MNT Bench:
Benchmarking Software and Layout Libraries 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*—As *Field-coupled Nanocomputing* (FCN) gains traction as a viable post-CMOS technology, the EDA community lacks public benchmarks to evaluate the performance of academic and commercial design tools. We propose *MNT Bench* to address this gap by providing a platform for researchers to compare algorithms across a diverse set of benchmarks generated by multiple physical design tools. These benchmarks span various clocking schemes and gate libraries, with *MNT Bench* being consistently updated to integrate the latest advancements in the field. In fact, using *MNT Bench*, we were able to provide layouts that are substantially better (in terms of area) than everything the community generated thus far.**

I. Introduction

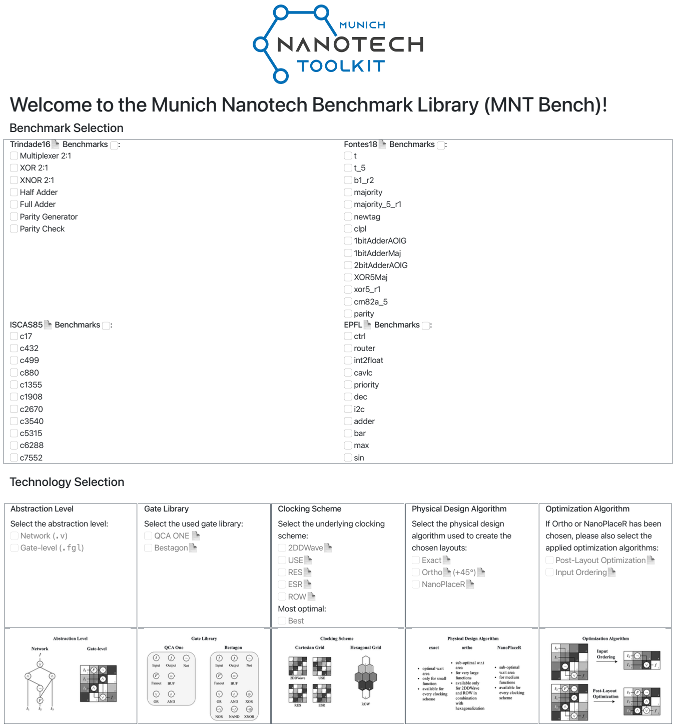
Since the recent experimental demonstration of a working nanoscale OR gate [1] with a footprint of less than $30\,\mathrm{nm}^2$ implemented using *Silicon Dangling Bonds* (SiDBs) [2] as a *Field-coupled Nanocomputing* (FCN) technology on a hydrogen-passivated silicon surface [3], new physical design methods [4]–[7] and optimization algorithms [8], [9] have been developed to improve the gate-level layouts suitable for FCN.

To further foster the collaboration between different research groups and facilitate the access to state-of-the-art layouts for benchmarking, simulation, and fabrication, this work introduces *MNT Bench* as part of the *Munich Nanotech Toolkit* (MNT), whose web interface, as seen in Figure 1, is available online,[1] as a pip package,[2] and as an open-source GitHub repository.[3] *MNT Bench* offers a wide range of gate-level layouts generated on top of different underlying clocking schemes using multiple gate libraries, physical design algorithms, and optimizations in combination with the network descriptions in verilog-format to serve as a benchmark for the development of new methodologies in the area of design automation for FCN.

II. MNT Bench

MNT Bench encompasses five major contributions:

1) Establishment of a website providing convenient access to benchmark files and tracking FCN domain advancements through regular updates.



Welcome to the Munich Nanotech Benchmark Library (MNT Bench)!

Benchmark Selection

Trindade16 Benchmarks □:
□ Multiplexer 2:1
□ XOR 2:1
□ XNOR 2:1
□ Half Adder
□ Full Adder
□ Parity Generator
□ Parity Check

ISCAS85 Benchmarks □:
□ c17
□ c432
□ c499
□ c880
□ c1355
□ c1908
□ c2670
□ c3540
□ c5315
□ c6288
□ c7552

Fontes18 Benchmarks □:
□ t
□ t_5
□ b1_r2
□ majority
□ majority_5_r1
□ newtag
□ clpl
□ 1bitAdderAOIG
□ 1bitAdderMaj
□ 2bitAdderAOIG
□ XOR5Maj
□ xor5_r1
□ cm82a_5
□ parity

EPFL Benchmarks □:
□ ctrl
□ router
□ int2float
□ cavlc
□ priority
□ dec
□ i2c
□ adder
□ bar
□ max
□ sin

Technology Selection

Abstraction Level
Select the abstraction level:
□ Network (.v)
□ Gate-level (.fgl)

Gate Library
Select the used gate library:
□ QCA ONE
□ Bestagon

Clocking Scheme
Select the underlying clocking scheme:
□ 2DDWave
□ USE
□ RES
□ ESR
□ ROW
Most optimal:
□ Best

