

Ending the Tyranny of the Clock: SAT-based Clock Number Assignment for Field-coupled Nanotechnologies

Marcel Walter, Jan Drewniok, and Robert Wille

<https://www.cda.cit.tum.de/research/nanotech/>

Abstract—This paper presents an algorithm for clock number assignment in the physical design process of *Field-coupled Nanocomputing* (FCN), a set of promising beyond-CMOS technologies that manipulate physical fields instead of electrical currents for computation and information transmission. Clocking has traditionally been a significant obstacle to the scalability of FCN physical design algorithms, requiring pre-defined clocking schemes that limit the quality of circuit layouts and add complexity to the design process. Our proposed method utilizes *Boolean Satisfiability* (SAT) solving to facilitate the assignment of clock numbers without the need for predefined clocking, while ensuring compliance with technological constraints on information flow and synchronization.

Via an experimental evaluation, we confirm the proposed algorithm’s *versatility* to reconstruct clock assignments for diverse clocking schemes in reasonable runtime, and its *scalability* up to layouts with a half-million tiles. Thereby, we are potentially paving the way for a new era of physical design algorithms that are not constrained by the limitations of predefined clocking schemes. This research suggests a move towards physical design strategies adapted from conventional design automation, potentially mitigating one of the major challenges to FCN’s further development.

I. INTRODUCTION & MOTIVATION

Field-coupled Nanocomputing (FCN, [1]), represents a novel category of advanced nanotechnologies that diverge from the traditional CMOS approach. Unlike conventional methods, these technologies leverage the manipulation of physical fields rather than electrical currents for information transmission and computational processes.

FCN technologies are distinguished by their potential for significantly low energy consumption, possibly surpassing the *Landauer Limit* [2]–[4], positioning them as an environmentally sustainable solution for the computational demands of the future, notably in data centers and for training large language models. Furthermore, FCN inherently implements the logic-in-memory concept, and possesses the theoretical capability to operate at terahertz frequencies [5], [6].

The journey of FCN from a theoretical framework to a contestant for the beyond-CMOS era of computing has been characterized by experimental breakthroughs in fabrication, namely, the development and commercial exploration of *Silicon Dangling Bonds* (SiDBs, [7]–[17]) by the research enterprise *Quantum Silicon Inc.*, which garnered

significant investment. SiDBs, atomic-scale quantum dots, have been employed in manufacturing FCN logic devices and wire segments that are smaller than 30 nm^2 [11]. In addition, the simulation of SiDBs [18]–[21] has led to the creation of standard gate libraries [22]–[25] and adaptations of the *Quantum-dot Cellular Automata* (QCA, [26]) concept, thereby enriching FCN’s versatility and integrating insights from the QCA domain.

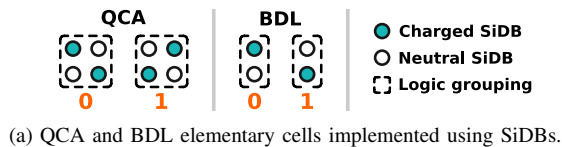
Despite these advancements, the physical design of FCN systems is still in its infancy, with information flow restrictions presenting a particularly daunting challenge [27]–[30]. In FCN systems, both combinational and sequential circuits must be clocked to enable information propagation and to ensure data synchronization [30]. Traditional approaches to clocking have relied on pre-assigned, regular clocking schemes, which often limit the result quality of circuit layouts obtained by physical design algorithms [31]–[34]. Furthermore, these algorithms become more complex in their design as they have to be tailored to specific clock topologies [35]. However, it is often unknown which particular clocking scheme is most suited for a circuit at hand [34], [35]. The need for more efficient and flexible design methodologies is evident, as the cumbersome nature of clocking hampers the scalability and practicality of FCN technologies.

This paper addresses these challenges by proposing an algorithm that could usher in a new era of FCN physical design where clocking is reduced from a main roadblock to an afterthought. To this end, we propose a novel method based on *Boolean Satisfiability* (SAT) solving that enables the assignment of clock numbers to unlocked FCN layouts, while adhering to all technological constraints regarding information propagation and synchronization, or proves that no such assignment exists.

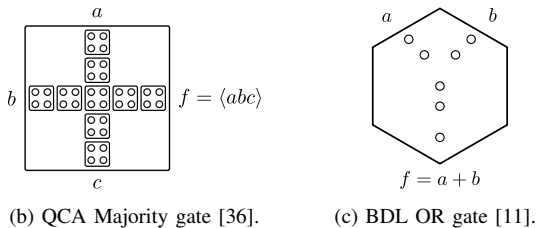
Via an experimental evaluation, we confirm the proposed algorithm’s *versatility* to reconstruct clock assignments for diverse clocking schemes in reasonable runtime, and its *scalability* up to layouts with a half-million tiles. Thereby, we are potentially paving the way for a new era of physical design algorithms that are not constrained by the limitations of predefined clocking schemes. This paper might, thus, be the missing link between conventional design automation and the FCN domain, and finally end the tyranny of the clock.

