

FPGA Design with Double-Gate Carbon Nanotube Transistors

M. H. Ben Jamaa^a, P.-E. Gaillardon^a, S. Frégonèse^b, M. De Marchi^c,
G. De Micheli^c, T. Zimmer^b, I. O'Connor^d, and F. Clermidy^a

^a Commissariat à l'Énergie Atomique (LETI), Minatec Campus, 38054 Grenoble, France

^b Laboratoire IMS, CNRS-UMR 5218, Université Bordeaux 1, 33405 Talence, France

^c Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

^d Institut des Nanotechnologies de Lyon (INL), Site ECL, 69134 Ecully, France

Double-gate carbon nanotube field effect transistors (DG-CNTFETs) are novel devices showing an interesting property allowing to control the p- or n-type behavior during the device operation. This opens up the opportunity for novel design paradigms. Based on a compact physical model of these devices, we demonstrate the benefit of designing *field-programmable gate arrays (FPGAs)* using fine-grain DG-CNTFET logic blocs rather than traditional look-up tables and coarse-grain DG-CNTFET logic blocs. In particular, we show a reduction by 13% to 48% on average in terms of delay of FPGA benchmarks.

Introduction

The scaling down of complementary metal-oxide-semiconductor (CMOS) technology has led to the emergence of novel post-CMOS devices, such as *carbon nanotubes (CNTs)*. One of the challenges of using CNT technology for building transistors is the chemical doping. Using undoped CNTs is possible, but it results in an ambipolar behavior of the *carbon nanotube field effect transistors (CNTFETs)*, meaning that undoped CNTFETs conduct under both positive and negative gate bias. This issue is addressed using a second gate, which controls whether the device operates as p- or a n-type [1]. The polarity of *double-gate* (or *dual-gate*) *CNTFETs (DG-CNTFETs)* can be selected during operation time.

This property offers the opportunity to design logic gates with reconfigurable devices, leading to more logic functions drawn on the same silicon area. We leverage this property by constructing a full reconfigurable logic system that is reminiscent of a *field-programmable gate array (FPGA)*. Reconfigurable circuits are gaining interest because of their low technology cost, fast design time and enhanced fault-tolerance compared to *application-specific integrated circuits (ASICs)* [2]. However, FPGAs necessitate a large number of configuration memories and have a higher cost in terms of area and power consumption compared to ASICs. Our approach to address these issues is to implement FPGAs using reconfigurable DG-CNTFET logic gates instead of *look-up tables (LUTs)* as *basic logic elements (BLEs)*. Our BLEs require less configuration memory, and their design is area efficient.

In order to assess the benefits of the proposed approach, we first build a family of basic logic elements with DG-CNTFETs, which we characterize using a compact model.

We then evaluate the FPGA performance on a large application benchmark for different architecture scenarios. The same design flow is used with LUT-based FPGAs. We demonstrate that the proposed approach with fine-grain DG-CNTFET gates is 48% faster than coarse grains and 13% faster than LUTs. FPGAs with fine-grain DG-CNTFET gates have a cost in terms of area with respect to LUTs (10%), while they are 45% smaller than coarse-grain DG-CNTFET gates.

The paper is organized as follows. We first survey previous works dealing with DG-CNTFET technology. Then we introduce a compact physical model for the considered devices. Subsequently, we introduce the design of reconfigurable DG-CNTFET FPGAs. Finally, we benchmark the presented approach and compare it with the standard LUT-based implementations.

Background and Related Work

Following more than one decade of research on CNTFETs, several issues have been addressed and some challenges have been identified. In this section, we first survey the state-of-the-art of CNTFET technology, before we focus on the physics of DG-CNTFETs.

Depending on the used materials and doping profile, different types of CNTFETs are reported in literature: the major distinction is between MOSFET-type CNT and Schottky-Barrier-type [3]. The first family is characterized by an intrinsic CNT channel and highly doped drain/source regions forming an ohmic contact to the metal. For the second family, the whole CNT along the channel and the drain/source regions is intrinsic. These devices have a Schottky barrier and are ambipolar, *i.e.*, they conduct both electrons and holes, showing a superposition of n- 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 [1].

