2017

# DESIGN AUTOMATION FOR CARBON NANOTUBE CIRCUITS CONSIDERING PERFORMANCE AND SECURITY OPTIMIZATION

Lin Liu
*Michigan Technological University*, lliu7@mtu.edu

# DESIGN AUTOMATION FOR CARBON NANOTUBE CIRCUITS CONSIDERING PERFORMANCE AND SECURITY OPTIMIZATION

By

Lin Liu

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Computer Engineering.

Department of Electrical and Computer Engineering

Dissertation Advisor: *Dr. Shiyan Hu*

Committee Member: *Dr. Zhaohui Wang*

Committee Member: *Dr. Nilufer Onder*

Committee Member: *Dr. Ye Sun*

Department Chair: *Dr. Daniel R. Fuhrmann*

# Dedication

To my parents, advisor and friends

I dedicate this dissertation to my dear parents, my advisor and my friends.

# Contents



[1]The material contained in this chapter was previously published in "*IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*" copyright [2015] IEEE. See Appendix C.1 for the copyright permission from IEEE.

[2]The material contained in this chapter was accepted to "*IEEE Transactions on Emerging Topics in Computing (TETC)*." See Appendix C.2 for the copyright permission from IEEE.

[3]The material contained in this chapter is submitted to a journal publication.

# List of Figures

# List of Tables

# Preface

This dissertation presents my research work in pursuing the Ph.D. degree in Computer Engineering at Michigan Technological University. This dissertation includes a previously published paper in Chapter 2, and an article which has been accepted in Chapter 3. This dissertation includes an article which is under preparation in Chapter 4.

Chapter 2 contains a paper published in *IEEE Computer Society Annual Symposium on VLSI*. As the first author, with the guidance of my advisor Dr. Shiyan Hu, I proposed the algorithm of a timing driven buffer insertion technique for carbon nanotube interconnects. The second author Yuchen Zhou and I implemented the algorithm using Python and C programming. I analysed the experimental results and the paper was written by me and my advisor.

Chapter 3 contains an article accepted to *IEEE Transactions on Emerging Topics in Computing*. As the first author, I analyzed the models of the resistance and capacitance of the bundled SWCNT interconnects considering unidimensional spatial correlation. The stochastic SWCNT interconnects buffering algorithm is proposed by my and my advisor Dr. Shiyan Hu. Yuchen Zhou, the second author, implemented the timing evaluation using Python and I implemented the stochastic SWCNT algorithm.

The article was written by me and my advisor.

Chapter 4 contains an article which is submitted to a journal publication. As the first author, I proposed a novel carbon nanotube physical unclonable function design through leveraging Lorenz chaotic system, which is resistant to machine learning modeling attacks. I implemented the simulation of the proposed design and generated the training and test data. I also implemented experiments of various machine learning attacks to evaluate the performance of the proposed physical unclonable function. The article was written by me and my advisor.

# Acknowledgments

In the journey of my PhD study, I have so many people to thank for. First of all, I would like to thank my advisor Dr. Shiyan Hu. We have been working together since 2011. He is a passionate researcher. His guidance and support have made me better and better on the research study. He always encourages me to propose and try novel ideas. It has been a great pleasure to work with Dr. Shiyan Hu. Without his advice and enthusiasm, it is impossible for me to finish my PhD study.

I would like to thank all my committee members, Dr. Shiyan Hu, Dr. Zhaohui Wang, Dr. Nilufer Onder and Dr. Ye Sun for taking the time to review and critique my dissertation and provide me with very valuable feedback.

I would like to thank my research group members, Yang Liu and Yuchen Zhou for their help and support. I would like also thank to my friend Jinxiang Liu for spending time helping me learn new programming language. Thank all my friends for their accompanying during those days.

Last but not least, a lot of appreciation to my parents for their support during these years. Thank you.

# Abstract

As prevailing copper interconnect technology advances to its fundamental physical limit, interconnect delay due to ever-increasing wire resistivity has greatly limited the circuit miniaturization. Carbon nanotube (CNT) interconnects have emerged as promising replacement materials for copper interconnects due to their superior conductivity. Buffer insertion for CNT interconnects is capable of improving circuit timing of signal nets with limited buffer deployment. However, due to the imperfection of fabricating long straight CNT, there exist significant unidimensional-spatially correlated variations on the critical CNT geometric parameters such as the diameter and density, which will affect the circuit performance. This dissertation develops a novel timing driven buffer insertion technique considering unidimensional correlations of variations of CNT. Although the fabrication variations of CNTs are not desired for the circuit designs targeting performance optimization and reliability, these inherent imperfections make them natural candidates for building highly secure physical unclonable function (PUF), which is an advanced hardware security technology. A novel CNT PUF design through leveraging Lorenz chaotic system is developed and we show that it is resistant to many machine learning modeling attacks. In summary, the studies in this dissertation demonstrate that CNT technology is highly promising for performance and security optimizations in advanced VLSI circuit design.

# Chapter 1

# Introduction

The heterogeneous system architecture which leverages multicore computing paradigm has become increasingly popular. Nevertheless, timing minimization is still a critical design challenge. Buffer insertion for carbon nanotube (CNT) interconnects is capable of significantly improving circuit timing of signal nets with limited buffer deployment. A timing driven buffer insertion technique for CNT interconnects is proposed, where the standard buffering algorithm has been enhanced to accommodate some features in the CNT timing modelling.

However, due to the imperfection of fabricating long straight CNT, there exist significant variations on the critical CNT geometric parameters such as the diameter and density, which will affect the circuit performance. On the other hand, the prevailing

CNT fabrication induces significant unidimensional spatial correlation. A unidimensional variation aware stochastic CNT interconnects buffering algorithm is developed to handle fabrication variations of CNTs in buffer insertion. To improve its time complexity, a novel importance sampling based timing evaluation technique is proposed considering unidimensional correlations of variations.

Although the fabrication variations of CNT are not desired for the circuits targeting performance and reliability, these inherent imperfections make the CNT based circuits natural candidates for building highly secure physical unclonable function (PUF). PUF is an advanced hardware security technology. Most conventional encryption approaches rely on the secure keys stored in flash or non-volatile memory, and they are vulnerable to physical attacks. PUFs exploit the hardware fabrication variations to generate the secure key, thus it is resistant to physical attacks.

CNT PUF designs have many advantages, such as low cost and significant randomness. However, they are still vulnerable to machine learning modeling attacks. Using the machine learning modeling attack, it is not necessary for the attacker to access the PUF layout and hardware. The attacker collects a large amount of challenge-response pairs, as the training data. Machine learning modeling attack methods are then used to predict the model of the PUF. Subsequently, the attacker can use the model to predict the responses given on the new challenges. It is demonstrated that machine learning modeling attacks can have high prediction rate, e.g., 99.9%. In

this dissertation, a novel CNT PUF design through leveraging Lorenz chaotic system is proposed. The Lorenz chaotic system could magnify the differences among corresponding responses of similar challenges, which makes the proposed PUF design resistant to machine learning modeling attacks. Through the study in the dissertation, we demonstrate that CNT technique is highly promising for performance and security optimization in advanced VLSI circuit design.

## 1.1 Buffering CNT Interconnects for Timing Optimization

As one of the most effective interconnect timing optimization techniques, copper buffer insertion is indispensable in physical design [6, 7, 8, 9, 10]. However, since copper interconnect technology has its fundamental physical limit, interconnect delay due to ever increasing wire resistivity has greatly limited the circuit miniaturization. The electromigration induced interconnect reliability issue resulting from the inherently low tolerable current density in copper interconnect aggravates the problem. Consequently, the novel on-chip interconnect material is highly desirable as a replacement of copper interconnect in nanoscale high-speed circuit design. As promising replacement materials, carbon nanotubes (CNTs) alleviate the above severe timing and reliability issues in copper interconnect due to their superior conductivity and current carrying

capabilities. CNTs have significantly larger carrier mean free paths and can conduct larger currents without deterioration compared to copper interconnects [11]. As a result, the issues such as electromigration that plaque the copper interconnects are mitigated. In addition, CNTs have high thermal conductivity and mechanical stability.

CNTs are miniaturized tubes consisting of rolled up sheets of carbon hexagons. There are two main types of CNTs with structural perfection. Single-walled carbon nanotubes (SWCNTs) are composed of a single graphite sheet seamlessly wrapped into a cylindrical tube while multi-walled carbon nanotubes (MWCNTs) are composed of an array of concentrically nested CNTs. Since a single CNT has much larger resistance than copper for global interconnect [2], it is desired to bundle CNTs in parallel, resulting in *bundled CNTs*, for better performance. According to [2], it is difficult to use MWCNTs for long-length ballistic transport, thus, this work will focus on the popular bundled SWCNTs. Various research efforts have been spent in CNT fabrication. Most of them explores chemical vapor deposition technologies and successful fabrication experience on CNT includes [12, 13, 14, 15, 16, 17].

