

A Heuristic Logical Effort Approach for Gate Sizing for CNFET-Based Circuits

Da Cheng, Fangzhou Wang, Feng Gao, and Sandeep K. Gupta

Computer Engineering Technical Report Number CENG-2014-03

Ming Hsieh Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, California 90089-2562

March 2014

A Heuristic Logical Effort Approach for Gate Sizing for CNFET-Based Circuits

Da Cheng, Fangzhou Wang, Feng Gao, and Sandeep K. Gupta
 Ming Hsieh Department of Electrical Engineering,
 University of Southern California, Los Angeles, CA, USA
 {dacheng, fangzhou, fenggao, sandeep}@usc.edu

ABSTRACT

Carbon nanotube (CNT) field-effect transistors (CNFETs), as one of the promising candidate emerging technologies, have distinctive device-level characteristics compared to conventional CMOS technology. Logical effort approach, which is an efficient approach for fast delay estimation in CMOS technology, however, is not universally applicable for CNFET-based circuits. In this work, we first identify scenarios where logical effort approach is not applicable. Then we propose a heuristic for delay estimation and gate sizing for such scenarios, as an important integral of the circuit design methodology for CNFET-based circuits. We have conducted two case studies on ripple-carry-adder and address decoder. Results show that logical effort approach can be used when CNFET sizes are large. For small CNFETs, logical effort approach result in up to 12.5% higher delays, where the proposed approach can provide exactly the designs with minimum delays.

I. INTRODUCTION

CNFETs have become one of the promising emerging technologies for the next generation of highly energy-efficient electronics [1][6][9]. Since the first Carbon nanotube (CNT) field-effect transistor (CNFET) was reported in 1998, great progress has been made in all the areas of CNFETs, including materials, devices, and circuits [16]. The first CNFET-based computer, Stanford Nanotube Computer, has been demonstrated in 2013 [11]. This 1-bit computer has 178 p-type CNFETs, and is fabricated using a 1 μ m lithography process. This initial demonstration has set the stage for the development of useful and performance-driven CNFET chips.

During intermediate stages of circuit design, designers use Elmore delay model for fast estimation of path delays. Performance of the final designs are evaluated using accurate library-based simulations. In CMOS technology, both capacitance (C) and drain current (I) are linearly dependent on channel width for a given functional gate [12][13]. As a result, the RC product (which can be computed by C/I) is a constant regardless of gate width, which makes *logical effort* (g) a constant. Then logical effort approach has been used as an efficient approach to estimate path delay and gate sizing for CMOS circuits [3].

Ali et al. [18] claim that logical effort approach is still applicable for CNFET-based circuit through empirical studies. However, this is not universally true according to our experiments. In this work, we theoretically identify the scenarios where logical effort approach is applicable for delay estimation and gate sizing for CNFET-based logic circuits. Based on these observations, we develop a logical-effort-based heuristic to address such heterogeneity as an integral of the circuit design methodology.

Section II introduces the background of CNFETs. Section III shows our problem statement. Section IV shows the key idea of this research. Section V proposes the design heuristic for delay estimation and gate sizing for CNFET-based circuits. Section VI evaluates the proposed heuristic using two case studies. Section VII summarizes this paper.

II. BACKGROUND

CNFETs are fabricated using CNTs, which are hollow, cylindrical nanostructures composed of a single sheet of carbon atoms, and have exceptional electrical, physical and thermal properties [2][4][11]. A top view of a CNFET is reproduced in Figure 1 [17]. In order to obtain both p-type and n-type FETs using CNTs, the polarities of the FETs are controlled using metals with different work functions. As electrons and holes have almost the same mobility, number of CNTs (N_{tub}) in pull-up and pull-down networks will be the same. In the rest of this section, we review CNFETs in following aspects: capacitance and current model, and area model.

