Exploiting Transistor-Level Reconfiguration to Optimize Combinational Circuits

Michael Raitza, Akash Kumar
Processor Design Group
Technische Universität Dresden,
Dresden, Germany,
<full.name>@tu-dresden.de

Marcus Völp
University of Luxembourg,
Luxembourg,
marcus.voelp@uni.lu

Dennis Walter
Technische Universität Dresden
Dresden, Germany,
<full.name>@tu-dresden.de

Jens Trommer,
Thomas Mikolajick,
Walter M. Weber
Namlab gGmbH
Dresden, Germany,
<full.name>@namlab.de

***Abstract*—Silicon nanowire reconfigurable field effect transistors (SiNW RFETs) abolish the physical separation of n-type and p-type transistors by taking up both roles in a configurable way within a doping-free technology. However, the potential of transistor-level reconfigurability has not been demonstrated in larger circuits, so far. In this paper, we present first steps to a new compact and efficient design of combinational circuits by employing transistor-level reconfiguration. We contribute new basic gates realized with silicon nanowires, such as 2/3-XOR and MUX gates. Exemplifying our approach with 4-bit, 8-bit and 16-bit conditional carry adders, we were able to reduce the number of transistors to almost one half. With our current case study we show that SiNW technology can reduce the required chip area by 16 %, despite larger size of the individual transistor, and improve circuit speed by 26 %.**

***Index Terms*—reconfigurable transistor, silicon nanowire transistor, RFET, TIGFET, FET, conditional sum adder, conditional carry adder, reconfigurable circuit, reconfiguration**

I. Introduction

Silicon nanowire (SiNW) reconfigurable transistor technology has benificial properties over current CMOS technology. It is able to take away the physical and logical separation of n-type and p-type transistors by design. SiNW transistors feature an additional polarity gate due to which its channel type does not have to be chosen at design time, enabling transistor-level reconfiguration. Also, as a doping-free technology, it enables p-type and n-type transistors to be freely intermixed in standard cell and full custom designs.

Even if SiNW technology cannot outperform standard CMOS technology in every aspect it can still function as a valuable backend compatible technology, as it can be produced on top of a ready-made CMOS design enhancing its functionality.

Due to functional, power supply and heat dissipation constraints reconfigurability is becoming an important aspect of digital circuit design. Also, the physical limitations that affect the realization of serial peak performance to outperform more complex parallel designs become more apparent. Silicon nanowire transistors as an emerging technology bring down the aspect of reconfigurability to the smallest element and the line that separates reconfiguration complexity and circuit overhead due to fixed reconfigurable basic elements can now be freely drawn. With transistor-level reconfigurability it is as easy to realize a three-transistor reconfigurable circuit as it is to define a lookup-table or a CGRA processing element.

Silicon nanowire (SiNW) reconfigurable field effect transistor (RFET) technology can serve both purposes in reconfigurable circuit development. It can be used to build hardware in the manner of CGRAs and FPGAs (e.g. shown in [4, 14]). Transistor level reconfigurability can also be used to build fixed function hardware with less transistors by exploiting internal reconfiguration.

Contributions: In this paper, we present a case-study of the well-known conditional sum adder circuit in its improved form proposed by Cheng et al. [2], which can be improved further using nanowire reconfigurable field effect transistors (RFETs) and internal reconfiguration on the transistor level. We make use of a gate that is reconfigurable from NAND to NOR. We contribute new gates like exclusive-OR (XOR) gates and multiplexer (MUX) gates that make use of transistor-level reconfiguration and other unique properties of nanowire transistors. Then we use these new gates to systematically improve the conditional sum adder and evaluate our improvements for different aspects and adder data widths. We use the term RFET to describe all versions of reconfigurable transistors we use in this work unless context demands for further specification.

Section II gives an overview of silicon nanowire technology, Section III describes our approach to improve the conditional carry adder, followed by our evaluation in Section IV and concluding remarks in Section V.