The remainder of this paper is structured as follows: Section II discusses the background on FCN required for the comprehension of this manuscript. Section III proposes

Marcel Walter, Jan Drewniok, and Robert Wille are with the Chair for Design Automation, Technical University of Munich, Germany. Marcel Walter is also with the University of Bremen, Germany. Robert Wille is also with the Software Competence Center Hagenberg GmbH (SCCH), Austria. E-mail: {marcel.walter, jan.drewniok, robert.wille}@tum.de



(a) QCA and BDL elementary cells implemented using SiDBs.



(b) QCA Majority gate [36]. (c) BDL OR gate [11].

Fig. 1: Elementary FCN devices and logic gates.

the novel SAT-based clock number assignment algorithm, which is then evaluated and discussed in Section IV. Finally, Section V concludes the paper.

II. FIELD-COUPLED NANOCOMPUTING

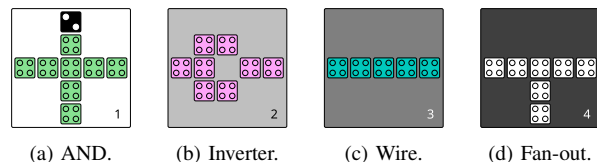
This section reviews the preliminaries of FCN technologies in an effort to keep this paper self-contained. To this end, Section II-A describes the elementary devices of information representation and manipulation. Section II-B covers the intricacies of FCN's information propagation and synchronization via clocking, which serves as a motivation for this work.

A. Cells and Gates

The fundamental unit of binary information in FCN technologies is the *cell*, which, despite variations across different technologies, exhibits three shared properties [1]: 1) it can be in a binary 0 or binary 1 state, 2) its logic state corresponds to the presence of a physical field (electric or magnetic) encoding said state, and 3) it influences adjacent cells to align their states via field coupling [26], [28], [37], [38]. This unique interaction allows for information transmission without electrical current, as cells can change states in response to an electric (or magnetic) field, propagating this change through adjacent cells [26].

Charge-based FCN cells, such as those used in QCA and SiDB technologies, operate on *Coulomb* interactions among quantum dots to define binary states [11], [26]. A QCA cell comprises four quantum dots positioned at the corners of a square, while an SiDB unit cell (called a BDL pair) uses just two dots. When one charge is shared between two quantum dots, *Coulomb* interactions dictate one of two possible charge distributions, which correspond to binary 0 and binary 1 as depicted in Figure 1a [7]–[10].

By arranging FCN cells spatially, logic gates and wire segments are created to process information through the same field interactions. The literature has introduced a variety of gate and wire segment designs across FCN technologies [22], [36], [39], [40]. For example, specific implementations like the QCA Majority gate and the BDL OR gate in Figure 1b and Figure 1c, respectively, highlight how different spatial arrangement of cells entail the gate's operation.



(a) AND. (b) Inverter. (c) Wire. (d) Fan-out.

Fig. 2: Gate tiles in QCA ONE implementation [40].

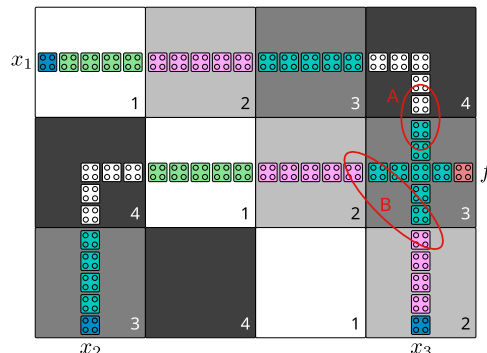


Fig. 3: Synchronization issues in FCN technologies, exemplarily depicted on a QCA circuit layout. The red A indicates a violation of local synchronization, while the red B marks a global synchronization problem.

Further implementations of QCA structures taken from the QCA ONE standard library can be seen in Figure 2 [40].

B. Clocking

In FCN circuits, *clocking* plays a pivotal role in ensuring signal stability and directing information flow in both combinational and sequential circuits, differing fundamentally from its function in CMOS technologies. To manage signal propagation effectively in larger circuits, FCN layouts are segmented into uniform tiles activated periodically by external coupling fields, called *clocks* [27], [28]. This tiling is visually represented by the black outlines around gates in Figure 1b and Figure 1c, where QCA utilizes square tiles within a Cartesian grid, and BDL adopts hexagonal tiling for its layout [22].

QCA circuits typically employ a four-phase clocking system, with each clock phase lasting $\frac{\pi}{2}$ of a full 2π *clock cycle*, facilitating a sequential pipeline for information flow across adjacent tiles numbered 1 through 4 as depicted in the tiles in Figure 2. Managing this flow requires meticulous design to prevent desynchronization, particularly by maintaining consecutive clock numbers of adjacent tiles and balancing wire lengths [30].

