Can Emerging Reconfigurable Nanotechnologies meet the Requirements of Logic Cell Architectures for FPGAs?

SHUBHAM RAI, Chair for Processor Design, TU Dresden, Germany
PALLAB NATH, Indian Institute of Technology Indore, India
ANSH RUPANI, Chair for Processor Design, TU Dresden, Germany
SANTOSH KUMAR VISHVAKARMA, Indian Institute of Technology Indore, India
AKASH KUMAR, Chair for Processor Design, TU Dresden, Germany

Field Programmable gate arrays (FPGA) have been one of the major drivers for configurable circuit implementations and are being employed for applications like neural networks and cloud computing. They boast of higher performance as compared to the conventional software systems. However, this versatility comes at a price of exponentially increasing complexity with an increase in the number of LUT inputs. A crucial part that defines the power consumption of field programmable gate arrays (FPGAs) is the underlying architecture of their logic cell. With the CMOS feature size shrinking, we are hitting a power wall with Dennard scaling due to the increase in static power dissipation. Although measures like power-gating are available, it is intriguing to have a fresh perspective towards newer reconfigurable nanotechnologies for electronic circuits. Emerging nanotechnologies based on materials like silicon, germanium and carbon exhibit runtime-reconfigurability at the device level and have low-leakage operation. In the present work, we consider the possibility of using these novel and feature-rich technologies for implementing logic cells in FPGAs. We investigate the ubiquitous LUT-based architecture and the recently proposed And-Inverter cones for logic cells. We carry out circuit-level simulations with 10nm reconfigurable nanotechnology model for individual components of FPGA logic cell architecture. A preliminary evaluation using VTR 8.0 over combinational (EPFL) and sequential (VTR) benchmarks show 11.51 and 13 % of improvements in power dissipation of logic components as well as 61.37% and 30.41% logic delay calculated as compared to the existing CMOS-based FPGA architecture for both the benchmark suites. For And-Inverter cones, preliminary investigations and evaluations show 80 % improvement in the average delay and 69 % reduction in delay-discrepancy as compared to the state-of-the-art.

ACM Reference Format:
Shubham Rai, Pallab Nath, Ansh Rupani, Santosh Kumar Vishvakarma, and Akash Kumar. 2019. Can Emerging Reconfigurable Nanotechnologies meet the Requirements of Logic Cell Architectures for FPGAs?. 1, 1 (May 2019), 20 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Field Programmable Gate Arrays (FPGAs) provide the freedom to adapt the hardware as per application requirement and is expected to reach into a $9.8 billion industry by 2022 [25]. They boast of features like compile time and runtime reconfigurability, short time-to-market, easy prototyping etc. which make them a reliable and viable option in digital systems applications [14].

Authors' addresses: Shubham Rai, Chair for Processor Design, TU Dresden, Germany; Pallab Nath, Indian Institute of Technology Indore, India; Ansh Rupani, Chair for Processor Design, TU Dresden, Germany; Santosh Kumar Vishvakarma, Indian Institute of Technology Indore, India; Akash Kumar, Chair for Processor Design, TU Dresden, Germany.

XXXX-XXXX/2019/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Most of the commercial FPGAs have look-up-table (LUT) based architectures [10]. These LUTs are the basic logic cells and are primarily based on static RAMs. A k-input LUT can implement any function possible with up to k inputs. The above functional completeness is due to the freedom provided by these LUTs but comes at the cost of compromised delay, area and power consumption as compared to application specific integrated circuits (ASICs) [26]. These problems are getting further aggravated because of CMOS scaling. Though CMOS scaling continues, the area, delay and power benefits over cost are getting further skewed and this calls for rethinking the FPGAs' logic cell architecture. The power consumption takes the critical factor, as evident by the fact that with operating voltage of 0.6V, around 50% of total energy consumption is wasted as leakage dissipation in modern FPGAs [39]. Many researchers have proposed techniques like guarded evaluation, clock gating, power gating, dual supply voltages, and power-aware CAD optimization [30, 37, 54, 55] but power consumption still remains one of the concerning issues.

While CMOS as a technology continues to be at the very core of logic cell architecture for FPGAs, several recently researched emerging nanotechnologies having reconfigurablity at the transistor level are exciting alternatives for implementing FPGA-like systems. Such devices are based on technologies and materials like silicon [22, 33] and germanium [51] nanowires, carbon nanotubes [31] and 2D materials like WSe_2 [44]. Reconfigurable nanotechnologies exhibit unique properties like i) runtime-reconfigurability between p- and n-type behavior with fully-symmetrical I-V characteristics in both configurations ii) low leakage power dissipation due to Schottky junctions [52] and iii) having multiple independent gates on the same channel creating a wired-AND effect [52] which make them suitable to implement FPGA architectures. Additionally, reconfigurable field effect transistors (RFETs) based on silicon and germanium nanowire follow the same manufacturing process as CMOS [46], further pushing such technologies for easier and cheaper industrial adoption.

Recent works like [19, 41, 43] have shown encouraging results for circuit optimizations using RFETs. Apart from the above works, several attempts have been made to explore programmable hardware implementations based on emerging nanotechnologies for FPGAs [12, 20]. The authors in [57] evaluated spintronics based FPGA architectures. These works indicate the need of a fresh perspective for novel FPGA logic cells and architectures based on emerging nanotechnologies.

Contributions: Overall, our contributions can be summarized as follows:

- We carry out detailed circuit simulation for individual logic components of an FPGA like MUX, crossbars, LUTs with RFET models and feed into VTR tool [45] to realize an FPGA architecture completely based on RFET technology. For RFET technology, we use silicon nanowire based TIGFET model available at [2]. This works hence promises of being an early level evaluation of FPGA architectures based on reconfigurable nanotechnology.
- With EPFL and VTR benchmark suites, we evaluate the proposed FPGA architecture and found average delay gains of 61.37% and 30.41% and power gains of 11.5% and 13% respectively as compared to CMOS based FPGA at the same technology node.
- An efficient implementation of *NAND-NOR* cell [24] based on reconfigurable technology is demonstrated followed by comparative analysis with respect to the existing *NAND-NOR* logic cell. Our design is better in terms of number of transistors and shows 81% improvement in the average delay for the RFET based NAND-NOR cone. Moreover, with the use of reconfigurable nanotechnology, we are able to solve the delay-discrepancy by 69% as compared to the state-of-the-art.
- We discuss the challenges and limitations related to development based on reconfigurable nanotechnologies for efficient programmable hardware.

Our focus in this paper is mainly for logic components and not on routing aspect of FPGA. We use CMOS based PTM 10nm model available at [1] for our baseline technology file as the closest approximation because TIGFETs model based on Silicon nanowire is still a part of an ongoing research. While our discussions are based on this

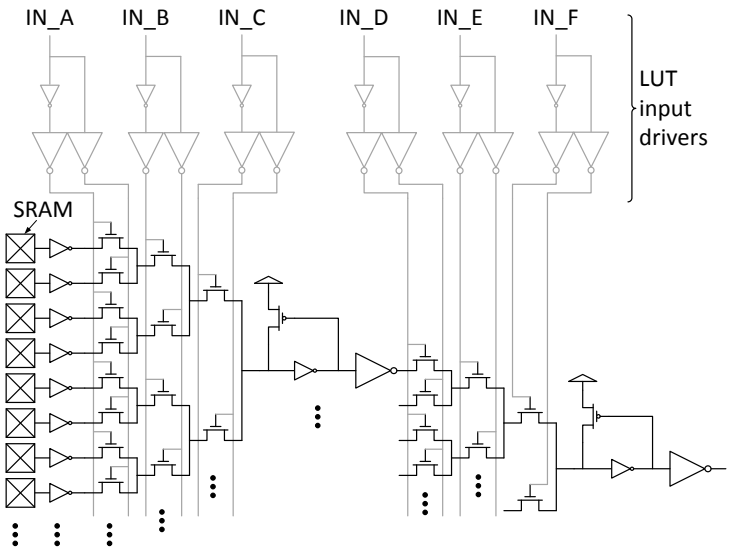


Fig. 1

Fig. 2. Standard LUT Design [3]

specific technology the idea and concepts described in the manuscript is applicable to any other technology agnostic reconfigurable FETs.

2 BACKGROUND AND RELATED WORKS

In this section, we discuss the logic cell architecture in FPGAs starting with the most commonly used *Look-up-table (LUT)* architecture. Various logic cells have been proposed in the literature to improve the basic architecture of an FPGA. This helps us to build upon the parameters which are crucial for an FPGA's architecture.

2.1 Standard LUT Design

Most of the commercial FPGAs consist of look-up tables (LUTs) as their primitive logic cell which are capable of performing any function of up to a given number of inputs. A generic LUT logic cell as shown in Figure 1 consists of rows of SRAM cells connected to a tree of MUXes [3]. The SRAMs hold the configuration bits while the tree of MUXes selects the bits to drive the output. The FPGAs' *Basic Logic Element (BLE)* typically comprises of a k-input LUT, a flip-flop and a MUX for routing outputs and implementing carry-chains.

Contemporary FPGA's BLEs are generally composed of coarse-grained LUTs with more than 4 inputs (typically 5 or 6) as it has been shown that they deliver the best performance as compared to fine-grained LUTs (LUTs with less than 4 inputs) [48].