It has been shown that bundled SWCNTs can outperform copper interconnects in signal wave transportation along a long global interconnect [2, 18, 19, 20, 21]. For example, it is shown in [2] that the resistance of bundled SWCNTs can be achieved 50% smaller compared to that of copper at the same size of a long interconnect at

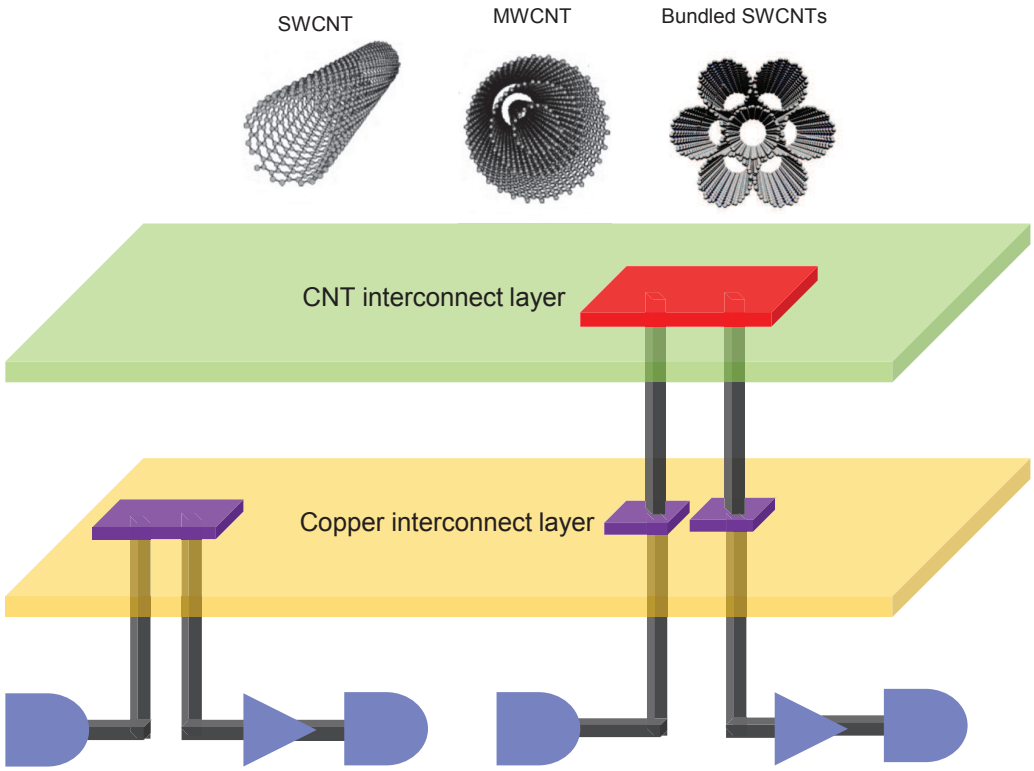



**Figure 1.1:** Copper buffering and CNT buffering.

$22nm$ technology node. Despite this, buffer insertion is still necessary to improve the timing of a bundled SWCNTS. Although there are works [2, 18] which consider CNT interconnect, they always use a two pin model since their perspective is from the device and interconnect modeling of CNTs. None of existing works consider the deployment of such an advanced technology into the VLSI physical design. To the best of the authors' knowledge, this work presents the first physical design technique considering carbon nanotube interconnects. Refer to Figure 2.1. The CNTs are replacing copper in global interconnect. In this work, bundled SWCNTs are mainly considered. The main contribution of this work is summarized as follows.

† The timing driven buffer insertion technique for bundled SWCNTs is proposed which can handle signal net buffering in VLSI design. To the best of authors' knowledge, this is the first buffer insertion optimization for CNT interconnects in the literature.

† Timing driven buffering algorithm for copper interconnect has been adapted to handle bundled SWCNTs.

† Our experiments are conducted with 500 scaled industrial nets and 10 different types of scaled buffers and inverters at 22nm technology. With the same timing constraint, CNT buffering can save over 50% buffer area compared to copper buffering. In addition, it is demonstrated that CNT buffering can effectively reduce the delay by up to 32%.

## 1.2 Stochastic Buffering For Bundled SWCNT Interconnects Considering Unidimensional Fabrication Variation

The heterogeneous system architecture which leverages multicore computing paradigm has become increasingly popular and it has been successfully deployed in many application scenarios. Nevertheless, there are still various design challenges

which need to be tackled, and timing minimization is a critical one among them. It is well known that interconnect delay has become the bottleneck of the circuit timing. However, existing copper based interconnects technologies are approaching their fundamental physical limits. Thus, novel interconnect materials such as carbon nanotube (CNT) become highly desirable. There are two types of CNTs regarding the electrical properties which are metallic CNTs (mCNTs) and semiconducting CNTs (sCNTs). Bundled metallic single walled carbon nanotubes (SWCNTs) have better electrical performance than copper in terms of e.g., superior conductivity and current carrying capabilities. Therefore, SWCNTs are suggested to be used as long global interconnects [2, 3, 11, 12]. However, due to the imperfectness of fabrication, the variations in the geometric parameters of CNTs will lead to significant timing yield reduction of the design [22].

Chemical vapor deposition (CVD) is the most popular technique for CNT fabrication [13, 14, 15, 16, 17]. In such a technique, CNTs grow along a single direction and their geometric parameters are controlled by the environment of the CVD system. For example, the diameters of tubes are highly dependent on the processing environment and their orientations are controlled by the direction of gas flow. CNT density variations are caused by non-uniform spacing between CNTs. Therefore, the number of CNTs in the bundled SWCNT interconnects may have a large variance. There are other variations from the imperfect fabrication process, such as the growth kinetics,

flow patterns variations, nucleation sites variations, and adsorption and diffusion variations, which can be translated to the variations on the resistance and capacitance of the bundled SWCNT interconnects. These variations could be partially captured by the density and diameter variations.

When using CNTs as interconnects, the main sources of variability include the diameters of CNT, the density of mCNTs in the bundle, the misalignment of CNTs and the presence of sCNTs [22, 23, 24, 25]. In this work, to illustrate our technique we focus on the diameter and density variations, while our approach can be easily adapted to handle other variations. The density and diameter variations will result in timing variations on the bundled SWCNT interconnects, impacting the timing yield of the CNT based circuit design.

In the literature, there are some works addressing the variations to carbon nanotube field effect transistor (CNFET) based circuit designs[26, 27]. The models for the impact of mCNTs on the CNFET and circuit delay are provided in [28, 29] and the impact of undeposited CNTs to the circuit delay is studied in [30, 31]. Shahi and Zarkesh-Ha propose an analytical model to predict gate delay variation induced from CNT density variation [32]. Zhang et al. consider spatial correlation in directional CNT growth which helps reduce the device-level failure by 350× [33]. A timing-driven placement method has been recently developed for CNFET circuits in [34].

Variations also manifest strongly in CNT interconnects. Refer to Figure 1.2. There

are some works modelling the fabrication impacts. For example, Nieuwoudt and Massoud model the variational impact on resistance, capacitance, inductance, and delay of bundled SWCNT interconnects [21]. However, it does not provide any closed form computation model for resistance and capacitance of the bundled SWCNT interconnects, and thus it cannot be extended to handle spatial correlation for our purpose. In [35], a statistical model is proposed to analyze the crosstalk noise induced by process variations on SWCNT based on a closed-form derivation. However, it focuses on noise analysis which cannot be adapted to modelling timing and its does not consider unidimensional spatial correlation as well. In this work, we will present a new closed-form model for the resistance and capacitance of bundled SWCNT interconnects, which is integrated into our buffer insertion algorithm for timing evaluations.

A striking difference compared to the copper counterpart is that there exists spatial correlation in bundled SWCNT interconnect fabrication [22, 25, 36]. Since CNTs grow along one dimension in the fabrication, the spatial correlation in variations in CNTs is in one dimension, in contrast to the two dimensional spatial correlation in the copper based design. Such a property will be leveraged in developing our interconnect optimization technique for CNT based circuits.

In the literature, there are very limited works studying the carbon nanotubes based physical design. The first CNT buffer algorithm is developed in [3]. However, that work does not consider the fabrication variations which could significantly impact

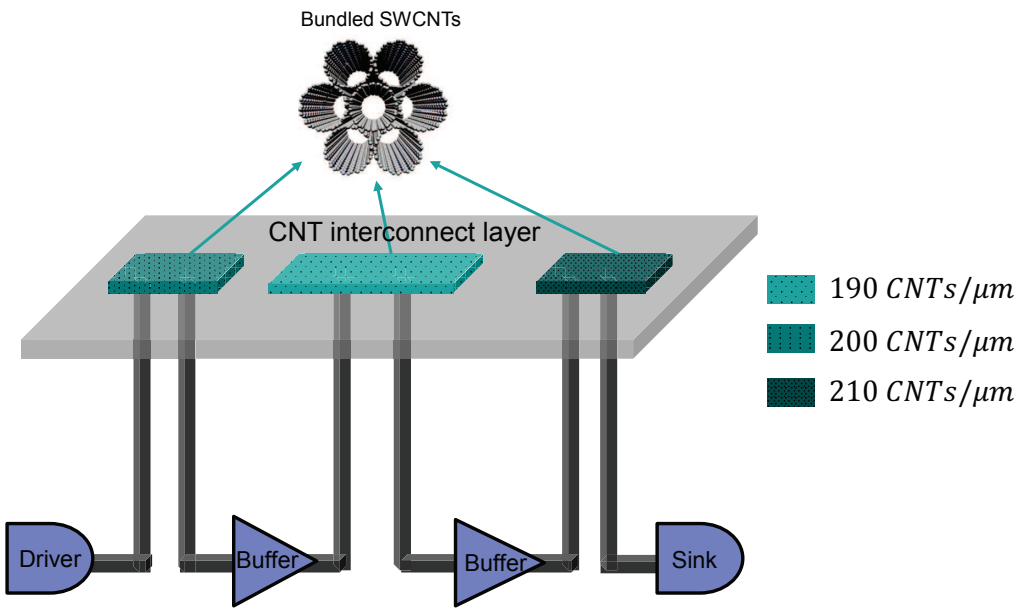



**Figure 1.2:** Illustration of fabrication variation aware buffer insertion problem for bundled SWCNT interconnects.

the performance of circuit design. This motivates us to model the variations, esp. the unidimensional spatial correlation of fabrication variations, on resistance and capacitance of bundled SWCNT interconnects, and develop a new stochastic CNT buffering algorithm based on this model. The contribution of this work is summarized as follows.

† Fabrication variation and unidimensional spatial correlation on the resistance and capacitance of bundled SWCNT interconnects are analyzed.

† A unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm is proposed. A new importance sampling based timing evaluation technique is also developed to improve the computational efficiency of the algorithm.

† To the best of our knowledge, this is the first work on the SWCNT interconnect optimizations considering variations.

† The simulation results on signal nets demonstrate that the proposed unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm saves over 30% buffer area over copper buffering on 50 nets while satisfying timing constraints. In addition, our proposed stochastic SWCNT interconnects buffering algorithm achieves much better performance than the best case design and the worst case design in terms of timing and buffer cost.

