Bridging the Gap Between Molecular FCN and Design Automation with SIM(7)-MolPDK: A Physically Simulated Standard-Cell Library

Bridging the Gap Between Molecular FCN and Design Automation with SIM(7)-MolPDK: A Physically Simulated Standard-Cell Library

Benjamin Hien[1], Danilo Quinci[2], Yuri Ardesi[2], Giuliana Beretta[2], Federico Ravera[2],
Marcel Walter[1,3], Robert Wille[1,3]
[1]Technical University of Munich, Germany [2]Politecnico di Torino, Italy
[3]Munich Quantum Software Company GmbH, Germany
Email: {benjamin.hien, marcel.walter, robert.wille}@tum.de, {yuri.ardesi, federico.ravera}@polito.it

***Abstract*—As CMOS technology approaches its physical and economic limits, alternative computing paradigms are being explored to overcome scaling, power, and manufacturing challenges. *Field-Coupled Nanocomputing* (FCN) is a promising post-CMOS approach that transmits information via electrostatic interactions rather than current flow. The molecular implementation of FCN—*Molecular Field-Coupled Nanocomputing* (MolFCN)—follows the Quantum-dot Cellular Automata (QCA) paradigm and offers room-temperature operation, ultra-high logic density, and ultra-low energy consumption. Recent advances in molecular device characterization and simulation make MolFCN circuit design more feasible than ever. However, most existing MolFCN circuits are manually designed under simplified assumptions, limiting their physical realism and scalability. While automated FCN design frameworks exist, they require verified *Standard-Cell Libraries* (SCLs), which are currently unavailable for molecular implementations. This work introduces SIM(7)-MolPDK, the first fully simulated MolFCN standard-cell library, enabling integration into an automatic FCN design framework. For the first time, physically realistic MolFCN circuits are synthesized automatically and validated through physical-level simulations, bridging the gap between molecular device research and design automation.**

I. Introduction and Motivation

Beyond power and scaling limitations, CMOS technologies face increasing manufacturing complexity and diminishing returns in performance-per-watt. As fabrication processes grow more intricate, defect rates and development costs continue to rise. Post-CMOS paradigms such as *Field-Coupled Nanocomputing* (FCN, [1]) offer a fundamentally different approach by transmitting information through electrostatic interactions rather than current flow.

Molecular Field-Coupled Nanocomputing (MolFCN) encodes the information in single molecules following the *Quantum-dot Cellular Automata* (QCA) paradigm and represents a strong candidate for post-CMOS nanocomputing [2], [3]. It enables room-temperature operation, overcoming one of the significant limitations of other QCA technologies [4]. Additionally, it benefits from chemical synthesis techniques enabling sub-nanometric control and packing densities far exceeding those achievable with traditional nanodevices [5]. MolFCN promises ultra-low power consumption combined with potentially high switching speeds, offering a compelling balance of energy efficiency and performance [6], [7].

Despite its potential, MolFCN technology faces several challenges that limit its practical adoption. Manual circuit design is time-consuming and error-prone, particularly for larger circuits. To address this, design automation tools are essential—but the distinct computing paradigm of FCN, which encodes information through electrostatic interactions rather than current flow, demands specialized tools that can handle its unique physical and timing constraints [8]–[14]. Fortunately, such FCN-specific tools already exist and have matured to support full design flows [10], [15]. These tools, much like their CMOS counterparts, rely on hierarchical abstractions to manage design complexity efficiently. One of the most critical elements in this hierarchy is the *Standard-Cell Library* (SCL), which provides verified logic and layout templates as building blocks for scalable circuit synthesis. However, MolFCN lacks such a library, creating a gap that prevents its seamless integration into existing FCN design flows.

This work addresses that gap by introducing the first fully simulated SCL in the *MolFCN Physical Design Kit (MolPDK)*, named *SIM(7)*, and corresponding design rules for molecular FCN. By integrating this library into an established FCN design framework [15], we enable the automated synthesis of MolFCN circuits that are functionally correct and physically realistic. To verify correctness, synthesized layouts are re-simulated using the *Self-Consistent Electrostatic Potential Algorithm* (SCERPA) [16] engine at the physical level.

The remainder of the paper is structured as follows. Section II provides the necessary background on MolFCN and its supporting tools. Section III reviews prior work on MolFCN simulation and FCN design automation. Section IV details the proposed molecular FCN standard-cell library SIM(7), including its construction, validation, and design rules. Section V presents experimental results demonstrating the integration of the library into a complete design flow. Finally, Section VI concludes the paper and outlines directions for future work.

II. Background

This section provides the necessary background on MolFCN and its supporting tools. It introduces the fundamental principles of MolFCN, outlines its simulation methodology, reviews physical-design flows for nanocomputing, and discusses the role of standard-cell libraries in enabling scalable design automation.

