Exploring Standard-Cell Designs for Reconfigurable Nanotechnologies: A Formal Approach

Michael Raitza, Steffen Märcker, Shubham Rai, and Akash Kumar
Technische Universität Dresden, Germany
E-mail: {michael.raitza, steffen.maercker, shubham.rai, akash.kumar}@tu-dresden.de

***Abstract*—Standard-cell design has always been a craft, and common field-effect transistors span only a small design space. This has changed with reconfigurable transistors. Boolean functions that exhibit multiple dual product-terms in their sum-of-product form yield various beneficial circuit implementations with reconfigurable transistors. In this work, we present an approach to automatically generate these implementations through a formal modeling approach. Using the 3-input XOR function as an example, we discuss the variations and show how to quantify properties like worst-case delay and power dissipation, as well as averages of delay and energy consumption per operation over different scenarios. The quantification runs fully automated on charge transport network models employing probabilistic model checking. This yields exact results instead of approximations obtained from experiments and sampling. The highlight of our work is that the proposed approach provides a comprehensive early technology evaluation flow.**

***Index Terms*—Circuit analysis, formal verification, nanoelectronics, probabilistic model checking, probability, quantitative analysis, reconfigurable logic, semiconductor device modeling**

I. Introduction

Emerging technologies have shown numerous benefits over established CMOS devices, especially in the low-power domain. Research includes mixed-dimensional heterostructures [1], 2-D graphene devices [2], [3] and reconfigurable transistors [4]–[8]. With emerging devices establishing themselves to underpin present or future electronic systems, analysis of circuit topologies is highly imperative. Transistor device characteristics strongly influence the optimal layout of standard cells and their applicability in a VLSI design. For instance, imbalances in p- and n-channel current strengths lead to VLSI designs that favor either OR-type behavior or AND-type behavior. Reconfigurable transistors have widened the design space of standard cells considerably, e. g., 8 topological different single-stage implementations of the Boolean function 3-MIN have been presented in [9]. For Boolean functions with certain properties, transistor-level reconfiguration has a strong influence on the standard cell layout and performance characteristics. Thus, it is valuable to have analysis tools and methodologies at hand that allow early evaluation of device influences on standard-cell designs. State-of-the-art modeling and analysis techniques, though, need a *technology readiness level* (TRL) of 4 or 5 (technology development stage) to provide useful results. They also do not provide the means to do an automated comprehensive analysis on implementation variants.

This research was supported in part by the German Research Foundation (DFG), Project SecuReFET (PN: 439891087).

We present an approach that is based on formal methods, in which we model both the transistor and the circuit, from very little input data, and that allows us to do quantitative analysis. The input device characteristics may come from finite element method simulations.

Our approach relies on a discrete charge transport network model for the electrical connections, which was similarly shown for SPICE simulation [10], and a flexible transistor model that captures the devices' interactions with its environment in the current time step. The model employs simple closed-form equations that are able to capture the non-linear effects of semiconductors instead of sets of significantly more complex differential equations. We completely separate the model from the implementation of experiment stimuli. Hence, we can reason about arbitrary non-deterministic or stochastic input stimulation by using suitable stimuli generators with the same network model. The abstract model facilitates the application of *probabilistic model checking* (PMC) [11], [12]. This enables us to directly compute extremal values, like maximum power dissipation or worst-case propagation delay. We can also compute expected values for metrics under stochastic input patterns, which can characterize application behavior, namely average delay or energy consumption per operation [9]. In contrast to formal methods such as PMC, simulation-based approaches generally rely on experimentation, sampling and insights to generate the relevant stimuli for the experiment.

In this work, we target the 3-XOR function to determine the impact of transistor device characteristics on standard cell structure and its performance, and a projected suitability in a process design kit (PDK). This function is particularly interesting, because it is usually costly to implement in contemporary CMOS technology. Every output value depends on every input signal regardless of their values, i. e., it is representable only as a full binary decision tree. However, in its sum-of-products (SOP) form, the 3-XOR function has numerous product-terms that are pairwise dual. This can be exploited with transistor reconfiguration, because implementing one product-term with reconfigurable transistors also implements its dual term, which is selected via said reconfiguration.

