

Late Breaking Results From Hybrid Design Automation for Field-coupled Nanotechnologies

Simon Hofmann*, Marcel Walter*, Lorenzo Servadei*, and Robert Wille*[‡]

*Chair for Design Automation, Technical University of Munich, Germany

[‡]Software Competence Center Hagenberg GmbH, Austria

Email: {simon.t.hofmann, marcel.walter, lorenzo.servadei, robert.wille}@tum.de

Abstract—Recent breakthroughs in atomically precise manufacturing are paving the way for *Field-coupled Nanocomputing* (FCN) to become a real-world post-CMOS technology. This drives the need for efficient and scalable physical design automation methods. However, due to the problem’s NP-completeness, existing solutions either generate designs of high quality, but are not scalable, or generate designs in negligible time but of poor quality. In an attempt to balance scalability and quality, we created and evaluated a hybrid approach that combines the best of established design methods and deep reinforcement learning. This paper summarizes the obtained results.

I. INTRODUCTION

Latest technology advancements in *Field-coupled Nanocomputing* (FCN) [1], e.g., in the form of *Silicon Dangling Bonds* (SiDBs) [2] and multi-million dollar investments by enterprises such as *Quantum Silicon Inc.* emphasize the need for novel approaches to the physical design of FCN to keep pace. However, state-of-the-art solutions for layout mapping of logic functions are at risk of falling behind these rapid domain shifts: Optimal solutions for the placement and routing of a netlist onto a layout can only be obtained for small functions using *exact* approaches with exponential runtime behavior [3]. On the other hand, *scalable* algorithms [4] are tailored toward specific layout topologies, and as such are able to determine solutions for larger input functions, but of sub-par quality.

In an attempt to address this problem, we created and evaluated a solution that utilizes established design automation techniques in a hybrid setup with deep reinforcement learning. To this end, we take advantage of recent developments in the field of machine learning-aided design automation for gate placement [5] in combination with efficient multi-path signal routing based on scalable path finding and fast approximations to graph coloring [6]. In this work, *Proximal Policy Optimization* (PPO) [7] is used to learn the placement of logic elements, which is further accelerated by incorporating an action mask calculated based on the structure of the netlist and the last partial placement, ensuring valid and compact solutions. Furthermore, the routing of placed gates is incorporated directly into the placement step using the aforementioned established routing strategies.

The results (outlined in Section III) further highlight the possibility of solving even larger functions with the presented approach, which is necessary for the emergence of FCN as a post-CMOS technology. An open-source implementation based on the *fiction* framework [8] is available at https://github.com/simonlhofmann/rl_fiction.

II. A HYBRID APPROACH FOR PLACEMENT & ROUTING

In this section, the proposed hybrid placement and routing algorithm, which includes the routing of a placed gate directly in the reinforcement learning loop, is described by means of Figure 1.

1) *Netlist Preparation*: At the beginning of a training cycle, the order of the gates to be placed is obtained via topological sorting. The next gate in the netlist to be placed is marked green.

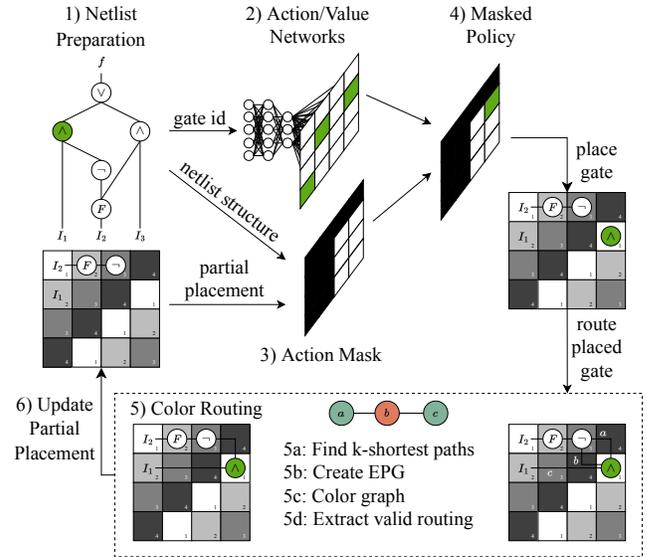


Fig. 1. The novel approach masks out invalid coordinates proposed by the action and value networks to determine promising placements. After each placement, color routing tries to route the new gate with its predecessors.

