Graphene Nanoribbon Based Complementary Logic Gates and Circuits

Jiang, Yande; Cucu Laurenciu, Nicoleta; Wang, He; Cotofana, Sorin Dan

DOI
10.1109/TNANO.2019.2903480

Publication date
2019

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Nanotechnology

Graphene Nanoribbon Based Complementary Logic Gates and Circuits

Yande Jiang, *Student Member, IEEE*, Nicoleta Cucu Laurenciu, *Member, IEEE*, He Wang, *Student Member, IEEE*, and Sorin Dan Cotofana, *Fellow, IEEE*

***Abstract*—As CMOS feature size is reaching atomic dimensions, unjustifiable static power, reliability, and economic implications are exacerbating, thereby prompting for conducting research on new materials, devices, and/or computation paradigms. Within this context, graphene nanoribbons (GNRs), owing to graphene's excellent electronic properties, may serve as basic structures for carbon-based nanoelectronics. In this paper, we make use of the fact that GNR behavior can be modulated via top/back gate contacts to mimic a given functionality and combine complementary GNRs for constructing Boolean gates. We first introduce a generic gate structure composed of a pull-up GNR performing the gate Boolean function and a pull-down GNR performing its complement. Then, we seek GNR dimensions and gate topologies required for the design of 1-, 2-, and 3-input graphene-based Boolean gates, validate the proposed gates by means of SPICE simulation, which makes use of a non-equilibrium Green's function Landauer formalism based Verilog-A model to calculate GNR conductance, and evaluate their performance with respect to propagation delay, power consumption, and active area footprint. Simulation results indicate that, when compared with 7 nm FinFET CMOS counterparts, the proposed gates exhibit 6× to 2 orders of magnitude smaller propagation delay, 2 to 3 orders of magnitude lower power consumption, and necessitate 2 orders of magnitude smaller active area footprint. We further present full adder (FA) and SRAM cell GNR designs, as they are currently fundamental components for the construction of any computation system. For an effective FA implementation, we introduce a 3-input MAJORITY gate, which apart of being able to directly compute FA's carry-out is an essential element in the implementation of error correcting codes codecs, which outperforms the CMOS equivalent carry-out calculation circuit by 2 and 3 orders of magnitude in terms of delay and power consumption, respectively, while requiring 2 orders of magnitude less area. The proposed FA exhibits 6.2× smaller delay, 3 orders of magnitude less power consumption, while requiring 2 orders of magnitude less area, when compared with the 7 nm FinFET CMOS counterpart. However, because of the effective carry-out circuitry, a GNR-based n-bit ripple carry adder, whose performance is linear in the carry-out path, will be 108× faster than an equivalent CMOS implementation. The GNR-based SRAM cell provides a slightly better resilience to dc-noise characteristics, while performance-wise has a 3.6× smaller delay, consumes 2 orders of magnitude less power, and requires 1 order of magnitude less area than the CMOS equivalent. These results clearly indicate that the proposed GNR-based approach is opening a promising avenue toward future competitive carbon-based nanoelectronics.**

***Index Terms*—Graphene, GNR, Graphene-based Boolean Gates, Carbon-Nanoelectronics.**

The authors are with the Department of Quantum and Computer Engineering, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft 2628CD, The Netherlands (e-mail: yande.jiang@tudelft.nl; n.cuculaurenciu@tudelft.nl; h.wang-13@tudelft.nl; s.d.cotofana@tudelft.nl).

I. Introduction

As DENNARD scaling nears the limit of atomic feature size, with in high power density and leakage, low reliability and yield, and increased IC production costs, new materials, structures, and computation paradigms are called upon [1], [2]. Graphene is one of the post Silicon front runners, which has enjoyed a surge of research popularity in the last decade, opening the way for a wide range of graphene-based applications, e.g., spintronics, photonics and optoelectronics, sensors, energy storage and conversion, flexible electronics, and biomedical applications [3]–[6].

Graphene consists of a single layer of carbon atoms arranged in a honeycomb lattice, and has a set of unique, remarkable properties, among which room temperature electron mobility 10× higher than Si, high thermal conductivity, thinness, and ballistic carrier transport [7]–[9]. Such properties provide a strong drive for investigating graphene as a potent contender to Si and to pursue avenues for carbon-based nanoelectronics [10]–[12]. Generally speaking, the main hindrances to graphene-based logic circuitry are design and manufacturing related [13], [14].

From the perspective of manufacturing, the principal ambition is to find a cost-effective, scalable and reliable manufacturing process, which allows mass-production with minimum defect density and highly reproducible features. Over the past few years, graphene researchers focused on Graphene Nanoribbon (GNR) fabrication and several approaches have been proposed to produce GNRs, such as top-down lithographic patterning [15], [16], chemical procedures [17], and longitudinally unzipping of high quality grown carbon nanotubes [18], [19]. A fast and inexpensive approach to fabricate GNRs as narrow as 9 nm with an ON/OFF current ratio of 70 at room temperature and carrier mobility of $300\,\mathrm{cm}^2 v^{-1} s^{-1}$ is presented in [20] and a surface-assisted synthesis method to produce atomically precise, low-edge-defect GNRs, e.g., 3-Armchair GNRs (1 hexagon width) and 6-Zigzag GNRs (6 hexagon width) is described in [21]. Such developments clearly indicate that GNR structures with various dimensions and geometries can be potentially fabricated in the close future.

From the design standpoint, there are several impediments to graphene-based Boolean logic that need consideration: (i) how to control the conductivity in order to obtain "on" and "off" states that are distinguishable, while not compromising the intrinsic highly advantageous properties of graphene (e.g., high carrier mobility), (ii) how to encode a specific Boolean logic transfer function onto electrical properties of graphene (e.g., conduction maps), (iii) how to find the appropriate external electrical means (e.g., top gates, back gate) that enable the graphene behaviour control and that can induce a specific logic functionality, (iv) how to make sure that digital circuits can be cascaded, i.e., clean and compatible/matching gate inputs and outputs electric levels, (v) understanding how the GNRs interact with each other when they are interconnected, and (vi) how to combine GNRs and construct graphene-based gates/circuits.

Past work in [22] proved that when a trapezoidal Quantum Point Contact (QPC) topology [23] is being augmented with top gates, and when its GNR geometry is changed, the GNR conductance can be modulated via external voltages such as top gate and back gate voltages, so that Boolean logic functions behaviour is being mirrored. This structure addresses the issue outlined in (i)–(iii), and constitutes a basic ingredient for Boolean gates construction. However, multiple aspects still need to be taken into consideration, chiefly, the manner to obtain Boolean gates which have clean and compatible primary inputs and outputs voltage levels by shaping and combining various GNRs.

In this paper, we address the (iv)-(vi) issues resulted from the electrical interaction of GNRs, which will enable the construction of graphene-based Boolean gates and circuits. For this purpose, we make use of the methodology for designing Boolean gates by means of two complementary GNRs, i.e., a pull-up GNR performing the targeted Boolean function and a pull-down GNR performing its inverse, introduced in [24]. The GNR structures have a conduction channel made of a graphene zigzag ribbon, which is situated between the drain and source contacts. The gate primary inputs voltages are applied via one/two top gate/s. Since each gate necessitates GNRs with a desired behaviour (e.g., conductance) which corresponds to the Boolean function that they mimic, we identify topologies which are able to yield the behaviour of each basic function, i.e., AND, NAND, OR, NOR, XOR, XNOR, INV, and BUFF. For this purpose, we conduct a design space exploration with regard to the GNR shape and its dimensions, and the top gates contacts topology, while abiding to particular constraints (e.g., gate output voltage values which are compatible with gate input voltage values, high ratio between the high and low conductance values of the GNR).

The proposed 1-, 2-, and 3-input GNR gates are validated in Cadence by means of SPICE simulation which employs a Verilog-A model, that calls internally a Simulink model in order to compute the GNR conductance using the Non-Equilibrium Green's Function (NEGF)-Landauer formalism [23], [25], [26]. To gain insight into the potential of our proposal, we evaluate the GNR gates with respect to delay, active area footprint, and power consumption, relative to the 7 nm FinFET CMOS [27] counterparts. Our results indicate that proposed 1-, 2- and 3-input graphene gates outperform 7 nm FinFET CMOS counterparts as follows: (i) they provide up to 6$\times$ and 2 orders of magnitude smaller propagation delay, (ii) they consumes 2 and 3 orders of magnitude lower power, and (iii) they require 2 orders of magnitude smaller active area footprint, respectively. We observe that, contrary to CMOS designs, the proposed GNR-based gates can yield effective power-delay trade-offs, at approximately the same area. This is because the graphene conductance main contributor is the nanoribbon geometry and its overall topology, rather than the effective area. Furthermore, the required active area is not proportional with gate's function complexity and fan-in, e.g., XOR and INV have similar footprints, which results in more compact circuit layout.