II. Silicon nanowire reconfigurable FET

Reconfigurable field effect transistors [7, 8] are an emerging technology with the potential to deliver an efficient ultra-fine grain reconfigurable hardware platform. The transistors used for analysis in this work feature a doping-free monocrystalline silicon nanowire channel with sharp metallic contacts of nickel silicide forming two Schottky junctions at source and drain. Several independent gates are patterned on top of this heterostructure. Typically the gate aligned above the drain contact is used to block the undesired carrier type and to set the device polarity. In a Schotty barrier bias (SBB) FET, both junctions are steered simultaneously [8]. For illustration, inputs acting as program gate will be drawn as ⊸. The channel resistance of all SiNW reconfigurable transistors is dominated by the source-sided Schottky barrier [15] and not by the channel length as in CMOS. As a consequence, the same channel can accommodate multiple gates without losing performance [14]. All implementations share, that they support the same on-current I_{on} due to the Schottky barriers being the limiting factor. As an example Figure 1 shows the design and

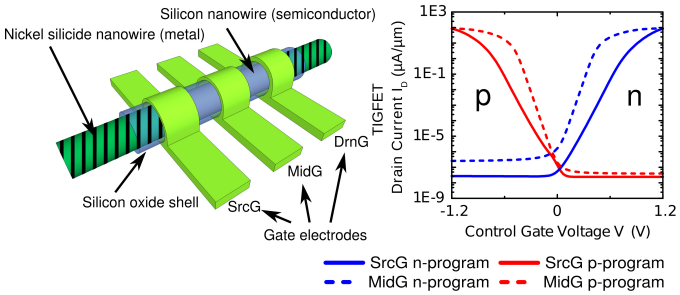


Fig. 1. Silicon nanowire three independent gate FET (TIGFET). The drain gate (DrnG) determines channel polarity. The graph shows p-type (red) and n-type (blue) behavior when using the middle gate (MidG) or the source gate (SrcG). Simulated results based on the model used in [14].

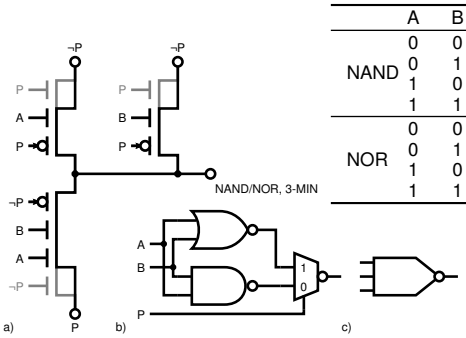
electrical characteristics of a device with three-independent gates (TIGFET).

Note, that in the on-state it exhibits equal conductivity for p-type and n-type configuration. This is a precondition to successful transistor-level reconfiguration as the channel width cannot be tuned separately for p-channel and n-channel transistors, as it is done in CMOS [6]. However, as shown in Figure 1 the subthreshold slope SS changes depending on whether the input gate is placed above the Schotty barrier or over the middle of the channel. This is reflected in Table I, where a $+$ SS signifies a faster dynamic switching. As a tradeoff, the leakage current is lower if the transistor is turned off at the source-gate. TIGFETs and MIGFETs (multiple independent gates) provide the possibility to employ both modes enabling an energy efficient multi-treshhold voltage design [17]. Besides reconfigurability all device types enable other features of functional enhancement. It was first shown by DeMarchi et al. [8] that the SBBFET concept with simultaneous junction control intrinsically yields the XOR function. A similar function can be built with TIGFETs and MIGFETs. Further, in both multigate concepts the source barrier and middle gates can be used independently as control gates, fulfilling the function of a wired-AND, only opening the channel if all control gates are active. Despite that, TIGFETs (and simple RFETs) cannot be used to drive bidirectional transmission gates due to their ambipolar characteristics in circumstances where the source drain voltage is inverted without the polarity gates following suit. Unidirectional transmission gates can be achieved with all types of transistors shown in [16]. The differences between implementations of SiNW transistors demonstrated in literature are summed up in Table I.

TABLE I

COMPARISON OF DIFFERENT SINW TRANSISTOR TYPES REGARDING ELECTRICAL AND FUNCTIONAL FEATURES. $+$ MEANS BETTER, $-$ MEANS WORSE PERFORMANCE; $\times$ MEANS ABLE TO IMPLEMENT. IN ENTRIES WITH TWO VALUES, THE LEFT CORRESPONDS TO MIDDLE GATES, THE RIGHT TO OUTER GATES.

	RFET [7]	SBBFET [8]	TIGFET [17]	MIGFET [14]
I_{on}	$=$	$=$	$=$	$=$
I_{off}	$+$	$-$	$-/+$	$-/+$
SS	$-$	$+$	$+/-$	$+/-$
Intrinsic XOR		$\times$	$(\times)$	$(\times)$
Transmission gate	$(\times)$	$\times$	$(\times)$	$(\times)$
Merge ser. paths			$\times$	$\times$



	A	B	P	Out ≡ 3-MIN
NAND	0	0	0	1
	0	1	0	1
	1	0	0	1
	1	1	0	0
NOR	0	0	1	1
	0	1	1	0
	1	0	1	0
	1	1	1	0

Fig. 2. **a)** Three-input minority gate as proposed in [11] with optimized series path proposed in [18] and truth table for reference. It uses RFETs, TIGFETs and MIGFETs to show optimization possibilities, black indicates the smallest possible configuration and gray indicates a faster extension; **b)** Equivalent circuit using CMOS NAND, NOR and MUX gates. **c)** 3-MIN Circuit symbol.

