Universal Logic Modules Based on Double-Gate Carbon Nanotube Transistors

Andrew Zukoski, Xuebei Yang, and Kartik Mohanram
Department of Electrical and Computer Engineering, Rice University, Houston
{andrew.zukoski,xbyang,kmram}@rice.edu

Abstract

Double-gate carbon nanotube field-effect transistors (DG-CNTFETs) can be controlled in the field to be either n-type or p-type through an extra polarity gate. This results in an embedded XOR behavior, which has inspired several novel circuit designs and architectures. This work makes the following contributions. First, we propose an accurate and efficient semi-classical modeling approach to realize the first SPICE-compatible model for circuit design and optimization of DG-CNTFETs. Second, we design and optimize universal logic modules (ULMs) in two circuit styles based on DG-CNTFETs. The proposed ULMs can leverage the full potential of the embedded XOR through the FPGA-centric lookup table optimization flow. Further, we demonstrate that DG-CNTFET ULMs in the double pass-transistor logic style, which inherently produces dual-rail outputs with balanced delay, are faster than DG-CNTFET circuits in the conventional single-rail static logic style that relies on explicit input inversion. On average across 12 benchmarks, the proposed dual-rail ULMs outperform the best DG-CNTFET fabrics based on tiling patterns by 37%, 12%, and 33% in area, delay, and total power, respectively.

Categories and Subject Descriptors: B.7.1 [Integrated circuits]: Types and Design Styles—Advanced technologies
General Terms: Design, Performance
Keywords: Carbon nanotubes, double-gate, universal logic module, double pass-transistor logic

1. Introduction

Carbon nanotube field-effect transistors (CNTFETs) offer high mobility for ballistic transport, high mechanical and thermal stability, and high resistance to electromigration, attracting strong interest as alternative device technologies for future nanoelectronics applications [1]. Although different families of CNTFETs have been fabricated and studied, the most important distinction is between MOSFET-type and Schottky-barrier-type (SB-type) CNTFETs [2, 3]. MOSFET-type devices are characterized by doped CNT channels and Ohmic contacts, and pose more engineering challenges due to the immature doping techniques for CNTs. In contrast, SB-type CNTFETs use intrinsic CNT channels with metallic drain and source contacts, and hence are more easily fabricated.

In contrast to unipolar CMOS devices whose polarity (n-type or p-type) is determined during fabrication, SB-type CNTFETs are ambipolar, i.e., they conduct both electrons and holes, showing a superposition of electron and hole currents. Recent work has demonstrated that ambipolar conduction can be controlled by fabricating a second gate — commonly referred to as the polarity gate — within the transistor [4]. Such a double-gate CNTFET (DG-CNTFET) can be configured as n-type or p-type if the polarity gate has a high or low voltage, respectively. This extra gate makes the DG-CNTFET polarity controllable in the field — with an embedded XOR behavior — inspiring novel circuit designs and architectures. In [5, 6], a compact in-field reconfigurable dynamic logic gate to map eight different logic functions of two inputs using only seven DG-CNTFETs was described. In [7], a generalized NOR gate was proposed as the core building block to realize in-field configurable logic arrays. In [8], a static logic gate library based on DG-CNTFETs was proposed. More recently, regular static logic fabrics [9] and clocked standard cells [10] have been proposed.

However, we believe that these works suffer from the following shortcomings. First, although there have been various models proposed for CNTFETs, none of them are suitable to simulate DG-CNTFET circuits. There have been approaches to approximate DG-CNTFETs by superposition of I-V data of individual n-type and p-type MOSFET-type CNTFETs; however, such approaches are only capable of simulating MOSFET-type CNTFETs that are substantially different from SB-type DG-CNTFETs, motivating research in accurate DG-CNTFET modeling. Second, although it is technologically possible to realize the full swing that is necessary for optimum DG-CNTFET operation on both gates, this requires that the DG-CNTFET circuits provide dual-rail outputs with balanced delays. However, to date, all circuit styles proposed in literature use single-ended logic styles and explicitly invert the output or the essential inputs to realize the complementary values necessary to drive DG-CNTFETs. Finally, state-of-the-art synthesis and mapping tools are limited in their ability to fully exploit the embedded XOR functionality in complex DG-CNTFET logic gates. Although DG-CNTFET logic gates provide gains over conventional logic gates, there is sufficient motivation to explore alternate design approaches that can utilize the embedded XOR more efficiently.

Motivated by these observations, this paper makes the following contributions. First, we propose a semi-classical modeling approach to realize the first SPICE-compatible model for circuit design and optimization of DG-CNTFETs. Our proposed semi-classical model is the first semi-classical model for the simulation of intrinsic DG-CNTFETs. When validated against the accurate quantum simulator NanoTCAD ViDES [11], we observe that the average difference in results is 10%–20% for a simulation time reduction of 3–4 orders of magnitude. When validated against fabricated planar DG-CNTFETs, the I-V results are within 2–4$\times$ of the reported data under similar device parameters and operating conditions, showing that our semi-classical model is both accurate and computationally efficient. The extracted I-V and C-V data from our semi-classical model is stored in a lookup table, which is then used by Verilog-A to implement a lookup table model for circuit simulation in SPICE.