Several other architectures like the fracturable-LUT [29] and the S44 [18] architecture have been proposed in the literature. Similarly, in order to look for a power efficient logic cell design with power-gating, while not deteriorating the performance of the FPGA, the authors in [17] proposed a logic cell as a combination of *Reconfigurable Hard Logics (RHLs)* and a 3-LUT. Their design showed improvement in terms of total static power and Power-Delay-Product (PDP) as compared to the traditional LUT based architectures. All these works were based on CMOS technology albeit at different technology nodes. The works [17, 32**?**] demonstrated that by

compromising on the versatility of the LUTs, better area, power and delay parameters can be achieved. However, for other parameters they could not show better optimality as compared to the LUT based architecture. In the present work, we will be focusing on the standard LUT architecture.

2.2 And-Inverter Cones

LUTs suffer when it comes to scaling, as their complexity increases exponentially with the increase in the number of inputs. Due to this reason, LUTs with more than 6 inputs have rarely been used. In an attempt to handle more inputs and increase the logical density of logic cells, *And-Inverter Cones (AICs)*, shown in Figure 3a were proposed [38]. These are an alternative for LUTs with a better compromise between hardware complexity, flexibility, delay and input and output counts. These are inspired by modern logic synthesis approaches [11] which employ and-inverter graphs (AIGs) for representing logic networks. AIC is a binary tree which consists of AND nodes with programmable conditional inversion and offers to tap intermediate results. AICs have a lot to offer as compared to the traditional LUT based FPGAs. The following points summarize the major benefits of using AICs over LUTs:

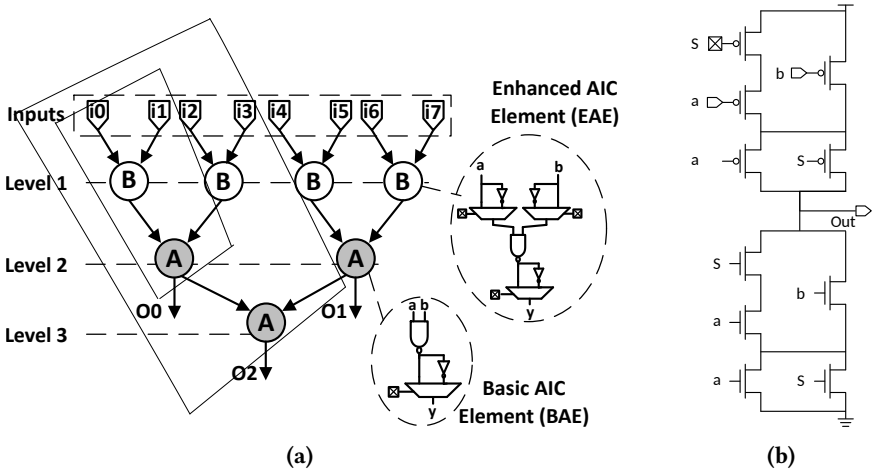


Fig. 3. (a) AND-Inverter Cones [38] (b)10 transistor design for NAND-NOR cell shown in [24]

- For a given complexity, AICs can implement a function with more number of inputs compared to an LUT.
- Since it is inspired by AIGs, area and delay increase linearly and logarithmically respectively with the number of inputs which is in contrast to the respective exponential and linear increase in case of LUTs [24].
- Intermediate results can be tapped out of AICs thereby reducing logic duplication.

In spite of the above benefits, they have issues caused due to programmable conditional inversion which leads to *delay discrepancy* and inefficient cluster design as discussed in [24]. Every level of an AIC consists of 2-input ANDs which can pass in normal or complemented form to the next stage. Since to implement a complemented form, an inverter as needed which adds extra delay in the combinational path. Hence, the propagation delay of the cell depends upon its configuration. In combinational circuits, different arrival times of signals that might otherwise transmit simultaneously, lead to signal competition and might cause glitches. Glitches often lead to instability and errors in the circuit. *NAND-NOR Cones* proposed in [24] tried to solve these issues in two ways: Proposing a novel transistor level NAND-NOR logic cell and employing architectural changes like using *Delay-balanced Dual-phased Multiplexer (DDM)* crossbars to tackle the delay discrepancy problem [24]. The

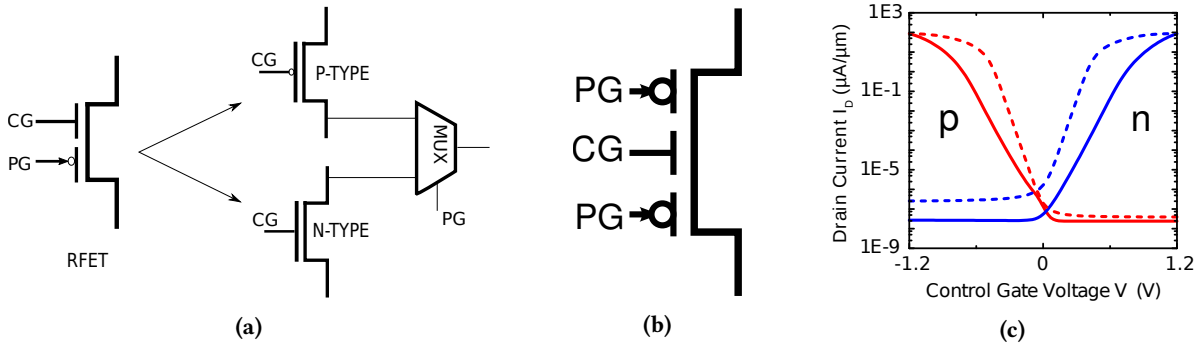


Fig. 4. Various Reconfigurable FETs (a) RFET based logic Gate and comparison with CMOS (b) Representative figure of RFET in [34] with program gates over both source and drain (c) The graph shows symmetry in p and n-type behavior for *SiNW* RFET [43]

proposed *NAND-NOR* cones were proven to have better overall delay values compared to AICs while also keeping the delay-discrepancy under check. In our work, we start with the NAND-NOR cones and demonstrate that the minority gate based on reconfigurable nanotechnology can be suitably replace it with lesser delay-discrepancy.

3 RUNTIME-RECONFIGURABLE EMERGING NANOTECHNOLOGY

In this section, we provide an overview of reconfigurable emerging nanotechnologies. We demonstrate how the unique properties exhibited by these technologies are perfectly suited to the demands of programmable hardware.

3.1 Nanowire based Reconfigurable FETs

Reconfigurable property is exhibited by transistors made from materials like silicon [22, 33], germanium [53] nanowires and graphene nanoribbons [21]. These transistors are ambipolar, i.e. they can show p-type or n-type behavior within the same device.

Our present work is inspired by SiNW based reconfigurable technology. Silicon Nanowire RFETs can have multiple independent gate terminals on a single channel in contrast to the CMOS technology as shown in Figure 4a [22] and Figure 4b [33]. The **control gate** (CG) receives the normal input to the transistor which controls the creation of a carrier channel. The **program gate** (PG) determines the type of the carrier thereby enabling encapsulated p and n-type functionality as seen in Figure 4c. Here the above ambipolarity is enabled by metal-NiSi2/intrinsicSi/metal-NiSi2 nanowire structures forming Schottky contacts. Results have shown that SiNW transistors can significantly reduce the transistor count in circuits when compared to CMOS [40, 43, 50]. An introductory review can be found in [36] with details on device physics for these nanowires in [56].

SiNW RFETs as a technology offers the following benefits over CMOS:

- It offers runtime reconfigurability which enables to get more functions per computational unit [43, 52]. The I-V curve shown in Figure 4c is symmetric which implies that both n and p-type configurations deliver similar ON and OFF currents [23].
- The technology also offers to add additional independent gates in the channel. Thus, three-independent gates FETs (*TIGFETs*) [59] to multi-independent gates FETs (*MIGFETs*) [50] have been shown experimentally. Each transistor with multiple independent gates can be visualized as an equivalent circuit consisting of several transistors in series. Unlike CMOS, where all the series resistances of the individual transistors add up, RFETs can have an equivalent resistance of R_{on}, where each independent gate contributes a resistance

of R_{on}/m, with m being the number of independent gates. This leads to a notable decrease in the number of series path resistances and the number of dynamic capacitances in a circuit.
- Being a Schottky device, the leakage power dissipation is extremely low as compared to the conventional CMOS technology.
- As compared to other emerging technologies, nanowire based RFETs follow a mature top-down manufacturing process as CMOS [46] which further makes it easier for industry adoption.
- Among the reconfigurable nanotechnologies, silicon nanowires are one of the earliest and most highly researched technologies. They are shown to provide better area [43], power and delay numbers [6].

The above properties make RFETs an exciting choice for programmable hardware requirements.

3.2 CNTFETs

Carbon Nanotube based Field Effect Transistors (CNTFETs) based on Schottky contacts are ambipolar FETs, capable of switching their nature from p-type to n-type and vice versa at runtime. This nature is similar to technologies like SiNW RFETs and Graphene SymFETs. CNTFETs have Schottky barriers at the metal contacts.

- The polarity of the transistor is set by changing the fringing gate field at CNT-metal contact, which in turn modulates the thickness of the Schottky barriers. This feature of changing the polarity by modifying the field is normally modeled as another gate input, referred to as the polarity gate, as suggested by other works like [8].
- They offer completely-symmetric p and n-type characteristics, similar to the behavior seen in SiNW RFETs.
- Due to the high mobility in carbon, the CNTFET devices show very fast performance with high I_{ON} currents.
- Though CNTFETs offer great freedom in controlling the polarity of the device at runtime, there are some disadvantages associated with CNTFETs. These devices are often associated with high manufacturing costs, variability and reliability issues. To solve the variability issues, authors in [28] have proposed a variation-aware compact model of CNTFET.

However, we assume that such emerging technologies will keep on improving in the future In the next subsections, we list down the works which have used emerging nanotechnologies for computing platforms.