A. MolFCN Fundamentals

MolFCN encodes the information in the charge distribution of molecular cells. Fig. 1(a) shows the MolFCN cell composed of two molecules. The charge distribution of the two molecules resembles the definition of the QCA cell, proposed by Lent et al. in 1993 [2]. Thanks to Coulomb repulsion, the charge distribution aggregates on the antipodal sites of the molecules, thus creating two possible configurations used for logic encoding, visualized in Fig. 1(b) [2]. A third state, with an intermediate charge state, can be forced by an external clock field and used to encode the NULL state [17], [18]. Fig. 1(c) shows an example of clocked molecular wire [19]. The NULL state allows for the control and guidance of information flow, enabling efficient information propagation, elaboration, and pipelining. Finally, complex layouts can be used to create logic gates, such as inverters and majority voters, or more complex devices [19].

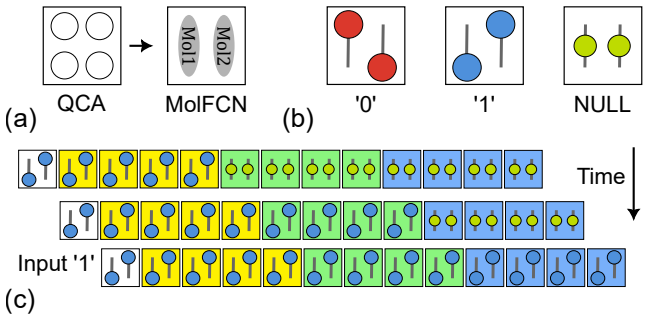

Fig. 1: Basics of MolFCN: (a) Realization of the MolFCN cell according to the Quantum-dot Cellular Automata paradigm; (b) Information encoding in MolFCN cells; (c) Clocked MolFCN wire.

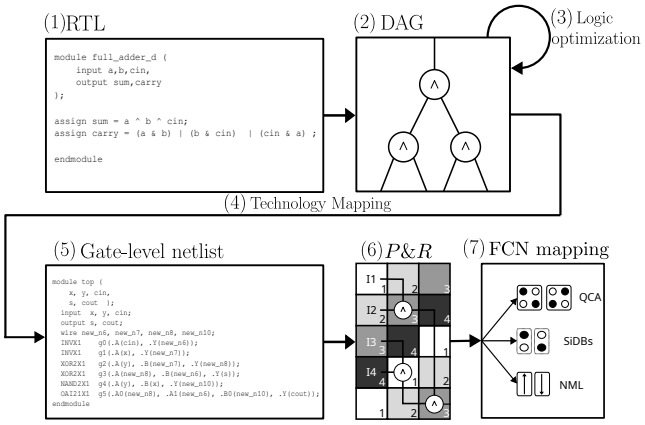

Fig. 2: Standard design flow for FCN circuit layouts.

B. MolFCN Simulation

The scientific community is currently exploring MolFCN devices and circuits using two simulation-based approaches: *QCADesigner* [20] and *SCERPA* [16]. Many works use QCADesigner to design and simulate QCA devices, potentially implementable with MolFCN technology. However, a recent study highlighted the need to validate QCADesigner results, as their validity depends on the molecular nature [21]. Indeed, the electrostatic properties of molecules—crucial in MolFCN technology—depend on their molecular state, which can be oxidized, neutral, or in a mixed condition, and must be considered in the simulation. The SCERPA tool, part of the ToPoliNano framework [10] and recently validated with DFT precision, enables the fast simulation of MolFCN devices by considering their effective physics and electrostatic characteristics [16], [22]. SCERPA relies on the *MoSQuiTo* methodology, which provides a physics-aware method for designing MolFCN devices [18]. It exploits *ab initio* characterization of molecular charge distribution to provide an adequate characterization of molecules under the effect of electrostatic fields, which play a key role in the MolFCN working principle. Then, SCERPA iteratively evaluates the field generated by the molecules and their resulting charge distribution through the so-called *Vin-Aggregated Charge Transcharacteristics* (VACT). This self-consistent approach enables the evaluation of signal propagation in realistic molecular structures, permitting simulation and optimisation of circuit layouts.

C. Physical-Design Flow for Nanocomputing

The automated layout-generation flow for FCN, shown in Fig. 2, begins with an abstract behavioural specification and concludes with a fabrication-ready cell-level layout.

Starting from a high-level *Register-Transfer Level* (RTL) description **(1)** in a hardware description language (e. g., Verilog or VHDL), only the functional behaviour is defined. This Boolean description is compiled into a *Directed Acyclic Graph* (DAG) **(2)**, such as an *And-Inverter Graph* (AIG) or *Majority-Inverter Graph* (MIG), which enables efficient logic optimizations **(3)** [23], [24] to reduce node count and critical-path depth. In the technology mapping step **(4)** [25], the optimised network is mapped to gate primitives from an FCN standard-cell library (SCL), producing a gate-level netlist **(5)**. Placement and routing (P&R) **(5)** [26]–[29] then positions gates on a two-dimensional grid and interconnects them according to the physical design rules. Finally, in the FCN mapping step **(6)**, the gate-level layout is translated into a concrete, cell-level layout for the target nanotechnology. For example, a square-cell QCA layout can be implemented using different QCA technologies, depending on the chosen SCL.

