An Efficient Gate Library for Ambipolar CNTFET Logic

M. Haykel Ben-Jamaa, *Member, IEEE,* Kartik Mohanram, *Member, IEEE,*
and Giovanni De Micheli, *Fellow, IEEE*

***Abstract*—Recently, several emerging technologies have been reported as potential candidates for controllable ambipolar devices. Controllable ambipolarity is a desirable property that enables the on-line configurability of n-type and p-type device polarity. In this paper, we introduce a new design methodology for logic gates based on controllable ambipolar devices, with an emphasis on carbon nanotubes as the candidate technology. Our technique results in ambipolar gates with a higher expressive power than conventional complementary metal-oxide-semiconductor (CMOS) libraries. We propose a library of static ambipolar carbon nanotube field effect transistor (CNTFET) gates based on generalized NOR-NAND-AOI-OAI primitives, which efficiently implements XOR-based functions. Technology mapping of several multi-level logic benchmarks that extensively use the XOR function, including multipliers, adders, and linear circuits, with ambipolar CNTFET logic gates indicates that on average, it is possible to reduce the number of logic levels by 42%, the delay by 26%, and the power consumption by 32%, resulting in a energy-delay-product (EDP) reduction of 59% over the same circuits mapped with unipolar CNTFET logic gates. Based on the projections in [1], where it is stated that defect-free CNTFETs will provide a 5× performance improvement over metal-oxide-semiconductor field effect transistors, the ambipolar library provides a performance improvement of 7×, a 57% reduction in power consumption, and a 20× improvement in EDP over the CMOS library.**

***Index Terms*—Ambipolar carbon nanotubes, ambipolar silicon nanowires, ambipolarity, logic design, logic synthesis.**

I. Introduction

THE CONTINUOUS scaling of metal-oxide-semiconductor field effect transistors (MOSFETs) led to the consideration of devices with intrinsic channel and Schottky barrier (SB) contacts. Such transistors are *ambipolar*, i.e., they behave either as n-type or p-type devices, depending on the bias conditions. A back gate can be used in order to control the device polarity. It has been shown recently that the unique property of in-field polarity control can yield denser and faster design of reconfigurable logic circuits [2], [3].

Different technologies represent potential candidates for ambipolar logic, including silicon nanowire field effect transistors [4], carbon nanotube field effect transistors (CNTFETs) [5], and graphene nanoribbons [6]. Devices fabricated in any of these technologies have two gates, controlling both current conduction and device polarity as illustrated in Fig. 1. The ultimate goal of design using these devices is to leverage their controllable ambipolarity at the logic level, which yields a very compact realization of the XOR function, and its potential embedding into more complex logic gates.

It has been suggested in previous work that embedding the XOR operation into other gates results in generalized logic gates [2], i.e., reconfigurable gates whose input polarities can be set in the field. This results in a higher *expressive power*, i.e., the potential to implement more complex functions using fewer physical resources. In [2], generalized NOR (GNOR) gates, combining NOR and XOR operations, were described. In [3], the investigation of the potentials of ambipolar logic gates was restricted to the characterization of a single universal reconfigurable 8-function gate. The intrinsic reason for the high expressive power lies in the presence of an inverting device behavior (as, for example, a p-device in the pull-down network or vice versa) enabling the realization of binate functions in single logic gates.

However, prior work with ambipolar devices has only demonstrated dynamic logic, where function monotonicity requirements limit the potential of multi-level logic implementations. Furthermore, multi-level logic synthesis that leverages the high expressive power of ambipolar devices has not been investigated in the literature. Unlike ambipolar 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 [8].

This paper exploits the unique in-field controllability of the device polarity of ambipolar transistors. It deals with design aspects of ambipolar gates, and it is applicable to SiNW, CNT, and graphene technologies. Whereas this paper proposes

Manuscript received December 6, 2009; revised April 20, 2010, July 25, 2010, and August 20, 2010; accepted August 24, 2010. Date of current version January 19, 2011. This work was supported in part by the Swiss NSF, under Grants 20021-109450/1 and ERC-2009-AdG-246810, by the U.S. NSF, under Grant CCF-0916636, and by the ERC, under Senior Grant NANOSYS ERC-2009-AdG-246810. This paper was recommended by Associate Editor I. Markov.

M. H. Ben-Jamaa is with Commissariat à l'Energie Atomique et aux Energies Alternatives, Grenoble 38000, France (e-mail: haykel.ben-jamaa@cea.fr).

K. Mohanram is with the Departments of Electrical and Computer Engineering and Computer Science, Rice University, Houston, TX 77005 USA.

G. D. Micheli is with École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2085250

0278-0070/$26.00 © 2011 IEEE

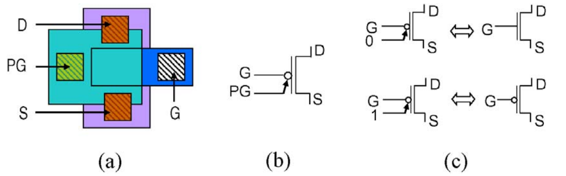


Fig. 1. Double gate ambipolar transistor. (a) Layout. (b) Symbol. (c) Configuration as n-type and p-type.

a general methodology, it specializes on SB CNTFETs as an example of the investigated devices in order to design a family of full-swing static logic gates in a transmission gate configuration. The contributions of this paper can be summarized as follows.

1) 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 complementary metal-oxide-semiconductor (CMOS) libraries, which targets the design of circuits that extensively use the XOR function.
2) This paper extends our approach in [9] by enhancing the library characterization in terms of area and delay with a power model.

In this paper, we demonstrate that logic gates with no more than three ambipolar devices each in the pull-up (PU) and pull-down (PD) networks, respectively, can implement 46 different functions, which can be extended to 158 different derived functions (Section III). This is in contrast to only seven functions with CMOS logic using the same topology. This core family of static logic gates can be extended 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. Technology mapping of several multi-level logic benchmarks, including multipliers, adders, and linear circuits, using ambipolar CNTFET logic gates indicates that on average, it is possible to reduce the number of logic levels by 42%, the delay by 26%, and the power consumption by 32%, resulting in an energy-delay-product (EDP) reduction of 59% over the same circuits mapped with unipolar CNTFET logic gates. Based on the prediction given in [1], stating that defect-free CNTFETs have a 5× better performance than MOSFETs, the performance improvement over the CMOS mapping is 7×, while the power consumption is reduced by 57% and the EDP by 20×. The proposed design approach in this paper can be generalized to other transistor technologies as long as the device ambipolarity can be controlled after manufacturing. However, the benefits will depend on the underlying technology.

This paper is organized as follows. Section II provides a background and surveys technologies and circuit design approaches based on ambipolar devices. Section III introduces the novel design approach based on transmission-gate static logic gates with ambipolar CNTFETs. Then, Section IV extends this static family to pseudo-logic using either transmission gates or pass-transistors. Section V is dedicated to the characterization of the designed libraries in terms of delay and weighted device count; then large logic circuits are synthesized and mapped using those gates in Section VI. In Section VII, a model to estimate the power consumption of the static transmission gates is introduced. The power consumption of the proposed library is estimated in Section VIII and compared to CMOS gates. Then, the power consumption of the previously synthesized and mapped circuits is estimated in Section IX. Section X summarizes the comparison between unipolar and ambipolar design and discusses future directions for further assessment of the ambipolar design methodology. In Section XI, we conclude this paper.

II. Background and Motivation

This section surveys previous works related to physics and technology of ambipolar CNTFETs, which illustrates the ambipolar technologies underlying our proposed design methodology. It also summarizes previous approaches to leverage the controllable ambipolarity at the circuit level.

A. *Ambipolar Technologies*

Several technologies represent potential platforms for the design of ambipolar logic gates. Recently, ambipolar behavior has been reported on silicon nanowire field effect transistors (SiNWFETs) [10]. Moreover, GNRFETs have an ambipolar behavior when their width is confined to less than 10 nm [6], [11]. Ambipolar behavior has also been reported in CNTFETs [12]. In this paper, we focus on CNT technology to illustrate the general design methodology for ambipolar gates, since CNTFETs have been shown to deliver higher performance than other technologies [1].

When intrinsic CNTs are used as the channel material in CNTFETs, then the fabricated devices have a Schottky barrier at the contacts and exhibit ambipolar behavior, i.e., they conduct both electrons and holes, showing a superposition of n-type and p-type behaviors. The Schottky barrier thickness is modulated by the fringing gate field at the CNT-to-metal contact, allowing the polarity of the device to be set electrically [12]. The ability to control CNTFET polarity (p-type or n-type) in the 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 [12]. Thus, CNTFETs can be used to realize *in-field programmable ambipolar devices*, i.e., devices whose p-type or n-type behavior can be programmed in the field using the polarity gate.