One of the earliest work using silicon nanowires or carbon nanotubes for reconfigurable platforms was done by DeHon [16]. Using bottom-up approaches to support finer feature-sizes he used stochastic placement (which does not promise perfect deterministic alignment), to propose programmable-logic arrays [16] and programmable inerconnects [15]. Such programmable-logic arrays using silicon nanowires were termed as *NanoPLA* and can be configured to any logical functions to deliver high-density computing platforms. While he proposed a PLA kind of architecture using dopant based nanowires technology, our work focuses on dopant-free nanowire technology as proposed in [22, 33, 51]. Further, the nanowire technology used in the present work employs top-down lithography techniques which promises more deterministic alignment [46].

Gaillardon et.al. used a novel fine-grained architecture which used domino logic for logic cell architecture and was capable of implementing a major portion of all 2-input functions [20]. The logic cell is shown in Figure 5a. Then using a $k \times k$ matrix based BLE they compared their implementation with k-input LUTs and showed improvements. However, there were certain limitations related to the proposed design: The major concern was the use of domino logic which are highly susceptible to noise. The proposed logic cell relies on a 4-phase pseudo clocking signal which consists of two pre-charge (*pc*) and two evaluation (*ev*) signals. Each logic cell is connected to these four signals, which are always switching in synchronization leading to higher power dissipation. Intermediate D-flipflops are needed to support pre-charge and evaluation of each domino stage [20]. Another important point is that the delay of a domino logic cell changes almost more than twice as much with process variations compared to static logic [4].

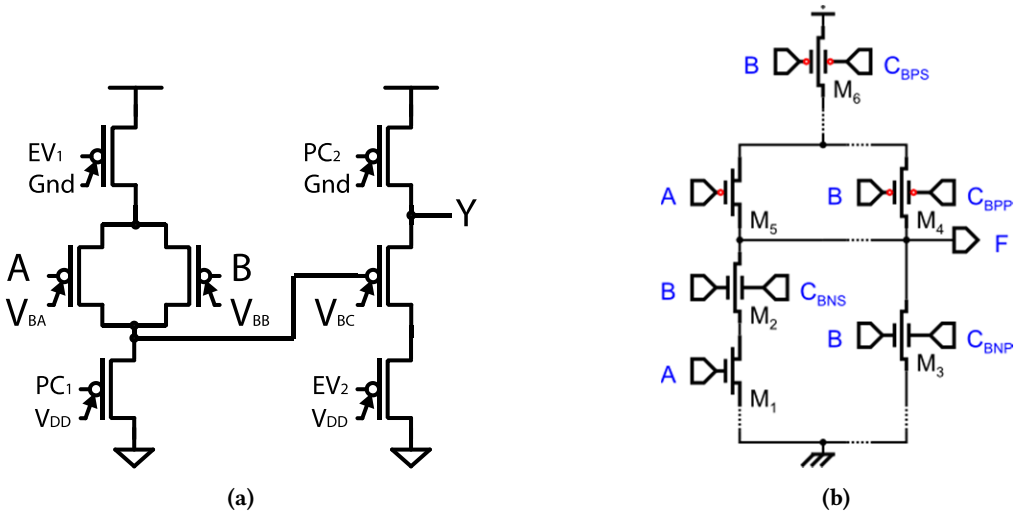


Fig. 5. Logic Cells Proposed in Literature Based on Reconfigurable Nanotechnologies (a) Novel FPGA Design Using Ultra-Fine Grained Logic Cell [20] (b) Logic Cell Design Using CNTFETs [12]

Another approach towards reconfigurable platforms was proposed by Cheng et al. in [12]. They used *Ambipolar Independent Double-Gate Field effect transistors* (Am-IDGFET) based on Carbon Nanotube technology to propose a fine-grained architecture. Am-IDGFET exhibits three states: N-type, P-type and OFF through the application of different back-gate voltages. Using Am-IDGFET, three nano-grained reconfigurable cells were proposed. One of the elementary cells proposed is shown in Figure 5b. These nano-grained cells were not functionally complete as compared to a *k*-input LUT. In order to solve the above problem, they proposed optimum interconnect topologies to obtain a cluster thereby achieving functional completeness.

While all the above works proposed novel designs for FPGA architecture based on reconfigurable FETs, we use the existing LUT-based FPGA architectures based on RFETs.

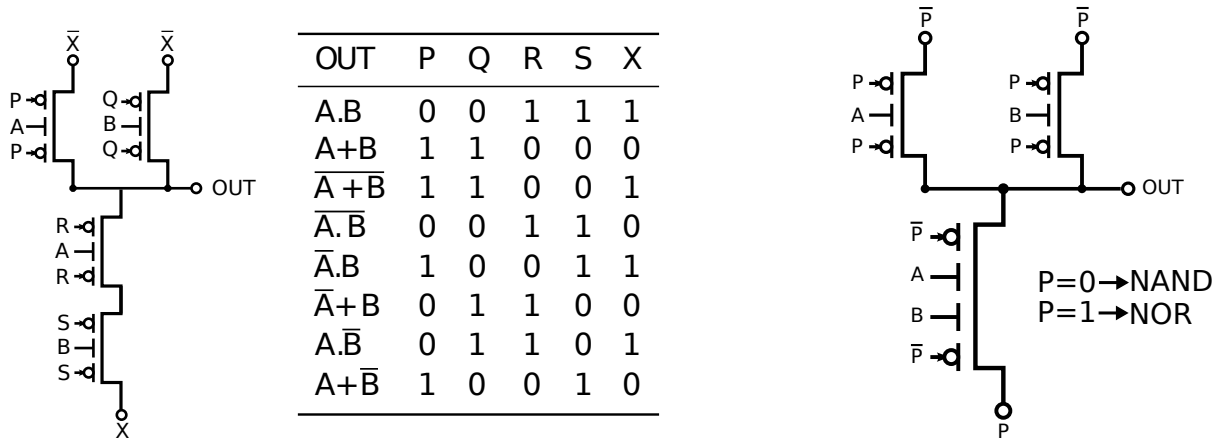
4 RFET BASED LOGIC CELL FOR AIC

Building on the capability of RFETs, we propose a logic cell design *4T* design which employs the runtime reconfigurability to switch between the pull-up and pull-down network by application of voltage potential. Since there is an electrical symmetry in the p-type and n-type functionality as shown in Figure 4c, the RFETs in the proposed logic cells can switch in runtime between the pull-up and the pull-down network. This yields more functions from the logic cell while encouraging quick dynamic reconfiguration.

4.1 4T Design

The design employs 4 RFETs in the main logic cell design as shown in Figure 6a. Each of the 4 RFETs is dynamically configured between n-type and p-type separately to obtain 8 functions from it. This feature enables to simulate the inverted behavior of inputs as shown in Figure 7. By changing the program gate input at runtime and consequently changing the n-channel device to a p-channel device, it is possible to simulate the inverted behavior of an input. For instance, for a gate input A, A=0 turns a PFET *on* while $\overline{A}$=1 turns an NFET *on*. [1]. With the help

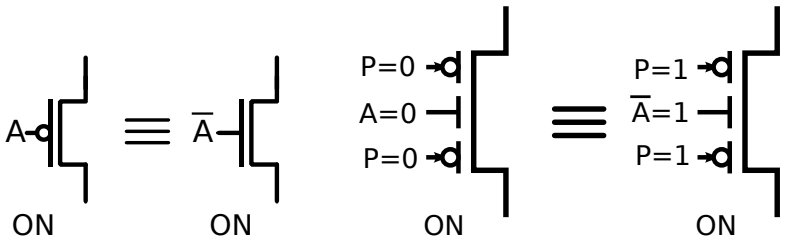
[1]Such modifications would need *Gate boosting and level shifter circuits* to truly realize this behavior. However, such evaluation is beyond the scope of this work



OUT	P	Q	R	S	X
A.B	0	0	1	1	1
A+B	1	1	0	0	0
$\overline{A+B}$	1	1	0	0	1
$\overline{A.B}$	0	0	1	1	0
$\overline{A}$.B	1	0	0	1	1
$\overline{A}$+B	0	1	1	0	0
A.$\overline{B}$	0	1	1	0	1
A+$\overline{B}$	1	0	0	1	0

(a) **(b)**

Fig. 6. Logic Cells Proposed in Literature Based on Reconfigurable Nanotechnologies (a) 4T design and its various reconfiguration (b) NAND-NOR Logic Cell with a multi-independent gate FET (MIGFET) [50]. This is also a MINORITY logic cell



(a) For CMOS **(b)** For RFETs

Fig. 7. Simulating inverted input with opposite polarity FET

of the modifications suggested above, we are able to make this cell functionally dense and implement the same number of functions as implemented by the cells in Figure 5 [12, 20].

The above technique is slightly unconventional with respect to CMOS logic because such runtime reconfiguration can lead to a p-type transistors in pull-down and n-type transistors in pull-up. Putting p-type transistors in pull-down and n-type transistors in pull-up in a logic cell results in an insufficient voltage swing at the output. This results in large short circuit currents and slow switching speed because the waveform slew rate is slow near V_{dd} and *GND*. These issues are common in contemporary FPGAs, where the multiplexers used for routing also consists of transistors in pass-transistor logic. To counter such issues, *Gate-Boosting* is an efficient technique to get a sufficient voltage swing at the output as discussed in the literature. This method of gate boosting with PMOS level-restorer has been extensively used for pass transistors in LUT-based FPGAs [13]. Here, a voltage larger than the supply voltage V_{dd} for logic high, and a voltage lower than ground *GND* for logic low is applied onto the transistor's gate terminal. This allows both n and p- type transistors to pass both logic high and low effectively, irrespective of them being in pull-up or pull-down network, consequently passing a full rail-to-rail

voltage at the output. While such techniques are shown to accelerate device aging, thicker gate oxides can be used to allow gate voltage boosting without compromising reliability.

