Layout-Driven Robustness Analysis for Misaligned Carbon Nanotubes in CNTFET-based Standard Cells

Matthias Beste
Karlsruhe Institute of Technology
Chair for Dependable NanoComputing
76131 Karlsruhe, Germany
Email: beste@kit.edu

Mehdi B. Tahoori
Karlsruhe Institute of Technology
Chair for Dependable NanoComputing
76131 Karlsruhe, Germany
Email: mehdi.tahoori@kit.edu

***Abstract*—Carbon Nanotube Field Effect Transistors (CNTFETs) are being considered as a promising successor to current CMOS technology. Since the alignment of CNTs cannot be fully controlled yet, the layout of CNTFET-based standard cells has to be designed robust against misalignment. As CNTFET-based designs become more prevalent, a systematic methodology for misalignment robustness evaluation becomes crucial. In this work we present a novel EDA tool "Layout-Driven Robustness Analysis" (LDRA) which enables designers to, for the first time, measure the robustness against misalignment. LDRA is validated and applied to various CNTFET-based standard cell layouts. The comparison of these cells reveals that robustness against misalignment is complex and depends on many factors. A CNT curve model is introduced and its influence on the robustness result is discussed. In conclusion, key factors for designing layouts robust against misalignment are proposed.**

I. Introduction

As scaling of CMOS technologies reaches physical limits, new device structures are necessary to replace traditional MOSFETs in integrated circuits [1]. One of the most promising devices is the *Carbon Nanotube FET* (CNTFET) [2]. While giving MOSFET-like behaviour, CNTFETs come with less power consumption and better performance. The benefits are provided by the nearly-ballistic carrier transport of the CNT, with almost no scattering and a higher channel control via the gate. In order to use *Carbon Nanotubes* (CNTs) in a robust process technology flow, two major technological challenges remain: (1) a method to filter the metallic CNTs from the semiconducting ones; and (2) the capability to precisely place CNTs on the wafer [3]. The first issue might be overcome by additional process steps, like burning the metallic CNTs [4], or by using a chirality-selective polymer wrapper [5]. The second issue is treated mainly on the design level, as the only available method on the physical level still leaves a small percentage of misaligned CNTs [6]. This results in misaligned carbon nanotubes, which affect the functionality of CNTFET-based standard cells. Therefore, the layout of these standard cells has to be designed robust against misalignment by default. As the number of proposed CNTFET-based cells and circuits increases rapidly, a method for analysing the robustness against misalignment is becoming a necessity.

In this paper, we present a new EDA tool, called *Layout-Driven Robustness Analysis* (LDRA), for measuring the robustness of CNTFET-based standard cell layouts. LDRA analyses the effect of misalignment on the probability of having shorts. It is based on the statistical generation of misaligned CNTs, which is followed by an analysis step. The new tool determines which layout variant is more robust, or whether layout updates can improve the robustness. Furthermore, it can be used to harden the design rules against robustness issues. LDRA reads every cell layout directly out of the *OpenAccess* (OA) database. As this database is also used by Cadence and Synopsys, the new tool is fully integrated in the existing design flow and can be easily combined with commercial EDA tools. The new tool can be set for either higher accuracy or lower runtime and has no dependencies on other EDA tools. Hence, LDRA is easily applicable to a given design; e.g. LDRA determines the misalignment *robustness* (R) for a CNTFET-based (robust) inverter to 99.1% in only 3.5 minutes. That is to say that, in 99.1% of the investigated cases, the inserted misaligned CNTs will not affect the correct functionality of the inverter.

The paper is organized as follows: Section II reviews two ideas for a robust CNTFET-based cell layout proposed by [3], [7] which will then be used as input for our new tool. In Section III we present the main concept, methodolodgy, implementation, and validation of LDRA. In Section IV, robustness results for different layouts are discussed. The influence of layout parameter tuning on the robustness is also demonstrated. Section V concludes the paper.

II. Misalignment immune CNTFET-based layouts

Two layout ideas for robust CNTFET-based standard cells are proposed by Bobba et al. in [3] and Mitra et al. [7], [8]. Since these layouts will be used as inputs to the LDRA tool, we introduce them here.