We further present GNR based designs of 1-bit Full Adder (FA) and SRAM cell, as they currently constitute the foundation for the construction of any computation system. For an effective FA implementation we design a 3-input MAJORITY gate, which apart of being able to directly compute FA's Carry-Out is an essential element in the implementation of Error Correcting Codes (ECC) decoders, that outperforms the CMOS equivalent Carry-Out calculation circuit by 2 and 3 orders of magnitude in terms of delay (0.109 ps vs 11.863 ps) and power consumption, respectively, while requiring 2 orders of magnitude less area. The proposed FA design exhibits 6.2$\times$ smaller delay, 3 orders of magnitude less power consumption, while requiring 2 orders of magnitude less area, when compared with the 7 nm FinFET CMOS counterpart. By consequence, a GNR-based n-bit Ripple Carry Adder, which performance is linear in the Carry-Out path, will be 108$\times$ faster than a CMOS implementation. The GNR based SRAM cell provides a slightly better resilience to DC noise characteristics, while performance-wise has a 3.6$\times$ smaller delay, consumes 2 orders of magnitude less power, and requires 1 order of magnitude less area than the CMOS equivalent.

The rest of this paper has the followig structure: Section II presents the proposed 1-, 2-, and 3-input GNR-based Boolean gates and their correspondant design methodology. Section III describes the simulation framework. Section IV and V presents, evaluates, and compares the proposed designs with state of the art CMOS equivalents. Finally, the paper ends with some concluding remarks in Section VI.

II. Complementary GNR Pair-Based Boolean Gates

Subsequently we describe the design methodology we employed fot the proposed GNR-based Boolean gates and we present the rationale behind the gates complementary construction.

We begin by noticing that there are 2 fundamental elements towards graphene-based circuits: (i) opening the graphene energy bandgap in order to switch off effectively the current, and (ii) finding how to control GNR conductance and how to enact the appropriate electrical response corresponding to a particular Boolean function. For this purpose, as GNR research vehicle to be build upon, we use a trapezoidal graphene Quantum Point Contact (QPC) which has zigzag shaped edges [23]. The GNR can be utilized as conduction channel between the source and drain contacts, which are biased by a voltage $V_d - V_s$. The bandgap opening problem can be solved to a certain extent, by

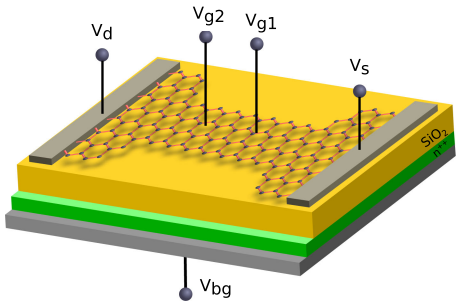

Fig. 1. Boolean gate graphene-based building block.

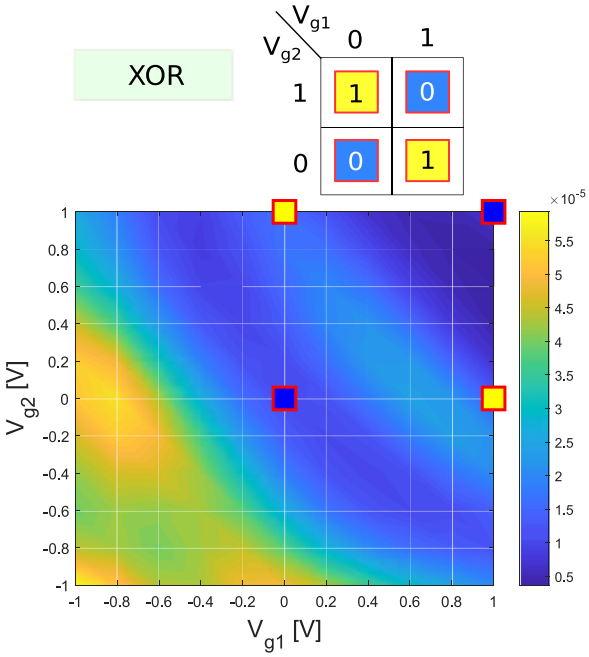

Fig. 2. 2-input XOR conductance map.

carving the GNR geometry. When carving the GNR gometry and adding top and back gates with various topologies, we can modulate the graphene conductance (via voltages externally applied on the GNR top gates), such that it mirrors a particular intended Boolean function. In Figure 1 is illustrated the main ingredient employed for the construction of GNR-based Boolean gates, i.e., a GNR structure which is augmented with 1 back gate and 2 top gates contacts. Figure 2 shows for example, the conductance map that we obtained for a GNR whose geometry was optimized in such a way that it is able to reflect the functionality of the Boolean XOR operator, with 0 V and 1 V associated to logic low and logic high voltage levels, respectively.

Subsequently, building upon the structure presented in Figure 1, we propose GNR-based complementary Boolean gates. For this purpose, we construct each gate using 2 GNR basic building structures, as depicted in Figure 3: a pull-down GNR, denoted as GNR_{dn}, which has its source terminal connected to the ground V_{SS}, and a pull-up GNR, denoted subsequently as GNR_{up}, which has its drain contact connected to the supply voltage V_{DD}.

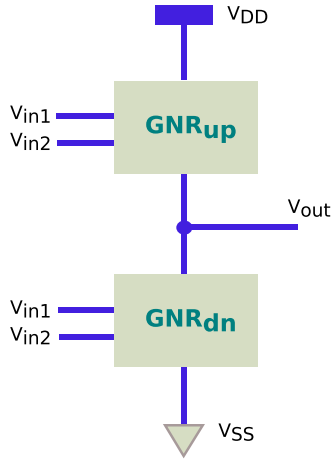

Fig. 3. GNR Boolean gate with complementary GNRs.

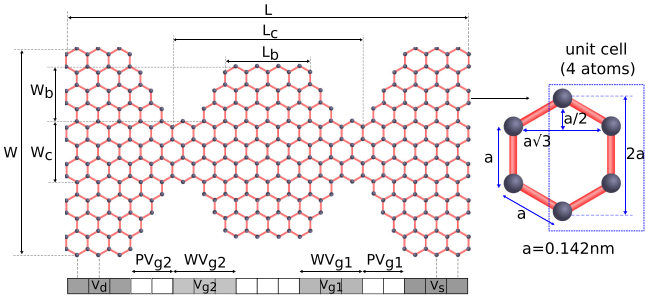

Fig. 4. GNR topology description parameters.

The pull-up and the pull-down GNRs perform complementary functions, e.g., a NAND gate is composed of a GNR_{up} which mirrors onto its conductance the NAND logical functionality, and of a GNR_{dn} whose conductance maps the AND logical functionality.

In order to obtain the suitable GNRs for every gate, we conduct a design space exploration, by changing a set of parameters, as defined in Figure 4: (i) nanoribbon geometry (i.e., width W and length L, constriction width W_c and length L_c, and extrusion top length L_b and width W_b), and (ii) the topology of the top gate contacts (i.e., contacts width W_{V_g} and their position relative to the drain and source contacts P_{V_g}).

The primary output voltage level of the gate illustrated in Figure 3, can be approximated as:

$$V_{\text{out}} = V_{DD} \cdot \frac{G_{up}}{G_{dn} + G_{up}}, \tag{1}$$

where G_{dn} and G_{up} are the conductances of the pull-down and pull-up GNR, respectively. Therefore, multiple aspects require to be taken into consideration as part of the design space exploration process, among which:

- A high ratio between the conductances of the pull-up and pull-down GNRs is the main contributor for achieving gate output voltages which are closer to the power supply and ground rails, as well as low leakage power. In particular, when the gate output voltage ought to pull-up to V_{DD},

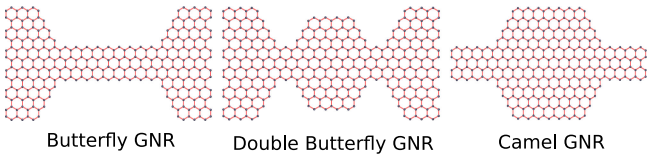

Fig. 5. GNR shapes for Boolean gates.

the ratio G_{up}/G_{dn} ought to be at least > 10, for obtaining $V_{\text{out}} \geq 91\% \cdot V_{DD}$. Conversely, when the gate voltage ought to pull-down to V_{SS}, the ratio G_{up}/G_{dn} ought to be less than $1/10$, for obtaining $V_{\text{out}} \leq 9.1\% \cdot V_{DD}$.