In the present work, we limit the functions to be implemented using this design to compute only NAND and NOR logic. NAND-NOR cells can be efficiently implemented using RFETs as shown in Figure 6a. Further, the two independent RFETs in the bottom leg can be replaced with a single multi-independent gate FET (MIGFET) [50] as shown in Figure 6b to have a more efficient implementation of Minority logic as the minority logic function encapsulates both the NAND and NOR functionality.

5 CIRCUIT SIMULATION OF INDIVIDUAL COMPONENTS

We carry our circuit simulation for various major components of FPGA logic architecture. We used a three-independent gate (TIGFET) based model for Silicon nanowires which is at 10nm technology node [2]. We compare our results with a corresponding high performance (HP) and low power (LP) 10nm PTM based on CMOS technology [1]. For each component, we use Cadence Virtuoso for our simulations.

Since the TIGFET model is still undergoing research, some important points have to be kept in mind regarding the TIGFET 10nm model. Theoretically speaking, though RFETs show fully-symmetrical p and n-type behavior and has been shown experimentally [59] and theoretically [52], in the present model, there is a considerable skew between the current drive in the n-type and p-type configuration. The n-type configuration is stronger than the p-type one, so much so, that it causes a width-p/width-n ratio of 6/1. The above reason leads to sizing issues with reconfiguration. For a given configuration, if we size the pull-up and pull-down network accordingly, the sizing would lose meaning when we reconfigure the logic cell as the pull-up network becomes pull-down and vice-versa. It is particularly detrimental to applications where we want minimal delay discrepancy between different configurations of the cell. Hence, all the simulations are done with identically sized transistors (wn=wp=1 and wn=wp=4 for now). This leads to better results as apparent from the calculations below. They have the same Vdd= 0.8V and the operating temperature is kept at 25 degree Celsius. For the corresponding PTM 10nm technology model, the ratio wp/wn is 1.2/1. This is representative of the fact that Schottky devices (RFETs) are comparatively slower as compared to the ohmic contact devices (CMOS).

Major components of an FPGA's CLB (configuration logic block) have been simulated using both the PTM and the TIGFET based model. To the best of our knowledge, such an extensive evaluation has been carried for the first time for an FPGA architecture. The results are shown in Table 2, 3, 4. It can be seen that the values of TIGFET lie somewhere in the middle of the HP and the LP PTM models. Another flexibility in terms of performance is provided by changing the values of wp and wn. This directly affects the delay and power numbers of individual components. In order to have the most optimum value of power and delay for our benchmark evaluation as mentioned in Section 6.1, we have used wp and wn equal to 4. Further, in most of the circuit design, the reduced transistor count in case of TIGFET, leads to delay and power gains over PTM models as has been shown before [5, 42]. The same trend can be seen in coming sections.

Technology	Static power per transistor (nW)
TIGFET wp =1, wn=1	0.018375
TIGFET wp =1, wn=1	0.0735
PTM HP	188.625
PTM LP	0.2

Table 1. Comparison of Static Power per transistor across technologies

Models	Design	Total Power (nW)	Delay (ps)	Static power (nW)	Dynamic Power (nW)	Energy per toggle (J)
TIGFET	4-LUT	31.4	72	0.154	31.246	3.1246E-16
(wn=1,wp=1)	6-LUT	40	155.6	0.258	39.742	3.9742E-16
TIGFET	4-LUT	121.9	19.1	0.615	121.285	1.21285E-15
(wn=4,wp=4)	6-LUT	156	38.9	0.934	155.066	1.55066E-15
PTM_HP	4-LUT	6384	18.8	3384	3000	3E-14
(wn=10,wp=10)	6-LUT	7292	22.8	4633	2659	2.659E-14
PTM_LP	4-LUT	106	66.2	3.8	102.2	1.022E-15
(wn=10,wp=10)	6-LUT	193	89	5.2	187.8	1.878E-15

Table 2. Simulation Results for 4-LUT and 6-LUT

5.1 Static Power

Silicon Nanowire based RFETs are Schottky based FETs which inherently leads to less leakage dissipation. This in turns leads to savings in static power dissipation. Static power dissipation for a single transistor has been shown in Table 1. It can be seen that TIGFET leads to 2 to 3 orders of magnitude lesser static power as compared to the PTM model. This ultimately leads to overall lesser leakage dissipation which is a major concern with lower technology nodes due to the higher number of off-transistors.

5.2 MUX-tree for 4 and 6-LUT

In order to compare the CMOS and the corresponding TIGFET model, we used 4-LUT and 6-LUT for our analysis. This was important because in contemporary architectures, more than 4-input LUT is the standard input size for LUTs. We have only used the MUX tree part of LUT for our analysis. The SRAM part is used during bitstream configuration and once the values are set it does not contribute to overall delay as it can be approximated with the assumption that there is no arrival time delay for the inputs. Once the SRAMs are set, a particular value is selected using the MUX-tree as an output of the LUT depending upon a particular set of input configuration. It can be seen from Table 2, while TIGFET based LUTs gain in the overall power, the delay is comparable with the PTM LP values. The PTM HP shows the best performance but shows 2 to 3 orders more in terms of total power consumption. Further, TIGFET (wp=4,wn=4) based model shows a comparable dynamic power consumption LUT designs as compared to the PTM LP model.

5.3 MUXes and Crossbars for Routing

To enable FPGA's reconfigurable behavior, MUXes give the freedom to generate any logic function possible. They form the backbone for FPGA routing structure both at the intra- as well as inter-CLB level. One can see that with the latest model, one get favorable values as compared to the CMOS based PTM models. Moreover, features like wired-AND functionality as proposed in [47], one can further reduce the area, delay and power for MUXes. Such functionality can further be useful to absorb multiple stages of MUX-chain with a single stage as has been shown in [43]. Such multi-gate functionality in larger circuits has been shown [42, 59]. However, in the present model, such designs cannot be employed due to unavailability of such than more than one control gate devices. An important point to note here is that such multi-gate designs can further optimize the basic LUT structure as well. From Table 3, we can see similar gains in terms of delay and power as shown in LUTs. Similarly, in terms of

Models	Design	Delay (ps)	Total Power (nW)	Static power (nW)	Dynamic Power (nW)	Energy per toggle (J)
TIGFET (wn=1,wp=1)	2x1MUX	24.1	10	0.147	9.853	9.853E-17
	4x1MUX	37.4	11.2	0.219	10.981	1.0981E-16
TIGFET (wn=4,wp=4)	2x1MUX	13.5	37.8	0.588	37.212	3.7212E-16
	4x1MUX	16.9	41.5	0.936	40.564	4.0564E-16
	i/p-crossbar	22.6	251	1.498	249.502	2.49502E-15
	Feedback-crossbar	18.53	206	1.633	204.367	2.04367E-15
PTM HP (wn=10, wp =10)	2x1MUX	12.5	3613	1509	2104	2.104E-14
	4x1MUX	15.2	4212	1930	2282	2.282E-14
	i/p-crossbar	21.5	8160	4553	3607	3.607E-14
	Feedback-crossbar	18.2	6996	5139	1857	1.857E-14
PTM LP (wn=10, wp =10)	2x1MUX	38.65	29.2	1.6	27.6	2.76E-16
	4x1MUX	49.5	49.7	2.2	47.5	4.75E-16
	i/p-crossbar	63.5	84.7	6.7	78	7.8E-16
	Feedback-crossbar	61.3	79.5	6.5	73	7.3E-16

Table 3. Simulation Results for MUXes and Crossbars

Design	T_setup	clk_to_Q	Total Power (nW)	Dynamic Power (nW)	Static Power (nW)
TSPC-DFF (TIGFET)	10 ps	7.88 ps	9.43	9.047	0.383
TSPC-DFF (PTM HP)	14 ps	15.4 ps	2274	707	1567
TSPC-DFF (PTM LP)	70 ps	85.2 ps	65	22.5	42.5

Table 4. Values of TSPC D-Flip-Flop

dynamic power, TIGFET (wp=4,wn=4) shows higher numbers for smaller circuits but gains for larger crossbars as compared to PTM LP.

5.4 True-Single-Phase-Clock D Flip-Flop

D flip flops are integral to FPGA CLB architecture. In order to carry out circuit simulation for flip-flops, we incorporated the TSPC D Flip Flop as proposed in [60] and evaluated it with the new TIGFET model. It is clear from Table 4, the values of the delay for TIGFET based DFF is better as compared to the CMOS based DFF. In the case of TSPC DFF, there was no need to do simulation for wp=4,wn=4 because that would lead to the lowest delay numbers of all the four. As of now, with wp=1,wn=1 TIGFET model, the *setup* and *clk_to_q* is still the lowest of the three. Hence, we take these numbers for our calculations. Additionally, this DFF offers dynamic reconfiguration to high and low performance and hence can be tuned to have additional savings in energy as mentioned in [58]. In our experiments, we use the high performance mode for simplicity.

6 EXPERIMENTS AND RESULTS

In this section, we first provide a a detailed comparison between the NAND-NOR cones and the 4T design based on the circuit simulation. Benchmark level evaluation for our 4M-scaled design with respect to CMOS based 4-LUT design is carried out at the end of this section.