III. Conditional carry adder exploiting internal reconfiguration

The optimization of the conditional sum adder as proposed by Cheng et al. [2] reduces the main overhead in conventional circuits: the size of the multiplexer network needed to select the appropriate sum, by shifting the sum calculation from the input of the network to its output. Only the carry signals are multiplexed in the network, and the authors, thus, named the circuit Conditional Carry Adder (CCA). The reconstruction imposes a slight additional delay on the circuit, as the final sum calculation of the topmost bit now lies on the critical path.

In our approach we exploit internal reconfiguration, that is reconfiguration properties of components used in the circuit. Reconfigurability can be either internal and inaccessible or external and, thus, accessible to the user. Mathematically speaking, reconfigurability means merging separate functions into a higher order function. For instance, an FPGA is regarded as an externally reconfigurable circuit, as its mathematical representation bears no common meaning or specific topic and its meaning or practical use is defined by the user. When the circuit's function bears a common meaning and applies to a specific topic, like in our example of the conditional carry adder, it can be regarded as an internally reconfigurable circuit. External reconfigurability usually incurs a higher performance overhead as internal reconfigurability as it has to be exposed to the user. Our approach is to systematically replace the elements of the conditional carry adder with more efficient SiNW reconfigurable variants and to restructure the multiplexer network to avoid unnecessary inverters.

A. Conditional carry calculation using 3-MIN gates

At each bit position in the conditional carry adder, a logical AND and a logical OR are computed from the two input values speculating the carry bit value of the previous position. Each stage is, therefore, multiplexed at least once. When the input signals for the two bits and the select signal (originating from the carry) are combined in one function, they form a three-input minority function (3-MIN) shown in the truth table in Figure 2.

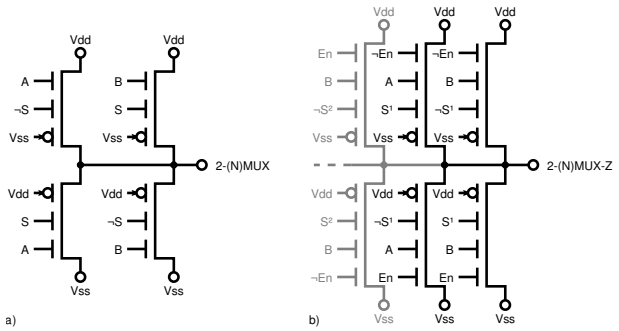


Fig. 3. **a)** Novel two-input static inverting multiplexer with TIGFETs. **b)** Enhanced version with tri-state output realized with MIGFETs.

To realize the 3-MIN functionality, we apply the reconfigurable NAND/NOR circuit, proposed in Heinzig et al. [6] and Trommer et al. [11]. Figure 2 a) shows how the input **P** (and its inverse) not only controls the polarity gates of all three transistors in this circuit but also how it serves as the transmission gate input as it is connected to the transistors' gates as well as the source contacts. This means the signals are not equally fast and must be chosen carefully. Figure 2 b) shows the 3-MIN gate built from NAND, NOR and MUX gates in standard CMOS technology. Although the 3-MIN function is commutative, the connections of the NAND and NOR gates to MUX gate are not. Swapping the inputs obviously means the select signal must be inverted for the function to remain equivalent. A similar gate pattern to Figure 2 b) occurs in the CCA in regular intervals (e.g. at Bit 3, 5 and 7) with the difference, that NAND and NOR are indeed swapped. As can be seen in Figure 2 a), the input **P** can be inverted with no additional cost. It is already available directly (**P**) and inverted (¬**P**) and just needs to be swapped in all places.

This circuit can be optimized in various ways depending on the design goals. As shown in Figure 2 a) we can easily employ transistors with various numbers of gates. We can use this, as shown by Zhang et al. [18], to shorten the serial path, making it equally fast to the parallel paths (as channel resistance does not change by adding gates). It also shows that, by adding gates and attaching signal ¬**P** to both outer gates (shown in lighter color), signals **A** and **B** all lie on inner gates and become equally fast to each other, which can be beneficial (also see Figure 1).