## 1.3 Lorenz Chaotic System Based CNT PUF

Physical unclonable function (PUF) is an emerging technology for security applications, such as true random number generation, secure key generation, low-cost authentication, etc [37, 38]. Most conventional encryption methodologies rely on the secure keys stored in flash or non-volatile memory, and they are vulnerable to physical attacks. As an alternative approach, PUF exploits the hardware fabrication variations and generates unpredictable secure information in a storage-less fashion. For example, Gassend et al. propose an arbiter PUF through leveraging the timing difference on the two identically designed paths due to fabrication variations [39]. The

input signal of the PUF is called the challenge and the output signal is called the response.

Given a PUF design, we can have many fabricated chips. The only differences among each fabricated chip are from the fabrication variations, which is not predictable nor clonable. The carbon nanotubes are promising candidates for highly secure PUF design thanks to their significant fabrication variations [23]. Chemical vapor deposition (CVD) is the most popular method for CNT fabrication, in which the pressure and temperature of the environment have significant effects on the fabricated features such as diameters and densities of CNTs. When CNTs are used as FETs and interconnects for reliable designs, the fabrication variations are not desired [21, 25, 40]. However, these inherent imperfections make the CNT based circuits natural candidates for building highly secure PUFs. Several carbon nanotube PUFs (CNT PUFs) are designed in the previous works such as [1, 41, 42, 43], which demonstrate significant advantages such as low cost and significant randomness. A secure empirical mode decomposition projection based CNT PUF design is developed in [41]. The other prominent one is discussed in [1] where self-assembled CNTs are used to design a random bit generation approach for low-cost and hard-to-forge security applications. In Figure 4.1, individual CNTs are placed between two layers which can be randomly connected or disconnected [1].

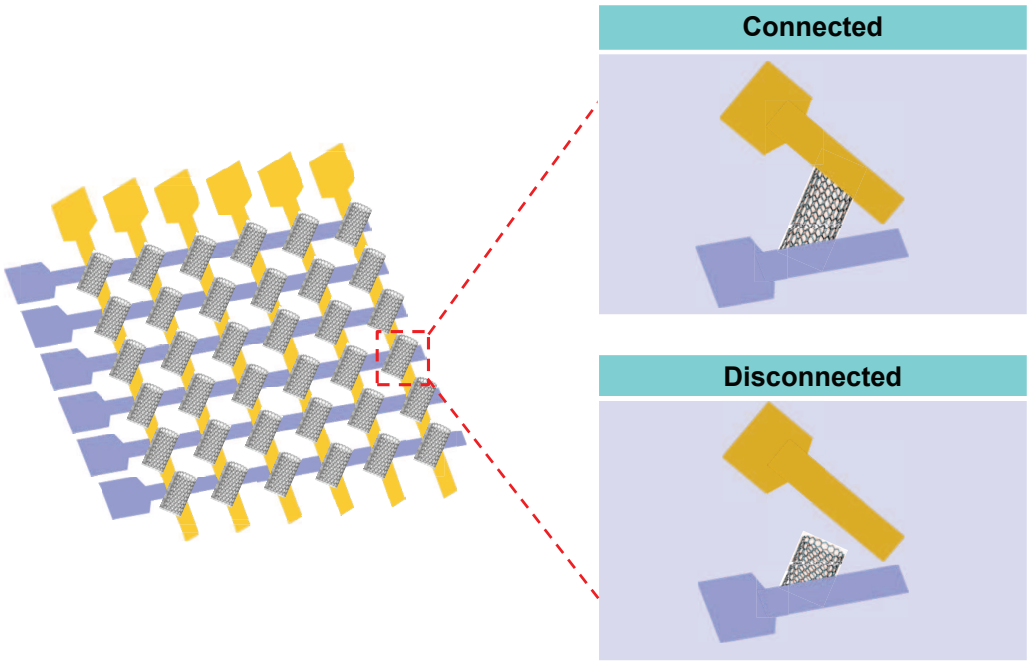



**Figure 1.3:** The illustration of 2D CNT bitarray crossbar structure (adapted from [1]).

Although CNT PUF designs have many advantages, they are still vulnerable to machine learning modeling attacks, where the attacker does not need to access the PUF hardware physsically. The attacker collects a large amount of challenge-response pairs as the training data. Machine learning modeling attack methods are then used to model the PUF. Subsequently, the attacker can use the model to predict the responses given new challenges. Deep learning (DL) and evolution strategy (ES) are the most prominent machine learning methods used for modeling attacks [44]. It is demonstrated in some works [45, 46, 47] that machine learning modeling attacks can achieve high prediction rate, e.g., 99.9%.

This motivates [48, 49, 50, 51] to design PUFs resistant to modeling attacks. In [48],

a secure physically-embedded data encryption architecture is proposed by replacing conventional weak arbiter PUF with a specific strong PUF proposed in [49]. However, it is not easy to build that specific strong PUF proposed in [49]. A circuit that relies on non-linear current mirrors is designed to generate modeling resistant PUF in [50]. The current sources are assumed to be ideal which is impractical. In [51], the authors propose a lockdown technique in the PUF based system by adding server authentication. It could effectively prevent the attacker to collect many challenge-response pairs. However, the lockdown technique is of low efficiency.

In this work, our objective is to design a CNT PUF which is resistant to machine learning modeling attacks. First, one needs to know how the modeling attack methods work. For most PUFs, similar challenges could generate similar responses [52, 53, 54]. Therefore, one possible method for preventing modeling attacks is to magnify the differences among responses of similar challenges. Since Lorenz chaotic system yields widely diverging outputs given similar inputs, it motivates us to develop a novel CNT PUF design by leveraging Lorenz chaotic system. To the best of our knowledge, this is the first such work in CNT PUF design. To demonstrate the effectiveness of our proposed Lorenz chaotic system based CNT PUF, various machine learning attacks are preformed, including Support Vector Machine (SVM), Deep Belief Networks (DBN), Logistic Regression (LR) and Evolution Strategies (ES). The experimental results demonstrate that the proposed Lorenz chaotic system based CNT PUF is robust to these attacks. The main contribution of this work is summarized as follows.

- In this work, a novel CNT PUF design is developed by leveraging Lorenz chaotic system. Lorenz chaotic system magnifies the differences among responses of similar challenges, which makes the proposed PUF design resistant to modeling attacks.

- To demonstrate the security performance of the proposed PUF, various machine learning methods are used on the proposed PUF, including SVM, DBN, LR and ES.

- The experimental results demonstrate that the machine learning modeling attack methods can achieve as high as 100% bit-wise prediction rates on the CNT PUF without Lorenz chaotic system, while they can only obtain less than 55% bit-wise prediction rates on the proposed Lorenz chaotic system based CNT PUF. To the best of our knowledge, this is the first work to leverage Lorenz chaotic system to CNT PUF.

- The significant security performance of the proposed PUF is mainly contributed by Lorenz chaotic system. However, if one uses Lorenz chaotic system only in the design, the parameters need to be induced by fabrication variations, which could be complicated. Therefore, one needs to combine CNT PUF and Lorenz chaotic system as discussed in this work.

# Chapter 2

# Buffering Single-Walled Carbon Nanotubes Bundle Interconnects for Timing Optimization[1]

## 2.1 Introduction

As one of the most effective interconnect timing optimization techniques, copper buffer insertion is indispensable in physical design [6, 7, 8, 9, 10]. However, since copper interconnect technology has its fundamental physical limit, interconnect delay due

[1]The material contained in this chapter was previously published in "*IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*" copyright [2015] IEEE. See Appendix C.1 for the copyright permission from IEEE.

to ever increasing wire resistivity has greatly limited the circuit miniaturization. The electromigration induced interconnect reliability issue resulting from the inherently low tolerable current density in copper interconnect aggravates the problem. Consequently, the novel on-chip interconnect material is highly desirable as a replacement of copper interconnect in nanoscale high-speed circuit design. As promising replacement materials, carbon nanotubes (CNTs) alleviate the above severe timing and reliability issues in copper interconnect due to their superior conductivity and current carrying capabilities. CNTs have significantly larger carrier mean free paths and can conduct larger currents without deterioration compared to copper interconnects [11]. As a result, the issues such as electromigration that plaque the copper interconnects are mitigated. In addition, CNTs have high thermal conductivity and mechanical stability.

CNTs are miniaturized tubes consisting of rolled up sheets of carbon hexagons. There are two main types of CNTs with structural perfection. Single-walled carbon nanotubes (SWCNTs) are composed of a single graphite sheet seamlessly wrapped into a cylindrical tube while multi-walled carbon nanotubes (MWCNTs) are composed of an array of concentrically nested CNTs. Since a single CNT has much larger resistance than copper for global interconnect [2], it is desired to bundle CNTs in parallel, resulting in *bundled CNTs*, for better performance. According to [2], it is difficult to use MWCNTs for long-length ballistic transport, thus, this work will focus on the

popular bundled SWCNTs. Various research efforts have been spent in CNT fabrication. Most of them explores chemical vapor deposition technologies and successful fabrication experience on CNT includes [12, 13, 14, 15, 16, 17].

