

Novel Library of Logic Gates with Ambipolar CNTFETs: Opportunities for Multi-Level Logic Synthesis

M. Haykel Ben Jamaa[†], Kartik Mohanram[‡], and Giovanni De Micheli[†]

[†]Swiss Federal Institute of Technology, Lausanne, Switzerland

[‡]Department of Electrical and Computer Engineering, Rice University, Houston

Abstract

This paper exploits the unique in-field controllability of the device polarity of ambipolar carbon nanotube field effect transistors (CNTFETs) to design a technology library with higher expressive power than conventional CMOS libraries. Based on generalized NOR-NAND-AOI-OAI primitives, the proposed library of static ambipolar CNTFET gates efficiently implements XOR functions, provides full-swing outputs, and is extensible to alternate forms with area-performance tradeoffs. Since the design of the gates can be regularized, the ability to functionalize them in-field opens opportunities for novel regular fabrics based on ambipolar CNTFETs. Technology mapping of several multi-level logic benchmarks — including multipliers, adders, and linear circuits — indicates that on average, it is possible to reduce both the number of gates and area by $\sim 38\%$ while also improving performance by $6.9\times$.

1. Introduction

Carbon nanotube field effect transistors (CNTFETs) are novel devices that are projected to outperform scaled CMOS technologies [1]. CNTFET-based devices offer high mobility for near-ballistic transport, high carrier velocity for fast switching, as well as better electrostatic control due to the quasi one-dimensional structure of CNTs [1]. Early CNTFETs suffered from high variability in CNT diameter and alignment, as well as manufacturing defects due to indistinguishable metallic and semiconducting CNTs. Recent progress in addressing challenges arising from variability and defects at the technological level [2,3] as well as by defect tolerant design [4] has showcased the strong potential of CNTFET-based technology for future nanoelectronics applications.

CNTFETs can be fabricated with Ohmic or Schottky contacts, leading to MOSFET-type or Schottky-barrier-type operation, respectively. *Schottky-barrier CNTFETs (SB-CNTFETs)* are ambipolar, *i.e.*, they conduct both electrons and holes, showing a superposition of p- and n-type behaviors. Ambipolar SB-CNTFETs can be controlled by an additional terminal, called the polarity gate, which sets the p- or n-type device polarity, while the actual gate terminal controls the current flow through the transistor [5]. This novel feature of ambipolar SB-CNTFETs was investigated in [6], where a compact and in-field reconfigurable universal 8-function logic gate was described. Furthermore, the use of ambipolar SB-CNTFETs to implement in-field reconfigurable generalized NOR (GNOR) gates, combining NOR and XOR operations, was described in [7]. It was shown that GNOR gates can be utilized in a two-level programmable logic array (PLA) to save circuit area, to map logic functions into compact and fast Whirlpool PLAs [8], and to realize AND-XOR planes that efficiently map n -bit adders [9].

However, prior work with ambipolar SB-CNTFETs has only demonstrated dynamic logic, where function monotonicity require-

ments limit the potential of multi-level logic implementations. Furthermore, multi-level logic synthesis that leverages the high expressive power of ambipolar SB-CNTFETs, *i.e.*, the potential to implement more complex logic functions with less physical resources has not been investigated in literature. Unlike ambipolar SB-CNTFET logic gates that implement XOR operations in a compact form, traditional libraries provide the universal NAND, NOR, and compound AOI/OAI gates but fail to efficiently implement circuits that contain one or more binate operations such as the XOR. This makes them inefficient for circuits such as n -bit adders and parity functions that are efficiently implemented using XOR gates [9].

This paper exploits the unique in-field controllability of the device polarity of ambipolar SB-CNTFETs to design a family of full-swing static logic gates based upon SB-CNTFETs in a transmission gate configuration. Based on generalized NOR-NAND-AOI-OAI primitives that embed XORs, the family is used to build a technology library with a significantly higher expressive power than conventional CMOS libraries. To the best of our knowledge, this is the first work to describe a library of ambipolar SB-CNTFET gates that can be cascaded and used for the synthesis and mapping of multi-level logic circuits. Logic gates with no more than three SB-CNTFETs each in the pull-up (PU) and pull-down (PD) networks respectively can implement 46 functions, as compared to only 7 functions with CMOS logic having the same topology. This core family of static logic gates forms the basis for compact extensions to a pseudo logic family with transmission gates in the PD network, a static logic family with pass transistors in the PU and PD networks, and a pseudo logic family based only on pass transistors in the PD network. Since the design of the gates can be regularized, the ability to functionalize them in-field opens opportunities for novel regular fabrics based on ambipolar SB-CNTFETs. Technology mapping of several multi-level logic benchmarks — including multipliers, adders, and linear circuits — indicates that on average, it is possible to reduce both the number of gates and area by $\sim 38\%$ while also improving performance by $6.9\times$. Although not reported here, energy per cycle gains over CMOS are expected to be consistent with the $2.5\times$ reduction reported in literature [1].

This paper is organized as follows. Section 2 provides a background to SB-CNTFETs. Section 3 describes the design of the static and pseudo ambipolar SB-CNTFET logic families. Section 4 describes library implementation, characterization, and results of synthesis and mapping. Section 5 discusses regular fabrics and directions for future work. Section 6 is a conclusion.