Figure 3 depicts a simple QCA layout with three primary inputs (PI) highlighted as blue cells and labeled x_1 , x_2 , and x_3 , respectively. It possesses one primary output (PO) at a Majority gate highlighted as an orange cell and labeled f . Upon cursory inspection, one could assume this layout to simply compute the Majority of x_1 , x_2 , and x_3 since there are no other gates in the layouts, merely wire segments transporting the input signals. However, the circuit suffers

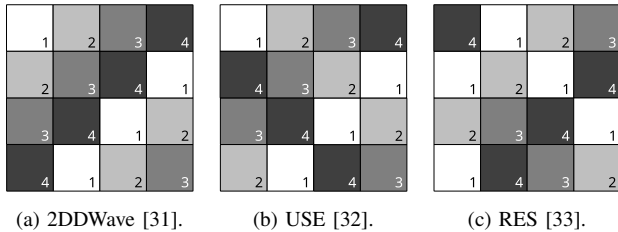


Fig. 4: Common clocking schemes for FCN circuit layouts.

from two issues: 1) at the intersection of the adjacent tiles marked with a red A, information is supposed to flow from the top to the bottom tile (from the PI toward the PO). However, the clock numbers do not allow for this propagation because no information can flow from 4 to 3. *Local Synchronization* is violated; and 2) the signals applied to the primary input x_2 have to pass four tiles before arriving at the gate with PO f , highlighted by the red B, while signals applied to x_3 only pass a single tile. Consequently, information that arrives at f is desynchronized and *Global Synchronization* is violated.

The distribution of clock signals is typically achieved through electrodes embedded within the circuit substrate, supporting various pre-defined regular *clocking schemes* detailed in the literature [27], [41]. The three most popular ones at the time of writing this manuscript are depicted in Figure 4 [31]–[33]. While these schemes provide standardized ways of implementing certain logic blocks, they often limit the result quality of large-scale non-uniform circuit layouts obtained by physical design algorithms, which additionally have to be tailored to each individual clock topology. Moreover, it is often unknown which particular clocking scheme is most suited for a circuit at hand. The need for more efficient and flexible design methodologies is evident, as the cumbersome nature of clocking hampers the scalability and practicality of FCN technologies.

III. PROPOSED SAT-BASED CLOCK NUMBER ASSIGNMENT

This section presents the novel SAT-based clock number assignment algorithm. First, its general idea is discussed in Section III-A. Afterward, the resulting SAT encoding is presented in Section III-B.

A. General Idea

The clock number assignment process requires an FCN layout without (or with partial) clocking information as input and outputs the same layout but with a valid clock number assignment, i. e., an assignment that ensures synchronization. By leveraging the strong reasoning capabilities of SAT solvers, we formulate the clock number assignment problem as a set of propositional logic constraints, drawing inspiration from *vertex coloring*. This formulation essentially encompasses the local synchronization constraint into propositional logic.

A key advantage of this approach is its ability to provide exact solutions—delivering a valid clocking scheme if one exists, or proving the layout’s unclockability, otherwise. Furthermore, the problem at hand is well-suited for the application of SAT solvers since it does not rely on iterative SAT solver calls, but can be formulated as a single holistic instance. In contrast to heuristic methods that may require iterative backtracking to find a solution, our method boosts efficiency of the search space traversal, especially in complex FCN designs characterized by numerous reconvergences and crossovers. The effectiveness of our algorithm is further enhanced by the implementation of strong symmetry breaking techniques, which prune the search space and effectively direct the SAT solver towards a satisfying solution.

B. SAT Encoding

The following constraints, inspired by vertex coloring, encode the clock number assignment problem in propositional logic, where the variable x_{tc} indicates that tile t receives clock number c , and k is the number of distinct clock signals in the technology (e. g., $k = 4$ for QCA):

- 1) Every tile has at least one clock number assigned:

$$\bigwedge_{t \in L} \bigwedge_{c \in \{1, \dots, k\}} x_{tc}$$

- 2) Every tile has at most one clock number assigned:

$$\bigwedge_{\substack{c, c' \in \{1, \dots, k\}, \\ c \neq c'}} \bigwedge_{t \in L} \neg x_{tc} \vee \neg x_{tc'}$$

- 3) Adjacent tiles have clock numbers assigned that allow for the flow of information:

$$\bigwedge_{(t, t') \in L} \bigwedge_{c \in \{1, \dots, k\}} \neg x_{tc} \vee \neg x_{t'(c+1 \bmod k)}$$

Here, let $(t, t') \in L$ denote a pair of adjacent tiles where information is supposed to be propagated from t to t' , i. e., t hosts a gate or wire segment that is a logic predecessor of the gate or wire segment hosted by t' .