The physics of the considered double-gate device is illustrated using the band diagram in Fig. 2. The fields A and B are integral to the channel, i.e., the CNT, and are controlled by the conventional and the polarity gate, respectively. By setting the polarity gate to a positive value ($V_{\mathrm{PG}} = V_{+}$), the band diagram becomes thin and transparent for electrons (e^{-}), which can tunnel through the SB, and it remains thick for holes (h^{+}), which are then stopped by the SB. A resulting electron current can flow along the channel, as long as the gate voltage is positive ($V_{\mathrm{G}} > 0$), thus making the device operate as a n-type transistor. The opposite happens when the polarity gate is set to

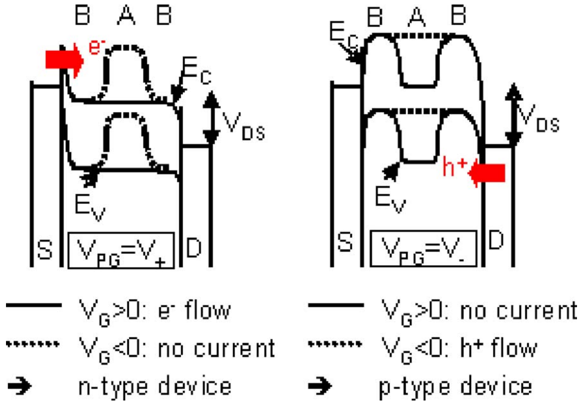


Fig. 2. Band diagram of a double-gate ambipolar CNTFET: the fields A and B are controlled by the conventional and the polarity gate, respectively. The band diagram of the n-type and p-type behaviors of the same device under different polarity gate biases are illustrated on the left and right sides, respectively.

a negative value ($V_{\mathrm{PG}} = V_{-}$). Then the SB becomes transparent for holes and the device operates as a p-type transistor.

B. Operation of Ambipolar Devices

A technique to manufacture in-field programmable CNTFETs based on double-gate devices has been proposed in [12]. The device layout, including a bottom gate that is different from the substrate has been introduced in Fig. 1(a); and its symbol used in this paper has been shown in Fig. 1(b). The device has two gates G and PG. The *gate* G turns the device on or off, as the regular gate of a MOSFET, while the polarity gate (PG) controls the type of polarity setting to p-type or n-type. If a large positive voltage is applied at PG, the device behaves as a n-type transistor, while a large negative voltage applied at PG would set the polarity to p-type. For simplicity with respect to logic design with ambipolar devices, which will be introduced in the following sections, the logic 0 at PG is defined as the required positive voltage to set the n-type polarity, while the logic 1 is defined as the required negative voltage to set the p-type polarity [Fig. 1(c)]. Note that logic 0 and 1 may correspond to different voltage levels at G and PG, and we discuss this in greater detail in Section X.

We assume in this paper that the technological aspects challenging the operation of CNTFETs, such as variability and lack of control of the chirality, diameter, and placement of the fabricated CNTs, will be addressed as the technology matures. Our focus is on the ideal operation of the ambipolar CNTFETs. Our proposed methodology can be generalized to other technologies, as long as the ambipolarity can be controlled in the field.

C. Previous Design Approaches with Ambipolar Devices

The novel in-field programmability of CNTFETs was investigated in previous works in order to extend the possibilities offered by MOSFETs. In [3] and [13], a compact in-field reconfigurable logic gate that maps eight different logic functions of two inputs using only seven CNTFETs in dynamic logic was presented. A full adder and an arithmetic logic unit (ALU) were designed using this reconfigurable logic gate in [14]. Similarly, another reconfigurable 6-function logic gate including XOR and XNOR was designed in [15] and used in a matrix-based regular architecture to map logic circuits.

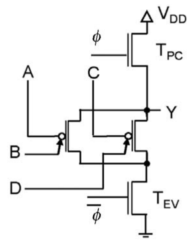


Fig. 3. Dynamic GNOR gate: $Y = \overline{(A \oplus B) + (C \oplus D)}$ [2].

In [2], the design of a 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. 3 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 $\mathrm{T_{PC}}$ and $\mathrm{T_{EV}}$ execute the usual "precharge" and "evaluate" operations in dynamic logic.

These GNOR-based PLAs offer the opportunity of mapping logic functions into the compact and fast Whirlpool PLAs [16]. Another option is the realization of AND-XOR PLAs. It has been shown [17] that such AND-XOR planes efficiently map specific families of logic functions, including adders. In [17], the design using XOR function has been investigated in an independent way on the underlying technology. Some functions, such as parity functions, can be efficiently implemented in circuits using XOR gates.

III. Ambipolar Static Transmission-Gate Logic

Previous approaches using ambipolar devices in logic design are based on dynamic logic. However, dynamic logic has two major weaknesses when combined with ambipolar devices. First, it is vulnerable to internal signal races. Second, in the example depicted in Fig. 3, 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 $V_{\mathrm{SS}} + |V_{\mathrm{Tp}}|$ at most. The output does not provide full swing, and worsens further when stages are cascaded, seriously compromising noise margins. Another disadvantage of dynamic logic is that dynamic logic gates implement functions that tolerate only monotonic transitions at the outputs.

We can possibly think of compensating the cascading issue by adding a restoration stage (inverter or a buffer) at the function output to restore the output swing. However, this represents a certain area and delay overhead to the logic gates. In addition, this option does not address the issue of monotonicity of the implemented functions. Similar to CMOS, it is possible extend the design of ambipolar gates to complementary static logic by inserting a PU network that represents the complement of the PD network. Whereas this solves the problem of monotonicity, the potential presence of n-type (p-type) CNTFET(s) in the PU (PD) network may still

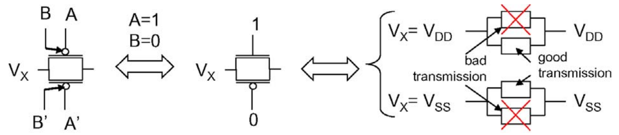


Fig. 4. CNTFET transmission gate: any passing configuration ($A \oplus B = 1$) prevents signal degradation.

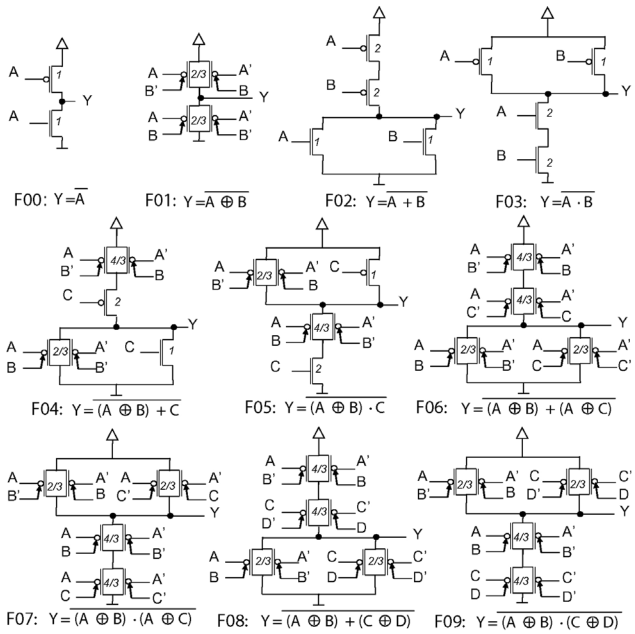

F00: Y=$\overline{A}$
F01: Y=$\overline{A \oplus B}$
F02: Y=$\overline{A+B}$
F03: Y=$\overline{A \cdot B}$
F04: Y=$\overline{(A \oplus B)+C}$
F05: Y=$\overline{(A \oplus B) \cdot C}$
F06: Y=$\overline{(A \oplus B)+(A \oplus C)}$
F07: Y=$\overline{(A \oplus B) \cdot (A \oplus C)}$
F08: Y=$\overline{(A \oplus B)+(C \oplus D)}$
F09: Y=$\overline{(A \oplus B) \cdot (C \oplus D)}$
1
2
2/3
4/3
Y

Fig. 5. Circuit implementation of ambipolar CNTFET logic gates with no more than two transmission gates or transistors in the PU/PD networks.

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.

In order to obtain a static design and guarantee full voltage swing in all configurations, we replace each CNTFET whose polarity is to be set during operation 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-type and p-type devices are in parallel to ensure that one of the two transistors restores the signal level in all cases (Fig. 4).

We combine this approach with the extension of the GNOR gates to generalized NAND and generalized AOI and OAI (GAOI and GOAI) configurations, by considering series-parallel combinations of transmission gates and transistors in the PU/PD paths. Fig. 5 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 Section V.

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 basic logic gates listed in Table II. Even though every transmission gate has two transistors, a topologically uniform comparison between CNTFET-based 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.

TABLE I
NUMBER OF GATES AND AVERAGE TRANSISTOR COUNT FOR DIFFERENT DESIGNS AND GATE STRUCTURES

Structure	Conventional Design		Generalized Design		
	Gates	Trans.	Gates (no sw.)	Gates (sw.)	Trans.
2	3	3.3	10	23	5.8
3	7	4.9	46	158	9.1

The structure of a gate designates the maximum number of pass-transistors or transmission gates it has in its PU or PD network.

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. 5. 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 F07: $\overline{(A \oplus B) \cdot (A \oplus C)}$ also implements $\overline{(\overline{A} \oplus B) \cdot (A \oplus C)}$, $\overline{(A \oplus B) \cdot (\overline{A} \oplus C)}$, and $\overline{(\overline{A} \oplus B) \cdot (\overline{A} \oplus C)}$. These four functions can be derived from F07 just by swapping the inputs $A \rightleftharpoons \overline{A}$, or equivalently, by swapping $B \rightleftharpoons \overline{B}$ and/or $C \rightleftharpoons \overline{C}$ accordingly. However, from the circuit implementation point of view, $\overline{(\overline{A} \oplus B) \cdot (A \oplus C)}$ and $\overline{(A \oplus B) \cdot (\overline{A} \oplus C)}$ are equivalent, given that they are derived from the same logic gate by swapping signals B and C. Then, the number of distinct gates that can be derived from the function of the gate F07 is 3 (instead of 4). The number of distinct gates obtained for every function by swapping polarities is included in Table II, and it sums up to 158 gates in total.