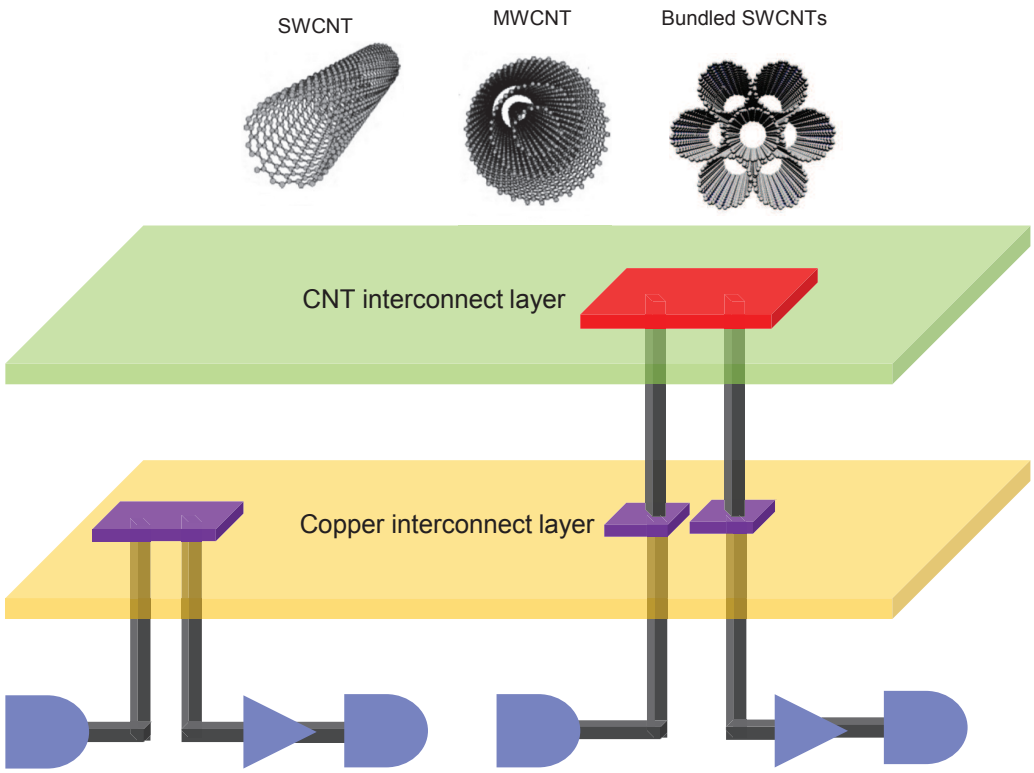



**Figure 2.1:** Copper buffering and CNT buffering.

It has been shown that bundled SWCNTs can outperform copper interconnects in signal wave transportation along a long global interconnect [2, 18, 19, 20, 21]. For example, it is shown in [2] that the resistance of bundled SWCNTs can be achieved 50% smaller compared to that of copper at the same size of a long interconnect at $22nm$ technology node. Despite this, buffer insertion is still necessary to improve the timing of a bundled SWCNTS. Although there are works [2, 18] which consider

CNT interconnect, they always use a two pin model since their perspective is from the device and interconnect modeling of CNTs. None of existing works consider the deployment of such an advanced technology into the VLSI physical design. To the best of the authors' knowledge, this work presents the first physical design technique considering carbon nanotube interconnects. Refer to Figure 2.1. The CNTs are replacing copper in global interconnect. In this work, bundled SWCNTs are mainly considered. The main contribution of this work is summarized as follows.

† The timing driven buffer insertion technique for bundled SWCNTs is proposed which can handle signal net buffering in VLSI design. To the best of authors' knowledge, this is the first buffer insertion optimization for CNT interconnects in the literature.

† Timing driven buffering algorithm for copper interconnect has been adapted to handle bundled SWCNTs.

† Our experiments are conducted with 500 scaled industrial nets and 10 different types of scaled buffers and inverters at 22nm technology. With the same timing constraint, CNT buffering can save over 50% buffer area compared to copper buffering. In addition, it is demonstrated that CNT buffering can effectively reduce the delay by up to 32%.

## 2.2 Problem Formulation

Consider a routing tree $T = (V, E)$ where $V = s_0 \cup V_s \cup V_n$, and $E \in V \times V$. Let $|V| = n$. Vertex $s_0$ is the source node and also called the root of the tree. $V_s$ is the set of sink nodes. Each sink, denoted by $s$, has a sink capacitance and required arrival time $RAT(s)$. $T$ is said to satisfy the timing constraint if its required arrival time at root is no earlier than the arrival time at root. Each edge, denoted by $e$, in $E$ represents a segment of wire, which has edge resistance $R(e)$, edge inductance $L(e)$ and edge capacitance $C(e)$. $V_n$ refers to the candidate buffer positions where the buffers can be inserted. In practice, they are discrete locations and are specified before buffer insertion algorithm by e.g., wire segmenting technique [55].

A buffer library $B$ which consists of a set of different types of buffers are given to the buffering problem. Let $|B| = m$. Each buffer, denoted by $b$, has its cost $W(b)$, input capacitance $C(b)$, driving resistance $R(b)$ and intrinsic delay $t(b)$. Following most existing buffering works [6, 7, 8, 9, 10], the underlying routing tree can be assumed to be binary since trees in other topologies can be converted to a binary one using the technique in [8]. Given a tree in carbon nanotube interconnect layer, a buffer assignment is to determine the locations and the types of buffers which will be inserted to the routing tree. Our buffer insertion problem is formulated as follows.

**Timing Constrained Minimum Cost Buffering for Carbon Nanotube Interconnects**: Given a binary routing tree with $n$ candidate buffer locations in carbon nanotube interconnect layer, a buffer library and a set of candidate buffer positions, to compute a buffer assignment solution such that the timing constraint is satisfied, and the total buffer cost is minimized.

## 2.3 Carbon Nanotube Interconnects

To tackle the fundamental physical limits on copper interconnects, CNTs have emerged as a promising replacements for Copper interconnects due to their better conductivity and current carrying capabilities. Table 2.1 from [4, 5] summarizes some major advantages of CNTs over copper interconnects. In fact, similar observations have been made from many other works [56, 57, 58, 59, 60].

**Table 2.1**
Comparison between CNT and copper interconnect [4, 5].

| Properties | CNT | Cu |
|---|---|---|
| Max. current density | $10^{10} A/cm^2$ | $10^6 A/cm^2$ |
| Mean free path | $1000nm$ | $40nm$ |
| Thermal conductivity | 6000 $W/mK$ | 400 $W/mK$ |

CNTs are miniaturized tubes consisting of rolled up sheets of carbon hexagons. Figure 2.2 shows an equivalent circuit model for an isolated single-walled carbon nanotube (SWCNT), which is proposed in [2]. It has become a popular model and it will

be explained how to compute the resistance and capacitance using this model.

### 2.3.1 Resistance for CNT

#### 2.3.1.1 Resistance for An Isolated SWCNT

The resistance of an isolated SWCNT, denoted by $R_{isolated}$, is divided into two parts, the quantum resistance $R_Q$ and scattering resistance $R_S$ as shown in Figure 2.2. Recall that the mean free path, denoted by $\lambda$, refers to the average distance between two subsequent collisions of electrons. The mean free path of electrons for a CNT is about $1\mu m$ as shown in Table 2.1, i.e., $\lambda = 1\mu m$. When $l \leq \lambda$ where $l$ is the length of a carbon nanotube, we have [61]

$$R_Q = \frac{h}{4e^2} = 6.45k\Omega, \tag{2.1}$$

where $e$ is the electronic charge and $h$ is Plank's constant. Thus, if the length $l$ of a CNT is less than $\lambda = 1\mu m$, the resistance of CNT is independent of length.

For the length greater than the mean free path, the distributed scattering resistance for an interconnect with length $l$ is [61, 62]:

$$R_S l = \frac{hl}{4e^2\lambda}. \tag{2.2}$$

For simplicity, one defines $R_S = 0$ when $l \leq \lambda$. In practice, the total resistance of a single CNT, denoted by $R_{isolated}$, is expressed as the sum of quantum resistance and scattering resistance as shown in the following equation [2]

$$R_{isolated} = R_Q + R_S l. \tag{2.3}$$

Comparing to copper global interconnect, a single SWCNT global interconnect has resistance of $6.45k\Omega/\mu m$, which is too large for timing minimization. However, if a bundled SWCNTs are used, the resistance can be significantly reduced.

#### 2.3.1.2 Resistance for a Bundled SWCNTs

The resistance of a bundle, denoted by $R_{bundle}$, is given by the following equation [62]:

$$R_{bundle} = R_{isolated}/N_{cnt}, \tag{2.4}$$

where $N_{cnt}$ is the number of CNTs contained in the bundle. It is clear that the

resistance decreases with increasing $N_{cnt}$.

#### 2.3.1.3 Contact Resistance

Due to the presence of imperfect metal and carbon nanotube contacts, contact resistance needs to be considered. According to [21], some research groups have accomplished to fabricate the contact resistances ranging from a few hundred ohms to a few kilohms which have similar magnitude with quantum resistance and scattering resistance.

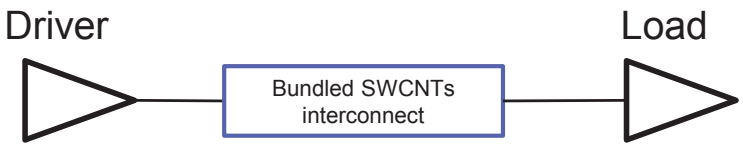
### 2.3.2 Capacitance for CNT

#### 2.3.2.1 Capacitance for An Isolated SWCNT

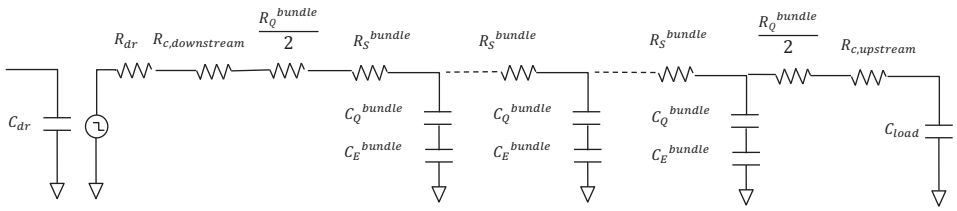
The capacitance of the CNT comes from two aspects. One is the electrostatic capacitance denoted by $C_E$, and the other is quantum capacitance denoted by $C_Q$.

