

Timing-Driven Placement for Carbon Nanotube Circuits

Chen Wang¹, Li Jiang¹, Shiyao Hu², Tianjian Li¹, Xiaoyao Liang¹, Naifeng Jing¹ and Weikang Qian¹

¹ Shanghai Jiao Tong University, Shanghai, China

² Michigan Technological University, Houghton, Michigan, USA

Email: ¹{wangchen_2011, ljiang_cs, ltj2013, liang-xy, sjtuj, qianwk}@sjtu.edu.cn, ²shiyao@mtu.edu

Abstract—Carbon nanotube field effect transistors (CNFETs), which use carbon nanotubes (CNTs) as the transistor channel, are promising substitution of conventional CMOS technology. However, due to the stochastic assembly process of CNTs, the number of CNTs in each CNFET has a large variation, resulting in a vast circuit delay variation and timing yield degradation. To overcome it, we propose a timing-driven placement method for CNFET circuits. It exploits a unique feature of CNFET circuits, namely, asymmetric spatial correlation: CNFETs that lie along the CNT growth direction are highly correlated in terms of their electrical properties. Our method distributes CNFETs of the same critical paths to different rows perpendicular to the CNT growth direction during both global and detailed placement phases, while optimizing the timing of these critical paths. Experimental results demonstrated that our approach reduces both the mean and the variance of circuit delay, leading to an improvement in timing yield.

I. INTRODUCTION

Power consumption has become one of the paramount concerns in designing very large scale integrated (VLSI) circuits as CMOS technology is scaled into the nanometer regime. To address this challenge, alternatives to CMOS technology are being actively explored. Among many choices, carbon nanotube field-effect transistors (CNFETs) are considered as one promising alternative to CMOS devices.

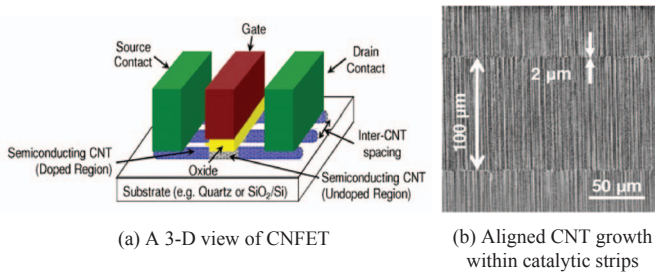


Fig. 1: CNFETs and CNTs [1].

As shown in Fig. 1(a), CNFET devices use carbon nanotubes (CNTs) as the transistor channel. They have a number of remarkable advantages over traditional MOSFETs, including strong driving capability and much smaller leakage current than CMOS devices [2]. Recent studies showed that compared to conventional CMOS circuits, those built with CNFETs could potentially improve the energy-delay product, a measure of energy efficiency, by more than an order of magnitude [3]. However, in order to build VLSI circuits entirely with CNFETs, some inherent limitations of CNFETs must be addressed [4] [5]. These include the misalignment of carbon nanotubes (CNTs), the existence of metallic CNTs, and CNT density variation. Recent advances in device and circuit design technology have provided effective solutions to overcome the challenges due to misaligned and metallic CNTs [6]. However, few effective methods are known to solve the CNT density variation problem, which significantly affects both the reliability and the performance of the circuits built with CNFETs.

CNT density variation is caused by the randomness during the CNT manufacturing process. The state-of-the-art fabrication method

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No. 61204042 and 61472243, in part by Shanghai Science and Technology Committee under Grant No.15YF1406000, and in part by U.S. NSF CAREER Award CCF-1349984.

for CNTs is using chemical vapor deposition. However, such a CNT growth technique does not generally allow precise control over the locations of the individual CNTs [1], as shown in Fig. 1(b). This causes the spacing between CNTs to vary significantly, leading to huge CNT density (i.e., CNT count per unit width) variations [7] [8]. Since the driving current of a CNFET is proportional to the CNT count in its channel, different CNFETs on the chip may have significantly different driving currents. As a consequence, there exhibits a vast variation in the delay of CNFET circuit, which in turn degrades timing yield [9].

To overcome the above challenges caused by the CNT density variation, in this paper, we proposed a timing-driven placement algorithm for CNFET circuits. In the previous work [10], a path-healing method that spreads critical-path modules across different columns of CNTs is applied during the detailed placement phase. This method effectively reduces the variance of the path delay. However, since the method only perturbs the cell locations during the detailed placement phase, the searching space of good CNT-variation-aware placements is strictly restricted. Furthermore, their method fails to consider the influence of interconnect wiring on the delay. From our experimental experiences, although sometimes a perturbation could improve the path delay variance, it could adversely increase the interconnect wiring, leading to an increase in the mean path delay. If the increase in the mean path delay outweighs the reduction of the path delay variance, the benefit of reducing the path delay variance is nullified.