There are two types of cells in this *generalized family*: those that are formed exclusively by pass-transistors and those that contain transmission gates. The first subset of the cells covers the gates F00, F02, F03, F10, F11, F12, and F13, which can be fabricated in any unipolar technology, for instance with MOSFETs or MOSFET-like CNTFETs. We refer to this subset in this paper as the *conventional gates*. The second subset of logic cells embed one or more XOR functions in an efficient manner. They cannot be fabricated with a unipolar technology, and they are therefore referred to as *non-conventional gates*. We define the structure of a logic gate as follows: primary, secondary, and ternary gates as those having exactly one, two, and three transmission-gate(s) and/or pass-transistor(s) in their PU and PD networks, respectively.

The existence of derived gates, generated by swapping the signals and their complements, offers a high flexibility in designing the generalized gates, which is more important for larger gate structures. Table I summarizes the number of logic gates obtained for different gate structures for both the generalized and conventional families. We considered both cases with and without signal polarity swapping. The results in Table I show that the generalized family with a ternary structure has 22× more gates than its conventional counterpart when the derived gates through polarity swapping are taken into account. This fact confirms the high expressive power of the designed gates. The average number of transistors is also provided and it is about 75 to 85% larger for the generalized implementation in comparison to the conventional implementation.

IV. Alternate Ambipolar Logic Families

If transistor count and gate area are more critical than power consumption, then a pseudo-logic implementation of the same set of logic gates listed in Table II is preferred to the previously introduced transmission-gate static logic implementation. As for standard CMOS gate, pseudo-logic can be derived from static logic by replacing the PU network by a PU CNTFET biased as a p-type device. The PU CNTFET is weaker than the PD devices in order to allow the output signal to fall within the tolerated margin. Fig. 6 (bottom-left quadrant) depicts the pseudo-logic implementation of F05 combined with the transmission-gate approach. The rest of the gates summarized in Table II can be designed in transmission gate pseudo-logic in a similar way.

An alternative approach to reduce the transistor count is to replace all transmission-gates by pass-transistors. However, this requires that ambipolar CNTFETs that are electrically configured as n-type or p-type be located in the PU or PD network, respectively. They therefore conduct with a high resistance and cause the output level to be degraded. In order to restore the full swing of the output, a restoration stage (buffer or inverter) is inserted. This requires two more transistors and an additional gate delay, which will be assessed in Section V. Fig. 6 depicts the pass-transistor implementations of F05 as an example in static (top-right quadrant) and pseudo-logic (bottom-right quadrant), respectively. The other gates summarized in Table II can be designed in pass-transistor (static and pseudo) logic in a similar way.

V. Area and Delay of Logic Gates

In order to design libraries of ambipolar CNTFETs, we first validated the correctness of our design approach by simulating the designed gates. Ambipolar CNTs are an emerging technology, and simulating ambipolar CNTFET gates is not an easy task, given the fact that at present, there is no general and SPICE-compatible model for SB CNTFETs (i.e., ambipolar CNTFETs). The SB CNTFET model presented in [18] is restricted to a specific CNT chirality, is not SPICE-compatible, and does not allow for the in-field controllability of the ambipolarity. On the other hand, the MOSFET-like CNTFET model released in [19] is SPICE-compatible and offers more freedom with respect to the choice of the CNT parameters, such as CNT count and chirality. However, it does not allow for in-field control of device polarity.

TABLE II
Ambipolar CNTFET Logic Gates With No More Than Three Series Transmission-Gates or Transistors in Each PU/PD Network

Gate	Basic Function	Derived Functions
F00	$\overline{A}$	1
F01	$\overline{A \oplus B}$	2
F02	$\overline{A + B}$	1
F03	$\overline{A \cdot B}$	1
F04	$\overline{(A \oplus B) + C}$	2
F05	$\overline{(A \oplus B) \cdot C}$	2
F06	$\overline{(A \oplus B) + (A \oplus C)}$	3
F07	$\overline{(A \oplus B) \cdot (A \oplus C)}$	3
F08	$\overline{(A \oplus B) + (C \oplus D)}$	3
F09	$\overline{(A \oplus B) \cdot (C \oplus D)}$	3
F10	$\overline{A + B + C}$	1
F11	$\overline{(A + B) \cdot C}$	1
F12	$\overline{A + (B \cdot C)}$	1
F13	$\overline{A \cdot B \cdot C}$	1
F14	$\overline{(A \oplus D) + B + C}$	2
F15	$\overline{(A \oplus D) + (B \oplus D) + C}$	3
F16	$\overline{(A \oplus D) + (B \oplus D) + (C \oplus D)}$	4
F17	$\overline{((A \oplus D) + B) \cdot C}$	2
F18	$\overline{((A \oplus D) + (B \oplus D)) \cdot C}$	3
F19	$\overline{((A \oplus D) + B) \cdot (C \oplus D)}$	4
F20	$\overline{((A \oplus D) + (B \oplus D)) \cdot (C \oplus D)}$	6
F21	$\overline{(A + B) \cdot (C \oplus D)}$	2
F22	$\overline{(A \oplus D) + (B \cdot C)}$	2
F23	$\overline{A + (B \oplus D) \cdot C}$	2
F24	$\overline{(A \oplus D) + (B \oplus D) \cdot C}$	4
F25	$\overline{A + (B \oplus D) \cdot (C \oplus D)}$	3
F26	$\overline{(A \oplus D) + ((B \oplus D) \cdot (C \oplus D))}$	6
F27	$\overline{(A \oplus D) \cdot B \cdot C}$	2
F28	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot C}$	3
F29	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot (C \oplus D)}$	4
F30	$\overline{(A \oplus D) + (B \oplus E) + C}$	3
F31	$\overline{(A \oplus D) + (B \oplus D) + (C \oplus E)}$	8
F32	$\overline{((A \oplus D) + (B \oplus E)) \cdot C}$	3
F33	$\overline{((A \oplus D) + B) \cdot (C \oplus E)}$	4
F34	$\overline{((A \oplus D) + (B \oplus D)) \cdot (C \oplus E)}$	6
F35	$\overline{((A \oplus D) + (B \oplus E)) \cdot (C \oplus D)}$	8
F36	$\overline{(A \oplus D) + ((B \oplus E) \cdot C)}$	4
F37	$\overline{A + ((B \oplus D) \cdot (C \oplus E))}$	3
F38	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus E))}$	6
F39	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus D))}$	8
F40	$\overline{(A \oplus D) \cdot (B \oplus E) \cdot C}$	3
F41	$\overline{(A \oplus D) \cdot (B \oplus D) \cdot (C \oplus E)}$	6
F42	$\overline{(A \oplus D) + (B \oplus E) + (C \oplus F)}$	4
F43	$\overline{((A \oplus D) + (B \oplus E)) \cdot (C \oplus F)}$	6
F44	$\overline{(A \oplus D) + ((B \oplus E) \cdot (C \oplus F))}$	6
F45	$\overline{(A \oplus D) \cdot (B \oplus E) \cdot (C \oplus F)}$	4
Total		158

Ambipolar behavior can be *emulated* in a SPICE-environment by using the Stanford MOSFET-like CNTFET model. A method has been presented in [3]: every ambipolar CNTFET can be replaced by two parallel MOSFET-like CNTFETs with opposite polarities as depicted in Fig. 7. Note that this technique only emulates the ambipolar behavior, and it does not allow for any control of the ambipolarity, which has to be realized manually in the HSPICE simulations. We followed this approach to emulate ambipolar CNTFETs by using the SPICE-compatible Stanford model for MOSFET-like CNTFETs [19]. Then, in order to control the polarity, we

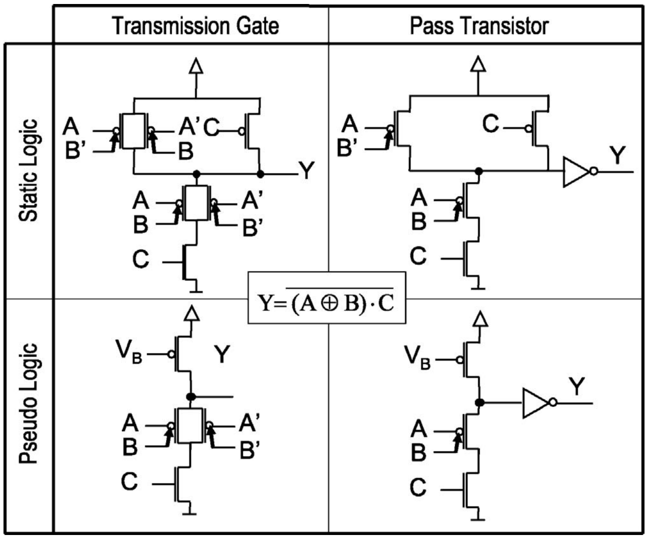
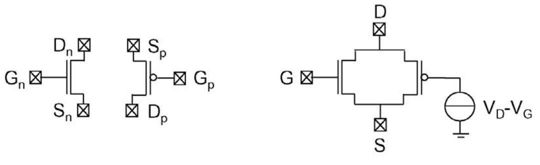


Fig. 6. Compact implementation of F05: $\overline{(A \oplus B) \cdot C}$. Any combination of transmission-gate/pass-transistor with static/pseudo-logic yields a possible design approach.



