Exploiting the Expressive Power of Graphene Reconfigurable Gates via Post-Synthesis Optimization

Sandeep Miryala, Valerio Tenace, Andrea Calimera, Enrico Macii, Massimo Poncino
Politecnico di Torino
Torino, Italy
andrea.calimera@polito.it

Luca Amarú, Giovanni De Micheli, Pierre-Emmanuel Gaillardon
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
pierre-emmanuel.gaillardon@epfl.ch

ABSTRACT

As an answer to the new electronics market demands, semiconductor industry is looking for different materials, new process technologies and alternative design solutions that can support Silicon replacement in the VLSI domain. The recent introduction of graphene, together with the option of electrostatically controlling its doping profile, has shown a possible way to implement fast and power efficient Reconfigurable Gates (RGs). Also, and this is the most important feature considered in this work, those graphene RGs show higher expressive power, i.e., they implement more complex functions, like Majority, MUX, XOR, with less area w.r.t. CMOS counterparts. Unfortunately, state-of-the-art synthesis tools, which have been customized for standard NAND/NOR CMOS gates, do not exploit the aforementioned feature of graphene RGs.

In this paper, we present a post-synthesis tool that translates the gate level netlist obtained from commercial synthesis tools to a more optimized netlist that can efficiently integrate graphene RGs. Results conducted on a set of open-source benchmarks demonstrate that the proposed strategy improves, on average, both area and performance by 17% and 8.17% respectively.

Categories and Subject Descriptors

B.5.2 [**Design Aids**]: Optimization

Keywords

Graphene, P-N junction, Reconfigurable Gate, Synthesis, Optimization

1. INTRODUCTION

Graphene is a single atomic layer of carbon atoms arranged in a honeycomb lattice structure. It was first isolated by mechanical exfoliation in 2004 by researchers at the University of Manchester [1] after decades of experiments from the scientists across the world.

GLSVLSI'15, May 20–22, 2015, Pittsburgh, PA, USA

http://dx.doi.org/10.1145/2742060.2742098

Although the mechanical exfoliation process is effective for research purposes, it is not suitable for industries. Thereafter, significant progress in research has led to other methods, e.g., SiC decomposition [2, 3]. The latest one, being Chemical Vapor Deposition (CVD), has finally reported to be a viable solution for mass-production [4].

High carrier mobility and high saturation velocity are what make graphene a perfect material for electronic devices. However, it also shows some drawbacks; the zero band-gap energy distribution, in particular, as the most critical. The latter results in heavy leakage currents when the material is turned-OFF, and hence, poor I_{on}/I_{off} current ratio. Needless to say this poses severe limitations for digital applications which need a clear separation between 0- and 1-logic. One possible way to open a band-gap is given by lithographic patterning of large graphene sheets into narrow stripes ($\sim 10nm$), called Graphene Nano Ribbons (GNRs) [5]. However, the patterning process alter the lattice structure of the material and degrades carrier mobility. An alternative strategy to implement graphene switches is to exploit its intrinsic properties rather than modifying them, that is, one can effectively steer carriers across pristine sheets of graphene by means of external voltage potentials as to build equivalent P-N junctions [6]. The latter, when properly connected, can implement reconfigurable logic gates [7], RGs hereafter, which show exceptional properties w.r.t. CMOS gates.

Just few recent works deal with, or are related to, graphene P-N junctions and RGs. In [8, 9], the authors explore multiple architectures for basic logic gates using graphene RGs and characterize them from power and performance, where as [10, 11] discusses the associated models for delay and power for each of the possible timing arcs for graphene RG logic gates. In [12] the authors propose the interesting comparison among various implementation styles for graphene RG-based circuits. From a CAD standpoint, in [13, 14] the authors proposed a novel data structure on the lines of BDD called Biconditional Binary Decision Diagrams (BBDD) which is suitable for representing RG-based circuits.

Till now, graphene RGs were used for realizing elementary Boolean logic gates, like AND/OR/INV/BUF, but not for complex functions, e.g., Majority (MAJ), multi-inputs XOR and others, like XOR followed by AND (XOR-AND) and XOR followed by another XOR (3-input XOR), all of them very common in many arithmetic circuits. Obviously the right use of such complex gates during synthesis and optimization may result in a drastic reduction of circuit complexity and better figures of merit, as area, performance and power.