In this work, we propose a CNT-density-variation-aware placement flow containing both the global and the detailed placement phases. By including the global placement phase, we enlarge the search space to find a better solution. In both phases, the modules of the same critical paths are distributed to different rows, which helps reduce the variance of the total path delay. At the mean time, we optimize the wirelength, which helps reduce the mean value of the total path delay. The experimental results showed that by applying our placement algorithm, the mean and variance of the circuit delay are reduced and the timing yield is greatly improved.

The main contributions in our work are summarized as follows:

- We propose a novel global placement algorithm for CNFET circuits based on the force-directed placement framework. In order to reduce the path delay variance, we introduce a new force which distributes the gates on the same critical path to different rows.
- We propose a detailed placement algorithm which further spreads the gates on the same path to different rows, while at the mean time optimizing the interconnect wirelength.

The remainder of this paper is organized as follows. In Section II, we introduce some background on CNFET gate, CNT density variation, and the force-directed quadratic placement algorithm. In Section III, we present our basic idea. In Sections IV and Section V, we elaborate our proposed global placement algorithm and detailed placement algorithm, respectively. In Section VI, we present the experimental results. Finally, we conclude the paper in Section VII.

II. PRELIMINARIES AND ASSUMPTIONS

A. CNFET Standard Cell Layout Style

There are two basic layout styles for CNFET standard cells. In the first layout style, the CNT growth direction is perpendicular to the row direction along which the cells are aligned, as shown in Fig. 2 on the

left. In the other layout style, these two directions are parallel to each other, as shown in Fig. 2 on the right. In this work, we consider the latter layout style, which is different than the layout style considered in [10].

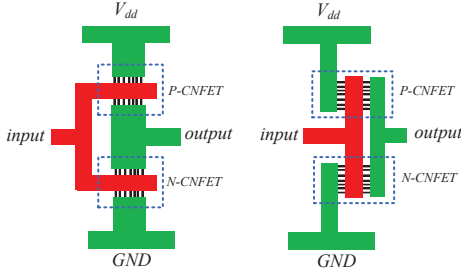


Fig. 2: Two standard cell layout styles with different CNT growth directions with regard to the Vdd/GND rails. On the left, the CNT growth direction is perpendicular to the Vdd/GND rails. On the right, the CNT growth direction is parallel to the Vdd/GND rails.

B. CNT Density Variation

Due to its stochastic assembly process, CNT density, defined as the CNT count per unit width, has a significant variation across the whole chip. However, this variation exhibits a strong spatial asymmetry. CNT densities in different locations along the CNT growth direction are highly correlated, while the densities in locations not along the CNT growth direction are highly independent. For example, as shown in Fig. 3 (a), the number of CNTs covered by the NOR2 gate, the inverter, and the NAND2 gate are all four, as these three gates are at the same row. In contrast, as shown in Fig. 3 (b), if these three gates are distributed to different rows, the number of CNTs covered by them are different. Strictly speaking, CNT densities are not the same for different locations along the CNT growth direction. However, [1] showed that the correlation coefficient of CNT count along its growth direction is above 0.9 up to a distance of $6\mu m$. Therefore, in this work, we assume that the CNT densities along the CNT growth direction do not change.

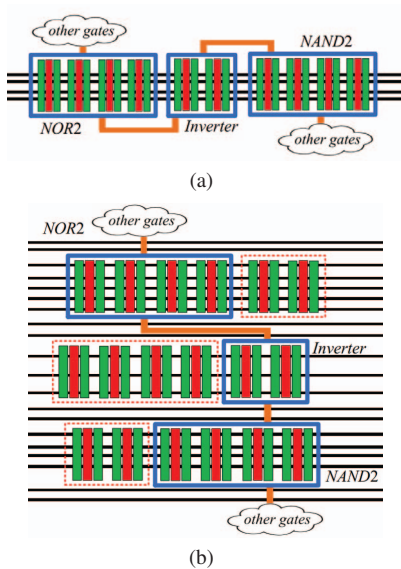


Fig. 3: Gates of the chosen critical path allocated to the same and the different CNT rows.

Since in our work we focus on a layout style in which the CNT growth direction is parallel to the row direction in the placement, the

CNT counts for the gates at the same row are the same, while the CNT counts for the gates at different rows are totally independent. As a result, the gates at the same row have similar electric properties such as gate delay and driving capability.

Due to variation, there could be no CNTs in a CNFET, but the probability is very small. Furthermore, our focus is on timing yield instead of functional yield. Thus, in our study, we assume that there is no zero-CNT transistor. A CNFET may also contain metallic CNTs. However, there exist well-established methods to remove them [11]. Thus, we assume that there are no metallic CNTs.

C. CNFET Gate Delay Model

The CNFET gate delay consists of two parts: the intrinsic delay of this gate, and the delay caused by the output load of the gate. Consider the case that one gate drives another gate. The gate delay τ_{gate1} of the driving gate can be represented as

$$\tau_{gate1} = \frac{(C_{gate1,intrinsic} + C_{load})V_{supply}}{n_1 I_0}, \quad (1)$$

where $C_{gate1,intrinsic}$ is the intrinsic gate capacitance of the driving gate, C_{load} is the load capacitance, V_{supply} is the supply voltage, n_1 is the CNT count of the driving gate, and I_0 is the on-current of each CNT. If the inter-CNT pitch is too small, screening effect occurs, which degrades I_0 [12]. This will increase both the gate delay and interconnect delay. For simplicity, we ignore the screening effect in this work.