(a) (b)

Fig. 7. Emulation of an ambipolar CNTFET using two MOSFET-like CNTFETs in parallel with opposite polarities. (a) n-type and p-type MOSFET-like CNTFETs. (b) Their parallel connection.

manually turned parallel transistors on and off in the HSPICE deck according to the value of the polarity control signals.

In this section, we compare different ambipolar CNTFET logic design families and CMOS. Since the static transmission-gate ambipolar CNTFET family requires both polarities of some inputs, it is possible to consider a dual-rail CMOS logic family for this purpose. Different logic families have been considered including single-rail pass-transistor logic, dual-rail complementary pass-transistor logic, and double pass-transistor logic [20]. These logic families are based on pass-transistors and require a signal inversion/restoration at the output in order to restore the signal swing. The proposed transmission-gate ambipolar CNT family does not require any pass-transistors and guarantees full swing in a manner similar to complementary static CMOS design. We therefore chose to compare the transmission-gate static CNT family with the static CMOS family.

A. Transistor Sizing

We designed the logic gates with equal rise and fall times, and the output current is equal to that of the unit inverter. Since electron and hole mobility is equal in CNTs, the on-resistance of p-type 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.

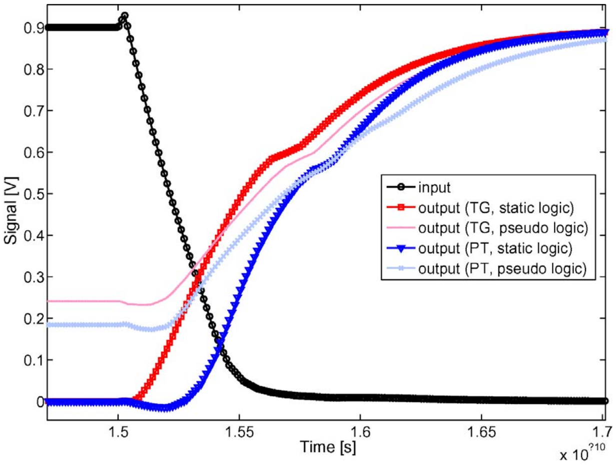


Fig. 8. Waveforms for F01: $Y = \overline{A \oplus B}$ when $B = 0$ and A falls from 1 to 0. All possible combinations of transmission gates (TG)/pass-transistors (TG) with static/pseudo-logic are depicted.

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 [21]. Hence, the resistance of a transmission gate is estimated as $R_{\mathrm{n}} \parallel 2R_{\mathrm{p}}$ if it conducts a low signal, and $2R_{\mathrm{n}} \parallel R_{\mathrm{p}}$ if it conducts a high signal. Since $R = R_{\mathrm{n}} = R_{\mathrm{p}}$ 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.

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 ($2A$ versus $4A/3$). 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× weaker than the PD network, which offers a good compromise between delay and area.

We simulated the dynamic behavior of the designed logic gates using HSPICE. In Fig. 8, we depicted the delay of the XNOR gate F01 for $B = 0$ when A falls from 1 to 0. The delay is illustrated for all four design approaches explained previously. Note that for both pass-transistor implementations, the signal A is not pulled completely down to $V_{\mathrm{SS}} = 0$ V.

B. Library Characterization

Fig. 9 summarizes the weighted device count and FO4 delay estimates for the library cells. Note that the additional gates

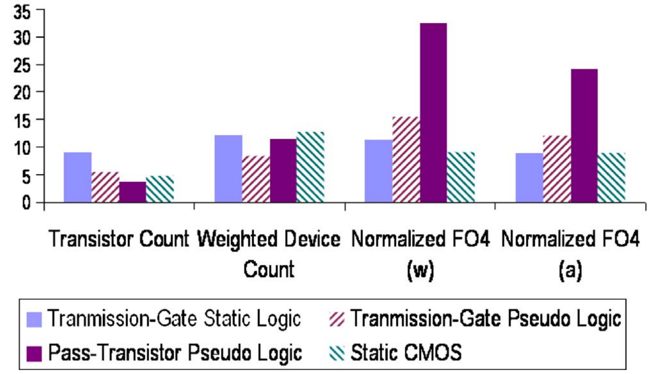


Fig. 9. Characterization of the designed CNTFET library compared to CMOS *with the same topology*: transistor count, weighted device count, normalized FO4 delay to the technology-dependent delay in the worst case (w) and on average (a).

obtained by swapping the signal polarities at the transmission gates (Section III) 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 three transistors in the PU and PD networks, respectively. The weighted device count was obtained by summing over the number of devices weighted with the respective ratio of their area to the one of a unit transistor in the PD network of an inverter with the strength set to 1. This metric takes into account only the impact of logic devices on area, without any consideration of the physical layout and the signal routing. The FO4 delay was calculated with the switch-level RC delay model [21] and is equal to the delay of a gate driving four 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 [21]. 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) in order to decouple the impact of technology from the design. Note that the intrinsic delay of CNTFETs is roughly 5.1× less than CMOS [1] for a variability-free CNT technology.

We observed 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-type 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 weighted device count is slightly smaller (12.3 versus 12.7) than the CMOS library, because most of the transmission-gate transistors are sized smaller since they are in parallel.

As expected, the CNTFET transmission gate pseudo-logic family has a 31% smaller average weighted device count than its static counterpart (8.5 versus 12.3); however, it is 33% slower (12 versus 9). Surprisingly, the CNTFET pass transistor pseudo-logic family is less area efficient (in terms of weighted device count) than its transmission gate counter-part. This confirms the conjecture in Section IV 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 weighted device count: 11.5 versus 12.3), while it is 2.7× slower (delay: 24.1 versus 9). This makes the CNTFET pass transistor family a sub-optimal 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. This increased the average delay and weighted device count of all logic families by 5–17%.

VI. Area and Delay of Large Logic Circuits

We used the tool ABC developed at Berkeley [22] 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 the characterized libraries. Note that the weighted device count was used as an estimate for area. The results for 15 benchmark circuits are summarized in Table III. Arithmetic circuits in the considered benchmarks are usually not derived directly from logic synthesis. However, this is only a small subset of the circuits that were considered. We included logic-only circuits, as well as established benchmark circuits like C2670 and C3540 that are a mixture of ALU and control. The synthesis and technology mapping was performed using the same flow across all benchmarks, and we have been careful to differentiate between the functions of the benchmarks.

In Section V, we showed that the transmission gate configuration outperforms the pass transistor configuration in terms of weighted device count and delay. We therefore considered only transmission gate implementations in static and pseudo-logic and we compared them to the CMOS library. For each family, the number of gates, the weighted device count, 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 (measured as weighted device count). Of the benchmarks, circuits that embed XOR operations, the adders, ALUs, error correcting circuits, and the multiplier C6288, return the best improvements in weighted device count and speed when implemented in CNTFET technology (see Fig. 10).

The implementation with both transmission gate CNT families requires on average ≈38% fewer gates and 40% less levels

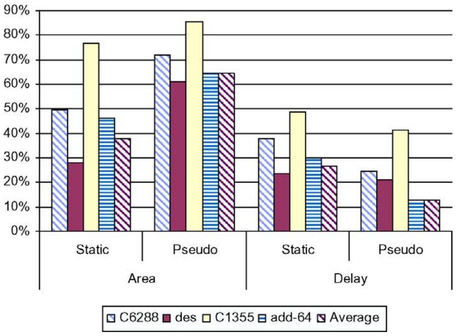


Fig. 10. Saving in terms of area (measured as weighted device count) and delay of circuits mapped with transmission-gate CNTFET libraries versus those mapped with MOSFET gates.

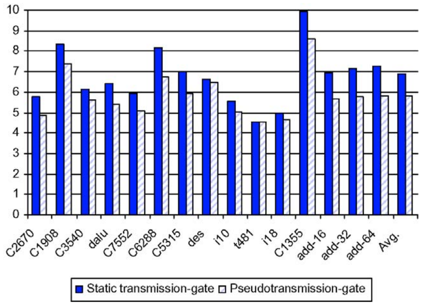


Fig. 11. Ratio of the absolute delay of CMOS implementation to CNTFET implementation.

of logic than CMOS. While the static logic CNTFET family saves 37.7% in terms of weighted device count on average compared to CMOS, the pseudo-logic CNTFET family saves 64.5% in terms of weighted device count on average.

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 (see Fig. 10). Delay was normalized to the technology-dependent intrinsic delay τ, and unipolar CNTFETs are expected to be 5.1× faster than CMOS [1]. We assumed the same intrinsic delay for unipolar and ambipolar CNTFETs to calculate the absolute delay of the logic circuits. Fig. 11 shows the cumulative benefits of technology and design that translate into an average speedup of 6.9× and 5.8× for static and pseudo CNTFET logic families, respectively, compared to CMOS. The largest speedup was calculated for the static CNTFET implementation of multipliers (*approx* 10×) and error correcting circuits (more than 8×). 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.

VII. Estimating Power Dissipation

The designed libraries of logic gates outperform CMOS in terms of expressive power. Technology mapping with the transmission-gate static or pseudo-logic families yields smaller circuits and fast designs. The purpose of this section is to present a simulation technique that allows us to assess the power consumption of the mapped circuits with ambipolar CNTFETs. We focus on the transmission-gate static logic CNTFET family for the rest of this paper, because it is the most promising family in terms of power consumption.