The quantum capacitance $C_Q l$ is obtained by [63]:

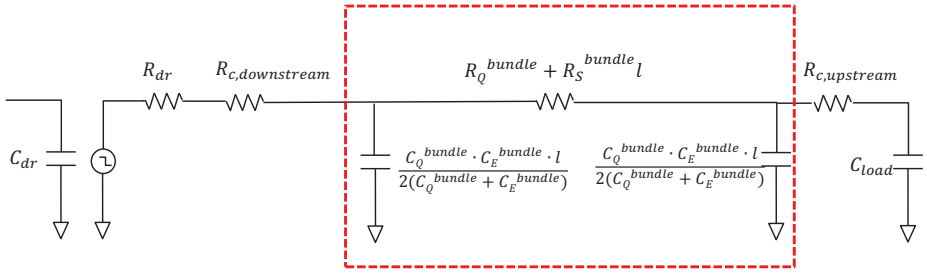
$$C_Q l = \frac{2e^2}{hv_f} l. \tag{2.5}$$



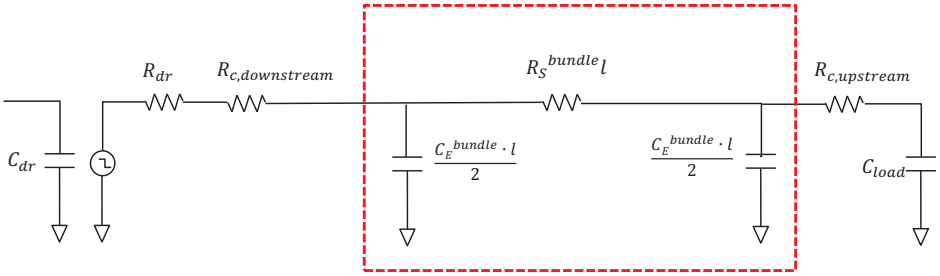

(a) Schematic of bundled SWCNTs interconnect




(b) Distributed equivalent circuit model for bundled SWCNTs interconnect




(c) Equivalent $\pi$ circuit model for bundled SWCNTs interconnect




(d) Simplified equivalent $\pi$ circuit model for bundled SWCNTs interconnect

**Figure 2.2:** Equivalent circuit model for bundled SWCNTs interconnect (adapted from [2]).

Since an SWCNT has four conducting channels, the net quantum capacitance of an isolated SWCNT is

$$C_Q^{CNT} l = 4C_Q l. \tag{2.6}$$

The quantum capacitance for a bundled SWCNT can be computed as

$$C_Q^{bundle} l = N_{cnt} C_Q^{CNT} l. \tag{2.7}$$

The electrostatic capacitance $C_E$ is calculated by treating the CNT as a thin wire, with diameter $d$ and the distance to the ground plane $y$. $C_E l$ can be calculated as follows

$$C_E l = \frac{2\pi\epsilon}{cosh^{-1}(y/d)} l, \tag{2.8}$$

where $\epsilon$ is the permittivity of free space. The electrostatic capacitance for a bundled SWCNTs $C_E^{bundle}$ is given by a parallel combination of all SWCNTs in the bundle. The electrostatic capacitance can be calculated using FastCap [64].

According to [2], besides quantum capacitance and electrostatic capacitance, capacitance between metallic and semiconducting SWCNTs within a bundle is not important. In addition, the effect of the quantum capacitance is small, the effective capacitance of an SWCNTs bundle is nearly equal to its electrostatic capacitance [2].

$$C_{bundle} l = C_E^{bundle} l. \tag{2.9}$$

### 2.3.3 Inductive Impact is Not Important

According to [2], the inductive impact is not important. It shows that an RC model for interconnect delay is accurate when the following inequality does not hold.

$$R_{dr}Cl < \frac{1}{2}RlCl < \sqrt{LC}l, \tag{2.10}$$

where $R_{dr}$ is the driver impedance and $R$, $C$ and $L$ are the per unit length interconnect resistance, capacitance and inductance. According to the simulation conducted in [2] for different size of driver and SWCNTs, Eq. 2.10 is never satisfied. Therefore, RC model is sufficient to handle bundled SWCNTs interconnect delay.

### 2.3.4 Elmore Delay Model for Bundled SWCNTs

This work uses the Elmore delay model for bundled SWCNTs proposed in [2]. Refer to Figure 2.2. The schematic of the driver, load and interconnect is shown in Figure 2.2(a). The interconnect is made of bundled SWCNTs. Elmore delay model for bundled SWCNTs with the driver and load capacitance is shown in Figure 2.2(c) which is derived from the distributed equivalent circuit model shown in Figure 2.2(b). $R_{dr}$ is the resistance of the driver and $C_{load}$ is the load capacitance connecting to the interconnect. $R_{c,downstream}$ is the contact resistance between the driver and the bundled

SWCNTs interconnect and $R_{c,upstream}$ is the contact resistance between the bundled SWCNTs interconnect and load capacitance. $R_Q^{bundle}$ and $R_S^{bundle}$ are the quantum and scattering resistance of bundled SWCNTs, respectively. $C_Q^{bundle}$ and $C_E^{bundle}$ are the quantum and electrostatic capacitance of bundled SWCNTs, respectively. Since the capacitance of bundled SWCNTs is approximately equal to the quantum capacitance of the bundled SWCNTS and quantum resistance is not important for long global interconnect, the $\pi$ model can be simplified to Figure 2.2(d).

## 2.4 Timing Buffering For Carbon Nanotube Interconnects

Our algorithm for carbon nanotube interconnect timing driven buffer insertion problem is based on the dynamic programming algorithm in [7]. In the algorithm, a 3-tuple $(Q, C, W)$ is used to characterize each buffering solution. $Q$ represents the required arrival time for each buffering solution, $C$ represents the downstream capacitance for each buffering solution, and $W$ is the cumulative buffer cost of the buffering solution. Working under the dynamic programming framework [7], the tree is processed in a bottom-up fashion and a set of candidate buffering solutions and the corresponding 3-tuple are propagated from sinks to driver. Precisely, a routing tree is traversed by depth first search, and the calculation/propagation for $Q, C, W$ begins when a sink is

reached. The algorithm will compute $Q$, $C$ and $W$ from sinks up to driver.

Pruning is an important technique in buffer insertion technique due to its effectiveness in reducing the number of solutions. Following [7], for any two solutions denoted by $\gamma_1, \gamma_2$ at the same node, $\gamma_2$ is said to be inferior to $\gamma_1$ and is thus pruned if $Q(\gamma_1) \geq Q(\gamma_2)$, $C(\gamma_1) \leq C(\gamma_2)$ and $W(\gamma_1) \leq W(\gamma_2)$. In other word, one will compare two solutions with the same set of processed candidate buffer locations by their required arrival time, downstream capacitance and cumulative buffer cost.

When the solutions are propagated all the way up to the driver, one can obtain all the non-inferior solutions. The one with smallest $W$ satisfying timing constraint will be returned. During the dynamic programming, there are four operations, namely, add wire, add buffer, add driver and branch merge. They are described as follows.

### 2.4.1 Add Buffer

This operation is invoked when a buffer is to be inserted at a candidate buffer location $v$. In any buffering solution $\gamma$, after a buffer insertion, a new solution $\gamma'$ will be generated. The cost $W(\gamma')$ will be computed as $W(\gamma') = W(\gamma) + W(b)$ if the buffer $b$ is inserted. Refer to Figure 2.3. Recall that the buffer resistance is $R(b)$, buffer capacitance is $C(b)$, and buffer intrinsic delay is $t(b)$. To handle the contact resistance, recall that the contact resistance for the contact linking the buffer $b$ with

the downstream CNT wire is $R_{c,downstream}(b)$, and the contact resistance for the contact linking the upstream CNT wires with the buffer $b$ is $R_{c,upstream}(b)$. The required arrival time needs to be updated considering the buffer delay and capacitance need to be set to the input capacitance of the buffer. Sinks can be similarly handled. We have

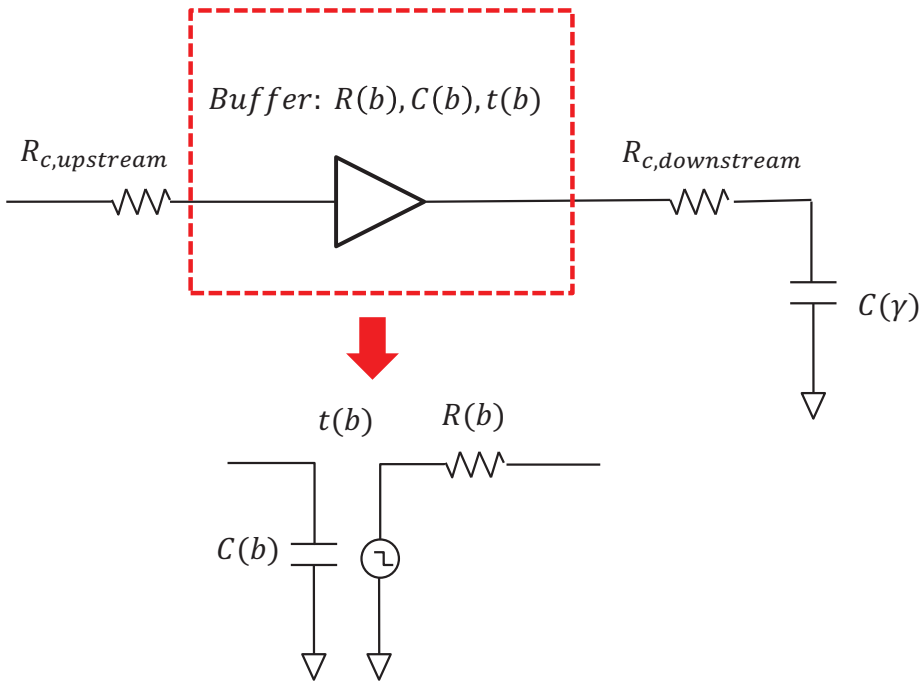



**Figure 2.3:** Circuit and parameters for add buffer.

$$
\begin{aligned}
Q(\gamma') &= Q(\gamma) - R(b) \cdot C(\gamma) - R_{c,downstream}(b) \\
&\quad \cdot C(\gamma) - R_{c,upstream}(b) \cdot C(b) - t(b) \\
C(\gamma') &= C(b) \\
W(\gamma') &= W(\gamma) + W(b).
\end{aligned}
\tag{2.11}
$$

### 2.4.2 Add Driver

This operation is to add the driver $b$ to the candidate buffering solution. It is similar to the add buffer operation with difference that one does not compute the delay due to the upstream contact resistance of the driver and one does not update the cumulative buffer cost.