Whereas the uncontrollable ambipolar behavior, is undesirable, the ability to select the CNTFET polarity (p- or n-type) in-field by controlling the fringing gate field suggests the innovation of using a second gate, the *polarity gate*, to control the electrical field at the CNT-to-metal junctions and to set the device polarity [1]. The physics of the considered dual-gate device is illustrated using its energy-band diagram. By setting the polarity gate to a high value, the drain and source regions let electrons e^- pass and block holes h^+ . Then, the device operates as a n-type transistor. The opposite happens when the polarity gate is set to a low value, and then the device operates as a p-type transistor.

Modeling Double-Gate Carbon Nanotube Field Effect Transistors

In order to assess the benefits of the considered devices at the circuit architecture level, we first need to simulate the electrical behavior of the building blocks of the circuits. This requires an accurate physical model, which, in turn, depends on the underlying technology. Therefore, we first start by explaining the technological assumptions. Then, we introduce the physical compact model.

Fabrication Process

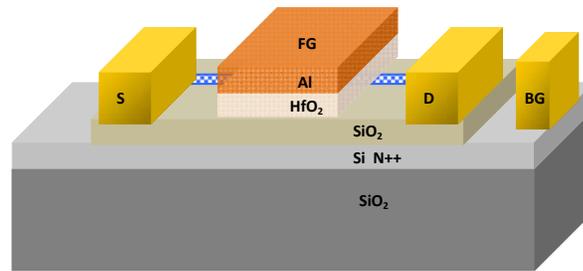


Figure 1: DG-CNTFET device showing the source (S), drain (D), front (FG) and back (BG) gates

At present there is no standard CNTFET process. All developed devices have been produced experimentally. In order to use a realistic and CMOS-compatible process flow, we suggest the following process steps (Figure 1). We start with a SOI wafer, on top of which intrinsic CNTs are deposited or transferred. Then, the gate oxide (HfO_2) and the metal (Al) of the top gate are deposited. Following these steps, the top gate is patterned. Then, the active area and the back gate are defined by SiO_2 and Si etch respectively. Subsequently, the metal is sputtered onto the contacts to drain, source and to both gates.

Physical Device Modeling

The physical model described in [4] is related to the structure in Figure 1, which is made of three different regions: source access, inner part (underneath the front gate) and drain access. In this structure, four energy barriers appear in the device: at the metal to source (or drain) access junction, two SB-like barriers appear, while at the source (or drain) access to the inner part junction, the barrier is more conventional and is of a pn-junction shape. Depending on the work function difference between the metal contact and the CNT, carriers at the metal-CNT interface encounter different barrier heights: Carriers with energies above the Schottky barrier height reach the channel by thermionic emission. On the other hand, carriers with energies below the Schottky barrier height have a probability to reach the channel according to a transmission function describing the tunnel effect which can be calculated from WKB approximation.

To overcome the complexity of WKB expression for compact modelling, an approximation based on works from [5] is applied. This effective barrier height model is described in [6]. The electron (hole) current is calculated through the Landauer equation, by integrating over energy from the dominating barrier to infinite. The dominating barrier position depends on the applied bias. In fact, the electron current can be limited by three barriers: (i) the Schottky barrier from source, (ii) the one from drain and (iii) the conduction (valence) band of the inner part.

The analytical expression of the drain current is given in [6]. In this model several other features are included. On the one hand, the band-to-band tunnelling has been developed for MOSFET-like CNTFET in [7] and has been validated through NEGF simulation. On the other hand, charges have been modeled according to the ballistic assumption and the analytical expression of charge in each region is given in [1]. The potential calculation inside the device is given in [4].