2. Background and motivation

Although different families of CNTFETs have been demonstrated in literature, the most important distinction is between MOSFET-type and Schottky-barrier-type CNTFETs [6]. Whereas the first family is characterized by doped CNT channels and Ohmic contacts, the second family uses intrinsic CNT channels that form a Schottky-barrier (SB) at the metallic drain and source contacts. SB-CNTFETs are ambipolar, *i.e.*, they conduct both electrons and

This research was supported in part by Swiss NSF grant 20021-109450/1 and in part by NSF grant CCF-0701547. The first author thanks Prof. Subhasish Mitra for useful discussions.

holes, showing a superposition of p- and n-type behaviors. The SB thickness can be modulated by the fringing gate field at the CNT-to-metal contact, allowing the polarity of ambipolar SB-CNTFETs (CNTFETs henceforth) to be set electrically [5]. Similar ambipolar behavior has also been reported in graphene nanoribbon field-effect transistors, and suggests the possible electrical polarity control of these novel devices as well [10].

Whereas the uncontrollable ambipolar behavior — that enables transistor conduction in either gate polarity — is undesirable, the ability to control CNTFET polarity (p- or n-type) in-field by controlling the fringing gate field suggests the innovation of using a second gate, termed the *polarity gate* throughout this paper, to control the electrical field at the CNT-to-metal junction and to set the device polarity [5]. Thus, CNTFETs can be used to realize *in-field programmable ambipolar devices*, i.e., devices whose p- or n-type behavior can be programmed in-field using the polarity gate.

Several techniques to manufacture such in-field programmable CNTFETs have been proposed in literature [5, 7]. A sample device cross-section is shown in Fig. 1.a with the layout drawn in Fig. 1.b. The gate G in region A turns the device on or off, as the regular gate of a MOSFET does; the polarity gate PG in region B controls the type of polarity setting to p- or n-type. If the polarity gate is set to 0, the device exhibits n-type behavior; the device exhibits p-type behavior if the polarity gate is set to 1. The symbol for the in-field programmable CNTFET is shown in Fig. 1.c and the configuration of p- and n-type devices is illustrated in Fig. 1.d.

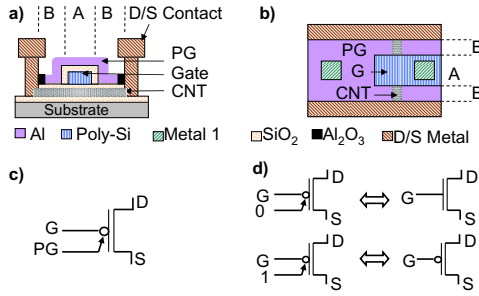


Figure 1: Ambipolar CNTFET: device (a), layout (b), symbol (c), and configuration for n- and p-type (d).

3. Ambipolar CNTFET logic families

The novel in-field programmability of CNTFETs was investigated in [6], where a compact in-field reconfigurable logic gate that maps eight different logic functions of two inputs using only seven CNTFETs was presented. In [7], the design of a *generalized NOR* (GNOR) gate was proposed as the core building block to realize in-field PLAs. It has a compact design and a high expressive power by combining both NOR and XOR operations in the output function. For example, the dynamic GNOR gate in Fig. 2 implements the function $Y = \overline{(A \oplus B) + (C \oplus D)}$ with a relatively small number of transistors, and makes use of the signals B and D as free variables. The transistors T_{PC} and T_{EV} execute the usual “precharge”

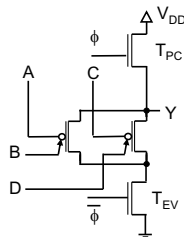


Figure 2: Dynamic GNOR gate: $Y = \overline{(A \oplus B) + (C \oplus D)}$ [7]

and “evaluate” operations in dynamic logic. However, this logic gate has two major weaknesses. First, it is based on dynamic logic that is vulnerable to internal signal races. Second, if both signals B and D are equal to 1, then the PD network will be formed exclusively by p-type devices. This can pull down the output to $\sim |V_{TP}|$ at most. The output does not provide full swing, worsening further when stages are cascaded, seriously compromising noise margins.

3.1 Transmission gate static logic family

The first innovation proposed in this paper is that analogous to CMOS gates, full swing can be restored by inserting a PU network that represents the complement of the PD network. However, the potential presence of n-type (p-type) CNTFET(s) in the PU (PD) network may still result in a degradation of the output signal. In fact, an n-type device in the PU network passes $V_{DD} - V_{TN}$ at most, and a p-type device in the PD network passes $V_{SS} + |V_{TP}|$ at least, causing signal degradation in both cases. To obtain full swing in all configurations, we replace each CNTFET whose polarity is to be set during operation time by a transmission gate formed by two CNTFETs controlled (at both the regular gate and the polarity gate) by complementary signals. In a transmission gate, both n- and p-type devices are in parallel to ensure that one of the two transistors restores the signal level in all cases (Fig. 3).

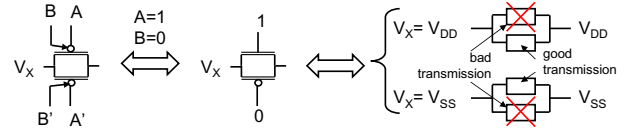


Figure 3: CNTFET transmission gate: any passing configuration ($A \oplus B = 1$) prevents signal degradation.

The second innovation proposed in this paper is to extend the GNOR gates to *generalized NAND* (GNAND) and *generalized AOI and OAI* (GAOI and GOAI) configurations, by considering series-parallel combinations of transmission gates and transistors in the PU/PD paths. Figure 4 illustrates the circuit implementation of all gates that can be obtained using no more than two transmission gates or transistors in series/parallel in the PU/PD networks. The derivation of transistor aspect ratios (W/L), indicated in the figure, will be explained in Sec. 4.