- To avoid spurious transients in the gate output voltage, the conductance which is modulated via the gate input voltages shouldn't manifest non-linearities.
- Conductance values which can enable a reasonable input to output propagation delay and power trade-off are preferable.
- Balanced output switching delay (i.e., "0" $\rightarrow$ "1" delay which resembles "1" $\rightarrow$ "0" delay).

By means of the design space exploration, we exposed 3 types of GNR shapes, which are depicted in Figure 5 and found to be the most suitable for the construction of GNR Boolean gates construction. Further, in Section IV we prove that by appropriately changing the dimensions of the GNR shapes, they deliver the necessary functionalities for constructing all the desired Boolean gates.

We observe that the GNR gates can be directly cascaded to construct networks of GNR gate to enable GNR-based circuit design, since the GNRs input voltages are compatible with their output voltages. Nevertheless, akin to the CMOS case, certain circuit topologies may result in signal integrity degradations, and in these situations, buffers, such as the one from Section IV, are necessary for restoring the logic high and logic low voltage levels.

III. Simulation Setup

In this section, we describe the formalism for deriving the electrical properties of GNRs and present the SPICE simulation setup of proposed GNR-based Boolean gates.

A. GNR Conduction Computation

To derive GNR conduction under certain bias condition we model graphene electronic ballistic transport by means of the Non-Equilibrium Green's Function (NEGF)-Landauer formalism. The GNR channel is described by a Hamiltonian matrix $H = H_0 + U$, which models the interactions between neighbor carbon atoms (via H_0), and incorporates all the external as well as the internal potentials (e.g., top gates voltages, and back gate voltage) via U. H_0 is constructed by using semi-empirical (tight-binding) calculations, as follows:

$$H_0 = \sum_{i,j} t_{i,j} \left|i\right\rangle \left\langle j\right|, \tag{2}$$

$$\text{where } t_{i,j} = \begin{cases} \tau, & \text{if atoms } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In our simulation we account for first nearest-neighbor (1NN) interactions, with hopping energy between atoms $\tau = -2.7$eV. The potential distribution matrix U is determined self-consistently as the solution of the 2D Poisson equation

$$\nabla \cdot [\epsilon(\mathbf{r})\ \nabla U(\mathbf{r})] = -\frac{\rho(\mathbf{r})}{\epsilon_0}, \tag{4}$$

where $\mathbf{r} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}}$ is a position vector in space, ϵ_0 denotes the vacuum permittivity, $\epsilon(\mathbf{r})$ is the dielectric permittivity of the materials at position $\mathbf{r}$, and ρ represents the net charge density distribution. The Poisson equation is numerically solved by making use of the finite difference method. On the two end sides of the channel, reside the drain and source contacts which have different electrochemical potentials that sustain the conduction in the channel. The interactions between the two contacts and the channel are modelled via the contact self-energy matrices Σ_1 and Σ_2, respectively. Once H and $\Sigma_{1,2}$ are computed, the transmission function $T(E)$, which models the probability of transmission of one electron from the drain to the source contact, is derived as a function of energy using:

$$T(E) = \text{Trace}\left[\Gamma_1\ G_R\ \Gamma_2\ G_R^{\dagger}\right] \tag{5}$$

where

$$G_R(E) = [EI - H - \Sigma_1 - \Sigma_2]^{-1}$$

$$\Gamma_{1,2} = i[\Sigma_{1,2} - \Sigma_{1,2}^{\dagger}].$$

The channel current is obtained using the Landauer formula:

$$I = \frac{q}{h}\int_{-\infty}^{+\infty} T(E) \cdot (f_0(E-\mu_1) - f_0(E-\mu_2))\ \mathrm{d}E, \tag{6}$$

where $f_0(E)$ is the Fermi-Dirac distribution function at temperature T, and $\mu_{1,2}$ denote the Fermi energy of the source and drain contacts. The conductance can then be written as:

$$G = \frac{I}{V_d - V_s}. \tag{7}$$

B. Mixed SPICE-Simulink Simulation

In order to validate the correct operation and evaluate the proposed GNR-based Boolean gates, we utilize SPICE simulation in Cadence [28]. The GNR of each gate is modeled using a Verilog-A model which has 5 pins (out of which 2 inout pins: source and drain, and 3 input pins: 2 top gates and 1 back gate) [29]. In order to permit multiple GNR shapes and gate topologies, we developed a parametric Verilog-A model which is able to take into account: the nanoribbon length L and width W, the constriction length L_c and width W_c, the extrusion top length L_b and width W_b, the position of the top gate contacts relative to the source/drain contacts $P_{V_{g1,2}}$, and the top gate contact widths $W_{V_{g1,2}}$, as defined in Figure 4. The Verilog-A model triggers internally a Simulink model which computes the GNR conductance as described in Section III-A. In this way, we benefit of pysics level, accurate results. The inter-communication

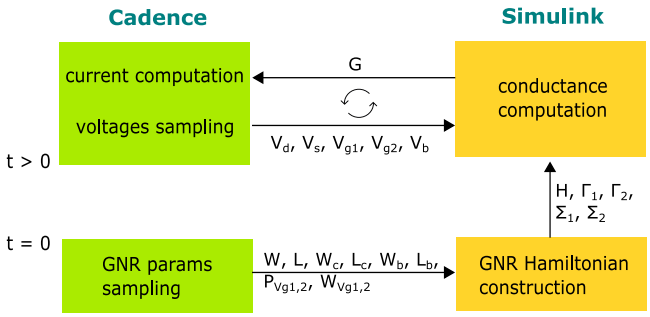

Fig. 6. Cadence-Simulink GNR Simulation Flow.

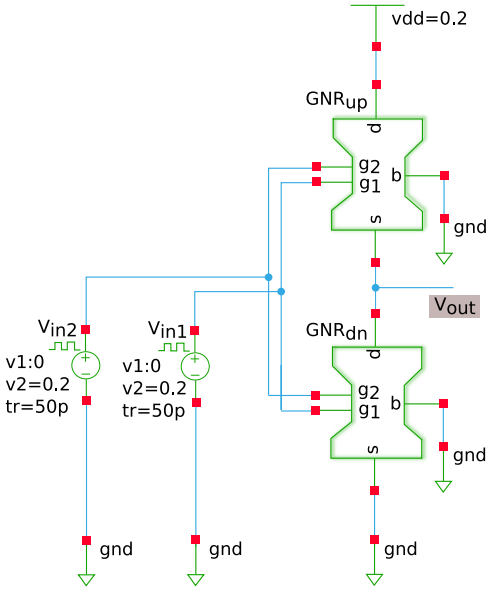

Fig. 7. GNR 2-input Gate SPICE Simulation Setup.

between Simulink [30] and Cadence is illustrated schematically in Figure 6. Based on the GNR geometry (described using the 10 parameters depicted in Figure 6), at the initial time step $t = 0$, Simulink computes the Hamiltonian matrix H, the source and drain contacts self-energy, Σ_1 and Σ_2, and their energy broadening factors, Γ_1 and Γ_2. Then, for every remaining transient simulation time step, Simulink receives from Cadence 5 inputs (V_d, V_s, V_{g1}, V_{g2}, and V_b voltages), and based on the matrices computed during the initial time step $t = 0$, it derives the afferent GNR conductance G and then passes this value back to Cadence. Once the conductance value is known to the Verilog-A model, the current through the GNR is updated via the relation: $I(d, s) = V(d, s) \cdot G$.

C. GNR Gates Simulation

Individual GNR gates are simulated in Cadence using a generic setup which is illustrated in Figure 7 for 2-input gates with the back gate voltage set to 0 V.

For each gate, there are two GNRs which are connected in series. We employ 0 V as logic low voltage level, and 0.2 V as logic high voltage level. For 1-input gates, i.e., inverter and buffer, the $g2$ pin is absent, while for 3-input gates a third pin denoted as $g3$ is added. The gate primary input voltages are periodic pulse signals with 50 ps rise time and fall time and 50% duty cycle. The primary input signals period is set to 400 ps for 1-input gates, 400 ps and 800 ps for 2-input gates, and 400 ps, 800 ps, and 1600 ps for 3-input gates.

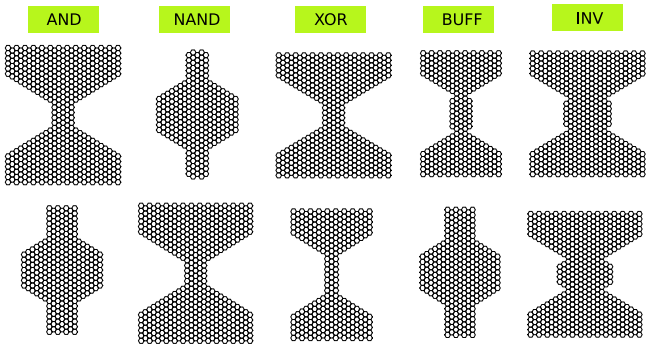

Fig. 8. GNR_{up} (top row) and GNR_{dn} (bottom row).