Clock number assignments defined thusly contain inherent symmetries. In any clocking scheme, every clock number c can be replaced with $c + 1 \bmod k$, yielding up to $k - 1$ new valid clocking schemes for $k - 1$ iterations of this modular arithmetic transformation. These symmetries largely inflate the search space for the SAT solver, causing sub-par reasoning efficiency. However, they can be effectively minimized via the application of symmetry breaking clauses, which greatly increase the SAT solver’s performance, by pre-assigning fixed clock numbers to some tiles.

However, without any knowledge about the structure of the layout at hand, pre-assigning clock numbers without compromising equisatisfiability is delicate. A common idea—adapted from vertex coloring—is to assign the first clock number to the first primary input in the layout. Additionally, any of its neighbors can be assigned the second clock number without loss of generality. This pre-assignment may continue on a single straight path through the entire layout until a

primary output is reached. For optimal effectiveness, the layout’s critical path, i.e., the longest possible path from any primary input to any primary output, is chosen for this endeavor.

- 4) *Every tile on the critical path P has ascending clock numbers pre-assigned (symmetry breaking):*

$$\bigwedge_{t_i \in P} x_{t(i \bmod k)}$$

This extended encoding not only captures the essential local synchronization constraint but also incorporates symmetry breaking techniques for increased reasoning performance. By pre-assigning clock numbers along the critical path, we significantly prune the search space and guide the SAT solver towards an efficient exploration of feasible clocking schemes.

Having this encoding formulated, an instance can be constructed for any given Layout L and number of clocks k . It can be passed to a SAT solver, which will either return SAT together with a satisfying variable assignment, called a *model*, or UNSAT if no such assignment exists. From the model, the clock numbers can be extracted unambiguously and assigned to L .

IV. EXPERIMENTAL EVALUATION

This section presents the results of an experimental evaluation of the proposed clock number assignment algorithm. First, Section IV-A discusses the applied experimental setup. Afterward, two sets of results for different experiments are presented: 1) a *versatility* experiment in Section IV-B that demonstrates the applicability to different clocking schemes, and 2) a *scalability* experiment in Section IV-C that investigates the algorithm’s performance on larger layouts.

A. Experimental Setup

The clock assignment algorithm discussed and proposed in this work has been implemented in C++17 on top of the *fiction*¹ framework as part of the *Munich Nanotech Toolkit* (MNT, [42]). We evaluated different SAT solvers using the *EPFL Synthesis Library bill* [43] and settled for ABC’s *bsat2* [44] due to its slight performance advantage over the competition.

Both experiments were run on a distinct set of FCN gate-level layouts, we obtained from established datasets—the specifics of which are elaborated on in the respective sections. From both sets of layouts, we removed all clocking information and tasked the proposed clock number assignment algorithm with obtaining valid solutions while we measured its runtime. Finally, every resulting layout was formally verified for correctness in terms of logic and information propagation via clocking using the FCN equivalence checker proposed in [45].

All evaluations were run on a Manjaro 24 (Linux 6.9.2 kernel) machine with an AMD Ryzen 7 PRO 5850U CPU with 1.90 GHz (up to 4.40 GHz boost) and 32 GB DDR4 main memory.

¹The code is publicly available on GitHub at: <https://github.com/cda-tum/fiction>.

TABLE I: Excerpt of a versatility evaluation on layouts of three different clocking schemes.

	BENCHMARK CIRCUIT [46]			CLOCK ASSIGNMENT
	Name	I / O	$w \times h = A$	$t[s]$
2DDWave [31]	mux21	3 / 1	4 × 3 = 12	0.00
	xnor2	2 / 1	3 × 6 = 18	0.00
	par_gen	3 / 1	7 × 9 = 63	0.00
	t	5 / 2	8 × 8 = 64	0.00
	t_5	5 / 2	8 × 8 = 64	0.00
	par_check	4 / 1	9 × 9 = 81	0.00
	clpl	11 / 5	6 × 20 = 120	0.00
	newtag	8 / 1	11 × 11 = 121	0.00
	majority_5_r1	5 / 1	11 × 11 = 121	0.00
	xor5_r1	5 / 1	14 × 14 = 196	0.00
	cm82a_5	5 / 3	25 × 25 = 625	0.00
	xor5Maj	5 / 1	30 × 43 = 1290	0.02
	parity	16 / 1	48 × 48 = 2304	0.04
USE [32]	mux21	3 / 1	5 × 5 = 25	0.00
	xnor2	2 / 1	6 × 6 = 36	0.00
	par_gen	3 / 1	9 × 9 = 81	0.00
	t	5 / 2	10 × 10 = 100	0.00
	t_5	5 / 2	10 × 10 = 100	0.00
	par_check	4 / 1	11 × 11 = 121	0.00
	newtag	8 / 1	13 × 13 = 169	0.00
	majority_5_r1	5 / 1	13 × 13 = 169	0.00
	clpl	11 / 5	15 × 15 = 225	0.00
	xor5_r1	5 / 1	16 × 16 = 256	0.00
	cm82a_5	5 / 3	35 × 35 = 1225	0.03
	xor5Maj	5 / 1	45 × 45 = 2025	0.07
	parity	16 / 1	70 × 70 = 4900	0.42
RES [33]	mux21	3 / 1	5 × 5 = 25	0.00
	xnor2	2 / 1	6 × 6 = 36	0.00
	par_gen	3 / 1	11 × 11 = 121	0.00
	t	5 / 2	12 × 12 = 144	0.00
	t_5	5 / 2	12 × 12 = 144	0.00
	par_check	4 / 1	15 × 15 = 225	0.00
	newtag	8 / 1	15 × 15 = 225	0.00
	majority_5_r1	5 / 1	15 × 15 = 225	0.00
	clpl	11 / 5	18 × 18 = 324	0.00
	xor5_r1	5 / 1	20 × 20 = 400	0.00
	cm82a_5	5 / 3	50 × 50 = 2500	0.07
	xor5Maj	5 / 1	75 × 75 = 5625	0.33
	parity	16 / 1	110 × 110 = 12100	1.48