In contrast to that, if transistor size is of importance, the source barrier gates can be used as normal control gates as shown in black. This would also reduce the capacitive load on signal **P**, as it would only have to drive three inputs (black) instead of six (black and gray).

B. Multiplexer network

The size of the multiplexer network grows in the order of $\mathcal{O}(n^2)$ of the number of input signals n. Thus, it is a valuable target for optimization. The smallest possible two-input multiplexer (2-MUX) in standard CMOS has two transmission gates and an inverter for the select signal, which sums up to six transistors. Its output signal has to be buffered within a circuit to reach sufficient fan-out, which adds another two transistors and inverts the signal. Inverting the signal constitutes

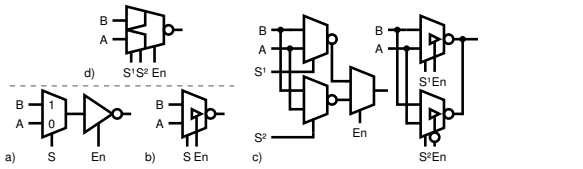


Fig. 4. **a)** CMOS circuit that realizes the enhanced 2-MUX from Fig. 3. **b)** New circuit symbol. **c)** Frequent MUX pattern in CCA and its replacement. **d)** Circuit equivalent to c) by combining both tri-state MUX gates as indicated in gray in Fig. 3 a).

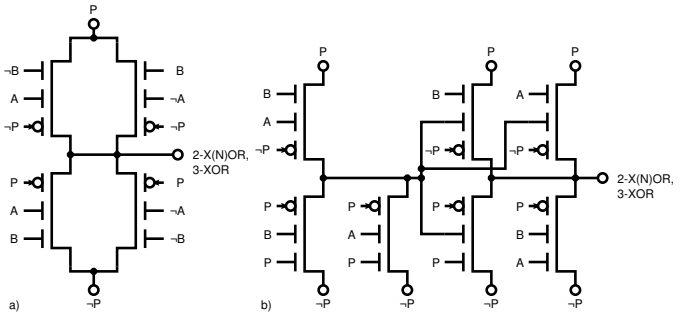


Fig. 5. Two novel XOR variants, which can be reconfigured as 2-XOR, 2-XNOR gates or used as functionally enhanced 3-XOR gates. **b)** shows a two stage design that saves one transistor (inverters for **A** and **B**) but adds a slight performance penalty to **A** and **B**.

no problem for the CCA, because this simply means that the input signals of the next multiplexer stage need to be swapped. Of course, some carry signals may arrive inverted at the final 3-XOR output stage and must therefore be inverted once more (which can be achieved inside the XOR at no further cost).

Using TIGFETs, inverting multiplexers can be built with six transistors, as seen in Figure 3 a) (counting the inverter for ¬**S**). The proposed multiplexer has a faster topology than its CMOS variant, because it is completely static and uses only one transistor stage from each input to the output. It is also to be noted, that in this design, as in the 2-NAND/NOR gate, there is only one transistor on each path from the power plane to the output, giving optimal fan-out capabilities.

Figures 3 b) and 4 shows another improvement for a common pattern in CCAs. The signals **A** and **B** are speculatively selected by $\mathbf{S}^1$ and $\mathbf{S}^2$ and finally multiplexed by **En** (see Figure 4 c). The first improvement step is to use 4-gate MIGFETs to build a 2-input inverting multiplexer with an additional tri-state enable input **En**. This allows us to connect two instances of those tri-state MUXes without the need for second MUX stage. The input of signal **En** can be inverted without further cost by inverting all uses of signal **En** and ¬**En** in the circuit.

We can build tri-state MUXes by adding a fourth gate to the transistors in a way that the corresponding pull-up pull-down networks are both enabled or disabled at the same time, as shown in Figure 3 b). Signal **En** simultaneously switches on or off a multiplexer such that instances can be linked together in an open drain design, as Figure 3 b) also shows in gray. This transistor configuration, when used with another combination of input signals, can directly be mapped to a 4-to-1 inverting multiplexer design similar to the 2-MUX design shown in Figure 3 a).

As indicated in gray, this circuit can be further optimized to spare one inverter that is used to generate ¬**En**. Due to its tri-state behavior, two tri-state MUXes can be connected and form the circuit shown in Figure 4 c); it is equivalent to the circuit shown in d) with important topological differences. Every signal crosses at most two stages, one inside the circuit itself and one inverter generating the signal's inverse; **A** and **B** cross only one stage. The **En** and ¬**En** signals power four gates each, whereas in the two-stage design in d) it was only two gates each. This means, that this optimization cannot be used where **En** lies on the critical path.