C_{load} consists of the input capacitance of the driven gate $C_{gate2,input}$ and the interconnect capacitance $C_{interconnect}$, as shown by the following equation:

$$C_{load} = C_{gate2,input} + C_{interconnect}. \quad (2)$$

The input capacitance of the driven gate $C_{gate2,input}$ can be calculated as

$$C_{gate2,input} = n_2 \cdot C_{g-total(CNT),1}, \quad (3)$$

where n_2 is the CNT count of the driven gate and $C_{g-total(CNT),1}$ is the capacitance of one CNT [12].

In our study, the distribution of the CNT count of a CNFET with regard to its gate width is approximated as a Gaussian distribution as proposed in [8]:

$$n(W) \sim Gauss\left(\frac{W}{\mu_s}, \frac{W\sigma_s^2}{\mu_s^3}\right), \quad (4)$$

where W is the CNFET gate width, μ_s is the semiconducting CNT inter-pitch mean value, and σ_s is the semiconducting CNT inter-pitch standard variation value.

D. Force-directed Quadratic Placement

Among all kinds of VLSI placement algorithms, the force-directed quadratic placement algorithm is the one with outstanding efficiency and placement quality [13]. The force-directed quadratic placement method, such as Kraftwerk2 [13], iteratively finds the optimal location for each movable module to minimize the total half perimeter wirelength. Such a minimization problem can be equivalently modeled as a mechanic system with several forces applied on the movable modules. The optimal solution corresponds to the equilibrium status where the sum of all the forces is zero.

In a force-directed quadratic placement, typically there are three forces involved, which are the net force, the hold force, and the move force. All forces can be decomposed into an x -component and a y -component. Since these two components are similar, we show the modeling of the x -components of these forces.

The net force is caused by the net connection between modules, which is defined as

$$\mathbf{f}_x^{net} = \mathbf{C}_x \mathbf{x} + \mathbf{d}_x, \quad (5)$$

where the vector \mathbf{x} is the new x -location of all movable modules, the matrix \mathbf{C}_x represents the connection among all the movable modules

along the x -axis, and the vector \mathbf{d}_x represents the connection between movable and fixed modules along the x -axis. The matrix \mathbf{C}_x and the vector \mathbf{d}_x are built from each two-point connection as follows. Suppose that two modules i and j are connected together and the x -component of their connection weight is $w_{ij}^{(x)}$. If both of the modules are movable, then $w_{ij}^{(x)}$ is added to the entries c_{ii} and c_{jj} on the diagonal of the matrix \mathbf{C}_x , and is subtracted from the off-diagonal entries c_{ij} and c_{ji} . If module i is movable and module j is fixed, then $w_{ij}^{(x)}$ is added to c_{ii} , and $w_{ij}^{(x)} \cdot x_j$ is subtracted from the entry d_i of the vector \mathbf{d}_x . If both modules are fixed, then they do not contribute to \mathbf{C}_x and \mathbf{d}_x .

The hold force is applied to balance the net force, which is defined as

$$\mathbf{f}_x^{\text{hold}} = -(\mathbf{C}_x \mathbf{x}' + \mathbf{d}_x), \quad (6)$$

where \mathbf{x}' is the x -location of movable modules at the beginning of each placement iteration.

The move force is used to reduce the overlap of modules and distribute them evenly in the placing area. It is defined as

$$\mathbf{f}_x^{\text{move}} = \mathbf{C}_x^{\text{move}} (\mathbf{x} - \mathbf{x}_{\text{target}}^{\text{move}}), \quad (7)$$

where $\mathbf{C}_x^{\text{move}}$ is a diagonal matrix collecting the connection weight between the movable modules and their target points of the move force, and $\mathbf{x}_{\text{target}}^{\text{move}}$ is calculated as

$$\mathbf{x}_{\text{target}}^{\text{move}} = \mathbf{x}' - \frac{\partial}{\partial x} \Phi_x^{\text{move}} \Big|_{\mathbf{x}'}, \quad (8)$$

where Φ_x^{move} is a potential matrix calculated by assigning charge to the modules and solving the Poisson's equation, and $\frac{\partial}{\partial x} \Phi_x^{\text{move}} \Big|_{\mathbf{x}'}$ is a vector collecting the x -gradient of the potential at the center location of each module.

At equilibrium, we must have

$$\mathbf{f}_x = \mathbf{f}_x^{\text{hold}} + \mathbf{f}_x^{\text{net}} + \mathbf{f}_x^{\text{move}} = \mathbf{0}. \quad (9)$$

By solving this equation, the new x -position of all the modules can be obtained. The y -positions of modules can be calculated in the same way.