6.1 4T vs NAND-NOR

The circuit simulation of up to 6-level NAND-NOR cone has been shown in Table 5 for both TIGFET 10nm and PTM models. It can be seen that in terms of only NAND and NOR configuration, 4T design shows the least average delays for various levels of AIC as compared to the PTM based designs in spite of the fact that the individual transistor in PTM HP is faster as compared to the TIGFET. Here, average delay refers to the average delay across all the four configurations of a single NAND or NOR. This is because the 4T design based NAND-NOR cell has a lower transistor count as compared to the one proposed in [24] which had 10 transistors. The smaller number of transistors leads to a fewer number of transistors in series for the pull-up and pull-down paths between the output and the V_{dd} and V_{ss} respectively. Moreover, the reconfiguration is due to an inherent reconfigurable property and not due to a circuit design aspect.

Similarly, delta_T refers to the deviation or the maximum difference between the four configurations and the average delay. This is also the measure of the delay-discrepancy. The average delay along with the delta_T is plotted for all the technologies in Figure 8. From the y-axis ranges, it is clear that the TIGFET based model performs better as compared to the PTM based model in terms of average delay. We can see that for the TIGFET based 4T, the delay discrepancy is the least when wp = 1 and wn = 1. While the delta_T is least for wp=1,wn=1 the average delay for the wp=4,wn=4 is the least. Such improvement can also help to ease out the crossbar which was required to manage this skew [24]. Variance refers to the maximum deviation with respect to the average delay per level. An important point to note is that level 4 and level 3 cones are the best with the least variance across all the technology nodes and particularly for the TIGFET models. This result helps to define the architecture and use appropriate level cones for the most optimum delay[2].

By using the NAND-NOR cell with MIGFETs, the gains can be summarized as:

- It improves the average delay of the cone owing to the simple transistor level architecture of the NAND-NOR cell, empowered by *device-level reconfigurability* to switch between the two configurations
- It reduces the delay discrepancy problem as the delay of the NAND-NOR cell is the same in both NAND and NOR configurations [50]. We believe that with better technology models, we can reduce the delay-discrepancy problem to a much greater extent which would truly ease out the overall architecture of AIC (on NAND-NOR) based architecture.

6.2 Benchmarks based evaluation

We carry out a detailed evaluation using both combinational and sequential benchmark suites for our proposed TIGFET based FPGA design and compare it with CMOS based 6-LUT design. LUT based logic cell is the ubiquitous architecture in contemporary FPGAs, and in order to truly evaluate nanotechnology, we carry out the below experiments. We used VTR [45] and EPFL [7] benchmarks for our analysis. However, we used PTM 10nm LP model and TIGFET 10nm (wp=4,wn=4) to feed the values in the architecture files. This was done due to the following reasons:

- In terms of comparison, PTM 10nm HP models gave 2 to 3 orders worst values in terms of power as compared to the TIGFET models. Hence, the comparison would have been highly skewed. PTM 10nm LP gave a more realistic comparison though it failed in terms of delay.

[2]Benchmark level evaluation is a part of our future work. In the present work we focuss on the conventional LUT based FPGA architecture

Table 5. Comparison of Delay of different configurations of a 6-level NAND-NOR cone with 4T and [24]

	NAND-NOR level	nor_a (ps)	nor_b (ps)	nand_a (ps)	nand_b (ps)	Average Delay (ps)	delta_T	Variance
TIGFET 10nm wp =1 wn =1	6	27	25.7	29.9	28.9	27.875	2.175	7.803
	5	21	24.5	24	25.5	23.75	2.75	11.579
	4	20	17.5	20.2	19.8	19.375	1.875	9.677
	3	13	16.4	14.3	16.9	15.15	2.15	14.191
	2	10.3	12	10.4	11.1	10.95	0.64	5.936
	1	5	8.1	4.6	5.7	5.85	1.25	21.368
	Geo. Mean					**15.224**	**1.642**	**10.785**
	Mean					**17.158**	**1.808**	**11.759**
TIGFET 10nm wp =4 wn =4	6	7	7.6	8.8	7.1	7.625	0.625	8.197
	5	6.1	7.3	7.3	6.3	6.75	0.65	9.630
	4	5	5.6	6.4	4.9	5.475	0.574	10.502
	3	3	5.3	4.9	4.1	4.325	1.325	30.636
	2	2.3	4.4	3.9	2.7	3.325	1.025	30.827
	1	1	3.2	2.4	1.3	1.975	0.975	49.367
	Geo. Mean					**4.472**	**0.822**	**18.387**
	Mean					**4.460**	**0.863**	**23.193**
PTM 10nm HP	6	32.1	28.2	36.1	30.2	31.65	3.45	10.900
	5	28.3	25.9	31.8	28.1	28.525	2.625	9.202
	4	25.4	21.1	29	22.6	24.525	3.425	13.965
	3	20.5	18.2	23.6	19.8	20.525	2.325	11.328
	2	17.6	13.3	20.8	14.4	16.525	3.225	19.516
	1	12	10.5	13.8	11.8	12.025	1.525	12.682
	Geo. Mean					**21.181**	**2.661**	**12.561**
	Mean					**22.296**	**2.763**	**12.932**
PTM 10nm LP	6	79.7	59.8	66.3	83.7	72.375	12.575	17.375
	5	73.4	55.4	58.6	78.6	66.5	11.1	16.692
	4	68	47.8	52.6	69.3	59.425	11.625	19.562
	3	60.2	42.1	43.4	63.3	52.25	10.15	19.426
	2	54.6	34.3	37.3	54.1	45.075	10.775	23.905
	1	50.9	28.9	27.2	48	38.75	11.55	29.806
	Geo. Mean					**54.464**	**11.271**	**20.694**
	Mean					**55.729**	**11.296**	**21.128**

- TIGFET 10nm (wp=4, wn=4) is used as compared to the (wn=1,2p=1) because of the better delay values. Further, stacked nanowire devices have been proposed earlier to have better performance [34, 35, 41].

6.2.1 Other Parameters in FPGA architecture files: An important consideration for VTR is to have all the data in the architecture file as it is important for delay and power calculations. Since we used 10nm technologies, we

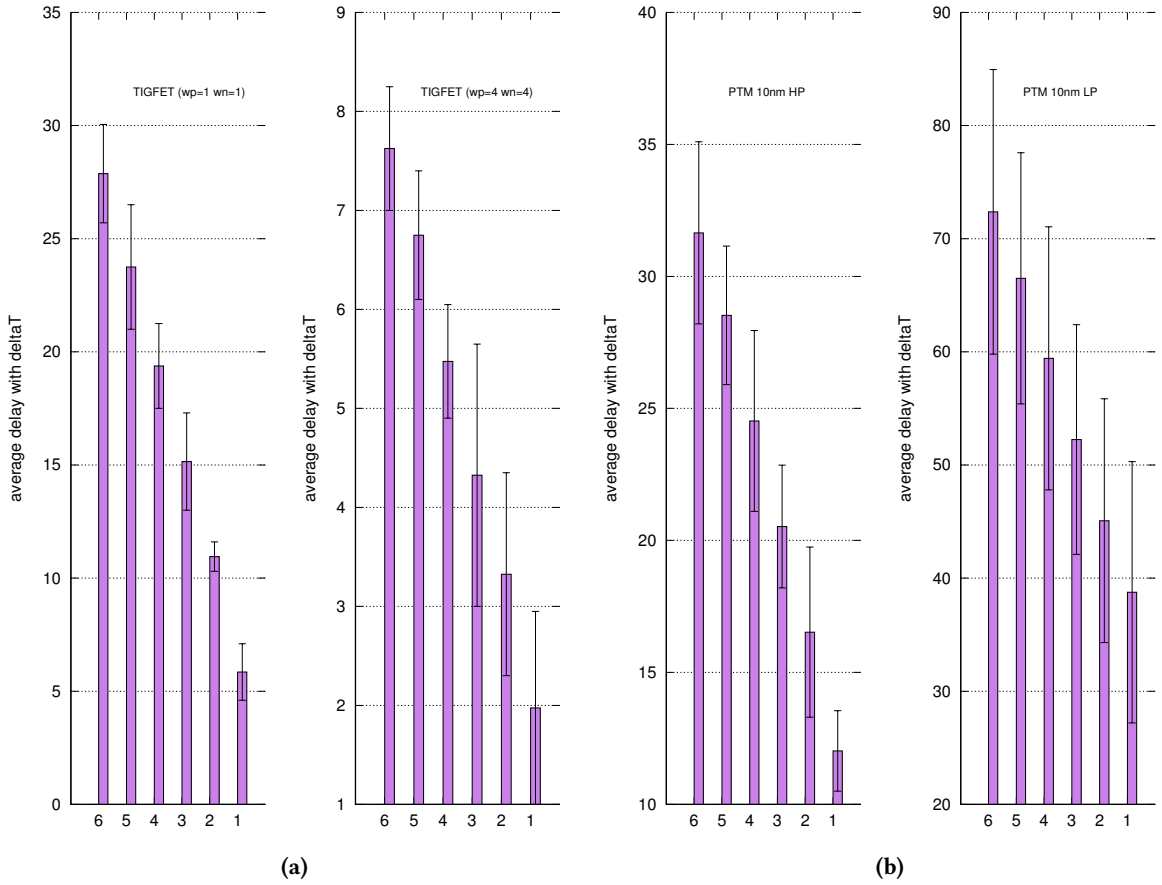


Fig. 8. Average delay and Delta_t (a) TIGFET (wp =1, wn =1) and (wp=4 and wn=4) (b) PTM High performance and low power. One can notice the delta_T shown by the black lines is the least in level 3 and 4

carried out circuit simulation to compute the other important parameters which play a major role in routing power and total delay calculations. All these parameters are listed in Table 6. The resistances and capacitances values for TIGFET lie in between the PTM LP and HP for both the MUX used in general routing as well as for the switchbox (SB) connection. Another important aspect is in terms of the resistance of minimum width nmos (R_minW_nmos) and pmos (R_min_pmos) transistors which is again lesser in TIGFET. Since there is no such distinction between the nmos and pmos transistors of TIGFET [52, 59], they have the same values in contrast to the values for PTM 10nm models. The minimum width transistor number is same for both the PTM models. For TIGFET, we have assumed the vertical stacking of four nanowires. We leave all the other parameters of architecture files like the carry chains and the multipliers in its default state.

