e}}_{\gamma_{n'+1}} \right\}$, which covers of the cases with respect to (2.44) and (2.45) we did not close out yet, apart from $\alpha^{*\prime\prime} = \hat{\mathfrak{e}}_{\gamma_n}$ which is handled last. Additionally, by (2.40), we find that only the states that are equal to $\hat{\mathfrak{e}}_{\gamma_{n'}}$ in any of the these cases may not be a fixed point of $\mathcal{D}$, and there may only be one such, since (2.40) tells us that with the converse we can relate two of $\alpha, \alpha', \alpha''$ as equal through $\top$. Thus, we may look at the remaining cases independently; excluding, primarily, cases that either relate any of $\overset{*}{\alpha}, \alpha^{*\prime}, \alpha^{*\prime\prime}$ as equal, since (2.40) then relates the corresponding pair in $\alpha, \alpha', \alpha''$ as equal. Further, we may find cases to which (2.44) does not apply at all and all states are fixed points of $\mathcal{D}$, from which we can easily get to the conclusion by the transitivity from the IH. It can thus be observed that this applies for all cases in which $\alpha^{*\prime\prime} \neq \hat{\mathfrak{e}}_{\gamma_{n'+1}}$, again excluding $\alpha^{*\prime\prime} = \hat{\mathfrak{e}}_{\gamma_{n'}}$ for now. Therefore, assume $\alpha^{*\prime\prime} = \hat{\mathfrak{e}}_{\gamma_{n'+1}} = \alpha''$ by (2.40) for the following cases.

$\alpha = \overset{*}{\alpha} \neq \hat{\mathfrak{e}}_{\gamma_{n'}} \neq \alpha^{*\prime} = \alpha'$**.** Here we may apply (2.45) to $\alpha \precsim_{n'+1} \alpha'$, giving us $\overset{*}{\alpha} \precsim_{n'} \alpha^{*\prime}$. Moreover, we have $\alpha^{*\prime} \precsim_{n'} \hat{\mathfrak{e}}_{\gamma_{n'}} = \mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'}} \in \mathfrak{F}_S^{\gamma_{n'}}\right)$ by (2.44). Our induction hypothesis then makes $\overset{*}{\alpha} \precsim_{n'} \hat{\mathfrak{e}}_{\gamma_{n'}}$ and so (2.44) makes $\alpha \precsim_{n'+1} \hat{\mathfrak{e}}_{\gamma_{n'+1}} = \alpha''$.

$\alpha = \overset{*}{\alpha} = \hat{\mathfrak{e}}_{\gamma_{n'}} \neq \alpha^{*\prime} = \alpha'$**.** Here we apply (2.44) since we have $\alpha = \overset{*}{\alpha} \precsim_{n'} \hat{\mathfrak{e}}_{\gamma_{n'}}$ by reflexivity, directly giving us $\alpha \precsim_{n'} \alpha''$.

$\alpha = \overset{*}{\alpha} \neq \hat{\mathfrak{e}}_{\gamma_{n'}} = \alpha^{*\prime} = \alpha'$**.** We first apply (2.45) to $\alpha \precsim_{n'+1} \alpha' \neq \hat{\mathfrak{e}}_{\gamma_{n'+1}}$ and get $\overset{*}{\alpha} \precsim_{n'} \alpha^{*\prime} = \hat{\mathfrak{e}}_{\gamma_{n'}} = \alpha'$. Therefore, (2.44) makes $\alpha \precsim_{n'+1} \hat{\mathfrak{e}}_{\gamma_{n'+1}} = \alpha''$.

$\alpha = \top \neq \overset{*}{\alpha} = \hat{\mathfrak{e}}_{\gamma_{n'}} \neq \alpha^{*\prime} = \alpha'$**.** Through applying (2.44) to $\alpha' \precsim_{n'+1} \alpha''$, we have that $\alpha^{*\prime} \precsim_{n'} \hat{\mathfrak{e}}_{\gamma_{n'}} = \mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'}}\right)$ but we know this not the case by $\alpha \neq \overset{*}{\alpha}$. Therefore this case gives us a contradiction, and can thus not apply.

$\alpha = \overset{*}{\alpha} \neq \hat{\mathfrak{e}}_{\gamma_{n'}} = \alpha^{*\prime} \neq \top = \alpha'$**.** Similar to the previous, we end up with a contradiction again, since something that is not $\hat{\mathfrak{e}}_{\gamma_{n'+1}}$ is ordered below it by $\alpha' \precsim_{n'+1} \alpha''$, but we have $\alpha' \neq \alpha^{*\prime} = \hat{\mathfrak{e}}_{\gamma_{n'}}$.

Finally, we must consider that $\alpha^{*\prime\prime}$ can be $\hat{\mathfrak{e}}_{\gamma_n}$ and $\alpha'' = \top \neq \alpha^{*\prime\prime}$. Then $\overset{*}{\alpha}$ and $\alpha^{*\prime}$ must be fixed points of $\mathcal{D}$ for this case to be any interesting, since otherwise α or α' can be related as equal to α'' through $\top$. So now we have $\alpha'' = \top = \mathcal{D}(\top)$ by (2.40), so then $\alpha^{*\prime} \precsim_{n'} \top$ by (2.45). We may apply the same equation to $\alpha \precsim_{n'+1} \alpha'$, and together with the IH, we get $\overset{*}{\alpha} \precsim_{n'} \top$, and hence by (2.45), we get $\alpha \precsim_{n'+1} \top = \alpha''$. □

A.2 Proof of the $\mathcal{D}$-Iteration Invariant

Let $\mathcal{S}$ be a (non-empty) system of SiDBs with local external potentials and potentials between them that provides our context. Here we prove the invariant property stated in (2.46).

A.2.1 Loop Invariant Properties

Corollary 1: $\forall n \in \mathbb{N}\left(\top \in \gamma_n\right)$**. Proof:** Trivial by $\top \in \gamma_0$ and (2.43). □

Lemma 2: $\forall n \in \mathbb{N}\left(\top \in \mathfrak{F}_S^{\gamma_n}\right)$**. Proof:** By induction on n. Base case: $n = 0$. By (2.40), (2.41) and $\gamma_0 = \{\top\}$ we have $\top \in \mathfrak{F}_S^{\gamma_0}$. Let $n' \in \mathbb{N}$; induction hypothesis: $\top \in \mathfrak{F}_S^{\gamma_{n'}}$. Then, by IH, (2.40), (2.41) and **Corollary 1**, we have $\mathcal{D}\left(\top \in \mathfrak{F}_S^{\gamma_n}\right) = \top \in \mathfrak{F}_S^{\gamma_{n'+1}}$. □

Lemma 3: $\forall n \in \mathbb{N}\left(\mathfrak{F}_S^{\gamma_n} \supseteq \mathfrak{F}_S^{\gamma_{n+1}}\right)$**. Proof:** Let $n \in \mathbb{N}$ and $\alpha \in \mathfrak{F}_S^{\gamma_{n+1}}$; to prove: $\alpha \in \mathfrak{F}_S^{\gamma_n}$. If $\alpha = \top$, then by **Lemma 2** we have $\alpha \in \mathfrak{F}_S^{\gamma_n}$. Now take $\alpha \neq \top$; since $\mathfrak{F}_S^{\gamma_{n+1}}$ is constructed from $\mathfrak{F}_S^{\gamma_n}$ by $\mathcal{D}$, with in particular through (2.40), we have $\alpha = \mathcal{D}\left(\alpha'\right)$ for some $\alpha' \in \mathfrak{F}_S^{\gamma_n}$. Now, taking such α', $\alpha \neq \top$ and (2.40) tell us that $\alpha = \mathcal{D}\left(\alpha'\right) = \alpha' \in \mathfrak{F}_S^{\gamma_n}$. □