Second, we propose a design approach based on universal logic modules (ULMs) to fully exploit the embedded XOR in ambipolar DG-CNTFETs. Universal logic modules (ULMs) can be suc-

This research was supported in part by NSF grant CCF-0916636 and in part by NSF CAREER Award CCF-0746850.

DAC'11, June 5-10, 2011, San Diego, California, USA.

cinctly described as m-input logic gates that are capable of implementing any n-input Boolean function ($m > n$, referred to as m, n-ULMs) through a combination of input permutation, input inversion, and output inversion. DG-CNTFETs open new possibilities for ULM design, allowing the XOR functionality inherent in ULMs to be efficiently implemented. Furthermore, ULMs are interconnect-configurable and compare favorably with traditional lookup tables (LUTs) in FPGAs. Using DG-CNTFETs for ULM design can thus leverage the FPGA-centric LUT optimization flow, which can address the limitations in XOR utilization inherent to conventional synthesis flows. Finally, ULMs exhibit strong regularity, which not only simplifies the design process, but also provides immunity to process variations.

We design 3,2- and 5,3-ULMs using DG-CNTFETs based on the single-rail static logic style [7] and the double pass-transistor logic (DPL) style. The DPL style was introduced in [12] as a fast dual-rail logic with balanced delay on both rails. We demonstrate that DG-CNTFET ULMs in the DPL style, which inherently produce dual-rail outputs with balanced delay, are faster than DG-CNTFET circuits in the conventional single-rail static logic style that relies on explicit input inversion. We validate and optimize both the single-rail and dual-rail ULMs using our SPICE-compatible DG-CNTFET model. We compare the proposed ULMs against the standard tiling cells proposed in [9], which are the most generic proposed to date in literature. On average across 12 benchmarks, the proposed dual-rail ULMs (single-rail ULMs) outperform the fabrics based on tiling patterns in [9] by 37% (36%), 12% (-23%), and 33% (36%) in area, delay, and total power, respectively.

This paper is organized as follows. Section 2 describes SPICE-compatible DG-CNTFET modeling. Section 3 describes the proposed single-rail and dual-rail ULMs based on DG-CNTFETs. Section 4 presents results. Section 5 is a conclusion.

2. DG-CNTFET modeling

In this section, we first introduce the background and principles of operation of DG-CNTFETs. We then propose and present the first semi-classical model for DG-CNTFETs, based on which a SPICE-compatible model is implemented.

2.1 DG-CNTFET background

DG-CNTFETs with coaxial and planar geometries have been proposed in literature. The DG-CNTFET with a coaxial gate geometry illustrated in Figure 1 has been shown to provide the best gate control [13], and can be used to explore the performance limits of the DG-CNTFET. In addition to the coaxial DG-CNTFET, our approach can also handle the planar DG-CNTFET structure, which has been experimentally demonstrated [14]. In both DG-CNTFETs, the regions on either end of the channel are controlled by the polarity gate (PG), which determines whether the device is p-type or n-type. The polarity gate surrounds the primary gate (G), which controls the region in the center of the channel and governs the switching of the transistor. The primary and polarity gates are ideally isolated from each other, and the isolation distance is assumed to be negligible.

In order to illustrate the operating principle for the DG-CNTFET, we present the energy band diagram along the channel direction under different V_{PG} and V_G in Figure 2. Note that when both V_{PG} and V_G are high/low, there is no energy barrier inside the channel, and the transistor is turned on due to electron/hole conduction. However, if one of V_{PG} and V_G is high and the other is low, there exists an energy barrier inside the channel, and the transistor is off. This behavior is equivalent to embedding the XNOR operation in the inputs driving the two gates of the DG-CNTFET.

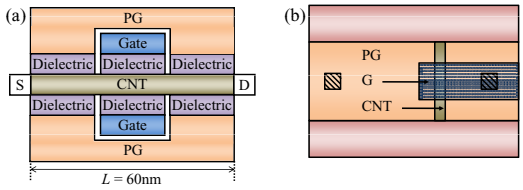

Figure 1: DG-CNTFET (a) cross-section and (b) layout. PG (G) denotes the polarity (primary) gate.

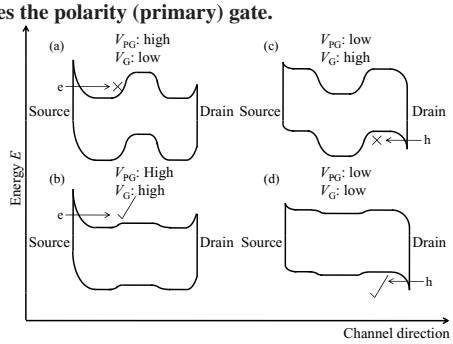

Figure 2: Energy profile for different primary gate and polarity gate voltages.

2.2 DG-CNTFET modeling

Although several models have been proposed for CNTFETs, none of them are suitable to simulate DG-CNTFET circuits. For example, although existing quantum simulators such as NanoTCAD ViDES [11] can be used to simulate DG-CNTFETs, they are very slow and computationally inefficient. Therefore, it is not practical to build a model for circuit simulation of DG-CNTFETs based on quantum simulators. In contrast, semi-classical models for CNTFETs [15–17] are fast and time-efficient, but cannot handle the double-gate structure. As a result, all existing works on DG-CNTFETs are based on the MOSFET-type CNTFET model proposed in [15] and rely on superposition of n-type and p-type I-V data of MOSFET-type CNTFETs to approximate the behavior of a DG-CNTFET. However, since MOSFET-type CNTFETs are inherently different from DG-CNTFETs, we believe that a DG-CNTFET model that is compatible with a circuit simulator such as SPICE is critical for circuit design studies of DG-CNTFETs.