D. Standard Cell Libraries

As illustrated in the design flow above, the SCL is a central element of automated FCN circuit design. It contains the physical implementations of Boolean functions that serve as reusable building blocks for constructing complete circuits.

Analogously to CMOS, design automation in FCN follows a hierarchical approach: complex systems are built from smaller, standardized components. The SCL enables this by providing a systematic, composable set of standard cells. To represent arbitrary logic networks, the library must be functionally complete—meaning it must contain a universal set of logic functions, such as $\{\text{AND}, \text{INV}\}$ or $\{\text{MAJ}, \text{INV}\}$, depending on the synthesis strategy and target technology.

In addition to logic gates, an FCN SCL typically includes essential routing primitives, such as wires and fan-outs, which ensure that the generated layouts satisfy the physical and topological constraints of the chosen technology.

III. Related Work

This section reviews prior work on the simulation of MolFCN circuits and on design automation for Field-Coupled Nanocomputing. It first outlines how molecular devices and circuits have been modeled and validated, and then considers tools and frameworks for integrating physically validated standard-cell libraries into complete design flows.

A. MolFCN Simulation Tools

Most of the works related to the MolFCN technology focus on the development and characterization of new molecules or understanding the basic phenomena related to molecular physics. At the system level, many works have proposed MolFCN circuits using QCADesigner, without considering the effective physics of the underlying molecules. With SCERPA, the first attempt at providing standard MolFCN cells proposed fixed dimensions self-assembled cells [30]. The logic operations are achieved through the correct operations of the clocking mechanisms, rather than the circuit layout, to avoid nanopatterning. A more recent work defined a full design flow—from the physical modeling of molecule interactions to the creation of logic primitives—favoring the development of an SCL compatible

with nanocomputing architectures [31]. It sets the current state-of-the-art of devices, which is the 1-bit full adder. However, the proposed cells are still non-standardized in terms of size, thus making it difficult to automatically place logic gates.

B. FCN Design Automation Tools

The design of FCN circuits is supported by a variety of software tools. However, most of these tools are ill-suited for the integration and validation of novel SCLs, such as the one proposed in this work.

Tools like *SiQAD* [8] are technology-specific, focusing exclusively on *Silicon Dangling Bonds* [32], [33]—an alternative FCN technology—, rather than MolFCN. *ToPoliNano* [9]–[11], on the other hand, is tailored for nanomagnetic logic and is closed-source, preventing the integration of new libraries. The foundational QCA tool, *QCADesigner* [12], and its derivatives are largely outdated and lack the necessary extensibility and automation infrastructure for modern design flows. Specialized tools such as *MagCAD/ToPoliNano* [9]–[11], [13] provide physical simulation and analysis for MolFCN, but do not offer a complete design automation environment, as they lack gate-level synthesis, placement, and routing capabilities.

A more suitable foundation for FCN design automation is provided by the *fiction* framework [10], [15]. It is a comprehensive, open-source C++17 library designed to be technology-independent. Its key strengths are its flexible data structures and, most importantly, its support for custom gate libraries. This allows for the straightforward integration of new standard cells, like the library proposed in this work, and their use across the entire physical design flow, including logic synthesis, placement, routing, clocking, and verification. The framework's ability to abstract technology-specific details while providing a full suite of design algorithms makes it the only viable platform for effortlessly incorporating and evaluating a novel, physically-validated standard-cell library. Therefore, we use *fiction* as the basis for all design automation tasks in this work.

IV. The Proposed SIM(7) Standard Cell Library

This section presents the MolFCN standard-cell library developed in this work, named SIM(7), and part of the *MolFCN Physical Design Kit* (MolPDK). It describes the construction and physical design of the cells, the simulation and validation process, and the design rules that ensure compatibility with automated synthesis and layout generation. The name SIM(7) denotes that the library is simulated with SCERPA (S) and constituted with an ideal molecule (IM). In addition, the library contains 7 gates: bus, inverter, L-wire, fan-out, 3-input majority voter, as well as 2-input AND and OR gates.

A. Standard Cell Design

The SIM(7) standard cells were implemented on a uniform grid of 10×10 MolFCN cells ($400\,\mathrm{nm}^2$) with the same dedicated pin locations across all standard cells. These uniformities are important for compatibility with state-of-the-art physical design algorithms in the FCN domain, as, e. g., implemented in the *fiction* framework [15].

MagCAD was used as the IDE to manually design the standard cells [11]. This size was chosen to provide sufficient space for the physical implementation of the required logic devices and clock zones, as well as to minimize possible crosstalk among adjacent standard cells [21]. The implemented devices include fundamental logic gates (AND, OR, majority voter, inverter) and interconnection elements (wires, L-wires, fan-outs).