The transistor device of choice is a Germanium-based reconfigurable field effect transistor (RFET) described in [13]. The device can be reconfigured between p- and n-type carrier transport dynamically via an additional input [4]–[7], [13]. An extended fabrication enables the placement of multiple transistor gates on a single device, exciting an intrinsic AND

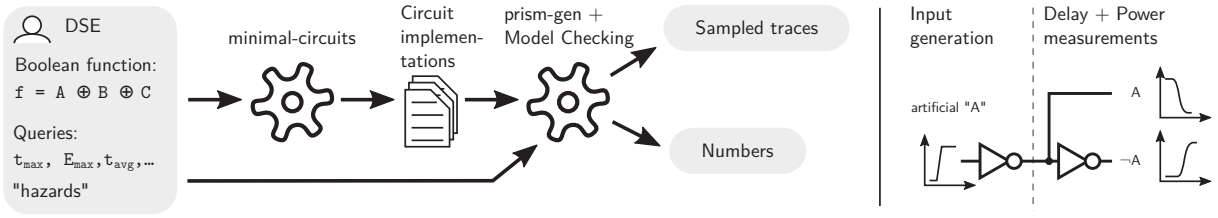

Fig. 1. *Left* The general workflow used in this work. A single Boolean function results in multiple circuits each of which is transformed into a charge transport network model. Instead of running a test protocol, the model is checked against queries, yielding exact numbers or sampled traces. *Right* Artificial inputs are transformed into signals with realistic shapes and slopes before being supplied to the circuit under test. Both inverters are modeled, but only the second is considered by delay and power quantification.

behavior, as shown for Silicon-based technology in [8], [14]. The use of this extended device is surmounted by [15], [16], which show improved signal integrity in dynamic logic gates, using multiple-gate devices and by [17], displaying non-traditional circuit design styles, such as reversible and asynchronous logic. The abstract model of this GeNW RFET is thoroughly described in our work in [9]. Its notable device characteristics under scrutiny include the strength of drain currents depending on device polarity, the effect of transistor reconfiguration and the difference between switching a Schottky barrier gate or a gate placed in the middle of the nanowire.

II. Workflow

Designers working on a PDK need to investigate the standard cells for the intended Boolean functions. Reconfigurable standard cells turn out to have varied performance characteristics, which makes the overall PDK perform best if it is tailored to the intended application. Our tool *minimal-circuits* generates complementary single-stage logic gates from a Python lambda expression of the Boolean function that shall be investigated. Viable implementations are selected according to structural criteria and combinations of input signals connected to the transistors described in Section III. Depending on the characteristics of the Boolean function, this may yield many implementations, e. g., 49 are determined for the 3-XOR function. *minimal-circuits* outputs the implementations in form of charge transport network models specified in the domain-specific language (DSL) of our transpiler *prism-gen*; see also the first step in Figure 1. See [18] for a formal definition of the model.

Our DSL provides transistor devices as library elements that were calibrated in [9] against FEM simulations. To perform the analyses described in Section III, it also provides various input generators as network templates that convert the input stimuli from digital to analogue signals. After matching a circuit implementation with a generator template, the resulting charge transport model is transpiled into a Markovian model in the input language of the probabilistic model checker PRISM [19].

One component of the analysis is the circuit model including the proper input generator, the other is a set of suitable queries formalizing the metrics of interest. The model checker computes the results fully automatically while covering all possible evolutions of network states in the model. The queries, used in this work, are specified and supplied as input files by our tools. A DSL to formulate queries against charge transport models, that stays in the domain of electrical signals and time, is left for a future extension. Meanwhile, new queries can be formulated in PRISM's query language.

Based on the discretization parameters, PRISM builds each circuit model and checks it against the selected queries, yielding numbers for power dissipation, delay and energy consumption. Furthermore, it also delivers all network states witnessing the result. As we show in Section III-D on abnormal situations, this enables designers to further investigate specific circuit behavior. The returned network states can be fed back into PRISM to obtain exact traces of arbitrary length for all inner circuit components. In Section III-D, this yielded the shapes of output hazards that turned out to be a dominating effect for a particular circuit implementation.