In this work, we build the first SPICE-compatible DG-CNTFET model based on the first semi-classical model that can simulate intrinsic DG-CNTFETs. Our semi-classical model is parameterizable to study the role of common design and process parameters such as CNT chirality, SB height, and gate dielectric thickness, and allows quick, accurate simulation and evaluation of a large design-space. In this work, we consider a (13,0) CNT channel with a 1.5nm-thick oxide as the gate insulator. The height of each SB is assumed to be half of the CNT energy band-gap (mid-gap SB). We assume a ballistic channel length of 60nm, which is divided into three regions. Without loss of generality, it is assumed that the gate in each region has the same length (20nm). The gate current is assumed to be negligible, and the drain current of the transistor is evaluated using the Landauer formula, which takes the form:

$$I = \frac{2q}{h} \int_{-\infty}^{\infty} \left(f(E - E_{FS}) - f(E - E_{FD}) \right) T(E)\, \mathrm{d}E$$

where I is the drain current; h is the Planck's constant; $f()$ is Fermi function; E_{FS} and E_{FD} are Fermi levels in source and drain contacts; and $T(E)$, the transmission coefficient, is the probability that a carrier reaches one contact from the other. The reader is referred

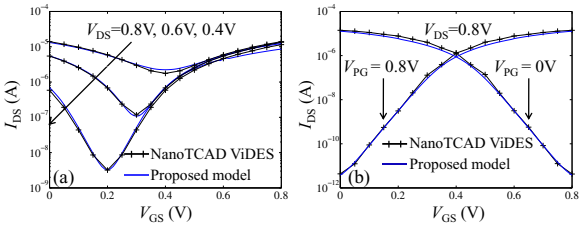

Figure 3: Comparison of coaxial DG-CNTFET model with NanoTCAD ViDES for (a) tied and (b) independent polarity gate and primary gate

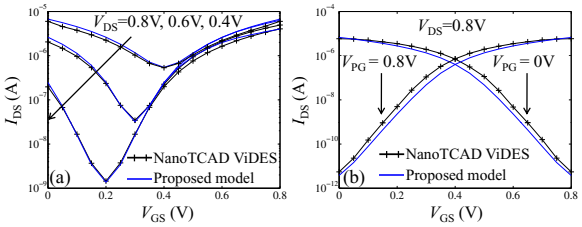

Figure 4: Comparison of planar DG-CNTFET model with NanoTCAD ViDES for (a) tied and (b) independent polarity gate and primary gate

to [18] for the details on the evaluation of the transmission coefficient $T(E)$.

The accuracy of the proposed model is illustrated in Figure 3 by comparing the results with that of NanoTCAD ViDES [11], a versatile NEGF-based quantum transport simulation framework. In Figure 3(a), the polarity gate and the primary gate are tied together, so the DG-CNTFET acts as a single-gate CNTFET and ambipolar conduction is clearly shown. In Figure 3(b), the polarity gate and the primary gate are controlled separately. When the voltage on the polarity gate is high and low, the DG-CNTFET exhibits n-type and p-type behavior, respectively, which can be understood from Figure 1(b). For a total of 85 bias points, the average difference observed between NanoTCAD ViDES and the proposed model is 10.5%, but the simulation time of the proposed model is 3–4 orders of magnitude lower. In Figure 4, we also present results for a planar DG-CNTFET to illustrate the applicability and accuracy of the proposed modeling approach. The average difference between NanoTCAD ViDES and the proposed model across 85 bias points is 20%. The simulation results also match the experimental results for the fabricated planar DG-CNTFET reported in Figure 12(a) of [14] under similar device parameters and operating conditions, with comparable I_{on} (4×10^{-8}A versus 1×10^{-8}A), I_{off} (1×10^{-13}A versus 5×10^{-14}A), and I_{on}/I_{off} (4×10^{5} versus 2×10^{5}). The differences can be attributed to (i) assumption of a ballistic channel without scattering and (ii) parasitics and other non-idealities related to the experiments. In summary, the proposed model is of sufficient accuracy to describe the I-V performance of the DG-CNTFET, and it is computationally efficient to enable circuit level simulation.

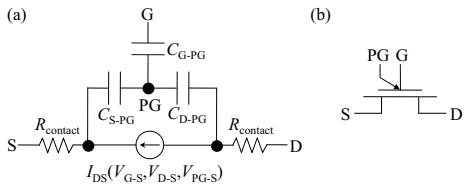

Figure 5: DG-CNTFET circuit (a) model and (b) symbol

Next, we build a SPICE-compatible circuit model for the DG-CNTFET, as shown in Figure 5. Note that without loss of generality, we consider coaxial DG-CNTFETs in the rest of this paper. The I-V and C-V data extracted from the semi-classical model are stored in a lookup table, and used by Verilog-A to implement a lookup table model for circuit simulation in SPICE. Note that the capacitance is extracted using the method described in [19]. Three capacitances, $C_{PG\text{-}S}$, $C_{PG\text{-}D}$, and $C_{PG\text{-}G}$ are considered, as shown in Figure 5(a). The capacitance between the primary gate and the source/drain ($C_{G\text{-}S/D}$) is assumed to be negligible since the primary gate is wrapped by the polarity gate. Since the I-V and C-V data are extracted from a single nanotube in the semi-classical model, when the width of the DG-CNTFET is increased in the SPICE model to include multiple nanotubes in the channel, the current and capacitance will increase correspondingly. The SPICE model also incorporates extrinsic effects such as the contact resistance, which is on the order of kΩ as reported in [20].