Despite the advantages of the expressive power of the transmission-gate static CNTFET family, an important fact is that the frequent utilization of embedded XOR functions may increase the dynamic power dissipation because of the high activity factor of these functions. The activity factor is defined as the number of times a gate switches from 0 to 1 and from 1 to 0 on average, when all its input combinations are applied. For 2-input NOR and NAND gates, only one input combination among the existing four changes the signal direction; then their activity factor is 25%. On the other hand, for 2-input XOR gates, the activity factor is 50%. Moreover, even when the embedded XOR gates are not switching, their static power is expected to be high, since they are formed by transmission gates whose leakage is twice as high as the static leakage of a single transistor with the same size.

A. Model of Power Dissipation

In order to study the power dissipation of static logic gates in ambipolar CNTFET technology, we consider the different components of power dissipation reported in static logic gates in CMOS technology [23]. The total power dissipation of a logic gate is modeled as follows:

$$P_{\mathrm{T}} = P_{\mathrm{D}} + P_{\mathrm{SC}} + P_{\mathrm{S}} + P_{\mathrm{G}} \tag{1}$$

where P_{D} denotes the dynamic power, P_{SC} the short-circuit power, P_{S} the static power, and P_{G} the power dissipation due to gate leakage. Dynamic power is dissipated whenever the gate switches from 0 to 1 and from 1 to 0 in order to charge or discharge the load capacitance. In this paper, we do not consider the dynamic power dissipated by the interconnect. Short-circuit power is dissipated during the switching phase when devices in both PU and PD networks are temporarily and simultaneously conducting current from V_{DD} to V_{SS}. Static power is dissipated when the gate is idle due to the subthreshold leakage. The power dissipation due to gate leakage is caused by the tunneling current through the gate oxide. The different components of the total power can be estimated as follows [23], [24]:

$$P_{\mathrm{D}} = \alpha \cdot C \cdot f \cdot V_{\mathrm{DD}}^2 \tag{2}$$

$$P_{\mathrm{SC}} \approx 0.15 \cdot P_{\mathrm{D}} \tag{3}$$

$$P_{\mathrm{S}} = I_{\mathrm{off}} \cdot V_{\mathrm{DD}} \tag{4}$$

$$P_{\mathrm{G}} = I_{\mathrm{g}} \cdot V_{\mathrm{DD}} \tag{5}$$

where α denotes the activity factor, C the load capacitance, f the operating frequency, V_{DD} the power supply, I_{off} the sum of all subthreshold currents, and I_{g} the sum of all gate leakage currents. The conjecture $P_{\mathrm{SC}} \approx 0.15 \cdot P_{\mathrm{D}}$ has been verified for CMOS technology [23] and is also assumed to be valid for CNTFETs.

TABLE III

Results for Technology Mapping: Gate Count, Weighted Device Count, Logic Depth, Normalized 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	
			Gate	Device	Levels	Norm.	Abs.	Gate	Device	Levels	Norm.	Abs.	Gate	Device	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 versus 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]			τ_1 = 3 ps					τ_1 = 3 ps					τ_2 = 15 ps				

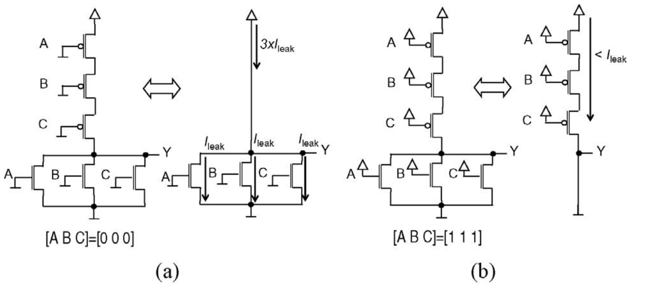


Fig. 12. Example of leakage. (a) High leakage through parallel transistors. (b) Lower leakage through series transistors.

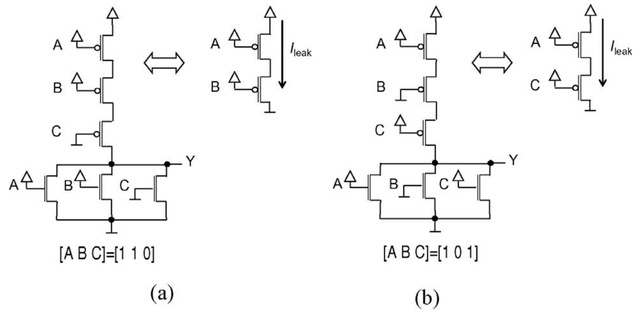


Fig. 13. Identical I_{off} patterns for different input vectors. (a) Input vector: [1 1 0]. (b) Input vector: [1 0 1].

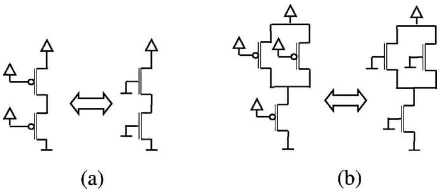


Fig. 14. Equivalence of leakage in PU and PD networks. (a) Leakage in a series connection for NAND/NOR gates. (b) Leakage in a parallel-series connection for AOI/OAI gates.

Generally, f and V_{DD} are fixed for a given process and design, C is given by the process and geometry, and α is statistically estimated for a given circuit and application. This gives analytical expressions for P_{D} and P_{CS}. However, the static leakage currents I_{off} and I_{g} do not have any analytical expression for CNTFET technology, and they strongly depend on the input vector. We therefore deploy a method that uses the SPICE model of CNTFETs in an efficient way by classifying the patterns generated by the input vectors in order to estimate P_{S} and P_{G}.

B. Pattern-Based Power Model

In order to estimate the static power, we need to consider all input vectors that strongly impact the static power. For example, given a 3-input NOR gate, depending on the input vector, we may have an increase of static power by a factor of more than 3× if we compare leaking parallel transistors (input [0 0 0]) to those that are in series (input [1 1 1]), as depicted in Fig. 12.

The number of input vectors increases exponentially with the number of inputs. We can avoid running a large number of simulations to quantify I_{off} for every input pattern by using the I_{off} pattern classification method [24]. This method is based on identifying the pattern of on-transistors and off-transistors for every given input vector. Then, the on-transistors are considered to have a negligible resistance and just replaced by a short circuit in the pattern. Also off-transistors that are shorted by parallel on-transistors are removed from the pattern. For instance, a 3-input NOR gate with the input vectors [1 1 0] and [1 0 1] generates the same I_{off} pattern (Fig. 13).

We used the same assumption in [24], which states that similar patterns in the PU and PD networks generate equal leakage currents and can be considered equivalent (Fig. 14). This is justified by the design of symmetrical (non-skewed) gates, i.e., having equal drive current for both PU and PD networks. Once an I_{off} pattern is mapped onto every input vector, only the set of different I_{off} patterns has to be quantified.

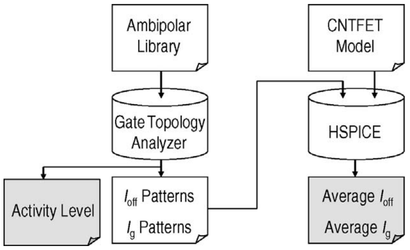


Fig. 15. Simulation flow.

Note that the gate leakage I_{g} is also a static current that occurs in the on-network while the off-current I_{off} is measured in the off-network. Consequently, it also depends on the input vector and it can be assessed by using the same topology analyzer.

C. Simulation Flow

The library characterization for power dissipation was carried out in two steps (Fig. 15). First, we performed the mapping between the I_{off}-current (I_{g}-current) patterns and the input vectors for every logic gate in the library by determining the topology of the logic gate given the input vector, to obtain a netlist of off-transistors. This *gate topology analyzer* also calculates the activity factor of every logic gate. Then, we performed circuit level simulations in order to determine the exact values of I_{off} and I_{g} by characterizing every pattern. Thus, for every logic gate, we obtained a vector of I_{off} and I_{g} values for every input vector, which were averaged and used to estimate the static power dissipation. This flow is depicted in Fig. 15.

VIII. Power Consumption of Logic Gates

We considered the static ambipolar CNTFET transmission-gate library described in Table II in Section III. We classified all I_{off} patterns obtained for the whole library, and obtained 26 different patterns shown in Fig. 16.

The load capacitance depends on the intrinsic drain capacitance and on the gate fan-out, assumed to be equal to 3. We assumed identical values for unit gate, drain, and source capacitances, as well as a 32 nm gate width and 3 CNTs per channel. Based on these assumptions, the unit capacitances can be derived from [1]. In order to compare the power dissipation of CNTFET logic gates with those in CMOS technology, we also characterized the logic gates taken from the considered library that are available in CMOS technology. Leakage currents I_{off} and I_{g} for a unit transistor as well as unit capacitances were estimated using the MASTAR simulator provided by the international technology roadmap for semiconductors [25]. In these simulations, we assumed the built-in model for 32 nm bulk technology with metal gate and strained channel. For both CNTFET and CMOS logic gates, we set the power supply and operation frequency to 0.9-V and 1 GHz, respectively.