Figure 1(a) shows the NAND2 standard cell layout, which is functionally affected by misaligned CNTs. The rectangles for Vdd, Out and Gnd are electrically connected to the active CNT layer. We assume that the CNT's preferred orientation is vertical, as depicted. The gates A and B are electrically isolated from the active layer. The active layer in the pull-up path (above Out and masked by the gates) is p-doped, while the pull-down path is n-doped. Due to continuously doped ungated CNTs between Vdd and Out, the latter is pulled up

978-3-9810801-8-6/DATE12/ © 2012 EDAA

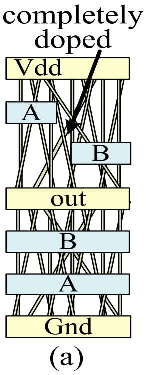

Fig. 1. NAND layouts (a) misalignment-vulnerable (b) misalignment immune by Mitra et al. [7] (c) misalignment immune, ideal stack from [3]

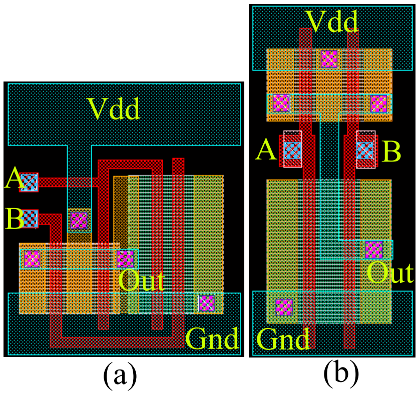

Fig. 2. (a) Variation 1 and (b) variation 2 of the robust NAND2 cell layout by Bobba et al. [3]

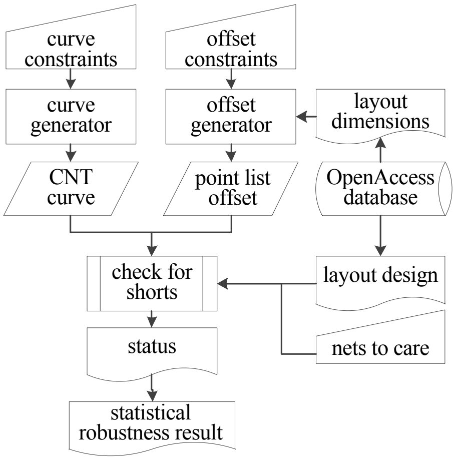

Fig. 3. Simplified internal flow of LDRA

to Vdd constantly. Solving this problem, Fig. 1(b) shows the misalignment-immune layout by Mitra et al. [7]. To achieve a nonconducting area in between the gates of the pull-up path, this area is either covered during the doping step or is etched afterwards. However, additional processing steps are necessary (i.e. treatment). Figure 1(c) shows the misalignment-immune layout idea from Bobba et al. [3], which entails no extra processing. By stacking the necessary gates in the proposed way, the authors insert additional intra-cell connections (here, for Out) instead of preparing the nonconducting area of 1(b). Two different variants of Fig. 1(c) are displayed in Fig. 2 (rotated by 90°, CNTs horizontal). In Figure 2(a) the pull-up and pull-down path share the same CNTs. The space between both pull paths can be reduced to the minimum possible dimension (2λ) which results in a compact cell layout. In Figure 2(b), two different blocks of CNTs are used for pull-up and pull-down path, which could lead to an unbalanced driver strength or unsymmetric characteristics [9]. The space between pull-up and pull-down is limited by the via dimension for the incoming signals. All proposed layout cells can be stacked easily on the Vdd and Gnd edges; on the remaining edges, additional steps are necessary to avoid inter-cell effects. For the rest of the paper, we assume that no metallic CNTs exist in the active CNT layer of the standard cell design, and that the preferred CNT orientation is perpendicular to the gates [3]. We present the robustness results for these layouts in Section IV-B.

III. Layout-Driven Robustness Analysis

The goal of Layout-Driven Robustness Analysis (LDRA) is to determine whether a given layout is robust against misaligned CNTs. One possible application is the combination of LDRA with an automatic cell layout generator to improve the cell robustness iteratively. In order to determine the robustness, LDRA performs a set of experiments and is therefore a simulation-based analysis tool. In general, two experiments are conceivable:

1) Apply a routing procedure for connecting two shapes of the layout and then compare the possible routes with the CNT curve model. Traversing all possible shape pairs would give the complete solution. The comparison of the CNT curve model with the found route gives a shape matching probability. The matching algorithm depends on the router and model implementations.
2) Generate a CNT curve by modeling parameters and overlay it with the shapes of the standard cell layout. Intersections of the CNT curve with shapes of the standard cell layout will be checked, resulting in either an electrical short or no short.