Each layout was designed by integrating the necessary clock zones for the control of information propagation, following a typical four-stage clocking scheme of FCN systems [17]. Table I summarizes

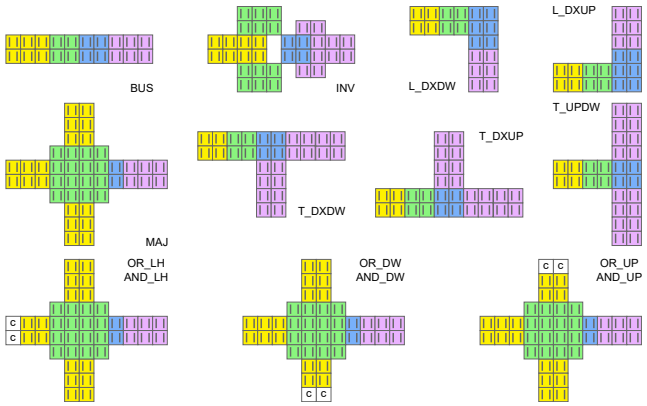

Fig. 3: Standard devices (wires and logic gates) composing the SIM(7) MolFCN standard cell library.

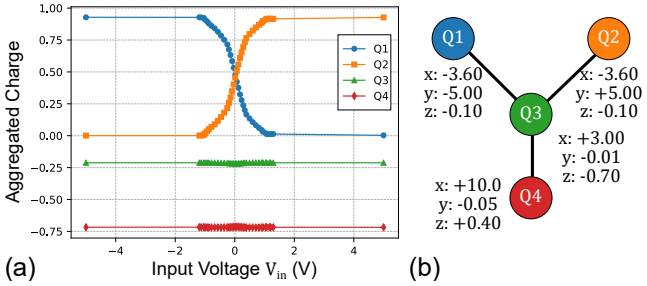

Fig. 4: Employed molecule: (a) Vin-Aggregated Charge Transcharacteristics (VACT); (b) Geometrical model of the Aggregated Charge system.

TABLE I: Summary of the proposed SIM(7) standard cells with their corresponding physical parameters.

Type	Inputs	Outputs	N. of cells	Size
BUS	4	4	20	10×10
INV	4	4	34	10×10
L	4	4	20	10×10
MAJ	12	4	40	10×10
T	4	8	28	10×10
AND	12	4	38	10×10
OR	12	4	38	10×10

the characteristics of the elementary MolFCN devices developed, showing the number of inputs, outputs, and molecular cells used for each type of device. Fig. 3 shows all the developed library elements.

All the devices are constituted with an ideal molecule in order to avoid molecule-dependent molecular cells. The intermolecular distance is fixed to $1\,\mathrm{nm}$. The characteristics of the molecule have been decided to favor bistability [34] and to resemble the geometry of the well-known bis-ferrocene [18], [35]. Fig. 4(a) shows the VACT of the used molecule, whereas Fig. 4(b) depicts the Aggregated Charge geometry used. The intermolecular distance is fixed to $1\,\mathrm{nm}$.

Functional validation of the designed layouts was performed using *BBchar* [36], [37], a MolFCN device characterization framework that integrates the SCERPA algorithm [16], [38] for the simulation of MolFCN circuits. Simulations were performed on QLL layout files generated for each implemented device. A test protocol was defined for each device by applying four-step clock signals (switch, hold, release, reset) with linear transitions between logic levels ($-1\,\mathrm{V}$ to $1\,\mathrm{V}$ in 7 steps). All library devices were simulated using termination

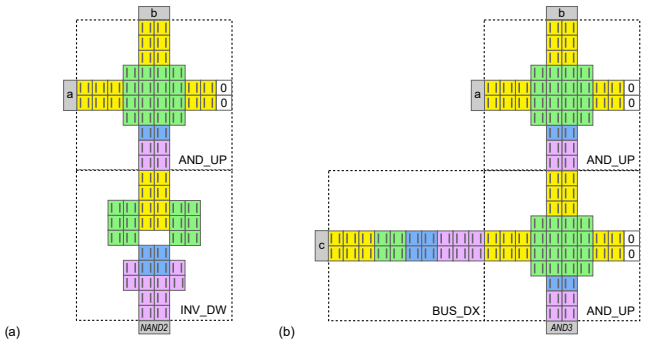

Fig. 5: Composed logic blocks from the proposed SIM(7) standard cells: (a) MolFCN NAND2 gate; (b) MolFCN AND3 gate.

structures to emulate their operation within a more complex circuit, following established techniques to characterize FCN circuits [37]. It is left for future work to design crosswires [31], [39] and L-shaped bent inverters [21], i. e., with orthogonal input and output pins.

B. Usage of Standard Cells

The modularity of SIM(7) enables the seamless construction of complex circuits by plugging together pre-verified building blocks. Each standard cell is internally divided into four clock zones, ensuring directional signal flow and proper pipelining. When cells are connected—for example, the output of one cell to the input of another—synchronization between adjacent cells is automatically preserved, as each new cell begins with clock zone 1, ensuring *local synchronization* [14].

Example 1. Consider the circuit in Fig. 5(a), resulting in a `NAND` function. It can be constructed by connecting the output of an `AND` cell to the input of an `INV` cell. The standard cells align their input and output ports with the global grid, enabling straightforward and error-free composition.

While this modular approach is highly convenient, connecting standard cells must also respect the *global synchronization* constraint [14], which requires that all signals feeding into a tile arrive at the same time.