I and O are the number of inputs and outputs in the circuit, respectively; w , h , and A are the width, height, and resulting area (in tiles) of the layout, respectively. $t[s]$ is the runtime for clock number assignment in seconds.

B. Versatility

To assess the proposed algorithm’s *versatility*, we entered layouts that were originally generated on the three clocking schemes 2DDWave [31], USE [32], and RES [33] reviewed in Section II-B. These varying floor plans resulted in structurally diverse circuit layouts, destined for the sake of this evaluation. The runtime results of the proposed clock number assignment algorithm are summarized in Table I. All resulting assignments were found to be logically equivalent to the respective original layouts.

The obtained runtime values for this set of circuits are negligible—with the only exception being *parity* on the RES clocking scheme as it is by far the largest. 2DDWave, USE, and RES differ not only in their topology but also in their characteristics. While 2DDWave only allows for a linear information flow, USE and RES enable the creation of feedback loops, making the circuit design vastly more complex. Furthermore, RES offers distinct tiles with three inputs and others with three outputs, while 2DDWave and USE allow for a maximum of two inputs/outputs per tile. The successful recreation of every evaluated clock topology

demonstrates the proposed algorithm’s versatility, but leaves open the question for its scalability, i. e., its runtime behavior on larger-scale circuit layouts. This is investigated next.

C. Scalability

To assess the proposed algorithm’s *scalability*, we took the combinational benchmark functions from the established *IWLS93* [47] dataset, and placed and routed them using the state-of-the-art physical design algorithm *ortho* [48] to obtain larger-scale 2DDWave-clocked FCN layouts. The runtime results of the proposed clock number assignment algorithm are summarized in Table II. All resulting assignments were found to be logically equivalent to the respective original layouts.

As can be seen, the runtime values do not surpass 0.01 s for layouts with up to 1000 tiles, and only exceed 1 s for the first time on layouts with more than 10 000 tiles in case of the benchmark called *pcler8*. On layouts with 100 000 tiles, the runtime reaches $125.31\text{ s} \approx 2\text{ min}$ in case of the *i4* function. Finally, for the largest tested benchmarks with over 500 000 tiles in their resulting layout, like *i7*, *alu2* and *i9*, the runtime is in the half-hour range with 1884.70 s, 1508.78 s, and 1720.21 s, respectively. The successful recreation of these large clocking floor plans demonstrates the proposed algorithm’s scalability.

D. Discussion

In the physical design process of integrated circuits, algorithmic performance is of utmost importance given the complexity of contemporary systems. The algorithm proposed in this work positions itself in a new era of FCN physical design that breaks free from the tyranny of the clock. As soon as layouts can be generated without the limiting factor of data flow restrictions, a novel design flow can be envisioned, in which decades of research and development in the domain of CMOS-based electronics are reused for FCN. However, this vision ultimately depends on the performance with which any layout obtained this way can be assigned a valid clocking scheme, or, conversely, deemed unclockable.

As the previous evaluations demonstrate, the proposed algorithm is versatile such that it applies to determine valid clocking schemes for any given layout. However, as is the nature of exact algorithms, its runtime increases sharply beyond a certain input size. This kind of algorithm is to be utilized deliberately in appropriate contexts. Often, SAT-based techniques find applications in the obtainment of exact sub-solutions, i. e., after a problem is broken down into manageable fractions, an exact algorithm can be applied to each of them (in parallel).

With runtime values of 0.01 s on layouts with around 1000 tiles (e. g., *cm82a* and *cm42a*), a strategy can be envisioned that subdivides the largest benchmark, *i9*, into 519 distinct sub-layouts of up to 1000 tiles each and calls the proposed algorithm on each of them, which would result in a total runtime of 5.19 s—a sharp contrast to the 1720.21 s measured by considering the entire layout as one large problem instance. One might argue that there is no guarantee of

TABLE II: Excerpt of a scalability evaluation on larger-scale 2DDWave-clocked layouts.