III. THE BASIC IDEA

Since the CNT density variation exhibits a strong spatial asymmetry, it is natural to ask how this asymmetry affects the timing of a path under different layouts. We consider two extreme layouts of a path. In the first layout, all the gates on the path are put on the same row. An example of this is shown in Fig. 3(a), where the NOR2 gate, the inverter, and the NAND2 gate on the same path are placed on the same row. In the second layout, all the gates on the path are distributed to different rows. An example of this is shown in Fig. 3(b), where the NOR2 gate, the inverter, and the NAND2 gate on the same path are placed on three different rows. We assume the number of gates on the path is n and the delay of the i -th gate is a random variable D_i . For simplicity, we assume that the delay of each gate obeys the same distribution, of which the mean is μ and the variance is σ^2 . The path delay S can be calculate as

$$S = \sum_{i=1}^n D_i,$$

which is also a random variable. The mean of the path delay can be calculated as

$$E[S] = E \left[\sum_{i=1}^n D_i \right] = \sum_{i=1}^n E[D_i] = n\mu.$$

Therefore, the mean path delay is independent of the path layout.

If all the gates are put on the same row, then their delays are the same, i.e., $D_1 = D_2 = \dots = D_n$. Thus, the variance of the path

delay can be calculated as

$$\text{Var}[S] = \text{Var} \left[\sum_{i=1}^n D_i \right] = \text{Var}[nD_1] = n^2 \text{Var}[D_1] = n^2 \sigma^2.$$

If all the gates are distributed to different rows, then their delays are independent. The variance of the path delay can be calculated as

$$\text{Var}[S] = \text{Var} \left[\sum_{i=1}^n D_i \right] = \sum_{i=1}^n \text{Var}[D_i] = n\sigma^2,$$

where the second equality is due to the basic property of independent random variables. Comparing the above two cases, we can conclude that distributing the gates on a path to different rows will reduce the variance of the path delay. In general, it can be shown that the more the gates are distributed to different rows, the smaller the path delay variance is.

In this work, we exploit the above basic theoretical observation in our placement method, which tries to distribute the modules on the same critical path to different rows as much as possible. We note that a similar idea is also used in [10]. However, in that work, the distribution of cells is only considered in the detailed placement phase, which is the last stage in the placement. As a result, that method only searches a small subset of all the placement solutions. Furthermore, the effect of the interconnect is not considered in that work. It only focuses on reducing the variance of the path delay. However, if not done properly, this could adversely increase the total wirelength of the critical path, leading to a degradation in the nominal delay value. If the nominal delay is significantly increased, the benefit of reducing the delay variance is compromised.

To address the above problems, in this work, we also consider distributing the modules on critical paths to different rows in the global placement phase, in addition to the detailed placement phase. Furthermore, we take into consideration the effect of interconnect on the total path delay, which leads to a more accurate delay estimate. We not only try to reduce the delay variance, but also try to control the nominal delay value by optimizing the total wirelength on the critical path.

IV. THE PROPOSED GLOBAL PLACEMENT ALGORITHM

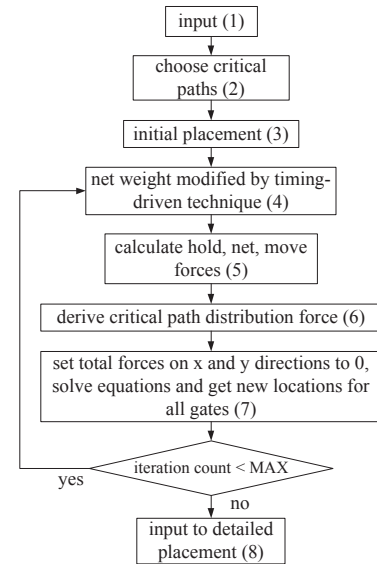


Fig. 4: Procedure of the proposed timing-driven global placement algorithm for carbon nanotube circuits.

To realize the basic idea, we propose a novel placement algorithm that attempts to distribute the gates on the critical paths to different

rows as much as possible. It achieves this in both the global placement phase and the detailed placement phase. In this section, we discuss the proposed global placement method. The proposed detailed placement method will be elaborated in Section V.

A. Overview of the Global Placement Method

Our global placement method is based on the force-directed quadratic placement. The main procedure is shown in Fig. 4.

The input to our global placement procedure includes a gate-level netlist and a critical path report, which can be generated by a standard timing analysis tool. From the input information, a number of top-ranked critical paths are selected. Our algorithm will spread gates on these paths to different rows. The details on how to select the critical paths will be discussed in Section IV-B.

After the initial placement, the main placement iteration begins. In each iteration, a net-based timing optimization technique proposed in [14] is first applied to modify the weight of the nets on each chosen critical paths, the details of which will be shown in Section IV-C. Next, the net force, hold force, and the density-based move force, introduced in Section II-D, are calculated. Then, the algorithm calculates a new distribution force for each critical-path module which reduces the y -direction overlap of the module with the other modules on the same critical path. The details about the new distribution force will be shown in Section IV-D. Up to this point, we have derived four forces for each module. Then, we add all the proper forces for each module and set their sum to zero. By solving this equation, the new location of each module is found. This concludes the main iteration.