We show this new gate in Figure 6 b) calculating **C7**. It incurs a slight performance penalty (see Table III) on the critical path but shows how it can be used to replace the pattern found in Figure 4 c). In a 16-bit CCA, it can be used in similar positions.

The reduction by one stage also drops one inversion step (the second stage MUX in Figure 4 c) is non-inverting), which would have to be compensated by a subsequent inverter, as we use inverting MUX gates in our proposed design.

C. Final sum calculation using novel 3-XOR gates

As proposed in Cheng et al. [2], the final sum calculation is done with three-input XOR functions (3-XOR), which cost at least eight transistors per gate in standard CMOS design (see Fang et al. [3]) (although 16 transistors for a 2-stage implementation and 22 transistors for a fully static implementation are more realistic).

Our realization of 3-XOR follows a novel transmission gate-like realization (see Figure 5). If all four branches in the standard 2-XOR layout are replaced by TIGFETs, we obtain a fully static eight transistor 2-XOR, which can be connected in series or reconfigured, via **P**, to achieve 3-XOR functionality (Figure 5 a). Another realization was proposed by Zukoski et al. [19], which is structurally comparable. Figure 5 b) displays a fully static 2-XOR, which can be connected in series or reconfigured in the same way via **P**. Both implementations can be used in our proposed optimization of the CCA, whereby our evaluation uses implementation a), as it is the faster implementation.

IV. Evaluation

Having shown how all the elements that constitute the adder can be redesigned with RFETs, we now turn our attention to the benefits they generate for the overall circuit when compared to a standard CMOS implementation.

In our optimization, we employ the 3-MIN gate to calculate the inverse of a conditional carry signal depending on the previous carry signal. As the **P** signal of the 3-MIN gate exhibits a higher delay than the select signal of a multiplexer, care must be taken on which NAND and NOR gates to replace. Our multiplexers invert their selected input signal. Consequently, whenever the carry signal is currently delivered in its inverse polarity, one of two things have to be done: (A) the carry signal has to be inverted once more if it is fed into the final XOR stage; or (B) the carry signal has to be connected to the select input if the next stage is a multiplexer and its **A** and **B** inputs have to be swapped. The MUX gates each have

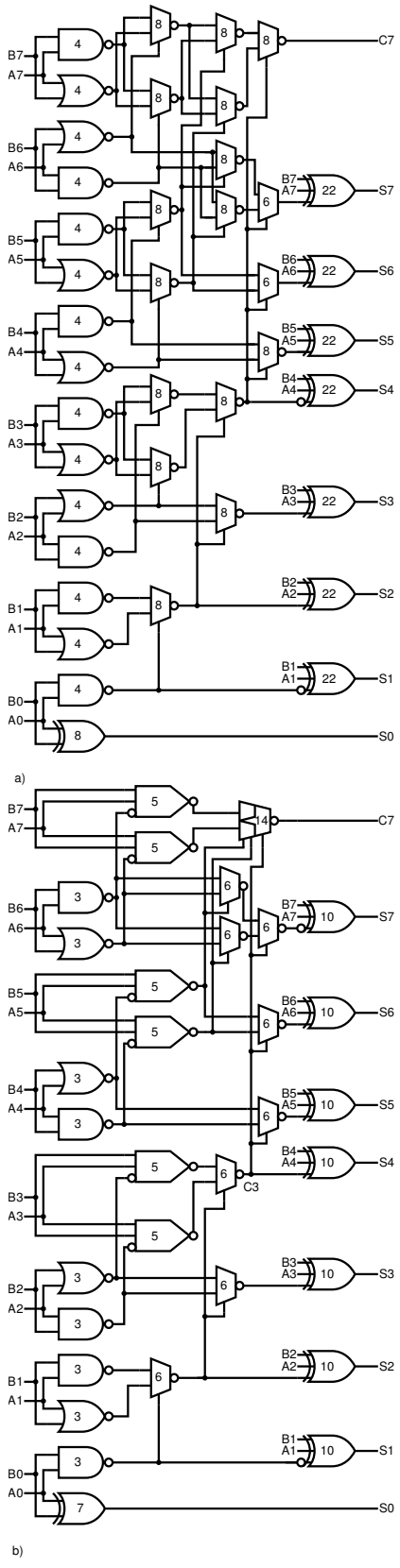


Fig. 6. a) 8-bit conditional carry adder (CCA, [2]), b) 8-bit CCA with RFETs. Numbers inside logic gates give the transistor count per gate.

TABLE II
LOGICAL EFFORT OF PROPOSED RFET GATES.