The workflow is described in detail in our work in [9].

III. Example Analysis of the 3-XOR Function

Our findings in [9] confirm that the different implementation variants of standard cells indeed offer specific trade-offs which in turn enable us to tailor a circuit closer to an application. To lift this potential, we envision to enhance current EDA techniques with PDKs that are able to capture the nuanced performance characteristics and sheer bandwidth of implementation variants. This will allow EDA tools to make informed choices when picking a particular implementation in the scope of a VLSI design.

For the remainder of this work, we use the 3-XOR function to generate and analyze circuits of interest around it. This function is particularly interesting because of its usability in computation circuits and because it contains many dual product-terms in its SOP form. Dual product-terms allow us to use reconfigurable transistors to implement both product-terms with a single device and select between them via reconfiguration.

We use a model of a Germanium nanowire-based Schottky MIGFET, which we described in [9]; a technology node of 24 nm with multiple independent gates on a reconfigurable nanowire, $V_{DD} = 1.2\,\mathrm{V}$, $V_{th} = 0.4\,\mathrm{V}$. Figure 1 (right) shows our setup for the input generation which is well-known from SPICE simulation. Realistic input signals to the circuit are obtained by inverting artificial input signals with a linear full swing transient of 100 fs. A second inverter generates the inverse input signal and contributes to both the momentary power dissipation and the delay of the circuit under test. The transistors of the circuit

and the inverters are fully modeled and have equal electrical characteristics to emphasize the circuit's sensitivity to the input signal patterns.

A Boolean function can be decomposed into two sub-functions for each input. We call the selection of a subordinate function with respect to an input *explicit reconfiguration*. In contrast, *implicit reconfiguration* is the implementation of a product term and its dual with a single RFET by reconfiguring between n- and p-type behavior. In particular, 3-XOR enables the explicit reconfiguration between 2-XOR and 2-XNOR, and its implementations can exploit implicit reconfiguration, because 3-XOR is self-dual.

Firstly, it is safe to say that, for the notion of (explicit) *reconfiguration* being distinct from mere computation, reconfiguration takes place orders of magnitude less often than computation in general. Hence, we quantify circuit behavior separately in two scenarios: *switching all inputs* and *switching 2 out of 3 inputs*.

Secondly, most electronic devices do not run at peak performance of their technology node or run in a low-energy environment, e. g., IoT devices. With enough slack to tolerate a worse worst-case delay, average delay and energy consumption become the much more valuable metrics to judge circuit quality.

Thirdly, a circuit's structure influences its sensitivity to its output load, which is why PDKs usually have slower but stronger variants of circuits in their portfolio as well.

Hence, we consider the following metrics under various output loads in both scenarios:

- Worst-case delay, momentary power dissipation and energy consumption,
- Average delay and energy consumption per operation.

A. Generating All Minimal Circuits

Our tool *minimal-circuits* generates 49 circuits for the 3-XOR function. These circuits are single-stage CMOS designs and they can be distinguished in three separate dimensions.

a) The variant: describes the circuit structure with respect to implicit reconfiguration. We have fully reconfigurable structures, where each transistor is reconfigured in an input signal, partially reconfigurable structures where some transistors employ reconfigurations and other do not, and a (minimal) static implementation without any reconfiguration. The number of structural variants depends on the Boolean function and amounts to 8 variants for 3-XOR.

b) The reconfiguration type: can apply to a certain variant and describes which inputs take part in circuit reconfiguration. Reconfiguration is typed into being fully balanced over all inputs (*bl*), balanced but with preference given to a certain input (*blA*, *blB* and *blC*), and restricted to a particular input (*A*, *B* and *C*).

c) The reconfiguration mode: describes how reconfiguration is implemented. A transistor, whose outer Schottky barrier gates are never used an input other than the reconfiguration signal itself, is called inner mode. In single mode, a single Schottky barrier gate is used by an input signal other than the reconfiguration input. Lastly, in transmission gate mode, the signals at both Schottky barrier gates and the source contact are pairwise distinct.