Once the maximum iteration number is reached, the global placement algorithm generates an intermediate placement result which will be fed into the detailed placement procedure for further processing.

B. Choosing Critical Paths

In our method, we choose a number of most critical paths that will potentially dominate the circuit delay to optimize their timing. The gates on these chosen paths will be distributed to different rows. How to choose these paths is a crucial problem.

The first question is how to estimate the path delay, which is used as a criteria in choosing the paths for optimization. Since at this early stage, placement and routing have not been performed yet, the exact delay information is unavailable. Furthermore, the delay value has variation for each chip even after placement and routing, since it depends on the CNT count of each gate, which is random. However, since the delay of a path is roughly proportional to the gate count and the total gate delay, therefore, we estimate the path delay using the sum of the mean CNFET gate delays, which are calculated by the delay model discussed in Section II-C.

The second question is how many number of critical paths we should choose. On the one hand, if we choose too few, then some paths not chosen could become a critical path due to delay variation. Then the effectiveness of the proposed technique is compromised. On the other hand, if we choose too many critical paths, the computation workload will be significantly increased. By analyzing the critical path report, we found that many critical paths in a circuit share a large number of common gates, and hence, can be grouped together. Based on this observation, our strategy is to first cluster the critical paths sharing a large number of common gates together, and then choose the longest critical path from each group.

C. Net-based Timing-driven Placement

In our proposed global placement, a net-based timing-driven technique similar to the one proposed in [14] is applied to optimize the delay of the critical paths. During each placement iteration, we first estimate the interconnect delay of each net by applying the Elmore delay model on the bounding box enclosing the net. Then we add the interconnect delays and the gate delays on each critical path to obtain a more accurate delay estimate. For each chosen critical path, we assign to it a criticality value, which is initialized to zero. During each placement iteration, the criticality values of the critical

paths change incrementally. For all the nets on the critical paths with larger criticality, their connection weights (i.e., the value $w_{ij}^{(x)}$ in Section II-D) are increased more, leading to larger net forces to shrink these nets. Specifically, in the i -th iteration, if a path is among the top 20% critical paths, its criticality at the i -th iteration $c_i = (c_{i-1} + 1)/2$; otherwise, $c_i = c_{i-1}/2$. For each net, the net weight is initialized as 1, and the net weight at the i -th iteration is $w_i = w_{i-1}(1 + c_i)$. By this means, the total interconnect length on the critical path is reduced, and the delay of the critical paths is optimized.

D. Distribute Critical Path Modules

In our proposed global placement algorithm, a new 1-dimensional spring-like force is introduced to distribute the modules on each chosen critical path to different rows. The force is defined as

$$\mathbf{f}^{\text{CP}} = \mathbf{f}_y^{\text{CP}} = \mathbf{C}_y^{\text{CP}}(\mathbf{y} - \mathbf{y}_{\text{target}}^{\text{CP}}), \quad (10)$$

where \mathbf{C}_y^{CP} is a diagonal matrix composed of the weight for each connection between a module and its target point, \mathbf{y} is a vector containing the new y -location of each module, and $\mathbf{y}_{\text{target}}^{\text{CP}}$ is a vector containing the y -location of the target point for each module. It should be noted that the distribution force \mathbf{f}^{CP} is a 1-dimensional force, which only has a component along the y -axis.

The target y -location for each module is determined as follows. First, we add a virtual charge to all the critical-path modules and compute the potential distribution along the y -axis Φ_y^{CP} by solving a 1-dimensional Poisson's equation. Next, the gradient of the potential along the y -axis $\frac{\partial}{\partial y}\Phi_y^{\text{CP}}$ is computed at the center of each chosen module. The target y -location is finally computed as

$$\mathbf{y}_{\text{target}}^{\text{CP}} = \mathbf{y}' - \frac{\partial}{\partial y}\Phi_y^{\text{CP}}\Big|_{\mathbf{y}'}, \quad (11)$$

where \mathbf{y}' is a vector containing the current y -location of each module.

At equilibrium, there are four forces applied to each module on the chosen critical paths: the net force, the hold force, the move force, and the new distribution force. For the other modules, only three common forces are applied to them: the net force, the hold force, and the move force.

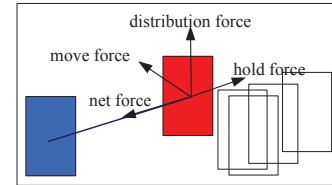


Fig. 5: All forces on a critical-path module during one placement iteration.