Lemma 4: $\forall n \in \mathbb{N}\left(\gamma_n \subseteq \mathfrak{F}_S^{\gamma_n}\right)$**. Proof:** By induction on n. By (2.39) and $\gamma_0 = \{\top\}$ we can form our IH for some $n' \in \mathbb{N}$: $\gamma_{n'} \subseteq \mathfrak{F}_S^{\gamma_{n'}}$. By (2.43) we have that $\gamma_{n'+1} = \gamma_{n'} \cup \left\{\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'}}\right)\right\}$. Let $\alpha \in \gamma_{n'+1}$. We first consider $\alpha \in \gamma_{n'}$. By IH we have that $\alpha \in \mathfrak{F}_S^{\gamma_{n'}}$, and by (2.40) we have that $\mathcal{D}(\alpha)$ is either $\top$, which **Lemma 2** tells us is in $\mathfrak{F}_S^{\gamma_{n'+1}}$, or α. In the latter case, but also in the case that $\alpha = \mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_{n'}}\right)$, we have $\alpha \in \left\{\mathcal{D}\left(\alpha'\right) \mid \alpha' \in \mathfrak{F}_S^{\gamma_{n'}}\right\} = \mathfrak{F}_S^{\gamma_{n'+1}}$. □

Lemma 5: $\forall n \in \mathbb{N}\left(\gamma_n = \mathfrak{F}_S^{\gamma_n} \implies \mathfrak{F}_S^{\gamma_n} = \mathfrak{F}_S^{\gamma_{n+1}}\right)$**. Proof:** Let $n \in \mathbb{N}$ such that $\gamma_n = \mathfrak{F}_S^{\gamma_n}$. It suffices to show that for all $\alpha \in \gamma_n$, $\alpha = \mathcal{D}(\alpha)$ holds. This naturally holds for $\alpha = \top$ by (2.40), so consider $\alpha \neq \top$. Then by (2.41) we have $\alpha \neq \hat{\mathfrak{e}}_{\gamma_n} = \hat{\mathfrak{e}}_{\mathfrak{F}_S^{\gamma_n}} = \top$, so (2.40) tells us $\alpha = \mathcal{D}(\alpha)$. □

Lemma 6: $\forall n \in \mathbb{N}\left(\gamma_n = \mathfrak{F}_S^{\gamma_n} \implies n \geq \left|\mathcal{G}_S\right|\right)$**. Proof:** By induction on n. For $n = 0$ it means that $n \geq \left|\mathcal{G}_S\right| = 0$ by $\gamma_0 = \{\top\}$ and (2.39). Let $n' \in \mathbb{N}$, and let $\gamma_{n'} = \mathfrak{F}_S^{\gamma_{n'}}$; our induction hypothesis now says that $n' \geq \left|\mathcal{G}_S\right|$. Therefore, it suffices to show the premise, i.e.: $\gamma_{n'+1} = \mathfrak{F}_S^{\gamma_{n'+1}}$. By **Lemma 5** we have $\mathfrak{F}_S^{\gamma_{n'}} = \mathfrak{F}_S^{\gamma_{n'+1}}$, and **Lemma 4** and (2.43) then tell us $\gamma_{n'} \subseteq \gamma_{n'+1} \subseteq \mathfrak{F}_S^{\gamma_{n'}}$. □

Lemma 7: $\forall n \in \mathbb{N}\left(\mathfrak{F}_S^{\gamma_n} \setminus \left(\gamma_n \cup \left\{\hat{\mathfrak{e}}_{\gamma_n}\right\}\right) = \mathfrak{F}_S^{\gamma_{n+1}} \setminus \gamma_{n+1}\right)$. **Proof:** By (2.40), we know that for all $n \in \mathbb{N}$,

$$\mathfrak{F}_S^{\gamma_{n+1}} = \left\{\mathcal{D}(\alpha) \mid \alpha \in \mathfrak{F}_S^{\gamma_n}\right\} = \left\{\alpha \mid \alpha \in \mathfrak{F}_S^{\gamma_n} \setminus \left\{\hat{\mathfrak{e}}_{\gamma_n}\right\}\right\} \cup \left\{\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_n}\right)\right\},$$

and since by (2.43), we know $\gamma_{n+1} = \gamma_n \cup \left\{\mathcal{D}\left(\hat{\mathfrak{e}}_{\gamma_n}\right)\right\}$, we can exclude the singleton on either side and get get:

$$\mathfrak{F}_S^{\gamma_{n+1}} \setminus \gamma_{n+1} = \left\{\alpha \mid \alpha \in \mathfrak{F}_S^{\gamma_n} \setminus \left\{\hat{\mathfrak{e}}_{\gamma_n}\right\}\right\} \setminus \gamma_n = \mathfrak{F}_S^{\gamma_n} \setminus \left(\gamma_n \cup \left\{\hat{\mathfrak{e}}_{\gamma_n}\right\}\right).$$

□

Corollary 8: $\forall n \in \mathbb{N}\left(\hat{\mathfrak{e}}_{\gamma_n}\left(\left\{\hat{\mathfrak{e}}_{\gamma_n}\right\}\right) = \hat{\mathfrak{e}}_{\gamma_{n+1}}\right)$. **Proof:** Follows directly from **Lemma 7** and (2.41). □

Proposition 9: $\forall n \in \mathbb{N}, \alpha \in \gamma_n, \alpha' \in \mathfrak{F}_S^{\gamma_n} \setminus \gamma_n\left(\mathfrak{e}_\alpha > \mathfrak{e}_{\alpha'}\right)$.
This can be regarded as 'the small loop invariant'. It asserts that the iteration over n traverses the error rank $\mathfrak{e}_{\alpha \in \mathcal{G}_S}$ downwards.
Proof: We prove it by induction on n. For the base case, let $n = 0$. Then only $\top \in \gamma_0$ and thus $\mathfrak{e}_\top = \infty$ satisfies our goal. Let $n' \in \mathbb{N}$; the induction hypothesis now states that $\forall \alpha \in \gamma_{n'}, \alpha' \in \mathfrak{F}_S^{\gamma_{n'}} \setminus \gamma_{n'}\left(\mathfrak{e}_\alpha > \mathfrak{e}_{\alpha'}\right)$. By **Lemma 7**, (2.43), (2.40) and **Corollary 1**, it suffices to show:

$$\forall \alpha \in \gamma_{n'+1} \subseteq \gamma_{n'} \cup \left\{\hat{\mathfrak{e}}_{\gamma_{n'}}\right\}, \alpha' \in \mathfrak{F}_S^{\gamma_{n'}} \setminus \left(\gamma_{n'} \cup \left\{\hat{\mathfrak{e}}_{\gamma_{n'}}\right\}\right) . \mathfrak{e}_\alpha > \mathfrak{e}_{\alpha'},$$

and by our IH, we only need to consider $\alpha = \hat{\mathfrak{e}}_{\gamma_{n'}} \in \gamma_{n'+1}$. By (2.41) and the unicity requirement of $\mathfrak{e}_{\alpha'}$ for all $\alpha' \in \mathcal{G}_S$ given above (2.46), we know that $\forall \alpha' \in \mathfrak{F}_S^{\gamma_{n'}} \setminus \gamma_{n'}\left(\mathfrak{e}_\alpha > \mathfrak{e}_{\alpha'}\right)$, and thus our goal is satisfied. □