A. Capacitance and current model

A CNFET capacitance model is developed in [5], where gate capacitance of a CNFET (C_g) mainly consists of three components: gate-to-channel capacitance (C_{gc}), outer fringe gate capacitance (C_{of}), and the gate-to-source/drain coupling

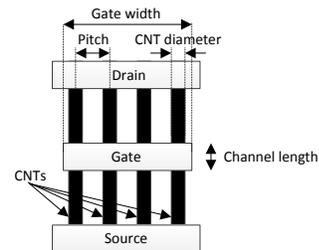


Figure 1. Structure of CNFET with 4 CNTs (adapted from [17]).

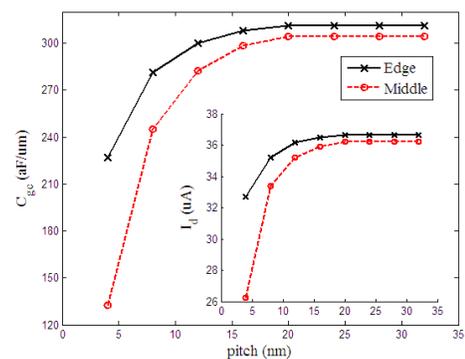


Figure 2. Screening effect on capacitance (C_{gc}) and current (I_d).

capacitance (C_{gtg}). Both C_{gc} and C_{of} are strongly affected by screening effect (or shielding) of neighboring channels, particularly, for closely spaced channels. Specifically, in order to calculate coupling capacitance between a CNT and gate metal electrode, the total effect of other CNTs can be lumped and approximated as the two nearest CNTs. This effect has a similar impact on current as well. The relationship between capacitance (current) and CNT pitch is reproduced in Figure 2 [5]. Parasitic capacitance is assumed to be equal to gate capacitance [10][14].

B. Area model

The minimum gate width (W_{gate}) of a typical transistor is 3λ , due to lithography limitations [8]. For 32nm technology, the minimum W_{gate} equals to 48nm. If we consider 4nm as the minimum pitch value for CNTs [5], there can be up to 12 CNTs in a CNFET gate for the minimum gate width. We use N_{ath} for the maximum allowable N_{tub} in this scenario. If a CNFET gate has more than N_{ath} CNTs, W_{gate} must be increased to cover all CNTs. Note that W_{gate} can only be increased by multiples of λ due to lithography limitations, i.e., W_{gate} can only have values such as 3λ , 4λ , 5λ , and so on. The width of a CNFET gate is computed using Equation (1), where P_{tub} is CNT pitch value, and N_{tub} is number of CNTs for a CNFET.

$$W_{gate} = MAX\left(3\lambda, \left\lceil \frac{N_{tub} \cdot P_{tub}}{\lambda} \right\rceil \cdot \lambda\right) \quad (1)$$

III. PROBLEM STATEMENT

Significant progress has been achieved on demonstration of functionality for CNFET-based circuits, e.g., a ring oscillator [15] and a 1-bit carbon nanotube computer [11]. Now it is helpful to look into performance-oriented issues in order to build fast circuits.

Our problem is to develop an approach which determines N_{tub} for each transistor so that the critical path delay of a circuit is minimized, under given design specifications.

Transistors in a logic circuit can be divided into three categories: (i) limited transistors, or Tr_{lmt} , whose sizes are limited to particular constraints required by design specifications, e.g., maximum input capacitance required by preceding logic, sizes of transistors on the critical path determined by logical effort, (ii) don't-care transistors, or Tr_{dc} , which are *not* directly driven by any transistors on the critical path, and (iii) down-scaled transistors, or Tr_{ds} , which are directly driven by transistors on the critical path (as they increase the internal or load capacitance, they have impact on critical path delay).

To minimize critical path delay, (i) Tr_{lmt} , which usually compose pull-up and pull-down networks for critical path, should be assigned relatively large size as they provide the driving current, and (ii) Tr_{ds} , which don't compose pull-up or pull-down networks but add to internal or load capacitance should be assigned the minimum size. In this work, we assign the minimum size, i.e., $N_{tub} = 1$, to both Tr_{ds} and Tr_{dc} . And our approach will determine N_{tub} for each transistor in Tr_{lmt} .