With no more than three transmission gates and transistors in the PU or PD networks, with a maximum of three inputs (applied to the gates) and three control inputs (applied to the polarity gates), we obtain 46 different logic gates listed in Table 1. Even though every transmission gate has two transistors, a topologically uniform comparison between CNTFET- and CMOS-based gates suggests that we consider CMOS gates with three inputs at most, instead of six. Then, with the same constraints and topology, we obtain only 7 CMOS-based logic gates (F00, F02, F03, F10, F11, F12, and F13), highlighting the higher expressive power of the proposed transmission-gate-based static logic family.

In this design approach, whenever the function $U \oplus V$ is implemented with transmission gate CNTFET, both polarities of U and V are needed, as illustrated in Fig. 4. By swapping the order in which the signals with different polarities are applied to the transmission gates, it is possible to implement $\overline{U \oplus V}$, $U \oplus \overline{V}$ and $\overline{U} \oplus \overline{V}$. Since $U \oplus V \equiv \overline{U} \oplus \overline{V}$ and $\overline{U \oplus V} \equiv U \oplus \overline{V}$, it is possible to implement one more function by utilizing the same resources. For example, the circuit implementing F05: $\overline{(A \oplus B) \cdot C}$ also implements $\overline{(\overline{A} \oplus \overline{B}) \cdot C}$ by swapping the inputs A and $\overline{A}$. Note that in Sec. 4.4, the technology mapping tool is aware of the existence of additional gates obtained by swapping signal polarities.

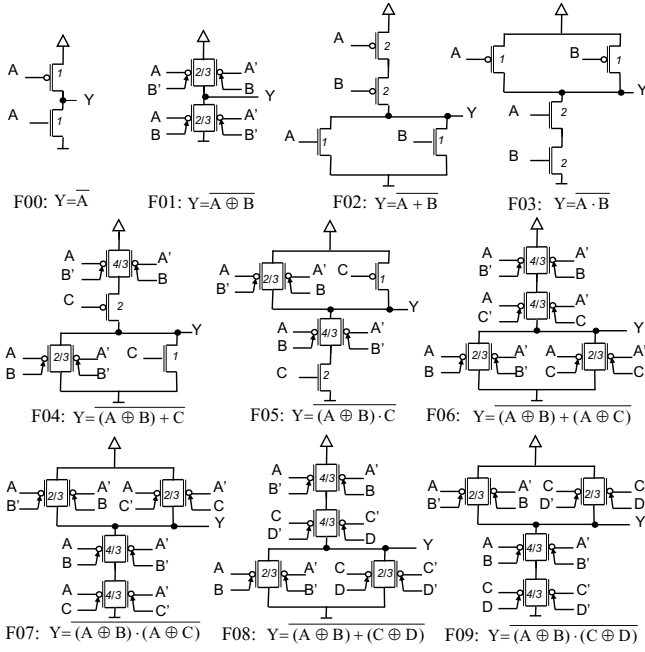


Figure 4: Circuit implementation of ambipolar CNTFET logic gates with no more than 2 transmission gates or transistors in the PU/PD networks

3.2 Alternate CNTFET families

In this section, we derive alternate CNTFET families with lower transistor count from the transmission gate static logic family. In the first approach, the transistor count can be reduced by replacing the PU network by a single PU transistor, resulting in a pseudo logic style. The PU CNTFET is weaker than the PD devices in order to allow the output signal to fall sufficiently and meet the noise margin. The gates are expected to be slower because of the weak PU network. Higher static power is also a potential concern. The pseudo logic implementation of the same set of logic functions listed in Table 1 can be derived, as illustrated in Fig. 5.a for F05.

The second approach to reduce transistor count is to replace all transmission gates by pass transistors, in static or pseudo logic configurations. Figures 5.b and 5.c illustrate the pass transistor implementations of F05, as an example, in static and pseudo logic styles respectively. However, this implies that CNTFETs that are electrically configured as p- or n-type can be located in the PU or PD network, respectively. Since this may degrade the output level, a restoration stage (inverter) is used to restore full swing at the output. The area-delay costs of this approach are assessed in Sec. 4.3.

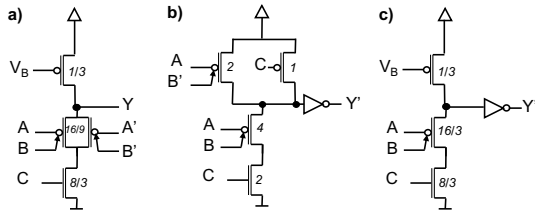


Figure 5: Compact implementation of F05: $(A \oplus B) \cdot C$: transmission gate pseudo logic (a), pass transistor static logic (b), and pass transistor pseudo logic (c)

4. Simulation results

We designed the logic gates with equal rise and fall times, and the output current is equal that of the unit inverter. Since electron

Table 1: Ambipolar CNTFET logic gates with no more than 3 series transmission-gates or transistors in each PU/PD network