Unfortunately, commercial synthesis tools are currently customized for managing Sum-of-Product forms, namely, they have been optimized for AND-OR-INV standard cells mapping; hence do not make, or rarely do, use of those complex functions achievable with RGs, even if included in the reference libraries used at the synthesis stage. This work addresses the above limitation proposing a new post-synthesis tool, currently part of a library package written in C, that parses a logic netlist collected from a standard synthesis flow and automatically recognizes and maps unusual complex functions by means of a structural matching&covering procedure. Such mapping is done as to exploit the maximum expressive power of RGs. With this method, it is possible to reduce not only total number of gates (17% on average) but also improve the average performance (8.17%) of graphene integrated circuits.

2. GRAPHENE RECONFIGURABLE GATE

A graphene reconfigurable logic gate, 3D view given in Figure 1-(a), consists of four layers [15]. The bottom layer has three metal back-gates ($\overline{U}$, S and U) isolated each other by oxide ($\approx 18nm$). A thick oxide layer is growth on top of these three back-gates, while a graphene sheet is deposited on top. Three front metal-to-graphene contacts (A, Z, B) are then connected to the graphene sheet. The central front-contact Z acts as an output pin and the other two front-contacts, namely A and B, will serve as inputs to the reconfigurable logic gate. Notice that the triangular form of the back-gates (bottom view in Figure1-(b)) is fundamental for proper functionality of the RG; interested reader can refer to [15] for more details.
The graphene material adapts its doping profile based on the polarity of the back-gate potentials, hence it requires bipolar voltages: $+V_{dd}/2$ to form n-type; $-V_{dd}/2$ to form p-type regions [16]. This infers logic '1'corresponds to $+V_{dd}/2$ and logic '0'corresponds to $-V_{dd}/2$.

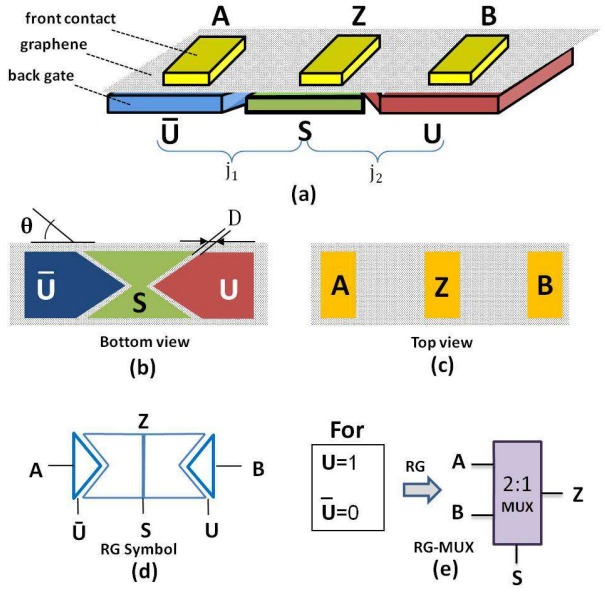

Figure 1: Structure of reconfigurable logic gate

An RG is made of two adjacent junctions (j_1 and j_2 in Fig. 1-(a)), the first between A-Z, the other between B-Z. It operates according to the type of junctions formed (i.e., p-n/n-p or p-p/n-n); alike potentials fed to two adjacent back-gates induce a p-p/n-n junction, while, opposite polarities form a p-n/n-p junction. The cross resistance of p-p/n-n is $\simeq 300\Omega$ (the ON State) while as for a p-n/n-p junction is $\simeq 3 \cdot 10^5 \Omega$ (the OFF state) [7]. Depending on the doping configuration, the inputs carriers injected at the two front-contacts A and B eventually reach the output front-contact Z. The back-gates U and $\overline{U}$ receives complementary potentials as to guarantee that only one of the two junctions can be ON at the same time and thus to avoid short-circuits across inputs A and B.
From a functional viewpoint the behavior of an RG is summarized as follows:

- When S and U receive opposite voltage signals the junction j_1 between A-Z is ON, while j_2 between B-Z is OFF. Hence, the output potential at Z follows that the input at A. This can be written in boolean form as $(S \oplus U) \cdot A$. Where $\oplus$ is represents the XOR operator.
- When S and U receive same voltage signals, the junction j_1 between A-Z is OFF, while the junction j_2 between B-Z is ON. Hence, the output Z follows the input at B. This can be written in boolean form as $(S \odot U)B$. Where $\odot$ is XNOR operator.
- The overall functionality of the Graphene Reconfigurable gate can be written as $Z = (S \oplus U) \cdot A + (S \odot U) \cdot B$

In order to support electrical simulation, we implemented a Verilog-A macro whose model card is integrated within SPICE simulator. Such a model is borrowed from [7] and [8] to which interested reader can refer for more accurate details.