3. Universal logic modules (ULMs)

In this section, we first discuss the limitations of related work in XOR-based synthesis and ambipolar-device regular logic fabrics. We next introduce universal logic modules (ULMs), and motivate their advantages for DG-CNTFET-based design. We finally present the design of ULMs based on 2 circuit styles using DG-CNTFETs.

3.1 Related work, limitations, and challenges

The novel in-field programmability of DG-CNTFETs was investigated in previous works in order to extend the functional possibilities offered by MOSFETs. The first explorations were reported in [5, 6], where a compact in-field reconfigurable dynamic logic gate that maps eight different logic functions of two inputs using only seven DG-CNTFETs was presented. In [7] the design of a generalized NOR (GNOR) gate was proposed as the core building block to realize in-field programmable logic arrays. The GNOR gate has a compact design and a high expressive power by combining both NOR and XOR operations in the output function. However, these designs are based on dynamic logic, which is vulnerable to internal signal races and/or fails to provide full swing at the outputs, which is necessary to drive DG-CNTFETs.

In [8], the first static logic based on DG-CNTFETs was proposed, using the transmission-gate style shown in Figure 6(a). Note that the transmission-gate based on DG-CNTFETs will only conduct when A and B have the same logic value, thereby efficiently implementing the XNOR function. This design can provide full swing at the output, and outperforms the 32nm CMOS technology with 20× lower EDP [8]. Using the transmission-gate transistor structure of [8], a 2-input XOR gate can be constructed using only 4 transistors. If the supply and ground terminals in this gate are replaced by a third input and it's complement, a 3-input XOR gate with rail-to-rail swing that also requires only 4 transistors can be constructed as shown in Figure 6(b). These device counts are comparable to NAND and NOR gates, unlike CMOS implementations.

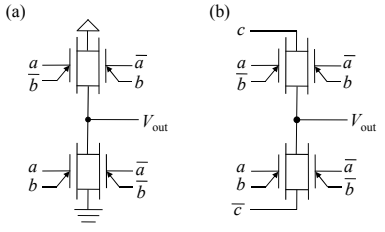

Figure 6: Schematic for (a) 2-input XOR [8] and (b) 3-input XOR (proposed in this paper) based on DG-CNTFETs

However, there are two important limitations. First, although it is technologically possible to realize the full-swing on both gates necessary for optimum DG-CNTFET operation, this requires that the DG-CNTFET circuits provide dual-rail outputs with balanced delays. However, to date, all circuit styles proposed in literature use single-ended logic styles and explicitly invert the output or the essential inputs to realize the complementary dual-rail signals necessary to drive DG-CNTFETs. Second, and more importantly, state-of-the-art synthesis and mapping tools are limited in their ability to fully exploit the embedded XOR functionality in complex DG-CNTFET logic gates. This is a result of the historical cost comparison between NAND/NOR gates and XOR gates: a 2-input NAND gate requires 4 CMOS transistors, while a 2-input XOR gate requires 8. This cost difference has created a long-standing and powerful incentive to minimize the use of XOR-intensive representations during synthesis and mapping.

Although there have been several tools proposed for XOR utilization and optimization, e.g., [21–23], they are of limited scalability and have not achieved widespread application. Mapping the ISCAS benchmark circuits to the ambipolar library described in [8] using the tool ABC, we observe that the gates implementing the functions $\overline{a+b}$, $\overline{a+b+c}$, and $\overline{a+bc}$ comprised 76.8% of the gates (not counting inverters). This high utilization of conventional gates demonstrates that even with favorable delay and area characteristics, conventional optimization makes poor use of XOR functionality and the expressive capabilities of DG-CNTFETs.

Motivated by the functional flexibility of ambipolar gates, the most recent work [9] on static DG-CNTFET logic has argued for regular logic fabrics, citing increased manufacturability — an important consideration for an emerging technology. In [9], different tiling patterns were chosen and evaluated based on the number of sub-functions that each tiling pattern can implement. However, in our simulations, we have observed that benchmarks synthesized and mapped to the sub-functions of the leading tiling (F21/F22 in [9]) utilize XOR functionality at only 26.5% of nodes. Motivated by this, we propose an alternate design approach based on universal logic modules (ULMs) that can make heavy use of the embedded XOR in the ambipolar DG-CNTFET and that can be easily incorporated into a conventional optimization flow for FPGAs.