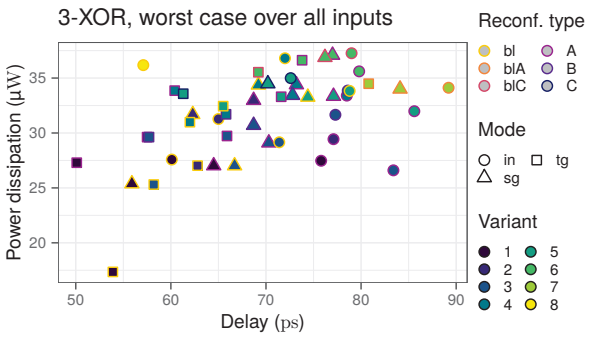

Fig. 2. All 49 minimal 3-XOR implementations as single-stage logic gates.

Some variants cannot use all reconfiguration types and modes. The static variant, for instance, is always balanced (as no implicit reconfiguration takes places) and uses inner mode (as Schottky barrier gates are slower than inner gates).

For the 49 implementations of 3-XOR, Figure 2 plots the maximum power dissipation against the maximum propagation delay for an output load $H = 1$. Both quantities, power and delay, are computed by model checking, i. e., no test runs or test cases of particular input combinations or timings are required. As PMC covers all possible traces and yields the maximum values and the traces that cause them, the method is both exhaustive and exact w. r. t. the model.

Apparently, two designs dominate the pack in Figure 2, both of which are new. The most power-efficient variant was found with *minimal-circuits* whereas the fastest design was first described in [20], but we optimized the distribution of the input signals B and C, cf. α in Figure 7 and [20]. This improves the delay by 5–10 %, depending on circumstance. Also note, that many power-efficient designs use a balanced reconfiguration type (light colored outline) but among the fastest implementations are circuits that use a single reconfiguration input (dark colored outlines).

B. Load-dependent Worst-Case Analysis

Circuits have to perform in various load situations. Hence, it is important to know, how strongly a high output load degrades circuit performance, given that reconfigurable variants use an input signal (at least partially) to drive the output.

Figure 3 shows the graphs for the loads $H \in \{1, 4, 7\}$ overlayed with each other when all three inputs may switch. Dominant designs are highlighted and connected by a Pareto curve for each load. The graph highlights also those instances of circuits that are not yet dominating but will be for other loads. At load $H = 7$ the static variant η dominates the delay, extending the Pareto front.

The shape of the three Pareto fronts and the relation of each circuit to its other incarnations reveals that the designs react very differently to an increased load. While the reconfigurable variants are already at their peak performance for $H = 1$ and

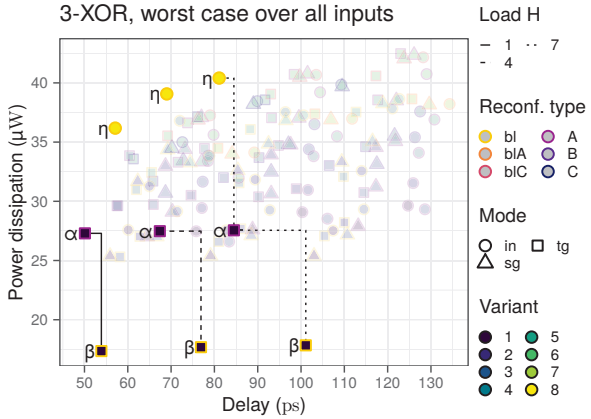

Fig. 3. Pareto-optimal 3-XOR implementations under loads $H = 1, 4, 7$, considering implicit reconfiguration. Greek letters correspond to circuit schematics shown in Figure 7.