3. DIGITAL LIBRARY WITH RGS

3.1 Basic Boolean Functions

The RG is used as a primitive to build basic 2-input logic gates [8]. Those used in this work, i.e., INV, OR, AND, XOR, have been summarized in Figure 2. All of them except the XOR, are implemented using a single RG, whereas the XOR needs two RGs. It is worth noticing that terminals U and $\overline{U}$ (not reported in the picture) are fed fixed voltages, i.e., $+V_{dd}/2$ and -$V_{dd}/2$ respectively.

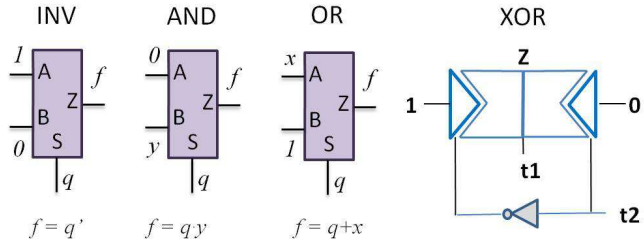

Figure 2: Input configuration for 2-inputs logic gates implementing basic Boolean functions.

Needless to say, any logic function can be implemented with proper connection of basic standard cells. However, a less intuitive, yet more efficient implementation, would better exploit the expressive power of RGs by using the terminals U and $\overline{U}$ as logic inputs; we refer to this implementation as the RG-EXP implementation. In the next subsections we

report a quantitative analysis of some complex logic functions typically used in arithmetic circuits, implemented as cascade of standard cells, the STC style, or the RG-EXP style.

3.2 MUltipleXer (MUX)

Implementing the 2:1 MUX using STC, as shown in Fig. 3(b), requires four RGs with total depth of three levels. Note that depth can be used as an estimation of the timing criticality. However, a single RG can naturally implement a 2:1 MUX by just fixing the potential of U to logic '1'; this represents the RG-EXP implementation illustrated in Fig. 3 (a). The area ratio is therefore 4:1, being the RG-EXP 75% more area efficient. The performance comparison between STC

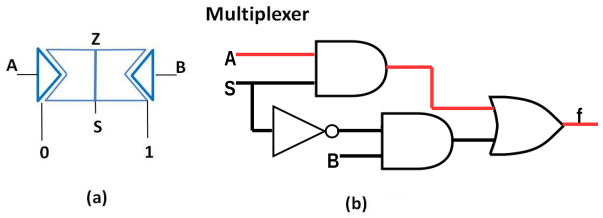

Figure 3: MUX using RG-EXP and STC.

and RG-EXP is reported in Table. 1, where the characterization is done for different input transition time (T_r) with fanout varying from 1 to 5. As can be seen RG-EXP performs substantially better, 55% of delay reduction in the best case (FO=1).

Table 1: Multiplexer function characterization using RG-EXP and STC

Tr (Sec)	2.85E-12		5.65E-12	
FanOut	RG-EXP	STC	RG-EXP	STC
1	9.20E-13	1.87E-12	1.10E-12	2.03E-12
2	1.58E-12	2.47E-12	1.85E-12	2.63E-12
3	2.21E-12	3.06E-12	2.53E-12	3.21E-12
4	2.82E-12	3.64E-12	3.18E-12	3.80E-12
5	3.42E-12	4.22E-12	3.81E-12	4.37E-12

