A Verilog-A Model for Reconfigurable Logic Gates Based on Graphene pn-Junctions

Sandeep Miryala, Mehrdad Montazeri, Andrea Calimera, Enrico Macii, Massimo Poncino

Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy
Email: {sandeep.miryala, andrea.calimera, enrico.macii, massimo.poncino}@polito.it

***Abstract*—Single layer sheets of graphene show special electrical properties that can enable the next generation of smart ICs. Recent works have proven the availability of an electrostatically controlled pn-junction upon which it is possible to design multi-function reconfigurable logic devices that naturally behave as multiplexers. In this work we introduce a stable large-signal Verilog-A model that mimics the behavior of the aforementioned devices. The proposed model, validated through the SPICE characterization of a MUX-based standard cell library we designed as benchmark, represents a first step towards the integration of Electronic Design Automation tools that can support the design of all-graphene ICs.**

I. Introduction and Graphene Physics

Isolated for the first time in 2004 [1], [2], graphene is a one-atom-thick sheet of graphite where all the carbon atoms form covalent bonds in a single plane. The resulting hexagonal crystal lattice provides graphene with a gapless energy band structure in which the conduction and the valence bands touch each other at the Fermi energy (E_F) [3]. Therefore, differently from semiconductors, where E_F falls in the bandgap, with graphene it is not possible to implement an "OFF" state.

This characteristic has been initially used as an argument to support the inadequacy of graphene in the implementation of electronic devices and has forced the research community to find new process techniques that can open a gap in the energy band structure. Graphene Nanoribbons (GNRs), which are narrow stripes of graphene, have been emerged as potential candidates that overcome this issue, thereby allowing the creation of graphene-based transistors [4].

However, cutting graphene into nanoribbons is not the only possible choice. Few recent works proposed the use of an equivalent graphene-based pn junction [5] to implement a new class of devices which show optic-like properties and that naturally behave as a reconfigurable multiplexer [6]; interconnections of multiple devices and proper signal assignments of inputs allows to realize all the basic logic Boolean functions. The analysis reported in [6] and [7] indicate that this reconfigurable gate can outperform CMOS technologies below the 22nm node in terms of both delay and power consumption.

Motivated by these encouraging results, in this work we propose a stable large-signal device model written in *Verilog-A*. In order to validate and test the stability of the model, we designed and characterized a standard cell library consisting of several logic cells implemented through the reconfigurable graphene primitive. Simulation data have been collected in a LUT-based timing library formatted using the EDA *Liberty* standard of Synopsys. The model we proposed, combined with the quantitative analysis reported in the experimental section, can support CAD developers during the implementation of new synthesis tools and optimization techniques that better exploit the potentials of a graphene-based reconfigurable technology.

978-3-9815370-0-0/DATE13/©2013 EDAA

II. PN Junction Logic Devices on Graphene

A. Graphene pn Junction

The Fermi energy E_F of a sheet of pristine graphene passes at zero eV where the valence band and the conductance band touch each other; this provides the material with semimetallic properties. Exploiting an *electrostatic doping* [3], it is however possible to shift the E_F down in the valence band, or up in the conduction band, obtaining p-type or n-type graphene respectively.

The structure of an electrostatically controlled pn junction is shown in Figure 1 and consists of two split metal back-gates isolated from the graphene sheet by a thick layer of oxide. The two metal "sticks" drawn in the figure represent the conceptual source and the probe that are emitting and receiving the carriers. One can apply a symmetric control voltage ($\pm U$) to the two back-gates: a negative voltage ($-U$) makes graphene above the the gate p-doped, whereas a positive voltage ($+U$) makes the above graphene n-doped.

