Automated Design of Misaligned-Carbon-Nanotube-Immune Circuits

Nishant Patil, Jie Deng H. -S. P. Wong and Subhasish Mitra
Departments of Electrical Engineering and Computer Science, Stanford University
{nppatil, jdeng, hspwong, subh}@stanford.edu

Abstract

Carbon Nanotube Field-Effect Transistors (CNFETs) are promising candidates as extensions to Silicon CMOS due to excellent CV/I device performance. An ideal CNFET inverter fabricated using a perfect CNFET technology can have 5.1 times faster FO4 delay and 2.6 times lower energy per cycle compared to a 32nm Silicon CMOS inverter. Two fundamental challenges prevent us from creating CNFET-based logic designs with the advantages quoted above: 1. Misaligned Carbon Nanotubes (CNTs), and 2. Metallic CNTs. Misaligned CNTs can result in incorrect logic function implementations. This paper presents a technique for designing CNFET-based arbitrary logic functions that are guaranteed to be correct even in the presence of a large number of misaligned CNTs.

Categories and Subject Descriptors

B.7 [Hardware] : Integrated Circuits

General Terms

Algorithms, Design, Reliability

Keywords

Carbon Nanotube Transistor, Misaligned Carbon Nanotubes Immune, CNT, CNFET, Fault tolerance, Circuits.

1. Introduction

Carbon Nanotube Field Effect Transistors (CNFETs) are promising extensions to Silicon CMOS [Wong 03]. Figure 1.1 shows the schematic of a CNFET inverter in the 32nm technology node. Parallel semiconducting Carbon Nanotubes (CNTs) are grown on or transferred to a substrate. The inverter consists of contacts, p+ doped source and drain regions of semiconducting CNTs for the pull-up PFET network, n+ doped source and drain regions of semiconducting CNTs for the pull-down NFET network, two gates for the pull-up and pull-down networks, undoped or intrinsic regions of CNTs under the gates, and local interconnects connecting the gates. The conductivity of the semiconducting regions of the CNTs is controlled by the corresponding gate. The distance between the gates and the contacts is limited by the lithographic feature size (lithographic pitch in Fig. 1.1) which is 64nm for the 32nm technology node. The inter-CNT distance (sub-lithographic pitch in Fig. 1.1) is not limited by lithography because the CNTs are grown through self-assembly. With this structure, a large portion of the existing design and manufacturing infrastructure for FET-based large scale electronic systems can be utilized.

Semiconducting CNTs are grown on or transferred to a substrate. The regions of logic cells are defined using lithography, and the CNTs outside these regions are etched away. Gate and contact regions are then defined using lithography. Next, CNT regions that create PFET transistors are doped p-type (NFET regions are lithographically masked during this step), and CNT regions that create NFET transistors are doped n-type (the PFET regions are lithographically masked during this step). The CNT regions under the gates remain undoped because they are masked during the doping steps in this self-aligned process [Chen 04]. Finally, the interconnects are defined using lithography.

An ideal CNFET inverter in 32nm technology is 5.1 times faster FO4 delay and consumes 2.6 times lower energy per cycle compared to a 32nm Silicon CMOS inverter [Deng 07]. However, there are two major barriers that must be overcome before large scale integration of CNFETs becomes feasible:

1. Misaligned CNTs: A major challenge in CNT fabrication is accurate positioning and placement of CNTs to make CNFETs [Han 05, Kocabas 06]. Figure 1.2a shows a CNFET NAND cell layout overlaid on an SEM image of CNTs. The misaligned CNT in Fig. 1.2a causes a Vdd to output short in this NAND cell because the portion of this CNT between Vdd and output is entirely p-doped. A misaligned CNT may also cause an incorrect logic function implementation as illustrated in Fig. 1.2b.