$$\begin{aligned} Q(\gamma') &= Q(\gamma) - R(b) \cdot C(\gamma) \\ &\quad - R_{c,downstream}(b) \cdot C(\gamma) - t(b) \\ C(\gamma') &= C(b) \\ W(\gamma') &= W(\gamma). \end{aligned} \tag{2.12}$$

**Table 2.2**
Different types of inverter and buffer parameters at 22nm node. (Note that the inverters in BUF are different from those in INV)

| | BUF_X1 | BUF_X2 | BUF_X4 | BUF_X8 | BUF_X16 |
|---|---|---|---|---|---|
| Resistance ($\Omega$) | 2310.0 | 1201.0 | 618.9 | 315.5 | 159.6 |
| Capacitance($fF$) | 0.21 | 0.44 | 0.88 | 1.76 | 3.51 |
| Intrinsic delay ($ps$) | 2.93 | 2.91 | 2.87 | 2.87 | 2.87 |
| Area ($nm^2$) | 15197.6 | 30395.2 | 60790.4 | 121580.8 | 243161.6 |
| | INV_X1 | INV_X2 | INV_X4 | INV_X8 | INV_X16 |
| Resistance ($\Omega$) | 1846.0 | 976.5 | 514.8 | 270.2 | 139.7 |
| Capacitance($fF$) | 0.44 | 0.87 | 1.74 | 3.49 | 6.97 |
| Intrinsic delay ($ps$) | 0.59 | 0.62 | 0.61 | 0.61 | 0.61 |
| Area ($nm^2$) | 10115.6 | 20231.2 | 40462.4 | 80924.8 | 161849.6 |

### 2.4.3 Add Wire

Since the resistance of bundled SWCNTs global interconnect is related to the length, it can simply assumed that the distance between two consecutive buffers is larger than $1\mu m$. Under this assumption, the resistance of bundled SWCNTs can be simply $\frac{6.45k\Omega}{N_{cnt}}/\mu m$. In this operation, one is to add a wire from location $v$ to its upstream location $u$ for a candidate buffering solution. Recall that the capacitance for the wire $(u, v)$ is computed as $C(u, v) = C_E^{bundle} \cdot l(u, v)$ and the resistance for the wire $(u, v)$ is computed as $R(u, v) = R_{bundle} = R_S l(u, v)/N_{cnt}$, where $l(u, v)$ is the length of wire $(u, v)$. We have

$$\begin{aligned} Q(\gamma_u) &= Q(\gamma_v) - R(u, v) \cdot [\frac{C(u, v)}{2} + C(\gamma_v)] \\ C(\gamma_u) &= C(\gamma_v) + C(u, v) \\ W(\gamma_u) &= W(\gamma_v). \end{aligned} \tag{2.13}$$

### 2.4.4 Branch Merge

This operation is to merge the solutions in two branches connected by a branching point. Since the solutions along each branch have been computed, one will compute the combinations among them. Suppose that there are a solution

$(Q(\gamma_1), C(\gamma_1), W(\gamma_1))$ at left branch and a solution $(Q(\gamma_2), C(\gamma_2), W(\gamma_2))$ at right branch. After merging, we have

$$\begin{aligned} Q(\gamma) &= min\{Q(\gamma_1), Q(\gamma_2)\} \\ C(\gamma) &= C(\gamma_1) + C(\gamma_2) \\ W(\gamma) &= W(\gamma_1) + W(\gamma_2). \end{aligned} \tag{2.14}$$

That is, one needs to set the merged required arrival time to be smaller required arrival time on two branches, the total downstream capacitance to be the sum on the downstream capacitance on two branches, and the total buffer cost to be the sum of buffer costs on two branches.

## 2.5 Experimental Results

### 2.5.1 Experimental Setup

The proposed carbon nanotube interconnect based timing driven minimum cost buffer insertion algorithm is implemented in C language and tested on a machine with 3.40GHz Intel Pentium CPU and 3GB memory. The results of CNT buffering are compared with copper buffering. In this work, the buffer cost is measured by buffer

area.

Our buffer library consists of 10 buffer types including 5 buffers and 5 inverters. Due to the lack of industrial buffer library at 22nm technology, a buffer library of 45nm technology [65] is scaled to 22nm technology. To calculate the resistance, capacitance and intrinsic delay of different types of buffers and inverters at 22nm node, the simulation is performed using ngspice [66]. The resistance, capacitance, intrinsic delay and gate area are shown in Table 2.2. Linear fitting is applied to obtain resistance and intrinsic delay. The capacitance of buffer is simulated using method in [67].

**Table 2.3**
Unit resistance and capacitance (for $1\mu m$) of global interconnects with Cu and bundled SWCNTs at 22nm node.

| Properties | Cu | CNT |
|---|---|---|
| Unit resistance ($\Omega$) | 14.50 | 6.45 |
| Unit capacitance ($fF$) | 0.16 | 0.16 |

Our experiments are performed to 500 global nets extracted from an industrial ASIC chip in an old technology. Due to the lack of industrial nets in 22nm technology, we scale wire lengths of these old technology nets to 22nm technology.

The parameters of copper and bundled SWCNTs are presented in Table 3.1. The unit resistance and unit capacitance are for $1\mu m$. The parameters of copper are obtained from ITRS 2007 [68]. Note that the feature size predicted by ITRS 2007 is smaller than the one in the industrial 22nm technology according to [69]. We

**Table 2.4**
Timing constrained minimum cost buffering results on 5 representative nets

| Test cases | | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| CNT w/o contact resistance | Area ($nm^2$) | 318666.0 | 364162.0 | 222543.0 | 50578.0 | 40462.4 |
| | # Buffers | 7 | 5 | 5 | 3 | 2 |
| | Delay (ps) | 754 | 611 | 676 | 1019 | 722 |
| CNT w/ contact resistance (100Ω) | Area ($nm^2$) | 379359.0 | 424855.0 | 222543.0 | 80924.8 | 40462.4 |
| | # Buffers | 7 | 6 | 5 | 4 | 2 |
| | Delay (ps) | 762 | 599 | 691 | 927 | 736 |
| Cu | Area ($nm^2$) | 955997.0 | 819412.0 | 475433.0 | 202312.0 | 91040.4 |
| | # Buffers | 18 | 17 | 12 | 10 | 5 |
| | Delay (ps) | 766 | 611 | 702 | 994 | 870 |

use the ITRS parameters since the resistance and capacitance information of the industrial 22nm technology are not available. The parameters of bundled SWCNTs are calculated as follows. Refer to Figure 2.4. The cross section area of the global interconnect is set to be $33 \times 88nm^2$. For global interconnect, the resistance of a single SWCNT is approximately $6.45k\Omega/\mu m$ since the effect of quantum resistance for global interconnect is small. The impact of different number of SWCNTs in the bundle to the CNT resistance can be observed from Figure 2.4. If there are 1000 metallic SWCNTs in the $33 \times 88nm^2$ area, the total resistance of bundled SWCNTs is $6.45k\Omega/\mu m/1000 = 6.45\Omega/\mu m$. Note that the density of bundled SWCNTs is $1000/(33 \cdot 88) = 0.34nm^2$ which is below the maximum density $0.66nm^2$ from ITRS 2011 [70]. The unit capacitances of bundled SWCNTs and copper are set to be the same according to [2]. In this work, one considers both the ideal contact resistance and the practical contact resistance. The ideal contact resistance means no contact resistance. In the following discussion, without considering contact resistance is identical to ideal contact resistance. The practical contact resistance is set to 100Ω which

is achievable according to [21].

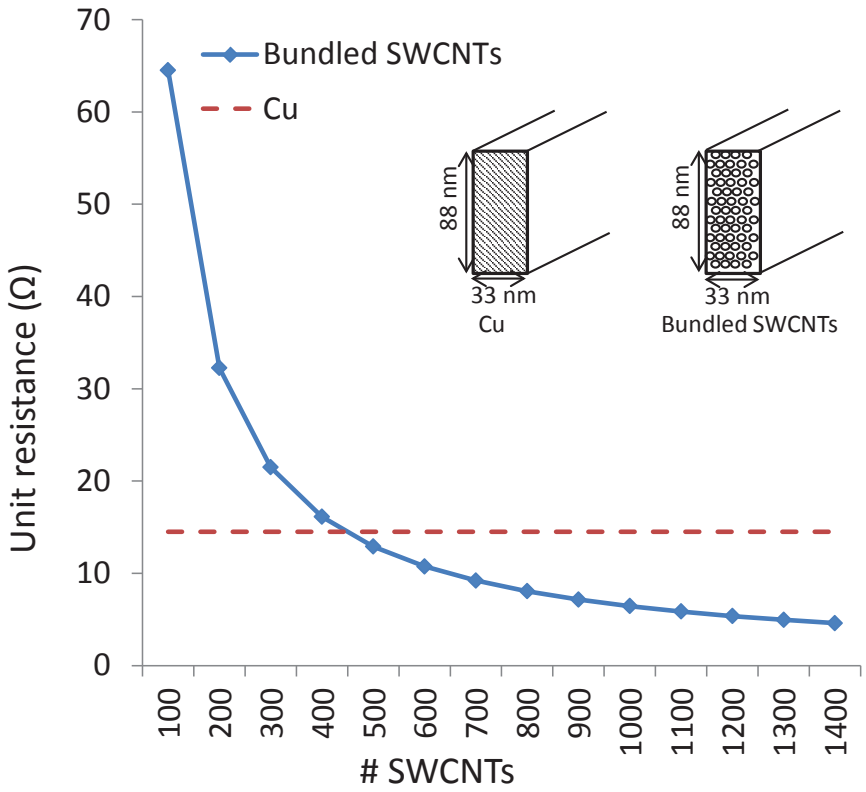



**Figure 2.4:** Resistance comparison and cross section area of Cu and bundled SWCNTs global interconnect in 22nm technology.