TABLE I
COMPLEMENTARY BOOLEAN GATE GNR DIMENSIONS

		(W, L)	(W_c, L_c)	(W_b, L_b)	(P_{V_g}, W_{V_g})
AND	GNR_{up}	$(41, 27\sqrt{3})$	$(8, 4\sqrt{3})$	$(0, 0)$	$(2\sqrt{3}, 6\sqrt{3})$
	GNR_{dn}	$(29, 25\sqrt{3})$	$(0, 0)$	$(9, 7\sqrt{3})$	$(6\sqrt{3}, 3\sqrt{3})$
NAND	GNR_{up}	$(29, 25\sqrt{3})$	$(0, 0)$	$(11, 7\sqrt{3})$	$(6\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(8, 4\sqrt{3})$	$(0, 0)$	$(2\sqrt{3}, 6\sqrt{3})$
XOR	GNR_{up}	$(41, 25\sqrt{3})$	$(8, 4\sqrt{3})$	$(0, 0)$	$(1\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(29, 25\sqrt{3})$	$(5, 7\sqrt{3})$	$(0, 0)$	$(6\sqrt{3}, 3\sqrt{3})$
BUFF	GNR_{up}	$(29, 25\sqrt{3})$	$(5, 7\sqrt{3})$	$(2, 6\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(29, 25\sqrt{3})$	$(0, 0)$	$(9, 7\sqrt{3})$	$(0, 6\sqrt{3})$
INV	GNR_{up}	$(41, 25\sqrt{3})$	$(14, 6\sqrt{3})$	$(2, 5\sqrt{3})$	$(6\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 25\sqrt{3})$	$(14, 6\sqrt{3})$	$(3, 4\sqrt{3})$	$(6\sqrt{3}, 3\sqrt{3})$

IV. GNR BOOLEAN GATES

In this section we propose and investigate the performance of 1-, 2-, and 3-input GNR basic Boolean gates.

A. 1- and 2-Input GNR Gates

We present in the following for every proposed GNR gate, its topology, and we validate in SPICE the gates correct operation. Further we evaluate them with respect to propagation delay, area, and power consumption, against 7 nm FinFET CMOS counterparts.

Figure 8 graphically illustrates the GNR shapes utilized in the proposed 1- and 2-input gates. We observe that, intuitively speaking, GNR_{up} and GNR_{dn} can be interchanged as part of two gates which perform inverse Boolean functions, i.e., we can use the same 2 GNRs for both AND gate and NAND gate per se.

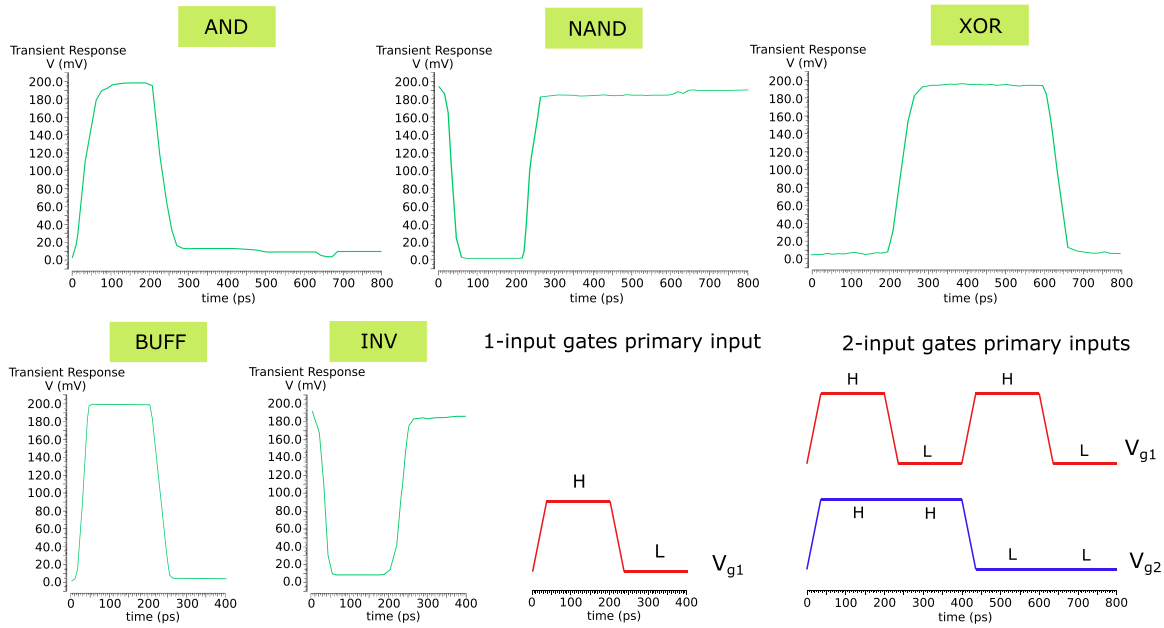

Fig. 9. GNR Gate SPICE Simulation Results.

Nevertheless, as the pull-up GNR is connected to V_{DD} and V_{out}, when connecting it as pull-down GNR to V_{out} and V_{SS} for the inverse gate, its conductance map might deviate from expected behaviour (might not properly mirror the same Boolean function). Therefore, for gates which perform complementary functions, it is necessary to use different GNRs even if they execute the same Boolean function. The geometry and contacts topology of the proposed GNR gates, which are optimized for an operating voltage of 0.2 V, are summarized in Table I. We note that all dimensions in the table are expressed in terms of $a = 0.142$ nm, the distance between adjacent carbon atoms, which holds true for all the other reporting GNR geometries tables included in the paper. GNR gate designs which operate on other power supply voltage values (e.g., varying from mV to V) are feasible and lead to varying delay-power-area tradeoffs, but they require identifying GNR topologies which are capable of delivering the intended functionality under the new biasing setup. In general, the delay and robustness requirements constrain the power supply voltage, but our choice for 0.2 V is motivated primarily by the fact that we wanted to probe the delay and power potential of graphene logic while maintaining the GNR dimensions within a feasible range.

The GNRs of all gates have similar total width and length, but they have different extrusion and constriction dimensions. The extrusion and constriction width impact on the conductance is big, which is not true for the influence of their length dimension. Therefore, as it can be inferred from Table I, the extrusion and constriction width parameters can vary significantly among GNRs which correspond to different Boolean functions. Also, it can be observed that the top gates contacts are situated closer to the source/drain contacts for the GNRs which map {AND, XOR, BUFF} Boolean operations, and further for {NAND, INV}. The width of the top gate contacts remains the same for all GNRs with 2 exceptions - the GNRs which map {AND, INV}.

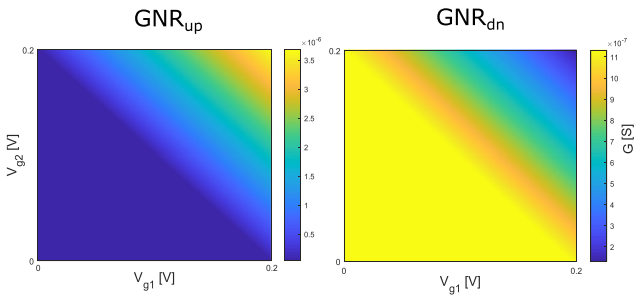

Fig. 10. AND gate GNR conductance maps.

For illustrating the complementary operation of proposed GNR-based Boolean gates, we consider the AND gate, and present the 2 conductance maps in Figure 10 which correspond to the gate pull-up and pull-down GNRs. As we can notice, the 4 corner conductance points are mirroring the logical AND functionality for GNR_{up}, and the inverted function (NAND) for GNR_{dn}. The 2 density plots are also pointing out the proposed gate robustness to voltage variations of the gate input. For example, $\approx 5\%$ variation of the input voltages results in $\approx 4.9\%$ and $\approx 4.4\%$ variation of GNR_{up} and GNR_{dn} conductance, respectively.