Gate	Function	Gate	Function
F00	$\overline{A}$	F23	$\overline{A + (B \oplus D) \cdot C}$
F01	$\overline{A \oplus B}$	F24	$\overline{(A \oplus D) + (B \oplus D) \cdot C}$
F02	$\overline{A + B}$	F25	$\overline{A + (B \oplus D) \cdot (C \oplus D)}$
F03	$\overline{A \cdot B}$	F26	$\overline{(A \oplus D) + ((B \oplus D) \cdot (C \oplus D))}$
F04	$\overline{(A \oplus B) + C}$	F27	$\overline{(A \oplus D) \cdot B \cdot C}$
F05	$\overline{(A \oplus B) \cdot C}$	F28	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot C}$
F06	$\overline{(A \oplus B) + (A \oplus C)}$	F29	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot (C \oplus D)}$
F07	$\overline{(A \oplus B) \cdot (A \oplus C)}$	F30	$\overline{(A \oplus D) + (B \oplus E) + C}$
F08	$\overline{(A \oplus B) + (C \oplus D)}$	F31	$\overline{(A \oplus D) + (B \oplus D) + (C \oplus E)}$
F09	$\overline{(A \oplus B) \cdot (C \oplus D)}$	F32	$\overline{((A \oplus D) + (B \oplus E)) \cdot C}$
F10	$\overline{A + B + C}$	F33	$\overline{((A \oplus D) + B) \cdot (C \oplus E)}$
F11	$\overline{(A + B) \cdot C}$	F34	$\overline{((A \oplus D) + (B \oplus D)) \cdot (C \oplus E)}$
F12	$\overline{A + (B \cdot C)}$	F35	$\overline{((A \oplus D) + (B \oplus E)) \cdot (C \oplus D)}$
F13	$\overline{A \cdot B \cdot C}$	F36	$\overline{(A \oplus D) + ((B \oplus E) \cdot C)}$
F14	$\overline{(A \oplus D) + B + C}$	F37	$\overline{A + ((B \oplus D) \cdot (C \oplus E))}$
F15	$\overline{(A \oplus D) + (B \oplus D) + C}$	F38	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus E))}$
F16	$\overline{(A \oplus D) + (B \oplus D) + (C \oplus D)}$	F39	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus D))}$
F17	$\overline{((A \oplus D) + B) \cdot C}$	F40	$\overline{(A \oplus D) \cdot (B \oplus E) \cdot C}$
F18	$\overline{((A \oplus D) + (B \oplus D)) \cdot C}$	F41	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot (C \oplus E)}$
F19	$\overline{((A \oplus D) + B) \cdot (C \oplus D)}$	F42	$\overline{(A \oplus D) + (B \oplus E) + (C \oplus F)}$
F20	$\overline{((A \oplus D) + (B \oplus D)) \cdot (C \oplus D)}$	F43	$\overline{((A \oplus D) + (B \oplus E)) \cdot (C \oplus F)}$
F21	$\overline{(A + B) \cdot (C \oplus D)}$	F44	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus F))}$
F22	$\overline{(A \oplus D) + (B \cdot C)}$	F45	$\overline{(A \oplus D) \cdot (B \oplus E) \cdot (C \oplus F)}$

and hole mobility is equal in CNTs, the on-resistance of p- and n-type CNTFETs is equal. Thus, unlike CMOS gates, the PU devices in CNTFET gates need not be larger than the PD devices. This yields smaller CNTFET gates compared to the CMOS gates implementing the same function.

We simulated the correct operation of the designed CNTFET families with the Stanford CNTFET model for unipolar devices [11], using a lithography pitch of 32 nm. At the time of writing this paper, no SPICE model for controllable ambipolar CNTFETs was available. Ambipolar behavior was modeled by fixing the polarity gate signals, *i.e.*, the device polarities during simulations, along the lines suggested in [6]. All results are in comparison to the 32nm technology node for CMOS.

4.1 Transmission gate static design

We denote by R_n (R_p) the on-resistance of the n-type (p-type) device. The resistance of a transistor conducting in the weak direction is roughly double its on-resistance [12]. Hence, the resistance of a transmission gate is estimated as $R_n \parallel 2R_p$ if it conducts a low signal, and $2R_n \parallel R_p$ if it conducts a high signal. Since $R = R_n = R_p$ holds for CNTFETs, the equivalent resistance of the transmission gate is always $\sim 2R/3$. These values were taken into account in sizing the transmission gates. Note that although the decrease of the on-resistance to $\sim 2R/3$ instead of R speeds up the gates, transmission gates with a unit on-resistance have a larger area ($2 \times 2A/3$) than unit transistors (A), which may offset the speed advantages due to the higher input capacitance.

4.2 Alternate CNTFET families

The pass transistors were sized to achieve equal rise and fall times and to drive as much current as a unit inverter. Since the pass transistors potentially operate as n-type in the PU network or p-type in the PD network, their worst-case on-resistance is $2R$. Thus, they were designed to be double the unit size (area = $2A$). Despite the reduction in transistor count of the pass transistor family over the transmission gate family, the area cost to achieve unit on-resistance is higher ($2A$ vs. $4A/3$). Consequently, transmission gates are preferable to pass transistors in static logic. In pseudo logic, pass

transistors may be useful because the logic gates require no inverted inputs, unlike other logic families. We assumed for pseudo logic gates (with either transmission gates or pass transistors) that the PU device is $4\times$ weaker than the PD network, which offers a good compromise between delay and area.

4.3 Library characterization

Table 2 summarizes the area and FO4 delay estimates for the library cells. Note that the additional gates obtained by swapping the signal polarities at the transmission gates (Sec. 3.1) have the same area and delay as the gates from which they were derived. Then, we compared them to their CMOS counterparts, whenever they exist with *the same topology* and with no more than 3 transistors in the PU and PD networks, respectively. The area of the logic gates was estimated in a normalized manner as the number of transistors multiplied by their respective aspect ratios (W/L), given that all gates were designed to drive the current of a unit inverter. The FO4 delay was calculated with the switch-level RC delay model [12] and is equal to the delay of a gate driving 4 instances of itself. In this model, the FO4-delay is given by $p + 4g$, where p is the parasitic (or intrinsic) delay of the logic gate and g is the logical effort [12]. The input capacitance of the polarity gate and the actual gate were assumed to be equal. Similar to MOSFETs, we also assumed that the gate capacitance of CNTFETs is roughly equal to the drain/source parasitic capacitances. We calculated the FO4 delay on average (for all inputs) and in the worst case (for the slowest input). The FO4 delay was normalized to the delay of a unit inverter τ (defined as the delay of a fanout-of-1 inverter with no parasitic capacitances). This metric is technology-dependent and CNTFETs are roughly $5.1\times$ better than CMOS [1].