Fine-Grain Reconfigurable Architecture

The dual-gate feature of the considered devices enables the possibility to leverage the in-field polarity control. The in-field control means the ability to control the device polarity

during the operation of the system, following the design and fabrication steps. This opportunity enables a reconfiguration of the circuit at a very fine grain, which is ultimately the device-level configuration. Today's most used reconfigurable circuits are FPGAs. These regular circuits formed by several reconfigurable logic blocks called *configurable logic blocks (CLBs)*, in addition to other logic modules and reconfigurable interconnects [8]. Every CLB consists of a set of N *basic logic elements (BLEs)*. A BLE is a K -input *look-up-table (LUT)*, whose output can be possibly routed to any other LUT input through a latch. Every CLB has I inputs coming from other CLB outputs.

A standard FPGA architecture is depicted in Figure 2, showing CLBs, connected to the routing lines through *connection blocks (CBs)*. The routing blocks are connected through *switch boxes (SBs)*. We focus on the BLE design and we optimize it in the following. The standard BLE design is based on LUTs (see Figure 3(a)). The K -input LUT is a set of 2^K *static random access memory (SRAM)* cells. In our approach, the novelty is threefold. We first replace the K -input LUT by a K -input logic gate designed with DG-CNTFETs, which can be reconfigured with both gate signals. Secondly, we suggest using the input signals not only to make the calculation, but also to perform the configuration by providing the power supply signals as additional inputs, that we multiplex with the initial ones. Finally, we allow the permutation of the power supply of the logic cell, since the *pull-up (PU)* and *pull-down (PD)* networks of a DG-CNTFET can be designed with the same size [9]. The obtained novel reconfigurable cell is depicted in Figure 3(b).

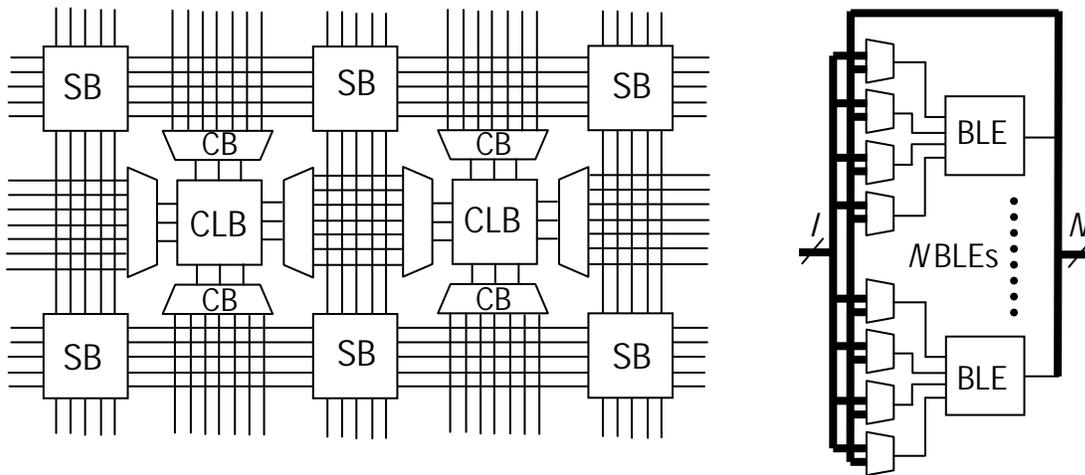


Figure 2: FPGA organization from [8]: Island-style FPGA (left), zoom-in into a CLB (right)