6.2.2 Tool Flow. To evaluate and compare the proposed TIGFET based FPGA design with the baseline 6-LUT cells, VTR8.0 [45] is used to run the experimental flow with the EPFL [7] and VTR benchmarks [45]. The baseline logic cell corresponds to the conventional 6-LUT as shown in Figure 1. We use ABC [11], a logic and technology mapper to perform logic optimization on the circuit and to map it onto the functions implemented by the gates.

Parameters	PTM 10nm LP	TIGFET 10nm	PTM 10nm HP
R_minW_nmos	11622	6206	8310
R_minW_pmos	13134	6206	10514
MUX_general(wires)			
R	735	221	309
Cin	1.72E-16	1.30E-16	9.80E-17
Cout	1.88E-15	1.65E-15	6.70E-16
Tdel	49.5	16.9 ps	15.2 ps
mux_trans_size	1	4	1
buf_size	5	16	5
MUX_SB_connection_driver (ipin_cblock)			
R	735	221	309
Cin	1.72E-16	1.30E-16	9.80E-17
Cout	1.88E-15	1.65E-15	6.70E-16
Tdel	49.5	16.9 ps	15.2 ps
mux_trans_size	1	4	1
buf_size	5	16	5

Table 6. Basic parameters required for architecture files in VTR

For power calculation, the computation of signal probability and transition density is done through ACE2.0 [27], an activity estimator tool. VPR [9] (a part of VTR) is used for packing, placing and routing the benchmark circuits, which gives the cluster, area, delay and power estimates.

The architecture files are created to include the area, delay and power values of each logic cell design. The CLB organization in both proposed and baseline architectures comprises of $N = 10$ BLEs and $I = 22$ where N is the number of BLEs in a CLB and I is the number of distinct inputs to a CLB. VTR in default flow, use the technology model to calculated power and delay numbers for both the logic and the routing part in an FPGA. In order to make use of power and delay numbers for individual components of a CLB, we use the numbers as shown in Section 5. We used the "absolute" tags as mentioned in the VTR documentation and inputted individual power measurements for components like MUX, Flip-flops and 6-LUTs. This makes the VTR to use these given power and delay numbers and not to calculate on the fly based on the technology models. This also backs our previous assumption of using the default architectural parameters as they contribute only to the interconnect parameters. We use the delay and power calculated as shown in Section 5 for our experiments. However, VTR presently does not work with PTM 10nm technology model[3]. In order to carry out the experiments, we used the existing 22nm PTM technology. We have done the calculations of individual parameters of delay and power as shown previously and we used scaling ratios as mentioned in [49] with our TIGFET 10nm model to bring everything at 22nm technology scale. For the routing and interconnect, since scaling ratios are not defined for parameters like resistance and capacitances as mentioned in Section 6.2.1, we used the default values of 22nm architecture files. Since, in the present work we have focussed only on the logical part of an FPGA, we have only focussed on primitive block's power and delay numbers in our results as the improvements are primarily because of the change in the technology.

[3]A bug has been filed for the same. If the bug is resolved then we will include the 10nm results to have more confidence in the result

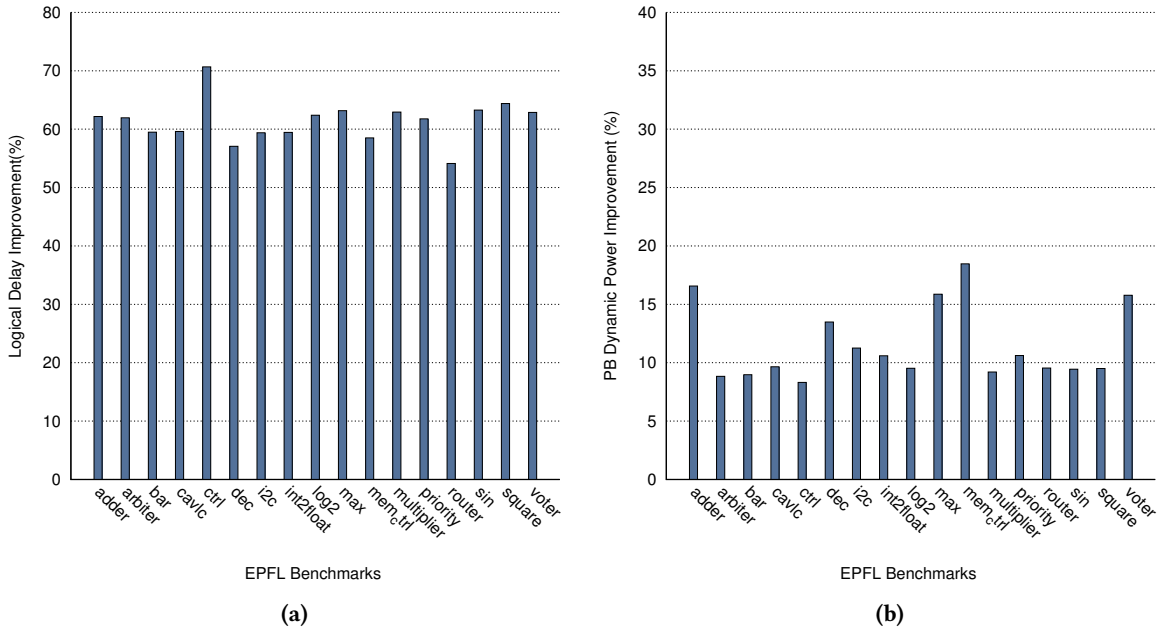


Fig. 9. Delay and Power Comparison for EPFL benchmarks (a) % Improvement in logic delay for the TIGFET based FPGA (b) PB Power Improvement % for TIGFET based FPGA

6.3 Results

The results of the experiments are shown in Figure 9 and 10. We show the percentage of improvements which the TIGFET based FPGA has over the PTM based FPGA. For each benchmark, the number of clb used is equal for both the technologies since the basic architecture is the same.

Delay: Figure 9a shows the percentage improvement in the logical delay of the TIGFET based model as compared to the PTM based model. The average percentage gain shown by the TIGFET based model is 61.37%. Similarly, Figure 10a shows the percentage improvement in critical path delay for the VTR benchmarks. The critical path delay comprises of the delay due to the logic components only. The average delay improvement comes out to be 30.41%. In both the experiments, the improvement is lesser when the number of clbs used are less. This is due to the fact that the logic delay is additive in nature and hence if the number of CLBs used are less, the total delay values will be less and hence the improvement in percentage is not too much across the technologies. The delay improvement is representative of the fact that individual components like the 6-LUT, flip-flops, and MUXes show better performance in TIGFET as compared to the PTM 10nm LP model.

PB Power: In terms of dynamic power calculations, PTM 10nm LP performs slightly better as compared to the TIGFET 10nm (wp=4, wn=4) model particularly for small circuits. In sequential benchmarks, additional contribution from flip-flops is accounted as well. An important point to note is that in VTR, in spite of using “absolute” tags for primitive blocks (pb_type), VTR still carries out some estimation using architecture file parameters under the heading "Other Estimation Methods ". This estimation is also added in power calculation for primitive blocks. Figure 9b and Figure 10b show the comparison of PB dynamic power between the two technologies for EPFL and VTR benchmarks respectively. The amount of gains is also dependent upon the size of the benchmarks. If the benchmarks need more number of CLBs, then the gain is pronounced. For combinational

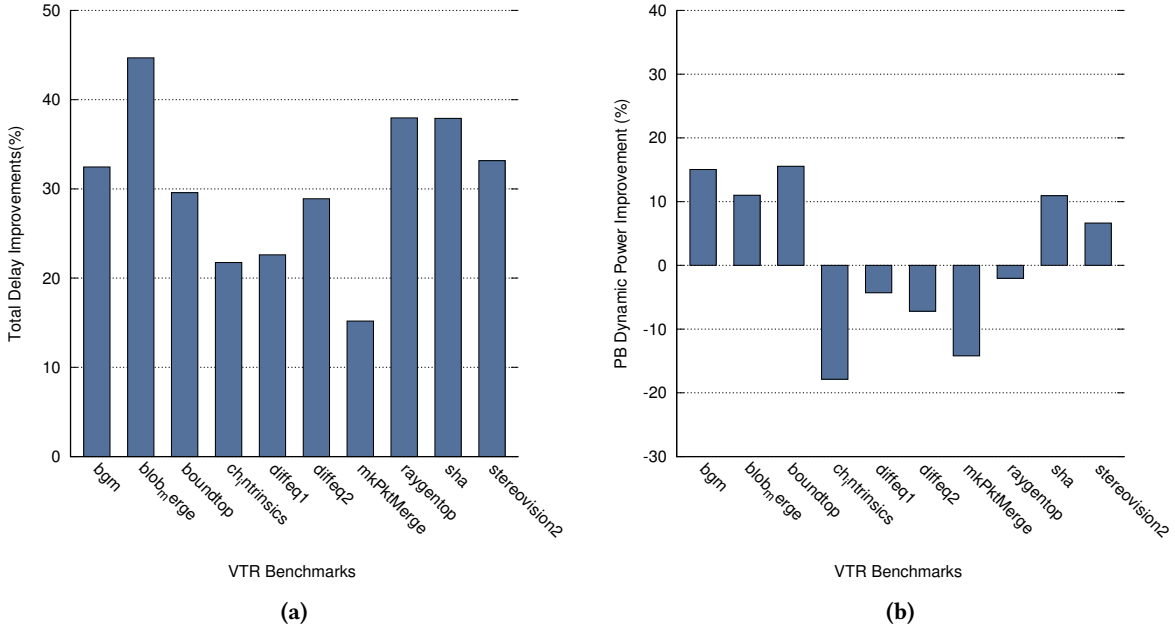


Fig. 10. Delay and Power Comparison for VTR sequential benchmarks (a) %Improvement in Total delay for the TIGFET based FPGA (b) PB Power Improvement % for TIGFET based FPGA

benchmarks, the overall improvements are almost uniform with an average of 11.51%. For sequential benchmarks, the average comes out to be 13%. In the case of sequential benchmarks, some benchmarks show negative gains because these are small benchmarks. Additionally, the gain is offseted by contributions due to "Other estimation methods" as explained above. We believe that further improvements can be achieved when VTR is used with native 10nm technology model.