The forces on a critical-path module are illustrated in Fig. 5. In the figure, both the red and the blue modules are two critical-path modules. The red module is subject to all the four forces. The new distribution force on the red module moves it away from the blue module to reduce their overlap along the y -axis. The net force pulls the red module towards the blue module. The hold force on the red module is in the opposite direction of the net force. The move force is caused by module overlapping, which moves the red module away from crowded places. At equilibrium, all these forces are balanced, and the following two equations should be satisfied simultaneously.

$$\mathbf{f}_x^{\text{tot}} = \mathbf{f}_x^{\text{hold}} + \mathbf{f}_x^{\text{net}} + \mathbf{f}_x^{\text{move}} = \mathbf{0} \quad (12)$$

$$\mathbf{f}_y^{\text{tot}} = \mathbf{f}_y^{\text{hold}} + \mathbf{f}_y^{\text{net}} + \mathbf{f}_y^{\text{move}} + \mathbf{f}_y^{\text{CP}} = \mathbf{0} \quad (13)$$

By solving these equations, we get the new x and y positions of all the modules as

$$\mathbf{x} = \mathbf{x}' - [\text{Inv}(\mathbf{C}_x + \mathbf{C}_x^{\text{move}})](\mathbf{C}_x^{\text{move}}\Phi_x^{\text{move}}), \quad (14)$$

$$\mathbf{y} = [\text{Inv}(\mathbf{C}_y^{\text{move}} + \mathbf{C}_y + \mathbf{C}_y^{\text{CP}})] [\mathbf{C}_y^{\text{move}}(\mathbf{y}' - \Phi_y^{\text{move}}) + \mathbf{C}_y \mathbf{y}' + \mathbf{C}_y^{\text{CP}} \mathbf{y}_{\text{target}}^{\text{CP}}], \quad (15)$$

where \mathbf{x} and \mathbf{y} are the vectors representing the new positions of the modules, \mathbf{x}' and \mathbf{y}' are the vectors representing the positions of the modules at the beginning of each iteration, \mathbf{C}_x and \mathbf{C}_y are connectivity matrices composed of connection weights, $\mathbf{C}_x^{\text{move}}$ and $\mathbf{C}_y^{\text{move}}$ are the matrices with move force weights, Φ_x^{move} and Φ_y^{move} are vectors containing the gradient of potential caused by module overlapping at their previous positions, \mathbf{C}_y^{CP} is a diagonal matrix composed of the weight for each connection between a module and its target point, and $\mathbf{y}_{\text{target}}^{\text{CP}}$ is a vector containing the y -location of the target point of each module.

V. THE PROPOSED DETAILED PLACEMENT ALGORITHM

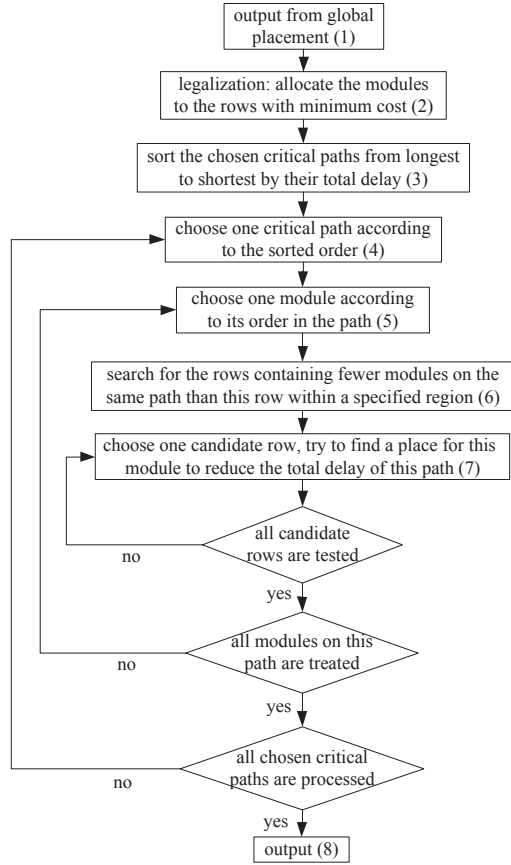


Fig. 6: Procedure of the proposed timing-driven detailed placement algorithm for carbon nanotube circuits.

After the global placement, the detailed placement is performed to remove overlap among modules and further improve the critical path timing. The detailed placement procedure is shown in Fig. 6.

The first step is legalization, which eliminates the overlap. We use a method similar to [15] for this step. We first sort all these modules in ascending order of their x -locations. Then each module is allocated to the row which gives the minimum value of the cost function $\Delta x^2 + c_{xy} \cdot \Delta y^2$, where Δx and Δy are the x and y distance from the current location of the module to the leftmost available location of each row, and c_{xy} is a coefficient.

After this, a greedy algorithm is applied to further distribute the modules on the same critical path to different rows while reducing the total estimated path delay. First, the chosen critical paths are sorted in descending order of their total delays. Then we select each critical path one by one in the sorted order for further optimization. For a

specific path, each module on the path is considered one by one in the order from the beginning of the path to the end.