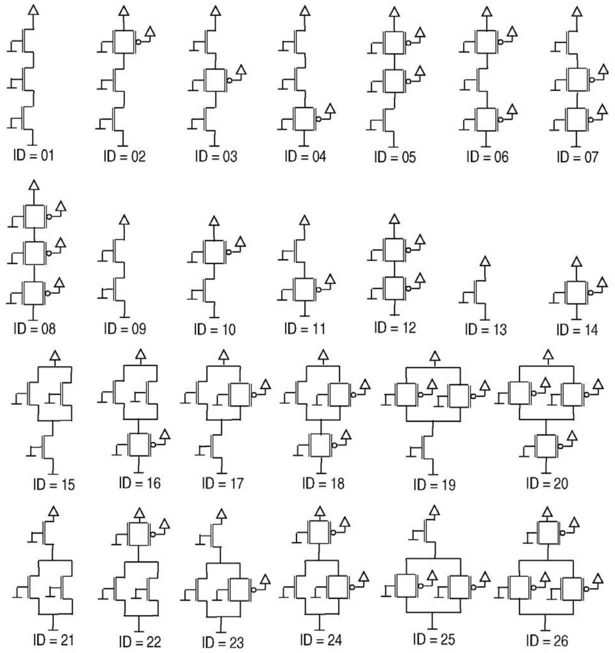


Fig. 16. List of 26 I_{off}-current patterns in the static ambipolar transmission-gate library.

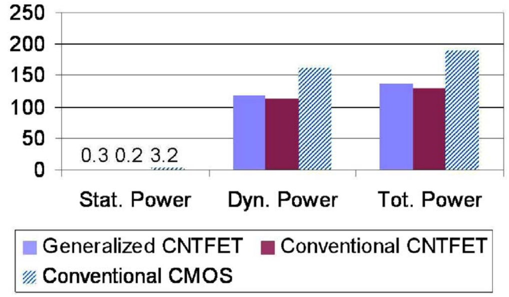


Fig. 17. Average power dissipation in the generalized CNTFET, conventional CNTFET, and conventional CMOS libraries.

Fig. 17 summarizes the characterization of the generalized CNTFET, conventional CNTFET, and standard CMOS libraries for power dissipation. Recall that the conventional gates are those that can be fabricated in a unipolar technology, for instance with MOSFET or MOSFET-like CNTFET devices. These results highlight the dynamic, static, and total power for both libraries. Short circuit power was assumed to be 15% of P_{D} [23]. Power dissipated as gate leakage was found to be about 10% of P_{S} for CMOS gates and less than 1% of P_{S} for CNTFET because of the high-κ dielectric used as gate insulator in CNTFETs [19].

The activity factor is a key factor that determines dynamic power consumption. We found that the CNTFET library shows on average the same activity factor as the CMOS library, despite the frequent presence of XOR functions. The XOR function has a higher activity factor when it is used as a stand-alone gate. However, when the XOR function is embedded in complex generalized gates as described in this paper, there is a negligible increase in the overall activity factor. In other words, the more complex the binate function, the lower is the activity factor. For instance, F01 (XOR2) has only two inputs and a high activity factor of 50%. However, F14 and F15, which are more complex, have an activity factor of only 12.5%.

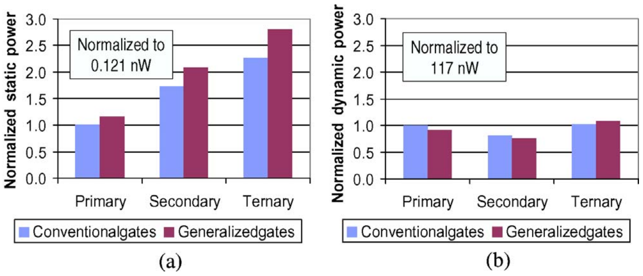


Fig. 18. Power dissipation versus CNTFET gate structure (number of transmission-gates and/or pass-transistors in the PU/PD networks). (a) Static power. (b) Dynamic power.

The CNTFET gates dissipate on average 27% less dynamic power than CMOS gates, which is mainly due to the lower CNTFET input capacitance, given the equal activity factors. As a matter of fact, under the given assumptions, the input capacitance of a CNTFET inverter is 36 aF, while it is 52 aF for CMOS inverters (31% difference). Static power of CNTFET gates is about one order of magnitude less than CMOS gates, because of the use of a thick insulator separating drain/source from the substrate of CNTFETs. On average, the CNTFET gates dissipate 28% less power than CMOS gates.

We also compared the power consumption of conventional gates with the power consumption of the whole generalized gates for different gate structures. Recall that the structure of a logic gate corresponds to the maximum expected number of transmission-gates and/or pass-transistors in its PU or PD network. Fig. 18(a) depicts the normalized P_S (to the static dissipation of an inverter) versus the structure of conventional and generalized gates. The change is linear with the gate structure because of the linear increase of number of leaking transistors in series in binary and ternary structures. Moreover, the generalized library dissipates on average 22% more static power than conventional gates, because of the utilization of transmission gates. The normalized dynamic power (to the dynamic power of an inverter) versus gate structure, depicted in Fig. 18(b), shows a different trend: P_D hardly increases with gate complexity, because it depends on the drive current that is equal to that of a unit inverter for all gates in this library.

IX. Power Consumption of Large Logic Circuits

Based on the obtained netlists for the previously synthesized logic circuits (see Section VI), power consumption and energy-delay-products were estimated by injecting 640K random input patterns, which were used to determine the circuit activity factor. The results for 12 benchmark circuits are summarized in Table IV. We considered CNTFET technology using both generalized and conventional gates in transmission-gate static logic and CMOS technology in static logic.

The CNTFET library using generalized gates returns the best technology mapping with more than 24% saving in terms of number of logic gates on average, given its higher expressive power. Both CNTFET technology with conventional gates and CMOS technology need approximately the same physical resources on average, because they implement the same set of gates. The conventional CNTFET technology library requires about 3% less gates due to the fact that the mapping is not fully identical. This is because some CNTFET gates are more compact or have less parasitics than their CMOS counterpart, since the p-type CNTFET can be sized equally to a n-type CNTFET in order to drive the same current (which is not valid for CMOS). This reduces the parasitics and intrinsic delays, which is reflected in the synthesized netlists. The compact design with the generalized CNTFET library is on average 25% faster than the conventional CNTFET library, and 7× faster than CMOS designs, because the intrinsic CNTFET delay is 5× lower than the MOSFET delay [1]. Circuits that embed XOR operations (multiplier, and error correcting circuits) require the fewest gates and can be mapped with the lowest delay when the generalized CNTFET library is used.

Static power is about two orders of magnitude less than dynamic power for both types of CNTFET families and one order of magnitude less for the CMOS family. This confirms the trend observed on the library characterized in Section VIII, and it is mainly due to the better isolation of CNTFETs in the off-state. The generalized CNTFET library is on average 28% more power-efficient than the conventional CNTFET library. The highest power saving was found for the multiplier C6288 and the error correcting circuits. The same trend can be seen when circuits mapped with the generalized CNTFET gates are compared with those mapped with CMOS gates, showing an average power saving of 55%.

The generalized CNTFET library outperforms the conventional CNTFET library in terms of EDP by 43% on average. The lowest EDP is found when circuits embed the XOR operation frequently (C1908, C6288, and C1355), because their delay and power consumption are lower with the generalized CNTFET implementation. The EDP of CMOS-based circuits is much larger than for circuits mapped with either CNTFET families. While the EDP of conventional CNTFET gates is expected to be 13× lower than for CMOS gates [1], the simulated EDP of circuits mapped with generalized CNTFET gates is on average 20× lower than CMOS circuits, resulting from the cumulative benefits of the proposed design technique and the technology boosters of CNT technology.

Finally, we would like to highlight some limitations of the proposed behavioral model. For instance I_{off} is not simulated accurately because the model is not physical. Consequently, actual static power may be larger than the simulated values. Dynamic power is however large enough to remain the dominant component of power consumption. On the other hand, there may be race conditions between the polarity and conventional gates that lead to unintended short-circuit currents. To assess the impact of this additional short-circuit power, a physical SPICE-compatible model for ambipolar CNTFETs may be used in the future.