3.3 MAJority Function (MAJ)

Fig. 4 shows the implementation of the majority function, i.e., $f(t1,t2,t3)_{MAJ} = t1 \cdot t2 + t2 \cdot t3 + t3 \cdot t1$, using STCs (a) and RG-EXP (b). Concerning the RG-EXP one, let us explain its functionality by examples. Consider the input configuration of the three inputs $t1$, $t2$ and $t3$ being "101". For such pattern the junction between A-Z is ON (as $t1'$=0 and $t2$=0), while the junction between B-Z is OFF (as $t1$=1 and $t2$=0). Hence the output voltage at Z follows that of the input signal at A, i.e., $t3$, which happens to be '1', which is the right value of the MAJ function when 2 or more inputs are at '1'. Similar reasoning can be done for remaining input patterns. The STC requires four RGs with total depth of tree levels; the area ratio is therefore 4:2, i.e., 50% area savings.
Concerning performance, the Table. 2 shows the delay characterized for different input transition time (T_r) and output load (FanOut). As for the 2:1 MUX gate, the RG-EXP has better performance irrespective of the fanout, 45% of delay reduction in the best case (FO=1), due to lower depth.

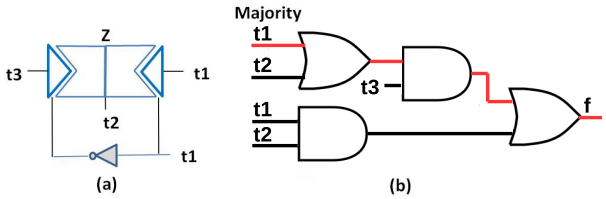

Figure 4: MAJ using RG-EXP and STC.

Table 2: Majority function characterization using RG-EXP and STC

Tr (Sec)	2.85E-012		5.65E-012	
FanOut	RG-EXP	STC	RG-EXP	STC
1	1.29E-012	2.24E-012	1.64E-012	2.63E-012
2	2.08E-012	2.88E-012	2.56E-012	3.31E-012
3	2.78E-012	3.49E-012	3.41E-012	3.96E-012
4	3.43E-012	4.09E-012	4.18E-012	4.60E-012
5	4.05E-012	4.69E-012	4.92E-012	5.22E-012

3.4 XOR-AND

Another complex function that can be efficiently implemented through graphene RG is the XOR-AND (i.e an XOR gate followed by an AND gate), which is a common path in arithmetic circuits. Fig. 5 shows the implementation of XOR-AND with STCs (b) and RG-EXP (a). To explain functionality, consider the combination of the three inputs $t1$, $t2$ and $t3$ to be "101". For such pattern the junction between A-Z is ON (as $t1'$=0, $t2$=0), whereas the junction between B-Z is OFF (as $t1'$=1, $t2$=0). Hence, the signal at Z follows the input on A, i.e., t3, which happens to be '1'. Similar reasoning can be done for remaining input patterns.

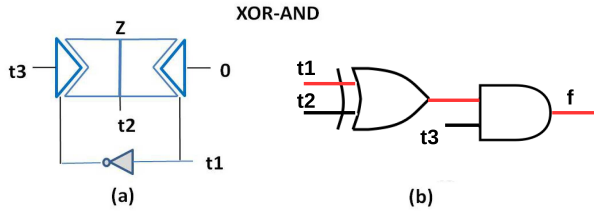

Figure 5: XOR-AND realization using RG-EXP and STC.

The area ratio between STC and RG-EXP is 1:1, i.e., no area savings. Concerning performance, the delay characterized for different input transition time (T_r) and output load (FanOut) is reported in Tab. 3. Again, the RG-EXP shows better performance, 35% in the best case (FO=1).

3.5 XOR-3

It is possible to realize a 3-inputs XOR gate by using a single RG. Fig. 6 shows the implementation of XOR-3 with STCs (b) and RG-EXP (a). Similar to previous cases different inputs patterns turn-ON/OFF one of the two junctions connecting the right input signal to output depending on the input pattern.
The STC implementation requires two XOR gates, each of them requiring two RGs (please refer to Fig.2). The resulting area ratio is 3:4, i.e., 25% area savings. In the best case, the performance savings achieved with RG-EXP is 25% (FO=1), but, due to the presence of an INV at the front contacts, the propagation delay of RG-EXP is worse for FO larger than 3. This suggest that the mapping of 3-inputs