Note that the static transmission gate XNOR gate has a lower FO4 delay than the unit inverter. This is because of the lower parasitic drain capacitance of the transmission gates in the XNOR, when compared to an inverter driving the same output current. Most of the cells designed with static transmission gates present this advantage. Thus, the normalized average FO4 delay of all CNTFET transmission gate static logic gates is comparable to that for all static CMOS gates, even though the CNTFET library implements more complex functions. Simultaneously, since equally sized p- and n-type CNTFETs devices have the same on-resistance, the CNTFET cells are more compact: despite the larger average number of transistors per gate in the CNTFET static library, its average area is slightly smaller (12.3 vs. 12.7) than the CMOS library. As expected, the CNTFET transmission gate pseudo logic family has a 31% smaller average gate area than its static counterpart (8.5 vs. 12.3); however, it is 33% slower (12 vs. 9). Surprisingly, the CNTFET pass transistor pseudo logic family is less area efficient than its transmission gate counter-part. This confirms the conjecture in Sec. 4.2 that larger area is needed for pass transistors in order to compensate for the high on-resistance of p-type (n-type) transistors operating in the PD (PU) network. This family is only 7% more compact than the transmission gate static logic family (average area: 11.5 vs. 12.3), while it is $2.7\times$ slower (delay: 9 vs. 24.1). This makes the CNTFET pass transistor family a bad choice for circuit design. All the CNTFET logic families need both polarities of inputs for XOR operations. Consequently, we included an output inverter in every gate, in order to provide both polarities of every output. The average delay and area of the logic families with output inverters are indicated in the penultimate row of Table 2.

4.4 Logic synthesis and mapping results

We used the tool ABC developed at Berkeley [13] for logic synthesis and technology mapping of several benchmark circuits. The

circuits were first synthesized using the `resyn2rs` script, followed by technology mapping using `genlib` libraries that were compiled for each logic family based on the area-delay values from Table 2. The results for 15 benchmark circuits are summarized in Table 3. In Sec. 3.2 and 4.3, we demonstrated that the transmission gate configuration outperforms the pass transistor configuration in terms of area and delay. We therefore considered only transmission gate implementations in static and pseudo logic and we compared them to a CMOS library. For each family, the number of gates, the normalized circuit area (to a unit transistor), the logic depth, the normalized delay (to the technology-dependent intrinsic delay τ [1]), and the absolute delay in picoseconds are reported. Whereas both CNTFET families reduce the implementation complexity, the static family is more efficient in terms of speed and the pseudo family is more attractive in terms of area. Of the benchmarks, circuits that embed XOR operations — the adders, ALUs, error correcting circuits, and the multiplier C6288 — return the largest area and speed improvements when implemented in CNTFET technology.

The implementation with both transmission gate CNT families requires on average $\sim 38\%$ fewer gates and 40% less logic levels than CMOS. While the static logic CNTFET family saves 37.7% area on average compared to CMOS, the pseudo logic CNTFET family saves 64.5% area on average. The area normalization factor was set to the area of a unit transistor, which is expected to be equal for MOSFETs and ambipolar CNTFETs [6], since the additional polarity gate is buried underneath the channel or defined on top of the actual gate. However, we may expect a negligible area cost due to the contact area of the polarity gate.

The circuits implemented in static and pseudo CNTFET families are 26.4% and 13.0% faster than the CMOS implementation respectively in terms of normalized delay. Delay was normalized to the technology-dependent intrinsic delay τ , and unipolar CNTFETs are expected to be $5.1\times$ faster than CMOS [1]. We assumed the same intrinsic delay for unipolar and ambipolar CNTFETs to calculate the absolute delay of the logic circuits. Figure 6 shows the cumulative benefits of technology and design that translate into an average speed-up of $6.9\times$ and $5.8\times$ for static and pseudo CNTFET logic families respectively compared to CMOS. The largest speed-up was calculated for the static CNTFET implementation of multipliers ($\sim 10\times$) and error correcting circuits (more than $8\times$). For delay calculations, we considered the worst case scenario when every signal, *i.e.*, either input or control signal, needs to charge or discharge an input capacitance equal to a unit drain/source intrinsic capacitance on every switching operation. Consequently, the reported estimates for the delay of the mapped circuits are the worst-case values. Even though the delay due to signal routing around ambipolar cells was not considered, its impact is expected to be mitigated due to the advantages of smaller CNTFET cell layout.

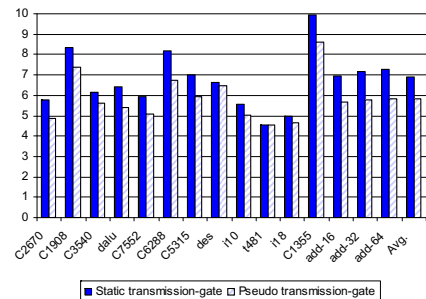


Figure 6: Ratio of the absolute delay of CMOS implementation to CNTFET implementation

Table 2: Characterization of the designed CNTFET library compared to CMOS *with the same topology*: transistor count (T), normalized area (A) to a unit transistor, normalized FO4 delay to the technology-dependent delay τ [1] in the worst case (w) and on average (a). Average performance of gates without and with output inverters is also indicated.