2) *Action/Value Networks*: A one-hot encoded vector of the next gate id is fed into the action and value networks, which have the same input dimension as the number of gates, two hidden layers of size 64, and an output dimension equal to the number of coordinates in the layout. The output, in the form of a probability distribution, predicts the most promising layout coordinates to place the current gate at. In Figure 1, the three positions with the highest probability are highlighted in green.

3) *Action Masks*: Based on the netlist structure, the current partial placement, the position of preceding and already placed gates, constraints imposed by the layout topology, information flow directions, and validity of the partial placement (e.g., the gate is not trapped, input/output pins are at the border, etc.), an action mask can be created that masks invalid subsequent placements.

4) *Masked Policy*: The action mask sets the probability of coordinates that are invalid for placement to 0, forcing the agent to place the gates only at valid positions. In Figure 1, the action mask allows only one of the three most probable coordinates, resulting in the choice of this position, as it now has the highest probability.

5) *Color Routing*: After each placement step, the placed gate must be connected to its predecessors without conflicting with the existing wiring and without violating the information flow directions. To this end, we employ an established multi-path FCN routing algorithm called *color routing* [6], whose steps are outlined in Figure 1. First, the k shortest paths from the two predecessors to the placed gate

TABLE I
COMPARATIVE EXPERIMENTAL EVALUATION OF THE STATE OF THE ART AGAINST THE PROPOSED HYBRID ALGORITHM.

BENCHMARK CIRCUIT [9], [10]	EXACT APPROACH [3]		HEURISTIC APPROACH [4]		PROPOSED APPROACH		ΔA		
Name	I / O	$ G $	$w \times h = A$	t	$w \times h = A$	t	$w \times h = A$	t	ΔA
2:1 MUX	3 / 1	9	$3 \times 4 = \mathbf{12}$	< 1	$6 \times 8 = 48$	< 1	$3 \times 4 = \mathbf{12}$	< 1	-75 %
XOR	2 / 1	9	$3 \times 6 = \mathbf{18}$	< 1	$5 \times 8 = 40$	< 1	$3 \times 6 = \mathbf{18}$	< 1	-55 %
XNOR	2 / 1	11	$3 \times 6 = \mathbf{18}$	< 1	$6 \times 9 = 54$	< 1	$3 \times 6 = \mathbf{18}$	4	-67 %
Half Adder	2 / 2	14	$4 \times 6 = \mathbf{24}$	< 1	$9 \times 10 = 90$	< 1	$4 \times 6 = \mathbf{24}$	4	-73 %
Parity Gen.	3 / 1	18	$4 \times 8 = \mathbf{32}$	< 1	$9 \times 14 = 126$	< 1	$8 \times 8 = 64$	2	-49 %
Parity Check.	4 / 1	26	$6 \times 8 = \mathbf{48}$	2	$12 \times 20 = 240$	< 1	$11 \times 11 = 121$	9	-50 %
clpl	11 / 5	30	$3 \times 15 = \mathbf{45}$	1	$18 \times 26 = 468$	< 1	$12 \times 12 = 144$	31	-69 %
XOR5_R1	5 / 1	40	$7 \times 11 = \mathbf{77}$	11	$14 \times 33 = 462$	< 1	$15 \times 15 = 225$	26	-51 %
cm82a	5 / 3	68	<i>timeout limit reached</i>		$26 \times 51 = 1326$	< 1	$25 \times 25 = \mathbf{625}$	210	-53 %
2bitAdderMaj	5 / 2	82	<i>timeout limit reached</i>		$26 \times 64 = 1664$	< 1	$29 \times 29 = \mathbf{841}$	180	-49 %
xor5Maj	5 / 1	102	<i>timeout limit reached</i>		$30 \times 79 = 2370$	< 1	$40 \times 40 = \mathbf{1600}$	2520	-32 %
parity	16 / 1	150	<i>timeout limit reached</i>		$48 \times 120 = 5760$	< 1	$50 \times 50 = \mathbf{2500}$	3550	-57 %

Runtime values are in seconds; the timeout limit is 24 h; w , h and A are the width, height and resulting area of the layout respectively; the final column compares the area difference ΔA between the heuristic and the proposed hybrid approach.

are enumerated. From these paths, an *edge intersection graph* (EPG) is created, and colored using an efficient approximation to vertex coloring. A valid routing can be extracted from the said coloring. For the experiments in Section III, k has been set to 50.

