Delay Model for Reconfigurable Logic Gates Based on Graphene PN-junctions

Sandeep Miryala, Andrea Calimera,
Enrico Macii, Massimo Poncino
Politecnico di Torino, 10129, Torino, ITALY

ABSTRACT

In this paper we address the problem of modeling the timing behavior of a new class of reconfigurable logic gates based on electrostatically controlled graphene pn-junctions. These gates naturally behave as a 2-to-1 multiplexer in which the polarity of the input select line can be dynamically reconfigured. Interconnection of multiple gates and proper assignments of the inputs signals allow to implement all the basic Boolean logic functions, and, at a larger scale, any digital circuit.
Exploiting the symmetric structure of the graphene device, we first identify the main in-to-out timing arcs of the gate, then we derive the parametric models that accurately describe the propagation delay across them as function of the input signal transition slope and the output capacitive load. Delay equations are fit to SPICE-level simulation data by means of bilinear and linear interpolation methods; the obtained regression coefficients are stored and used as substitutes of the huge lookup-tables typically adopted in CMOS timing libraries.
Experimental results show that the proposed model allows accurate timing analysis with average and peak errors below 0.6% and 3.1% respectively.

Categories and Subject Descriptors

B.6.3 [**Design Aids**]: Simulation

Keywords

Graphene, PN junction, reconfigurable gate, timing model

1. INTRODUCTION

Transparent and stretchable, with high thermal-mechanical resistance and unique electrical qualities [1], graphene is emerging as one of the viable candidate to replace silicon in the post-CMOS electronic era. First isolated in 2004 [9, 10, 8], graphene is available in nature in the form of stacked sheets, i.e., graphite. It shows a two-dimensional (2D) structure in which carbon atoms are packed in an honeycomb lattice with a bond length a_{C-C} of 1.42Å, Figure 1 (a). The resulting energy band structure E is analytically reported in the Equation 1 and plotted in Figure 1 (b)[1].

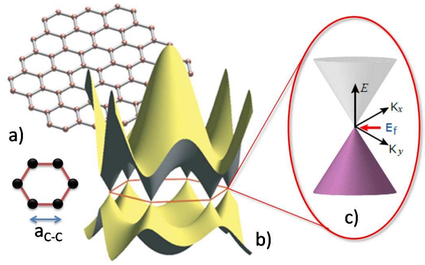


Figure 1: The graphene honeycomb lattice (a), its energy band structure (b) and the linear energy dispersion around the Fermi Energy E_F (c)

$$E^{\pm}(\mathbf{k}) = \pm\gamma\sqrt{1 + 4\cos\left(\frac{\sqrt{3}a}{2}k_x\right)\cos\left(\frac{a}{2}k_y\right) + 4\cos^2\left(\frac{a}{2}k_y\right)} \quad (1)$$

The positive and negative energy branches are the conduction band (E^+) and the valence band (E^-) respectively; the vector $\mathbf{k} = \{k_x, k_y\}$ represents the 2D wavevector; γ is a fitting parameter whose value can range from 2.7eV to 3.3eV; $a = \sqrt{3}a_{C-C}$ is the side length of the parallelogram representing the primitive cell in the Bravis lattice [12]. From the plot one can notice that conduction and valence curves touch each other near the edges of the Brillouin Zone, i.e., at zero energy, where the Fermi energy E_F passes (Figure 1 (c)).
The gapless energy spectrum gives graphene semi-metallic properties that do not allow the material to implement the *off* state. Nevertheless, recent works have jointly exploited (i) the massless propagation of carriers in graphene, and (ii) the existence of a tunneling property called *Klein's tunneling*, to devise an electrostatically controlled pn-junction [6, 2] that allows the control of the current flow. Such a pn-junction serves as basic element for a more complex reconfigurable gate, the target of this work, that conceptually implements a reconfigurable 2-to-1 multiplexer [13] (RG-MUX hereafter). The RG-MUX consists of a graphene sheet with co-planar back-gates through which it is possible to alter the doping profile of the graphene sheet; this allows to select which of the two input signals connected at the front metal

[1] Interested readers can refer to [11] and [12] for a formal derivation, which is out of the scope of this work.

GLSVLSI'13, May 2–3, 2013, Paris, France.

contacts of the device is forwarded to the output. The analysis reported in [13], [3] and [7] indicates that the RG-MUX can outperform CMOS technologies below the 22nm node in terms of both delay and power consumptions.