2. Metallic CNTs: Metallic CNTs cannot be used to make CNFETs because their conductivity cannot be controlled by the gate. Current CNT synthesis techniques yield between 10% to 30% metallic CNTs [Li 04]. Metallic CNTs can be removed by selective chemical etching [Zhang 06]. However, such etching can introduce significant variations in CNFET circuits [Deng 07]

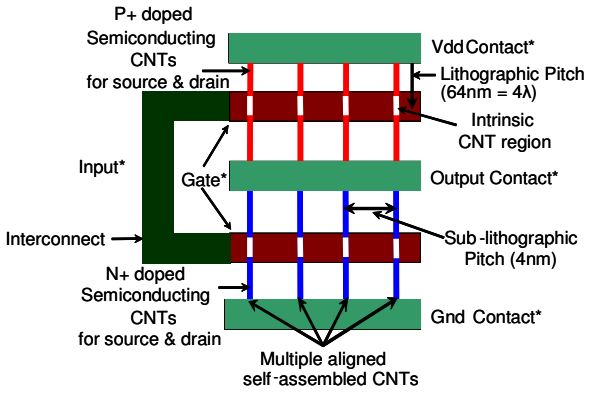


Figure 1.1. A CNFET inverter.

Both misaligned CNTs and metallic CNTs are major challenges for CNFET circuits. This paper focuses on the first challenge of misaligned CNTs and provides a new design solution to address the problem. Because of very high misalignment rates, discarding defective chips with misaligned CNTs or reconfiguring around defective cells via testing may be very expensive. Traditional fault tolerance techniques like Triple Modular Redundancy (TMR) are also very expensive, and may not be adequate for such high defect rates.

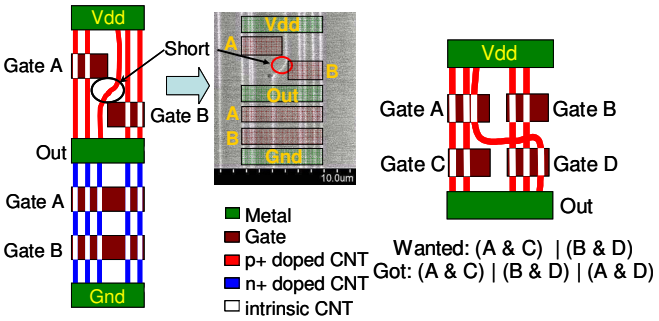
The major contributions of this paper are:

- A technique for designing CNFET logic circuits that are guaranteed to implement correct logic functions even in the presence of a large number of misaligned CNTs. We refer to these circuits as *misaligned-CNT-immune circuits*.

DAC 2007, June 4–8, 2007, San Diego, California, USA

- An automated algorithm to determine whether or not a given CNFET logic circuit design is misaligned-CNT-immune.
- Simulation results to compare energy, delay and area of misaligned-CNT-immune circuit designs compared to designs that are not guaranteed to be misaligned-CNT-immune.

Section 2 describes the algorithm for determining whether a CNFET-based logic circuit is misaligned-CNT-immune or not. In Sec. 3, we present an technique for designing misaligned-CNT-immune circuits. Section 4 presents simulation results. Related work is discussed in Sec. 5. We conclude in Sec. 6.



(a) (b)

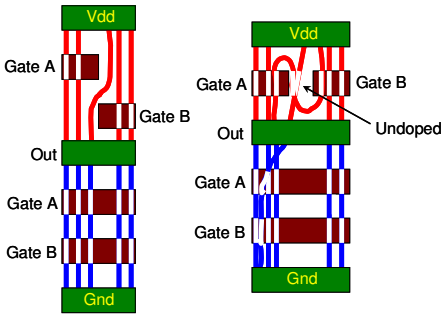
Figure 1.2. (a) Short inside NAND gate caused by misaligned CNT. (b) Incorrect logic function due to misaligned CNT.