3.2 Universal logic modules

In this paper, we propose ULMs utilizing ambipolar DG-CNTFETs for regular logic fabrics. An m, n-ULM is a logic gate with m inputs that is capable of implementing any n-input Boolean function through configuration of the $m > n$ input pins. ULMs are interconnect-programmable devices: functions are implemented by wiring input variables, their complements, logical 1, and/or logical 0 to the input pins in various combinations. ULMs have received sustained interest for over forty years [24–28], and three critical advantages as logic blocks for regular fabrics have emerged:

1. ULMs are configurable using only interconnects. Each ULM is patterned as a regular cell, and only metalization layers are required to specify functionality. This reduces the effects of process variations as only a single cell must be optimized.
2. Like an n-input lookup table (LUT), an m, n-ULM can take advantage of a simplified synthesis and optimization flow that is very similar to FPGAs. The optimization process of mapping to LUTs eliminates some of the bias against XOR functionality inherent to conventional synthesis flows.
3. ULMs are smaller and faster than LUTs. LUTs require 2^n SRAM cells and a 2^n-to-1 multiplexor that is expensive in both area and delay. While not field-programmable, ULMs retain the functional flexibility of LUTs while exhibiting the attractive performance of ASIC-style logic circuits.

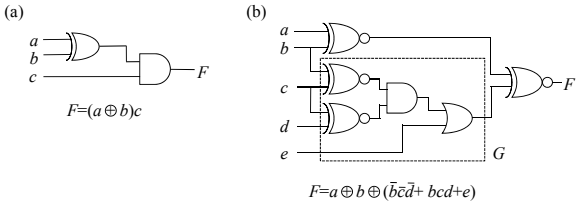

Figure 7: (a) Schematic for the (a) 3,2-ULM and (b) 5,3-ULM

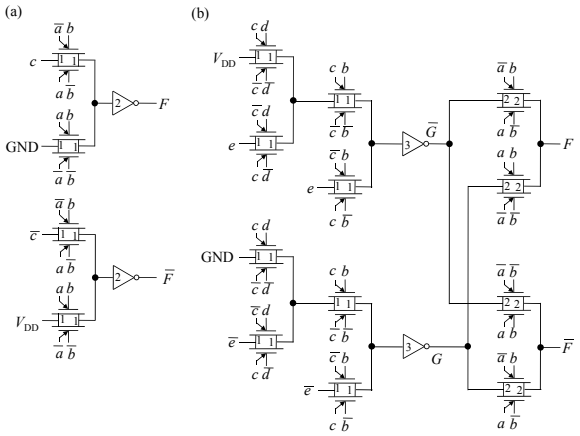

Figure 8: DPL-style circuit of the (a) 3,2-ULM and (b) 5,3-ULM. Optimal transistor sizes are annotated for reference.

Designing a ULM that exploits the unique capabilities of ambipolar DG-CNTFETS opens an avenue to 'trick' a synthesis and optimization flow into utilizing XOR-rich functions — XOR functionality need not be explicitly optimized for or against. The design of ULMs has conventionally focused on implementing all members of NP- or NPN-equivalence classes. These classes are defined as those Boolean functions that are equivalent under the operations of input negation, input permutation, and output negation (see [29] for a discussion of equivalence classes). XOR functionality embedded in ULM designs represents an opportunity to utilize the expressiveness inherent to ambipolar DG-CNTFET transistors.

3.3 Dual-rail 3,2-ULM and 5,3-ULM design

In Figures 7(a) and 7(b), we present proposed designs for the 3,2-ULM and the 5,3-ULM, respectively. In Figures 8(a) and 8(b), we present the corresponding circuit schematics for the implementation of these ULMs in the double pass-transistor logic style (DPL-style). Both designs provide a complementary pair of outputs, as required for DPL-style fanout gates and input complementation in Boolean matching. Both of these designs heavily leverage the ambipolar characteristics of DG-CNTFETS: in both designs, the only transistors not implemented in their field-programmable configuration are the inverters. CMOS-based implementations of these functions will require an additional 2 and 12 transistors for the 3,2-ULM and the 5,3-ULM, respectively.

There are six possible 3,2-ULM designs [29]. Three of these are NPN complete, and three of these are NP complete (not requiring output inversion to implement all two input functions). Implemented as dual-rail transmission-style gates, the question of output inversion is moot, as both complemented and uncomplemented out-

puts are readily available. As a result, we implement the function

$$f(a,b,c) = \overline{(a \oplus b)c} \quad (1)$$

There are four NPN classes of functions of two or fewer variables, represented by the functions: 0, x_1, x_1x_2, and $x_1 \oplus x_2$. Within our 3,2-ULM, the constant functions can be implemented by hardwiring the three inputs, and both the buffer and inverter functions can be (simultaneously) implemented by wiring inputs b to ground and c to V_{DD} ($\overline{b}$ to V_{DD} and $\overline{c}$ to ground). The implementation of a function such as x_1x_2, for example, will require wiring x_1 to input c, $\overline{x_1}$ to $\overline{c}$, x_2 to b, $\overline{x_2}$ to $\overline{b}$, GND to a, and V_{DD} to $\overline{a}$. With output f representing the AND function, $\overline{f}$ represents the NAND function. Inversions to the inputs can be accomplished through swapping the wiring assignments of a and $\overline{a}$.

It has been demonstrated that it is not possible to realize a 4,3-ULM [29]. Of the several designs for 5,3-ULMs proposed in literature, we have implemented the function first proposed by Edwards [26]:

$$f(a,b,c,d,e) = a \oplus b \oplus (bcd + \overline{b}\overline{c}\overline{d} + e) \quad (2)$$

This function (see schematic in Figure 7(b)) is rich in XOR functionality. Other proposed 5,3-ULMs result in smaller CMOS implementations, but do not result in smaller implementations with DG-CNTFET technology. For brevity, we do not illustrate the wiring configurations for the 5,3-ULM.