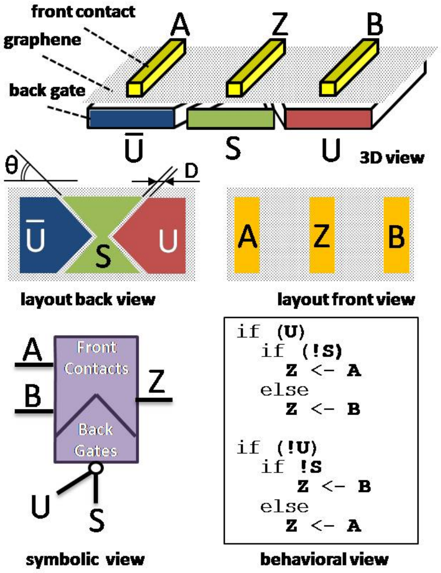
The main contribution of this work is to provide a compact timing model for the RG-MUX. It is worth emphasizing that the proposed model has been conceived with the aim of supporting future logic synthesis tools for graphene ICs, and not as a substitute of analytical equations for device-level simulations. The methodology we followed recalls the NLDM-based approach (NLDM stands for Non-Linear Delay Model) widely employed in CMOS technologies for the creation of standard cells timing libraries. The symmetric structure of the device allowed us to group the timing arcs of the RG-MUX in two main classes, one including *back-to-output* timing arcs, i.e., those from back-gates to the output, the other including *front-to-out* timing arcs, i.e., those that traverse the device from the front metal contacts to the output. SPICE-level characterization data obtained simulating a verilog-A model of the device have been used for model fitting. We demonstrate that the regression coefficients obtained using bilinear and linear interpolation (for back-to-out and front-to-out arcs respectively) are constant over the entire characterization space, thereby enabling LUT-free static timing analysis.

Experimental results show that the proposed model allows accurate timing analysis with an average and peak error below 0.6% and 3.1% respectively w.r.t. SPICE simulations.

2. BACKGROUND

2.1 PN-Junction Graphene Device

Figure 2 shows 3D, back and front views of the RG-MUX as introduced in [13]. The device consists of a graphene sheet, three back-gates, $\bar{U}$, S and U, that are isolated from the graphene by a thick layer of oxide, and three upper metal-to-graphene contacts, A, B, (inputs) and Z (output).



```
if (U)
  if (!S)
    Z <- A
  else
    Z <- B

if (!U)
  if !S
    Z <- B
  else
    Z <- A
```

Figure 2: RG-MUX design views.

The voltages applied at the back-gates $\bar{U}$, S and U, control the doping profile of the two resulting back-faced pn-junctions, S-$\bar{U}$ on the left and S-U on the right; they perform an electrostatic doping [1] through which the Fermi energy E_F can be shift down in the valence band, or up in the conduction band, leading to p-type or n-type graphene respectively.

Pins $\bar{U}$ and U are the *configuration pins* and are always controlled by a complemented logic voltage; the pin S works as selective pin of the MUX. When U='1' ($\bar{U}$='0'), the graphene region above the U gate results as n-doped (the Fermi energy E_F moves up in the conduction band due to positive control voltage), while that above the $\bar{U}$ gate as p-doped (the Fermi energy E_F moves down in valence band due to a negative control voltage). Notice that the '1'-logic value is $+V_{dd}/2$ and the '0'-logic value is $-V_{dd}/2$. Under such configuration, when S='0', the central graphene becomes p-type forming a pp-junction on the left and a pn-junction on the right; the pp-junction shows a low resistance (R_{pp}), whereas the pn-junction a high resistance (R_{pn}). Therefore, the output Z follows the input signal associated with the smallest resistive path, i.e., A. On the contrary, when S ='1', the central graphene becomes n-type forming a low resistive nn-junction on the right (R_{nn}); this forces the output Z to follow B. A dual behavior is observed when U='0' ($\bar{U}$='1'). Figure 2 gives the symbolic and behavioral view of the gate.

2.2 Electrical Model

Borrowing the work of [13], we implemented a Verilog-A compact device model whose equivalent circuit is drawn in Figure 3.

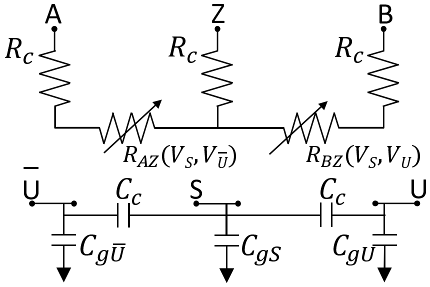


Figure 3: RG-MUX electrical model.