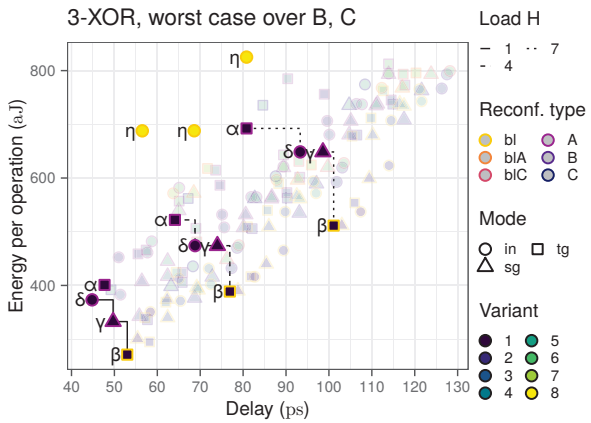

Fig. 4. Pareto-optimal reconfigurable 3-XOR implementations considering explicit reconfiguration between 2-XOR/ 2-XNOR.

do not significantly increase power dissipation for higher loads, the static implementation is able to deliver more energy faster. This is due to it having twice the number of paths from a power source to the output than the other two dominating variants, which means that in the worst case, it excites a strong cross current in the moment of switching but also provides a high-energy, high-speed configuration. In contrast, the delays of reconfigurable the variants start falling behind the static implementation with increasing distances.

The situation turns out to be more complex for the *explicit reconfiguration scenario* where only 2 out of 3 inputs are considered, while one input is left out of the measurement. For the reconfiguration types *bl*, *blA* and *A* this is input A.As much as five circuits become Pareto-optimal variants in Figure 4. The y-axis has changed and now displays the maximum energy per operation, because the maximum power dissipation turns out to often occur only during circuit reconfiguration of A, a case we have explicitly excluded in this scenario. All three reconfiguration modes, and both balanced and restricted

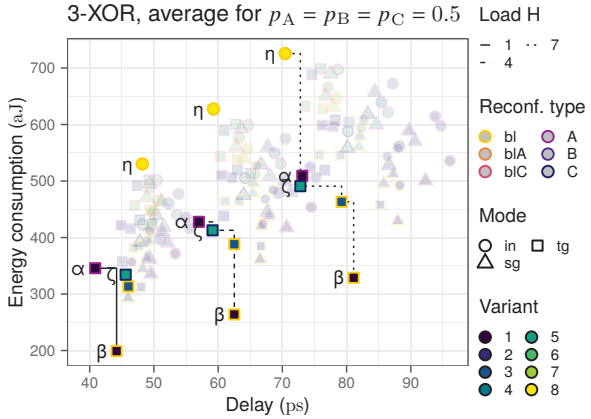

Fig. 5. Pareto-optimal 3-XOR variants under various loads considering average performance.

reconfiguration types, are found in dominating candidates. The energy-optimal design β consumes less than 50 % energy compared to the static variant η while still being faster. Also, the circuit η still does not dominate the delay at $H = 7$.

It is obvious from the Pareto fronts, that concentrating on a particular reconfiguration input, A in case of the three violet bordered circuits, enables strong performance gains for the other two inputs when the reconfiguration use-case is previously known to the circuit designer. Yet, as also comes without a surprise, the square-drawn circuits that use the transmission gate reconfiguration mode, are among the best-performing designs. Transmission gate circuits are know for their performance benefits but also for strong susceptibility to output load. Thus, it surprised us that most of the Pareto-optimal designs for $H = 7$ both, in Figure 3 and Figure 4, are still transmission gate mode designs.

C. Average Performance Analysis

In the majority of use cases, a technology node is not driven to its performance limits. This implies that the average performance of a circuit is actually a very relevant metric, even if a particular design can no longer achieve maximum conceivable performance. And this is where probabilistic model checking proves to be a valuable and superior tool to explore the available design space compared to simulation-based approaches.

PMC allows the designer to specify stochastic input modules that excite the circuit under test and to compute the average on the long run for metrics like energy consumption and propagation delay. The long-run average describes the average of a quantity under the assumption that the experiment entered its normal operation phase after its warm-up phase. Thus, its deviation from the true value is bounded by an ε environment, i. e., numerical precision, rather than the quality of a sample drawn from a set of experiments.