Gate	CNTFET Technology												CMOS Technology			
	Transmission Gate Static Logic				Transmission Gate Pseudo Logic				Pass Transistor Pseudo Logic				Static Logic			
	T	A	FO4/ τ_1 (w)	FO4/ τ_1 (a)	T	A	FO4/ τ_1 (w)	FO4/ τ_1 (a)	T	A	FO4/ τ_1 (w)	FO4/ τ_1 (a)	T	A	FO4/ τ_2 (w)	FO4/ τ_2 (a)
F00	2	2	5	5	2	1.7	7	7	2	1.7	7	7	2	2	5	5
F01	4	2.7	4	4	3	2.1	5.7	5.7	2	3	13.7	13.7	–	–	–	–
F02	4	6	8	8	3	3	8.3	8.3	3	3	8.3	8.3	4	10	8.7	8.7
F03	4	6	8	8	3	5.7	13.7	13.7	3	5.7	13.7	13.7	4	8	7.3	7.3
F04	6	7	8.2	6.6	5	3.4	8.8	7.4	3	4.3	15	13.2	–	–	–	–
F05	6	7	8.2	6.6	5	6.6	13.7	10.8	3	13.7	27	23.4	–	–	–	–
F06	8	8	10.7	8	5	3.9	11	8.6	3	5.7	27	19.9	–	–	–	–
F07	8	8	10.7	8	5	7.4	18.1	13.4	3	11	48.3	34.1	–	–	–	–
F08	8	8	6.7	6.7	5	3.9	7.4	7.4	3	5.7	16.3	16.3	–	–	–	–
F09	8	8	6.7	6.7	5	7.4	11	11	3	11	27	27	–	–	–	–
F10	6	12	11	11	4	4.3	9.7	9.7	4	4.3	9.7	9.7	6	21	12.3	12.3
F11	6	11	10.5	9.8	4	8.3	13.7	13.7	4	8.3	13.7	13.7	6	16	10.7	9.8
F12	6	11	10.5	9.8	4	7	15	13.2	4	7	15	13.2	6	17	10.3	9.9
F13	6	12	11	11	4	12.3	20.3	20.3	4	12.3	20.3	20.3	6	15	9.7	9.7
F14	8	13.3	11.2	9.4	5	4.8	10.1	8.9	4	5.7	16.3	13.7	–	–	–	–
F15	10	14.7	11.3	10.6	6	5.2	12.3	10.1	4	7	28.3	19	–	–	–	–
F16	12	16	20	12	7	5.7	16.3	11	4	8.3	40.3	24.3	–	–	–	–
F17	8	12.3	10.5	8.4	5	9.2	13.7	11.3	4	11	24.3	20.8	–	–	–	–
F18	10	13.7	13.5	9.8	6	10.1	17.2	12.7	4	13.7	45.7	28.9	–	–	–	–
F19	10	13.3	12.3	10.1	6	10.1	18.1	13.5	4	13.7	48.3	31.6	–	–	–	–
F20	12	14.7	18	10.7	7	11	25.2	14.6	4	16.3	69.7	37.7	–	–	–	–
F21	8	12	11	8.3	5	9.2	14.6	12.2	4	11	27	23.4	–	–	–	–
F22	8	12	11	8.3	5	7.4	15.4	10.7	4	8.3	16.3	16.3	–	–	–	–
F23	8	12.3	10.5	8.4	5	7.9	13.7	10.4	4	9.7	25.7	19	–	–	–	–
F24	10	13.3	12.3	9.5	6	7	15.4	12.4	4	11	37.7	24.3	–	–	–	–
F25	10	13.7	13.5	9.8	6	8.8	26.6	14.1	4	12.3	49.7	29.7	–	–	–	–
F26	12	14.7	18	10.7	7	9.2	23.4	14.6	4	7	31	17.7	–	–	–	–
F27	8	13.3	11.2	9.4	5	13.7	20.3	16.8	4	16.3	36.3	28.3	–	–	–	–
F28	10	14.7	14	10.6	6	15	20.3	10.7	4	20.3	68.3	40.3	–	–	–	–
F29	12	16	20	12	7	16.3	37.7	21.7	4	24.3	104.3	56.3	–	–	–	–
F30	10	14.7	11.3	11	6	5.2	14.1	12.5	4	7	17.7	16.6	–	–	–	–
F31	12	16	14.7	10.4	7	5.7	12.8	9.3	4	8.3	29.7	21.1	–	–	–	–
F32	10	13.7	8.8	8.2	6	10.1	13.7	10.5	4	13.7	24.3	23.2	–	–	–	–
F33	10	13.3	11	8	6	10.1	14.6	11.4	4	13.7	27	25.8	–	–	–	–
F34	14	12.7	14	9.2	7	11	18.1	12.4	4	16.3	48	31.3	–	–	–	–
F35	12	14.7	14	9.2	7	11	18.1	12.4	4	16.3	48.3	31.3	–	–	–	–
F36	10	13.3	11	8	6	8.3	15.4	10.7	4	11	27	20.6	–	–	–	–
F37	10	13.7	10.8	8.5	6	10.1	13.7	10.5	4	13.7	24.3	13.2	–	–	–	–
F38	12	14.7	14	9.2	7	9.2	19.9	12.8	4	13.7	51	29.7	–	–	–	–
F39	12	14.7	12.7	9.2	7	9.2	16.3	12.8	4	13.7	40.3	29.7	–	–	–	–
F40	10	14.7	11.3	9	6	15	20.3	15.6	4	20.3	36.3	33.1	–	–	–	–
F41	12	16	14.7	10.4	7	16.3	27	18.5	4	24.3	72.3	46.7	–	–	–	–
F42	12	16	9.3	9.3	7	5.7	9.2	9.2	4	8.3	19	19	–	–	–	–
F43	12	14.7	8.7	8.2	7	9.2	12.8	11.6	4	13.7	29.7	26.1	–	–	–	–
F44	12	16	9.3	9.3	7	16.3	16.3	16.3	4	24.3	40.3	40.3	–	–	–	–
F45	12	14.7	8.7	9.2	7	11	11	11	4	16.3	32.5	24.1	–	–	–	–
Av. w/o INV	9.1	12.3	11.3	9	5.6	8.5	15.6	12	3.7	11.5	32.5	24.1	4.9	12.7	9.1	9
Av. w/ INV	11.1	14.3	–	10.7	7.6	10.2	–	13.8	5.7	13.1	–	25.5	–	–	–	–
τ	$\tau_1 = 0.59$ ps				$\tau_1 = 0.59$ ps				$\tau_1 = 0.59$ ps				$\tau_2 = 3.00$ ps			