Our ULM designs were validated using SPICE simulations based on our DG-CNTFET model. We report min/max delays on transitions appropriate for a configured ULM — many configurations of the 5,3-ULM have inputs wired to more then one pin (for instance, implementing functions of the NPN class represented by $f(x_1,x_2,x_3) = \overline{x_1x_2x_3} + x_1x_2$ requires wiring inputs a and $\overline{a}$ to $\overline{e}$ and e, respectively, and inputs c and $\overline{c}$ to GND and V_{DD}, respectively. To account for this, we tested multi-bit transitions arising from each of the ten NPN equivalence class configurations.

In addition to our DPL-style designs, we implemented, sized, and tested static, complementary gates for the 5,3- and 3,2-ULMs. These designs are similar to those proposed in [9], utilizing complementary pull-up and pull-down networks with appropriate and compact XOR implementations. These gates are single-rail, utilizing unit-size inverters (again, as in [9]) on the inputs to generate complementary signals when required for transmission gates. For the 5,3-ULM, we implement the sub-function G (see Figure 7(b)) as a complementary network, and implement the remaining 3-input XOR using the structure from Figure 6(b). An intermediate unit size inverter generates the necessary $\overline{G}$ signal.

Finally, we report delay characteristics for the F21/F22 regular logic tiling based on the transistor sizings reported in [9]. In all delay simulations, V_{DD} was 0.9 volts and the input slew was 5 picoseconds. We report the sum over transistor widths for each gate as an indicator of circuit area. All the gates considered in this paper exhibit full rail-to-rail swing.

In summary, our DG-CNTFET implementations of 3,2- and 5,3-ULMs are attractive because they (i) leverage ambipolar capabilities to reduce device count from similar CMOS implementations, (ii) provide a complementary, balanced pair, i.e., dual-rail outputs to drive fanout gates, and (iii) can be integrated into the traditional LUT-based flow for FPGAs. In the following section we will discuss circuits synthesized using our ULMs, and compare the results to the leading existing proposal for regular logic cells.

4. Results

In this work we compare the simulation results between three logic families: the F21/F22 tiling pattern [9], the proposed single-rail static logic ULMs, and the DPL-style ULMs. The comparison between DG-CNTFET and CMOS logic circuits is omitted here as previous research has demonstrated a 4× lower intrinsic CV/I delay at comparable gate lengths for the SB-type CNTFET over scaled CMOS. This advantage has also been recently confirmed for CNTFETs at a channel length of 15nm [30]. Therefore, since DG-CNTFETs are a type of SB-type CNTFET, it is reasonable to believe that DG-CNTFETs will maintain these advantages and outperform scaled CMOS at 32nm and beyond.

Table 1: Min/max delays for the ULMs and F21/F22 tiling (ps)

Module		Rise delay (ps)		Fall delay (ps)		Area
		f	$\overline{f}$	f	$\overline{f}$	
Single-rail	3,2-ULM	5.4/5.8	—	3.1/4.8	—	17
	5,3-ULM	6.6/11.8	—	7.8/13.1	—	35
Dual-rail	3,2-ULM	4.1/4.1	4.1/4.1	4.1/4.1	4.1/4.1	16
	5,3-ULM	4.2/8.8	4.2/8.8	4.2/8.8	4.2/8.8	44
F21/F22 tiling [9]		2.5/6.3	—	2.5/6.3	—	28

We used the tool ABC developed at Berkeley [31] for logic synthesis and technology mapping of several benchmark circuits. The circuits were first synthesized using the `resyn2rs` script, followed by technology mapping using libraries for each family based on the area-delay values from Table 1. Mapping to the ULMs was accomplished by specifying a LUT architecture composed of 2- and 3-LUTs. The 2- and 3-LUTs were provided with area equal to the 3,2- and 5,3-ULMs and delay equal to the worst case among all transitions from Table 1. FPGA mapping was performed with the area optimization option enabled. Mapping to the regular tiling was performed by enumerating the sub-functions available under the operations of hardwiring input pins and duplicating inputs inside a genlib gate library for use by ABC. The area for the regular tiling is based on the values specified in [9], and the delay characteristics were derived from our SPICE simulations.

The results for 12 benchmark circuits are summarized in Table 2. We report area, delay, dynamic power, and static power. Area was calculated using the sum of the transistor sizes, i.e, weighted device count. Delay was calculated using static timing analysis. Dynamic power consumption was calculated using switching activity extracted using random pattern simulation and worst-case switching capacitance. Static power was estimated as the average power across all input patterns to the respective modules.

Examining the results for the three logic families, we see that in comparison to the F21/F22 tiling pattern of [9], our single-rail ULMs offer average improvements in area, dynamic power, and static power of 36%, 36%, and 37%, respectively. However, this comes with a 23% increase in circuit delay on average. The dual-rail DPL-style ULMs outperform the F21/F22 tiling simultaneously in terms of area, delay and dynamic power, with an improvement of 37%, 12%, and 34%, respectively, on average across all the benchmarks. The static power consumption of the dual-rail ULMs was the highest of the three sets of circuits explored. As the dynamic power consumption is several orders of magnitude larger than static power consumption (μW versus nW), this is not a critical concern. Indeed, when total power is considered, the dual-rail DPL-style ULMs offer a 33% improvement over the F21/F22 tiling.