Example 2. Consider the MolFCN layout for a `AND3` function in Fig. 5(b). It is constructed by connecting two `AND` cells. However, due to the global synchronization constraint, a delay must be added to the third input using a wire, ensuring all signals arrive simultaneously.

Although such adjustments are manageable in small examples, they quickly become intractable in larger circuits. This is why design automation tools are essential: they handle these challenges systematically, especially the combined placement, routing, and clocking problem, which is known to be $\mathcal{NP}$-complete [40].

Example 3. Consider the MolFCN layout for the `c17` benchmark [41] shown in Fig. 6. This circuit was automatically generated using the *fiction* framework, demonstrating that the same modular design principles scale beyond small illustrative examples. While the framework is capable of constructing substantially larger circuits, current physical simulation tools cannot yet handle such sizes efficiently.

V. Experimental Evaluation

This section presents the experimental evaluation of the proposed MolFCN SCL. The primary objectives are to 1) demonstrate functional correctness of the individual cells under physical-level simulation, 2) validate their integration into a complete design flow, and 3) assess the correctness of automatically synthesized and laid out circuits using the library.

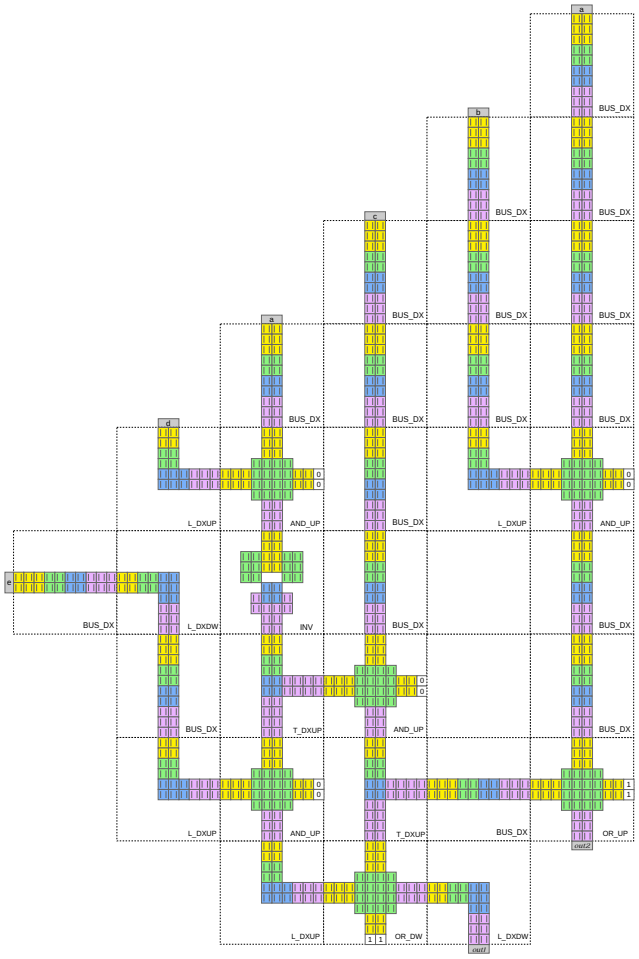

Fig. 6: Constructing circuit layouts from the proposed SIM(7) standard cells using the design automation framework *fiction*.

A. Evaluation Setup

To enable design automation, the proposed SIM(7) was integrated into the open-source *fiction* framework (cf. Section III-B).[1] Using this extended toolchain, we synthesized layouts for a representative set of benchmark circuits from Verilog specifications [26]. The resulting physical layouts were then simulated using the SCERPA engine to verify logic functionality at the molecular level.

B. Cell-Level Verification

The first set of experiments focuses on verifying the functionality of the individual standard cells in the proposed SIM(7) library. All the cells are verified on all the possible input combinations. Fig. 7 shows SCERPA simulation results for a subset of the most representative cells. For each device, the input and output waveforms are plotted, illustrating signal propagation through the respective clock zones. Each waveform depicts the voltage generated by the charge distribution of the device output molecule, directly representing the

[1]The *fiction* framework is openly available on GitHub as part of the *Munich Nanotech Toolkit* (MNT) [42]: https://github.com/cda-tum/fiction

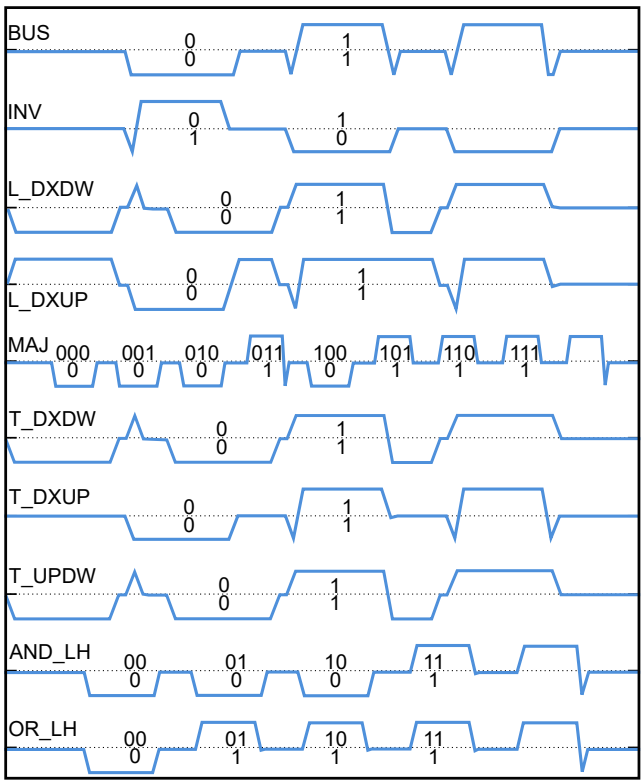

Fig. 7: Simulation results of the proposed SIM(7) standard cells.