IV. KEY IDEAS

Library-based simulations provide accurate and reliable estimates for circuit performance based on a given library. However, this approach is not an efficient approach to identify the optimal design as (i) number of transistors in a circuit is usually large, which requires substantial simulation time, and (ii) design space is usually large, especially for new technologies, where more variations have to be considered. Elmore delay model enables fast estimations of critical path delays and gate sizing without the complexity of constructing libraries and simulating circuits. The optimal circuit design can be obtained analytically, i.e., by enumerating different circuit configurations.

In CMOS technology, input and output capacitance (C) are both linearly dependent on channel width for a given functional gate [12][13]. At the same time, drain current (I) is linearly dependent on the channel width as well. As a result, the RC product (which can be computed by C/I) becomes a constant regardless of the gate width. As *logical effort* (g) is a constant, logical effort approach has been used as an efficient approach to estimate path delay and size gates without enumeration of design configurations [3].

For CNFET technology, however, the gate capacitance and drain current of a transistor are no more linearly dependent on N_{tub} . (Note that N_{tub} in the context of CNFET corresponds to the channel width in the context of CMOS.) Based on Figure 2, we are able to compute the gate capacitance and drain current for a transistor with a certain N_{tub} . The RC product, as well as gate capacitance (approximated using C_{gc}) and drain current, are shown in Figure 3. It can be seen that current and capacitance change with N_{tub} in different manners. Hence, the RC product is *not* a constant, and the idea of logical effort doesn't apply to CNFETs in a strict sense. However, as N_{tub} in a transistor increases beyond a certain point, which is marked as "Break point" at $N_{tub} = 10$ in Figure 3, the RC product doesn't change a lot. In other words, logical effort approach might still be applicable for CNFETs which have more than 10 CNTs, i.e., $N_{tub} \geq 10$. In this work, we propose a hybrid approach for delay estimations and gate sizing by *enumerating* N_{tub} for transistors whose sizes are below a certain threshold, and applying logical effort approach to *fast identify* the optimal N_{tub} for transistors above this threshold.

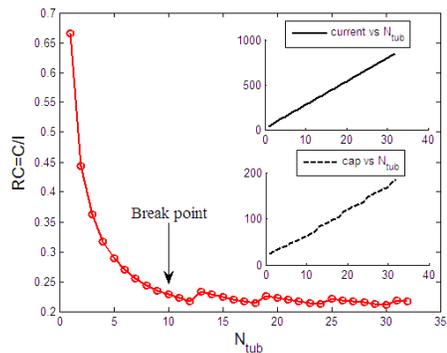


Figure 3. Ratio of capacitance and current for an inverter with N_{tub} .

V. THE PROPOSED HEURISTIC

We use $N_{le,th}$ to represent the threshold below which a transistor cannot be sized using logical effort approach. In our approach, we first identify small transistors, i.e., transistors with $N_{tub} < N_{le,th}$. For logic circuits *without branches*, gates in the beginning always have larger sizes than those gates in the end in each path. Hence transistors with smallest N_{tub} will always appear in the first few stages. For logic circuits *with branches*, transistors in following few stages after each *branch* might have $N_{tub} < N_{le,th}$ especially when fan-out is high.

Then we apply following heuristic to a circuit given input transistor sizes, which are subject to design specifications, and *branch* information. Figure 4 illustrates the proposed heuristic for two scenarios, i.e., without and with branches.

The proposed heuristic. We start by enumerating N_{tub} for transistors which might be *logical-effort-infeasible*, including those in the first few stages in the circuit and in each branch. During enumeration, transistors sizes are set to be greater than or equal to that of their predecessors. If transistors have $N_{tub} \geq N_{le,th}$ in stage i , we apply logical-effort approach to the following transistors from stage i to next branch point, which are logical-effort-feasible.