X. Summary: Ambipolar Versus Unipolar Design

This section has two purposes. On the one hand, it summarizes the results detailed in the previous sections. On the other hand, it discusses the results in a normalized manner that is not specific to CNT technology. This will highlight

TABLE IV

LOGIC SYNTHESIS AND TECHNOLOGY MAPPING: GATE COUNT, DELAY (ps), P_D (μW), P_S (μW), P_T (μW) AND ENERGY-DELAY-PRODUCT (10^{-24} J · s), SIMULATED AT f = 1 GHz AND V_{DD} = 0.9 V

Benchmark		CNTFET Technology (Generalized Gates)						CNTFET Technology (Conventional Gates)						CMOS Technology					
Circuit	Function	No.	Delay	P_D	P_S	P_T	EDP	No.	Delay	P_D	P_S	P_T	EDP	No.	Delay	P_D	P_S	P_T	EDP
C2670	ALU and control	541	52	10.95	0.10	12.70	0.66	631	62	14.52	0.14	16.83	1.04	632	320	20.34	1.84	25.42	8.13
C1908	Error correcting	261	50	4.23	0.05	4.91	0.25	569	90	11.34	0.13	13.17	1.19	544	452	15.81	1.63	19.98	9.04
C3540	ALU and control	871	80	17.35	0.18	20.13	1.61	1126	109	24.06	0.26	27.93	3.04	1084	551	32.24	3.29	40.70	22.41
dalu	Dedicated ALU	892	68	13.29	0.19	15.48	1.06	1142	79	17.24	0.26	20.08	1.59	1046	401	22.38	3.20	29.26	11.73
C7552	ALU and control	1229	59	24.68	0.24	28.62	1.69	1722	77	40.74	0.38	47.23	3.65	1615	401	55.45	4.85	69.10	27.71
C6288	Multiplier	1645	161	31.53	0.31	36.57	5.88	3405	245	79.40	0.78	92.09	22.57	3653	1268	114.20	11.09	143.53	181.96
C5315	ALU and selector	1163	58	23.69	0.24	27.47	1.59	1368	88	31.96	0.31	37.06	3.28	1496	448	48.53	4.41	60.66	27.20
des	Data encryption	3429	40	59.02	0.72	68.59	2.75	3483	59	64.71	0.78	75.19	4.41	3668	301	98.34	11.26	125.48	37.82
i10	Logic	1680	82	23.37	0.34	27.21	2.24	1979	95	31.29	0.43	36.41	3.47	2073	486	45.90	6.00	59.39	28.88
t481	Logic	860	54	6.92	0.19	8.15	0.44	709	58	5.08	0.15	6.00	0.35	743	290	7.73	2.24	11.36	3.30
i8	Logic	961	37	19.72	0.21	22.89	0.86	987	37	19.98	0.22	23.19	0.87	974	191	29.06	2.93	36.65	7.00
C1355	Error correcting	212	27	3.34	0.04	3.88	0.10	428	62	10.73	0.10	12.43	0.78	607	320	18.16	1.83	22.89	7.33
Average		1145	64	19.84	0.23	23.05	1.59	1462	89	29.25	0.33	33.97	3.85	1511	452	42.35	4.55	53.70	31.04
Improvement versus CMOS		24.2%	7.1×	53.4%	94.5%	57.1%	19.5×	3.2%	5.1×	30.9%	92.7%	36.7%	8.1×	–	–	–	–	–	–
Delay normalization		τ_1 = 3 ps						τ_1 = 3 ps						τ_2 = 15 ps					

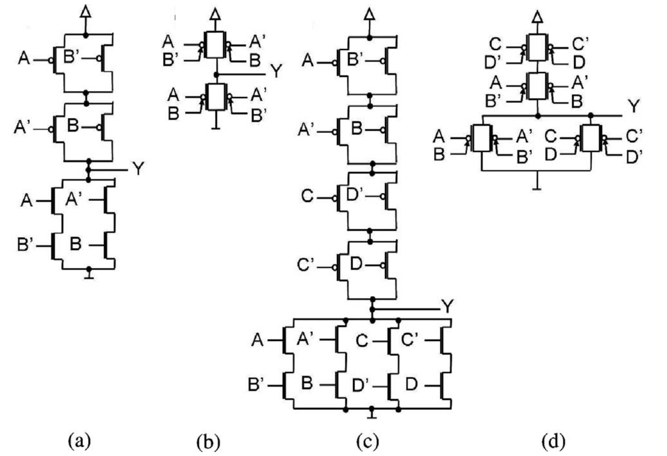


Fig. 19. Comparison of unipolar versus ambipolar implementation of binate functions. (a) Unipolar F01. (b) Ambipolar F01. (c) Unipolar F07. (d) Ambipolar F07.

the benefits of ambipolar design over unipolar design, which can be applied to other ambipolar technologies such as silicon nanowires [4]. Recall that an ambipolar device is a device that has both n-type and p-type polarities, and that we implicitly only consider ambipolarity that is controllable in the field through a second gate, i.e., we can control in the field whether a transistor operates as an n-type or p-type device.

A. Design with Ambipolar Logic Gates

We demonstrated in this paper that the main benefits of ambipolarity are the design of compact XOR/XNOR functions that are embedded into NAND, NOR, AOI, and OAI structures. This necessitates the presence of both polarities of signals to be XOR-ed. Fig. 19 compares the implementation of F01 ($Y = \overline{A \oplus B}$) and F07 ($Y = \overline{(A \oplus B) + (C \oplus D)}$) in the unipolar and the proposed ambipolar design styles. In both cases, it is assumed that both polarities of signals are available.

The unipolar design is similar to standard CMOS design. By using both polarities of inputs in a unipolar design, more binate functions (e.g., XNOR) than usual CMOS with single polarity inputs can be implemented. However, the internal structure of such an XNOR gate is larger compared to the ambipolar design [Fig. 19(a) versus Fig. 19(b)]. For more complex gates, e.g., F07, the unipolar implementation using both signal polarities requires a large number of transistors in series, as depicted in Fig. 19(c), which is not desirable in scaled technologies. The complexity of the cell makes it bulky and slow. Consequently, when it comes to synthesis of large multi-level logic circuits, the logic synthesizer has the tendency to avoid the utilization of such unipolar gates. However it utilizes the equivalent ambipolar gates, because they are smaller and faster.

It is worth highlighting that the unipolar CNTFET implementation used for comparison is reminiscent of static CMOS design. Other logic families can be used for comparison, such as those based on pass-transistor or transmission-gate implementations [20]. Recall that the pass-transistor and transmission-gate implementations in the sense of MOSFET-based design are different from the pass-transistor and transmission-gate implementation in the sense of CNTFET-based design as introduced in Section III. In pass-transistor (transmission-gate) logic using MOSFETs, input signals are connected to source/drain terminals of transistors instead of their gate terminals, while in CMOS logic style, all input signals are connected to the gate terminals [Fig. 20(a) and (b)]. In the proposed ambipolar CNTFET static library (using either pass-transistors or transmission-gates), all input signals are connected to the gate terminals of the transistors [Fig. 20(c)] in a similar way to CMOS design. This motivated the comparison of the ambipolar implementation with the CMOS-style family.

B. Multi-Level Logic Synthesis with Ambipolar Logic Gates

Logic synthesis and technology mapping were performed with the ABC tool and based on two libraries, the ambipolar CNTFET and the standard CMOS library. In order to allow for a technology-independent comparison of ambipolar and unipolar design approaches, we looked into the weighted device count, logic depth, and normalized delay of the critical path for logic circuits. We remind the reader that the weighted device count was obtained by summing the number of devices weighted with the respective ratio of their area to the one of a unit transistor in the PD network of an inverter with the

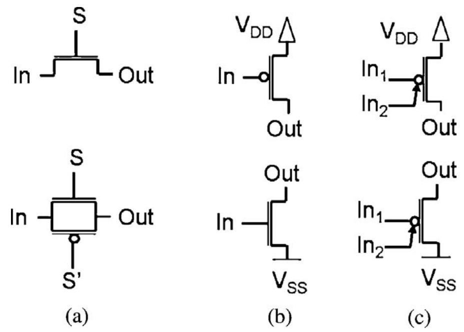


Fig. 20. Input signal configuration in different logic family styles. (a) Pass-transistor (top) and transmission-gate (bottom). (b) CMOS in pull-up (top) and pull-down (bottom). (c) Ambipolar design in pull-up (top) and pull-down (bottom).

strength set to 1. The delay was normalized to the intrinsic delay τ of a single device, which is given by the technology.

For power consumption we compared the total power of circuits synthesized with the ambipolar CNTFET family (using all generalized NAND/NOR/AOI/OAI structures) and the unipolar CNTFET family (using conventional NAND/NOR/AOI/OAI structures). These results showcase the benefits of the ambipolar design, but they are specific to CNT technology under the assumption of no variability, as modeled by Stanford in [19].

From the results detailed in Section VI we highlight that the ambipolar implementation requires about 42% less levels of logic than the unipolar implementation for the set of circuits that were considered in this paper. The weighted device count and the number of logic gates used to map the circuits were also about 38% less for the ambipolar design. On average, the ambipolar design saved about 26% of the normalized delay compared to the standard unipolar design.

The results explained in Section IX (Table IV) show that the ambipolar design dissipates less power than the unipolar design, which is consistent with the fact that it requires less physical resources: 32% saving in total power and 59% saving in EDP. These figures are specific to CNT technology.

C. Future Directions

The previous figures give an insight into the impact of ambipolar design on the circuit-level benefits. Here, we motivate directions for future research that will provide a more in-depth assessment of the benefits of the ambipolar design methodology.

First, this paper has been carried out at the logic synthesis and technology mapping level to validate the usefulness of the proposed ambipolar design approach. In order to assess the impact of signal routing on area, delay and power, placement and routing are required. The previously assessed weighted device count is just an indicator of area saving coming from logic. The actual area will depend on cost of routing. The additional parasitic capacitances coming from routing will impact the figures describing the normalized delay, power, and EDP. With current technology, those parasitics are still large because of the large contacts [26]. However, the reductions in logic depth reported in this paper are independent of these physical considerations.

Next, the CNTFET model used in this paper describes ideal CNTs, i.e., the variability of state-of-the-art CNTs was not included. This variability affects the number and spacing of transistors in every device, their nature (semiconducting/metallic), their chirality, and the quality of the contact to the metal line. All these physical sources of variability have an impact on the drive strength of single devices, their threshold voltage, intrinsic delay, and power dissipation. It is therefore necessary to include redundancy and fault tolerance techniques if state-of-the-art technology is utilized in chip fabrication.