For each module, we will decide the candidate rows to which the module can be relocated to improve the distribution of the critical-path modules. The candidate rows are searched within a bounding box defined by this module and its preceding and succeeding modules. If a row contains fewer modules on the same critical path than the current row the module belongs to, then it is identified as a candidate row. For each candidate row, a location within the bounding box is further searched to allocate this module to reduce the path delay. If such a location is found, then the module is moved to this new location. This procedure iterates until all the modules on all the chosen critical paths are processed. By this means, the modules on the chosen critical paths are further distributed to different rows, and the total delays of these critical paths are further reduced. We can run a number of iterations of this detailed placement process to get the most satisfactory solution.

VI. EXPERIMENTAL RESULTS

In this section, we study the effectiveness of our proposed placement algorithm. The technology node investigated by us is 16nm, i.e., the gate width and length of the CNFET are 16nm.

A. Experiment Setup

For each benchmark, four placement methods are applied: global placement and legalization by *Cadence SoC Encounter* (PL1), global placement and legalization by *Cadence SoC Encounter* and detailed placement by our proposed method (PL2), our proposed global placement and legalization method (PL3), and our proposed global placement and legalization method together with our proposed detailed placement method (PL4).

Since CNT density variation causes the delay variation, the quality of a placement should be measured using statistical values. To obtain the statistics of delay for each placement, 2000 Monte-Carlo simulations are performed. A number of the critical paths with the longest nominal delay values are chosen for delay evaluation. Note that this set of critical paths includes those paths which are the optimization target of our placement algorithm. In each Monte-Carlo simulation, the largest delay value among these paths is recorded as the circuit delay.

In each simulation, a unique CNT count value for each row is randomly generated following the probability density function (4). This CNT count is assigned to each gate on the same row. With the CNT count, we compute the gate delay using the model shown in Section II-C. The interconnect delay is computed by using the Elmore delay model under the assumption that the nets are routed in a star shape [16]. Some important parameters used in the experiments are listed in Table I.

The circuit delays for all the 2000 Monte-Carlo simulations are collected. Their distribution is used to find the 99% timing yield margin.

B. Experiments on Multiplier

In our experiment, the unsigned 32-bit multiplier is used as the benchmark. We produce the netlist and critical path delay reports using *Synopsys Design Compiler*.

TABLE I: Parameters used in Monte-Carlo simulation.

parameter	value
CNFET gate width	16 nm
CNFET gate length	16 nm
CNT inter-pitch mean	4 nm
CNT inter-pitch standard deviation	2 nm
interconnect unit resistance	5.38 $\Omega/\mu\text{m}$
interconnect unit capacitance	0.16 fF/ μm

Table II shows the delay statistics for the four placement results. Comparing PL2 with PL1, we can see that the mean and standard

TABLE II: Results of delay mean and standard deviation of the unsigned 32-bit multiplier.

	PL1	PL2	PL3	PL4
circuit delay mean (ps)	780.5	684.6	656.3	582.7
circuit delay standard deviation (ps)	96.0	77.7	84.2	82.7
99% timing yield margin (ps)	1097.8	924.4	962.0	888.6

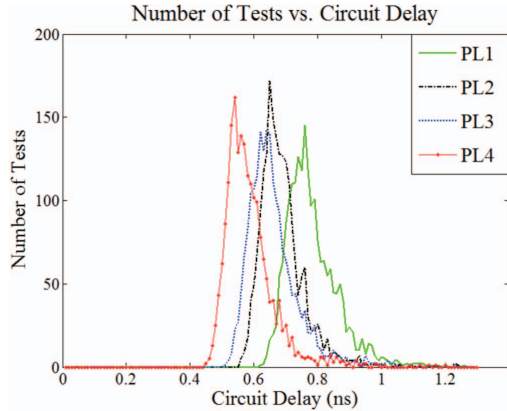


Fig. 7: Circuit delay distribution for 2000 Monte-Carlo simulation tests of the unsigned 32-bit multiplier.

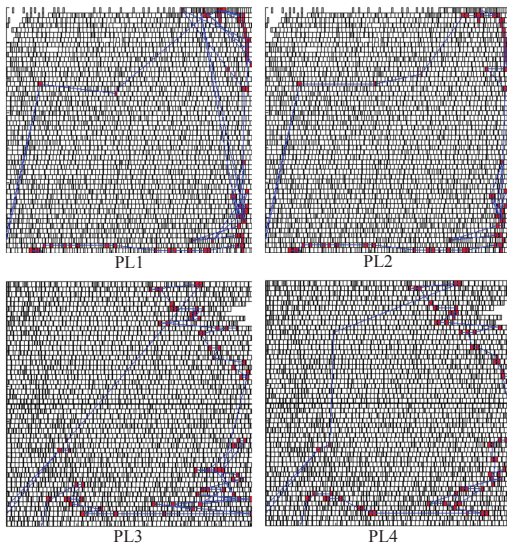


Fig. 8: Circuit layouts for different placements. Each white block represents a gate. The red cells are the gates on the longest critical path. The blue lines show the connection of the longest critical path.

deviation of the delay of the layout produced by PL2 is much smaller, showing the effectiveness of our proposed detailed placement. Comparing PL3 with PL1, we can see that the mean value of the delay of the circuit produced by PL3 is much smaller, showing that our proposed global placement effectively optimizes the circuit delay. Among all four placement methods, PL4 gives the best results. The circuit layouts for these four placements are shown in Fig. 8. The side length of each layout is $24.89\mu m$, and the layout area is $619.46\mu m^2$.