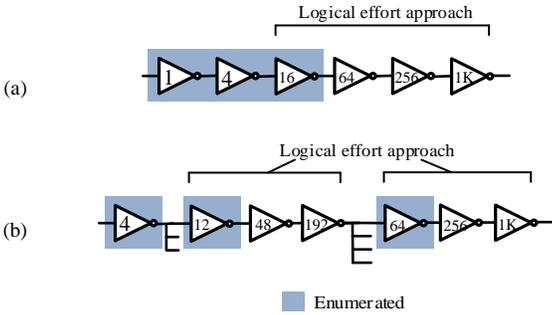


Figure 4. The proposed heuristic. (a) Without branches. (b) With branches.

VI. EXPERIMENTAL STUDIES

In this section, we evaluate the proposed heuristic by conducting two case studies on ripple-carry-adder and address decoder.

A. Ripple-Carry-Adder (RCA)

Figure 5 shows the transistor-level schematic of a 1-bit full adder built with compound gates, where \overline{Carry}_i is used to compute Sum_i . Elmore delay model is shown in Figure 5(b), where resistance variables (i.e., R_a, R_b , and R_c) and capacitance variables (i.e., C_a, C_b , and C_c) are obtained using Equation (2)-(5). Note that C_D^i , C_G^i , and R_i represent drain capacitance, gate capacitances, and resistance for CNFET i .

In a typical RCA design built by cascading such 1-bit full adders, $carry_i$ and $carry_{i+1}$ are on the critical path. Usually the size of input transistors of the critical path is subject to particular constraints due to design specifications. Then following transistors on the critical path are sized accordingly. Hence, in a RCA design, Tr_{lmt} include transistors which compose pull-up and pull-down networks for \overline{Carry}_i , or $carry_i$, e.g., CNFET 1-6, 13, and 14, which are highlighted in

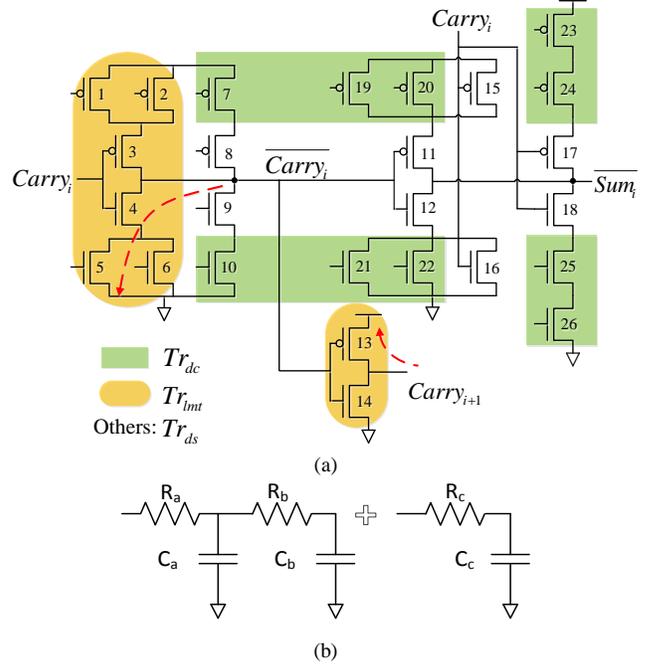


Figure 5. (a) Schematic of 1-bit full adder built with compound gates. (b) Elmore delay model of 1-bit full adder.

$$C_a = C_D^4 + C_D^5 + C_D^6 \quad (2)$$

$$C_b = C_D^3 + C_D^4 + C_D^8 + C_D^9 + C_G^{11} + C_G^{12} + C_G^{13} + C_G^{14} \quad (3)$$

$$C_c = C_G^3 + C_G^4 + C_G^{15} + C_G^{16} + C_G^{17} + C_G^{18} \quad (4)$$

$$R_a = R_5, R_b = R_4, R_c = R_{13} \quad (5)$$

ellipses in Figure 5(a). And Tr_{ds} include transistors which add to internal or load capacitance to \overline{Carry}_i , or $carry_i$, e.g., CNFET 8, 9, 11, 12, 17, and 18. The remaining transistors, which are highlighted in rectangles, belong to Tr_{dc} .