$(h=4)$		N	g	d
3-MIN implementation; Figure 2				
3-MIN	**A**, **B**	5	2.0	11.0
	$\mathbf{P_{min}}$		7.0	12.0
	$\mathbf{P_{max}}$		10.0	15.0
XOR implementation; Figure 5 a)				
2-XOR	**A**, **B**	8	2.0	10.0
3-XOR	**A**, **B**	10	4.0	17.3
	P		8.0	21.0
XOR implementation; Figure 5 b)				
2-XOR	**A**, **B**	7	1.5	10.4
3-XOR	**P**	9	15.0	30.9
2×2-XOR in series				
3-XOR	**A**, **B**	14	2.2	16.8

$(h=4)$		N	g	d
2-MUX implementation; Figure 3 a)				
2-MUX	**A**, **B**	6	1.0	6.0
	S		2.0	10.0
2-MUX-Z implementation; Fig. 4 b)				
2-MUX-Z	**A**, **B**	8	1.0	7.0
	S		2.0	11.0
	En		3.0	15.0
extended 2-MUX-Z; Figure 4 d);				
	A, **B**	14	2.0	14.0
	$\mathbf{S^1}$, $\mathbf{S^2}$		2.0	14.0
	En		3.0	18.0

only one active transistor in the output path at any given time, which makes them as effective as inverters in their driving capabilities. Therefore, we need fewer intermediate buffers in the multiplexer network, which further improves speed and power consumption.

To make a fair comparison, we implemented the CMOS variant of the CCA also with NAND and NOR gates, which are switched by standard MUX gates. These multiplexers drive an inverter to reduce the load on the input signal because of their transmission gate characteristics. In this design, there is also almost no need for additional buffers in the network (apart from the ones hidden in the inverting MUX gates). The final XOR stage is implemented in a static CMOS 3-XOR gate requiring 22 transistors.

Figure 6 shows the two circuits in CMOS technology and SiNW RFET technology side by side.

A. Logical effort

In this paper we apply the logical effort theory to give technology independent circuit results as described in Sutherland et al. [10]. This theory allows us to compare the speed of both adder circuits and their elements. It has been proven to be a viable design tool to describe and optimize the delay characteristics of VLSI circuits regardless of the used technology. See [1] for further analysis. It describes the propagation delay t_{PD} through an arbitrary combinational gate by:

$$t_{\mathrm{PD}} = \tau d, \qquad \text{with} \qquad d = gh + p. \tag{1}$$

Delay d is normalized to the intrinsic inverter delay τ in the same technology, which is the basic single stage circuit with exactly one transistor active in the output path. The fan-out of the circuit is given by h, which is also called the electrical effort. p is the parasitic delay and g the logical effort. The theory uses the conventional RC delay model in static CMOS logic gates. The delay d is proportional to the RC delay of the pull-up or pull-down network charging the output capacitance when linearly approximated. The logical effort g is a measure for the topological complexity of a logic gate. Thus, a specific logical effort g_s is given as the input capacitance C_s for signal s in the logic gate normalized to the input capacitance C_{INV} of an inverter in the same technology. For SiNW RFET technology the inverter input capacitance $C_{\mathrm{RFET,INV}} = 2$ but for standard CMOS $C_{\mathrm{CMOS,INV}} = 3$. This difference is due to the fact that in standard CMOS technology the p-channel transistor in the pull-up network has to have twice the width than the n-channel transistor to have equal performance. A wider channel and gate creates a larger gate capacitance. This also means that in standard CMOS technology it is important to avoid serial paths in pull-up networks as the added resistance must be compensated by even wider gates. In turn SiNW RFET pull-up and pull-down networks perform equally due to device symmetry.

TABLE III
SIZE AND DELAY OF CCA WITH DIFFERENT INPUT WIDTHS IN CMOS AND RFET IMPLEMENTATION. BOLD NUMBERS REPRESENT THE IMPLEMENTATION INCLUDING THE 2-MUX-Z GATE SHOWN IN FIGURE 6 B).