The OFF state leakage power consumption of logic resources (CLBs) for TIGFETs can be expected to be up to two to three orders of magnitude less as compared to PTM model. This can be readily accounted to the Schottky junctions in nanowire-based RFETs due to which the RFETs have a very low leakage current (of the order of $10^{-14}A$) [46]. Table 1 shows static power comparison across technologies. As we see that RFETs are capable of reducing the static power up to 3 orders of magnitude, we expect that employing similar RFET-based implementations in routing can lead to pronounced total power savings as well. Since routing consideration is not possible at this moment with 22nm technology, we have left the static power consumption of the overall FPGA for future work.

7 CONCLUSION AND FUTURE WORK

The above analysis on combinational and sequential benchmarks clearly shows that from an overall perspective, reconfigurable nanotechnology can be a suitable fit for FPGA applications. The present work attempts to bridge the gap between the two ends where we have reconfigurable nanotechnology as an enabler at one end and at the other end we have reconfigurable circuits as an application. From the delay calculations, it is apparent that the logic cells can suitably replace CMOS as the overall delay is better (61.37% for EPFL and 30.41% for VTR benchmarks) as compared to the contemporary LUT based architecture. In terms of dynamic power consumption,

an average gain of 11.5% and 13% is achieved for EPFL and VTR benchmarks. The prominent benefit lies in the static power dissipation for RFET based design. Static power dissipation is considered to be one of the most crucial problems in modern FPGAs as it amounts to 70% of the total power dissipation.

There are various spheres which remain untouched as far as the use of reconfigurable technologies in FPGAs is concerned. One of them is that, by using reconfigurability at the device level, system level reconfigurability can be achieved easily. This was shown to a great extent for NAND-NOR based AIC designs where inherent device reconfigurability leads to much better power and delay numbers even compared to high performance PTM models. This has been realizable by changing the entire functionality of the logic cell by changing the pull-up network to pull-down network at runtime. Similarly, reconfiguration of this sort allows us to bring about a significant change in logic by just changing the content of one configuration bit, as opposed to k-input LUTs where changing the functionality requires rewriting the truth table of the function by changing the contents of all 2^k SRAMs. This would play considerably in decreasing the size of bitstreams, and also speed up the partial reconfiguration of FPGAs with smaller partial bitstreams. This prospect of tightly coupling reconfigurable nanotechnology with the overlaying system is very promising for conventional FPGA architectures.

Secondly, it would be interesting to investigate how routing power and delay will be affected if we use RFETs as the underlying technology for switchboxes and interconnects. We have done the basic calculation for that and we assume with the availability of VTR working with PTM 10nm technology model, we can get much better analysis. Further, using wired-AND models as suggested in [47] can improve upon existing interconnect and routing architecture for FPGA. With the development of more accurate models for emerging reconfigurable nanotechnologies especially with multi-gate RFETs, better estimates for delay and power can be achieved.

REFERENCES

[1] Ptm - predictive technology model. http://ptm.asu.edu/, 2013.

[2] Tigfet-10nm-model. https://github.com/LNIS-Projects/TIGFET-10nm-model, 2019.

[3] E. Ahmed and J. Rose. The effect of lut and cluster size on deep-submicron fpga performance and density. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(3):288–298, March 2004.

[4] Massimo Alioto, Gaetano Palumbo, and Melita Pennisi. Understanding the effect of process variations on the delay of static and domino logic. *IEEE Trans. Very Large Scale Integr. Syst.*, 18(5):697–710, May 2010.

[5] L. Amarù, P. E. Gaillardon, and G. De Micheli. Efficient arithmetic logic gates using double-gate silicon nanowire fets. In *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*, pages 1–4, June 2013.

[6] L. Amarú, P. E. Gaillardon, S. Mitra, and G. De Micheli. New Logic Synthesis as Nanotechnology Enabler. *Proceedings of the IEEE*, 103(11):2168–2195, nov 2015.

[7] Luca Amarú, Pierre-Emmanuel Gaillardon, and Giovanni De Micheli. The epfl combinational benchmark suite. In *Proceedings of the 24th International Workshop on Logic & Synthesis (IWLS)*, number CONF, 2015.

[8] M. H. Ben-Jamaa, K. Mohanram, and G. De Micheli. An efficient gate library for ambipolar cntfet logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(2):242–255, Feb 2011.

[9] Vaughn Betz and Jonathan Rose. *VPR: a new packing, placement and routing tool for FPGA research*, pages 213–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[10] Vaughn Betz, Jonathan Rose, and Alexander Marquardt. *Architecture and CAD for deep-submicron FPGAs*, volume 497. Springer Science & Business Media, Norwell, MA, USA, 2012.

[11] Robert Brayton and Alan Mishchenko. *ABC: An Academic Industrial-Strength Verification Tool*, pages 24–40. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[12] K. Cheng, S. Le Beux, and I. O'Connor. Hybrid topologies for reconfigurable matrices based on nano-grain cells. In *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8, Nov 2017.

[13] Charles Chiasson and Vaughn Betz. Should fpgas abandon the pass-gate? In *FPL*, volume 13, pages 1–8, 2013.

[14] David Chinnery and Kurt Keutzer. *Closing the gap between ASIC & custom: tools and techniques for high-performance ASIC design*. Springer Science & Business Media, 2002.

[15] André DeHon. Design of programmable interconnect for sublithographic programmable logic arrays. In *Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-programmable Gate Arrays*, FPGA '05, pages 127–137, New York, NY, USA, 2005. ACM.

[16] Andre DeHon and Michael J. Wilson. Nanowire-based sublithographic programmable logic arrays. In *Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays*, FPGA '04, pages 123–132, New York, NY, USA, 2004. ACM.

[17] Z. Ebrahimi, B. Khaleghi, and H. Asadi. Peaf: A power-efficient architecture for sram-based fpgas using reconfigurable hard logic design in dark silicon era. *IEEE Transactions on Computers*, 66(6):982–995, June 2017.

[18] Wenyi Feng, Jonathan Greene, and Alan Mishchenko. Improving fpga performance with a s44 lut structure. In *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '18, pages 61–66, New York, NY, USA, 2018. ACM.

[19] Pierre-Emmanuel Gaillardon, Michele De Marchi, Luca Amarù, Shashikanth Bobba, Davide Sacchetto, Yusuf Leblebici, and Giovanni De Micheli. Towards structured ASICs using polarity-tunable Si nanowire transistors. *Proceedings of the 50th Annual Design Automation Conference on - DAC '13*, page 1, 2013.

[20] Pierre Emmanuel Gaillardon, Xifan Tang, Gain Kim, and Giovanni De Micheli. A novel fpga architecture based on ultrafine grain reconfigurable logic cells. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(10):2187–2197, oct 2015.

[21] Naoki Harada, Katsunori Yagi, Shintaro Sato, and Naoki Yokoyama. A polarity-controllable graphene inverter. *Applied Physics Letters*, 2010.

[22] André Heinzig, Stefan Slesazeck, Franz Kreupl, Thomas Mikolajick, and Walter M. Weber. Reconfigurable silicon nanowire transistors. *Nano Letters*, 12(1):119–124, 2012. PMID: 22111808.

[23] AndrÃĺ Heinzig, Thomas Mikolajick, Jens Trommer, Daniel Grimm, and Walter M. Weber. Dually active silicon nanowire transistors and circuits with equal electron and hole transport. *Nano Letters*, 13(9):4176–4181, 2013. PMID: 23919720.

[24] Zhihong Huang, Xing Wei, Grace Zgheib, Wei Li, Yu Lin, Zhenghong Jiang, Kaihui Tu, Paolo Ienne, and Haigang Yang. Nand-nor: A compact, fast, and delay balanced fpga logic element. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '17, pages 135–140, New York, NY, USA, 2017. ACM.

[25] Global Market Insights. Field programmable gate array market size set to exceed usd 9.98 billion by 2022, with over 8.4% cagr from 2015 to 2022.

[26] Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 26(2):203–215, 2007.

[27] J. Lamoureux and S. J. E. Wilton. Activity estimation for field-programmable gate arrays. In *2006 International Conference on Field Programmable Logic and Applications*, pages 1–8, Aug 2006.

[28] Chi-Shuen Lee, Eric Pop, Aaron D Franklin, Wilfried Haensch, and Hon-Sum Philip Wong. A compact virtual-source model for carbon nanotube fets in the sub-10-nm regimeâĂŤpart ii: Extrinsic elements, performance assessment, and design optimization. *IEEE Transactions on Electron Devices*, 62(9):3070–3078, 2015.