For simplicity, we assign $N_{tub} = 1$ to both Tr_{ds} and Tr_{dc} . Based on Elmore delay model, we enumerate all possible combinations of N_{tub} for transistors in Tr_{lmt} and obtained the optimal N_{tub} configurations for different N_{tub} values of input transistor (i.e., CNFET 3 and 4) for an 8-bit RCA, which is shown in Table 1, where CNFET 4 is used as a measure of input driving capability. The obtained design from above exhaustive approach is taken as the ground truth for the design with minimum delay. We also obtain designs using the proposed heuristic and logical effort approach.

Then we compute delay values using Elmore delay model for the designs which are obtained using the three approaches, i.e., exhaustive approach, the proposed heuristic, and logical effort approach. *Results show that the proposed heuristic provides the same optimal design as the exhaustive approach, which is different from that obtained by the logical effort approach.* These different designs are shown in Table 1. The delay difference between the two design decreases as N_{tub} increases for Tr_{lmt} . Also there is not much difference between the obtained optimal delays if CNFET 4 has more than 16 CNTs. It is also important to note that all other Tr_{lmt} have more

than 10 CNTs. This supports our prediction in the previous section, i.e., logical effort approach may still be applicable beyond “Break point” at $N_{tub}=10$ in Figure 3.

Table 1. Elmore delays for RCA with different Tr_{imt} using the exhaustive approach, the proposed heuristic, and logical effort approach.

$(N_4, N_5, N_{13}, N_{dc}, N_{ds})$	Exhaustive/ proposed heuristic	Logical effort approach
(1, 1, 1, 1, 1)	13.60	15.30
(2, 2, 5, 1, 1)	8.76	8.51
(4, 4, 7, 1, 1)	5.97	5.69
(8, 8, 11, 1, 1)	4.06	3.94
(16, 16, 12, 1, 1)	3.16	3.09
(32, 32, 24, 1, 1)	2.67	2.67

B. Address decoder

Address decoder (AD) is one of the most commonly-used logic circuit and an important component of memory design. A typical address decoder in SRAM designs is designed with identical paths, where each path drives a word line. As a result, all paths are critical paths. Hence, address decoder is a special case of logic circuits in terms of adding redundant CNTs as it comprises only Tr_{imt} . In this work, we use 4-to-16 decoder as a case study. We assume each word line drives 16 SRAM cells. Gate capacitance of an access transistor is C_g . As each SRAM cell has two access transistors, i.e., M2 and M5 in Figure 6, load capacitance for each word line is $32C_g$.

Table 2 shows delays for address decoders with different input capacitance constraints. Results show that the proposed heuristic provides more accurate delay estimations than logical effort approach. Especially the propose heuristic provides the same designs as the exhaustive approach when $N_{tub} \geq 4$ for the primary input CNFETs.

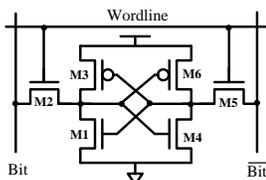


Figure 6. 6T CNFET SRAM cell.

Table 2. Elmore delays for AD with different Tr_{imt} using exhaustive approach, the proposed heuristic, and logical effort approach.