2. Determining Misaligned-CNT Vulnerability

We use a graph abstraction of the layout of a CNFET-based circuit to determine whether one or more misaligned CNTs can result in incorrect logic function implementation. Figure 2.1 shows two possible structures for a NAND circuit (the wires connecting the PFET and the NFET inputs are not shown for clarity). These two structures will be used to illustrate our technique. The two gates in the pull-up network of Fig. 2.1a (corresponding to inputs A and B) are not at the same horizontal level to create a compact layout in the horizontal direction. In Fig. 2.1b, the two gates in the pull-up network are at the same horizontal level separated by an undoped region. The CNTs in this undoped region of Fig. 2.1b can also be etched away. As discussed earlier in Sec. 1, the regions outside the cell boundaries are either devoid of CNTs (removed by etching) or have undoped non-conducting CNTs. In the ideal case, all CNTs should grow on the substrate in one direction from one contact to another passing under the gates. However, in reality, not all CNTs are perfectly aligned (as discussed in Sec. 1). Consider the misaligned CNT in Fig. 2.1a, which has all doped regions because it is misaligned. This CNT will cause a short between the Vdd and output nodes. On the other hand, any misaligned CNT in Fig 2.1b will not cause a malfunction. This is because any CNT in the pullup network of Fig. 2.1b either passes under the gates corresponding to inputs A or B, or passes through the undoped region between the two gates (in which case it will not conduct unless either A or B is turned on). Alternatively, the undoped CNTs between the gate regions can be just etched away in the misaligned-CNT-immune design. Assuming that CNTs are sufficiently dense, the chance that no CNT passes through a given gate (e.g., gate A or gate B) is very remote.

The first step in our automated analysis technique is to divide the cell into pull-up and pull-down regions. We analyze each of these regions separately. Each region is decomposed into a finely divided *square grid*. The dimension of each side of a square in this grid is equal to the smallest lithography feature size. Each square in this grid has a *label*: Contact (C), Doped (D), Undoped or Etched (UD) or Gate (G). For a gate, we also include the input variable associated with that gate (e.g, G_A in Fig. 2.2). For a contact, we label a Vdd contact as C_V, ground contact as C_G, output contact as C_O, and any other intermediate contact as C.

We create a graph where each square in the grid is a node in the graph. The label associated with a node is the same as that associated with the corresponding square. The details are given in Table 2.1. A CNT in a given square can grow into any of its adjacent squares in the grid. Hence, there is an edge between two nodes in the graph if and only if the corresponding squares in the grid are adjacent, i.e. the squares corresponding to those two nodes have a boundary or a vertex in common. Two nodes with an edge between them are referred to as *neighboring nodes*.



(a) (b)

Figure 2.1. (a) Misaligned-CNT-vulnerable NAND cell. (b) Misaligned-CNT-immune NAND cell.

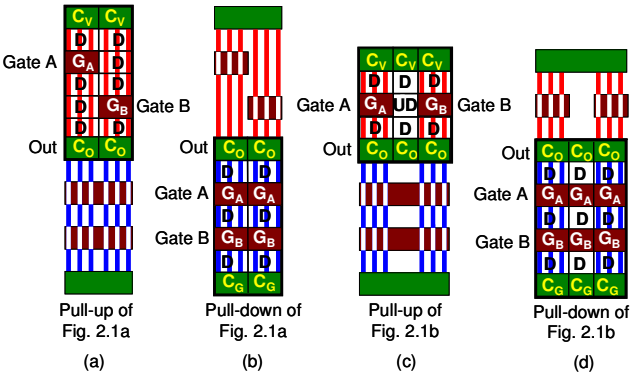


Figure 2.2. Grid Decomposition of NAND cell layouts in Fig. 2.1.

To reduce the number of nodes and edges in the graph, we combine neighboring nodes with the same label into a single node. The set of neighbors of this new combined node comprises the union of the neighbors of its constituent nodes. Figure 2.3 shows the reduced versions of the graphs corresponding to the layouts in Figure 2.2. It can be proved that a reduced graph is equivalent to the original layout. Hence, we can work on the reduced graph to determine whether a given layout is immune to misaligned CNTs.

Each node in the graph has an associated Boolean function as defined in Table 2.1. For example, the Gate node G_A has the associated function A since the CNT through that gate conducts when A = 1 (For a PFET the CNT conducts when A′ = 1). For a node with label D, the associated function is 1 since the doped region of the CNT always conducts. For an undoped region the corresponding Boolean function is 0 since an undoped region of the CNT does not conduct.