Physical Design Algorithm
Select the physical design algorithm used to create the chosen layouts:
□ Exact
□ Ortho (+45°)
□ NanoPlaceR

Optimization Algorithm
If Ortho or NanoPlaceR has been chosen, please also select the applied optimization algorithms:
□ Post-Layout Optimization
□ Input Ordering

Physical Design Algorithm
exact
- optimal w.r.t area
- only for small function
- available for every clocking scheme

ortho
- sub-optimal w.r.t area
- for very large functions
- available only for 2DDWave and ROW in combination with hexagonalization

NanoPlaceR
- sub-optimal w.r.t area
- for medium functions
- available for every clocking scheme

Figure 1: *MNT Bench* provides an intuitive web interface, facilitating the selection of desired benchmark functions and enabling users to apply filters based on their requirements.

2) Generation of layouts for various clocking schemes and gate libraries utilizing available physical design and optimization algorithms. These can be filtered according to the user's criteria on the website, allowing new design automation tools to be objectively benchmarked.
3) Generation of the best layouts in terms of area for multiple benchmarks using the optimal combination of design automation tools for each function, which can be downloaded for benchmarking, simulation, or fabrication.
4) Development and implementation of a novel gate-level file format (*.fgl*), which offers a standardized and human-readable representation of FCN layouts.

[1] https://www.cda.cit.tum.de/mntbench
[2] https://pypi.org/project/mnt.bench
[3] https://github.com/cda-tum/mnt-bench

Table I: Most efficient layouts w.r.t. area discovered thus far for multiple benchmarks sets, available in *MNT Bench*.