Figure 9 shows the {INV, BUFF, AND, NAND, XOR} GNR gates input voltages and their response. We observe that all gates exhibit correct operation in line with the corresponding Boolean function. The presence of small spikes can be observed on the output voltage evolution. We attribute these spurious transients on one hand to the feedback currents of the input voltage sources, and on the other hand to non-linearities which are present in the

TABLE II
1- AND 2-INPUT GATES DELAY, AREA, AND POWER

	$\tau_p[ps]$		Active Area $[nm^2]$		Total Power $[nW]$	
	GNR	CMOS	GNR	CMOS	GNR	CMOS
AND	1.38	9.618	$4.272 \cdot 10^1$	$1.452 \cdot 10^3$	4.628	$5.886 \cdot 10^2$
NAND	2.15	7.556	$4.146 \cdot 10^1$	$9.680 \cdot 10^2$	2.370	$5.415 \cdot 10^2$
XOR	7.48	9.168	$4.038 \cdot 10^1$	$2.420 \cdot 10^3$	1.734	$5.923 \cdot 10^2$
BUFF	0.42	2.040	$3.283 \cdot 10^1$	$9.680 \cdot 10^2$	0.937	$4.704 \cdot 10^2$
INV	0.27	1.110	$5.431 \cdot 10^1$	$4.840 \cdot 10^2$	0.947	$4.621 \cdot 10^2$

dependence of the GNR conductance on the voltages to which the GNR is subjected.

Table II summarizes the input to output propagation delay, the active area requirements, and the power consumption for the proposed gates and for 7 nm FinFET CMOS [27] ($V_{DD} = 0.7$ V) counterparts. For a fair area-wise comparison, we only consider the conduction channels area of the encompassed devices, instead of the total standard cell footprint (which is not available for GNR gates). As far as the power is concerned, we measure in SPICE the total power for all 4 clock cycles. The tabulated results show a propagation delay reduction for the GNR gates, relative to the CMOS counterparts, which varies from 23% for the XOR gate, up to 6× for the AND gate, and 2 orders of magnitude lower power consumption in all cases. Moreover, the GNR gates necessitate 1 to 2 orders of magnitude smaller active area footprint than the most advanced CMOS technology node [31] counterparts. While for the {AND, NAND, XOR} CMOS gates, the propagation delay and power figures are similar, we observe that this is not the case for the GNR gates. For instance, the GNR AND delay is 4.4× smaller than the GNR XOR gate delay. However, the GNR AND power consumption is 1.6× higher than that of the one of the GNR XOR. This is a direct consequence of our design choice towards a fast AND gate at the expense of increased power consumption. However, when designing the GNR gates one may opt for other trade-offs.

In Table II it can be noticed that the active area of different CMOS gates can vary by up to 4×, while in the case of the GNR gates the variation resides within 65%. Therefore, we can arrive at the conclusion that while, generally speaking, complex Boolean logic translates into a larger CMOS circuit area realization, this is not the case for GNR, where a complex Boolean functionality can be achieved with very little area implications. For example, if we consider the XOR gate relative to the NAND gate, the occupied area for the CMOS case increases by 1.5×, while the area is similar (2.7% reduction) for the GNR case.

B. 3-Input GNR Gates

As higher than 2 gate fan-in might be of interest in practical implementations, in this section, we seek GNR topologies appropriate for the implementation of 3-input gates, namely {AND3, NAND3, OR3, NOR3} and investigate the characteristics of the obtained GNR gates. Note that besides those we also propose 3-input XOR and MAJORITY gates but we discuss them in the more relevant context of the Full Adder implementation presented in Section V-A.

TABLE III
3-INPUT GNR GATE DIMENSIONS

		(W, L)	(W_c, L_c)	(W_b, L_b)	(P_{V_g}, W_{V_g})
AND3	GNR_{up}	$(41, 27\sqrt{3})$	$(8, 4\sqrt{3})$	$(0, 0)$	$(5\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(14, 6\sqrt{3})$	$(5, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
NAND3	GNR_{up}	$(35, 27\sqrt{3})$	$(0, 0)$	$(11, 5\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(14, 8\sqrt{3})$	$(2, 6\sqrt{3})$	$(5\sqrt{3}, 3\sqrt{3})$
OR3	GNR_{up}	$(35, 27\sqrt{3})$	$(0, 0)$	$(14, 11\sqrt{3})$	$(5\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(2, 2\sqrt{3})$	$(0, 0)$	$(3\sqrt{3}, 3\sqrt{3})$
NOR3	GNR_{up}	$(41, 27\sqrt{3})$	$(8, 6\sqrt{3})$	$(0, 0)$	$(5\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(2, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$

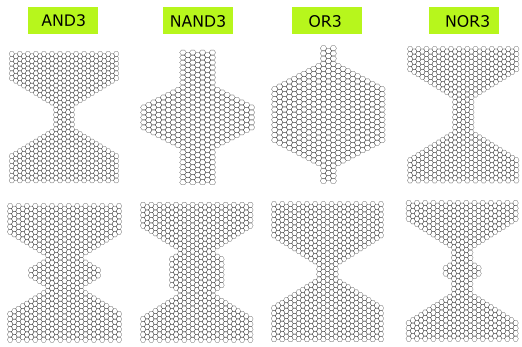

Fig. 11. 3-input Gate GNR Shapes.

TABLE IV
3-INPUT GNR BACK GATE BIAS

	V_b $[V]$			
	AND3	NAND3	OR3	NOR3
GNR_{up}	0	−0.4	0.2	−0.4
GNR_{dn}	−0.1	0	−0.1	0

The identified GNR gates dimensions and shapes are presented in Table III and Figure 11, respectively. For the 3-input gates, P_{Vg} defines the position of the first and third top gates with respect to the drain and source contacts, respectively. The second top gate is situated in the middle in-between the other two top gates. We note that for all the gates introduced in Section IV-A, topologies able to operate under the same 0 V back gate voltage bias have been sought. To implement 3-input gates, we extended the Design Space Exploration (DSE) by letting the back gate voltage also vary as other back gate voltage values can facilitate a more appropriate top gate control on the conductance, and induce a higher ON/OFF current ratio by modulating the Fermi energy level at the Dirac point. The applied back gate voltages for each identified GNR topology are presented in Table IV.

TABLE V
3-INPUT GNR GATES PROPAGATION DELAY, AREA, AND POWER VS 7 NM FINFET CMOS

	$\tau_p[ps]$		Active Area $[nm^2]$		Power $[nW]$		Power-Delay Product $[ps\text{-}nW]$	
	GNR	CMOS	GNR	CMOS	GNR	CMOS	GNR	CMOS
AND3	2.538	$1.116 \cdot 10^1$	$5.665 \cdot 10^1$	$1.936 \cdot 10^3$	6.234	$2.326 \cdot 10^2$	15.82	$4.461 \cdot 10^3$
NAND3	3.195	7.635	$4.387 \cdot 10^1$	$1.452 \cdot 10^3$	2.777	$8.701 \cdot 10^2$	8.871	$2.621 \cdot 10^3$
OR3	2.273	8.547	$5.092 \cdot 10^1$	$1.936 \cdot 10^3$	0.836	$6.472 \cdot 10^2$	1.900	$2.968 \cdot 10^3$
NOR3	2.132	$1.092 \cdot 10^1$	$4.771 \cdot 10^1$	$1.452 \cdot 10^3$	1.035	$9.868 \cdot 10^2$	2.207	$3.257 \cdot 10^3$
XOR3	1.583	$1.373 \cdot 10^1$	$5.179 \cdot 10^1$	$4.840 \cdot 10^3$	1.654	$1.768 \cdot 10^3$	2.618	$2.427 \cdot 10^4$
MAJ3	0.109	$1.099 \cdot 10^1$	$5.078 \cdot 10^1$	$2.180 \cdot 10^4$	3.388	$3.482 \cdot 10^3$	0.371	$3.826 \cdot 10^4$

TABLE VI
EXTENDED DSE DELAY, AREA, AND POWER

	$\tau_p[ps]$		Active Area $[nm^2]$		Total Power $[nW]$	
	GNR_$v1$	GNR_$v2$	GNR_$v1$	GNR_$v2$	GNR_$v1$	GNR_$v2$
XOR	7.48	0.96	40.38	51.51	1.73	1.40
INV	0.27	0.24	54.31	45.23	0.95	0.73

TABLE VII
DIMENSIONS OF GNR 1-BIT FULL ADDER GATES

		(W, L)	(W_c, L_c)	(W_b, L_b)	(P_{V_g}, W_{V_g})
XOR	GNR_{up}	$(41, 27\sqrt{3})$	$(14, 6\sqrt{3})$	$(2, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(2, 2\sqrt{3})$	$(7\sqrt{3}, 3\sqrt{3})$
XOR3	GNR_{up}	$(41, 27\sqrt{3})$	$(14, 8\sqrt{3})$	$(8, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(2, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$
MAJ3	GNR_{up}	$(41, 27\sqrt{3})$	$(14, 8\sqrt{3})$	$(2, 4\sqrt{3})$	$(5\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(14, 8\sqrt{3})$	$(2, 2\sqrt{3})$	$(3\sqrt{3}, 3\sqrt{3})$

