Late Breaking Results: Physical Co-Design for Field-coupled Nanocomputing

Simon Hofmann*, Marcel Walter*, and Robert Wille*‡

*Chair for Design Automation, Technical University of Munich, Germany
‡Software Competence Center Hagenberg GmbH, Austria
Email: {simon.t.hofmann, marcel.walter, robert.wille}@tum.de

***Abstract*—*Field-coupled Nanocomputing* (FCN), a class of post-CMOS technologies operating at the nanoscale without the flow of electricity, is becoming a reality due to advancements in simulating and manufacturing logic gates using *Silicon Dangling Bonds* (SiDBs). Efficient physical design methodologies are crucial for the performance, area efficiency, reliability, and manufacturability of FCN circuits. However, despite considerable progress in developing algorithms and tools tailored to FCN physical design, achieving efficient results still requires a co-design approach, necessitating expert manual refinement similar to the CMOS design process. To this end, we introduce a GUI-based tool that combines *both* automation and expert adjustments, enabling designers to easily optimize and modify FCN layouts. To demonstrate its potential, a designer used the tool to reduce the area of the best-known layout for the benchmark circuit *cm82a* by over 15 % in less than a minute. Additionally, the tool is publicly available as open-source at https://github.com/cda-tum/mnt-designer.**

I. Introduction

Field-coupled Nanocomputing (FCN, [1]), which operates by leveraging the repulsion of physical fields instead of electric current, is rapidly transitioning from theoretical research to practical implementation due to recent advancements in the simulation [2]–[4] and manufacturing [5] of logic gates using *Silicon Dangling Bonds* (SiDBs, [6]), heralding a new era in ultra-low-power and high-speed computing devices at the nanoscale.

As FCN technologies become more tangible, the significance of efficient physical design methodologies becomes increasingly paramount. The quality of physical design algorithms not only impacts the performance and area efficiency of FCN circuits but also influences their reliability and manufacturability.

In recent years, considerable progress has been made in developing algorithms and tools tailored to FCN physical design [7]–[17]. These automatic approaches have been instrumental in advancing the field. However, they often encounter limitations when addressing the complex design constraints inherent to FCN, such as planarity and signal balancing. This is reminiscent of the conventional CMOS design flow, where, despite sophisticated automation, expert designers frequently engage in post-processing steps to refine and enhance layouts manually or to fix *Design Rule Violations* (DRVs) [18].

To allow for such a *co-design* in the FCN domain, we introduce a novel, completely open-source GUI-based tool (available as part of the *Munich Nanotech Toolkit* (MNT, [19]) at https://github.com/cda-tum/mnt-designer) that empowers designers to easily optimize and modify FCN layouts. This tool offers technology independence, enabling designers to create gate-level layouts and compile them to the cell level for multiple technological implementations. It provides import and export functionalities for various high-level descriptions and layout formats, integrating state-of-the-art physical design [9]–[13] and post-layout optimization algorithms [17], [20] accessible via an intuitive interface with adjustable parameters. Additionally, it offers DRV and equivalence checking [21], enhancing design reliability—features often lacking in other tools. Designers can fine-tune layouts by repositioning and rewiring individual gates, bridging the gap between automatic design and expert manual refinement, offering a flexible platform for iterative improvement. With these combined advantages, we aim to offer a more flexible and user-friendly tool compared to existing solutions [3], [22]–[26].

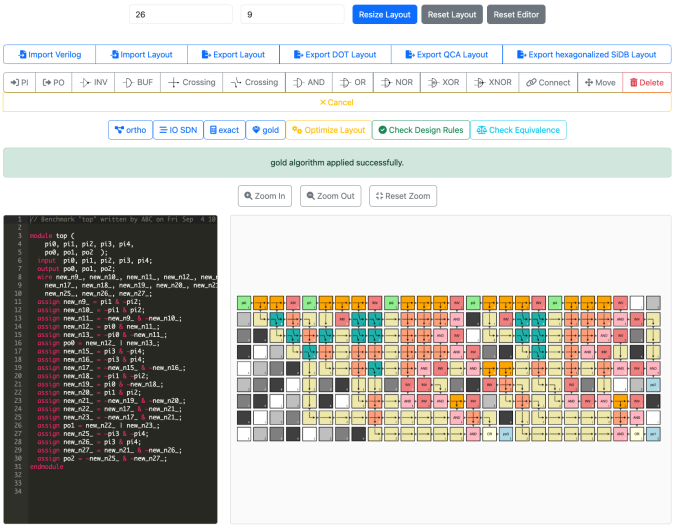
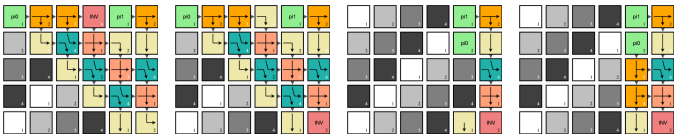


Fig. 1: Tool interface showcasing various features, including the high-level description and generated layout for *cm82a* [27].