### 2.5.2 Experimental Results

**Table 2.5**
Average result for timing constrained minimum cost buffering on 500 nets

| Test cases | Area ($nm^2$) | Area ratio | # Buffers | Delay (ps) | # Solutions | CPU($s$) |
|---|---|---|---|---|---|---|
| CNT w/o contact | 107816.70 | 0.42 | 3.4 | 1125.8 | 2193.2 | 3.79 |
| CNT w/ contact | 105494.80 | 0.41 | 3.5 | 1127.9 | 1827.9 | 3.15 |
| Cu | 255110.10 | 1.00 | 7.7 | 1248.9 | 2250.0 | 3.54 |

**Table 2.6**
Timing minimization (without considering cost) on 5 nets

| Test cases | | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| CNT w/o contact resistance | Area ($nm^2$) | 3307950.0 | 2867260.0 | 2477160.0 | 3039520.0 | 1945290.0 |
| | # Buffers | 50 | 51 | 44 | 44 | 32 |
| | Delay (ps) | 376 | 216 | 314 | 249 | 188 |
| CNT w/ contact resistance (100Ω) | Area ($nm^2$) | 1463910.0 | 1468890.0 | 1408250.0 | 1458970.0 | 1094230.0 |
| | # Buffers | 36 | 31 | 31 | 24 | 18 |
| | Delay (ps) | 423 | 263 | 347 | 302 | 229 |
| Cu | Area ($nm^2$) | 2851920.0 | 2745490.0 | 2269040.0 | 2872350.0 | 2142860.0 |
| | # Buffers | 65 | 55 | 56 | 48 | 36 |
| | Delay (ps) | 479 | 317 | 382 | 363 | 276 |

Two sets of experiments are conducted which are timing constrained minimum cost buffering and timing minimization without cost minimization, respectively.

For timing constrained minimum cost buffering, the results on five representative nets are shown in Table 2.4 and the results on 500 nets are shown in Table 2.5. We make the following observations.

† One can see that in order to achieve the similar delay, the CNT buffering saves more than 50% buffer area over copper buffering. Averaging over 500 nets, CNT buffering without considering contact resistance saves 58% buffer area and CNT buffering with 100Ω contact resistance saves 59% buffer area. Take net 1 in Table 2.4 for an example, CNT buffering without considering contact resistance saves 67% buffer area and CNT buffering with 100Ω contact resistance saves 60% buffer area.

† The total number of buffers in CNT buffering is much (about 2×) smaller than that of copper buffering thanks to the fact that wire resistivity of bundled

SWCNTs is much lower than that of copper for global interconnect as shown in Table 3.1.

† One can see that the contact resistance does not have significant impact on the performance for CNT interconnect timing constrained minimum cost buffering.

† It would be interesting in investigating the delay-area tradeoff between copper buffering and CNT buffering. For this, net 3 in Table 2.4 is chosen to run the buffering algorithm while keeping all non-dominated solutions. One generates delay-area tradeoff curves for copper buffering and CNT buffering, respectively. Refer to Figure 2.5. It is clear that CNT buffering always outperforms the copper buffering in terms of timing and buffer area.

The above results are obtained through setting certain timing constraint and compute the minimum area solutions. One may be interested in the best achievable timing in both of CNT buffering and copper buffering. The results of five representative nets for buffering timing minimization without considering cost are shown in Table 2.6. It demonstrates that CNT buffering can reduce timing by up to 32% which is obtained from net 5. In addition, the contact resistance has some impact on the performance of CNT buffering such as area and timing.

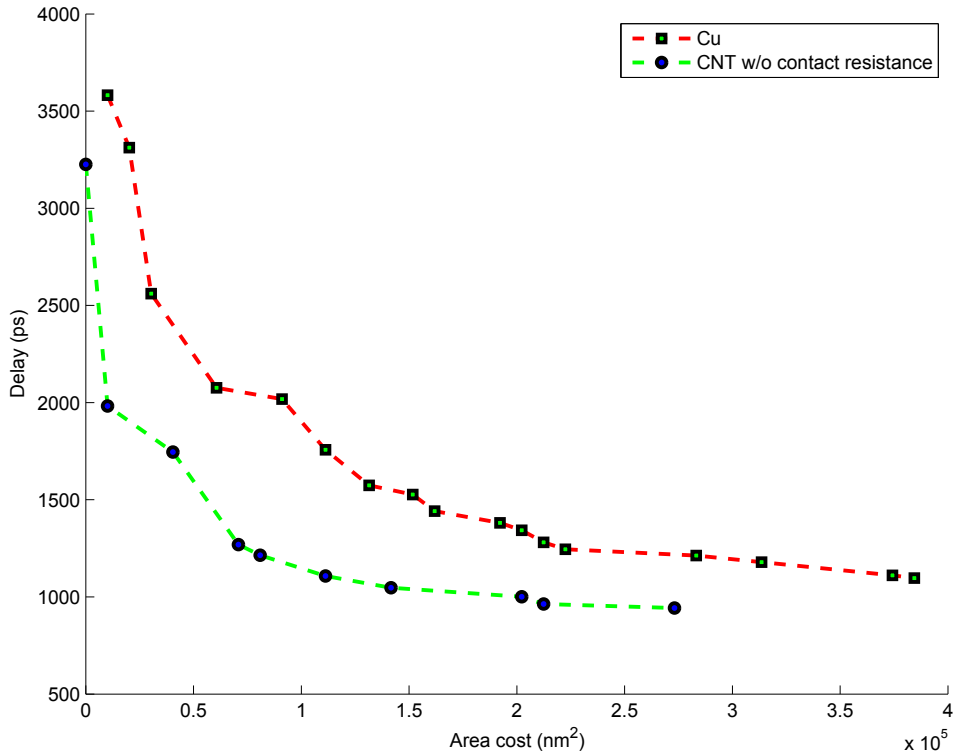



**Figure 2.5:** Area and delay comparison between Cu and CNT.

## 2.6 Summary

Carbon nanotube interconnects have become a promising replacement material for copper interconnects thanks to their superior conductivity. This work develops the first timing driven buffer insertion technique for carbon nanotube interconnects. In the experimental results, it demonstrates that with the same timing constraint, CNT buffering can save over 50% buffer area compared to copper buffering. In addition, CNT buffering can effectively reduce the delay by up to 32% without considering cost.

# Chapter 3

# Stochastic Buffering For Bundled SWCNT Interconnects Considering Unidimensional Fabrication Variation[1]

## 3.1 Introduction

The heterogeneous system architecture which leverages multicore computing paradigm has become increasingly popular and it has been successfully deployed in

[1]The material contained in this chapter was accepted to "*IEEE Transactions on Emerging Topics in Computing (TETC)*." See Appendix C.2 for the copyright permission from IEEE.

many application scenarios. Nevertheless, there are still various design challenges which need to be tackled, and timing minimization is a critical one among them. It is well known that interconnect delay has become the bottleneck of the circuit timing. However, existing copper based interconnects technologies are approaching their fundamental physical limits. Thus, novel interconnect materials such as carbon nanotube (CNT) become highly desirable. There are two types of CNTs regarding the electrical properties which are metallic CNTs (mCNTs) and semiconducting CNTs (sCNTs). Bundled metallic single walled carbon nanotubes (SWCNTs) have better electrical performance than copper in terms of e.g., superior conductivity and current carrying capabilities. Therefore, SWCNTs are suggested to be used as long global interconnects [2, 3, 11, 12]. However, due to the imperfectness of fabrication, the variations in the geometric parameters of CNTs will lead to significant timing yield reduction of the design [22].

Chemical vapor deposition (CVD) is the most popular technique for CNT fabrication [13, 14, 15, 16, 17]. In such a technique, CNTs grow along a single direction and their geometric parameters are controlled by the environment of the CVD system. For example, the diameters of tubes are highly dependent on the processing environment and their orientations are controlled by the direction of gas flow. CNT density variations are caused by non-uniform spacing between CNTs. Therefore, the number of CNTs in the bundled SWCNT interconnects may have a large variance. There are other variations from the imperfect fabrication process, such as the growth kinetics,

flow patterns variations, nucleation sites variations, and adsorption and diffusion variations, which can be translated to the variations on the resistance and capacitance of the bundled SWCNT interconnects. These variations could be partially captured by the density and diameter variations.

When using CNTs as interconnects, the main sources of variability include the diameters of CNT, the density of mCNTs in the bundle, the misalignment of CNTs and the presence of sCNTs [22, 23, 24, 25]. In this work, to illustrate our technique we focus on the diameter and density variations, while our approach can be easily adapted to handle other variations. The density and diameter variations will result in timing variations on the bundled SWCNT interconnects, impacting the timing yield of the CNT based circuit design.

In the literature, there are some works addressing the variations to carbon nanotube field effect transistor (CNFET) based circuit designs[26, 27]. The models for the impact of mCNTs on the CNFET and circuit delay are provided in [28, 29] and the impact of undeposited CNTs to the circuit delay is studied in [30, 31]. Shahi and Zarkesh-Ha propose an analytical model to predict gate delay variation induced from CNT density variation [32]. Zhang et al. consider spatial correlation in directional CNT growth which helps reduce the device-level failure by 350× [33]. A timing-driven placement method has been recently developed for CNFET circuits in [34].

Variations also manifest strongly in CNT interconnects. Refer to Figure 3.1. There

are some works modelling the fabrication impacts. For example, Nieuwoudt and Massoud model the variational impact on resistance, capacitance, inductance, and delay of bundled SWCNT interconnects [21]. However, it does not provide any closed form computation model for resistance and capacitance of the bundled SWCNT interconnects, and thus it cannot be extended to handle spatial correlation for our purpose. In [35], a statistical model is proposed to analyze the crosstalk noise induced by process variations on SWCNT based on a closed-form derivation. However, it focuses on noise analysis which cannot be adapted to modelling timing and its does not consider unidimensional spatial correlation as well. In this work, we will present a new closed-form model for the resistance and capacitance of bundled SWCNT interconnects, which is integrated into our buffer insertion algorithm for timing evaluations.

A striking difference compared to the copper counterpart is that there exists spatial correlation in bundled SWCNT interconnect fabrication [22, 25, 36]. Since CNTs grow along one dimension in the fabrication, the spatial correlation in variations in CNTs is in one dimension, in contrast to the two dimensional spatial correlation in the copper based design. Such a property will be leveraged in developing our interconnect optimization technique for CNT based circuits.