Lemma 10: $\forall n \in \mathbb{N}\left(\hat{\mathfrak{e}}_{\gamma_n} = \vec{\mathfrak{e}}_n\right)$. **Proof:** By strong induction on n. The base case for $n = 0$ is trivial since by (2.39) we have that $\mathfrak{F}_S^{\gamma_0} = \mathcal{G}_S \cup \{\top\}$, of which $\gamma_0 = \{\top\}$ takes off the latter part in (2.41). This leaves us with $\hat{\mathfrak{e}}_{\gamma_0}$ being the $\alpha \in \mathcal{G}_S$ for which $\mathfrak{e}_\alpha$ is maximal, which is $\vec{\mathfrak{e}}_0$ by definition of $\vec{\mathfrak{e}}$ above (2.46). Let $n' \in \mathbb{N}$; our induction hypothesis now says $\forall i \in \mathbb{N}\left(i \leq n' \hat{\mathfrak{e}}_{\gamma_i} = \vec{\mathfrak{e}}_i\right)$, and thus in particular $\hat{\mathfrak{e}}_{\gamma_{n'}} = \vec{\mathfrak{e}}_{n'}$. Let $A = \mathfrak{F}_S^{\gamma_{n'}} \setminus \gamma_{n'}$. Then $\forall \alpha \in A\left(\mathfrak{e}_\alpha < \mathfrak{e}_{\vec{\mathfrak{e}}_{n'}}\right)$ by (2.41) and the IH, and uniqueness of $\mathfrak{e}_\alpha$ for all $\alpha \in \mathcal{G}_S$. By **Corollary 8**, then, we have $\hat{\mathfrak{e}}_{\gamma_{n+1}} = \hat{\mathfrak{e}}_{\gamma_n}\left(\left\{\vec{\mathfrak{e}}_{n'}\right\}\right)$. From this, (2.41) and the aforementioned uniqueness property, it follows either one of two things hold: $\hat{\mathfrak{e}}_{\gamma_{n+1}} = \top$, or $\hat{\mathfrak{e}}_{\gamma_{n+1}} \in A$ and $\hat{\mathfrak{e}}_{\gamma_{n+1}} \neq \vec{\mathfrak{e}}_{n'}$ and $\forall \alpha \in A\left(\mathfrak{e}_\alpha < \mathfrak{e}_{\hat{\mathfrak{e}}_{\gamma_{n+1}}} < \mathfrak{e}_{\vec{\mathfrak{e}}_{n'}}\right)$. In the former case, it must be that $\gamma_{n'+1} = \mathfrak{F}_S^{\gamma_{n'+1}}$ by (2.41). Then, by **Lemma 6**, we have $n' + 1 \geq \left|\mathcal{G}_S\right|$, and thus $\vec{\mathfrak{e}}_{n'+1} = \top = \hat{\mathfrak{e}}_{\gamma_{n'+1}}$.

For the latter case, take $n'+2 < \left|\mathscr{G}_S\right|$, since $\top \neq \hat{\mathbf{e}}_{\gamma_{n'+1}}$ and $\mathbf{e}_{\hat{\mathbf{e}}_{\gamma_{n'+1}}} < \mathbf{e}_{\vec{\mathbf{e}}_{n'}}$ otherwise implies $\hat{\mathbf{e}}_{\gamma_{n'+1}} = \vec{\mathbf{e}}_{n'+1}$ directly by definition of $\vec{\mathbf{e}}$. It is now left to show $\mathbf{e}_{\vec{\mathbf{e}}_{n'+2}} < \mathbf{e}_{\hat{\mathbf{e}}_{\gamma_{n'+1}}}$, that is, $\vec{\mathbf{e}}_{n'+2} \in A$. From our strong induction hypothesis, we know that for all $i \in \mathbb{N}$ with $i \leq n'$, we have $\hat{\mathbf{e}}_{\gamma_i} = \vec{\mathbf{e}}_i$, and therefore by (2.43) we know that $\mathbf{e}_{\vec{\mathbf{e}}_{n'+2}} \notin \gamma_{n'}$ since our states are unique. By the same reasoning, now using (2.40), we can see that $\mathbf{e}_{\vec{\mathbf{e}}_{n'+2}} \in \mathscr{G}_S \supseteq \mathfrak{F}_S^{\gamma_{n'}}$ cannot have been transformed to $\top$ in any of the n' applications of $\mathcal{D}$ to $\mathbf{e}_{\vec{\mathbf{e}}_{n'+2}}$, hence it must be in A. □

Proposition 11:

$$\forall \alpha \in \mathscr{G}_S, n', n \in \mathbb{N}\Big(0 < n < n' \wedge \hat{\mathbf{e}}_{\gamma_n} \in \gamma_{n'} \setminus \{\top\} \wedge \alpha \prec_n \hat{\mathbf{e}}_{\gamma_n} \implies \exists i \in \mathbb{N}$$

$$\Big(i < n \wedge \vec{\mathbf{e}}_i = \alpha = \mathcal{D}\left(\alpha \in \mathfrak{F}_S^{\gamma_i}\right) \wedge \forall i \leq j < n\,.\,\vec{\mathbf{e}}_j \in \gamma_{n'} \subseteq \mathfrak{F}_S^{\gamma_{n'}} \wedge \vec{\mathbf{e}}_j \prec_n \hat{\mathbf{e}}_{\gamma_n}\Big)\Big).$$

With this proposition, we have a means of translation from statements about the irreflexive and transitive ordering $\prec$, extracting transitively entailed relations as well as associated denotations in our formal system.
Proof: Take any $\alpha \in \mathscr{G}_S$ and some $n' \in \mathbb{N}$ such that $n' \geq 2$; we prove the proposition by induction on $n \in \mathbb{N}$. For the base case, let $n = 1 < n'$ such that $\hat{\mathbf{e}}_{\gamma_1} \in \gamma_{n'} \setminus \{\top\}$ and $\alpha \prec_1 \hat{\mathbf{e}}_{\gamma_1}$. There is only one $i \in \mathbb{N}$ below $n = 1$, so take $i = 0$; we want $\hat{\mathbf{e}}_{\gamma_1} \succ_1 \vec{\mathbf{e}}_0 = \alpha = \mathcal{D}\left(\alpha \in \mathfrak{F}_S^{\gamma_0}\right) \in \gamma_{n'} \subseteq \mathfrak{F}_S^{\gamma_{n'}}$. This holds by considering that (2.44) must apply to our assumption, and thus $\hat{\mathbf{e}}_{\gamma_0} = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_0}\right) = \vec{\mathbf{e}}_0$ by **Lemma 10**. By (2.43), we get then that $\vec{\mathbf{e}}_0 \in \gamma_1 \subseteq \gamma_{n'}$ and the rest follows by **Lemma 4**.