Table 3: XOR-AND function characterization using RG-EXP and STC

Tr (Sec)	2.85E-012		5.65E-012	
FanOut	RG-EXP	STC	RG-EXP	STC
1	1.01E-12	1.50E-12	1.15E-12	1.64E-12
2	1.69E-12	2.10E-12	1.75E-12	2.24E-12
3	2.17E-12	2.68E-12	2.34E-12	2.83E-12
4	2.75E-12	3.27E-12	2.92E-12	3.41E-12
5	3.33E-12	3.85E-12	3.50E-12	3.99E-12

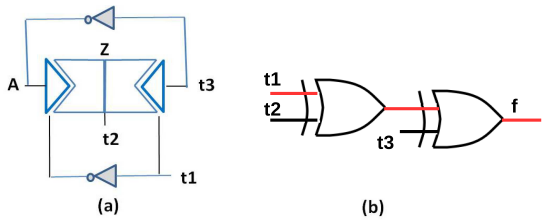

Figure 6: XOR-3 realization using RG-EXP and STC.

XOR with RG-EXP is convenient depending on the current FO.

Table 4: Three input XOR function characterization using RG-EXP and STC

Tr	2.85E-012		5.65E-012	
FanOut	RG-EXP	STC	RG-EXP	STC
1	1.23E-012	1.64E-012	1.49E-012	1.73E-12
2	2.03E-012	2.22E-012	2.46E-012	2.31E-12
3	2.74E-012	2.79E-012	3.33E-012	2.88E-12
4	3.39E-012	3.37E-012	4.12E-012	3.46E-12
5	4.02E-012	3.95E-012	4.86E-012	4.04E-12

4. DESIGN FRAMEWORK FOR RG-BASED CIRCUITS

4.1 Motivational Example

To understand the need of a alternative synthesis strategy and motivate the post-synthesis optimization tool proposed in this work, we briefly report a handcraft analysis for a simple arithmetic benchmark, a 3-bit ripple-carry adder. Fig. 7 reports an abstract view (Acyclic Directed Graph) of its netlist resulted from a standard synthesis flow performed with a commercial tool. Assuming each node has a unit delay, a Static Timing Analysis (STA) returns a critical path delay of 6 units. In reality the delay of each of these nodes differs as it is a function of fanout. However, an attempt to minimize the number of nodes along the critical path increases the probability of reducing the critical path delay. The number in the square box on each edge report the worst-case arrival time of the signals. The worst critical path, highlighted using dotted line, travels from the primary input `a0` to the primary output `carry`.
The sub-tree having as root `g10` and leaves `a1` and `b1` can be covered by the MAJ function (please refer to Fig.4) providing local area and delay savings. The same hold for node `g14`, while node `g20` matches with the structure of the 3-inputs XOR and it might be replaced using the corresponding XOR-3 cell. After these few transformation the new critical path has depth of 4 and the number of standard cells is reduced from 15 to 10. Simulation results show an improvement in the performance by 20% and an area improvement by 30%.

4.2 Post-Synthesis Mapping

The tool we implemented to perform the post-synthesis optimization is written in C language; it is composed of the following components all of them integrated together:

1. Verilog parser
2. Optimization algorithm
3. Static Timing Analysis (STA) engine

Once the circuit is read by the parser, we archive information in a table-like data structure. Each entry of the table represents a single gate in the benchmark. Connectivity of the gates is conserved by means of original pin names assigned by the commercial synthesis tool.
In order to optimize area/timing features of the input circuits, we implemented optimization algorithm that identifies the RG-EXP functions as discussed in the previous section. After identifying those patterns, we replace them and possibly remove the covered STC gates forming the function in the original netlist with RG-EXP counterparts. When the internal cells covered by the RG-EXP have FO outside the local sub-tree, namely, they drive other external cells in the circuit, they are replaced with a replica as to guarantee the functionality of the circuit. From the Fig. 7, it can be seen that the output of nodes g9 and g6 has fanout of more than 1, i.e., these outputs are input to g16 and g17 respectively. Hence, we have to replicate the nodes g9 and g6 for the functionality of the circuit.

```
Data: Verilog netlist
Result: Optimized graphene-based netlist
Parse verilog file;
Compute worst-case path with out PST;
Procedure : For identifying majority function
foreach gate do
    if gate function is OR then
        if gate inputs are both AND then
            if AND input is OR then
                Instantiate a new RG-MUX device;
                Create RG-MUX connections;
                if gates in pattern has fanout = 1 then
                    Remove gates;
                end
            end
        end
    end
end
Procedure: For identifying XOR-AND function;
Procedure: For identifying XOR-3 funtion;
Procedure: For identifying Multiplexer;
Compute worst-case path after optimization;
```