The second method allows more freedom to refine the CNT curve model without altering the other components. In addition the number of CNT curves to generate (i.e. effort) is free of choice in contrast to the first method, where all shape pairs have to be treated. We therefore focus only on the second idea in this paper. Selected parts of the LDRA are now described. First, the curve of a misaligned CNT is generated, as decribed in Section III-A. Second, the offset position of the CNT relative to the design is determined. Third, the checking algorithm is applied, described in Section III-B. The outcome of this analysis is information on whether this misaligned CNT, at this specific position, creates a short (defect). This experiment is repeated for a sufficiently large number of CNT curves and offset points to get statistical results. Figure 3 illustrates the simplified flow through the three main steps of the LDRA. To ensure that CNT curve placement is done on the entire layout, the offset points are distributed grid-like on the layout, as shown in Fig. 4 (grey dots). The grid width and the layout dimensions specify the number of offset points (#offset_points), which leads to a constant granularity for comparing different layouts. This grid is traversed in a serpentine way. Neighboring offsets will deal mostly with the same design shapes, as the common shape width is bigger than the granularity of the grid. This shifting method of the CNT over the layout benefits from the previous fact, resulting in significantly reduced runtime and refined results.

A. Modeling the misaligned CNT curve

The misaligned CNT curve consists of a list of points, which is projected to a polyline object in the OA database. All dimensions stated in this paper are in OA unit length.

The calculation algorithm for the points is now described. The user specifies the number of points (#points) and $[\mu, \sigma]$ for generating normal distributed values via the Wolfram's Mathematica Mathlink Interface [10], where μ is the mean and σ the standard deviation. Each value is treated as an $\Delta angle$. Without offset, the first point is always (0,0) and μ is generally $0°$. This avoids spiral-like CNT curves; however single self-intersections are possible due to the nature of the CNT curve. In combination with the user-specified *segment length* (segLen) and well-known trigonometric equations, the next point can be calculated. For a start angle ($\alpha_0 = 0°$) and $\Delta angle_0 = -60°$ the second point would follow as shown in Fig. 4 (black boxes).

$$x_{(n+1)} = x_n + segLen \cdot \sin(\alpha_n + \Delta angle_n) \tag{1}$$

$$y_{(n+1)} = y_n + segLen \cdot \cos(\alpha_n + \Delta angle_n) \tag{2}$$

$$\alpha_{(n+1)} = \alpha_n + \Delta angle_n \tag{3}$$

The CNT curve modeling has a progressive junction from misaligned CNT curves to non-misaligned ones due to the chosen $[\mu, \sigma]$. If there might be a harmful effect on the functionality of the standard cell (i.e. faults), the corresponding CNT can be called misaligned after the application of LDRA. Four CNT curves are illustrated in Fig. 5. The equations above represent a rather simple model of a CNT curve, however our method covers every possible CNT curve. In our future work, we will investigate statistics from actual fabricated CNT devices to refine the model. Since the LDRA algorithm can also load arbitrary point lists, there is no restriction to LDRA implied by the model simplicity.

B. Checking Algorithm

The input of the checking algorithm is the CNT curve (i.e. point list), the CNT offset, the layout (i.e. list of geometrical objects or layout shapes), and the electrical nodes of the cell (i.e. net objects). The checking algorithm is partioned into three sections, illustrated in Fig. 6: (1) finding layout shapes intersecting with the CNT curve, (2) building a connection graph of the CNT curve, and (3) checking the graph.

In the first part, each point on the CNT curve is checked separately. The aim is to find the list of layout shapes in the design which contain the specified point. On the layout level, a standard cell design consists of geometrical objects, which can be retrieved by the OA database application programming interface (API). Unfortunately, the API gives always the entire shape list, which takes a great deal of time. In contrast, the check of whether a known geometrical object contains a special point is fast. In order to improve performance, retrieving a new shape list from the API is avoided by first looking at neighboring points on the CNT curve. Considered points are:

- same point but with the former CNT offset
- preceding point with the same CNT offset
- next point with the former CNT offset

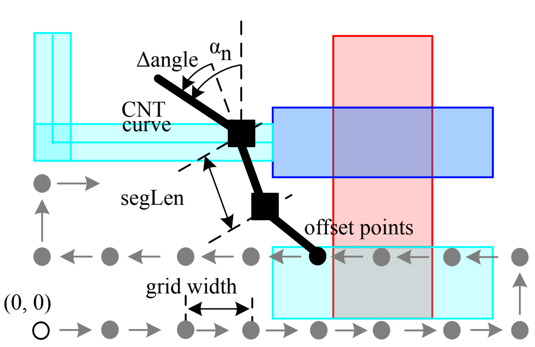

Fig. 4. Part of the CNT curve with a specific offset

If no shapes are found which are also valid for the current point on the CNT curve, a new shape list has to be retrieved. This technique accelerates the checking algorithm run time by 10%. Layout shapes can be linked with an electrical node. To reduce the complexity, the shapes are sorted and filtered. For each point on the CNT curve there remains at most one shape, leading to a list of pairs (point, shape). Only the shapes which have a link to one of the following (user-specified) electrical nodes are proccessed in the second part:

- SupplyNets: Vdd, Gnd
- GateNets: InA, InB
- OutputNet: Out

The second part builds a connection graph whose edges denote the electrical connections of the misaligned CNT. The vertices are nets (i.e. electrical nodes) from the lists above. An edge out of two nets is created if the corresponding shapes are adjacent in the list of pairs. The graph is connected and undirected, but can contain cycles. It has also no duplicated or loop-like edges. Figure 5 shows four CNT curves and the derived connection graphs.

The third part of the checking algorithm searches the graph for predefined edge types. If an edge (SupplyNet, SupplyNet) exists, this will lead to a *critical short*. Critical shorts affect other cells due to shared resources. An edge (SupplyNet, OutputNet) will give a *functional short*, e.g. Vdd2Out (stuck-at 1), Gnd2Out (stuck-at 0). If there are only edges of the types (GateNet, SupplyNet) or (GateNet, OutputNet) the result is *no short*.

For getting statistically relevant results, the checking algorithm is executed for a number of CNT curves and offset points based on the required confidence interval. We define the *total number of runs* (#Runs) and the overall *robustness* (R in [%]) as follows:

$$\#\text{Runs} = \#\text{CNT\_curves} \times \#\text{offset\_points} \tag{4}$$

$$\text{R} = 100 \times \#\text{no\_short}/\#\text{Runs} \tag{5}$$

C. Validation on NAND

In order to validate the LDRA algorithm and find an appropriate set of algorithm parameters, we designed a CNTFET-based NAND standard cell layout, displayed in Fig. 5(left). The NAND design is very similar to Fig. 1(c), except that the intra-cell connections are on the same layer as the gates or output, respectively. The intra-cell connections will act as gate

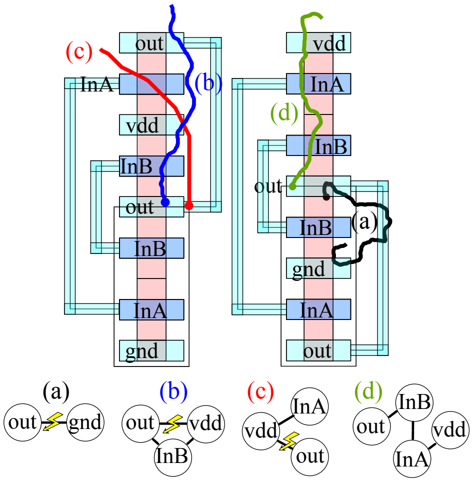

Fig. 5. Reference layout NAND & NOR with CNT curves and their connection graphs (a) $\sigma = 20°$ (b),(d) $\sigma = 10°$ (c) $\sigma = 2°$

TABLE I
INFLUENCE OF ALGORITHM PARAMETERS ON THE ROBUSTNESS, SINGLE MISALIGNED CNTS, BOLD DEFAULT VALUES, % RELATIVE TO #RUNS

no variance	1	2	3	4	5
Vdd2Out [%]	4.5	4.5	4.4	4.5	4.5
Gnd2Out [%]	0.0	0.1	0.1	0.1	0.0
no shorts [%]	95.5	95.4	95.5	95.5	95.5
σ	0	2	**10**	16	20
Vdd2Out [%]	0.0	1.8	4.5	3.6	2.8
Gnd2Out [%]	0.0	0.1	0.1	0.0	0.0
no shorts [%]	100.0	98.1	95.5	96.4	97.2
segLen/#points	10/800	15/533	20/400	30/267	40/200
Vdd2Out [%]	13.6	13.2	12.1	11.1	10.5
Gnd2Out [%]	1.6	1.8	1.7	1.2	1.2
no shorts [%]	84.8	85.0	86.2	87.7	88.3
segLen	**10**	15	20	30	40
Vdd2Out [%]	4.5	7.0	9.0	10.8	10.8
Gnd2Out [%]	0.0	0.3	0.5	1.0	1.1
no shorts [%]	95.5	92.7	90.5	88.2	88.0
#points	50	100	**200**	400	800
Vdd2Out [%]	0.1	1.6	4.5	8.8	13.8
Gnd2Out [%]	0.0	0.0	0.0	0.4	1.4
no shorts [%]	99.9	98.4	95.5	90.9	84.8
run time [s]	378	1173	1527	3021	5967
#CNT_curves	10	100	**1000**	10000	
Vdd2Out [%]	4.0	4.4	4.5	4.5	
Gnd2Out [%]	0.0	0.0	0.1	0.0	
no shorts [%]	96.0	95.6	95.5	95.5	
run time [s]	16	156	1518	15314	
grid width	10	**50**	100	150	200
Vdd2Out [%]	4.6	4.5	4.4	4.3	4.3
Gnd2Out [%]	0.0	0.1	0.0	0.1	0.0
no shorts [%]	95.6	95.5	95.6	95.7	95.7
run time [s]	37453	1542	401	171	106