The two resistors R_{AZ} and R_{BZ} model the resistive graphene path from the input pins A and B to the output Z respectively. Their value ranges from $R_{nn} = R_{pp}$ (that is the on state) when the back-gates are polarized with concordant voltages, to R_{pn} (that is the off state) when the back-gates are polarized with discordant voltages.

The value of R_{nn} is given by:

$$R_{nn} = R_0/N_{ch} \qquad (2)$$

where $R_0 = \frac{h}{4q^2}$ is the quantum resistance per mode and N_{ch} is the number of excited modes in the graphene sheet. The following table reports the equation of N_{ch} for R_{AZ} and R_{BZ}:

R_{AZ}	R_{BZ}
$N_{ch} = \frac{W}{\pi} min(K_{F,S}, K_{F,\bar{U}})$	$N_{ch} = \frac{W}{\pi} min(K_{F,S}, K_{F,U})$
$K_{F,\bar{U}} = \frac{2\pi E_{F,\bar{U}}}{h\nu_F}$	$K_{F,U} = \frac{2\pi E_{F,U}}{h\nu_F}$
$K_{F,S} = \frac{2\pi E_{F,S}}{h\nu_F}$	

where W is the device width, while $K_{F,\bar{U}}$, $K_{F,S}$, $K_{F,U}$ are the Fermi wave vectors on the graphene regions above the back-gates $\bar{U}$, S and U respectively; all them depend on the Fermi energy E_F whose level is function of the voltage V_g applied at the corresponding back gate as:

$$E_F = \frac{1}{\gamma t_{ox}} \left(\sqrt{\epsilon^2 + 2\gamma\epsilon q |V_g| t_{ox}} - \epsilon \right) \quad (3)$$

In the above equation t_{ox} is the oxide thickness, ϵ is the permittivity of the oxide, the constant $\gamma = 4\pi q^2/h^2\nu_F^2$, h is the Planck's constant, ν_F is the Fermi velocity which is of the order of 10^6m/s.

The R_{pn} can be estimated including the transmission probability T_θ of the carriers across the pn-junction:

$$R_{pn} = \frac{R_o}{N_{ch} T(\theta)} \quad (4)$$

$T(\theta)$ depends on (i) the angle θ between the electron's wave vector (K_F) and the normal of the junction and (ii) the distance D of the transition regions (please refer to back view in Figure 2). Equation 5 gives the expression for $T(\theta)$.

$$T(\theta) = \begin{cases} \cos^2(\theta) e^{-\pi D min(K_{F,S}, K_{F,\bar{U}}) \sin^2\theta} & \text{for } R_{AZ} \\ \cos^2(\theta) e^{-\pi D min(K_{F,S}, K_{F,U}) \sin^2\theta} & \text{for } R_{BZ} \end{cases} \quad (5)$$

$K_{F,\bar{U}}$, $K_{F,S}$, $K_{F,U}$ are the same used for N_{ch}, while θ is 45^o by construction.

The electrical model also includes parasitics of the metal contacts. R_c (assumed to be 10Ω) represents the resistance of a metal-to-graphene contact [13] and it has been applied at the three front metal contacts, A, B and Z.

The lumped capacitance C_g at the back-gates (i.e., $C_{g\bar{U}}$ at $\bar{U}$, C_{gS} at S, C_{gU} at U) consists of the series of the oxide capacitance C_{ox} and the quantum capacitance of 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 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 (Equation 3) and its value per unit of area is given by $C_q = \gamma E_F$.

3. DELAY MODEL

Figure 4 shows the four different timing arcs in the RG-MUX gate: from the select pin S to the output Z (*S-to-Z*); from the reconfiguration pin U to the output pin Z (*U-to-Z*), from input data pins A or B to output pin Z (*A-to-Z* or *B-to-Z*).

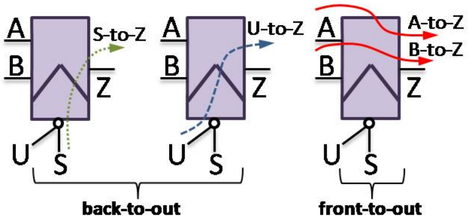


Figure 4: In-to-out RG-MUX timing arcs.

Thanks to the symmetric structure of the device (i) all the timing arcs show same low-to-high (LH) and high-to-low (HL) propagation delays and (ii) the arcs *A-to-Z* and *B-to-Z* are electrical equivalent, hence, they have same propagation delay. For these properties, we can simplify the modeling procedure by focusing on LH delays only and ignoring the equivalent path *B-to-Z* during characterization.