$(h=4)$	N_{CMOS}	N_{RFET}	$\frac{N_{\mathrm{RFET}}}{N_{\mathrm{CMOS}}}$	D_{CMOS}	D_{RFET}	$\frac{D_{\mathrm{RFET}}}{D_{\mathrm{CMOS}}}$
4-bit CCA	144	82	0.56	43.6	26.2	0.60
8-bit CCA	352	202 **196**	0.57 **0.56**	49.6	37.2 **38.2**	0.75 **0.77**
16-bit CCA	826	480 **474**	0.58 **0.57**	68.5	51.9 **52.5**	0.76 **0.77**

The full delay of a specific path from input to output can be calculated by accounting the logical effort of all gates along the path including the added effort due to branching. While the critical path characterizes a combinational circuit, the method is more general, allowing to calculate the delay of any path. Number of transistors N, logical effort g and propagation delay d of our logic gates are calculated described in [12] and are shown in Table II. When using logical effort to calculate paths through transmission gates, special care must be taken. The gate that drives the transmission gate must be taken into account to devise g. For our proposed logic gates we used an inverter at each end of the pull-up and pull-down network. In the CCA we used the logic gate actually driving the input.

The general formula to calculate the delay D for a whole path with J stages in an arbitrary circuit is (see [12] for details):

$$D = JF^{\frac{1}{J}} + \sum_{i=1}^{J} p_i \quad \text{with} \quad F = \prod_{i=1}^{J} g_i \prod_{i=1}^{J} b_i \times h \qquad \text{with} \quad b_i = 1 + \frac{C_i^{\mathrm{onpath}}}{C_i^{\mathrm{offpath}}} \quad , \tag{2}$$

where b designates the branching effort at each stage. As is noted in the top left corner of the tables, we have taken $h = 4$ as the standard load for calculating the logical effort and delays for the gates and the circuits respectively.

B. Applied improvements

Logic gate reconfiguration saves numerous transistors in the input stages of the adder, ¼ for each NAND or NOR gate and almost ⅔ for each combination of NAND, NOR, MUX gates that can be replaced by two 3-MIN gates. Each carry bit that is speculatively calculated but whose selection does not influence the length of the critical path, can be replaced by a 3-MIN gate. This reduces the transistor count from 24 (NAND, NOR, 2 MUX) down to 10 transistors (2 3-MIN) without impacting performance. According to Table II, the input **P** is structurally slower than **A** and **B**, as it has to drive more transistors and is

used as an input to transmission gates. Nevertheless, input **P** can be connected to one of the summand bits preconfiguring the 3-MIN gate into its 2-NAND or 2-NOR function. The second summand and the incoming carry bit then just perform the preconfigured function with no overhead. The delay of input **P** is too high when used for the summand bits **A1**, **B1** and **A2**, **B2**. We therefore refrain from this optimization.

Table III shows the critical path delay D for different bit widths of the CMOS and RFET CCAs as well as the transistor count N. The critical path always goes from input **A0** or **B0** along the MUX select inputs to **S3**, **S7** or **S15**, respectively. Fan-out of the output is assumed to be $h = 4$, and the input is assumed to be fed by an inverter to simplify the effort calculations. The data shows, that the same function can be achieved with RFETs with about half the amount of transistors, whereby exhibiting a structural performance gain from 26 % up to 40 %. For increasing bit widths the critical path delay of the circuit converges at around ¾ of the CMOS path delay, because the multiplexer network becomes the main contributing factor and the RFET implementation has a critical path length, that is only 75 % of its CMOS counterpart.

C. Area Consumption

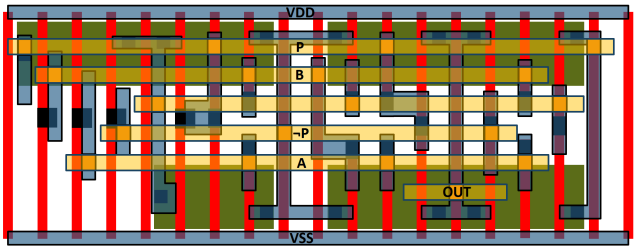


Fig. 7. Cell design of 3-XOR circuit in Fig. 5 a) in a 22 nm FDSOI process. Gates are red; zones of nanowires are green; first metal layer is semi-translucent blue, second metal layer is semi-translucent yellow.

Figure 7 shows a preliminary layout of the 3-XOR (Fig. 5 a) realized in 22 nm fully depleted silicon on insulator (FDSOI) technology. Although it hasn't been produced in this technology, it adheres to the design rules and serves as a comparison to top-of-the-line standard CMOS technology. We were able to build the cell in the same height as for CMOS. The nanowire cell is wider than its CMOS counterpart due to the geometric structure of the nanowires, especially their additional program gate. Nevertheless, the area of the 3-XOR SiNW cell is 114 % in relation to CMOS and the area of the 2-MUX cell is only 65 % compared to CMOS. Comparing the areas of all 2-MUX and 3-XOR cells of both CCAs – those two types of gates take the most area of the circuit – shows that the SiNW design takes only 84 % of the space of the CMOS design for the shown 8-bit CCA. As was projected by other authors, nanowires can also compete with CMOS technology in the area as well as the number of transistors [5], power delay product [9] and on-current I_{on} [13].