or output as well. Therefore, the design cells can be stacked together in a horizontal and vertical way.

In Table I, the influence of various parameters on the robustness of the NAND is shown. No critical shorts have occured, and are therefore omitted from the table. The table is partioned into seven experiments. For each, the altered parameter is shown in the first line and results are in the succeeding lines. The percentage values represent the number of shorts over the total number of investigated cases, while *no short* gives the robustness (R) according to Eq. 5. First, LDRA is used five times with the same setup, which is indicated with no variance. Five times 1000 CNT curves each with 200 points, are generated on demand, giving almost the same results. This testifies that the 5000 normal distributed value lists [μ=0°,σ=10°] are well balanced. The results are highly reproducible, because they are obtained from a statistical approach with a sufficient sample size. As a second variation, the deviation σ is investigated from 0° to 20°. This results in a slightly increased percentage of functional shorts (Vdd2Out and Gnd2Out shorts) and consequently the robustness (indicated with *no shorts*) is decreased. A higher curvature of the misaligned CNT leads to more shorts. CNT curves with different σ are illustrated in Fig. 5. In the third experiment, the ratio of segLen to #points per CNT curve is altered. The CNT length hence remains constant, while the CNT curve is discretized. For the invesitgated range, this seems to have no effect on the results. The fourth and fifth experiment alter the segLen or #points per CNT curve respectively. Both parameters have the effect of increasing the CNT curve length, the robustness then falls e.g. from 99.9% to 84.8% which emphasizes that the ratio between CNT length and the overall layout dimensions is a key factor for robustness [11]. The amount of CNT curves (#CNT_curves) is altered between 10 and 10,000. This shows that even hundreds of CNT curves are sufficient to get a representative results. The #points per curve and the #CNT_curve increase the problem size linearly, hence the runtime increases in the same way. The #offset_points for this NAND design is 2079 if the grid width is 50. For other grid widths the runtime scales well with #Runs. The default values (bold in Table I) are chosen as a trade off runtime and accuracy: ($\sigma = 10°$, #points=200, #CNT_curves=1000, grid width=50). With a reasonable runtime of 25 minutes, the noise in the results is less than 0.1%. In conclusion, all significant influences (the curvature and length of the CNT curve) on the robustness results are reasonable and were explained. However, the ratio between the length of the CNT

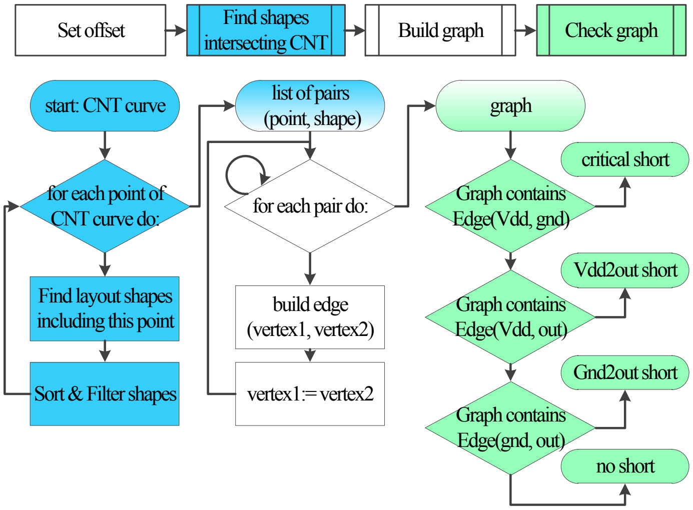
Set offset
Find shapes intersecting CNT
Build graph
Check graph
start: CNT curve
for each point of CNT curve do:
Find layout shapes including this point
Sort & Filter shapes
list of pairs (point, shape)
for each pair do:
build edge (vertex1, vertex2)
vertex1:= vertex2
graph
Graph contains Edge(Vdd, gnd)
critical short
Graph contains Edge(Vdd, out)
Vdd2out short
Graph contains Edge(gnd, out)
Gnd2out short
no short

Fig. 6. Flow of the checking algorithm

curve and the layout dimensions needs careful attention to obtain reasonable results.