Another important property is that the timing behavior along the arcs is primarily determined by the type of terminals involved. This allows us to identify two main categories: *back-to-out* arcs, i.e., those involving the back-gates (*S-to-Z* and *U-to-Z*), which show a non-linear relationship w.r.t. input transition times t_{in} and capacitive load C_l; *front-to-out* arcs, i.e., those involving the front metal contacts (*A-to-Z* and *B-to-Z*), which show a linear behavior w.r.t. C_l only. For the former category we use a non-linear delay model (NLDM), whereas a linear delay model (LDM) is successfully employed for the latter category.

3.1 NLDM for back-to-out timing arcs

Table 1 resumes LH back-to-out timing arcs.

Table 1: Back-to-out timing arcs.

	A	B	S	U	Z
S-to-Z	0	1	0→1	1	0→1
U-to-Z	0	1	0	0→1	0→1

As introduced in Section 2, the device shows an RC electrical behavior. As per the Elmore delay model [5] the propagation delay Dp is proportional to $R_{device} \cdot C_l$, with R_{device} the in-to-out device resistance and C_l the output load capacitance at pin Z. R_{device} is not a constant, but it changes dynamically with the voltages applied at the back gates S and U, like the channel resistance of MOS transistors changes with the gate voltage. The input signals are modeled by a linear ramp with slope that varies linearly with the transition time t_{in}. This results into $Dp \propto R_{device}(t_{in}) \cdot C_l$, which is a non linear equation that well fits with the non-linear delay model (NLDM) Synopsys employed for CMOS gates characterization [4]:

$$Dp(t_{in}, C_l) = A \cdot t_{in} + B \cdot C_l + C \cdot t_{in} \cdot C_l + D \quad (6)$$

Each timing arc has four coefficients associated with it (A, B, C, D in Equation 6) that are calculated using the bilinear interpolation method pictorially described in Figure 5. A 2D lookup table (LUT) stores the delay measures obtained from SPICE. Each entry in the table represents the delay of the timing arc at predefined points in the $\{t_{in}, C_l\}$ characterization space. When determining the response of the device for a given coordinate (t_{in-x}, C_{l-x}), the four points in the table that bound the desired coordinate are determined: (t_{in1}, C_{l1}), (t_{in1}, C_{l2}), (t_{in2}, C_{l1}), and (t_{in2}, C_{l2}) in Figure 5. The four coordinates determine four entries in the characterization table, i.e., f_{11}=LUT(t_{in1}, C_{l1}), f_{12}=LUT(t_{in1}, C_{l2}), f_{21}=LUT(t_{in2}, C_{l1}), f_{22}=LUT(t_{in2}, C_{l2}) in Figure 5.

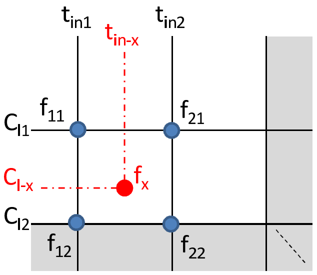


Figure 5: Bilinear interpolation.

The coefficients A, B, C and D of the characteristic equation that describes the gate's delay in the region enclosed in the square (f_{11}, f_{12}, f_{21}, f_{22}) are finally calculated by solving the system of equations reported below:

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{21} \\ f_{22} \end{bmatrix} = \begin{bmatrix} t_{in1} & C_{l1} & t_{in1} \cdot C_{l1} & 1 \\ t_{in1} & C_{l2} & t_{in1} \cdot C_{l2} & 1 \\ t_{in2} & C_{l1} & t_{in2} \cdot C_{l1} & 1 \\ t_{in2} & C_{l2} & t_{in2} \cdot C_{l2} & 1 \end{bmatrix} \times \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \tag{7}$$

which results in the following regression formulas:

$$\begin{cases} A = \frac{C_{l2}(f_{21}-f_{11})+C_{l1}(f_{12}-f_{22})}{(t_{in2}-t_{in1})(C_{l2}-C_{l1})} \\ B = \frac{t_{in2}(f_{12}-f_{11})+t_{in1}(f_{21}-f_{22})}{(t_{in2}-t_{in1})(C_{l2}-C_{l1})} \\ C = \frac{(f_{11}-f_{12})+(f22-f21)}{(t_{in2}-t_{in1})(C_{l2}-C_{l1})} \\ D = \frac{(t_{in2}C_{l2}f_{11}-t_{in1}C_{l2}f_{21}-t_{in2}C_{l1}f_{12}+t_{in1}C_{1}f_{22})}{(t_{in2}-t_{in1})(C_{l2}-C_{l1})} \end{cases} \tag{8}$$