Algorithm 1: Post Synthesis Tool exploiting expressive power of Graphene RG

For each of the four RG-EXPs, we implemented a different match&covering procedure that run iteratively over the circuit. In Alg. 1, we provide the pseudo-code of the entire post-synthesis algorithm, with an unrolled description

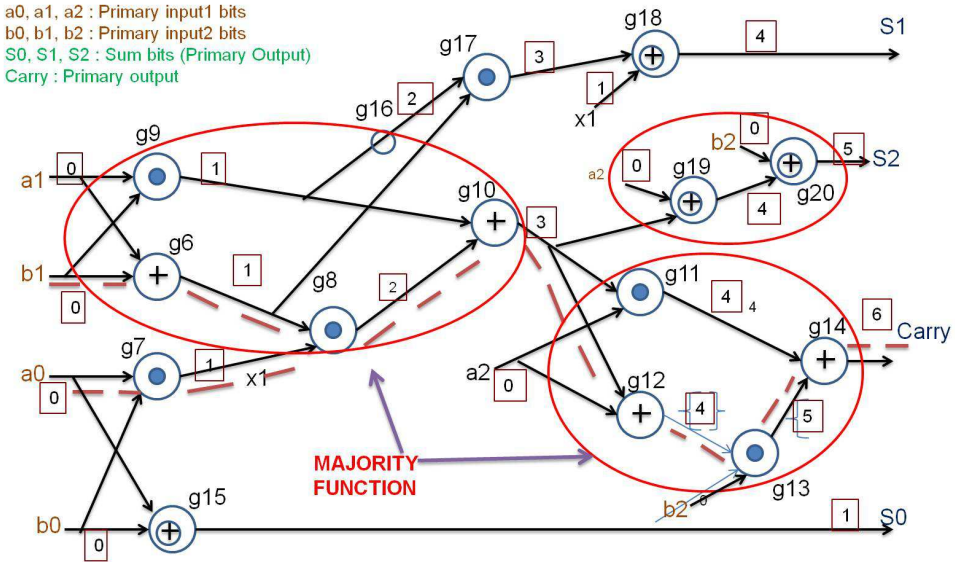

Figure 7: Directed acyclic graph of 3-bit adder.

of the procedure for recognizing MAJ functions. The latter works as follows. First it iterates among the entire set of gates in the circuit. Once an OR gate is recognized it checks whether its two inputs are both connected to AND gates. If that's true, we end up checking whether one of the two AND-gates receive an input from an OR gate. Once the pattern is recognized, it covers the sub-portion of the circuit with the RG-EXP MAJ function. The last step is to check whether it is possible to remove the gates which belong to the recognized pattern. This is possible only if the gates have fanout of 1. In the algorithm we also consider the associative property of the logic gates. In a similar way, there are other procedures to identify the other RG-EXP functions, i.e., XOR-AND, XOR-3, MUX. Finally, a STA is performed once again to identify the worst-case path and measure delay using HSPICE. The STA algorithm is straightforward, as it traverses from primary inputs to primary outputs by considering the worst case arrival-time of each node having delay of logic gates stored in a dedicated look-up table (LUT). Finally traversing the graph in the reverse order, i.e., from PO to PI, paths having maximum delay are marked critical paths.

5. SIMULATION RESULTS

5.1 Implemented Flow

The overall framework used in this work is shown in Fig. 8. The RTL code written in Verilog or VHDL is initially synthesized using conventional synthesis tools. For our work, we have adopted a commercial synthesis tool with technology libraries from STMicroelectronics at 45nm technology node. We used the reduced library set comprising of AND2, OR2, XOR2, NOT, BUF as these cells can be realized using a single Graphene Reconfigurable gate. The synthesized gate level netlist is the main input for the post-synthesis tool which identifies complex logic functions as discussed in the previous section and maps them to graphene RG-EXP. However, the gates that have fanout grater than '1' will be duplicated so that their output signal remains intact. The timing analyzer performs STA on the gate level netlist from the commercial synthesis tool and our Post Synthesis Tool (PST) to extract the critical path. This critical path is simulated in HSPICE to determine the worst-case performance of the circuit.