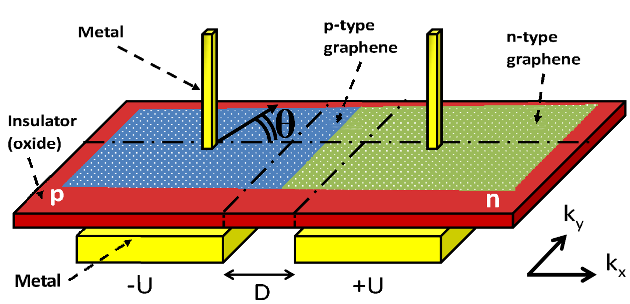

Fig. 1. Graphene-based pn junction.

As demonstrated in [8], the existence of the Klein tunneling allows carriers from the p-region to cross the junction with a probability $T(\theta)$ that depends on (i) the angle θ between the electron's wave vector $\mathbf{k}$ and the normal of the junction and (ii) the width D of the pn transition region. Equation. 1 gives the analytical expression of $T(\theta)$:

$$T(\theta) = cos^2(\theta)e^{-\pi \mathbf{k} D \sin^2 \theta} \quad (1)$$

Notice that the probability is 1 (regardless of D) when $\theta = 0$, i.e., when the wave vector is orthogonal to the junction, and it reaches 0 at $\pi/2$.

B. Multi-Function Graphene Device

Figure 2 shows the 3D view of the reconfigurable logic gate as presented in [6]. It consists of a graphene sheet, three isolated metal back-gates, U, A and $\bar{U}$, and three upper metal-to-graphene contacts, B, F and C. The back-gates U, A and $\bar{U}$ are used to control the doping profile of the two back-faced pn-junctions ($A-\bar{U}$ on the left and $A-U$ on the right); this allows to dynamically redirect the charges from B-to-F or C-to-F.

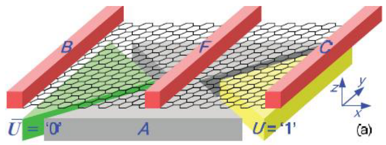

Fig. 2. 3D view of the reconfigurable logic gate [6].

From a functional point of view, the gate behaves as a Multiplexer with B and C being the the data inputs, F the output, and A the selection signal: $F = B$ when $A = 0$, $F = C$ when $A = 1$.

Figure 2 shows the case U ="1" ($\bar{U}$ ="0"). Notice that the "1" logic value is $+V_{dd}/2$ and the "0" logic value is $-V_{dd}/2$. The graphene region above the gate A changes its doping concentration following the voltage of the signal A. When A ="1", the central graphene becomes n-type, forming a pn junction on the left and a nn junction on the right. Such configuration allows a very high electrical resistance (R_{pn}) from B to F, and a very low resistance path from C to F (R_{nn}). Therefore, the output at F follows the signal at C. On the contrary, when A ="0", the central graphene becomes p-type, forming a low resistive pp junction on the left; this makes the output at F to follow B.

The triangular shape of the back-gates allows the pn-junctions to have a 45° rotated normal. As we show later in the text, this impacts on the resulting transmission probability $T(\theta)$, and thus, on the electrical properties of the device.

In this work, the back-gate U is fixed at logic "1" ($\bar{U}$="0")[1], but one could fix U="0" changing the polarity of the multiplexer: $F = C$ when $A = 0$, or $F = B$ when A="1".

III. Verilog-A Model

A. RC Electrical Model

In the reconfigurable device described in Section II-B, the two programmable pn junctions $A-\bar{U}$ and $A-U$ connect the output F with the inputs B and C respectively. When set in the nn (or pp) mode, i.e., when the back-gate voltages are concordant ($A = \bar{U}$ or $A = U$), the junction exhibits a low-resistance (R_{nn} or R_{pp}). When set in the pn (or np) mode, i.e., opposite voltages at the back-gates ($A \neq \bar{U}$ or $A \neq U$), the junctions show a high-resistance (R_{pn} or R_{np}). It is worth mentioning that (i) the device shows a symmetric structure, therefore $R_{nn} = R_{pp}$, and (ii) an electrostatic controlled graphene pn junction it is not a true pn diode: when reverse-biased, current flows in the opposite direction, that is $R_{pn} = R_{np}$.