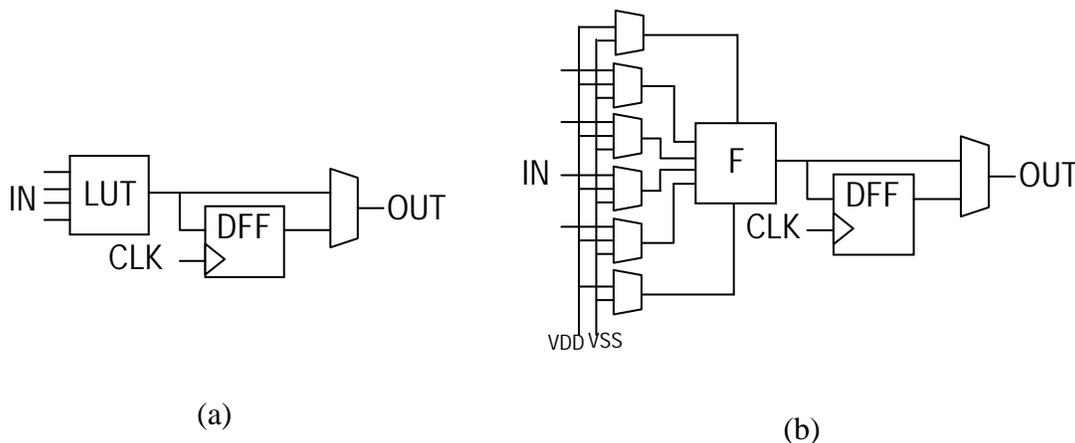


Figure 3: (a) LUT-based BLE, (b) BLE based on a reconfigurable DG-CNTFET gate

Simulation Results

We used a set of logic circuits taken from the MCNC and ISCAS'89 benchmarks [10]. We defined a gate library corresponding to the reconfigurable 4-input logic gate (Figure 4), which we used in order to synthesize the benchmark using the ABC tool [11]. It has been reported that gates similar to the used one have a high degree of reconfigurability [9]. The gate library was characterized using the presented DG-CNTFET compact model. We then performed the technology mapping with a library of 4-input LUTs ($K=4$) using ABC as well. Subsequently, we performed the logic packing of the mapped circuit into CLB with (N,I) set to (10,22) then (1,4) using T-VPACK [12]. Finally, the placement and routing were carried out using VPR [12].

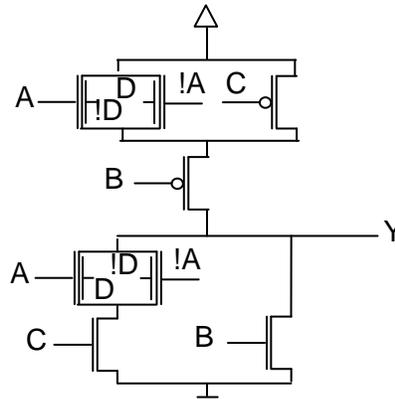


Figure 4: Reconfigurable 4-input DG-CNTFET gate

We compared the geometric average of the synthesis results over the 17 benchmarks for 3 scenarios S1 to S3: reconfigurable gate and LUT implementation at a fine granularity $(N,I) = (1,4)$ for S1 and S2 respectively and reconfigurable gate implementation with a coarse granularity $(N,I) = (10,22)$ for S3. We first note that the reconfigurable gate implements only 17 functions, compared to the 2^{16} functions implemented by the LUT. However, the logic gate efficiently implements most of the functions required for logic synthesis.

TABLE I. Implemented architectural scenarios for FPGA synthesis

Scenario	N	I	CLB area (norm.)	Intra-CLB delay (ps)	Inter-CLB delay (ps)
S1	1	4	2419	46.9	25.1
S2	1	4	2560	50.4	25.4
S3	10	22	17167	199.8	423.3

Table I shows the normalized CLB area (to unit transistors area including contacts), the delay between neighboring CLBs and the delay with the CLBs for the three scenarios simulated with the compact DG-CNTFET model. We notice that the fine-grain architectures naturally provide a smaller CLB area and a faster delay within CLBs because of the lighter input multiplexing, and a faster delay between CLBs because of the lower load of CLB in- and outputs. On the other hand, gate-based CLBs are slightly slower than the LUT-based CLBs because of the more compact BLE design, and they have a smaller area because of the smaller number of required SRAM cells, thanks to the multiplexing of reconfiguration and logic on the same input signals.