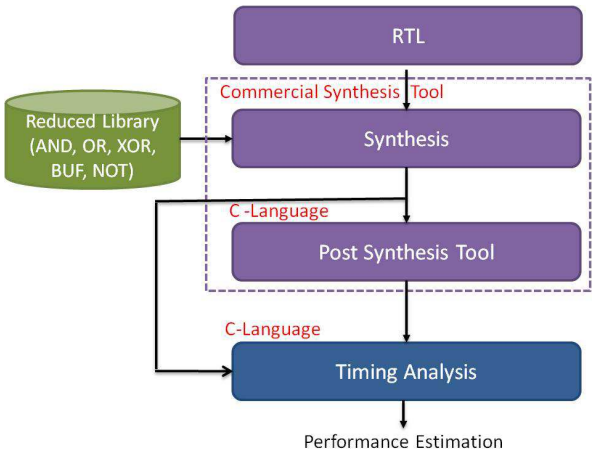

Figure 8: Simulation Framework.

5.2 Collected Results

We compare the results of various benchmarks mapped using conventional synthesis and STCs against those obtained with the proposed tool and RG-EXP.

Tab. 5, reports the number of Primary Inputs (PI) and Primary Outputs (PO) of each benchmark. For the two synthesis strategies (colomns STC and $RG-EXP$), we annotated the depth of the critical path (column *Depth*), the total number of cells (column $\#cells$) and worst-case propagation delay (column *Delay*) from 50% of input signal to 50% of the output signal; notice that delay figures have been obtained through detailed SPICE simulations. We have also annotated the number of RG-EXP functions, i.e., MAJ, MUX, XOR-AND and XOR-3 mapped after the post-synthesis optimization. Finally, column *%Imp.* represents the performance improvements resulting from the proposed synthesis strategy.

Table 5: Standard Logic Synthesis Vs. Post-Synthesis Optimization: Area/Performance Analysis

			Standard Cell Mapping			Post Synthesis Tool							
	# PI	# PO	Depth	# Cells	Delay	Depth	# Cells	Delay	# MAJ	# XOR-3	# XOR-AND	# MUX	% Imp.
Add 3bit	6	4	6	15	3.66	4	10	2.92	2	1	0	0	-20.20
Add 8bit	16	9	21	50	10.90	10	40	7.28	7	1	0	0	-33.40
Add 16bit	32	17	45	106	21.80	18	88	13.20	15	1	0	0	-39.40
Mult 4bit	8	8	19	69	10.60	14	60	8.60	3	9	1	0	-18.90
Voter 31	30	1	23	169	13.20	18	97	11.10	20	28	1	0	-16.00
c499	41	32	17	186	10.10	14	150	8.68	2	40	0	0	-14.10
cordic	23	2	17	113	9.47	16	99	8.98	0	0	13	5	-5.17
Maj. 11	11	1	9	42	6.65	7	37	5.33	1	8	4	0	-19.80
Maj. 15	15	1	11	56	6.63	9	49	5.65	1	9	7	0	-14.80
DES	256	245	26	3164	13.30	26	2117	13.30	0	2	86	4	0.00
Comp	11	3	9	40	5.38	9	38	5.38	0	0	6	0	0.00
Sin	34	63	19	1763	9.96	19	1723	9.96	0	0	0	50	0.00
ALU4	14	8	20	1572	10.40	20	1563	10.40	0	0	0	10	0.00
CLA 4bit	9	7	9	29	5.01	8	25	5.20	0	4	10	0	3.77
CLA 8bit	17	9	17	40	8.88	16	32	9.99	32	8	8	0	12.50
c1355	41	32	18	218	10.70	15	182	11.90	2	40	0	0	11.20
Adder_Big	32	16	42	242	21.10	35	230	22.10	4	16	9	1	4.65
Maj. 9	9	1	10	33	6.02	9	29	6.18	0	6	4	0	2.66
Total				**7907**			**6569**						**8.17**

Benchmarks from `Add_3bit` till `majority-15` significantly exploits utilization of complex functions and RG-EXPs. For these circuits, the benefits are not only in terms of area but also in terms of performance, as a reduction of the critical path depth corresponds to shorter propagation delay. These circuits make extensive use of MAJ functions, especially to evaluate the carry, which represents the timing critical path of the circuit.