[1]This choice simplifies the interconnection of multiple devices in a row, similarly to what is done with regular CMOS standard cells where fixed metal contacts work as V_{dd} and V_{gnd}.

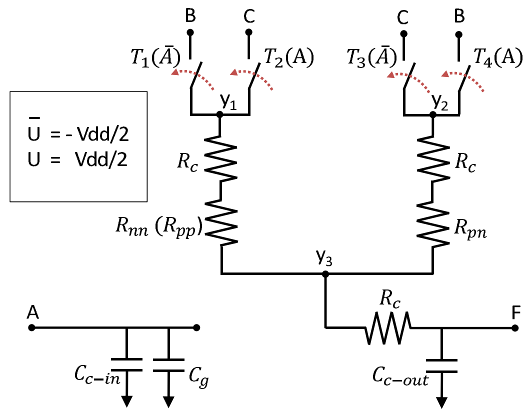

Fig. 3. Electrical schematic of the reconfigurable device.

The value of R_{nn} is simply defined by the geometrical width W of the graphene sheet, as shown in the following equation:

$$R_{nn} = R_o/N_{ch} = \frac{h\pi}{4q^2 W k_F} \tag{2}$$

where $R_o = h/4q^2$ is the quantum resistance per mode in the graphene sheet, and $N_{ch} = W k_F/\pi$ represents the number of excited modes, with k_F the Fermi wave vector.

The value R_{pn} is function of the transmission probability $T(\theta)$, with θ=45° as physically defined by the geometrical shape of the back-gates:

$$R_{pn} = \frac{R_o}{T(\theta)N_{Ch}} = \frac{h\pi}{4q^2 W K_F T(45°)} \tag{3}$$

where $T(45°)$ can be estimated from Equation1.

Figure 3 shows the equivalent electrical model of the device. It consists of four switches T_1, T_2, T_3, T_4, controlled by the input voltage A. With $V(A)$ ="1", T_2 and T_4 are closed, while T_1 and T_3 open; hence, C is connected to y_3 through the low resistive path R_{nn} (left branch) while B is isolated from y_3 thanks to the high resistive path R_{pn} (right branch). If $V(A)$ ="0", T_1 and T_3 are closed, while T_2 and T_4 open; hence, B is connected to y_3 through the low resistive path R_{nn} (left branch), and C is isolated from y_3 thanks to the high resistive path R_{pn} (right branch).

The electrical model also includes resistive and capacitive parasitics. R_c (assumed to be 10Ω) represents the resistance of a metal-to-graphene contacts and it has been applied for the three front metal contacts B, C and F[6].

The lumped capacitance C_g at the back-gate A consists of the series of the oxide capacitance C_{ox} and the quantum capacitance from the graphene sheet C_q, i.e., $C_g = 1/(C_{ox}^{-1} + C_q^{-1})$. C_{ox} is the conventional parallel plate capacitance and its value per unit of area is given by $C_{ox} = \frac{\epsilon}{t_{ox}}$, where ϵ and t_{ox} are the the dielectric constant and the dielectric thickness respectively. The quantum capacitance C_q is a function of the density of carriers at the Fermi energy E_F, whose analytical equation is given by:

$$E_F = \frac{1}{\gamma t_{ox}}\left(\sqrt{\epsilon^2 + 2\gamma\epsilon q V_g t_{ox}} - \epsilon\right) \tag{4}$$

where $\gamma = (4\pi q^2)/(h^2 v_F^2)$ (q is the electron charge, h the Plank's constant, v_F the Fermi velocity), ϵ is the permittivity of the oxide and V_g the gate control voltage applied at the corresponding back gate. The quantum capacity per unit of area is therefore given by $C_q = \gamma E_F$. To notice that we do not consider C_g for the back-gates U and $\bar{U}$ as they do not contribute to the dynamic of the device.

The elements C_{c-in} at node A and C_{c-out} at node F represent the capacitive coupling among the back-gates and the front metal contacts respectively. Both are function of the distance between the metal contacts and the coupling surface area.