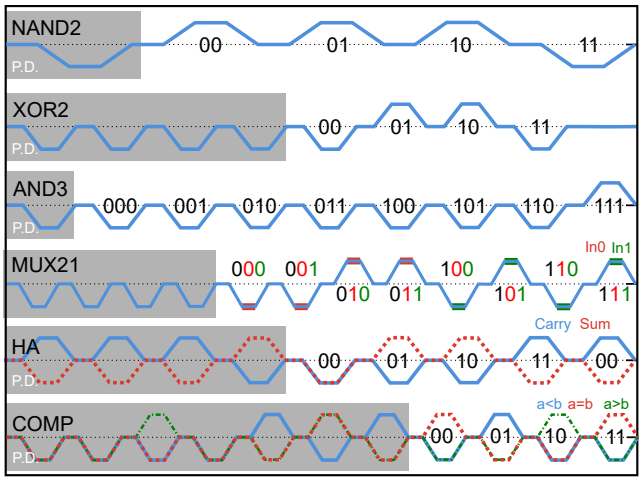

Fig. 8: Simulation results of synthesized circuits using SIM(7).

device logic output. The presence of spikes reflects the physics-aware model used in the SCERPA tool. The correct output is highlighted with labels denoting input (first line) and output combinations (second line).

C. Circuit-Level Verification

To evaluate the applicability of SIM(7) in a full design flow, we synthesized and laid out several multi-gate circuits, including basic combinational and arithmetic components. These designs were generated automatically using *fiction* and subsequently verified at the molecular level with SCERPA.

Fig. 8 presents representative simulation results for selected circuits: a two-input NAND gate (**NAND2**), a two-input XOR gate (**XOR2**), a three-input AND gate (**AND3**), a 2:1 multiplexer (**MUX21**), a half-adder (**HA**), and a 1-bit comparator (**COMP**). The waveforms, compared to Fig. 7, exhibit fewer spikes. Indeed, Fig. 8 shows the logic output of the devices, rather than the voltage, allowing for easier analysis of logically complex devices. Each device is subdivided into several clock regions, which are responsible for the propagation delay ($P.D.$) observed in the waveforms, which can be measured in terms of clock cycles (cc). In all cases, the simulated outputs match the expected truth tables:

- **NAND2**: Low output only when the two inputs are high.
- **XOR2**: High output only when the two inputs differ.
- **AND3**: High output only when all inputs are high.
- **MUX21**: Correctly forwards one of the two inputs depending on the selection signal.
- **HA**: Produces correct sum and carry signals for all input combinations.
- **COMP**: Correctly signals the majority, minority or equivalence for all input combinations.

TABLE II: Metrics for the considered devices: propagation delay, number of SCs, SC arrangement, and occupied area.

Device	P.D. [cc]	#SCs	SC Arrangement	Area [nm^2]
NAND2	2	2	2×1	800
XOR2	5	12	5×4	8000
AND3	2	3	2×1	800
MUX21	5	13	5×4	8000
HA	5	14	6×4	9600
COMP	8	24	9×7	25200

Finally, Table II reports the metrics for the considered devices. The propagation delay, the number of SCLs used in the design, the arrangement of the SCLs, and the area occupied by the cells.

As discussed in Section IV-B, larger circuits could be generated using the *fiction* framework; however, the simulation of large circuits requires considerations which are out of the scope of this paper, thus left for future work.

D. Summary

The evaluation confirms that the proposed SCL, SIM(7), can be seamlessly integrated into an automated MolFCN design flow. Both cell-level and circuit-level simulations show correct functional behavior under physical-level simulation, demonstrating that the generated layouts are realizable and logically correct. These results validate the library as a foundation for scalable MolFCN circuit synthesis.

VI. Conclusion and Future Work

This work introduces the first standard-cell library for *Molecular Field-Coupled Nanocomputing* (MolFCN), named SIM(7), marking a significant step toward circuit design in this emerging technology. The library is simulated with physical precision through the *Self-Consistent Electrostatic Potential Algorithm* (SCERPA). The proposed library, integrated into the *fiction* design automation framework, enables the automatic synthesis of MolFCN logic circuits from Verilog specifications that achieve 100 % functional correctness, with competitive area efficiency. All the generated circuits have been verified with SCERPA, demonstrating the capability of *fiction* to generate automatic and realistic layouts. Future studies will target large benchmark circuits, the incorporation of devices such as crosswires and L-shaped inverters, and the development of defect-tolerant placement, as well as dynamic and energy-dissipation models.