For CMOS technologies the selection of the data stored in the 2D characterization LUT plays an important role in determining the accuracy of the model. This is due to the fact that the error on the NLDM coefficients may increase significantly when the four coordinates used as interpolation points (f_{11}, f_{12}, f_{21}, f_{22}) are too far from the coordinate of interest (t_{in-x}, C_{l-x}). Obviously, for a given characterization range, i.e., $t_{in} \in [t_{in-min} : t_{in-max}]$; $C_l \in [C_{l-min} : C_{l-max}]$, the larger the number of entries in the LUT, the more accurate the delay estimation [4]. However, as we show later in the experimental section, this issue is immaterial for the RG-MUX. We prove that using the four coefficients calculated at the most outer corners of the characterization space, i.e., (t_{in-min}, C_{l-min}), (t_{in-min}, C_{l-max}), (t_{in-max}, C_{l-min}), (t_{in-max}, C_{l-max}), we get negligible errors across the entire range, whatever the size of the LUT. This represents a key feature that might enable LUT-free static timing analysis in future RG-MUX based designs flows.

3.2 LDM for front-to-out timing arc

Table 2 resumes the two equivalent LH front-to-out timing arcs.

Table 2: Front-to-out timing arcs.

	A	B	S	U	Z
A-to-Z	0→1	0	0	1	0→1
B-to-Z	0	0→1	1	1	0→1

As one can observe from the table, both S and U are kept at fixed values. Hence, differently from the back-to-out arcs, the resulting device resistance R_{device} is a constant, that is, it does not change with the input signals and their transition time t_{in}. This results into $Dp \propto R_{device} \cdot C_l$, which is a linear equation that well fits with a linear delay model (LDM):

$$Dp(C_l) = E \cdot C_l + F \tag{9}$$

The bilinear interpolation described for back-to-out arcs reduces to a simple one-dimension linear interpolation. A 1D LUT stores the delay measures obtained from SPICE simulations. Each entry represents the delay at predefined C_l points in the characterization space. When determining the response of the device for a given C_{lx}, the two points in the table that bound the desired coordinate are determined, i.e., C_{l1} and C_{l2}. The corresponding delay values at these points, i.e., $f_1 = LUT(C_{l1})$ and $f_2 = LUT(C_{l2})$, are then used to calculate the coefficients E and F using the following formulas:

$$\begin{cases} E = \frac{(f_2-f_1)}{(C_{l2}-C_{l1})} \\ F = f_1 - E \cdot C_{l1} \end{cases} \tag{10}$$

As for *back-to-out* arcs, we show that the optimal selection of the C_l points used for the construction of the characterization LUT does not represent an issue.

4. MODEL VALIDATION

4.1 Methodology flow

Figure 6 shows the flow we implemented for testing and validating the proposed timing models.

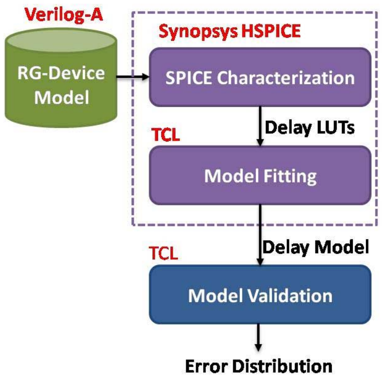


Figure 6: Methodology flow and tools.

For the simulation we used a standard SPICE netlist in which the RG-MUX, instantiated with a macro containing the Verilog-A model presented in Section 2, is driven by piece-wise linear voltage sources that emulate input signals. The characterization spans in the range $t_{in} \in \{10ps : 25ps\}$, $C_l \in \{1e^{-20}F : 1e^{-16}F\}$. A total of 50 equally spaced points in these ranges are considered for characterization. The output generated by Synopsys HSPICE are then collected in dedicated 50x50 LUTs; each timing arc has a LUT associated with it.

The interpolation methods presented in Section 3 have been integrated within procedures written in TCL, the de-facto standard language for modern EDA tools. The model fitting tool takes as input the characterization LUT of the timing arc under analysis and returns the model coefficients for a specific t_{in-x}, C_{l-x} point of interest.