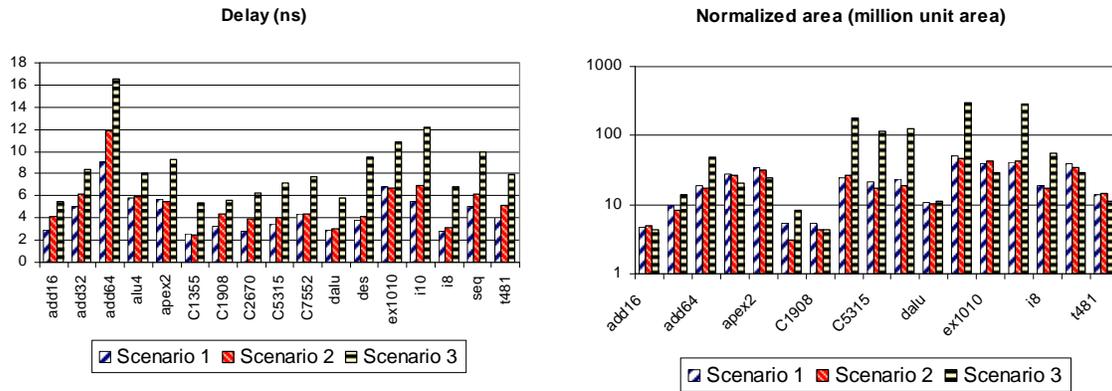


Figure 5: Delay and area comparison between the considered scenarios for different circuits

Figure 5 illustrates the delay and normalized area of the benchmark in the considered architectural scenarios. On average, the fine-grain logic implementation is 48% faster than the coarse-grain logic implementation, and it is 13% faster than the LUT implementation. The fine-grain implementation has therefore the highest performance. This comes, however, with a cost in terms of area. Because of the larger number of required CLBs, namely for pure routing, the fine-grain logic-based circuits are on average 10% larger than the LUT-based ones. On the other hand, they are 45% smaller in size than the coarse-grain logic-based counterparts because of the less inter-CLB routing resources.

Conclusions

In this paper, we introduced a novel family of DG-CNTFETs, which shows the unique property of in-field controllability. Based on a compact physical model of the considered devices, we characterized logic gates, which have been specifically chosen because of their efficient reconfigurability. We enhanced the FPGA architecture with those properties and we mapped a benchmark of logic functions in different architectural scenarios. The approach demonstrated its efficiency with fine-grain architectures, showing a delay reduction up to 48% with respect to coarse-grain architectures and 13% with respect to state-of-the-art LUT-based architectures. While the fine-grain architecture has a cost in terms of area of 10% with respect to the LUT implementation, it is 45% smaller in size than the coarse-size architecture.

References

- 1 Y. Lin et al, *IEEE Trans. on Nanotechnology*, **4**, 481 (2005).
- 2 M. H. Ben Jamaa et al., *Proc. of DATE Conference* (2009).
- 3 I. O'Connor et al., *IEEE Trans. Circuits and Systems I*, **54**, 2365 (2007).
- 4 M. Najari et al, *IEEE Trans. on Electron Devices*, **58** (1), 206 (2011).
- 5 J. Knoch et J. Appenzeller, *Physica Status Solidi (A) Applications and Materials*, **205**, 679 (2008).
- 6 S. Frégonèse et al, *IEEE Trans. on Electron Devices*, **58** (1), 206 (2011).
- 7 S. Frégonèse et al, *IEEE Trans. on Electron Devices*, **56**, 2224 (2009).
- 8 V. Betz et al., "Architecture and CAD for Deep-Submicron FPGAs", Kluwer Academic Publishers, New York (1999).
- 9 M. De Marchi et al., *Proc. of International Symposium on Nanoscale Architectures*, 65 (2010).
- 10 BLIF circuit benchmarks, <http://cadlab.cs.ucla.edu/~kirill/>
- 11 ABC: Berkeley logic synthesis tool, <http://www.eecs.berkeley.edu/~alanmi/abc/>
- 12 Versatile packing, placement and routing tool for FPGA, <http://www.eecg.utoronto.ca/vpr/>