IV. Experimental results

To compare the robustness of standard cell layouts against misalignment, we assume that all layouts are designed for the same process technology and therefore deal with the same CNT curves. Furthermore, we assume that only one misaligned CNT is placed in the layout at the same time. Table II presents the results for the layout types mentioned above and shows the layout dimensions, however a detailed geometric description is left out due to limited space. To model the influence of layout scaling, the CNT curve length is scaled instead (segLen=20). The remaining algorithm parameters are the default values, introduced in Section III-C. In the following selected aspects of the results are discussed in more details.

A. Comparing NAND, NOR and INV

In Figure 5 the reference layouts for NAND (left) and NOR (right) are displayed. LDRA gives for both a robustness of 95.4%, which means that in 95.4% of all investigated cases no short occur. Table II illustrates, furthermore, that nearly all of the possible shorts for this NAND2 cell are Vdd2Out shorts, because Vdd and Out have only a small gap in between. The relatively few Gnd2Out shorts and the ratio to Vdd2Out shorts can be explained by the longer distance between Out and Gnd, in combination with a small viewing angle. The results show that the probability of having misaligned CNTs, which result in critical shorts, is almost zero.

The NOR cell can derived from the NAND cell by mirroring horizontally and replacing Vdd and Gnd. Hence, the results for Vdd2Out and Gnd2Out are exchanged. The INV layout can be derived by reducing the NAND layout Fig. 5 by one gate. The INV has $1/3$ of the NAND area which lowers #Runs respectively. Due to the limited curvature of the CNT shape in combination with the small distance between Vdd, Out and Gnd the robustness is determined to be 99.12%.

B. NAND2 implementations

The NAND2 layout of Fig. 5 is the same as of Fig. 1(c) except that the latter has no participating intra-cell connections. These connections avoid critical shorts for long CNT curve, but they cause an increase of Vdd2Out and Gnd2Out Shorts of $\approx$6.5x and 1.5x, respectively. Therefore, the connections have no shielding effect and lead to lower robustness (e.g. 98.47% to 95.46%).

Comparing the results of Fig. 1(a), (c) and (b: with no treatment) shows that Fig. 1(b) has the most shorts and is therefore less robust. Figure 1(b) has the shortest pull-up path, which means that the distance between Vdd and Out is more important than the placement of the gates. The layout of Fig. 1(a) benefits from this effect compared to Fig. 1(b). With no gap in between the gates of the pull-up path, the layout of Fig. 1(c) is the most robust one in this triple. Closing the gap of Fig. 1(b) leads to the NAND version by Mitra et al. [7], Fig. 1(b: with treatment). Comparing these two version clarifies that the number of Vdd2Out shorts can be efficiently reduced by the additional treatment. However, a fraction of Vdd2Out and a negligible amount of Gnd2Out shorts remain which leads to 99.55% robustness.

The last three lines of Table II compares the robustness of Fig. 1(b) [7], Fig. 2(a), and Fig. 2(b) [3]. The layout of Fig. 2(b) gives the lowest robustness (97.77%), because the misaligned CNT curves can bypass the gates on top and on bottom, whereas on top the probability is higher due to two possible Vdd2Out paths. Figure 2(a) shows the most robust layout in this evaluation with 99.86% no shorts. As the gates wrap Vdd and Out well, the amount of shorts is almost zero. Only a few (e.g. 3096 of 2.1M) runs lead to a Vdd2Out short, resulting from the low probability of bypassing Gate A from Out to Vdd.

In conclusion, NAND layouts are more affected by Vdd2Out shorts while NOR layouts are more affected by Gnd2Out shorts. Most of the cases lead to no shorts, while the remaining cases are statistically enough to point out the robustness correctly. The distance between the nets and the view angle between two geometric objects have more influence on the robustness than the arrangement of the gates. Using longer CNT curves lowers the robustness, because the probability of shorting two geometric objects with a fixed distance increases.