The model validation stage consists of dedicated algorithms that calculate the error distribution across the characterization LUT when using a fixed set of coefficients. The main goal is to understand if it is possible to find a fixed set of coefficients that work over the entire characterization space with a reasonable error. In other words, we want to check if is possible to implement LUT-free timing analysis.

Algorithms 1 and 2 report the pseudo-code used in the validation stage for *back-to-out* and *front-to-out* timing arcs respectively. Let us first consider the former case, i.e., Algorithm 1. It takes as input the characterization LUT, and two arrays Cl and tin that contains the indices of the LUT. At each iteration of the **for** loop i, the model coefficients are calculated using as entries the four corners ($f11$, $f12$, $f21$, $f22$ in the code) of a submatrix of length i having the top-left corner stacked in the origin ($f11 = LUT(tin(1), Cl(1))$). The size of the submatrix increases with i from 2x2 to 50x50. The obtained coefficients (line 8) are then used in the two inner **for** loops j and k that iterate over all the points in the LUT for calculating the average and peak interpolation errors. The error is defined as the the difference between

Algorithm 1 NLDM error analysis

```
1: INPUT: LUT, tin, Cl, LUT_Size;;
2: for i = 2 to LUT_Size do
3:   f11=LUT[tin(1),cl(1)];
4:   f12=LUT[tin(1),cl(i)];
5:   f21=LUT[tin(i),cl(1)];
6:   f22=LUT[tin(i),cl(i)];
7:   calculate A, B, C, D;
8:   err_total=0;
9:   err_max=0;
10:   for j=1 to LUT_Size do
11:     for k=1 to LUT_Size do
12:       Dp= A*tin(k)+ B*Cl(j)+ C*tin(k)*Cl(j)+ D;
13:       err=abs((Dp - LUT(j,k));
14:       err_tot = err_tot + err;
15:       if err>err_max
16:           err_max=err;
17:       end if
18:     end for
19:   end for
20:   Avg_Err[i]=err_tot/(LUT_Size*LUT_Size);
21:   Peak_Err[i]=err_max;
22: end for
```

the delay calculated using NLDM (line 13) and that measured with HSPICE ($LUT(j,k)$). Avg_Err and $Peak\_Err$ and then stored in two dedicated arrays (line 21, 22).

Algorithm 2 refers to *front-to-out* timing arcs for which the propagation delay depends only on the load Cl. The code works as for the NLDM case, but submatrix now reduces to a 1D array (tin is now fixed). At each iteration of the **for** loop i, the two model coefficients are calculated using as entries the first and last items ($f1$ and $f2$) of a subarray of length i having the first item stacked at 1 ($f1 = LUT(Cl(1))$). The size of the array increases with i from 2 to 50. The inner **for** loop j is then used to calculate average and peak errors across the 1D LUT. Avg_Err and $Peak\_Err$ and then stored in two dedicated arrays.

Algorithm 2 LDM error analysis

```
1: INPUT: LUT, Cl, LUT_Size;
2: for i = 2 to LUT_Size do
3:   f1=LUT(Cl(1));
4:   f2=LUT(Cl(i));
5:   calculate E, F;
6:   err_tot=0;
7:   err_max=0;
8:   for j=1 to LUT_Size do
9:     Dp= E*cl(j)+ F;
10:     err=abs((Dp - LUT(j,1));
11:     err_tot=err_tot+err;
12:     if err>err_max
13:         err_max=err;
14:     end if
15:   end for
16:   Avg_Err[i]=err_tot/(LUT_Size);
17:   Peak_Err[i]=err_max;
18: end for
```

4.2 Results

As intuitively described in Section 3, due to the symmetric structure of the device (i) timing arcs have same LH and HL delays and (ii) the arcs *A-to-Z* and *B-to-Z* are equivalent. To sustain this claim, Table 3 shows the delays of the *front-to-out* arcs measured with HSPICE for min/max input transition times (columns t_{in-min} and t_{in-max}) and min/max load capacitance (columns C_{l-min} and C_{l-max}) of the characterization space. It is worth emphasizing that the obtained data are compliant with the numbers reported in [3]. It can be inferred that (i) LH and HL delays are the same, and (ii) there is no delay difference between the two *front-to-out* timing arcs, therefore, the delay model and the model coefficients that hold for *A-to-Z* are also valid for *B-to-Z*. As additional comment, one can observe that delays are independent from t_{in}, therefore, we can safely apply the LDM.