B. Verilog-A: Implementation Details

Verilog-A is a general-purpose Hardware Description Language for circuit modeling and simulation. The basic paradigm behind verilog-A descriptions is to represent the circuit in terms Kirchhoff's Voltage and Current Laws (KVL and KCL) between the nodes.

The verilog-A code for the reconfigurable logic gate is shown in Algorithm 1.
Line 1 and 2 include the the variables that make the verilog-A specific for electrical modeling along with those that define universal constants.
Lines 3 to 5 represent the module block. The module name is defined in Line 3, `dev`, while Line 4 and 5 define the ports direction.
Lines 8 and 9 define the internal variables and parameters (values are omitted for the sake of space) used to calculate the values of electrical elements.
Lines 12 to 16 define the equations described in SectionII through which the simulator calculates the value of R_{nn} and R_{pp}, both function of the voltage at the control port A.
Lines 17 to 20 define the equations for input parasitic capacitance $C_{in} = C_g + C_{c-in}$.
Line 21 defines the KCL at the input port A: $C_{in} \cdot dV(A)/dt$.
Lines 22 to 28 define the behavior of the four switches which are controlled by the voltage at node A; e.g., when A is at '1', $T2$ is closed and the voltage at Y_1 is set to the voltage of the input port C.
Lines 29 to 31 define the I-V equations from the nodes Y_1 and Y_2 to the intermediate node Y_3
Line 32 finally describes the KCL at the output port F: $C_{c-out} \cdot dV(F)/dt$.

IV. SIMULATING THE VERILOG-A MODEL

A. Case of Study: A MUX-based Logic Gate Library

Figure 4 shows the logic gates designed using as primitive the re-configurable graphene device. We call these logic gates *RG-gates* hereafter. Some have different implementations (called architectures), whereas some others (NOT, BUF, and XOR) have a single one.

The actual contribution of this section it is not the figure itself (the MUX-equivalents of the basic gates are somehow straightforward), but rather (i) to verify the functionality of the Verilog-A model when used to implement multi-stages RG-gates and (ii) characterize the performance of those gates. The obtained results, collected in look-up tables (LUTs) and formatted using the Synopsys .LIB standard, make the RG-gates usable as conventional cells in standard CMOS synthesis flow; moreover, they provide designers with new rules of thumb for future design optimizations.

Algorithm 1 Verilog-A code for the reconfigurable logic gate

```
1: `include ``disciplines.vams"
2: `include ``constants.vams"
3: module dev(A,B,C,F);
4: input A, B, C;
5: output F;
6: real Tox, d, Ef, pi, q, h, Vf, Y, E, EOT, Kf, C_ox,
     C_q, T, R_nn, R_pn, R_c;
7: real C_c-in, C_c-out, C_g, C_in, A;
8: analog
9: begin
10:     Y = (4*pi*q^2)/(h^2*Vf^2);
11:     E_f = (sqrt(pow(E,2)+
12:         +(2*Y*E*q*abs(V(a))*EOT)) - E)/(Y*EOT);
13:     Kf = (2*pi*abs(E_f))/(h*Vf);
14:     R_nn = (pi*h)/(4*q^2*w*Kf);
15:     R_pn = R_nn/T;
16:     C_ox = E/Tox;
17:     C_q = Y*Ef;
18:     A = w^2 + 2*w*d;
19:     C_g = A*(C_ox*C_q)/(C_ox + C_q);
20:     C_in = C_g + C_c-in;
21:     I(A) <+ C_in*ddt(V(A));
22:     if (V(A) >= 0)
23:       V(Y1) <+ V(C);
24:       V(Y2) <+ V(B);
25:     else
26:       V(Y1) <+ V(B);
27:       V(Y2) <+ V(C);
28:     end if
29:     V(Y3,Y1) <+ I(Y3,Y1)*(R_nn + R_c);
30:     V(Y3,Y2) <+ I(Y3,Y2)*(R_pn + R_c);
31:     V(Y3,F) <+ R_c*I(Y3,f);
32:     I(F) <+ C_c-out*ddt(V(F));
33: end
34: endmodule
```