Whereas the DPL-style designs use transmission gates and increase the number of devices in the dual-rail ULMs in comparison to their single-rail counterparts, the transistors are narrower, resulting in positive results in terms of power and area. The narrower transistors for DPL-style ULMs result in smaller dynamic power usage for synthesized circuits, and the fast DPL-logic results in a performance increase relative to both the F21/F22 and single-rail

Table 2: Results for optimization: number of patterns/ULMs, area (sum of transistor sizes), logic levels, delay (ps), dyanmic power P_D (μW at 1GHz and 0.9 V), and static power P_S (nW at 0.9 V)

Benchmark		F21/F22 tiling [9]						Single-rail static logic ULMs						Dual-rail DPL-style ULMs					
Circuit	Function	No.	Area	Levels	Delay	P_D	P_S	No.	Area	Levels	Delay	P_D	P_S	No.	Area	Levels	Delay	P_D	P_S
C2670	ALU and control	426	11928	14	88.2	8.66	13.7	504	7636	16	100.1	5.75	10.1	504	7596	16	70.3	4.93	39.3
C1908	Error correcting	230	6440	15	94.5	3.08	7.37	209	3679	19	111.7	1.78	3.85	209	3540	19	78.5	1.89	9.71
C3540	ALU and control	737	20636	21	132.3	10.6	23.6	807	14277	27	159.6	7.04	15.1	807	13780	27	111.9	7.45	40.6
dalu	Dedicated ALU	754	21112	17	107.1	7.93	24.2	884	16018	27	168.4	6.15	17.6	807	15676	27	117.2	6.54	64.4
C7552	ALU and control	961	26908	15	94.5	17.5	30.8	813	16341	20	140.9	11.2	20.6	892	17444	21	96.7	11.6	128.8
C6288	Multiplier	1529	43092	46	289.8	26.9	49.0	1018	21068	47	400.1	13.0	26.3	1018	22140	47	270.4	14.5	185.5
C5315	ALU and selector	906	25368	18	113.4	15.5	29.0	1040	18319	20	152.5	11.2	20.3	1046	17900	21	105.5	11.6	69.0
des	Data encryption	2826	79128	13	81.9	35.7	90.6	2793	52719	14	82.7	23.9	60.0	2793	52837	14	58	26.9	285.8
i10	Logic	1477	41356	24	151.2	16.2	47.3	1497	26774	31	182.9	10.5	29.1	1486	26304	31	128.3	11.1	96.3
t481	Logic	682	19096	13	81.9	3.45	21.9	753	12891	13	76.9	2.05	13.0	753	12188	13	53.9	2.12	19.01
i8	Logic	726	20328	8	50.4	10.7	23.3	938	16000	11	63.8	7.81	16.1	938	15092	11	45.1	8.07	21.3
C1355	Error correcting	230	6440	9	56.7	3.61	7.37	182	3094	10	58	1.71	3.09	182	2912	10	41	1.76	3.64
Average		1.0	1.0	1.0	1.0	1.0	1.0	1.02	0.64	1.22	1.23	0.64	0.63	1.02	0.63	1.23	0.88	0.66	2.20

ULMs. Additionally, because complementary outputs are generated simultaneously, explicit input inverters are not required in circuits, reducing the delay associated with each ULM module.

In both single- and dual-rail ULM implementations, node-count and the number of levels of logic are larger than the F21/F22 tiling. This is a result of ABC's preference for the 3,2-LUT over the 5,3-LUT: the area and delay costs associated with the 5,3-ULM incentivizes utilization of the 3,2-ULM on the critical paths, which results in an increase in the logic depth — but an overall improvement in delay — for synthesized circuits.

The novel design possibilities associated with ambipolar electronics rely on the availability of complemented signals. These results demonstrate that our DPL-style ULMs are capable of exploiting ambipolar properties through simultaneous generation of complemented signals.

5. Challenges and opportunities

Mounting evidence shows that the ability to control ambipolarity in the field presents new logic design possibilities. The realization of a relatively inexpensive XOR gate opens the door to compact yet highly expressive logic gates. However, this advantage cannot be fully exploited by existing design tools, and requires the availability of complemented signals. In this paper, we presented single- and dual-rail universal logic modules (ULMs) as candidates for DG-CNTFET based regular logic fabrics. These gates are able to leverage LUT-based mapping tools to better exploit ambipolar behavior. The dual-rail, DPL-style ULM gates generate balanced, complementary outputs and compared to previously proposed regular tilings [9], provide improvements in area, delay and total power of 37%, 12%, and 33%, respectively.

References

[1] P. Avouris, Z. Chen, and V. Perebeinos, "Carbon-based electronics," *Nature Nanotechnology*, vol. 2, pp. 605–615, 2007.

[2] S. Heinze *et al.*, "Carbon nanotubes as Schottky barrier transistors," *Phys. Rev. Lett.*, vol. 89, p. 106801, 2002.

[3] A. Javey *et al.*, "Ballistic carbon nanotube field-effect transistors," *Nature*, vol. 424, pp. 654–657, 2003.