BENCHMARK				QCA ONE [15] GATE LIBRARY							BESTAGON [16] GATE LIBRARY						
Set	Name	I / O	$\|N\|$	$w \times h$	=	A	t	Algorithm	Clk. Scheme	ΔA	$w \times h$	=	A	t	Algorithm	Clk. Scheme	ΔA
Trindade16 [11]	2:1 MUX	3 / 1	4	3 × 4	=	12	< 1	exact	2DDWave	±0 %	3 × 5	=	15	< 1	exact	ROW	−16.7 %
	XOR	2 / 1	4	4 × 4	=	16	< 1	exact	RES	±0 %	2 × 3	=	6	< 1	exact	ROW	±0 %
	XNOR	2 / 1	6	3 × 5	=	15	< 1	exact	2DDWave	−6.3 %	2 × 3	=	6	< 1	exact	ROW	±0 %
	Half Adder	2 / 2	6	4 × 5	=	20	< 1	exact	USE	−16.7 %	3 × 5	=	15	< 1	exact	ROW	±0 %
	Full Adder	3 / 2	5	5 × 11	=	55	< 1	exact	2DDWave	−21.4 %	3 × 9	=	27	< 1	exact	ROW	±0 %
	Parity Gen.	3 / 1	10	4 × 7	=	28	< 1	exact	ESR	±0 %	3 × 4	=	12	< 1	exact	ROW	±0 %
	Parity Check.	4 / 1	15	4 × 11	=	44	2	exact	2DDWave	−8.3 %	4 × 5	=	20	< 1	exact	ROW	−28.6 %
Fontes18 [12]	t	5 / 2	11	4 × 7	=	28	< 1	exact	2DDWave	−6.7 %	5 × 8	=	40	< 1	exact	ROW	±0 %
	b1_r2	3 / 4	12	4 × 10	=	40	2	exact	2DDWave	±0 %	4 × 7	=	28	1	exact	ROW	±0 %
	majority	5 / 1	17	5 × 7	=	35	1	exact	2DDWave	−22.2 %	5 × 9	=	45	< 1	exact	ROW	−18.2 %
	newtag	8 / 1	17	4 × 10	=	40	1	exact	2DDWave	−9.1 %	8 × 9	=	72	< 1	exact	ROW	±0 %
	clpl	11 / 5	10	2 × 19	=	38	70	exact	RES	±0 %	11 × 16	=	176	< 1	exact	ROW	±0 %
	1bitAdderAOIG	3 / 2	15	5 × 10	=	50	6	exact	USE	±0 %	3 × 9	=	27	< 1	exact	ROW	±0 %
	1bitAdderMaj	3 / 1	29	3 × 6	=	18	< 1	exact	2DDWave	−85.7 %	3 × 7	=	21	< 1	exact	ROW	±0 %
	2bitAdderMaj	5 / 2	54	5 × 8	=	40	36	exact	USE	−93.8 %	5 × 12	=	60	< 1	exact	ROW	±0 %
	xor5Maj	5 / 1	70	8 × 11	=	88	629	exact	2DDWave	−93.2 %	5 × 6	=	30	< 1	exact	ROW	±0 %
	cm82a_5	5 / 3	42	16 × 17	=	272	57	NPR, PLO	2DDWave	−24.7 %	5 × 14	=	70	20	exact	ROW	−6.7 %
	parity	16 / 1	103	32 × 34	=	1088	< 1	ortho, InOrd (SDN), PLO	2DDWave	−44.5 %	9 × 22	=	198	< 1	ortho, InOrd (SDN), 45°, PLO	ROW	−68.3 %
ISCAS85 [13]	c17	5 / 2	8	4 × 7	=	28	< 1	exact	2DDWave	±0 %	5 × 8	=	40	< 1	exact	ROW	±0 %
	c432	37 / 7	414	120 × 266	=	31920	< 1	ortho, InOrd (SDN)	2DDWave	−62.4 %	119 × 303	=	36057	< 1	ortho, InOrd (SDN), 45°	ROW	−50.1 %
	c499	41 / 32	816	371 × 687	=	254877	< 1	ortho, InOrd (SDN)	2DDWave	−12.1 %	163 × 435	=	70905	< 1	ortho, InOrd (SDN), 45°	ROW	−15.5 %
	c880	60 / 26	639	266 × 621	=	165186	< 1	ortho, InOrd (SDN)	2DDWave	−10.8 %	267 × 588	=	156996	< 1	ortho, InOrd (SDN), 45°	ROW	−19.4 %
	c1355	41 / 32	1064	365 × 701	=	255865	< 1	ortho, InOrd (SDN)	2DDWave	−43.7 %	171 × 417	=	71307	< 1	ortho, InOrd (SDN), 45°	ROW	−15.0 %
	c1908	33 / 25	813	322 × 693	=	223146	< 1	ortho, InOrd (SDN)	2DDWave	−22.4 %	225 × 496	=	111600	< 1	ortho, InOrd (SDN), 45°	ROW	−30.9 %
	c2670	233 / 64	1463	473 × 1166	=	551518	< 1	ortho, InOrd (SDN)	2DDWave	−47.0 %	499 × 1061	=	529439	< 1	ortho, InOrd (SDN), 45°	ROW	−31.1 %
	c3540	50 / 22	1987	723 × 1744	=	1260912	< 1	ortho, InOrd (SDN)	2DDWave	−28.3 %	814 × 1720	=	1400080	< 1	ortho, InOrd (SDN), 45°	ROW	−27.4 %
	c5315	178 / 123	3628	1137 × 2715	=	3086955	< 1	ortho, InOrd (SDN)	2DDWave	−47.7 %	1230 × 2535	=	3118050	1	ortho, InOrd (SDN), 45°	ROW	−39.0 %
	c6288	32 / 32	6467	1330 × 5714	=	7599620	< 1	ortho, InOrd (SDN)	2DDWave	±0 %	1248 × 2883	=	3597984	< 1	ortho, InOrd (SDN), 45°	ROW	−13.2 %
	c7552	207 / 107	4501	1330 × 3267	=	4345110	< 1	ortho, InOrd (SDN)	2DDWave	−45.3 %	1271 × 2618	=	3327478	1	ortho, InOrd (SDN), 45°	ROW	−21.7 %
EPFL [14]	ctrl	7 / 25	409	80 × 164	=	13120	< 1	ortho, InOrd (SDN)	2DDWave	−78.7 %	84 × 203	=	17052	< 1	ortho, InOrd (SDN), 45°	ROW	−69.5 %
	router	60 / 3	490	103 × 212	=	21836	< 1	ortho, InOrd (SDN)	2DDWave	−80.6 %	111 × 245	=	27195	< 1	ortho, InOrd (SDN), 45°	ROW	−76.4 %
	int2float	11 / 7	545	155 × 362	=	56110	< 1	ortho, InOrd (SDN)	2DDWave	−55.9 %	169 × 375	=	63375	< 1	ortho, InOrd (SDN), 45°	ROW	−45.4 %
	cavlc	10 / 11	1600	484 × 1149	=	556116	< 1	ortho, InOrd (SDN)	2DDWave	−40.4 %	549 × 1156	=	634644	< 1	ortho, InOrd (SDN), 45°	ROW	−33.1 %
	priority	128 / 8	2349	479 × 684	=	327636	< 1	ortho, InOrd (SDN)	2DDWave	−81.1 %	352 × 937	=	329824	< 1	ortho, InOrd (SDN), 45°	ROW	−84.6 %
	dec	8 / 256	320	418 × 466	=	194788	< 1	ortho, InOrd (SDN)	2DDWave	±0 %	425 × 892	=	379100	< 1	ortho, InOrd (SDN), 45°	ROW	−39.7 %
	i2c	136 / 127	2728	774 × 1573	=	1217502	< 1	ortho, InOrd (SDN)	2DDWave	−64.4 %	886 × 1880	=	1665680	< 1	ortho, InOrd (SDN), 45°	ROW	−64.9 %
	adder	256 / 129	2541	893 × 2169	=	1936917	< 1	ortho, InOrd (SDN)	2DDWave	−19.2 %	512 × 1659	=	849408	< 1	ortho, InOrd (SDN), 45°	ROW	−49.8 %
	bar	135 / 128	6672	2314 × 6193	=	14330602	< 1	ortho, InOrd (SDN)	2DDWave	−12.4 %	2964 × 6470	=	19177080	6	ortho, InOrd (SDN), 45°	ROW	−2.9 %
	max	512 / 130	6110	2461 × 6607	=	16259827	1	ortho, InOrd (SDN)	2DDWave	−11.3 %	2370 × 5982	=	14177340	9	ortho, InOrd (SDN), 45°	ROW	−15.1 %
	sin	24 / 25	11437	3900 × 9079	=	35408100	2	ortho, InOrd (SDN)	2DDWave	−19.5 %	4085 × 8707	=	35568095	9	ortho, InOrd (SDN), 45°	ROW	−10.5 %