For circuits from `DES` to `ALU4`, the main benefits are in terms of area, while the performance are not affected as no RG-EXP matching are found across the critical path.

For circuits from `CLA-4` to `Majority-9`, due to the presence of many XOR-3 gates, we register substantial area improvements at the cost of some delay penalties. What happens is that it worsens the worst-case performance of the circuit and the reason is well understood from the characterization data of XOR-3 in the previous section.

Overall, on an average, there is 17% improvement in the area of the circuit by adopting PST along with a performance improvement of 8.17%.

6. CONCLUSION AND FUTURE WORK

To exploit the expressive power of graphene RGs a PST is therefore essential or future synthesis tools has to consider these complex functions during optimization. Currently, PST optimizes the netlist to reduce area and for some circuits it reduces the depth of critical path. It may also hamper the performance for some other circuits. However, in our future work we would like to implement the timing driven approach which solely focuses on improving the performance of the circuit. Also, we would like to use the efficiency of BBDD data structure on the portion of the circuit which optimizes performance.

7. REFERENCES

[1] A. K. Geim and K. S. Novoselov, "The rise of graphene," *Nature materials*, vol. 6, no. 3, pp. 183–191, 2007.

[2] G. Deligeorgis, G. Konstantinidis, M. Dragoman, and R. Plana, "Fabrication of graphene devices, issues and prospects," in *CAS Conference*, vol. 01, pp. 21–25, Oct. 2010.

[3] C. Berger *et al.*, "Ultrathin epitaxial graphite: 2D electron gas properties and a route toward graphene-based nanoelectronics," *The Journal of Physical Chemistry B*, vol. 108, no. 52, pp. 19912–19916, 2004.

[4] C. Xu, H. Li, and K. Banerjee, "Modeling, analysis, and design of graphene nano-ribbon interconnects," *IEEE Transaction on Electron Devices*, vol. 56, no. 8, pp. 1567–1578, 2009.

[5] M. Y. Han, B. Özyilmaz, Y. Zhang, and P. Kim, "Energy band-gap engineering of graphene nanoribbons," *Phys. Rev. Lett.*, vol. 98, p. 206805, May 2007.

[6] 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.

[7] S. Tanachutiwat, J. U. Lee, W. Wang, and C. Y. Sung, "Reconfigurable multi-function logic based on graphene p-n junctions," in *DAC*, pp. 883–888, june 2010.

[8] 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 Conference*, pp. 877–880, March 2013.

[9] V. Tenace, A. Calimera, E. Macii, and M. Poncino, "Pass-XNOR Logic: A new Logic Style for PN-Junction based Graphene Circuits," in *DATE'14: ACM/IEEE Design, Automation and Test in Europe*, pp. 1–4, Mar. 2014.

[10] S. Miryala, A. Calimera, E. Macii, and M. Poncino, "Delay Model for Reconfigurable Logic Gates Based on Graphene PN-junctions," in *GLSVLSI Conference*, pp. 227–232, 2013.

[11] S. Miryala, A. Calimera, E. Macii, and M. Poncino, "Power modeling and characterization of graphene-based logic gates," in *PATMOS International Workshop*, pp. 223–226, Sept 2013.

[12] S. Miryala, A. Calimera, M. Poncino, and E. Macii, "Exploration of different implementation styles for graphene-based reconfigurable gates," in *ICICDT Conference*, pp. 21–24, May 2013.

[13] L. Amaru, P.-E. Gaillardon, and G. De Micheli, "Biconditional BDD: A novel canonical BDD for logic synthesis targeting XOR-rich circuits," in *DATE Conference Exhibition*, pp. 1014–1017, March 2013.

[14] R. I. Bahar, H. Cho, G. D. Hachtel, E. Macii, and F. Somenzi, "Timing analysis of combinational circuits using adds," in *EDAC'94: European Design and Test Conference*, pp. 625–629, 1994.

[15] C.-Y. Sung and J. U. Lee, "Graphene: The ultimate switch," *IEEE Spectrum*, 2012.

[16] K. S. Novoselov *et al.*, "Electric field effect in atomically thin carbon films," *Science*, vol. 306, no. 5696, pp. 666–669, 2004.