Now let $0 < m \in \mathbb{N}$ and assume our induction hypothesis to hold for m. We want to prove our proposition for $m+1$, so we assume $m < n' - 1$. Further, we assume $\top \neq \hat{\mathbf{e}}_{\gamma_{m+1}} \in \gamma_{n'}$ and $\alpha \prec_{m+1} \hat{\mathbf{e}}_{\gamma_{m+1}}$. From this we find by (2.44) that $\alpha \precsim_m \hat{\mathbf{e}}_{\gamma_m} = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_m}\right)$ since $\alpha \neq \top$ which means that $\alpha = \mathcal{D}\left(\alpha \in \mathfrak{F}_S^{\gamma_{m-1}}\right)$ by (2.40). Notice also that we get $\hat{\mathbf{e}}_{\gamma_m} = \vec{\mathbf{e}}_m \in \gamma_{m+1} \subseteq \gamma_{n'} \subseteq \mathfrak{F}_S^{\gamma_{n'}}$ by **Lemma 10**, (2.43), and **Lemma 4** together with $m+1 \leq n'$. Hence, since naturally $m < m+1$, we satisfy our goal when assuming $\alpha = \vec{\mathbf{e}}_m$, so instead assume $\alpha \prec_m \hat{\mathbf{e}}_{\gamma_m}$.

Since $\top \neq \hat{\mathbf{e}}_{\gamma_m} = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_m}\right) \in \gamma_{m+1} \subseteq \gamma_{n'}$ by (2.43) and $m+1 \leq n'$, we can apply our induction hypothesis. This produces an $i \in \mathbb{N}$ with $i < m < m+1$ for which $\vec{\mathbf{e}}_i = \alpha = \mathcal{D}\left(\alpha \in \mathfrak{F}_S^{\gamma_i}\right)$ and, for all $i \leq j < m$, we have $\vec{\mathbf{e}}_j \in \gamma_{n'} \subseteq \mathfrak{F}_S^{\gamma_{n'}}$, and $\vec{\mathbf{e}}_j \prec_m \hat{\mathbf{e}}_{\gamma_m}$, which implies $\vec{\mathbf{e}}_j \prec_{m+1} \hat{\mathbf{e}}_{\gamma_m}$ by (2.45). Then by transitivity of $\prec$, we get $\alpha = \vec{\mathbf{e}}_i \prec_{m+1} \hat{\mathbf{e}}_{\gamma_m} \prec_{m+1} \hat{\mathbf{e}}_{\gamma_{m+1}}$. Finally, when taken together which what we know about $\hat{\mathbf{e}}_{\gamma_m}$, the required properties are now shown for all $i \leq j \leq m$. □

A.2.2 Loop Invariant Theorem

$$\forall n \in \mathbb{N} \,.\, \top \in \gamma_n \wedge \forall \alpha \in \gamma_n \setminus \{\top\} \,.\, \exists i \in \mathbb{N} \,.\, i < n \wedge \alpha = \vec{\mathbf{e}}_i \wedge$$
$$\mathsf{prob}_\alpha \left\{ \vec{\mathbf{e}}_{i'} \mid i' \in \mathbb{N}, i' = i+1 \vee \left(i' \leq i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n \right) \right\} \geq P^{\min}$$

Proof

Starting off, we can separate the first part of the invariant from the rest, i.e. $\forall n \in \mathbb{N} \left(\top \in \gamma_n \right)$, which is satisfied by **Corollary 1**. Now let $n \in \mathbb{N}$ and $\alpha \in \gamma_n$ such that $\top \not\sim \alpha$, i.e., $\alpha \neq \top$ by reflexivity of the preorder. Then, by (2.43) there must be some $i \in \mathbb{N}$ with $0 < i < n$ such that $\alpha = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_i} \right) = \hat{\mathbf{e}}_{\gamma_i}$. **Lemma 10**, then tells us $\alpha = \vec{\mathbf{e}}_i$, but also, by (2.40) it means that $\mathsf{Relevant}_{\gamma_i}(\alpha)$, and thus, by (2.42) we have $\mathsf{prob}_\alpha \left\{ \alpha' \in \mathfrak{F}_S^{\gamma_i} \;\middle|\; \alpha' \precsim_i \alpha \vee \alpha' = \hat{\mathbf{e}}_{\gamma_i}(\{\alpha\}) \right\} \geq P^{\min}$. Thereby, it suffices to show that:

$$\left\{ \alpha' \in \mathfrak{F}_S^{\gamma_i} \;\middle|\; \alpha' \precsim_i \alpha \vee \alpha' = \hat{\mathbf{e}}_{\gamma_i}(\{\alpha\}) \right\}$$
$$= \left\{ \vec{\mathbf{e}}_{i'} \mid i' \in \mathbb{N}, i' = i+1 \vee \left(i' \leq i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n \right) \right\}.$$

By applying **Corollary 8** and **Lemma 10** in respective order and using the fact that $\alpha = \hat{\mathbf{e}}_{\gamma_i}$, we can show that $\hat{\mathbf{e}}_{\gamma_i}(\{\alpha\}) = \vec{\mathbf{e}}_{i+1}$. Furthermore, we have $\vec{\mathbf{e}}_i = \alpha \precsim_i \alpha \in \gamma_{i+1} \subseteq \gamma_i \subseteq \mathfrak{F}_S^{\gamma_i}$ by reflexivity, (2.43) and **Lemma 4**, thus it is left to prove $\left\{ \alpha' \in \mathfrak{F}_S^{\gamma_i} \mid \alpha' \prec_i \alpha \right\} = \left\{ \vec{\mathbf{e}}_{i'} \mid i' \in \mathbb{N}, i' < i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n \right\}$.

Let A_k denote $\left\{ \alpha' \in \mathfrak{F}_S^{\gamma_k} \;\middle|\; \alpha' \prec_k \hat{\mathbf{e}}_{\gamma_k} \right\}$ for all $k \in \mathbb{N}$. We extend this notation to the following in order to talk about *slices* of this set, where the size of the slice grows 'backwards' from k as the origin: for all $m \in \mathbb{N}$ with $m < k$, we say that 'A_k up to m' is defined as $A_k \uparrow m \coloneqq A_k \cap B_{k,m}$, where $B_{k,m} \coloneqq \left\{ \vec{\mathbf{e}}_j \mid k - m - 1 \leq j < k \right\}$. Let us confirm that $A_i = A_i \uparrow i - 1$, i.e. $\forall \alpha' \in \mathfrak{F}_S^{\gamma_i} \left(\alpha' \prec_i \alpha \implies \alpha' \in B_{i,i-1} = \left\{ \vec{\mathbf{e}}_j \mid 0 \leq j < i \right\} \right)$: let such α' such that $\alpha' \prec_i \alpha = \hat{\mathbf{e}}_{\gamma_i}$. Then by **Proposition 11** we have $\alpha' = \vec{\mathbf{e}}_j$ for some $j \in \mathbb{N}$ and $j < i$ so $\alpha' \in B_{i,i-1}$. Therefore, it suffices to prove the following by induction on m, since our goal is proven for $m = i - 1$:

$$\forall m \in \mathbb{N} \,.\, m < i \implies A_i \uparrow m = \left\{ \vec{\mathbf{e}}_{i'} \mid i - m - 1 \leq i' < i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n \right\}.$$

For the base case, take $m = 0 < i$. We then are required to prove the following equality: $\left\{ \vec{\mathbf{e}}_{i-1} \mid \vec{\mathbf{e}}_{i-1} \in \mathfrak{F}_S^{\gamma_i} \wedge \vec{\mathbf{e}}_{i-1} \prec_i \alpha \right\} = \left\{ \vec{\mathbf{e}}_{i-1} \mid \vec{\mathbf{e}}_{i-1} \in \gamma_n \right\}$. By **Proposition 11** and $i < n$, we have that the first side of the implicit biconditional (left to right) is shown. Now, for the other direction, assume $\vec{\mathbf{e}}_{i-1} \in \gamma_n$. Since $\vec{\mathbf{e}}_{i-1} \neq \vec{\mathbf{e}}_i \neq \top$, then also, by **Lemma 10**, (2.43) and reflexivity, $\hat{\mathbf{e}}_{\gamma_{i-1}} = \mathcal{D}\left(\hat{\mathbf{e}}_{\gamma_{i-1}} \right) = \vec{\mathbf{e}}_{i-1} \precsim_{i-1} \hat{\mathbf{e}}_{\gamma_{i-1}}$ so then by (2.44) we get $\vec{\mathbf{e}}_{i-1} \prec_i \hat{\mathbf{e}}_{\gamma_i} = \alpha$, and further $\vec{\mathbf{e}}_{i-1} \in \gamma_i \subseteq \mathfrak{F}_S^{\gamma_i}$ by (2.43) and **Lemma 4**.