In Table II, we further compare the 99% timing yield margins for different placement results. The 99% timing yield margin is defined as a boundary value such that the percentage of the tested circuits with delay values below that boundary value is 99%. Again, the results from the table demonstrate the effectiveness of our proposed global and detailed placement methods. Finally, The delay distribution of the

2000 Monte-Carlo simulations is shown in Fig. 7. It shows a same trend as the 99% timing yield margin.

It should be noted that the screening effect between the CNTs occurs when the inter-CNT pitch is small. This effect degrades the CNT drive current and worsens the circuit delay [12]. As a result, the mean and deviation of the delay distribution will increase if the screening effect is considered. Its further influence on different placement results will be studied in more details in our future work.

VII. CONCLUSIONS

In this paper, we propose a new timing-driven placement flow for carbon nanotube circuits exploiting a unique feature of CNFET circuits, namely, asymmetric spatial correlation. We consider both global and detailed placement phases. The proposed global placement algorithm is a force-directed quadratic method. A new force which distributes the cells on the same path to different rows is introduced. The proposed detailed placement algorithm further spreads all the modules on the same path to different rows while optimizing the estimated delay of this path.

REFERENCES

- [1] J. Zhang, N. Patil, A. Hazeghi, H.-S. Wong, and S. Mitra, "Characterization and design of logic circuits in the presence of carbon nanotube density variations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1103–1113, 2011.
- [2] J. Deng and H.-S. Wong, "A compact SPICE model for carbon-nanotube field-effect transistors including nonidealities and its application-part II: Full device model and circuit performance benchmarking," *IEEE Trans. Electron Devices*, vol. 54, no. 12, pp. 3195–3205, 2007.
- [3] G. Hills, J. Zhang, C. Mackin, M. Shulaker, H. Wei, H.-S. P. Wong, and S. Mitra, "Rapid exploration of processing and design guidelines to overcome carbon nanotube variations," in *DAC'13*. ACM, 2013, p. 105.
- [4] J. Zhang, A. Lin, N. Patil, H. Wei, L. Wei, H.-S. Wong, and S. Mitra, "Robust digital VLSI using carbon nanotubes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 4, pp. 453–471, 2012.
- [5] N. Patil, A. Lin, J. Zhang, H.-S. Wong, and S. Mitra, "Digital VLSI logic technology using carbon nanotube FETs: Frequently asked questions," in *DAC'09*, pp. 304–309.
- [6] N. Patil, J. Deng, A. Lin, H.-S. Wong, and S. Mitra, "Design methods for misaligned and mispositioned carbon-nanotube immune circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1725–1736, 2008.
- [7] J. Zhang, S. Bobba, N. Patil, A. Lin, H.-S. Wong, G. De Micheli, and S. Mitra, "Carbon nanotube correlation: Promising opportunity for CNFET circuit yield enhancement," in *DAC'10*, pp. 889–892.
- [8] J. Zhang, N. Patil, A. Hazeghi, and S. Mitra, "Carbon nanotube circuits in the presence of carbon nanotube density variations," in *DAC'09*, pp. 71–76.
- [9] B. Ghavami, M. Raji, and H. Pedram, "Timing yield estimation of carbon nanotube-based digital circuits in the presence of nanotube density variation and metallic-nanotubes," in *ISQED'11*, pp. 1–8.
- [10] M. Beste, S. Kiamehr, and M. B. Tahoori, "Layout-aware delay variation optimization for CNFET-based circuits," in *VLSID'14 and ICES'14*. IEEE, 2014, pp. 393–398.
- [11] M. Shulaker, J. Van Rethy, G. Hills, H. Wei, H.-Y. Chen, G. Gielen, H.-S. Wong, and S. Mitra, "Sensor-to-digital interface built entirely with carbon nanotube FETs," *Solid-State Circuits, IEEE Journal of*, vol. 49, no. 1, pp. 190–201, 2014.
- [12] N. Patil, J. Deng, S. Mitra, and H.-S. Wong, "Circuit-level performance benchmarking and scalability analysis of carbon nanotube transistor circuits," *IEEE Trans. Nanotechnol.*, vol. 8, no. 1, pp. 37–45, 2009.
- [13] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Kraftwerk2-a fast force-directed quadratic placement approach using an accurate net model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 8, pp. 1398–1411, 2008.
- [14] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," in *DAC'98*. ACM, 1998, pp. 269–274.
- [15] D. Hill, "Method and system for high speed detailed placement of cells within an integrated circuit design," Apr. 9 2002, US Patent 6,370,673.
- [16] B. M. Riess and G. G. Ettl, "SPEED: Fast and efficient timing driven placement," in *ISCAS'95*, vol. 1. IEEE, 1995, pp. 377–380.