II. A Tool for the Co-Design of FCN

This section outlines the main functionalities of the proposed physical design tool that, for the first time, enables a co-design approach for FCN. The tool is introduced using the exemplary benchmark circuit *cm82a* [27]. An overview of its interface is shown in Fig. 1.

The designer begins by importing a high-level description of the circuit, e. g. in Verilog, displayed in the text editor on the left side of the interface. From there, multiple physical design



(a) Cut-out of the layout from Fig. 1 (b) Inverter is moved to another position. (c) Wires from *pi0* to other gates are deleted. (d) *pi0* is rewired to the other gates.

Fig. 2: Optimizing the layout for the *cm82a* function using the proposed design tool.

algorithms are available to generate a gate-level layout. These include *ortho* [11], IO SDN [12], *exact* [9], and *gold* [10], which can be executed via the buttons shown in Fig. 1. By selecting the state-of-the-art algorithm *gold* [10], the tool produces the best-known layout, which is shown on the right.

The designer can then manually refine this layout through direct interaction. In FCN, wire segments contribute equally to area and delay costs as standard gates, making their minimization crucial. While automatic methods can reduce excess wiring by repositioning gates, certain optimizations are best achieved by expert designers. The proposed tool facilitates this via the movement of gates to new positions, the placement of additional gates (e. g., AND, OR, and XNOR), the deletion of unnecessary wiring, and the reconnection of gates to their fanins and fanouts. Throughout this process, the designer can continuously perform DRV-checks to ensure that the design remains functional.

Example. *The manual refinement of the layout from Fig. 1 is illustrated in Fig. 2. In Fig. 2a, an inverter (red tile labeled* INV*) located at the top border of the layout is identified. This inverter is moved along its wire path to the bottom right corner in Fig. 2b. With the complex wiring removed, the primary input in the top left corner (green tile labeled* pi0*) can be repositioned beneath* pi1 *as shown in Fig. 2c. Finally, in Fig. 2d,* pi0 *is reconnected to its three outgoing connections, freeing up four entire columns in the layout.*

Table I compares the area of layouts generated for the benchmark function *cm82a* using different combinations of physical design algorithms from the literature. Starting from the baseline layout produced by *ortho*, several algorithms achieved area reductions up to $81.25\,\%$ compared to said baseline. With the proposed tool, the current best-known layout generated using *gold*, which cannot be improved by automatic post-layout optimization algorithms, was further improved by an additional $15.38\,\%$, resulting in an overall area reduction of $84.13\,\%$ relative to the baseline.

III. Conclusion

This work introduces a GUI-based tool designed to enable physical co-design for FCN. By combining *both* automatic design and expert manual refinement, it empowers designers to optimize and modify layouts more effectively. This was demonstrated by improving the best-known layout generated by a state-of-the-art algorithm for the benchmark circuit *cm82a* by an additional $15.38\,\%$ in less than a minute.

Table I: Layout dimensions for the benchmark circuit *cm82a*.

Algorithm Combination	w	$\times$	h	$=$	A	ΔA
ortho [11]	26	$\times$	48	$=$	1248	$\pm 0\,\%$
IO SDN [12]	24	$\times$	37	$=$	888	$-27.85\,\%$
NPR [15]	25	$\times$	25	$=$	625	$-49.92\,\%$
ortho [11] + PLO [17]	16	$\times$	23	$=$	368	$-70.51\,\%$
NPR [15] + PLO [17]	16	$\times$	23	$=$	368	$-70.51\,\%$
IO SDN [12] + PLO [17]	18	$\times$	19	$=$	342	$-72.60\,\%$
gold [10]	26	$\times$	9	$=$	234	$-81.25\,\%$
gold [10] + Proposed Tool	22	$\times$	9	$=$	**198**	$-84.13\,\%$

NPR, PLO, IO SDN, *gold* and *ortho* are abbreviations for the physical design tools NanoPlaceR [15], Post-Layout Optimization [17], Input Ordering Signal Distribution Network [12], Graph-Oriented Layout Design [10], and the OGD-based heuristic physical design method [11], respectively; w, h and A are the width, height and resulting area of the layout respectively; the final column lists the area difference ΔA between each algorithm combination and the baseline *ortho*.