5. Discussion and opportunities

Since the proposed CNTFET logic gates have a higher expressive power than their CMOS counterparts, their regular structure motivates their use to design regular fabrics. A regular fabric is a set of resources (gates, memory, interconnect...) laid out in a regular manner, which can be mask- or in-field configured to implement specific logic functions. Several regular gate and logic arrays have been recently proposed to reduce the design risk due to increasing variability in current and future CMOS nodes, e.g., [8, 14–16].

The baseline architecture of an ambipolar CNTFET regular fabric is depicted in Fig 7.a. Two types of logic blocks are inter-

leaved. Their respective outputs are routed throughout the circuit by means of an interconnection network, which can be configured with SRAM cells in a similar manner to FPGAs. A detailed view of the two types of logic blocks is presented in Fig. 7.b and 7.c. The main components of the logic blocks are generalized NOR and NAND gates whose circuit implementation with CNTFET technology is presented in Fig. 8. The design takes advantage of their identical physical layout rotated by 180°. Depending on the signals connected to the inputs of the generalized gates, they can be configured in order to implement a large set of cells from the library presented in Sec. 3.1. The design of the generalized gates in

Table 3: Results for technology mapping: Gate count, normalized circuit area (to area of unit transistor), logic depth, normalized circuit delay (to technology-dependent intrinsic delay τ [1]) and absolute delay (in ps) for different benchmarks and technologies

Benchmark			CNTFET Transmission gate static logic						CNTFET Transmission gate pseudo logic						CMOS static logic					
Name	I/O	Function	Gates			Delay			Gates			Delay			Gates			Delay		
			No.	Area	Levels	Norm.	Abs.		No.	Area	Levels	Norm.	Abs.		No.	Area	Levels	Norm.	Abs.	
C2670	233/140	ALU and control	416	3292.5	12	105.2	62.1		467	1883.9	11	125.3	73.9		674	5687.0	16	120.0	360.0	
C1908	33/25	Error correcting	201	1562.2	12	106.5	62.8		207	893.6	13	120.2	70.9		502	4641.0	22	175.0	525.0	
C3540	50/22	ALU and control	642	6228.7	19	180.7	106.7		664	3475.4	19	197.6	116.6		956	8823.0	29	218.2	654	
dalu	75/16	Dedicated ALU	679	6662.3	16	163.6	96.5		713	3956.8	17	193.5	114.2		1100	9181.0	28	205.9	617.7	
C7552	207/108	ALU and control	904	6747.6	17	149.1	88.0		987	4235.7	17	174.4	102.9		1860	13933.0	24	173.6	520.8	
C6288	32/32	Multiplier	1389	11672.9	48	397.8	234.7		1322	6558.0	48	481.6	284.1		2767	23192.0	89	639.8	1919.4	
C5315	178/123	ALU and selector	894	7600.6	16	145.6	85.9		986	4553.2	17	172.2	101.6		1465	12048.0	27	200.2	600.6	
des	256/245	Data encryption	2583	25781.1	10	88.1	52.0		2500	13920.0	9	90.8	53.6		3560	35781.0	15	115.3	345.9	
i10	257/224	Logic	1279	11264.2	19	200.0	118.0		1287	6296.2	21	222.3	131.2		1965	16394.0	29	218.8	656.4	
t481	16/1	Logic	670	6379.0	12	113.7	67.1		598	3516.0	11	114.0	67.3		804	8259.0	13	102.2	306.6	
i18	133/81	Logic	674	6642.0	8	83.6	49.3		714	3698.6	9	89.8	53.0		836	7968.0	11	82.1	246.3	
C1355	41/32	Error correcting	207	1260.2	9	63.9	37.7		215	776.6	9	73.6	43.4		579	5376.0	16	125.0	375.0	
add-16	33/17	16-bit adder	128	834.4	19	179.2	105.7		132	540.0	20	220.0	129.8		217	1548.0	33	244.6	733.8	
add-32	65/33	32-bit adder	256	1656.7	35	340.5	200.9		260	1091.4	36	421.6	248.7		441	3084.0	65	479.1	1437.3	
add-64	129/65	64-bit adder	512	3321.0	67	663.1	391.2		516	2194.1	68	824.8	486.6		889	6156.0	129	948.3	2844.9	
Average			762.3	6727.0	21.3	198.7	117.2		771.2	3839.3	21.7	234.8	138.5		1241.0	10804.7	36.4	269.9	809.7	
Improvement vs. CMOS			38.6%	37.7%	41.5%	26.4%	6.9×		37.9%	64.5%	40.4%	13.0%	5.8×		-	-	-	-	-	-
Delay normalization factor [1]			$\tau_1 = 0.59$ ps						$\tau_1 = 0.59$ ps						$\tau_2 = 3.00$ ps					