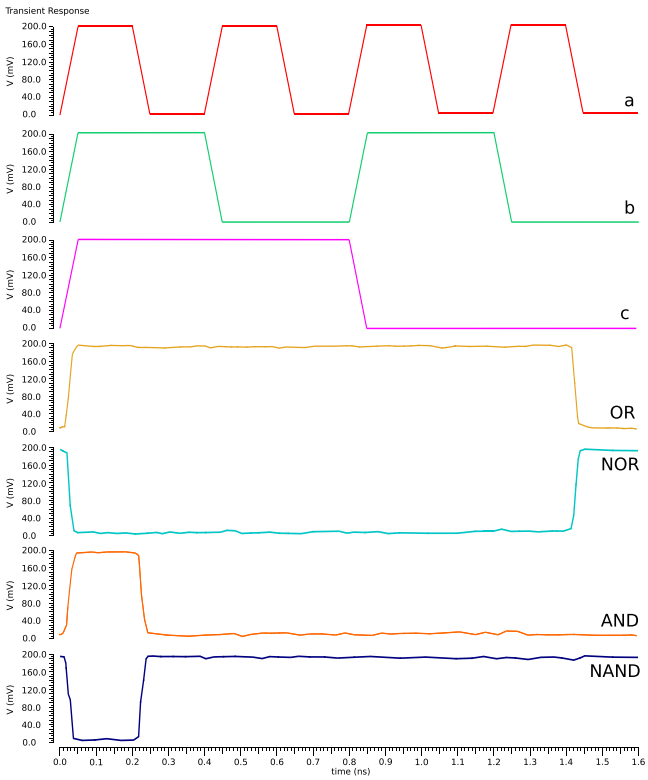

Fig. 12. 3-input gate SPICE simulation results.

As a result of this DSE extension we are also able to identify 1- and 2-input gate designs with slightly better performance than the ones proposed in Section IV-A, which is the case for the XOR and INV gates (with geometries presented in Table VII and Table X, respectively) that we employ for the Full Adder and SRAM cell designs in Section V. Table VI reflects the due to non-zero back bias ($V_b = 0$ DSE (_$v1$) vs $V_b \neq 0$ extended DSE (_$v2$)) performance improvement for these 2 gates and indicates 7× and 11% delay reduction, 19% and 23% lower power consumption, for the XOR and INV gate, respectively, while requiring roughly the same active area. We note that the 1- and 2-input gates proposed in Section IV-A were optimized for low power thus by setting a high performance focus for the design space exploration, we can potentially obtain GNR topologies that reduce the gate delay by at least one order of magnitude.

We performed SPICE simulation and validated the 3-input gates correct functionality as indicated by the plots in Figure 12.

Table V summarizes the delay, area, and power consumption for 3-input GNR gates (we also included the MAJORITY gate MAJ3 introduced in Section V for sake of completeness) and CMOS counterparts. We observe that the 3-input GNR gates provide propagation delay, power consumption, and power-delay product reductions of 2× and 2 orders of magnitude, 2 and 3 orders of magnitude, and 2 and 5 orders of magnitude, respectively, for NAND3 and MAJ3, respectively, while requiring about 1 to 2 orders of magnitude smaller active area. We observe that both 2-input and 3-input GNR gates occupy roughly the same area. This implies that we can increase the gate fan-in and the gate functional complexity with little to no impact on the active area footprint. The same cannot be said about CMOS where the area generally increases with the gate complexity increase. Also, the fact that we can implement a 3-input MAJORITY gate with 2 GNRs is quite significant and has positive implications on other implementations, e.g., Error Correcting Codes (ECC) codecs, LDPC [32]–[35], which performance heavily depends on the effectiveness of the utilized MAJORITY gate implementations.

All these results suggest that, potentially speaking, GNR-based logic gates can substantially outperform advanced CMOS

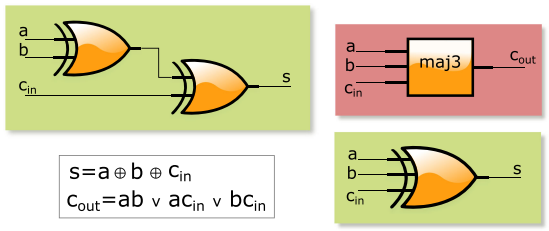

Fig. 13. 1-bit Full Adder.

TABLE VIII
FA GATES GNR BACK GATE BIAS

	XOR		XOR3		MAJ3	
	GNR_{up}	GNR_{dn}	GNR_{up}	GNR_{dn}	GNR_{up}	GNR_{dn}
V_b [V]	0	−0.1	0.1	0	0	−0.1

counterparts and can open a novel avenue towards future post-Si nanoelectronics. To get a glimpse on the possible implications of our proposal on potential carbon based computing platform performance we propose in the next section GNR based implementations of two fundamental computing and storage circuit elements.

V. BASIC GNR CIRCUITS

In this Section, we make use of proposed GNR gates to design the most frequently utilised computation and storage elements, i.e., the Full Adder and the SRAM cell.

A. 1-Bit Full Adder

As adders are the most ubiquitous basic building blocks of any computing system, we consider a 1-bit Full Adder, with 3 1-bit inputs (a, b, Carry-In), and 2 1-bit outputs (Sum, Carry-Out) and evaluate and compare different GNR and 7 nm FinFET CMOS implementations. For the CMOS case we use the optimized 28 transistors standard cell. For the GNR case, as illustrated in Figure 13, we make use of a single 3-input MAJORITY gate, realised with 2 GNRs only, for computing the Carry-Out output since it is faster, smaller, and consumes less power than any counterpart designs relying on multiple 2-input gates (e.g., 6 NAND gates). As an adder critical path typically resides in the carry propagation path, and since for GNR-based implementations 2-input gates and 3-input gates may yield similar performance, we consider two designs for computing the Sum output (i.e., using 2 2-input XOR gates and using 1 3-input XOR gate).

We summarize in Table VII the topology and dimensions of the GNR gates relevant for the Full Adder implementation and graphically illustrate in Figure 14 the employed GNR shapes. The back gate voltages applied to the adder gates comprising GNRs are included in Table VIII.

Figure 15 presents SPICE simulation results for the GNR based Full Adder implementation and one can observe that the Sum and Carry-Out outputs exhibit the correct functionality.

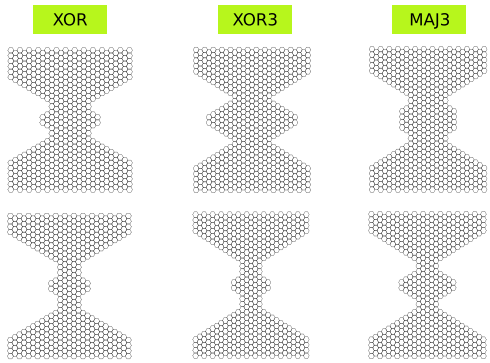

Fig. 14. FA GNR_{up} (top row) and GNR_{dn} (bottom row).

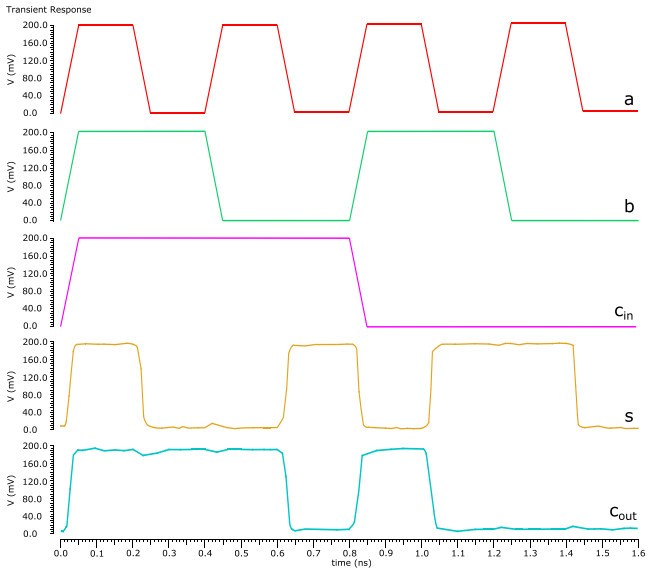

Fig. 15. GNR FA SPICE Simulation Results.