Table 3: LH and HL delays for front-to-out arcs.

		t_{in-min}		t_{in-max}	
		A-to-Z	B-to-Z	A-to-Z	B-to-Z
C_{l-min}	LH	1.09e-14	1.09e-14	1.06e-14	1.06e-14
	HL	1.09e-14	1.09e-14	1.06e-14	1.06e-14
C_{l-max}	LH	13.8e-14	13.8e-14	13.8e-14	13.8e-14
	HL	13.8e-14	13.8e-14	13.8e-14	13.8e-14

A more interesting analysis is graphically proposed in the Figures 7, 8, 9, which show the average (top) and peak (bottom) error distribution as returned from Algorithms1 and 2, i.e., the arrays $Avg\_Err[i]$ and $Peak\_Err[i]$.

For all the timing arcs (*S-to-Z* in Figure7, *U-to-Z* in Figure8, and *A-to-Z* in Figure 8) the percentage of error decreases for larger i, i.e., as the square length of the submatrix (subarray for LDM) get larger. Minimum average and peak errors, obtained for i=50, are always below 0.6% and 3.1% respectively, that is, accurate enough to integrate these models along with their coefficients in future LUT-free STA tools for graphene digital ICs.

Table 4 shows the model coefficients that guarantee the minimum error, namely, those extracted using as interpolation points the most outer corner of the characterization LUT (i.e., i=50). Notice that the arcs *S-to-Z* and *U-to-Z* use NLDM, hence have coefficients A, B, C and D, whereas *A-to-Z* and *B-to-Z* uses LDM which has two coefficients E and F. Since the RG-MUX is symmetric w.r.t. inputs A and B, the coefficients for the two *front-to-out* timing arcs are same.

Table 4: Model coefficients for Low-to-High and High-to-Low in-to-out propagation delays.

Back-to-out	A	B	C	D
S-to-Z	0.36	$6.30 \cdot 10^{3}$	-$4.53 \cdot 10^{13}$	-$2.00 \cdot 10^{-14}$
U-to-Z	0.36	$4.38 \cdot 10^{3}$	$2.13 \cdot 10^{13}$	$6.22 \cdot 10^{-16}$
Front-to-out	E	F	-	-
A-to-Z	1380	$-1.7 \cdot 10^{-16}$	-	-
B-to-Z	1380	$-1.7 \cdot 10^{-16}$	-	-

As further proof of model accuracy, Figure 10 finally plots model (dotted lines) vs. HSPICE (discrete points) as function of C_l; model coefficients are those reported in Table4. It can be seen that both the NLDM and the LDM well match with the SPICE responses of the device.

5. CONCLUSIONS

In this paper we applied conventional LUT-based timing modeling approaches to identify the delay model of a new class of pn-junction based reconfigurable graphene devices.