C. Parameter Tuning

The robustness can obviously be improved by tuning the geometric dimensions. Before LDRA was introduced, measuring the influence of the tuning could not be quantified. Now, as the application of LDRA and the corresponding robustness results are already demonstrated, this additional merit is easy, and is shown here for two examples. The results in Table II for Fig. 1(a) assume that the ratio between the width of gate A, the gap, and the width of gate B are $25\%, 50\%, 25\%$ which leads to a robustness of 97.88% (90.62% for a segLen=20). Modifying the ratio to $33\%, 33\%, 33\%$ will increase the robustness to 98.56% (93.31% for segLen=20). For Figure 1(b) a similar improvement of 1.3% (2.7% for segLen=20) can be obtained by this modification. The second example treats the reference NAND layout shown in Fig. 5. By doubling the spacing of the intra-cell connections the robustness can be improved from 95.46% to 97.19% (90.46% to 92.21% for segLen=20). In conclusion, the proper tuning of layout parameters (with respect to the cell area) will lead to highly improved robustness results. This again shows the benefit of LDRA in exploring layout design parameters on robustness.

D. Multiple misaligned CNTs in one layout cell

Due to the quality of the technology process multiple misaligned CNTs can occur per cell. The measured robustness R reflects a probability value and has thus an exponential dependency on the number of misaligned CNTs per cell.

$$R_n = (R_1)^n \tag{6}$$

Having only n=5 misaligned CNTs at the same time, the cell layout Bobba et al. Var.1 with the robustness R_1=99.86% for a

TABLE II
ROBUSTNESS RESULTS FOR MULTIPLE LAYOUTS, DIMENSIONS GIVEN IN OA UNIT LENGTH, SINGLE MISALIGNED CNT

Layout	Reference	Width	Height	Area	#Runs	segLen=10							
						Shorts				Shorts in % to #Runs			
				[1000x]	[1000x]	Crit.	Vdd2Out	Gnd2Out	No	Crit.	Vdd2Out	Gnd2Out	No
NAND2	NAND Figure 5(a)	1600	3065	4904	2079	0	93797	616	1984587	0.00	4.51	0.03	95.46
NOR2	NOR Figure 5(b)	1600	3065	4904	2079	0	1311	93986	1983703	0.00	0.06	4.52	95.42
INV	INV	955	1725	1647	756	27	3307	3330	749336	0.00	0.44	0.44	99.12
Vulnerable	NAND Figure 1(a)	1200	2685	3222	1485	0	31100	373	1453527	0.00	2.09	0.03	97.88
w/o treatment	NAND Figure 1(b)	1200	2300	2760	1296	0	88096	345	1207559	0.00	6.80	0.03	93.18
Ideal stack	NAND Figure 1(c)	600	3065	1839	945	0	13958	465	930577	0.00	1.48	0.05	98.47
Mitra et al. [7]	NAND Figure 1(b)	1200	2300	2760	1296	0	5486	350	1290164	0.00	0.42	0.03	99.55
Bobba et al. Var 1 [3]	NAND Figure 2(a)	1295	3065	3969	2268	0	3096	0	2264904	0.00	0.14	0.00	99.86
Bobba et al. Var 2 [3]	NAND Figure 2(b)	1800	1725	3105	1406	15	19948	11399	1374638	0.00	1.42	0.81	97.77

Layout	Reference	Width	Height	Area	#Runs	segLen=20							
						Shorts				Shorts in % to #Runs			
				[1000x]	[1000x]	Crit.	Vdd2Out	Gnd2Out	No	Crit.	Vdd2Out	Gnd2Out	No
NAND2	NAND Figure 5(a)	1600	3065	4904	2079	0	187748	10489	1880763	0.00	9.03	0.50	90.46
NOR2	NOR Figure 5(b)	1600	3065	4904	2079	0	17136	172550	1889314	0.00	0.82	8.30	90.88
INV	INV	955	1725	1647	756	3082	8217	6726	737975	0.41	1.09	0.89	97.62
Vulnerable	NAND Figure 1(a)	1200	2685	3222	1458	722	130019	848	1345775	0.05	8.76	0.60	90.62
w/o treatment	NAND Figure 1(b)	1200	2300	2760	1296	1196	163604	7608	1123592	0.09	12.62	0.59	86.70
Ideal stack	NAND Figure 1(c)	600	3065	1839	945	1179	29104	6588	908129	0.12	3.08	0.70	96.10
Mitra et al. [7]	NAND Figure 1(b)	1200	2300	2760	1296	1146	17207	8304	1269343	0.09	1.33	0.64	97.94
Bobba et al. Var 1 [3]	NAND Figure 2(a)	1295	3065	3969	2268	0	7301	21	2260678	0.00	0.32	0.00	99.68
Bobba et al. Var 2 [3]	NAND Figure 2(b)	1800	1725	3105	1406	3051	41166	35554	1326229	0.22	2.93	2.53	94.33

single misaligned CNT will be lowered to R_n=99.3%, where R_n is the robustness with n misaligned CNTs. The cell Bobba et al. Var.2, which has the robustness R_1=94.33% for the single misaligned case, has only the robustness of 74.68% in case of multiple misaligned CNTs. Even if the robustness of different layouts dealing with a single misaligned CNTs appears similar, these values will spread widely for multiple tubes.