References

[1] N. Anderson and S. Bhanja, Eds., *Field-Coupled Nanocomputing - Paradigms, Progress, and Perspectives*. Springer, 2014.

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

[3] S. Ng *et al.*, "SiQAD: A Design and Simulation Tool for Atomic Silicon Quantum Dot Circuits," *IEEE TNANO*, vol. 19, pp. 137–146, 2020.

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

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

[6] R. Achal *et al.*, "Lithography for robust and editable atomic-scale silicon devices and memories," *Nat. Commun.*, vol. 9, no. 1, 2018.

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

[8] R. E. Formigoni *et al.*, "A Survey on Placement and Routing for Field-Coupled Nanocomputing," *JICS*, vol. 16, pp. 1–9, 2021.

[9] M. Walter *et al.*, "An Exact Method for Design Exploration of Quantum-dot Cellular Automata," in *DATE*, 2018, pp. 503–508.

[10] S. Hofmann *et al.*, "A* is Born: Efficient and Scalable Physical Design for Field-coupled Nanocomputing," in *IEEE-NANO*, 2024, pp. 80–85.

[11] M. Walter *et al.*, "Scalable Design for Field-Coupled Nanocomputing Circuits," in *ASP-DAC*, 2019, pp. 197–202.

[12] ——, "Versatile Signal Distribution Networks for Scalable Placement and Routing of Field-coupled Nanocomputing Technologies," in *ISVLSI*, 2023.

[13] S. Hofmann *et al.*, "Scalable Physical Design for Silicon Dangling Bond Logic: How a 45° Turn Prevents the Reinvention of the Wheel," in *IEEE-NANO*, 2023, pp. 872–877.

[14] G. Li *et al.*, "A QCA Placement and Routing Algorithm Based on the SA Algorithm," *Int. J. Electron*, 2023.

[15] S. Hofmann *et al.*, "Thinking Outside the Clock: Physical Design for Field-coupled Nanocomputing with Deep Reinforcement Learning," in *ISQED*, 2024.

[16] B. Zhang *et al.*, "Quantum-dot Cellular Automata Placement and Routing with Hierarchical Algorithm," *Nano Commun. Netw.*, vol. 39, p. 100495, 2024.

[17] S. Hofmann *et al.*, "Post-Layout Optimization for Field-coupled Nanotechnologies," in *NANOARCH*, 2023.

[18] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Addison-Wesley Publishing Company, 2010.

[19] M. Walter *et al.*, "The Munich Nanotech Toolkit (MNT)," in *IEEE-NANO*, 2024, pp. 454–459.

[20] S. Hofmann *et al.*, "Late Breaking Results: Wiring Reduction for Field-coupled Nanotechnologies," in *DAC*, 2024.

[21] M. Walter *et al.*, "Verification for Field-coupled Nanocomputing Circuits," in *DAC*, 2020.

[22] K. Walus *et al.*, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata," *TNANO*, vol. 3, no. 1, pp. 26–31, 2004.

[23] F. Riente *et al.*, "MagCAD: Tool for the Design of 3-D Magnetic Circuits," *JXCDC*, vol. 3, pp. 65–73, 2017.

[24] R. E. Formigoni *et al.*, "Ropper: A Placement and Routing Framework for Field-Coupled Nanotechnologies," in *SBCCI*. ACM, 2019.

[25] F. Riente *et al.*, "ToPoliNano: A CAD Tool for Nano Magnetic Logic," *IEEE TCAD*, vol. 36, no. 7, pp. 1061–1074, 2017.

[26] F. Peng *et al.*, "Spars: A Full Flow Quantum-Dot Cellular Automata Circuit Design Tool," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 4, pp. 1233–1237, 2021.

[27] G. Fontes *et al.*, "Placement and Routing by Overlapping and Merging QCA Gates," in *ISCAS*, 2018.