For the inductive case, let $m' \in \mathbb{N}$ and assume the above expression holds for m' as our induction hypothesis. We now prove it holds for $m' + 1$, so we

assume $m' < i - 1$ since otherwise the implication would hold vacuously. Let $C = \left\{\vec{\mathbf{e}}_{i'} \mid i - m' - 1 \leq i' < i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n\right\}$. Now we get:

$$\begin{aligned}
&\left\{\vec{\mathbf{e}}_{i'} \mid i - m' - 2 \leq i' < i \wedge \forall i' \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n\right\} \\
&= C \cup \left\{\vec{\mathbf{e}}_{i-m'-2} \mid \vec{\mathbf{e}}_{i-m'-2} \in \gamma_n \wedge \forall i - m' - 1 \leq j < i \,.\, \vec{\mathbf{e}}_j \in \gamma_n\right\} \\
&= C \cup \left\{\vec{\mathbf{e}}_{i-m'-2} \mid \vec{\mathbf{e}}_{i-m'-2} \in \gamma_n \wedge |C| = m' + 1\right\} \\
&= A_i \uparrow m' \cup \left\{\vec{\mathbf{e}}_{i-m'-2} \mid \vec{\mathbf{e}}_{i-m'-2} \in \gamma_n \wedge \left|A_i \uparrow m'\right| = m' + 1\right\}
\end{aligned}$$

by our induction hypothesis. Rewriting the other side, we get:

$$\begin{aligned}
A_i \uparrow m' + 1 &= A_i \uparrow m' \cup \left(A_i \cap \left\{\vec{\mathbf{e}}_{i-m'-2}\right\}\right) \\
&= A_i \uparrow m' \cup \left\{\vec{\mathbf{e}}_{i-m'-2} \mid \vec{\mathbf{e}}_{i-m'-2} \prec_i \alpha\right\},
\end{aligned}$$

where $\vec{\mathbf{e}}_{i-m'-2} \in \mathfrak{F}_S^{\gamma_i}$ may be left out by **Proposition 11**. Therefore, it now suffices to show $\vec{\mathbf{e}}_{i-m'-2} \prec_i \alpha \iff \vec{\mathbf{e}}_{i-m'-2} \in \gamma_n \wedge \left|A_i \uparrow m'\right| = m' + 1$.

We first prove $\Longrightarrow$, i.e. let $\vec{\mathbf{e}}_{i-m'-2} \prec_i \alpha$. Then by **Proposition 11** and the unicity of $\vec{\mathbf{e}}_j$ for all $j \in \mathbb{N}$, $j < \left|\mathscr{G}_S\right|$ by definition of $\vec{\mathbf{e}}$ and $\left|\mathscr{G}_S\right| = 3^{|S|}$, we get that $\vec{\mathbf{e}}_{i-m'-2} \in \gamma_n$ by $i < n$ and (2.43), and since $\forall i' - m' - 2 \leq j < i$, we have $\vec{\mathbf{e}}_j \in \mathfrak{F}_S^{\gamma_i}$ and $\vec{\mathbf{e}}_j \prec_i \alpha$. Therefore, $A_i \uparrow m'$, which can have at most $m' + 1$ elements, has $m' + 1$ elements for which the condition to be contained in it is satisfied, hence $|A_i \uparrow m'| = m' + 1$.

Lastly, we prove $\Longleftarrow$; so let $\vec{\mathbf{e}}_{i-m'-2} \in \gamma_n$ and $\left|A_i \uparrow m'\right| = m' + 1$. From the former and $i - m' - 2 < \left|\mathscr{G}_S\right|$, we can deduce by **Lemma 10** and (2.43) that $\mathcal{D}\left(\vec{\mathbf{e}}_{i-m'-2} \in \mathfrak{F}_S^{\gamma_{i-m'-2}}\right) = \vec{\mathbf{e}}_{i-m'-2} \precsim_{i-m'-2} \hat{\mathbf{e}}_{\gamma_{i-m'-2}}$ by reflexivity, and thus (2.44) tells us that then $\vec{\mathbf{e}}_{i-m'-2} \prec_{i-m'-2} \hat{\mathbf{e}}_{\gamma_{i-m'-1}} = \vec{\mathbf{e}}_{i-m'-1}$ from which it also follows that $\vec{\mathbf{e}}_{i-m'-2} \prec_i \vec{\mathbf{e}}_{i-m'-1}$ by (2.45). Furthermore, from $\left|A_i \uparrow m'\right| = m' + 1$ and the general fact that $\left|A_i \uparrow m'\right| \leq m' + 1$ and again the unicity argument, we have in particular, by **Proposition 11**, that $\vec{\mathbf{e}}_{i-m'-1} \prec_i \alpha$. Now, by transitivity of $\prec_i$, we get our goal $\vec{\mathbf{e}}_{i-m'-2} \prec_i \alpha$, thus concluding the proof. □

Appendix B

ClusterComplete Simulations of 83-DB Crossover Layouts

As a demonstration of the shift in computational intractability, physical simulations were carried out using *ClusterComplete* in base 3 on a crossover gate extended with input and output ports, accumulating to 5 tiles from the *Bestagon* gate library in connection. [34] The layout was synthesised within *fiction* for each of the four input combinations. [35] The input ports are the two tiles at the top of each layout, connecting to the crossover gate, which connects to two output perturber-terminated output ports on the bottom of each layout.

ClusterComplete was invoked on a system with a i7 4770k @ 3.50 GHz and 16 GB of 1666 MHz (DDR3) RAM. The respective ground states are shown in Figure B.1 through Figure B.4. It can be observed that only the **00** input combination is *technically* operational, as observed at the output BDL pairs next to the respective output perturbers, though BDL wire kinks are seen in the respective ground states of each input combination (see [25]).