To determine whether the pull-up (pull-down) network implements the correct function in the presence of misaligned CNTs, we need to traverse all possible paths between the Vdd contact (Gnd contact) node and the output contact in the corresponding graph. For the pull-down network, we compare against the complement of the function implemented by the standard logic cell. In the case of the pull-up network, we apply De-Morgan's law and then complement the variables. This step simplifies the analysis without loss of correctness.

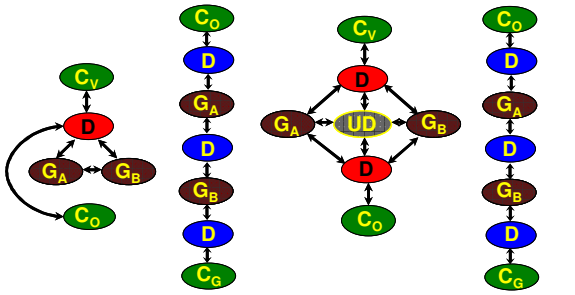


Figure 2.3. Reduced graphs for NAND cell layouts in Fig. 2.2.

Table 2.1. Labels associated with nodes in the graph.

Node type	Node Label	Boolean Function
Gate with input variable A	G_A	A
Gate with input variable A′	$G_{A'}$	A′
Doped Region	D	1
Undoped Region	UD	0
Vdd contact	C_V	1
Gnd contact	C_G	1
Output contact	C_O	1
Any intermediate contact	C	1

Let us first consider paths with no loops. The *Boolean function associated with each path* is obtained by AND-ing the Boolean functions associated with the nodes along the path. This Boolean function of a path represents the switch level function implemented by a CNT traversing that path. For example, in Fig. 2.3c, the path C_v-D-G_A-D-C_O has the Boolean function A and the path C_v-D-UD-D-C_O has the Boolean function 0. In Fig. 2.3a, the path C_v-D-C_O has the Boolean function 1. For paths with loops, we traverse the loops only once, since the Boolean expression associated with the path will not change with multiple traversals of the same loops.

The OR of the Boolean functions associated with all paths must be identical to the intended function (function without misalignment) for the circuit to be immune to misaligned CNTs. This is because the paths represent all possible CNT misalignment scenarios. In our implementation, we used standard equivalence checking techniques for this purpose [Beer 96]. The reader can verify that the OR of the Boolean functions of all paths in the graphs of Fig. 2.3b and 2.3d are identical to the intended function implemented by the pull-down network of a NAND cell. These graphs have special properties and we refer to them as *straight line graphs*. A misaligned CNT in the pull-up network of the misaligned-CNT-vulnerable cell may create a short between Vdd and output because the Boolean function of the path C_v-D-C_O in Fig. 2.3a is 1 (which includes all possible minterms). However, the pull-up network of the misaligned-CNT-immune NAND is immune to misaligned CNTs because the Boolean function of any path cannot include minterms that are not present in the original function.

3. Sufficient Condition for Misaligned-CNT-Immune Design

We implement misaligned-CNT-immune layouts of the pull-up and pull-down networks. We first discuss the case where the pull-up and pull-down networks are expressed in Sum of Products (SOP) or Product of Sums (POS) form.

For a network specified in POS form, the misaligned-CNT-immune layout implementation is shown in Fig. 3.1a. This structure is immune to misaligned CNTs since each sum term is immune to misaligned CNTs (by the same argument as the SOP case). Any CNT (misaligned or straight) that spans multiple sum terms must pass through a contact and this does not change the function implemented by the cell.

For a network specified in SOP form, the misaligned-CNT-immune layout implementation is shown in Fig. 3.1b. Any path in the corresponding graph between the Vdd/Gnd contact and the output contact must pass through all the gates corresponding to a product term or through an undoped region. Hence, the Boolean function corresponding to that path cannot include minterms that are not present in the SOP representation of the network. Hence the overall layout in Fig. 3.1b is immune to misaligned CNTs.