Name	BENCHMARK CIRCUIT [47], [48]			CLOCK ASSIGNMENT
	I / O	$w \times h =$	A	$t[s]$
majority	5 / 1	$16 \times 26 =$	416	0.00
b1	3 / 4	$18 \times 27 =$	486	0.00
con1	7 / 2	$22 \times 36 =$	792	0.01
cm138a	6 / 8	$25 \times 34 =$	850	0.01
cm82a	5 / 3	$25 \times 39 =$	975	0.01
cm42a	4 / 10	$31 \times 34 =$	1054	0.01
cm152a	11 / 1	$30 \times 48 =$	1440	0.03
decod	5 / 16	$49 \times 53 =$	2597	0.07
xor5	5 / 1	$35 \times 79 =$	2765	0.07
cm85a	11 / 3	$38 \times 75 =$	2850	0.08
i1	25 / 13	$57 \times 73 =$	4161	0.16
cm163a	16 / 5	$46 \times 92 =$	4232	0.16
cmb	16 / 4	$51 \times 90 =$	4590	0.18
cm162a	14 / 5	$51 \times 92 =$	4692	0.18
tcon	17 / 16	$61 \times 78 =$	4758	0.20
misex1	8 / 7	$57 \times 91 =$	5187	0.25
cm151a	12 / 2	$54 \times 104 =$	5616	0.27
x2	10 / 7	$58 \times 97 =$	5626	0.27
pcler	19 / 9	$71 \times 106 =$	7526	0.44
pm1	16 / 13	$76 \times 103 =$	7828	0.51
cm150a	21 / 1	$57 \times 139 =$	7923	0.49
cu	14 / 11	$76 \times 119 =$	9044	0.61
parity	16 / 1	$62 \times 147 =$	9114	0.56
sqrt8ml	8 / 4	$72 \times 137 =$	9864	0.72
rd53	5 / 3	$74 \times 135 =$	9990	0.84
cc	21 / 20	$87 \times 117 =$	10179	0.90
pcler8	27 / 17	$87 \times 141 =$	12267	1.32
sqrt8	8 / 4	$82 \times 152 =$	12464	1.22
squar5	5 / 8	$89 \times 150 =$	13350	1.47
ldd	9 / 18	$112 \times 173 =$	19376	2.34
comp	32 / 3	$98 \times 213 =$	20874	2.48
unreg	36 / 16	$121 \times 182 =$	22022	3.83
misex2	25 / 18	$129 \times 201 =$	25929	5.53
b9	41 / 20	$132 \times 224 =$	29568	7.19
count	35 / 16	$153 \times 211 =$	32283	7.43
o64	130 / 1	$132 \times 259 =$	34188	16.22
i3	132 / 6	$139 \times 258 =$	35862	18.07
sct	19 / 15	$151 \times 239 =$	36089	10.19
lal	26 / 19	$150 \times 242 =$	36300	9.96
9symml	9 / 1	$181 \times 287 =$	51947	23.95
5xp1	7 / 10	$175 \times 299 =$	52325	21.89
z4ml	7 / 4	$188 \times 290 =$	54520	21.19
f51m	8 / 8	$216 \times 350 =$	75600	34.60
sao2	10 / 4	$213 \times 374 =$	79662	40.41
c8	28 / 18	$212 \times 389 =$	82468	36.67
my_adder	33 / 17	$227 \times 398 =$	90346	61.97
i2	201 / 1	$223 \times 435 =$	97005	107.41
i4	192 / 6	$243 \times 438 =$	106434	125.31
apex7	49 / 37	$259 \times 449 =$	116291	87.68
9sym	9 / 1	$252 \times 494 =$	124488	96.90
cht	47 / 36	$298 \times 524 =$	156152	132.69
rd73	7 / 3	$287 \times 547 =$	156989	138.28
i5	133 / 66	$371 \times 567 =$	210357	425.34
example2	85 / 66	$393 \times 585 =$	229905	362.54
ttt2	24 / 21	$375 \times 635 =$	238125	254.11
clip	9 / 5	$410 \times 738 =$	302580	532.06
i6	138 / 67	$463 \times 692 =$	320396	640.32
duke2	22 / 29	$475 \times 768 =$	364800	901.20
vg2	25 / 8	$476 \times 819 =$	389844	925.29
i7	199 / 67	$554 \times 907 =$	502478	1884.70
alu2	10 / 6	$577 \times 886 =$	511222	1508.78
i9	88 / 63	$612 \times 847 =$	518364	1720.21

I and O are the number of inputs and outputs in the circuit, respectively; w , h , and A are the width, height, and resulting area in tiles of the layout, respectively. $t[s]$ is the runtime for clock number assignment in seconds.

equisatisfiability between the overall problem formulation and the iterative one on sub-layouts, as clock numbers on the junctions might not match. However, as discussed in

Section III-B, clocking schemes can be transformed into symmetrical equivalent ones via simple modular arithmetic.