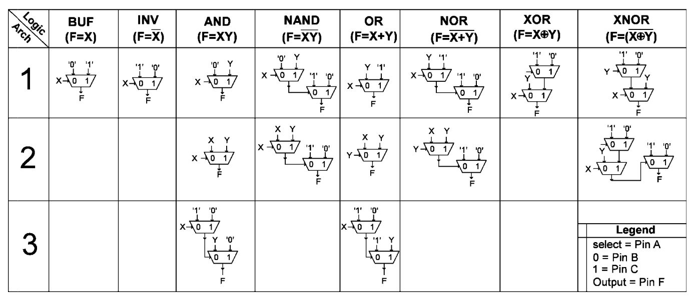

Fig. 4. Logic gates architectures using the re-configurable gate as a primitive.

B. Library Characterization

Each gate has multiple LUTs, each one describing the output rise/fall propagation delay and the output rise/fall transition time; gate delays are measured for different fanout and input rise/fall time ($T_{r/f}$). The propagation delay is defined as 50% of input signal to 50% of output signal; output transition time is measured from 20% to 80% for rising and 80% to 20% for falling. A fanout (FO) of n indicates that the cell is connected to n cells of the same type.

From the collected results we first observed that, as for the case of conventional CMOS technologies, RG-gates get slower for larger fanout and for larger input transition times. We omitted those numbers for the sake of space.

Table I reports the worst case propagation delays under different FO;data refers to Architecture 1 of Figure 4. For INV, BUF, AND and OR the worst case delay is observed from the select

	Rising propagation delay [ps]				
Gate	FO1	FO2	FO3	FO4	FO5
INV	0.486	0.680	0.861	0.987	1.110
BUF	0.486	0.680	0.861	0.987	1.110
AND	0.486	0.680	0.861	0.987	1.110
OR	0.486	0.680	0.861	0.987	1.110
NAND	0.567	0.634	0.672	0.706	0.736
NOR	0.567	0.634	0.672	0.706	0.736
XOR	0.486	0.680	0.861	0.987	1.110
XNOR	0.486	0.680	0.861	0.987	1.110

TABLE I. PROPAGATION DELAY FOR THE RG-GATES

pin of the multiplexer to the main output F. All those cells have same delay as they are implemented with a single stage. In general, this is not true in conventional CMOS technologies where different logic gates are typically implemented with different internal architectures.

Cells like NAND and NOR gates, which have one extra stage more w.r.t. INV and BUF, should have a delay that is roughly 2x larger. However, this is not the case as can be seen from the table. The effect can be explained by observing the in-to-out timing path. The worst case timing arc is measured from the select pin of the first stage to the main output F, passing through the select pin of the second gate. The total delay is therefore given by the sum of the delay of the first stage, same as INV with FO1 (regardless of the main output FO) and the delay of the second stage. The latter has been observed being very small (due to a fast transition of the signal generated by the first stage) and less sensitive from the output FO (as the select pin, i.e., the back-gate of the device, is physically isolated from F). Thus, the overall delay is greater than that of INV for FO1 and slowly increases with the output FO.

Also the gates XOR and XNOR, implemented with two stages, should have same delay of NAND and NOR gates. Instead they have the same speed of cells implemented with a single stage (e.g., INV, BUF). Main reason of this effect is that the worst case delay path is observed from the select pin of the second stage X to the main output F, and not from Y to F as one can intuitively expect. In fact, we observed that longest timing paths are usually those that involve the select pin of the multiplexer. This effect is representative of an important property: *the delay is primarily determined by the type of terminals involved in the critical path rather than the mere number of stages traversed*. Although this shares some similarities with CMOS devices, where the pin assignment has a strong impact on the resulting performance of the gate, in CMOS the number of traversed gates is a more meaningful estimate of propagation delay.