References

[1] N. G. Anderson and S. Bhanja, *Field-coupled Nanocomputing: Paradigms, Progress, and Perspectives*. New York: Springer, 2014.

[2] C. S. Lent, B. Isaksen, and M. Lieberman, "Molecular Quantum-Dot Cellular Automata," *Journal of the American Chemical Society*, vol. 125, no. 1, pp. 1056–1063, 2003.

[3] F. Ravera, G. Beretta, Y. Ardesi, M. Krzywiecki, M. Graziano, and G. Piccinini, "A roadmap for molecular field-coupled nanocomputing actualization," in *2023 IEEE Nanotechnology Materials and Devices Conference (NMDC)*. IEEE, Oct. 2023.

[4] Y. Wang and M. Lieberman, "Thermodynamic Behavior of Molecular-Scale Quantum-Dot Cellular Automata (QCA) Wires and Logic Devices," *IEEE Transactions on Nanotechnology*, vol. 3, no. 3, pp. 368–376, 2004.

[5] C. S. Lent, "Bypassing the transistor paradigm," *Science*, vol. 288, no. 5471, pp. 1597–1599, Jun. 2000.

[6] R. Listo, F. Ravera, G. Beretta, Y. Ardesi, G. Piccinini, and M. Graziano, "Unveiling Charge Dynamics in Molecular Field-coupled Nanocomputing," in *IEEE-NANO*. IEEE, 2024, pp. 424–429.

[7] J. Timler and C. S. Lent, "Power Gain and Dissipation in Quantum-Dot Cellular Automata," *journal of applied physics*, vol. 91, no. 2, pp. 823–831, 2002.

[8] S. S. H. Ng, R. A. Wolkow, K. Walus, J. Retallick, H. N. Chiu, R. Lupoiu, L. Livadaru, T. Huff, M. Rashidi, W. Vine, and T. Dienel, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *TNANO*, vol. 19, pp. 137–146, 2020.

[9] F. Riente, G. Turvani, M. Vacca, M. R. Roch, M. Zamboni, and M. Graziano, "ToPoliNano: A CAD Tool for Nano Magnetic Logic," *IEEE TCAD*, vol. 36, no. 7, pp. 1061–1074, 2017.

[10] U. Garlando, M. Walter, R. Wille, F. Riente, F. S. Torres, and R. Drechsler, "ToPoliNano and Fiction: Design Tools for Field-coupled Nanocomputing," in *DSD*, 2020, pp. 408–415.

[11] U. Garlando, F. Riente, D. Vergallo, M. Graziano, and M. Zamboni, "ToPoliNano & MagCAD: A Complete Framework for Design and Simulation of Digital Circuits Based on Emerging Technologies," in *SMACD*, 2018, pp. 153–156.

[12] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata," *TNANO*, vol. 3, no. 1, pp. 26–31, 2004.

[13] F. Riente, U. Garlando, G. Turvani, M. Vacca, M. R. Roch, and M. Graziano, "MagCAD: Tool for the Design of 3-D Magnetic Circuits," *JXCDC*, vol. 3, pp. 65–73, 2017.

[14] F. S. Torres, M. Walter, R. Wille, D. Große, and R. Drechsler, "Synchronization of clocked field-coupled circuits," in *2018 IEEE 18th International Conference on Nanotechnology (IEEE-NANO)*, 2018, pp. 1–5.

[15] M. Walter, R. Wille, F. S. Torres, D. Große, and R. Drechsler, "Fiction: An Open Source Framework for the Design of Field-coupled Nanocomputing Circuits," *arXiv preprint arXiv:1905.02477*, 2019.

[16] Y. Ardesi, R. Wang, G. Turvani, G. Piccinini, and M. Graziano, "SCERPA: A Self-Consistent Algorithm for the Evaluation of the Information Propagation in Molecular Field-Coupled Nanocomputing," *IEEE TCAD*, vol. 39, no. 10, pp. 2749–2760, 2019.

[17] K. Hennessy and C. S. Lent, "Clocking of Molecular Quantum-dot Cellular Automata," *J. Vac. Sci. Technol. B*, vol. 19, no. 5, pp. 1752–1755, 2001.

[18] Y. Ardesi, A. Pulimeno, M. Graziano, F. Riente, and G. Piccinini, "Effectiveness of Molecules for Quantum Cellular Automata as Computing Devices," *Journal of Low Power Electronics and Applications*, vol. 8, no. 3, 2018. [Online]. Available: https://www.mdpi.com/2079-9268/8/3/24

[19] P. D. Tougaw and C. S. Lent, "Logical devices implemented using Quantum Cellular Automata," *Journal of Applied Physics*, vol. 75, no. 3, pp. 1818–1825, 1994.

[20] K. Walus, T. Dysart, G. Jullien, and R. Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 3, no. 1, pp. 26–31, 2004.

[21] Y. Ardesi, G. Beretta, M. Vacca, G. Piccinini, and M. Graziano, "Impact of Molecular Electrostatics on Field-Coupled Nanocomputing and Quantum-Dot Cellular Automata Circuits," *Electronics (Basel)*, vol. 11, no. 2, p. 276, Jan. 2022.