TABLE IX
FA DELAY, AREA, AND POWER CONSUMPTION

	$\tau_p[ps]$		Active Area $[nm^2]$		Total Power $[nW]$	
	GNR	CMOS	GNR	CMOS	GNR	CMOS
FA	1.910	11.863	$1.538 \cdot 10^2$	$3.004 \cdot 10^4$	6.188	$7.915 \cdot 10^3$

When using a single 3-input GNR XOR gate for computing the Sum, we obtain a delay of 2.878 ps, while when using 2 cascaded 2-input GNR XOR gates, we measure a delay of 1.910 ps. Thus, we opted for the latter logic implementation of the Full Adder Sum output bit. The Carry-Out delay is determined by the 3-input GNR MAJORITY gate with a measured value of 0.109 ps. Table IX summarizes the propagation delay, area, and power consumption measured figures for the 1-bit GNR-based and CMOS-based Full Adders, and indicate that the GNR FA has 6.2× smaller delay, requires 2 orders of magnitude smaller area, and consumes 3 orders of magnitude less power than the CMOS counterpart. We note however that for implementations of Ripple Carry Adders (RCA), which are the quite common, the

TABLE X
GNR SRAM COMPONENT DIMENSIONS

		(W, L)	(W_c, L_c)	(W_b, L_b)	(P_{V_g}, W_{V_g})
INV	GNR_{up}	$(35, 27\sqrt{3})$	$(0, 0)$	$(14, 11\sqrt{3})$	$(5\sqrt{3}, 3\sqrt{3})$
	GNR_{dn}	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(5, 2\sqrt{3})$	$(5\sqrt{3}, 3\sqrt{3})$
	$GNR_{L/R}$	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(0, 0)$	$(5\sqrt{3}, 3\sqrt{3})$

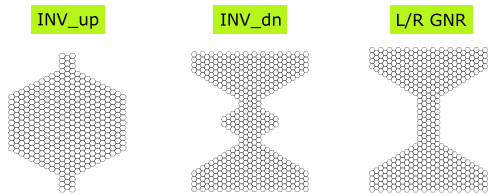

Fig. 16. SRAM Cell GNR Topologies.

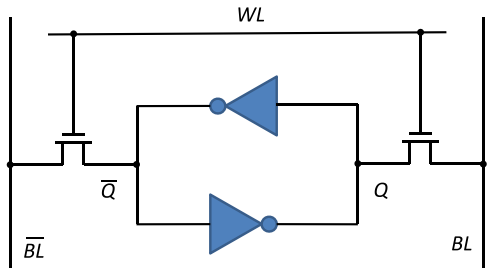

Fig. 17. SRAM Cell.

Carry-Out delay is the one determining the overall adder performance. Thus, as the Carry-Out delay is 0.109 ps and 11.863 ps, for the GNR and CMOS FA, respectively, an n-bit GNR RCA will be 108× faster than the CMOS counterpart.

B. SRAM Cell

Further, we consider an SRAM cell, illustrated in Figure 17, which is widely utilized for data storage, and investigate its performance when designed using GNRs relative to the 6T 7 nm FinFET CMOS counterpart. The dimensions and shapes of the two left/right access GNRs and of the GNRs belonging to the inverter gate are presented in Table X and Figure 16, respectively. As back gate voltages, we use -0.1 V and 0 V for the inverter GNR_{up} and GNR_{dn}, respectively, and 0 V for the left/right access GNR.

We analyze cell robustness to variability during the information retention state for both CMOS and GNR configurations, which is characterized by the Static Noise Margin (SNM) defined as the minimum amount of DC noise required in order to flip the SRAM cell state. The SNM value is given by the side of the biggest square embeddedable between the two DC characteristics of the cross-coupled inverters, illustrated in Figure 18 and Figure 19 for CMOS and GNR cells, respectively. Simulation results indicate an SNM value of 0.25 V ($\approx 35.7\%$ from $V_{DD} = 0.7$ V) for the 7 nm FinFET CMOS configuration and of 0.072 V ($\approx 36\%$ from $V_{DD} = 0.2$ V) for the GNR counterpart, thus we can conclude that the two memory cells exhibit similar

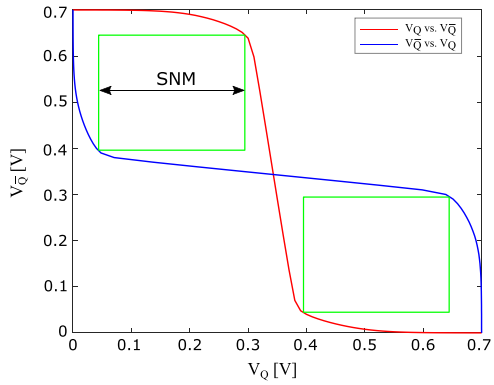

Fig. 18. 6T SRAM SNM Diagram.

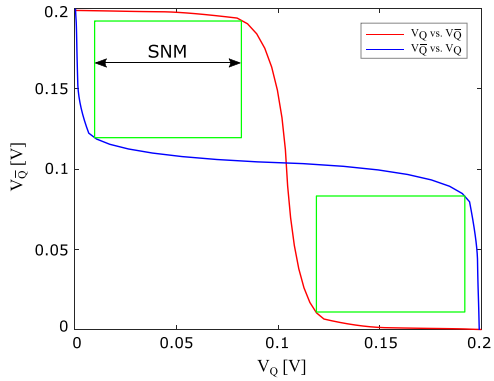

Fig. 19. GNR SRAM SNM Diagram.

TABLE XI
SRAM DELAY, AREA, AND POWER CONSUMPTION

	$\tau_p [ps]$		Active Area $[nm^2]$		Total Power $[nW]$	
	GNR	CMOS	GNR	CMOS	GNR	CMOS
SRAM	0.763	2.729	$6.776 \cdot 10^1$	$9.68 \cdot 10^2$	4.429	$2.622 \cdot 10^2$

DC noise voltage tolerance. Performance-wise, as presented in Table XI, the GNR SRAM cell provides 3.6× smaller delay, consumes 2 orders of magnitude smaller power, and requires 1 order of magnitude less active area than the CMOS SRAM cell.

We conclude, based on the simulation results presented in Section V-A and Section V-B, that GNR-based implementations can potentially outperform CMOS counterparts and that the proposed approach is opening a promising avenue towards future carbon-based nanoelectronics.

VI. CONCLUSION

In this paper, we proposed 1-, 2-, and 3-input GNR Boolean gates and investigated their potential as building structures for post-CMOS circuits. For this purpose, we introduced a generic GNR Boolean gate which is constructed using two GNRs arranged in a complementary manner (one GNR executes the

gate Boolean function, and the other GNR executes the inverted Boolean function). Then, we identified a set of suitable GNR geometries and gate topologies, while taking into account the gate output switching behaviour, and presented 1-input {BUFF, INV}, 2-input {AND, NAND, XOR}, and 3-input {AND, NAND, XOR, MAJORITY} gate designs. We validated the correct operation and evaluated the proposed gates in Cadence. We modelled the GNR conductance using a Verilog-A model which relies on the NEGF-Landauer formalism via an internally triggered Simulink model. Simulation results indicated that, when compared against 7 nm FinFET CMOS counterparts, the proposed gates exhibit 6× to 2 orders of magnitude smaller propagation delay, 2 to 3 orders of magnitude lower power consumption, and require 2 orders of magnitude less active area footprint. We further presented Full Adder (FA) and SRAM cell GNR designs, as they are currently fundamental components for the construction of any computation system. For an effective FA implementation we introduced a 3-input MAJORITY gate, which apart of being able to directly compute FA's Carry-Out, is an essential element in the implementation of Error Correcting Codes (ECC) decoders, that outperforms the CMOS equivalent Carry-Out calculation circuit by 2 and 3 orders of magnitude in terms of delay and power consumption, respectively, while requiring 2 orders of magnitude less area. The proposed GNR FA exhibits 6.2× smaller delay, 3 orders of magnitude less power consumption, while requiring 2 orders of magnitude less area, when compared with the 7 nm FinFET CMOS counterpart, and that a GNR-based n-bit Ripple Carry Adder is potentially 108× faster than an equivalent CMOS implementation. The GNR based SRAM cell provides a slightly better resilience to DC noise characteristics, while performance-wise has a 3.6× smaller delay, consumes 2 orders of magnitude less power, and requires 1 order of magnitude less area than the CMOS equivalent. Our investigations clearly suggest that GNR-based implementations can potentially outperform CMOS counterparts and that the proposed approach is opening a promising avenue towards future carbon-based nanoelectronics.

REFERENCES

[1] J. J. Liou, F. Schwierz, and H. Wong, *Nanometer CMOS.* Singapore: Pan Stanford Publishing, 2010.

[2] K. Rupp and S. Siegfried, "The economic limit to Moore's law," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 1, pp. 1–4, Feb. 2011.

[3] A. C. Ferrari *et al.*, "Science and technology roadmap for graphene related 2D crystals, and hybrid systems," *Nanoscale*, vol. 7, no. 11, pp. 4587–5062, 2015.