Input	Runtime	Ground State Energy	No. Metastable States
00	$1.96 \cdot 10^5$ s	3.6227 eV	1067
01	$1.67 \cdot 10^5$ s	3.6823 eV	1068
10	$1.54 \cdot 10^5$ s	3.6675 eV	1299
11	$2.03 \cdot 10^5$ s	3.7272 eV	1299

Table B.1: *ClusterComplete* simulation statistics for the SiDB logic layouts in Figure B.1 through Figure B.4. Each of these layouts consists of 83 SiDBs, and hence there are $3^{83} = 3.9908384 \cdot 10^{39}$ global states in total, which would all be enumerated if *ExGS* were to be invoked. This total number of global states is approximately equal to $6 \cdot \sqrt{\text{no. physical bits of information in the observable universe.}}$ [31]

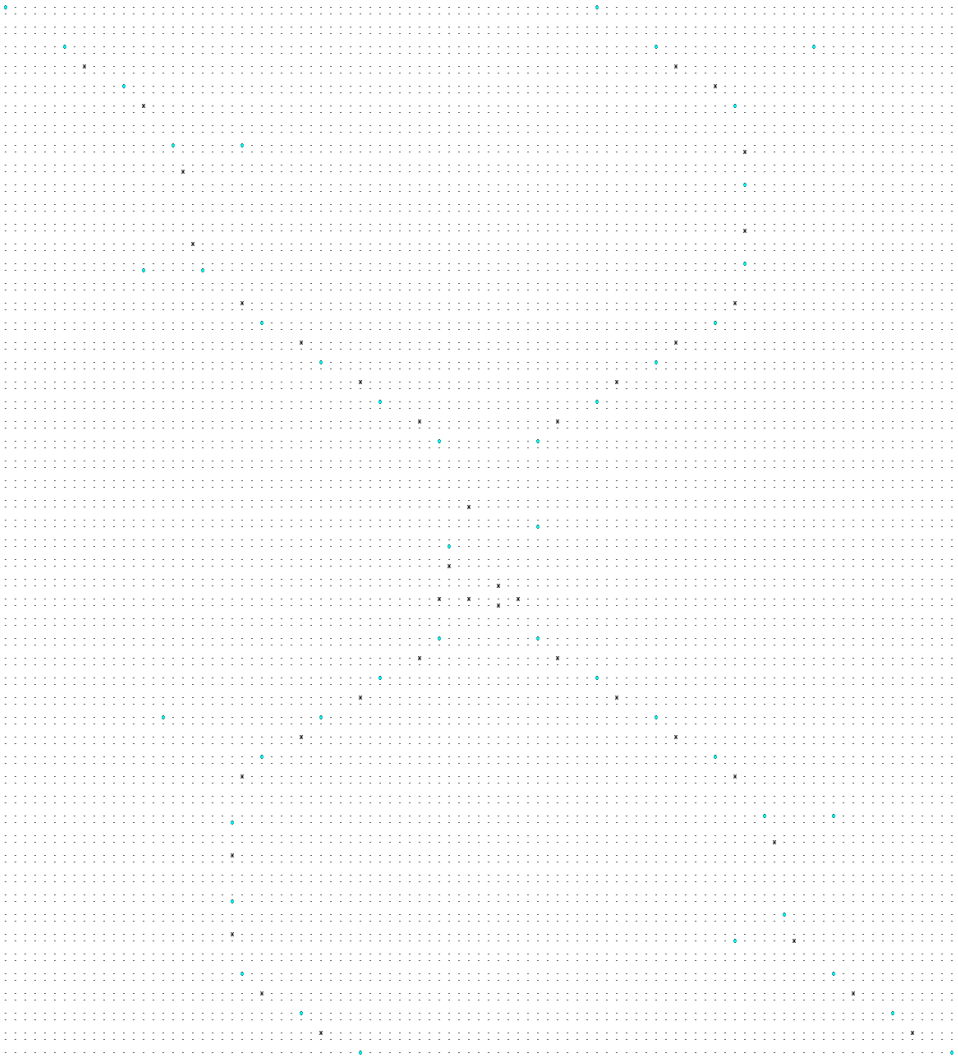


Figure B.1: Ground state for the **00** input combination.

Runtime	Ground State Energy	No. Metastable States
54 h 26 m 49 s	3.6227 eV	1067

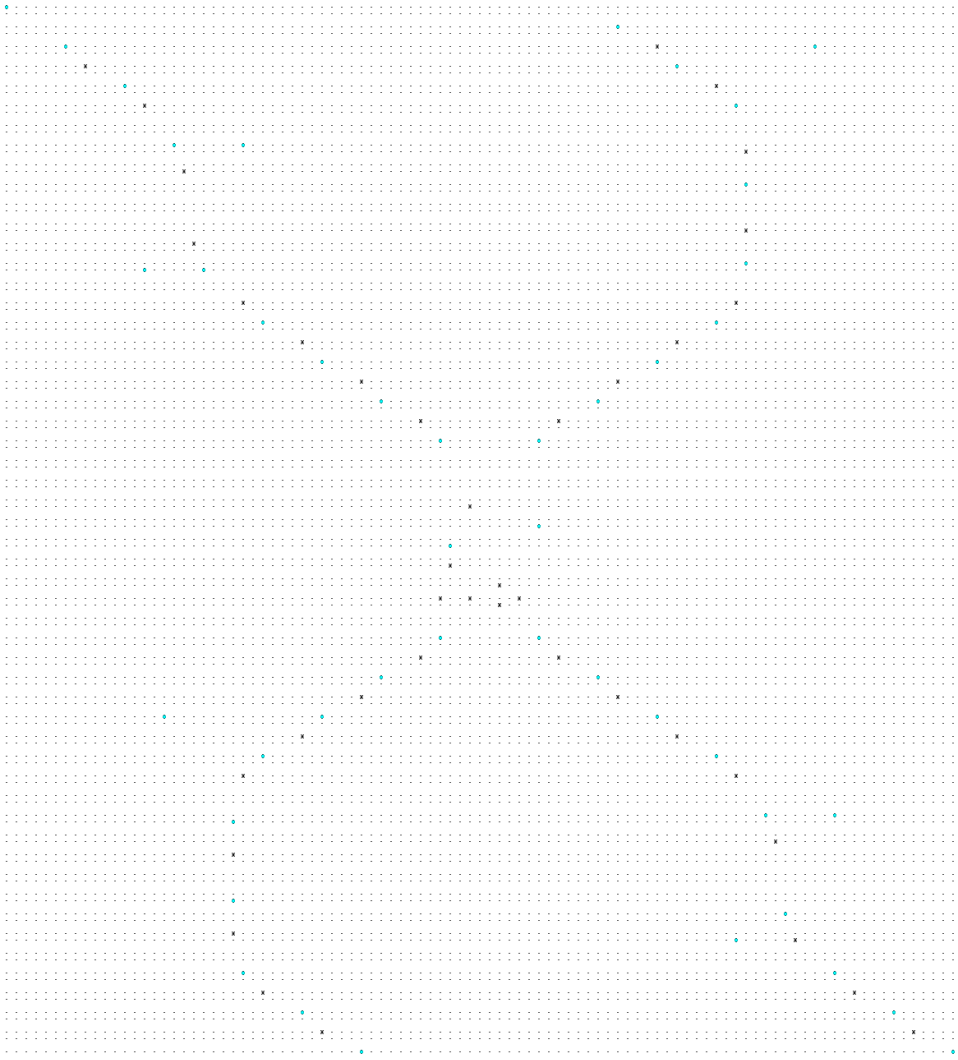


Figure B.2: Ground state for the **01** input combination.