[29] David Lewis, Elias Ahmed, Gregg Baeckler, Vaughn Betz, Mark Bourgeault, David Cashman, David Galloway, Mike Hutton, Chris Lane, Andy Lee, Paul Leventis, Sandy Marquardt, Cameron McClintock, Ketan Padalia, Bruce Pedersen, Giles Powell, Boris Ratchev, Srinivas Reddy, Jay Schleicher, Kevin Stevens, Richard Yuan, Richard Cliff, and Jonathan Rose. The stratix ii logic and routing architecture. In *Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-programmable Gate Arrays*, FPGA '05, pages 14–20, New York, NY, USA, 2005. ACM.

[30] Fei Li, Yan Lin, Lei He, and Jason Cong. Low-power fpga using pre-defined dual-vdd/dual-vt fabrics. In *Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays*, FPGA '04, pages 42–50, New York, NY, USA, 2004. ACM.

[31] Yu-Ming Lin, J. Appenzeller, J. Knoch, and P. Avouris. High-performance carbon nanotube field-effect transistor with tunable polarities. *IEEE Trans. Nanotechnol.*, 4(5):481–489, 2005.

[32] T. Luo, H. Liang, W. Zhang, B. He, and D. Maskell. A hybrid logic block architecture in fpga for holistic efficiency. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 64(1):71–75, Jan 2017.

[33] M. De Marchi, D. Sacchetto, S. Frache, J. Zhang, P. E. Gaillardon, Y. Leblebici, and G. De Micheli. Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire fets. In *2012 International Electron Devices Meeting*, pages 8.4.1–8.4.4, Dec 2012.

[34] M. De Marchi, D. Sacchetto, J. Zhang, S. Frache, P. E. Gaillardon, Y. Leblebici, and G. De Micheli. Top-down fabrication of gate-all-around vertically stacked silicon nanowire fets with controllable polarity. *IEEE Transactions on Nanotechnology*, 13(6):1029–1038, Nov 2014.

[35] H. Mertens, R. Ritzenthaler, A. Chasin, T. Schram, E. Kunnen, A. Hikavyy, L. ÃĚ. Ragnarsson, H. Dekkers, T. Hopf, K. Wostyn, K. Devriendt, S. A. Chew, M. S. Kim, Y. Kikuchi, E. Rosseel, G. Mannaert, S. Kubicek, S. Demuynck, A. Dangol, N. Bosman, J. Geypen, P. Carolan, H. Bender, K. Barla, N. Horiguchi, and D. Mocuta. Vertically stacked gate-all-around si nanowire cmos transistors with dual work function metal gates. In *2016 IEEE International Electron Devices Meeting (IEDM)*, pages 19.7.1–19.7.4, Dec 2016.

[36] Thomas Mikolajick, André Heinzig, Jens Trommer, Sebastian Pregl, Matthias Grube, Gianaurelio Cuniberti, and Walter M Weber. Silicon nanowires–a versatile technology platform. *physica status solidi (RRL)-Rapid Research Letters*, 7(10):793–799, 2013.

[37] M. Münch, B. Wurth, R. Mehra, J. Sproch, and N. Wehn. Automating rt-level operand isolation to minimize power consumption in datapaths. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '00, pages 624–633, New York, NY, USA, 2000. ACM.

[38] Hadi Parandeh-Afshar, Hind Benbihi, David Novo, and Paolo Ienne. Rethinking fpgas: Elude the flexibility excess of luts with and-inverter cones. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '12, pages 119–128, New

York, NY, USA, 2012. ACM.

[39] He Qi, Oluseyi Ayorinde, and Benton H Calhoun. An energy-efficient near/sub-threshold fpga interconnect architecture using dynamic voltage scaling and power-gating. In *Field-Programmable Technology (FPT), 2016 International Conference on*, pages 20–27. IEEE, 2016.

[40] S. Rai, M. Raitza, and A. Kumar. Technology mapping flow for emerging reconfigurable silicon nanowire transistors. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 767–772, March 2018.

[41] S. Rai, A. Rupani, D. Walter, M. Raitza, A. Heinzig, T. Baldauf, J. Trommer, C. Mayr, W. M. Weber, and A. Kumar. A physical synthesis flow for early technology evaluation of silicon nanowire based reconfigurable fets. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 605–608, March 2018.

[42] S. Rai, J. Trommer, M. Raitza, T. Mikolajick, W. M. Weber, and A. Kumar. Designing efficient circuits based on runtime-reconfigurable field-effect transistors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 1–13, 2018.

[43] M. Raitza, A. Kumar, M. VÃűlp, D. Walter, J. Trommer, T. Mikolajick, and W. M. Weber. Exploiting transistor-level reconfiguration to optimize combinational circuits. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 338–343, March 2017.

[44] Giovanni V. Resta, Surajit Sutar, Yashwanth Balaji, Dennis Lin, Praveen Raghavan, Iuliana Radu, Francky Catthoor, Aaron Thean, Pierre-Emmanuel Gaillardon, and Giovanni de Micheli. Polarity control in wse2 double-gate transistors. *Scientific Reports*, 2016.

[45] Jonathan Rose, Jason Luu, Chi Wai Yu, Opal Densmore, Jeffrey Goeders, Andrew Somerville, Kenneth B. Kent, Peter Jamieson, and Jason Anderson. The vtr project: Architecture and cad for fpgas from verilog to routing. In *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays - FPGA '12*, FPGA '12, page 77, New York, New York, USA, 2012. ACM Press.

[46] M. Simon, A. Heinzig, J. Trommer, T. Baldauf, T. Mikolajick, and W. M. Weber. Top-down technology for reconfigurable nanowire fets with symmetric on-currents. *IEEE Transactions on Nanotechnology*, 16(5):812–819, Sept 2017.

[47] M. Simon, J. Trommer, B. Liang, D. Fischer, T. Baldauf, M. B. Khan, A. Heinzig, M. Knaut, Y. M. Georgiev, A. Erbe, J. W. Bartha, T. Mikolaiick, and W. M. Weber. A wired-and transistor: Polarity controllable fet with multiple inputs. In *2018 76th Device Research Conference (DRC)*, pages 1–2, June 2018.

[48] S. Singh, J. Rose, P. Chow, and D. Lewis. The effect of logic block architecture on fpga performance. *IEEE Journal of Solid-State Circuits*, 27(3):281–287, March 1992.

[49] Aaron Stillmaker and Bevan Baas. Scaling equations for the accurate prediction of cmos device performance from 180 nm to 7 nm. *Integration*, 58:74–81, 2017.

[50] J. Trommer, A. Heinzig, T. Baldauf, T. Mikolajick, W. M. Weber, M. Raitza, and M. VÃűlp. Reconfigurable nanowire transistors with multiple independent gates for efficient and programmable combinational circuits. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 169–174, March 2016.

[51] J Trommer, A Heinzig, S Slesazeck, U Mühle, M Löffler, D Walter, C Mayr, T Mikolajick, and WM Weber. Reconfigurable germanium transistors with low source-drain leakage for secure and energy-efficient doping-free complementary circuits. In *Device Research Conference (DRC), 2017 75th Annual*, pages 1–2. IEEE, 2017.

[52] Jens Trommer, André Heinzig, Tim Baldauf, Stefan Slesazeck, Thomas Mikolajick, and Walter M. Weber. Functionality-enhanced logic gate design enabled by symmetrical reconfigurable silicon nanowire transistors. *IEEE Transactions on Nanotechnology*, 14(4):689–698, July 2015.

[53] Jens Trommer, AndrÃľ Heinzig, Uwe MÃijhle, Markus LÃűffler, Annett Winzer, Paul M. Jordan, JÃijrgen Beister, Tim Baldauf, Marion Geidel, Barbara Adolphi, Ehrenfried Zschech, Thomas Mikolajick, and Walter M. Weber. Enabling energy efficiency and polarity control in germanium nanowire transistors by individually gated nanojunctions. *ACS Nano*, 11(2):1704–1711, 2017. PMID: 28080025.

[54] T. Tuan, A. Rahman, S. Das, S. Trimberger, and S. Kao. A 90-nm low-power fpga for battery-powered applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(2):296–300, Feb 2007.

[55] Qiang Wang, Subodh Gupta, and Jason H. Anderson. Clock power reduction for virtex-5 fpgas. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '09, pages 13–22, New York, NY, USA, 2009. ACM.

[56] WM Weber, A Heinzig, J Trommer, D Martin, M Grube, and T Mikolajick. Reconfigurable nanowire electronics–a review. *Solid-State Electronics*, 102:12–24, 2014.

[57] Stephen M. Williams and Mingjie Lin. Architecture and circuit design of an all-spintronic fpga. In *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '18, pages 41–50, New York, NY, USA, 2018. ACM.

[58] Danfeng Zhang, Yao Wang, G. Edward Suh, and Andrew C. Myers. A hardware design language for timing-sensitive information-flow security. *SIGPLAN Not.*, 50(4):503–516, March 2015.

[59] J. Zhang, M. De Marchi, D. Sacchetto, P. E. Gaillardon, Y. Leblebici, and G. De Micheli. Polarity-controllable silicon nanowire transistors with dual threshold voltages. *IEEE Transactions on Electron Devices*, 61(11):3654–3660, Nov 2014.

[60] Jian Zhang, Xifan Tang, Pierre Emmanuel Gaillardon, and Giovanni De Micheli. Configurable circuits featuring dual-threshold-voltage design with three-independent-gate silicon nanowire fets. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(10):2851–2861, Oct 2014.