[4] S. Stankovich *et al.*, "Graphene-based composite materials," *Nature*, vol. 442, pp. 282–286, 2006.

[5] W. Ren and H. M. Cheng, "The global growth of graphene," *Nature Nanotechnol.*, vol. 9, pp. 726–730, 2014.

[6] M. J. Marmolejo and J. Velasco-Medina, "Review on graphene nanoribbon devices for logic applications," *Microelectron. J.*, vol. 48, pp. 18–38, 2016.

[7] V. Skakalova and A. B. Kaiser, *Graphene: Properties, Preparation, Characterisation and Devices.* Amsterdam, The Netherlands: Elsevier, 2014.

[8] J. Nilsson *et al.*, "Electronic properties of graphene multilayers," *Phys. Rev. Lett.*, vol. 97, no. 26, 2006, Art. no 266 801.

[9] A. H. Castro Neto *et al.*, "The electronic properties of graphene," *Rev. Mod. Phys.*, vol. 81, no. 1, pp. 109–162, 2009.

[10] W. Choi and J. W. Lee, *Graphene Syntehsis and Applications.* Boca Raton, FL, USA: CRC Press, 2012.

[11] J. M. Allen, V. C. Tung, and R. B. Kaner, "Honeycomb carbon: A review of graphene," *Chem. Rev.*, vol. 110, no. 1, pp. 132–145, 2010.

[12] K. Matsumoto, *Frontiers of Graphene and Carbon Nanotubes—Devices and Applications.* New York, NY, USA: Springer-Verlag, 2015.

[13] Z. F. Wang *et al.*, "Emerging nanodevice paradigm: Graphene-based electronics for nanoscale computing," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 5, no. 1, pp. 1–19, 2009.

[14] M. R. Stan *et al.*, "Graphene devices, interconnect and circuits—challenges and opportunities," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 69–72.

[15] S. Ryu *et al.*, "Raman spectroscopy of lithographically patterned graphene nanoribbons," *ACS Nano*, vol. 5, pp. 4123–4130, 2011.

[16] L. Liu *et al.*, "Nanosphere lithography for the fabrication of ultranarrow graphene nanoribbons and on-chip bandgap tuning of graphene," *Adv. Mater.*, vol. 23, pp. 1246–1251, 2011.

[17] Z. Chen *et al.*, "Synthesis of graphene nanoribbons by ambient-pressure chemical vapor deposition and device integration," *J. Amer. Chem. Soc.*, vol. 138, no. 47, pp. 15 488–15 496, 2016.

[18] D. Kosynkin *et al.*, "Longitudinal unzipping of carbon nanotubes to form graphene nanoribbons," *Nature*, vol. 458, pp. 872–876, 2009.

[19] F. Cataldo *et al.*, "Graphene nanoribbons produced by the oxidative unzipping of single-wall carbon nanotubes," *Carbon*, vol. 48, pp. 2596–2602, 2010.

[20] W. Xu *et al.*, "Rapid fabrication of designable large-scale aligned graphene nanoribbons by electro-hydrodynamic nanowire lithography," *Adv. Mater.*, vol. 26, pp. 3459–3464, 2014.

[21] L. Talirz, P. Ruffieux, and R. Fasel, "On-surface synthesis of atomically precise graphene nanoribbons," *Adv. Mater.*, vol. 28, pp. 6222–6231, 2016.

[22] Y. Jiang, N. Cucu Laurenciu, and S. D. Cotofana, "On carving basic Boolean functions on graphene nanoribbons conduction maps," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.

[23] I. G. Karafyllidis, "Current switching in graphene quantum point contacts," *IEEE Trans. Nanotechnol.*, vol. 13, no. 4, pp. 820–824, Jul. 2014.

[24] Y. Jiang, N. Cucu Laurenciu, and S. D. Cotofana, "Complementary arranged graphene nanoribbon-based Boolean gates," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures*, 2018, pp. 1–7.

[25] S. Datta, *Quantum Transport: Atom to Transistor.* Cambridge, U.K.: Cambridge Univ. Press, 2005.

[26] I. Nikiforidis, I. Karafyllidis, and P. Dimitrakis, "Simulation and parametric analysis of graphene p-n junctions with two rectangular top gates and a single back gate," *J. Phys. D, Appl. Phys.*, vol. 51, pp. 1–6, 2018.

[27] Arizona State University, "Predictive technology model." 2018. [Online]. Available: http://ptm.asu.edu/

[28] Cadence. 2018. [Online]. Available: https://www.cadence.com/

[29] Y. Jiang, N. Cucu Laurenciu, and S. D. Cotofana, "Non-equilibrium green function-based Verilog-A graphene nanoribbon model," in *Proc. IEEE 18th Int. Conf. Nanotechnol.*, 2018, pp. 1–4.

[30] Simulink, 2018. [Online]. Available: https://www.mathworks.com/

[31] TSMC Future Research and Development Plans. 2018. [Online]. Available: http://www.tsmc.com/english/dedicatedFoundry/technology/future_rd.htm

[32] S. S. Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Trans. Signal Process.*, vol. 5, no. 11, pp. 5692–5703, Nov. 2008.

[33] T. T. Nguyen-Ly *et al.*, "Flexible, cost-efficient, high-throughput architecture for layered LDPC decoders with fully-parallel processing units," in *Proc. Euromicro Conf. Digit. Syst. Des.*, 2016, pp. 230–237.

[34] N. Cucu Laurenciu *et al.*, "Error correction code protected data processing units," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures*, 2016, pp. 37–42.

[35] A. Amaricai, V. Savin, O. Boncalo, N. Cucu-Laurenciu, J. Chen, and S. Cotofana, "Timing error analysis of flooded LDPC decoders," in *Proc. IEEE Int. Conf. Microwaves, Commun., Antennas Electron. Syst.*, 2015, pp. 1–5.

Yande Jiang (S'17) received the M.Sc. degree in electronic science and technology from the Faculty of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 2015. He is currently working toward the Ph.D. degree in computer engineering at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His research interests include graphene-based nano-electronics with a focus on digital circuit design and VLSI placement and routing. He is an Affiliate Member of the HiPEAC.

Nicoleta Cucu Laurenciu (M'12) received the M.Sc. degree in computer engineering, in 2010, and the Ph.D. degree in electrical engineering, in 2017, both from Delft University of Technology, Delft, the Netherlands. She is currently a Postdoctoral Researcher with the Faculty of Electrical Engineering, Computer Science and Mathematics, the Computer Engineering Laboratory, Delft University of Technology, The Netherlands. Her research interests include reliability aware design methodologies, artificial intelligence based computation paradigms, emerging nanodevices based computing, and foremost graphene based nano-computing. She is an Affiliate Member of the HiPEAC.

He Wang (S'18) received the M.Sc. degree in computer science and technology from the Faculty of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 2016. He is currently working toward the Ph.D. degree in computer engineering at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His research interests include graphene-based neuromorphic computing. He is an Affiliate Member of the HiPEAC.

Sorin Dan Cotofana (M'93–SM'00–F'17) received the M.Sc. degree in computer science from "Politehnica" University of Bucharest, Romania, in 1984, and the Ph.D. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands. He is currently with the Faculty of Electrical Engineering, Mathematics and Computer Science, the Computer Engineering Laboratory, Delft University of Technology, The Netherlands. His current research interests include the following: 1) the design and implementation of dependable/reliable systems out of unpredictable/unreliable components; 2) aging assessment/prediction and lifetime reliability aware resource management; and 3) unconventional computation paradigms and computation with emerging nano-devices. He has coauthored more than 250 papers in peer-reviewed international journal and conferences, and received 12 best paper awards in international conferences, such as 2016 IEEE/ACM International Symposium on Nanoscale Architectures, 2012 IEEE Conference on Nanotechnology, 2012 ACM/IEEE International Symposium on Nanoscale Architectures, 2005 IEEE Conference on Nanotechnology, and 2001 International Conference on Computer Design. He was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I (2009–2011), IEEE TRANSACTIONS ON NANOTECHNOLOGY (2008–2014), and *Nano Communication Networks* (2010–2014). Also, he was a Senior Editorial Board Member of the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS (2016–2017), Steering Committee Member for the IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS (2014–2018), Chair of the GigaNano IEEE CASS Technical Committee (2013–2015), and IEEE Nano Council CASS representative (2013–2014). Furthermore, he has been actively involved as a Reviewer, TPC Member, and Track, Program, and General Co-Chair in the organization of numerous international conferences. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NANOTECHNOLOGY. He is an ACM Member and was appointed by the IEEE Circuits and Systems Society (CASS) as a Distinguished Lecturer for the period 2019–2020.