Figure 5 shows the same set of circuits as the previous figures under the assumption that each input has the probability $p_i = 0.5$ to contribute to the next input event that changes the

output. What distinguishes this experiment from the first worst-case setup in Figure 3, is, that the results contain also the many fast and cheap switching events that occur under realistic circumstances. So, it is no surprise that the average delay and energy consumption are about 20 % less compared to the worst-case scenario. The reconfigurable implementations stand out even more against the static implementation, comparing the Pareto fronts. The static implementation barely becomes faster than all reconfigurable variants for output loads around $H = 7$ or more. The three sets of experiments are still distinguishable in the figure and so is the spread of the last set for $H = 7$, which is a result of the variable load sensitivity of the various implementations.

It was shown in [9] that implementations of 3-MIN can be categorized into different groups, when comparing average energy consumption per operation, which correspond to the structure of the circuit implementations. Those designs with more electrical paths from the input to the output characteristically consumed more energy. But for the 3-XOR logic gate, this no longer seems to be the case.

For average analysis of the reconfiguration scenario, the switching probability of input A is decreased to 10^{-4}. Figure 6 depicts the delay and energy results but from the perspective of the changes caused by increasing the load from $H = 1$ by 3 units to $H = 4$ and then to $H = 7$. We can see that the jumps in delay and energy consumption still follow a distinct pattern. This pattern remains stable for both increments of the load H, first from 1 to 4 and second from to 4 to 7. The Pareto fronts now marks those circuits with the least sensitivity regarding output load, even though in absolute numbers the dominating designs may start off so bad, that they do not perform best in their immediate area. The static variant, without a doubt is the least delay-sensitive but one of the most energy sensitive designs, which is why it is found in the top-left corner.

D. Abnormal Situations

Figure 4 shows one peculiar detail for the static implementation η. For $H = 1$ and $H = 4$, it seems to consume the same amount of energy per operation in the worst case, although it has a considerably higher delay in the latter case. The reason is, that this high energy consumption is caused by an input switching event that does not change the output but creates a transient hazard. This event dominates all the others and explains the missing correlation to the worst-case delay. Would we have used only input signal combinations that excite an output switch, which would likely have happened in a simulation, we would have missed this case.

E. Circuits

The previous paragraphs showed that it is necessary to create a comprehensive picture about the implementation possibilities of a particular logic function, because the performance numbers vary drastically depending on the planned usage environment. Figure 7 picks out the six most relevant reconfigurable implementations which are Pareto-optimal under at least one set of conditions. It also lists key performance quantities, where

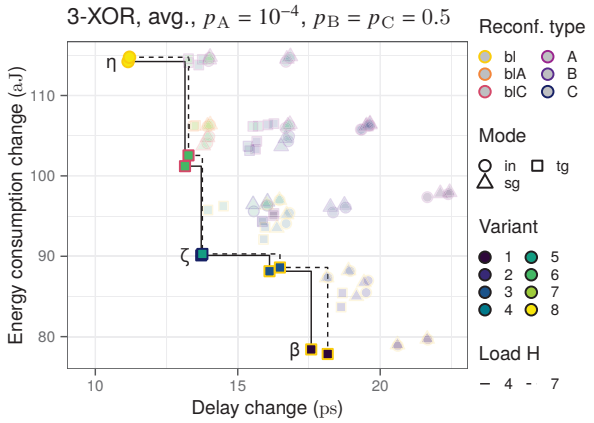

Fig. 6. Energy and delay sensitivity analysis considering explicit reconfiguration. Delay and energy show the *change* for a load increase from 1 to **4**, and from 4 to **7**.

bold quantities mark the leading implementation among the group of six circuits. We have to cut the interpretation of the results into multiple dimensions. The circuit structure, given by its variant in our taxonomy, is paramount to its key performance numbers, and the four best performing reconfigurable designs for low output loads (α–δ) are variant 1 designs.

Circuit α, which is fully identified as variant 1, type A (reconfiguration restricted to A), mode tg (transmission gate), is the fastest implementation for a low output load. While circuit β follows closely, it can outperform any other implementation in power dissipation and energy per operation due to its unique feature, that allows it to work with only two inverters to generate ¬B and ¬C where all other implementations have to use three.