Runtime	Ground State Energy	No. Metastable States
46 h 17 m 3 s	3.6823 eV	1068

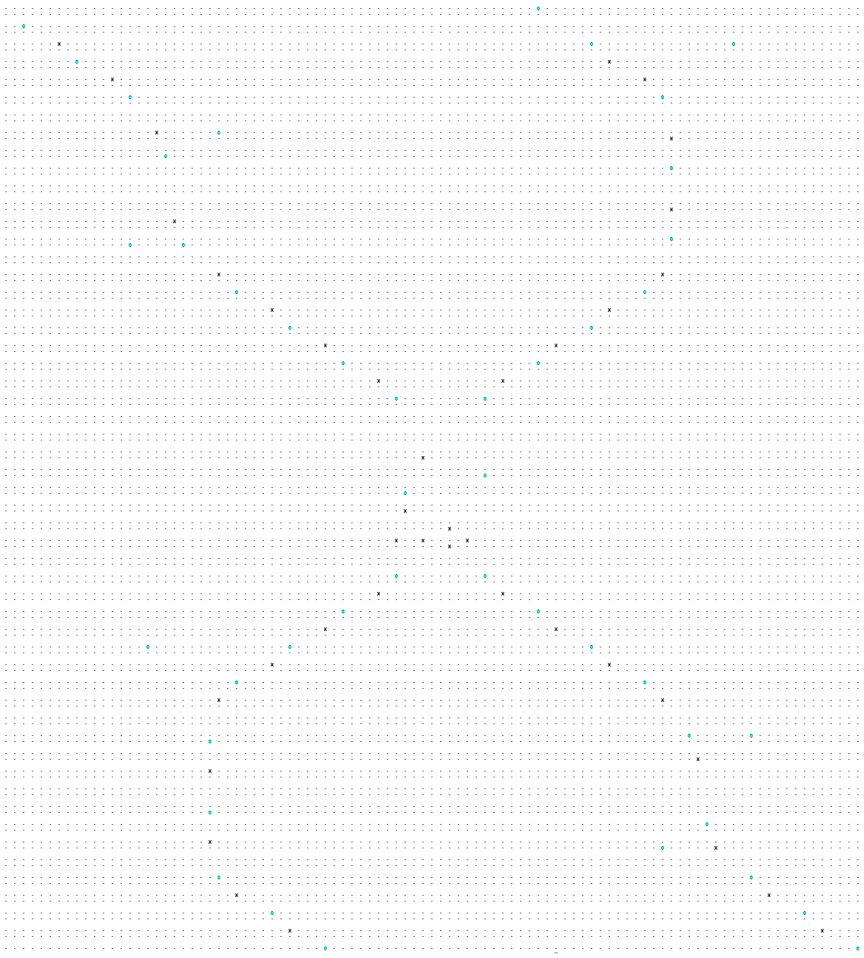


Figure B.3: Ground state for the **10** input combination.

Runtime	Ground State Energy	No. Metastable States
42 h 48 m 43 s	3.6675 eV	1299

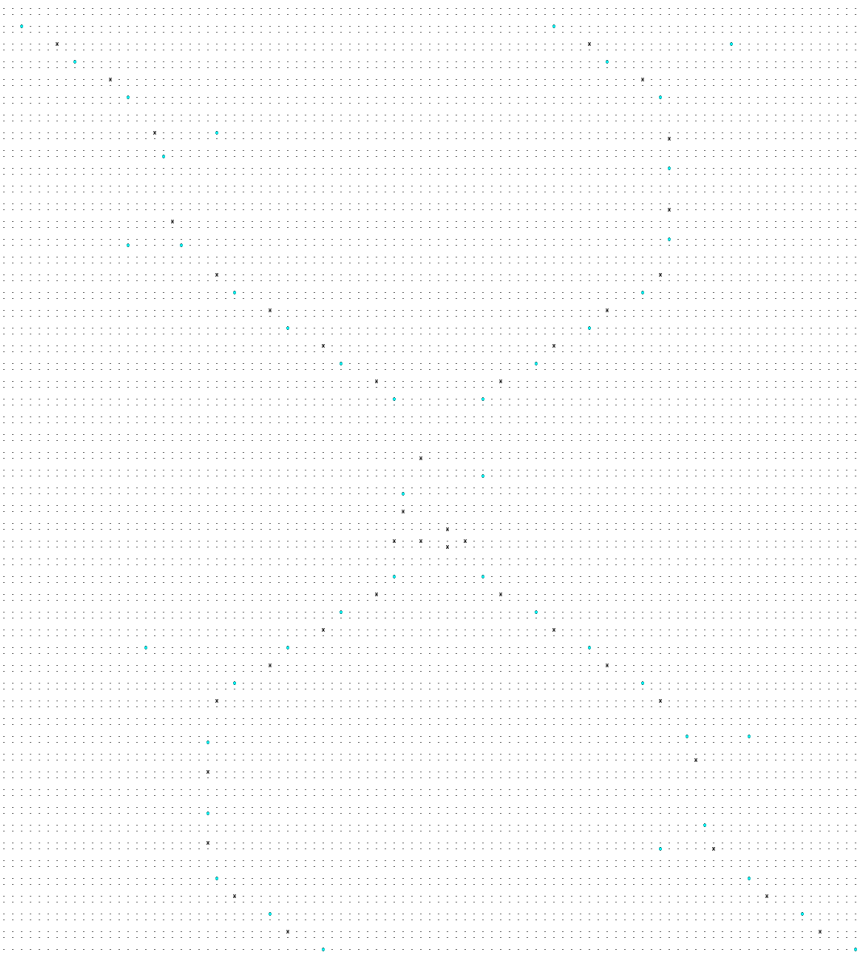


Figure B.4: Ground state for the **11** input combination.

Runtime	Ground State Energy	No. Metastable States
56 h 24 m 44 s	3.7272 eV	1299

Appendix C

Validity Witness Partitioning

Herein we describe a more elaborate procedure for evaluating $\mathscr{T}$, originally defined in (3.10), to account for variables that witness validity for multiple states and thereby impair pruning efficacy. The problem is easily understood with the example given below, after which a simple solution is proposed that is bounded by a worst-case time complexity that is factorial in the number of overlapping witnesses; the approach comes down to a brute-force of all permutations.

Example 3. Let $p \in \mathscr{P}$ such that $p_{\mathfrak{c}}^{\dagger} = \{v, v'\}$ and $p_{\mathfrak{m}} = \{\!\{z, z'\}\!\}$ with $v \neq v'$ and $z \neq z'$, and $\rho_{p_{\mathfrak{c}}}$ and ξ are such that:

- $\left(\rho_{p_{\mathfrak{c}}}(v) \mathbin{\overline{\underline{\oplus}}} \xi(v)\right) \cap \left(\tau_v(z) \cup \tau_v\left(z'\right)\right) = \varnothing$,
- $\left(\rho_{p_{\mathfrak{c}}}\left(v'\right) \mathbin{\overline{\underline{\oplus}}} \xi\left(v'\right)\right) \cap \tau_{v'}(z) \neq \varnothing$, and
- $\left(\rho_{p_{\mathfrak{c}}}\left(v'\right) \mathbin{\overline{\underline{\oplus}}} \xi\left(v'\right)\right) \cap \tau_{v'}\left(z'\right) \neq \varnothing$.

Then $\mathscr{T}(p, \xi)$ accepts, since it counts one witness for z and z' each, satisfying the respective counts $p_{\mathfrak{m}}(z)$ and $p_{\mathfrak{m}}(z')$, while v does not accept either state here. Thereby, we have the information required to deny $\mathscr{T}(p, \xi)$, but this is not used in (3.10).