The analysis continues considering the dependence from FO, columns 1 to 5. There are two main delay "groups" of cells: one including INV, BUF, AND, OR, XOR and XNOR which are slower (except for FO1) and more sensitive to load variations (2.3x difference from FO1 to FO5), and an another one, made up of NAND and NOR, that is faster and less sensitive to load variation (1.3x difference from FO1 to FO5). In general, it appears that *faster cells are less immune to load variations*.

As final remark, Figure 5 compares the delay of those gates that have multiple implementations, namely, AND, NAND, OR, NOR and XNOR. The characterization is done with a fanout of 4. It can be inferred from the plot that when the selector pin of the multiplexer (i.e., the back-gate of the

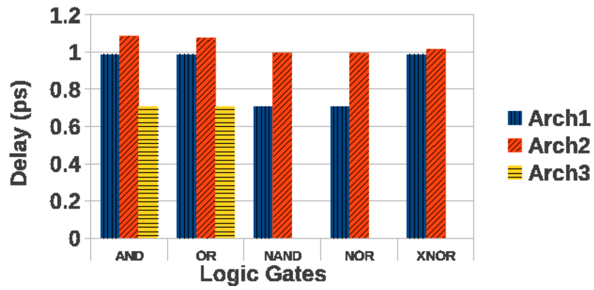

Fig. 5. Delay comparison for multi-architecture gates.

graphene device) is short-circuited to one of the data input (i.e., one of the front metal contacts B or C in the graphene device), e.g., signal X in the first stage of the NAND gate, the cell gets slower. This is the case for NAND and NOR, where the second architecture $Arch2$ is slower than $Arch1$. The same motivation holds for AND and OR gates, where $Arch2$ is slower. Also for OR and AND gates it can be seen that $Arch3$ is faster than $Arch2$.
Concerning the XNOR gate, $Arch2$ is slower than $Arch1$ because of the additional stage.

V. CONCLUSIONS

In this paper we introduced a Verilog-A behavioral model for a graphene-based multi-functional logic gate. The model was used to characterize basic logic gates implemented by means of the primitive graphene-based device. Our approach was oriented towards a conventional paradigm based on library-based synthesis. Experimental results demonstrated the stability of the proposed model and allowed us to identify common design patterns that will be used for future circuit optimizations.

REFERENCES

[1] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov, "Electric field effect in atomically thin carbon films," *Science*, vol. 306, no. 5696, pp. 666–669, 2004.

[2] K. S. Novoselov, D. Jiang, F. Schedin, T. Booth, V. Khotkevich, S. Morozov, and A. K. Geim, "Two-dimensional atomic crystals," in *Proceedings of the National Academy of Sciences of The United States of America*, 2005, pp. 10 451–10 453.

[3] A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov, and A. K. Geim, "The electronic properties of graphene," *Reviews of Modern Physics*, vol. 81, pp. 109–162, Jan 2009.

[4] M. Lemme, T. Echtermeyer, M. Baus, and H. Kurz, "A graphene field-effect device," *IEEE Electron Device Letters*, vol. 28, no. 4, pp. 282–284, apr 2007.

[5] V. V. Cheianov, V. Fal'ko, and B. L. Altshuler, "The focusing of electron flow and a veselago lens in graphene p-n junctions," *Science*, vol. 315, no. 5816, pp. 1252–1255, 2007.

[6] S. Tanachutiwat, J. U. Lee, W. Wang, and C. Y. Sung, "Reconfigurable multi-function logic based on graphene p-n junctions," in *DAC'10: ACM/IEEE Design Automation Conference*, june 2010, pp. 883 –888.

[7] P. Chenyun and A. Naeemi, "Device- and system-level performance modeling for graphene p-n junction logic," in *ISQED'12: IEEE International Symposium on Quality Electronic Design*, march 2012, pp. 262 –269.

[8] V. V. Cheianov and V. I. Fal'ko, "Selective transmission of dirac electrons and ballistic magnetoresistance of n-p junctions in graphene," *Phys. Rev. B*, vol. 74, p. 041403, Jul 2006.