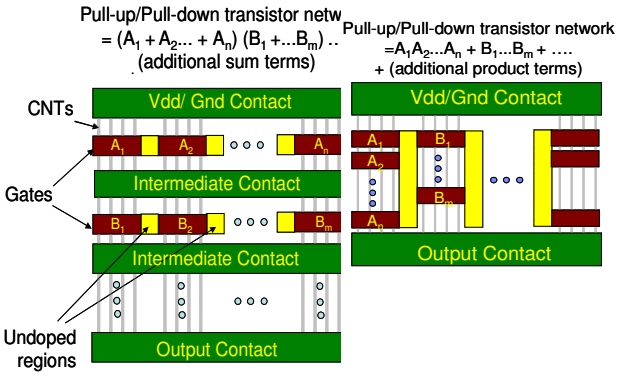


Figure 3.1 Misaligned-CNT-immune layout for (a) Product of Sums (b) Sum of Products.

An arbitrary representation of the network may not be in either SOP or POS forms. For example, consider the network represented by the expression A + (B + C) (D + E). We first implement the term (B + C) (D + E) similar to Fig 3.2. Next, we implement A in parallel with this structure and separated by an undoped region.

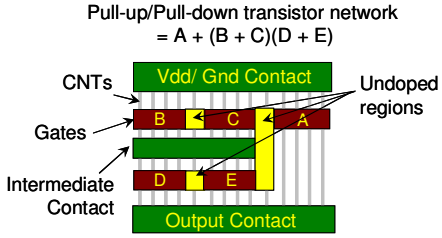


Figure 3.2. Misaligned-CNT-immune layout for network represented by the function A + (B + C) (D + E).

4. Simulation Results

As discussed in Sec. 1, a misaligned-CNT-immune layout may impose energy, delay or area penalties because of additional undoped or etched CNT regions between gates. In this section, we present simulation results to quantify these penalties for standard logic library cells. For circuit simulations, we use a CNFET HSPICE model that is implemented with practical device non-idealities and calibrated using experimental CNT data [Deng 06]. For each standard cell, we simulated a five stage fanout of 4 (FO4) chain of that cell, and measured the delay through and the switching energy consumed by the 3rd stage. We used 0.9V power supply for CNFET circuits at the 32nm technology node. Table 4.1 shows the penalties for misaligned-CNT-immune cells compared to misaligned CNT vulnerable cells. Area penalties were calculated directly from the cell layouts. Negative penalties imply improvements.

The previous simulations did not take into account situations where CNTs cross each other due to misalignment. If the misalignment occurs under the gate region, the upper CNT partially shields the gate from the lower CNT. Hence, the conductivity of the lower CNT is controlled by the gate as well as the upper CNT and the on-current through the lower CNT decreases. In the worst case where two CNTs overlap over the entire gate region, the on-current of the bottom CNT degrades by 11%. We are currently working with chemists on collecting statistics on occurrences of such misalignment scenarios to quantify the overall delay penalties

Table 4.1. Penalties for Misaligned-CNT-immune design.

	Penalties		
Cell Type	Area*	Energy*	Delay [max {rise, fall}]*
nand2	-1%	3%	-7%
nand3	11%	15%	10%
nand4	21%	18%	13%
nor2	-1%	5%	1%
nor3	11%	16%	10%
nor4	21%	12%	-5%
aoi21 [f = ~(ab+c)]	-2%	1%	1%
oai21 [f = ~((a+b)c)]	-2%	1%	1%
Full Adder	12%	10%	7%

* Negative penalty implies improvement

5. Related Work

Several earlier publications addressed the problem of high defect rates in nanoscale circuits fabricated using self-assembly. It was observed in [Butts 02] that the defect rates of nanotechnology will be large and conventional fault tolerance techniques may not be sufficient. Several publications discuss the use of test, reconfiguration and fault tolerance techniques for such purposes [Dehon 05, Snider 04, Strukov 06]. The design technique in this paper addresses the misaligned CNT problem at a much lower level of abstraction, at the layout itself. It is demonstrated that the misaligned-CNT problem can be successfully solved at this level of abstraction, reducing the burden on higher-level techniques.