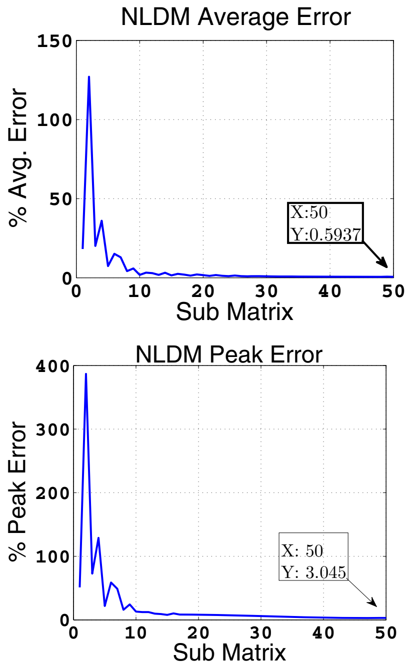


Figure 7: S-to-Z: average (top) and peak (bottom) error

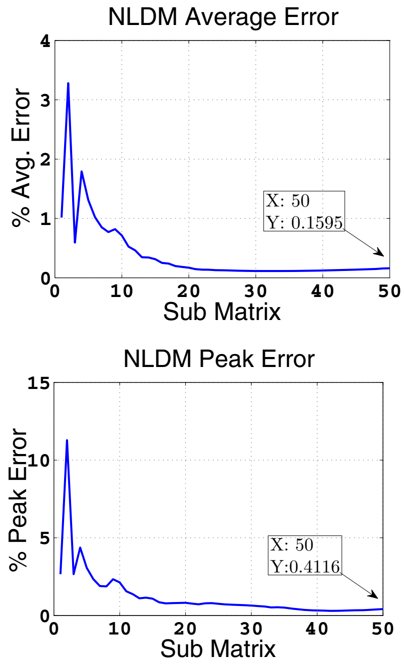


Figure 8: U-to-Z: average (top) and peak (bottom) error

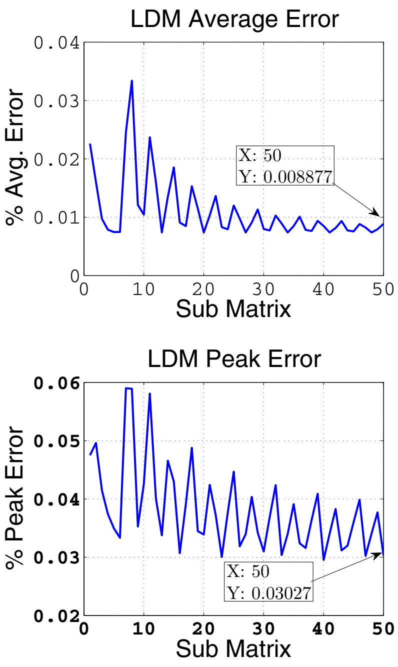


Figure 9: A-to-Z: average (top) and peak (bottom) error

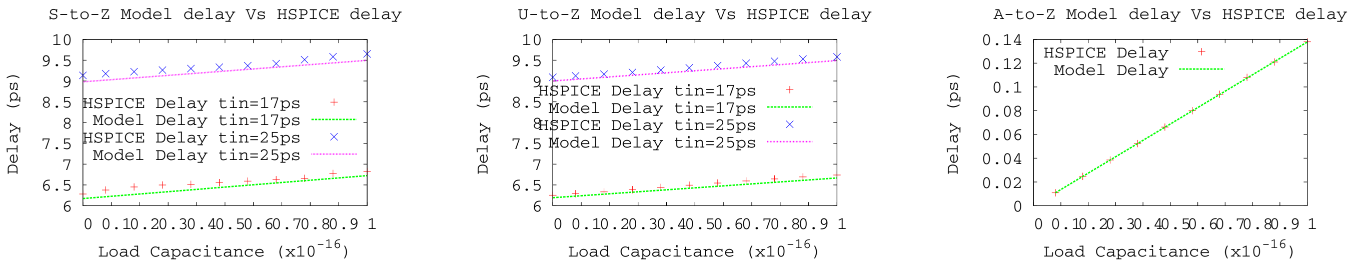


Figure 10: Model vs. HSPICE: S-to-Z(left), U-to-Z(center), A-to-Z(right)

We implemented a dedicated strategy that allowed us to extract the regression coefficients of the models and validate them for a wide range of input signal transition times and load capacitances.

The analysis has shown that the proposed model allows accurate delay estimation for all the in-to-out timing arcs of the device (average and peak estimation errors below 0.6% and 3.1%) with no need of storing large characterization LUT as typically done in todays' CMOS timing libraries.

6. REFERENCES

[1] 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*, 81:109–162, Jan. 2009.

[2] 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*, 315(5816):1252–1255, 2007.

[3] 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*, pages 262–269, Mar. 2012.

[4] J. F. Croix and D. F. Wong. Fast and accurate technique to optimize characterization tables for logic synthesis. In *DAC'97: ACM/IEEE Design Automation Conference*, pages 337–340, Jun. 1997.

[5] W. C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of Applied Physics*, 19(1):55–63, Jan. 1948.

[6] M. I. Katsnelson, K. S. Novoselov, and A. K. Geim. Chiral tunneling and the klein paradox in graphene. *Nature Physics*, 2(9):620–625, Sep. 2006.

[7] S. Miryala, M. Montazeri, A. Calimera, E. Macii, and M. Poncino. A verilog-a model for reconfigurable logic gates based on graphene pn-junctions. In *DATE'13: ACM/IEEE Design, Automation and Test in Europe*, pages 1–4, Mar. 2013.

[8] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, M. I. Katsnelson, I. V. Grigorieva, S. V. Dubonos, and A. A. Firsov. Two-dimensional gas of massless dirac fermions in graphene. *Nature*, 438(7065):197–200, 2005.

[9] 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*, 306(5696):666–669, 2004.

[10] 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*, pages 10451–10453, 2005.

[11] H.-S. Philip Wong and D. Akinwande. *Carbon nanotubes and graphene device physics*. Cambridge University Press, 2010.

[12] R. Saito, G. Dresselhaus, and M. Dresselhaus. *Physical properties of carbon nanotubes*. World Scientific, 1998.

[13] 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*, pages 883–888, Jun. 2010.