In scenarios where circuit reconfiguration is not explicitly considered, implementations using transmission gate mode have an propagation delay advantage. The transmission gate mode avoids using the signal that is connected to the source terminal also at a transistor gate, which keeps the transitions steep and takes some load off the signal that must drive the output load. In reconfiguration scenarios, though, single mode and inner mode implementations, e. g., γ and δ, dominate the delay performance. The reason is, that inside the same variant, here variant 1, inner and single mode circuits have at each transistor only one (single) or no (inner) Schottky barrier gate that is driven by a non-reconfiguration input. In transmission gate mode, the same signal must be connected to both Schottky barrier gates to prevent unwanted ambipolar transistor behavior which in turn also increases the load on that signal.

In higher output load environments, the larger variants like ζ shown in Figure 7 have an advantage over the variant 1 designs, as they can use their partially static implementation to drive the load. Figure 6 shows that circuit ζ is less delay sensitive than the smaller implementations α–δ. It is also the absolutely fastest design in that scenario, between $H = 2$ and $H = 6$.

F. Computation Parameters and Performance

For the 3-XOR function, we analyzed 3 loads and generated 2 PRISM models for average quantities and 1 for worst-case

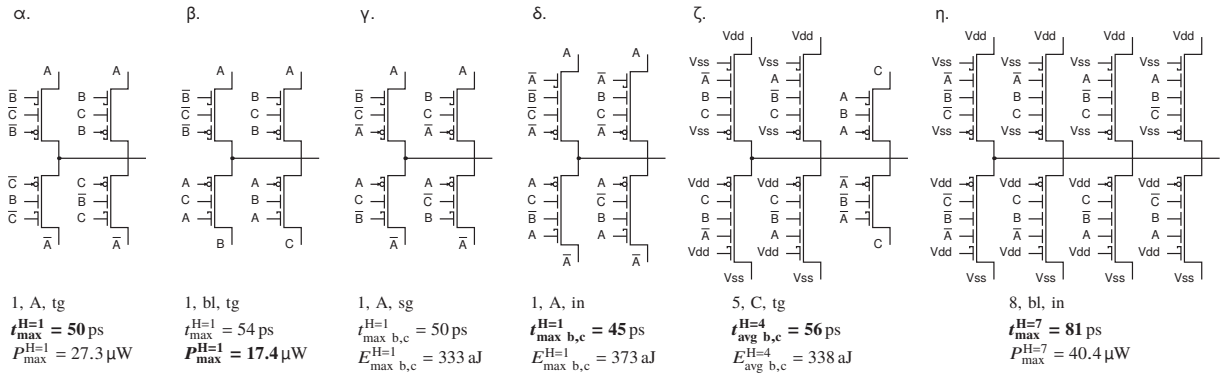

Fig. 7. Schematic drawings of Pareto-optimal implementations. Best performance among the shown circuits is given in bold.

quantities for each of the 49 circuits, resulting in 441 experiments.

The circuit model that is generated for each experiment uses a time resolution of 100 fs, which is why we used delays to a precision of 1 ps, and a voltage resolution of 10 µV. By scaling currents to micro Ampere, we achieve numerical stability. The power dissipation, being in the range of 10 µW, is close to the center of the numeric range of 64-bit floating point numbers.

Given these parameters, each model can be analyzed with at most 20 GiB of RAM. Model checking 441 experiments took a total of 1290 hours wall clock time, which amounts to 3 hours per experiment on average. This makes comprehensive design space exploration (DSE) of standard cells feasible.

IV. Conclusion

In this work, we showed that reconfigurable standard cells have a wide variety of performance characteristics and that their utility varies between applications. This demonstrates the necessity of a thorough DSE in multiple application scenarios quantifying average and worst-case characteristics. The effects of changes to the circuit structure also turn out to be irregular. That is why there is no simple algorithm with which the best designs can be created from scratch, and exhaustive analysis is needed to catch dominant effects that cannot be described with a predefined input stimulus. Formal methods provide the necessary features to enable this analysis in very early stages of the technology evaluation.