This paper might, thus, be the missing link between conventional design automation and the FCN domain, and finally end the tyranny of the clock.

V. CONCLUSION

This work introduced a novel algorithm for clock number assignment in the domain of *Field-coupled Nanocomputing* (FCN), addressing a major bottleneck in the scalability and flexibility of its physical design. By leveraging SAT-solving, our method circumvents the limitations imposed by traditional, predefined clocking schemes, offering a more adaptable and efficient approach to managing clock numbers while adhering to inherent technological constraints.

Via an experimental evaluation, we confirmed the proposed algorithm's *versatility* to reconstruct clock assignments for diverse clocking schemes in reasonable runtime, and its *scalability* up to layouts with a half-million tiles. Thereby, we are potentially paving the way for a new era of physical design algorithms that are not constrained by the limitations of predefined clocking schemes. This paper might, thus, be the missing link between conventional design automation and the FCN domain, and finally end the tyranny of the clock.

To support open research and open data, the implementation and the simulation results are publicly available as part of the *Munich Nanotech Toolkit* (MNT).

REFERENCES

- [1] N. G. Anderson *et al.*, *Field-coupled Nanocomputing: Paradigms, Progress, and Perspectives*. New York: Springer, 2014.
- [2] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM J. Res. Dev.*, vol. 5, no. 3, pp. 183–191, 1961.
- [3] R. W. Keyes *et al.*, "Minimal Energy Dissipation in Logic," *IBM J. Res. Dev.*, vol. 14, no. 2, pp. 152–157, 1970.
- [4] C. S. Lent *et al.*, "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling," *Nanotechnology*, vol. 17, no. 16, pp. 4240–4251, 2006.
- [5] J. Timler *et al.*, "Power Gain and Dissipation in Quantum-dot Cellular Automata," *J. Appl. Phys.*, vol. 91, no. 2, pp. 823–831, 2002.
- [6] L. Livadaru *et al.*, "Dangling-bond Charge Qubit on a Silicon Surface," *New Journal of Physics*, vol. 12, no. 8, p. 083018, Aug. 2010.
- [7] M. B. Haider *et al.*, "Controlled Coupling and Occupation of Silicon Atomic Quantum Dots at Room Temperature," *Physical Review Letters*, vol. 102, no. 4, p. 046805, 2009.
- [8] T. R. Huff *et al.*, "Atomic White-Out: Enabling Atomic Circuitry through Mechanically Induced Bonding of Single Hydrogen Atoms to a Silicon Surface," *ACS Nano*, vol. 11, no. 9, pp. 8636–8642, Sep. 2017.
- [9] N. Pavliček *et al.*, "Tip-induced Passivation of Dangling Bonds on Hydrogenated Si(100)-2×1," *Applied Physics Letters*, vol. 111, no. 5, p. 053104, 2017.
- [10] R. Achal *et al.*, "Lithography for Robust and Editable Atomic-Scale Silicon Devices and Memories," *Nature Communications*, vol. 9, no. 1, p. 2778, Jul. 2018.
- [11] T. Huff *et al.*, "Binary Atomic Silicon Logic," *Nature Electronics*, vol. 1, pp. 636–643, 2018.
- [12] R. A. Wolkow *et al.*, *Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics*. Springer, 2014, pp. 33–58.
- [13] M. Rashidi *et al.*, "Automated Atomic Scale Fabrication," US Patent 20 220 130 033, 2022.
- [14] T. Huff *et al.*, "Electrostatic Landscape of a Hydrogen-Terminated Silicon Surface Probed by a Moveable Quantum Dot," *ACS Nano*, vol. 13, no. 9, pp. 10 566–10 575, 2019.
- [15] R. Achal *et al.*, "Detecting and Directing Single Molecule Binding Events on H-Si (100) with Application to Ultradense Data Storage," *ACS Nano*, vol. 14, no. 3, pp. 2947–2955, 2019.
- [16] F. Altincicek, "Atomically Defined Wires on P-Type Silicon," *Bulletin of the American Physical Society*, 2022.
- [17] J. Pitters *et al.*, "Atomically Precise Manufacturing of Silicon Electronics," *ACS Nano*, 2024.
- [18] S. S. H. Ng *et al.*, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *TNANO*, vol. 19, pp. 137–146, 2020.