V. Conclusion

In this work, we have presented a case study of reconfigurable field effect transistors for improving the size and circuit delay of combinational circuits. For this, we showed new architectures for 2-MUX and XOR gates that make use of reconfiguration and improvements through serial path optimization using multi-gate technology. With preliminary cell designs of 3-XOR and 2-MUX cells, in a state-of-the-art silicon process technology, we could give a glimpse on the area consumption of nanowire circuits and could show that they are able to compete with CMOS. On the example of a fast block adder, we reduced transistor count to 50 % and achieved up to 40 % in circuit-level performance when compared to optimized CMOS circuits. We showed that reconfiguration enables us to increase the design density, that is, the ability to perform a certain function in a given number of transistors by improving the different elements of the adder. Our proposed adder is an example of a internally transistor-level reconfigurable system.

References

[1] Rommel M. Anacan et al. "Logical Effort Analysis of various VLSI design algorithms". In: *ICCSCE*. 2015, pp. 19–23.

[2] Kuo-Hsing Cheng et al. "Improved 32-bit Conditional Sum Adder for Low-Power High-Speed Applications". In: *Information Science and Engineering, Journal of* (2006), pp. 975–989.

[3] Sung-Chuan Fang et al. "A new direct design for three-input XOR function on the transistor level". In: *TCAS I* (1996), pp. 343–348.

[4] Pierre-Emmanuel Gaillardon et al. "A Novel FPGA Architecture Based on Ultrafine Grain Reconfigurable Logic Cells". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 23.10 (Oct. 2015), pp. 2187–2197.

[5] Pierre-Emmanuel Gaillardon et al. "Advanced System on a Chip Design Based on Controllable-polarity FETs". In: *DATE*. 2014, 235:1–235:6.

[6] André Heinzig et al. "Dually active silicon nanowire transistors and circuits with equal electron and hole transport". In: *Nano letters* 13.9 (2013), pp. 4176–4181.

[7] André Heinzig et al. "Reconfigurable silicon nanowire transistors". In: *Nano letters* 12.1 (2011), pp. 119–124.

[8] Michele de Marchi et al. "Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs". In: *IEDM*. IEEE. 2012, pp. 8–4.

[9] Kaushal Nigam et al. "DC Characteristics and Analog / RF Performance of Novel Polarity Control GaAs-Ge Based Tunnel Field Effect Transistor". In: *Superlattices and Microstructures* (2016).

[10] Ivan E. Sutherland et al. *Logical effort: designing fast CMOS circuits*. San Francisco, CA, USA: Morgan Kaufmann, 1999.

[11] Jens Trommer et al. "Elementary aspects for circuit implementation of reconfigurable nanowire transistors". In: *IEEE electron device letters* 35.1 (2014), pp. 141–143.

[12] Jens Trommer et al. "Functionality-Enhanced Logic Gate Design Enabled by Symmetrical Reconfigurable Silicon Nanowire Transistors". In: *Nanotechnology, IEEE Transactions on* 14.4 (July 2015), pp. 689–698.

[13] Jens Trommer et al. "Material Prospects of Reconfigurable Transistor (RFETs) – From Silicon to Germanium Nanowires". In: *MRS Proceedings*. Vol. 1659. 2014, pp. 225–230.

[14] Jens Trommer et al. "Reconfigurable Nanowire Transistors with Multiple Independent Gates for Efficient and Programmable Combinational Circuits". In: *DATE*. 2016.

[15] Walter M. Weber et al. "Non-Linear Gate Length Dependence of On-Current in Si-Nanowire FETs". In: *ESSDERC*. 2006, pp. 423–426.

[16] Jian Zhang et al. "Configurable circuits featuring dual-threshold-voltage design with three-independent-gate silicon nanowire FETs". In: *TCAS I* 61.10 (2014), pp. 2851–2861.

[17] Jian Zhang et al. "Dual-threshold-voltage configurable circuits with three-independent-gate silicon nanowire FETs". In: *ISCAS*. 2013, pp. 2111–2114.

[18] Jian Zhang et al. "Polarity-Controllable Silicon Nanowire Transistors With Dual Threshold Voltages". In: *Electron Devices, IEEE Transactions on* 61.11 (Nov. 2014), pp. 3654–3660.

[19] Andrew Zukoski et al. "Universal logic modules based on double-gate carbon nanotube transistors". In: *DAC*. June 2011, pp. 884–889.