Runtime values are in seconds; I, O and N are the number of inputs, outputs, and nodes in the unoptimized benchmark function, respectively; w, h and A are the width, height, and resulting area (in tiles) of the layouts, respectively. NPR, PLO, InOrd (SDN), 45°, exact, and ortho are abbreviations for the physical design tools NanoPlaceR [5], Post-Layout Optimization [9], Input Ordering Signal Distribution Network [8], Hexagonalization [7], and the SMT-based exact [4] and OGD-based heuristic [6] physical design methods, respectively; ΔA compares the layout area with the previous state of the art.

5) Integration of robust read and write utilities for the new file format into the open-source tool *fiction* [10] as part of the MNT.

In addition to generating layouts for all feasible combinations of gate libraries, clocking schemes, physical design algorithms, and optimizations, *MNT Bench* goes a step further by providing the most efficient gate-level layouts in terms of area discovered thus far for commonly encountered benchmarks in the domain [11]–[14]. The area of these layouts, along with the corresponding clocking scheme and physical design algorithm utilized in their creation, is detailed in Table I. Here, ΔA represents the achieved layout area reduction through optimal combinations of physical design tools.

In contrast to the previous state of the art, these layouts require significantly less area, e. g. for the *router* function using the *Bestagon* library, only 23.6 % compared to [7].

III. Conclusion

To support the recent developments in FCN, *MNT Bench* offers public benchmarks to evaluate the performance of academic and commercial design tools. On top of a new file standard for FCN gate-level layouts and read/write utilities integrated into *fiction*, *MNT Bench* provides the most efficient layouts w.r.t. to area discovered thus far, setting a benchmark for the physical design of FCN layouts. To keep track of further developments, *MNT Bench* is constantly updated and maintained by the Chair for Design Automation at the Technical University of Munich. Improved layouts can be sent to nanotech.cda@xcit.tum.de for inclusion.

References

[1] T. Huff *et al.*, "Binary atomic silicon logic," *Nat. Electron.*, vol. 1, no. 12, pp. 636–643, 2018.

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

[3] M. Haider *et al.*, "Controlled Coupling and Occupation of Silicon Atomic Quantum Dots at Room Temperature," *Phys. Rev. Lett.*, vol. 102, p. 046805, 2009.

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

[5] S. Hofmann *et al.*, "Late Breaking Results From Hybrid Design Automation for Field-coupled Nanotechnologies," in *DAC*, 2023, pp. 1–6.

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

[7] 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.

[8] M. Walter *et al.*, "Versatile Signal Distribution Networks for Scalable Placement and Routing of Field-coupled Nanocomputing Technologies," in *ISVLSI*, 2023, pp. 1–6.

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

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

[11] A. Trindade *et al.*, "A Placement and Routing Algorithm for Quantum-dot Cellular Automata," in *SBCCI*, 2016, pp. 1–6.

[12] G. Fontes *et al.*, "Placement and Routing by Overlapping and Merging QCA Gates," in *ISCAS*, 2018, pp. 1–5.

[13] F. Brglez *et al.*, "A neutral netlist of 10 combinational benchmark circuits and a targeted translator in FORTRAN," in *ISCAS*, June 1985.

[14] L. G. Amarù *et al.*, "The EPFL Combinational Benchmark Suite," in *IWLS*, 2015.

[15] D. Reis *et al.*, "A Methodology for Standard Cell Design for QCA," in *ISCAS*, 2016, pp. 2114–2117.

[16] M. Walter *et al.*, "Hexagons Are the Bestagons: Design Automation for Silicon Dangling Bond Logic," in *DAC*, 2022, pp. 739–744.