[4] Y. Lin *et al.*, "Ambipolar-to-unipolar conversion of carbon nanotube transistors by gate structure engineering," *Nano Letters*, vol. 4, pp. 947–950, 2004.

[5] I. O'Connor *et al.*, "CNTFET modeling and reconfigurable logic-circuit design," *IEEE Trans. on Circuits and Systems I*, vol. 54, pp. 2365–2379, 2007.

[6] J. Liu *et al.*, "Novel CNTFET-based reconfigurable logic gate design," in *Proc. Design Automation Conference*, pp. 276–277, 2007.

[7] M. H. B. Jamaa *et al.*, "Programmable logic circuits based on ambipolar CNFET," in *Proc. Design Automation Conference*, pp. 339–340, 2008.

[8] M. H. B. Jamaa *et al.*, "Logic circuits with ambipolar CNTFETs: Novel opportunities for multi-level logic synthesis," in *Proc. Design Automation and Test in Europe*, pp. 622–627, 2009.

[9] D. Marchi *et al.*, "Regular fabric design with ambipolar CNTFETs for FPGA and structured ASIC applications," in *Proc. Intl. Symposium on Nanoscale Architectures*, pp. 65–70, 2010.

[10] K. Jabeur *et al.*, "Reducing transistor count in clocked standard cells with ambipolar double-gate FETs," in *Proc. Intl. Symposium on Nanoscale Architectures*, pp. 47 –52, 2010.

[11] G. Fiori and G. Iannaccone, "NanoTCAD ViDES," 2008. DOI: 10254/nanohub-r5116.3.

[12] M. Suzuki *et al.*, "A 1.5ns 32-b CMOS ALU in double pass-transistor logic," *Journal of Solid State Circuits*, vol. 28, pp. 1145–1151, 1993.

[13] J. Guo *et al.*, "A numerical study of scaling issues for Schottky-barrier carbon nanotube transistors," *IEEE Trans. on Electron Devices*, vol. 51, pp. 172–177, 2004.

[14] Y.-M. Lin *et al.*, "High-performance carbon nanotube field-effect transistor with tunable polarities," *IEEE Trans. Nanotechnology*, vol. 4, pp. 481–489, 2005.

[15] J. Deng and H. Wong, "A compact SPICE model for carbon-manotube field-effect transistors including nonidealities and its application — Part I: Model of the intrinsic channel region," *IEEE Trans. on Electron Devices*, vol. 54, pp. 3186–3194, Dec. 2007.

[16] A. Balijepalli *et al.*, "Compact modeling of carbon nanotube transistor for early stage process-design exploration," in *Proc. Intl. Symposium Low Power Electronics and Design*, pp. 2–7, 2007.

[17] A. Hazeghi *et al.*, "Schottky-barrier carbon nanotube field-effect transistor modeling," *IEEE Trans. on Electron Devices*, vol. 54, no. 3, pp. 439–445, 2007.

[18] X. Yang and K. Mohanram, "Modeling and performance investigation of the double-gate carbon nanotube transistor," *IEEE Electron Device Lett.*, vol. 32, pp. 231–233, 2011.

[19] Y. Tsividis, *Operation and modeling of the MOS transistors*. WCB/McGraw-Hill, 1999.

[20] N. Patil *et al.*, "Circuit-level performance benchmarking and scalability analysis of carbon nanotube transistor circuits," *IEEE Trans. Nanotechnology*, vol. 8, pp. 37–45, 2009.

[21] N. Song and M. A. Perkowski, "Exorcism-mv-2: Minimization of exclusive sum of products expressions for multiple-balued input incompletely specified functions," in *Proc. Intl. Symposium on Multiple-Valued Logic*, pp. 132–137, 1993.

[22] J. P. Hansen and M. Sekine, "Synthesis by spectral translation using Boolean decision diagrams," in *Proc. Design Automation Conference*, pp. 248–253, 1996.

[23] A. Bernasconi *et al.*, "Efficient minimization of fully testable 2-spp networks," in *Proc. Design, Automation and Test in Europe*, pp. 1–6, 2006.

[24] D. C. Forslund, "The universal logic block (ULB) and its application to logic design," *Annual Symposium on Switching and Automata Theory*, pp. 236–250, 1966.

[25] S. Yau and C. Tang, "Universal logic modules and their applications," *IEEE Transactions on Computers*, vol. C-19, no. 2, pp. 141 – 149, 1970.

[26] C. Edwards, "A special class of Universal logic gate (ULG) and their evaluation under the walsh transform," *Computers and Digital Techniques*, vol. 44, pp. 49–59, 1979.

[27] X. Chen and X. Wu, "Derivation of universal logic modules, for n$\geq$3, by algebraic means," *Computers and Digital Techniques*, vol. 128, pp. 205 –211, 1981.

[28] M. Hutter, "Designing universal logic modules," in *Proc. Intl. Conference on Electronics, Circuits and Systems*, pp. 709–712, 2002.

[29] S. Hurst, D. Miller, and J. Muzio, *Spectral Techniques in Digital Logic*. Academic Press Inc. (London) Ltd., 1985.

[30] A. D. Franklin *et al.*, "Length scaling of carbon nanotube transistors," *Nature Nanotechnology*, vol. 5, pp. 858–862, 2010.

[31] "ABC Logic synthesis tool." Please visit the URL http://www.eecs.berkeley.edu/~alanmi/abc/ for further details.