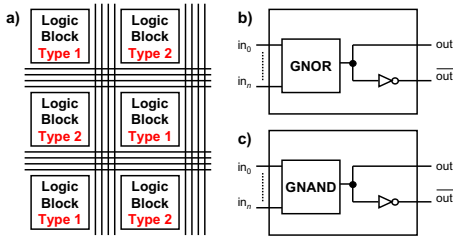


Figure 7: Baseline architecture of a CNTFET regular fabric (a) and type 1 and type 2 logic blocks (b and c)

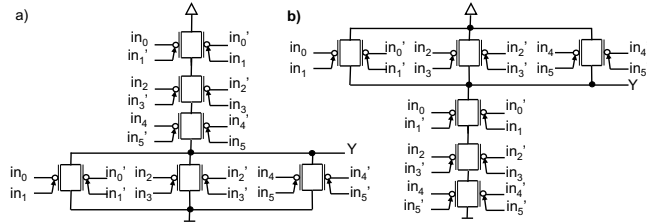


Figure 8: (a) GNOR and (b) GNAND gates for regular fabrics

the other logic families can be derived in a straightforward manner from the static transmission gate family.

In-field programmable regular fabrics offer a simple design flow, reconfigurability, and immunity to process variability. The regularity and symmetry allows easy bounding of delays, and if the local routing delay is small enough, then the gates can be designed with dynamic logic with no risk of internal signal races. This yields a more robust dynamic logic, while taking advantage of its lower power and area compared to static logic.

6. Conclusions

This paper described novel design guidelines for logic gates based on in-field reconfigurable ambipolar CNTFETs. The logic gates embed XOR operations efficiently, offering higher expressive power than equivalent CMOS gates. When used to map several benchmark circuits, the proposed static transmission gate CNTFET li-

brary requires on average 37.7% less area than static CMOS, with further reductions possible with pseudo logic CNTFETs. Multipliers, adders and error correcting circuits showed the highest improvement with up to 76.6% savings in area. The lower logic depth and lower intrinsic delay of CNTFETs result in an average circuit speed-up of 6.9×. The regular structure of the proposed gates can be exploited to manufacture regular fabrics, offering enhanced flexibility and robustness within a simple design framework.

References

- [1] J. Deng *et al.*, “Carbon nanotube transistor circuits: Circuit-level performance benchmarking and design options for living with imperfections,” in *Proc. Intl. Solid-State Circuits Conference*, pp. 70–588, 2007.
- [2] G. Zhang *et al.*, “Selective etching of metallic carbon nanotubes by gas-phase reaction,” *Science*, vol. 314, pp. 974–977, 2006.
- [3] S. J. Kang *et al.*, “High-performance electronics using dense, perfectly aligned arrays of single-walled carbon nanotubes,” *Nature Nanotechnology*, vol. 2, pp. 230–236, 2007.
- [4] N. Patil *et al.*, “Automated design of misaligned-carbon-nanotube-immune circuits,” in *Proc. Design Automation Conference*, pp. 958–961, 2007.
- [5] Y.-M. Lin *et al.*, “High-performance carbon nanotube field-effect transistor with tunable polarities,” *Trans. Nanotechnology*, vol. 4, pp. 481–489, 2005.
- [6] I. O’Connor *et al.*, “CNTFET modeling and reconfigurable logic-circuit design,” *IEEE Trans. Circuits and Systems I*, vol. 54, pp. 2365–2379, 2007.
- [7] M. H. Ben-Jamaa *et al.*, “Programmable logic circuits based on ambipolar CNTFET,” in *Proc. Design Automation Conference*, pp. 339–340, 2008.
- [8] F. Mo and R. K. Brayton, “Whirlpool PLAs: a regular logic structure and their synthesis,” *Proc. Intl. Conference Computer-aided Design*, pp. 543–550, 2002.
- [9] T. Sasao, *Switching theory for logic synthesis*. Kluwer Academic Publishers, 1999.
- [10] A. K. Geim and K. S. Novoselov, “The rise of graphene,” in *Nature Materials*, vol. 6, pp. 183–191, 2007.
- [11] “Stanford University CNTFET model.” Please visit the URL <http://nano.stanford.edu/models.php> for further details.
- [12] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Pearson - Addison Wesley.
- [13] “ABC Logic synthesis tool.” Please visit the URL <http://www.eecs.berkeley.edu/~alanmi/abc/> for further details.
- [14] V. Kheterpal *et al.*, “Design methodology for IC manufacturability based on regular logic-bricks,” in *Proc. Design Automation Conference*, pp. 353–358, 2005.
- [15] Y. Ran and M. Marek-Sadowska, “Designing via-configurable logic blocks for regular fabric,” *IEEE Trans. VLSI Systems*, vol. 14, pp. 1–14, 2006.
- [16] J. Brockman *et al.*, “Design of a mask-programmable memory/multiplier array using G4-FET technology,” in *Proc. Design Automation Conference*, pp. 337–338, 2008.