Gate sizes	Exhaustive	Proposed heuristic	Logical effort approach
(1, 5, 12, 19, 29, 44)	4.71	4.96	5.12
(2, 8, 17, 33)	3.19	3.45	3.49
(4, 12, 22, 38)	2.65	2.65	2.96
(8, 14, 24, 40)	2.29	2.29	2.58
(16, 24, 34, 47)	2.03	2.03	2.28
(32, 46)	1.01	1.01	1.15

VII. CONCLUSIONS

In this work we identify scenarios where traditional logical effort approach is not applicable for delay estimation and gate sizing for CNFET-based circuits based on their distinctive device-level characteristics. Then we develop a heuristic to take into account such heterogeneity, as an important integral of the design methodology for CNFET-based circuits. We conduct two case studies on ripple-carry-adder and address decoder to

evaluate logical effort approach and the proposed heuristic with respect to exhaustive approach. Results show that the proposed heuristic provides more accurate delay estimations than logical effort approach. Specifically, logical effort approach can be used when CNFET sizes are large. For small CNFETs, logical effort approach result in up to 12.5% higher delays, where the proposed approach can provide exactly the designs with the minimum delay.

VIII. REFERENCES

- [1] A. D. Franklin et al., “Sub-10nm Carbon nanotube transistor,” *Nano Letter*, 2012.
- [2] A. Javey, et al. “Ballistic carbon nanotube transistors,” *Nature*, 2003.
- [3] I. Sutherland et. al., “Logical Effort: Designing Fast CMOS Circuits,” Morgan Kaufmann, 1999.
- [4] J. Appenzeller, “Carbon nanotubes for high-performance electronics-progress and prospect,” in *Proceedings of IEEE*, 2008.
- [5] J. Deng, “Device Modeling and Circuit Performance Evaluation for Nanoscale Devices: Silicon Technology Beyond 45nm Node and Carbon Nanotube Field Effect Transistors,” PHD Thesis, 2007.
- [6] J. Deng et al., “Carbon Nanotube Transistor Circuits Circuit level performance bench marking and design options for living with imperfections,” in *Proceedings of ISSCC*, 2007.
- [7] J. Zhang et al., “Carbon Nanotube Circuits in the presence of CNT Density Variations,” in *Proceedings of DAC*, 2009.
- [8] J. Zhang et al., “Carbon Nanotube Correlation: Promising Opportunity for CNFET Circuit Yield Enhancement,” in *Proceedings of DAC*, 2010.
- [9] L. Wei et al., “A Non-Iterative Compact Model for Carbon Nanotube FETs Incorporating Source Exhaustion Effects,” in *Electron Devices Meeting (IEDM)*, pp. 1-4, 2009.
- [10] M. Ali, R. Ash, and M. C. Jeske, “Logical effort of CNFET-based circuits in the presence of Metallic Tubes,” in *Proceedings of IEEE-NANO*, 2012.
- [11] M. M. Shulaker et al., “Carbon Nanotube Computer,” *Nature*, 501(7468), 526-530, 2013.
- [12] N. Weste and D. Harris, “CMOS VLSI Design: A Circuits and Systems Perspective,” Addison-Wesley, Fouth edition, 2010.
- [13] S. M. Kang and Y. Leblebici, “CMOS digital integrated circuits,” McGraw-Hill, New York, 2003.
- [14] Y. B. Kim and F. Lombardi, “A novel design methodology to optimize the speed and power of the CNFET circuits,” in *Proceedings of MWSCAS*, 2009.
- [15] Z. Chen et al., “An Integrated Circuit Assembled on A Single Carbon Nanotube,” *Science*, 311(5768), 1735-1735, 2006.
- [16] W. Wang et al., “Design Method for 6T CNFET Misalignment Immune SRAM Circuit,” in *Proceedings of MWSCAS*, pp. 1-4, 2011.
- [17] W. Wang et al., “High SNM 6T CNFET SRAM Cell Design Considering Nanotube Diameter and Transistor Ratio,” in *Proceedings of EIT*, pp. 1-4, 2011.
- [18] M. Ali, R. Ashraf, and M. Chrzanowska-Jeske, “Logical Effort of CNFET-based Circuits in the Presence of Metallic Tubes,” in *Proceedings of IEE-NANO*, pp. 1-6, 2012.