6. Conclusions

Misaligned CNTs impose a major barrier to practical implementations of CNFET-based logic circuits. This paper shows that it is possible to overcome this barrier by designing CNFET-based logic circuits that are inherently immune to misaligned CNTs. The energy, delay and area costs associated with such a design technique are significantly lower than traditional fault-tolerance techniques. Moreover, this design technique can be easily automated as also demonstrated in this paper. We are currently working with chemists to create hardware prototypes of misaligned-CNT-immune designs. Major areas of future research include: 1. Automated techniques for synthesizing misaligned-CNT-immune circuit layouts together with global energy, delay and area optimization. This is in contrast to the sufficient condition we use in this paper for generating misaligned-CNT-immune circuits; and 2. Techniques for designing CNFET-based logic circuits immune to metallic CNTs.

7. Acknowledgments

This work is partially supported by the MARCO Focus Center Research Program (GSRC).

8. References

[Beer 96] Beer I., *et al.*, "RuleBase: an Industry-Oriented Formal Verification Tool", *Proc. Design Automation Conf.*, 1996.

[Butts 02] Butts M., A. DeHon and S.C. Goldstein, "Molecular electronics: devices, systems and tools for gigagate, gigabit chips," *Proc. Intl. Conf. CAD*, pp. 433-440, 2002.

[Chen 04] Chen, J., C. Klinke, A. Afzali and P. Avouris,. "Air-stable chemical doping of carbon nanotube transistors [CNFETs] *Device Research Conference*, 2004. 62nd DRC. Conference Digest, Vol., Iss., 21-23 June 2004

[DeHon 05] DeHon, A., and H. Naeimi, "Seven Strategies for Tolerating Highly Defective Fabrication," *IEEE Design and Test of Computers*, Vol. 22, No. 4, pp. 306-315, 2005.

[Deng 06] Deng J., and H.-S. P. Wong, "A Circuit-Compatible SPICE model for Enhancement Mode Carbon Nanotube Field Effect Transistors," *Proc. Intl. Conf. Simulation of Semiconductor Processes and Devices*, pp. 166-169, 2006.

[Deng 07] Deng J., *et al*., "Carbon Nanotube Transistor Circuits: Circuit-Level Performance Benchmarking and Design Options for Living with Imperfections", *Proc. Intl. Solid State Circuits Conf*, 2007.

[Han 05] Han S., X. Liu and C. Zhou, "Template-Free Directional Growth of Single-Walled Carbon Nanotubes on a- and r-Plane Sapphire", *J. Am. Chem. Soc.,* vol. 127, pp. 5294 – 5295, 2005.

[Kocabas 06] Kocabas C., M. Shim and J.A. Rogers, "Spatially Selective Guided Growth of High-Coverage Arrays and Random Networks of Single-Walled Carbon Nanotubes and Their Integration into Electronic Devices", *Journal of the American Chemical Society 128*, pp. 4540-4541, 2006.

[Li 04] Li Y., *et al*., "Preferential Growth of Semiconducting Single-Walled Carbon Nanotubes by a Plasma Enhanced CVD Method," *Nano Letters*, vol. 4, pp. 317, 2004.

[Snider 04] Snider, G., P. Kuekes and R.S. Williams, "CMOS-like logic in defective, nanoscale crossbars," *Nanotech. 15*, pp. 881-891, 2004.

[Strukov 06] Strukov D.B., and K. K. Likharev, "Defect-Tolerant Architectures for Nanoelectronic Crossbar Memories", *Journal of Nanoscience and Nanotechnology*, Aug. 2006.

[Wong 03] Wong H.-S. P., et al., "Carbon Nanotube Field Effect Transistors – Fabrication, Device Physics, and Circuit Implications", *Proc. Intl. Solid State Circuits Conf.*, pp. 370 – 371, 2003.

[Zhang 06] Zhang G., et al., "Selective Etching of Metallic Carbon Nanotubes by Gas-Phase Reaction", Science 10 November 2006 314: 974-9.