To enable stricter pruning, we will create sets of witnesses for each state in the given multiset m that we will then attempt to *partition*. If no partition can be found that assigns $m(z)$ witnesses to each $z \in m$, then m can be pruned. More precisely, let $p \in \mathscr{P}$ and $\xi : p_{\mathfrak{c}}^{\dagger} \to \mathcal{P}(W)$, and for all $z \in p_{\mathfrak{m}}$ let $\mathscr{W}_z \coloneqq \left\{ v \in p_{\mathfrak{c}}^{\dagger} \mid \left(\rho_{p_{\mathfrak{c}}}(v) \cup \xi(v)\right) \cap \tau_v(z) \neq \varnothing \right\}$ be the set of witnesses for z. Then (3.10) transcribes to $\mathscr{T}(p, \xi) \overset{\text{def}}{\iff} \forall z \in p_{\mathfrak{m}} \,.\, p_{\mathfrak{m}}(z) \leq \left|A_z\right|$. Additionally, however, we may form, beside (3.12), a secondary pruning objective:

$$\mathcal{S},\ \mathscr{T}\left(p, \pi_p\right) \vdash \neg \mathsf{Partitionable}\left(p_{\mathfrak{m}}, \left(\mathscr{W}_z\right)_{z \in p_{\mathfrak{m}}}\right) \iff p\ \mathfrak{pruned}.$$

Algorithm C.1: $\mathsf{Partitionable}_{\mathscr{Z}}$ for $\mathscr{Z} = \left(z_i\right)_{1 \le i \le n}$

Input: $(m, \mathscr{W}, i, n \ge 1)$, where $m \in \mathcal{M}(\mathcal{Z})$, $\mathscr{W} = \left(\mathscr{W}_z \subseteq V\right)_{z \in m}$ for some $V \subseteq \mathcal{V}$ with $|V| = |m|$, and $1 \le i \le n$ with $n \in \mathbb{N}$
Output: m is partitionable or not

if $i = n$ **then**
  return $\left|\mathscr{W}_{z_n}\right| \ge m\left(z_i\right)$
if $m\left(z_i\right) = 0$ **then**
  return $\mathsf{Partitionable}_{\mathscr{Z}}(m, \mathscr{W}, i+1, n)$
foreach $v \in \mathscr{W}_{z_i}$ **do**
  $m' \leftarrow m - \left\{\!\left\{z_i\right\}\!\right\}$;
  $\mathscr{W}' = \left(\mathscr{W}'_z\right)_{z \in m} \leftarrow \mathscr{W}$;
  foreach $i \le j \le n$ **do**
    $\mathscr{W}'_{z_j} \leftarrow \mathscr{W}'_{z_j} \setminus \{v\}$
  if $\mathsf{Partitionable}_{\mathscr{Z}}\left(m', \mathscr{W}', i, n\right)$ **then**
    return true
return false

Algorithm C.2:
$\mathsf{Partitionable} : \mathcal{P}\left(\left\{\left(m, \left(\mathscr{W}_z \subseteq V\right)_{z \in m}\right) \mid m \in \mathcal{M}(\mathcal{Z}),\ V \subseteq \mathcal{V},\ |V| = |m|\right\}\right)$

Input: m, $\mathscr{W} = \left(\mathscr{W}_z\right)_{z \in m}$
Output: m is partitionable or not

foreach $z \in m$ **do**
  $\mathsf{UniqueWitnesses} \leftarrow \mathscr{W}_z \setminus \bigcup_{z' \in m,\ z' \ne z} \mathscr{W}_{z'}$;
  $m(z) \leftarrow m(z) - |\mathsf{UniqueWitnesses}|$;
  $\mathscr{W}_z \leftarrow \mathscr{W}_z \setminus \mathsf{UniqueWitnesses}$
if $m = \varnothing$ **then**
  return true
$n \leftarrow |\mathrm{dom}(m)|$;
$\mathscr{Z} \leftarrow \left(z_i\right)_{1 \le i \le n}$ where $\left\{z_i \mid 1 \le i \le n\right\} = \{z \in m\}$;
return $\mathsf{Partitionable}_{\mathscr{Z}}(m, \mathscr{W}, 1, n)$

Observing the structure of Algorithm C.1, we can bound the time-complexity for solving $\mathsf{Partitionable}\left(p_{\mathfrak{m}}, \left(\mathscr{W}_z\right)_{z \in p_{\mathfrak{m}}}\right)$ by a factorial:

$$\mathcal{O}\left(\left|\bigcup_{z \in m}\left(\mathscr{W}_z \setminus \left(\mathscr{W}_z \setminus \bigcup_{z' \in m,\ z' \ne z} \mathscr{W}_{z'}\right)\right)\right|!\right) = \mathcal{O}\left(\left|\bigcap_{z \in m} \mathscr{W}_z\right|!\right).$$

Post Scriptum:
The Semantics of Life

In each unitary moment in time, the happiness or fulfilment of a perceiver is evaluated, given the perceived state of all the things in the universe in that moment. Rather than the resulting measure of fulfilment viewed as a gradient scale, we can observe it as a binary state; either the subject is fulfilled at the given moment, or there are things conditional to this state.

The typical case is the latter. All phenomena in consciousness are taken into account, and evaluated to their degree of contributing to happiness. Essentially, a list is iterated in which the leading items are of more prominent or immediate importance in their contribution to the current state than those that follow. Take for example the following sequence of concerns that may arise for a particular subject at a particular time upon the self-assessment "how am I doing?":

Do I have enough breath?
Do I have the ability to stand and walk?
Do I experience cramps or other uncomfortable pains in my body?
Am I hungry?
Did I get enough sleep last night?
Do I have to cycle through pouring rain?
What is my relation with my bad habits?
How is my love life?
Am I in good physical shape?
How do I feel about my career?
Do I like the view from my apartment?
Is my sister still mad at me from our argument last week?
...

Considering the fleeting nature of desires, the ordering of these phenomena may change with each instant, as for instance the breath may be of less concern with oxygen-filled lungs, and the thought of starvation will subside upon satiation at the dinner table. The structure of a preordered set fits nicely here, since, most importantly, the contribution of phenomena to the current

state of being may be ordered in a non-strictness manner, in which a natural transitive property emerges. Furthermore, without the requirement of equality for two phenomena that are equally ordered, this produces a framework in which such items in the checklist can be trialed in non-deterministic order.

Indeed, the word "checklist" implies that each item on the list is a binary assessment, since for each open question there is a possibly (uncountably) infinite set of yes/no questions upon introduction of transcendental limits for each point on the n-dimensional gradient associated with the original question. Therefore, the question of fulfilment becomes an inquiry for the satisfaction of all binary requirements for fulfilment itself.

To be more descriptive, we define the function of existence. In each infinitesimally brief moment in the existence of a perceiver, a function that is henceforth referred to as R is given the set of all things in current perception, and produces a subset with a certain preorder associated with the instant. This allows us to give rise to the evasive semantics of true happiness and, dually, weltschmertz. Formally, the denotations are as follows: for set P of phenomena in perception at any transcendental moment in time:

An experiencer assumes the state of fulfilment if and only if $R(P) = \emptyset$.

And on the other side:

A subject experiences weltsmertz if and only if $R(P) \neq \emptyset$.

From these formalisms we can extract the following characterisations:

Bliss is characterised by the vacuity of its requirement, and dually, weltschmertz is characterised by the non-vacuity of the requirements of fulfilment.

Then what about an instant in which a subject is dealt with a P that can be characterised as life-threatening? Should then a subset of P not be included in R(P)? While such a subset ought to be respected if existence in this physical realm is to be prolonged, one need not suffer in reaction. Observe the distinction between problem, and cause of suffering.

Furthermore, if the inclusive boundary into non-existence is crossed, we have two interpretations: either the experiencer has ceased to exist, and the function of existence can not be executed whereby the statements above the hold vacuously, or we interpret that $P = \emptyset$ for all that can be conceived as time in the after, of which the only subset is $\emptyset$. Therefore $R(\emptyset) = \emptyset$ into infinity, implying eternal fulfilment beyond the crossing.