6) *Update Partial Placement*: The agent receives a reward of +1 only if the routing is successful, i. e., conflict-free. Additionally, the reward is scaled according to the resulting layout area as a synchronous goal to penalize area overhead. Furthermore, if the routing step was successful, the agent utilizes the updated partial placement to compute the next action mask for subsequent gates. Otherwise, it attempts a different placement and returns to step 5. Depending on the number of placement steps, the action and value networks are updated using PPO [7]. The validity of partial placements is constantly monitored to prematurely terminate unpromising runs, thus facilitating the initiation of a fresh layout. This practice improves the efficacy of the proposed approach.

III. EXPERIMENTS

This section presents the results obtained by the proposed hybrid physical design algorithm using a set of benchmark circuits that are common in the domain [9], [10] and compares its effectiveness against the state of the art [3], [4]. Table I summarizes the obtained results. Due to the NP-completeness of the physical design problem, the exact approach [3] is constrained by its exponential runtime behavior. Consequently, we applied a time-out limit of 24 h, which was already reached when applied to the moderately-sized *cm82a* benchmark, which comprises 68 gates. At the same time, the heuristic approach can handle that in negligible time, but requires almost a factor of 6 more area. In contrast, the proposed hybrid approach provides a balance between efficiency and quality. Notably, even the challenging *parity* function, encompassing 150 gates, can be effectively solved in under 1 h, achieving a significant layout reduction of 57 %.

The findings confirm the previously raised issues pertaining to the subpar scalability of the exact approach and the unsatisfactory quality of the heuristic approach. However, by balancing the contradicting demands of scalability and quality, the proposed approach seamlessly integrates the strengths of both methods, thus paving the way for future investigations in the domain of design automation for FCN. Impressively, across all functions in the two benchmark sets presented in Table I, the proposed hybrid approach yields a substantial area reduction, ranging from 32 % to 75 %, with an average improvement of 57 % relative to the heuristic method.

IV. CONCLUSION

As *Field-coupled Nanocomputing* is becoming a reality, efficient physical design automation methods are needed to adapt to evolving technology. In this work, we present the first results of such an approach to overcome the bottleneck of today’s state of the art and provide a balance between quality and scalability. The proposed hybrid algorithm for FCN physical design demonstrates remarkable proficiency in obtaining valid circuit layouts for logic functions of up to 150 gates within 1 h. Notably, while optimal solutions can only be obtained for functions with a maximum of approximately 45 gates, the proposed approach facilitates the generation of layouts for all circuits in the benchmark sets, underscoring its versatility and broad applicability. The amalgamation of deep reinforcement learning with established physical design strategies culminates in a powerful tool for the creation of compact and valid FCN circuit layouts. Prospective endeavors will center on the extension of this approach to diverse layout topologies and clocking schemes, enabling rigorous comparisons with other heuristics [9], [10], and the scaling of the approach to accommodate functions comprising thousands of gates. Intriguingly, the proposed approach can also be harnessed for the design of sequential circuits. This LBR paper serves as an initial step toward the efficient creation of circuit layouts for FCN.

REFERENCES

- [1] N. G. Anderson and S. Bhanja, Eds., *Field-Coupled Nanocomputing - Paradigms, Progress, and Perspectives*. Springer, 2014.
- [2] R. Achal *et al.*, “Lithography for robust and editable atomic-scale silicon devices and memories,” *Nat. Commun.*, vol. 9, no. 1, p. 2778, 2018.
- [3] M. Walter, R. Wille, D. Große, F. S. Torres, and R. Drechsler, “An Exact Method for Design Exploration of Quantum-dot Cellular Automata,” in *DATE*, 2018, pp. 503–508.
- [4] M. Walter, R. Wille, F. S. Torres, D. Große, and R. Drechsler, “Scalable Design for Field-Coupled Nanocomputing Circuits,” in *ASP-DAC*, 2019, pp. 197–202.
- [5] A. Mirhoseini *et al.*, “A graph placement methodology for fast chip design,” *Nature*, vol. 594, no. 7862, pp. 207–212, 2021.
- [6] M. Walter and R. Wille, “Efficient Multi-Path Signal Routing for Field-coupled Nanotechnologies,” in *NANOARCH*, 2022.
- [7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *CoRR*, vol. abs/1707.06347, 2017.
- [8] 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,” 2019.
- [9] A. Trindade, R. Ferreira, J. A. M. Nacif, D. Sales, and O. P. V. Neto, “A Placement and routing algorithm for Quantum-dot Cellular Automata,” in *SBCCI*, 2016, pp. 1–6.
- [10] G. Fontes, P. A. R. L. Silva, J. A. M. Nacif, O. P. V. Neto, and R. Ferreira, “Placement and Routing by Overlapping and Merging QCA Gates,” in *ISCAS*, 2018, pp. 1–5.