References

[1] D. Jariwala *et al.*, "Mixed-dimensional van der waals heterostructures," *Nature Materials*, vol. 16, pp. 170–181, Aug. 2016.

[2] G. Fiori *et al.*, "Simulation of graphene nanoribbon field-effect transistors," *IEEE Electron Device Letters*, vol. 28, no. 8, pp. 760–762, Aug. 2007.

[3] B. Radisavljevic *et al.*, "Single-layer MoS2 transistors," *Nature Nanotechnology*, vol. 6, no. 3, pp. 147–150, 2011.

[4] A. Heinzig *et al.*, "Reconfigurable silicon nanowire transistors," *Nano Letters*, vol. 12, no. 1, pp. 119–124, 2011.

[5] M. De Marchi *et al.*, "Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire fets," in *Electron Devices Meeting, IEDM '12, 2012 IEEE Int.*, 2012, pp. 8–4.

[6] J. Zhang *et al.*, "Dual-threshold-voltage configurable circuits with three-independent-gate silicon nanowire fets," in *Circuits and Systems, ISCAS '13, IEEE Int. Symp.*, 2013, p. 2111.

[7] A. Heinzig *et al.*, "Dually active silicon nanowire transistors and circuits with equal electron and hole transport," *Nano Letters*, vol. 13, no. 9, pp. 4176–4181, 2013.

[8] J. Trommer *et al.*, "Reconfigurable nanowire transistors with multiple independent gates for efficient and programmable combinational circuits," in *Proc. 2016 Design, Automation & Test in Europe Conf. & Exhib., DATE '16*, Mar. 2016, pp. 169–174.

[9] M. Raitza *et al.*, "Quantitative characterization of reconfigurable transistor logic gates," *IEEE Access*, vol. 8, pp. 112 598–112 614, 2020.

[10] V. Vaidya *et al.*, "SPICE optimization of organic FET models using charge transport elements," *IEEE Trans. Electron Devices*, vol. 56, no. 1, pp. 38–42, 2009.

[11] J. Katoen, "The probabilistic model checking landscape," in *Proc. 31st Annu. ACM/IEEE Symp. on Logic in Computer Science, LICS '16*, 2016, pp. 31–45.

[12] C. Baier *et al.*, *Principles of model checking*. MIT Press, 2008.

[13] J. Trommer *et al.*, "Enabling energy efficiency and polarity control in germanium nanowire transistors by individually gated nanojunctions," *ACS Nano*, vol. 11, no. 2, pp. 1704–1711, 2017.

[14] M. Simon *et al.*, "A wired-and transistor: Polarity controllable fet with multiple inputs," in *2018 76th Device Research Conf. (DRC'18)*, Jun. 2018, pp. 1–2.

[15] D. Vana *et al.*, "C^{2}TIG: Dynamic C^{2}MOS design based on three-independent-gate field-effect transistors," *IEEE Transactions on Nanotechnology*, vol. 19, pp. 123–136, 2020.

[16] J. Trommer *et al.*, "Inherent charge-sharing-free dynamic logic gates employing transistors with multiple independent inputs," *IEEE Journal of the Electron Devices Society*, 2020.

[17] P.-E. Gaillardon *et al.*, "Three-independent-gate transistors: Opportunities in digital, analog and RF applications," in *17th Latin-American Test Symposium, 2016, LATS '16*, IEEE, 2016, pp. 195–200.

[18] S. Märcker, "Model checking techniques for design and analysis of future hardware and software systems," Ph.D. dissertation, Technische Universität Dresden, 2020.

[19] M. Kwiatkowska *et al.*, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd Int. Conf. on Computer Aided Verification, CAV '11*, vol. 6806, 2011, pp. 585–591.

[20] J. Zhang *et al.*, "Configurable circuits featuring dual-threshold-voltage design with three-independent-gate silicon nanowire fets," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 10, pp. 2851–2861, 2014.