- [19] J. Drewniak *et al.*, "QuickSim: Efficient and Accurate Physical Simulation of Silicon Dangling Bond Logic," in *IEEE-NANO*, 2023, pp. 817–822.
- [20] —, "Temperature Behavior of Silicon Dangling Bond Logic," in *IEEE-NANO*, 2023, pp. 925–930.
- [21] —, "The Need for Speed: Efficient Exact Simulation of Silicon Dangling Bond Logic," in *ASP-DAC*, 2024.
- [22] M. Walter *et al.*, "Hexagons are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, vol. 22, 2022.
- [23] M. D. Vieira *et al.*, "Three-Input NPN Class Gate Library for Atomic Silicon Quantum Dots," *IEEE Design & Test*, 2022.
- [24] R. Lupoiu *et al.*, "Automated Atomic Silicon Quantum Dot Circuit Design via Deep Reinforcement Learning," 2022. [Online]. Available: <http://arxiv.org/abs/2204.06288>
- [25] A. N. Bahar *et al.*, "Atomic Silicon Quantum Dot: A New Designing Paradigm of an Atomic Logic Circuit," *TNANO*, pp. 807–810, 2020.
- [26] C. S. Lent *et al.*, "Quantum Cellular Automata," *Nanotechnology*, vol. 4, no. 1, p. 49, 1993.
- [27] K. Hennessy *et al.*, "Clocking of Molecular Quantum-dot Cellular Automata," *Journal of Vacuum Science & Technology B*, vol. 19, no. 5, pp. 1752–1755, 2001.
- [28] C. S. Lent *et al.*, "A Device Architecture for Computing with Quantum Dots," *Proceedings of the IEEE*, vol. 85, no. 4, pp. 541–557, 1997.
- [29] F. Sill Torres *et al.*, "On the Impact of the Synchronization Constraint and Interconnections in Quantum-dot Cellular Automata," *MICPRO*, vol. 76, pp. 103–109, 2020.
- [30] —, "Synchronization of Clocked Field-Coupled Circuits," in *IEEE-NANO*, 2018.
- [31] V. Vankamamidi *et al.*, "Clocking and Cell Placement for QCA," in *IEEE-NANO*, vol. 1. IEEE, 2006, pp. 343–346.
- [32] C. A. T. Campos *et al.*, "USE: A Universal, Scalable, and Efficient Clocking Scheme for QCA," *TCAD*, vol. 35, no. 3, pp. 513–517, 2016.
- [33] M. Goswami *et al.*, "An efficient clocking scheme for quantum-dot cellular automata," *Int. J. Electron. Lett.*, vol. 8, no. 1, pp. 83–96, 2020.
- [34] M. Walter *et al.*, "One-pass Synthesis for Field-coupled Nanocomputing Technologies," in *ASP-DAC*. ACM New York, NY, USA, 2021, pp. 574–580.
- [35] S. Hofmann *et al.*, "Thinking Outside the Clock: Physical Design for Field-coupled Nanocomputing with Deep Reinforcement Learning," in *ISQED*, 2024.
- [36] P. D. Tougaw *et al.*, "Logical devices implemented using Quantum Cellular Automata," *J. Appl. Phys.*, vol. 75, no. 3, pp. 1818–1825, 1994.
- [37] C. S. Lent *et al.*, "Molecular quantum-dot cellular automata," *J. Am. Chem. Soc.*, vol. 125, no. 4, pp. 1056–1063, 2003.
- [38] G. H. Bernstein *et al.*, "Magnetic QCA systems," *Microelectronics Journal*, vol. 36, no. 7, pp. 619–624, 2005.
- [39] D. Giri *et al.*, "A Standard Cell Approach for MagnetoElastic NML Circuits," in *NANOARCH*, 2014, pp. 65–70.
- [40] D. A. Reis *et al.*, "A methodology for standard cell design for QCA," in *ISCAS*, 2016, pp. 2114–2117.
- [41] J. Retallick *et al.*, "Low-Energy Eigenspectrum Decomposition (LEED) of Quantum-Dot Cellular Automata Networks," *TNANO*, vol. 20, pp. 104–112, 2021.
- [42] M. Walter *et al.*, "The Munich Nanotech Toolkit (MNT)," in *IEEE-NANO*, 2024.
- [43] M. Soeken *et al.*, "The EPFL Logic Synthesis Libraries," 2019.
- [44] R. Brayton and A. Mishchenko, "ABC: An Academic Industrial-Strength Verification Tool," in *Computer Aided Verification*. Springer, 2010, pp. 24–40.
- [45] M. Walter *et al.*, "Verification for Field-coupled Nanocomputing Circuits," in *DAC*, 2020.
- [46] S. Hofmann *et al.*, "MNT Bench: Benchmarking Software and Layout Libraries for Field-coupled Nanocomputing," in *DATE*, 2024.
- [47] K. McElvain, "IWLS'93 Benchmark Set: Version 4.0," 1993.
- [48] M. Walter *et al.*, "Scalable Design for Field-coupled Nanocomputing Circuits," in *ASP-DAC*. ACM New York, NY, USA, 2019, pp. 197–202.