[22] Y. Ardesi, F. Mo, M. Vacca, G. Piccinini, and M. Graziano, "Guesstimation of Molecular Ensemble Electrostatics Properties through SCERPA-DFT Calculation: Molecular Field-coupled Nanocomputing as a Case Study," *Adv. Theory Simul.*, no. e00812, Jun. 2025.

[23] A. Mishchenko, S. Chatterjee, and R. Brayton, "DAG-aware AIG Rewriting a Fresh Look at Combinational Logic Synthesis," in *DAC*, 2006.

[24] A. Mishchenko, R. Brayton, J.-H. R. Jiang, and S. Jang, "Scalable Don't-Care-Based Logic Optimization and Resynthesis," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 4, no. 4, pp. 1–23, 2011.

[25] A. T. Calvino, H. Riener, S. Rai, A. Kumar, and G. De Micheli, "A Versatile Mapping Approach for Technology Mapping and Graph Optimization," in *ASP-DAC*. IEEE, 2022, pp. 410–416.

[26] M. Walter, R. Wille, D. Große, F. S. Torres, and R. Drechsler, "An exact method for design exploration of quantum-dot cellular automata," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 503–508.

[27] M. Walter, R. Wille, F. S. Torres, D. Große, and R. Drechsler, "Scalable design for field-coupled nanocomputing circuits," in *2019 24th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2019, pp. 1–6.

[28] S. Hofmann, M. Walter, and R. Wille, "A* is born: Efficient and scalable physical design for field-coupled nanocomputing," in *2024 IEEE 24th International Conference on Nanotechnology (NANO)*, 2024, pp. 80–85.

[29] B. Hien, M. Walter, S. Hofmann, and R. Wille, "A Fully Planar Approach to Field-coupled Nanocomputing: Scalable Placement and Routing Without Wire Crossings," 2025. [Online]. Available: https://arxiv.org/abs/2504.09012

[30] Y. Ardesi, G. Beretta, C. Fabiano, M. Graziano, and G. Piccinini, "A Reconfigurable Field-coupled Nanocomputing Paradigm on Uniform Molecular Monolayers," in *ICRC*, 2021, pp. 124–128.

[31] Y. Ardesi, U. Garlando, F. Riente, G. Beretta, G. Piccinini, and M. Graziano, "Taming Molecular Field-coupling for Nanocomputing Design," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 19, no. 1, pp. 1–24, 2022.

[32] R. A. Wolkow, L. Livadaru, J. Pitters, M. Taucer, P. Piva, M. Salomons, M. Cloutier, and B. V. Martins, "Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics," in *Field-Coupled Nanocomputing*, 2013.

[33] T. Huff, H. Labidi, M. Rashidi, L. Livadaru, T. Dienel, R. Achal, W. Vine, J. Pitters, and R. A. Wolkow, "Binary atomic silicon logic," *Nat. Electron.*, vol. 1, no. 12, pp. 636–643, 2018.

[34] Y. Ardesi, M. Graziano, and G. Piccinini, "A model for the evaluation of monostable molecule signal energy in molecular field-coupled nanocomputing," *Journal of Low Power Electronics and Applications*, vol. 12, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2079-9268/12/1/13

[35] V. Arima, M. Iurlo, L. Zoli, S. Kumar, M. Piacenza, F. Della Sala, F. Matino, G. Maruccio, R. Rinaldi, F. Paolucci, M. Marcaccio, P. G. Cozzi, and A. P. Bramanti, "Toward Quantum-Dot Cellular Automata Units: Thiolated-Carbazole Linked Bisferrocenes," *Nanoscale*, vol. 4, no. 3, pp. 813–823, 2012.

[36] G. Beretta, Y. Ardesi, M. Graziano, and G. Piccinini, "vlsi-nanocomputing/BBchar: v1.0.1," 2023.

[37] G. Beretta, "Modeling and Simulation Strategies for Advancing Molecular FCN: Pioneering Design Rules Toward Implementation from Molecules to Devices," PhD thesis, Politecnico di Torino, Turin, Italy, 2024, available at https://iris.polito.it/handle/11583/2991352.

[38] G. Beretta, Y. Ardesi, G. Piccinini, and M. Graziano, "vlsi-nanocomputing/SCERPA: SCERPA v4.0.1," 2022.

[39] ——, "Robustness of the In-Plane Data Crossing for Molecular Field-Coupled Nanocomputing," in *IEEE-NANO*, 2023, pp. 732–736.

[40] M. Walter, R. Wille, D. Große, F. Sill Torres, and R. Drechsler, "Placement & Routing for Tile-based Field-coupled Nanocomputing Circuits is NP-complete," in *jetc*, 2019.

[41] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *ISCAS*. IEEE Press, 1985, pp. 677–692.

[42] M. Walter, J. Drewniok, S. Hofmann, B. Hien, and R. Wille, "The Munich Nanotech Toolkit (MNT)," in *IEEE-NANO*, 2024.