Finally, it is also necessary to address the challenge of the voltage range of the polarity gate. In [5], a difference in the voltage range between the control and the polarity gate was reported for the fabricated ambipolar devices. In the proposed approach, it is important to have both voltage ranges identical, otherwise voltage shifters would be required. Fixing the voltage range is a matter of technology: voltages can be tuned by choosing metals and insulators that result in the appropriate work functions. In [27], a physical model of ambipolar devices including the interaction between both gates and the impact of technology parameters has been proposed. This opens up the opportunity to optimize the used materials according to the desired voltage range.

XI. Conclusion

In this paper, we described novel design guidelines for logic gates based on ambipolar devices such as CNTFETs. Several new logic functions including one or more embedded XOR operations can be implemented efficiently in the ambipolar library. The novel design techniques are based on a combination of transmission-gate or pass-transistor with static or pseudo-logic. Different tradeoffs between the four possible design styles were analyzed. The designed library was utilized to map synthesized multi-level logic circuits, including adders, multipliers, and ALUs. The transmission-gate static logic provides the most attractive approach for ambipolar circuit design, resulting in 38% lower weighted device count on average in comparison to circuits mapped with unipolar gates. The ambipolar library outperforms the unipolar library by reducing the number of logic levels by 42%, the delay by 26%, and the power consumption by 32%. Based on the predictions of the performance of defect-free CNTFETs versus MOSFETs in [1], the combination of the ambipolar design methodology with the CNT technology results, on average, in 7× lower delay, 57% less power consumption, and 20× less EDP in comparison to circuits mapped in CMOS technology.

Acknowledgment

The authors would like to thank Prof. S. Mitra for helpful discussions, M. D. Marchi for part of our logic simulations, and M. Choudhury for the power simulator.

References

[1] J. Deng, N. Patil, K. Ryu, A. Badmaev, C. Zhou, S. Mitra, and H.-S. P. Wong, "Carbon nanotube transistor circuits: Circuit-level performance benchmarking and design options for living with imperfections," in *Proc. IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 70–588.

[2] M. H. Ben-Jamaa, D. Atienza, Y. Leblebici, and G. D. Micheli, "Programmable logic circuits based on ambipolar CNFET," in *Proc. DAC*, 2008, pp. 339–340.

[3] I. O'Connor, L. Junchen, F. Gaffiot, F. Pregaldiny, C. Lallement, C. Maneux, J. Goguet, S. Fregonese, T. Zimmer, L. Anghel, T.-T. Dang, and R. Leveugle, "CNTFET modeling and reconfigurable logic-circuit design," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 54, no. 11, pp. 2365–2379, Nov. 2007.

[4] A. Colli, S. Pisana, A. Fasoli, J. Robertson, and A. C. Ferrari, "Electronic transport in ambipolar silicon nanowires," *Physica Status Solidi (B)*, vol. 244, no. 11, pp. 4161–4164, 2007.

[5] Y.-M. Lin, J. Appenzeller, and P. Avouris, "Novel carbon nanotube FET design with tunable polarity," in *Proc. IEEE IEDM Tech. Dig.*, 2004, pp. 687–690.

[6] A. K. Geim and K. S. Novoselov, "The rise of graphene," *Nature Mater.*, vol. 6, no. 3, pp. 183–191, 2007.

[7] R. Murgai, R. K. Brayton, and A. Sangiovanni-Vincentelli, *Logic Synthesis for Field-Programmable Gate Arrays*. Norwell, MA: Kluwer, 1995.

[8] T. Sasao, *Switching Theory for Logic Synthesis*. Norwell, MA: Kluwer, 1999.

[9] M. Ben Jamaa, K. Mohanram, and G. De Micheli, "Novel library of logic gates with ambipolar CNTFETs: Opportunities for multi-level logic synthesis," in *Proc. DATE Conf. Exhibit.*, Apr. 2009, pp. 622–627.

[10] A. Colli, A. Tahraoui, A. Fasoli, J. M. Kivioja, W. I. Milne, and A. C. Ferrari, "Top-gated silicon nanowire transistors in a single fabrication step," *ACS Nano*, vol. 3, no. 6, pp. 1587–1593, 2009.

[11] M. Choudhury, Y. Yoon, J. Guo, and K. Mohanram, "Technology exploration for graphene nanoribbon FETs," in *Proc. DAC*, 2008, pp. 272–277.

[12] Y.-M. Lin, J. Appenzeller, J. Knoch, and P. Avouris, "High-performance carbon nanotube field-effect transistor with tunable polarities," *IEEE Trans. Nanotechnol.*, vol. 4, no. 5, pp. 481–489, Sep. 2005.

[13] J. Liu, I. O'Connor, D. Navarro, and F. Gaffiot, "Novel CNTFET-based reconfigurable logic gate design," in *Proc. Annu. ACM IEEE Des. Autom. Conf.*, 2007, pp. 276–277.

[14] J. Liu, I. O'Connor, D. Navarro, and F. Gaffiot, "Design of a novel CNTFET-based reconfigurable logic gate," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2007, pp. 285–290.

[15] I. O'Connor, J. Liu, D. Navarro, and F. Gaffiot, "Dynamically reconfigurable logic gate cells and matrices using CNTFETs," in *Proc. Int. Conf. Des. Technol. Integr. Syst. Nanoscale Era*, Mar. 2008, pp. 1–6.

[16] F. Mo and R. K. Brayton, "Whirlpool PLAs: A regular logic structure and their synthesis," in *Proc. Int. Conf. Comput.-Aided Design*, 2002, pp. 543–550.

[17] T. Sasao, "EXMIN2: A simplification algorithm for exclusive-OR-sum-of-products expressions for multiple-valued-input two-valued-output functions," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 12, no. 5, pp. 621–632, May 1993.

[18] *Stanford University Ambipolar CNFET Model*. (2008) [Online]. Available: http://nano.stanford.edu/model.php?id=25

[19] *Stanford University MOSFET-Like CNFET Model*. (2008) [Online]. Available: http://nano.stanford.edu/model.php?id=23

[20] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1090, Jul. 1997.

[21] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, MA: Pearson/Addison-Wesley.

[22] *ABC Logic Synthesis Tool* [Online]. Available: http://www.eecs.berkeley.edu/∼alanmi/abc/

[23] K. Nose and T. Sakurai, "Analysis and future trend of short-circuit power," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 9, pp. 1023–1030, Sep. 2000.

[24] R. Gu and M. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 707–713, May 1996.

[25] *International Technology Roadmap for Semiconductors*. (2007) [Online]. Available: www.itrs.net/reports.html

[26] R. Chau, S. Datta, M. Doczy, B. Doyle, B. Jin, J. Kavalieros, A. Majumdar, M. Metz, and M. Radosavljevic, "Benchmarking nanotechnology for high-performance and low-power logic transistor applications," *IEEE Trans. Nanotechnol.*, vol. 4, no. 2, pp. 153–158, Mar. 2005.

[27] S. Fregonese, C. Maneux, and T. Zimmer, "A compact model for double gate carbon nanotube FET," in *Proc. ESSDERC*, 2010, pp. 452–455.

M. Haykel Ben-Jamaa (S'08–M'10) graduated in the field of electrical engineering from Technische Universität München, München, Germany, and Ecole Centrale Paris, Paris, France, and received the Ph.D. degree from École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in September 2009.

He is currently a Post-Doctoral Researcher with Commissariat à l'Energie Atomique et aux Energies Alternatives, Grenoble, France. He is working on the design aspects for nano-electronics with a tight link to emerging fabrication technologies. His work covers regular logic circuits such as field-programmable gate array, emerging memories, and 3-D integration.

Dr. Ben-Jamaa received the EDA Outstanding Dissertation Award from DATE 2010. He served many conferences as a TPC member or chair, including DATE in 2008, NOCs in 2010, and VLSI-SoC in 2010.

Kartik Mohanram (S'00–M'04) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 1998, and the M.S. and Ph.D. degrees in computer engineering from the University of Texas, Austin, in 2000 and 2003, respectively.

He is currently with the Departments of Electrical and Computer Engineering and Computer Science, Rice University, Houston, TX. His current research interests include computer engineering and systems, nano-electronics, and computational biology.

Dr. Mohanram is a recipient of the NSF CAREER Award, the ACM/SIGDA Technical Leadership Award, and the A. Richard Newton Graduate Scholarship.

Giovanni De Micheli (S'79–M'79–SM'80–F'94) is currently a Professor and Director of the Institute of Electrical Engineering and of the Integrated Systems Center, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is a Program Leader of the Nano-Tera.ch Program. Previously, he was a Professor of Electrical Engineering with Stanford University, Stanford, CA. His current research interests include several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips, and 3-D integration. He is also interested in heterogeneous platform design including electrical components and biosensors, as well as in data processing of biomedical information.

He was the recipient of the 2003 IEEE Emanuel Piore Award. He is a Fellow of ACM. He received the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000 and the 1987 D. Pederson Award for the Best Paper on the IEEE TCAD/ICAS. He was the Division 1 Director from 2008 to 2009, Co-Founder, and President Elect of the IEEE Council on EDA from 2005 to 2007, the President of the IEEE CAS Society in 2003, and the Editor-in-Chief of the IEEE TCAD/ICAS from 1987 to 2001. He is and has been the chair of several conferences, including DATE in 2010, pHealth in 2006, VLSI SoC in 2006, DAC in 2000, and ICCD in 1989.