V. CONCLUSIONS

Misaligned CNTs in standard cell layouts are, at the moment, inevitable due the immature technology. Therefore, each CNTFET-based cell layout has to be designed robust against this misalignment. As the number of proposed CNTFET-based cells and circuits is rapidly increasing, an independent checking method for robustness against misalignment is necessary. In this paper, we introduced the *Layout-Driven Robustness Analysis* (LDRA), which enables the measuring of the robustness against misaligned CNTs for standard cell layouts. We showed that LDRA provides reasonable and accurate results, even for complex layout cells. LDRA can be applied to all possible layouts and enables the comparison of different layouts in terms of their robustness against misaligned CNTs. Based on this, a technique was demonstrated to use LDRA for improving the robustness iteratively. In the same way, developers are now able to harden their design rules for misaligned CNTs. LDRA is highly scalable and models varieties of CNT shapes for matching every CNT process. Even customized CNT shapes can be processed and evaluated.

Our goal for the future work is to refine the CNT shape model based on statistics from actual processed CNT devices. This will enable the extension of the proposed LDRA to model an entire active layer with randomly placed CNT shapes. In contrast to a rectangle assumption for an active area or uniformly distributed shapes, this idea is capable in determining the inter-cell robustness, the driver strength, the leakage and the dynamic behavior of standard cells while having an environment with misaligned CNTs.

REFERENCES

[1] S. Fregonese *et al.*, "Computationally Efficient Physics-Based Compact CNTFET Model for Circuit Design," *IEEE Transactions on Electron Devices*, vol. 55, no. 6, pp. 1317–1327, 2008.

[2] J. Deng and H.-S. Wong, "A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application - Part I: Model of the Intrinsic Channel Region," *IEEE Transactions on Electron Devices*, vol. 54, no. 12, pp. 3186–3194, 2007.

[3] S. Bobba, J. Zhang, A. Pullini, D. Atienza, and G. De Micheli, "Design of compact imperfection-immune CNFET layouts for standard-cell-based logic synthesis," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2009, pp. 616–621.

[4] S. Kim, I. S. Park, H.-J. Kwon, W.-J. Chang, and S. W. Lee, "Current on/off ratio enhancement through the electrical burning process in ambient with/without oxygen for the generation of high-performance aligned single-walled carbon nanotube field effect transistors," *Applied Physics Letters*, vol. 97, no. 17, p. 173102, 2010.

[5] A. Vijayaraghavan *et al.*, "Toward Single-Chirality Carbon Nanotube Device Arrays," *ACS nano*, vol. 4, no. 5, pp. 2748–2754, 2010.

[6] Y. Li *et al.*, "Preferential Growth of Semiconducting Single-Walled Carbon Nanotubes by a Plasma Enhanced CVD Method," *Nano Letters*, vol. 4, no. 2, pp. 317–321, 2004.

[7] S. Mitra, J. Zhang, N. Patil, and H. Wei, "Imperfection-immune VLSI logic circuits using Carbon Nanotube Field Effect Transistors," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2009, pp. 436–441.

[8] N. Patil, J. Deng, H.-S. P. Wong, and S. Mitra, "Automated design of misaligned-carbon-nanotube-immune circuits," in *Proceedings of the 44th annual Design Automation Conference*, ser. Design Automation Conference (DAC), 2007, pp. 958–961.

[9] A. Lin, N. Patil, H. Wei, S. Mitra, and H.-S. Wong, "ACCNT - A Metallic-CNT-Tolerant Design Methodology for Carbon-Nanotube VLSI: Concepts and Experimental Demonstration," *IEEE Transactions on Electron Devices*, vol. 56, no. 12, pp. 2969–2978, 2009.

[10] "Mathematica C++ API Online Documentation," 2011. [Online]. Available: http://reference.wolfram.com/mathematica/tutorial/MathLinkDeveloperGuide-Unix.html#32679

[11] J. Zhang *et al.*, "Carbon nanotube circuits in the presence of carbon nanotube density variations," in *Proceedings of the 46th Annual Design Automation Conference*, ser. Design Automation Conference (DAC), 2009, pp. 71–76.