In the literature, there are very limited works studying the carbon nanotubes based physical design. The first CNT buffer algorithm is developed in [3]. However, that work does not consider the fabrication variations which could significantly impact

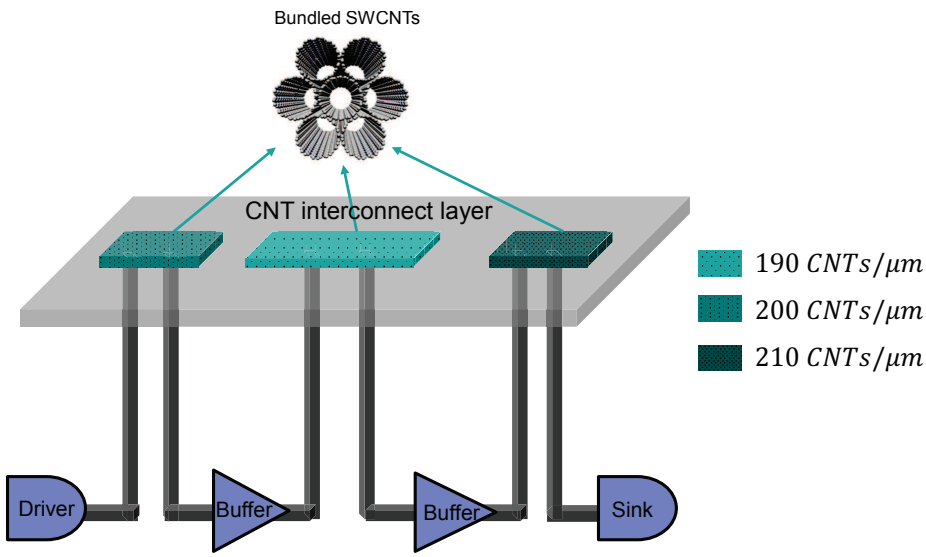



**Figure 3.1:** Illustration of fabrication variation aware buffer insertion problem for bundled SWCNT interconnects.

the performance of circuit design. This motivates us to model the variations, esp. the unidimensional spatial correlation of fabrication variations, on resistance and capacitance of bundled SWCNT interconnects, and develop a new stochastic CNT buffering algorithm based on this model. The contribution of this work is summarized as follows.

† Fabrication variation and unidimensional spatial correlation on the resistance and capacitance of bundled SWCNT interconnects are analyzed.

† A unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm is proposed. A new importance sampling based timing evaluation technique is also developed to improve the computational efficiency of the algorithm.

† To the best of our knowledge, this is the first work on the SWCNT interconnect optimizations considering variations.

† The experimental results on signal nets demonstrate that the proposed unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm saves over 30% buffer area over copper buffering on 50 nets while satisfying timing constraints. In addition, our proposed stochastic SWCNT interconnects buffering algorithm achieves much better performance than the best case design and the worst case design in terms of timing and buffer cost.

## 3.2 Preliminaries

### 3.2.1 Overview of The Deterministic CNT Buffering Algorithm

In the literature, there are several buffer insertion algorithms for copper interconnects such as [7, 8, 9, 10]. A similar algorithm for buffering CNT interconnects is developed in [3]. However, it is only for deterministic optimization which does not consider the fabrication variations. In contrast, this work develops a new variation aware buffer insertion, which actually utilizes the technique in [3] as a component. For

completeness, some details of [3] are included as follows.

The inputs to the buffer insertion problem include a routing tree and a buffer library. Let $T = (V, E)$ denote the routing tree, where $V = s_0 \cup V_s \cup V_c$, and $E \in V \times V$, where $s_0$ is the driver, $V_s$ is the set of sinks, and $V_c$ is the set of candidate buffer locations. Each sink $s$ has a sink capacitance and a required arrival time. In the deterministic buffer insertion problem, a buffered tree satisfies the timing constraint if and only if its required arrival time at the driver is no earlier than the arrival time. Each edge $e$ in the tree has a resistance and a capacitance. In the buffer insertion literature, the routing tree is typically assumed to be binary [7, 8, 9] since otherwise the tree can be easily converted to a binary one [8]. A set of candidate buffer locations along the routing are also given in practice (which can be computed using e.g., the technique in [55]). A buffer library $B$ is available to the buffer insertion problem. Each buffer type $b$ has a cost $w_{bj}$, an input capacitance $c_{bj}$ and a driving resistance $r_{bj}$. Given a binary routing tree in a nanotube wire layer and a buffer library, the deterministic buffer insertion algorithm asks to determine the locations and types of buffers to be inserted while satisfying the timing constraint.

In the deterministic CNT buffering algorithm, a 3-tuple $(Q, C, W)$ characterizes a buffering solution, where $Q$ is the required arrival time, $C$ is the downstream capacitance, and $W$ is the cumulative buffer cost. In the algorithm, a set of candidate buffering solutions represented by those 3-tuples are propagated from the sinks to the

driver. The propagation process starts with a sink and it computes $Q$, $C$ and $W$ all the way to the driver. During this process, inferior solutions are pruned for speedup. At the driver, the solution with the minimum buffer cost satisfying timing constraint is returned. Let $\gamma$ denote a solution. For any two solutions $\gamma_1, \gamma_2$ at the same node, $\gamma_1$ is inferior to $\gamma_2$ if $Q(\gamma_1) \leq Q(\gamma_2)$, $C(\gamma_1) \geq C(\gamma_2)$ and $W(\gamma_1) \geq W(\gamma_2)$. The algorithm has three operations which are add buffer, add wire and branch merge. Note that the add driver can be easily implemented using add buffer [3].

#### 3.2.1.1 Add Buffer

To insert a buffer at a candidate buffer location $v$, the buffering solution $\gamma$ will be updated to $\gamma'$. Let $R(b)$ denote the buffer resistance, $C(b)$ denote buffer capacitance, and $t(b)$ denote buffer intrinsic delay for a buffer type $b$. Eqn. 3.1 is used in [3] to update a solution. Clearly, the required arrival time is updated considering the buffer delay, the capacitance is set to the input capacitance of $b$, and $W(\gamma')$ is computed as $W(\gamma') = W(\gamma) + W(b)$.

$$\begin{aligned} Q(\gamma') &= Q(\gamma) - R(b) \cdot C(\gamma) - t(b) \\ C(\gamma') &= C(b) \\ W(\gamma') &= W(\gamma) + W(b). \end{aligned} \tag{3.1}$$

#### 3.2.1.2 Add Wire

[3] treats the resistance of bundled SWCNT interconnects as a deterministic value which is $\frac{6.45k\Omega}{N_{cnt}}/\mu m$, where $N_{cnt}$ is the number of nanotubes in the bundled SWCNT interconnects. To add a wire from a location $v$ to its upstream location $u$, the buffering solution can be updated as in Eqn. 3.2.

$$\begin{aligned} Q(\gamma_u) &= Q(\gamma_v) - (R_q(u,v) \\ &\quad + R_s(u,v)) \cdot [\frac{C(u,v)}{2} + C(\gamma_v)] \\ C(\gamma_u) &= C(\gamma_v) + C(u,v) \\ W(\gamma_u) &= W(\gamma_v). \end{aligned} \tag{3.2}$$

#### 3.2.1.3 Branch Merge

This operation is performed when two branches are to be merged in the routing tree. Given two solutions $(Q(\gamma_1), C(\gamma_1), W(\gamma_1))$ and $(Q(\gamma_2), C(\gamma_2), W(\gamma_2))$ associated with different branches, they can be merged as in Eqn. 3.3. Clearly, the merged required arrival time is the minimum of those on two branches, the total downstream capacitance is the sum of those downstream capacitance on two branches, and the total buffer cost is also the sum of buffer costs on two branches.

$$
\begin{aligned}
Q(\gamma) &= min\{Q(\gamma_1), Q(\gamma_2)\} \\
C(\gamma) &= C(\gamma_1) + C(\gamma_2) \\
W(\gamma) &= W(\gamma_1) + W(\gamma_2).
\end{aligned}
\tag{3.3}
$$

### 3.2.2 Problem Formulation

Considering the imperfectness of CNT fabrication, variations of geometric parameters such as density and diameter impact the buffer insertion assignment. The target of our variation aware buffer insertion is to guarantee the timing of buffered routing trees satisfying timing constraints after fabrication with certain high probability. In this work, 99% probability is chosen to demonstrate our technique, while others ratios can be easily handled. The timing corresponding to the 99% probability is called 99% timing which can be computed using simulations as follows. Given a buffered routing tree, one generates $n$ samples to simulate the fabrication process and the timing of each sample is evaluated. The timings of these $n$ samples are then sorted according to the increasing order and the $99\% n - th$ largest timing in this list is denoted as the 99% timing of the buffered tree. Given the unidimensional variation model described in Section 3.3, our problem is formulated as follows.

**Unidimensional Variations Aware Timing Constrained Minimum Cost Buffering for Bundled SWCNT Interconnects**: Given a binary routing tree

with a set of candidate buffer locations in bundled SWCNT routing layers, variation models of bundled SWCNT interconnects and a buffer library, to compute a buffer assignment solution such that the 99% timing of the routing tree satisfies the timing constraint and the total buffer cost is minimized.

## 3.3 Unidimensional Variation Model of Bundled SWCNT Interconnects

### 3.3.1 Variation Model of Resistance and Capacitance of SWCNT Interconnects

Due to the lack of precise control over CNT growing during the fabrication process, there can be significant variations. Refer to Figure 3.2 for a deterministic bundled SWCNT interconnects model [3]. $R_{dr}$ and $C_{dr}$ denote the resistance and capacitance of the driver. $R_{c,downstream}$ is the contact resistance between the driver and the bundled SWCNT interconnects and $R_{c,upstream}$ is the contact resistance between the bundled SWCNT interconnects and load capacitance $C_{load}$. $R_S^{bundle}$ is the scattering resistance of bundled SWCNT interconnects, and $C_E^{bundle}$ is the electrostatic capacitance of bundled SWCNT interconnects. Note that this model only considers

resistance and capacitance of bundled SWCNT interconnects since the inductance is negligible for prevailing designs and the RC model is as accurate as RLC model according to [2, 3, 71]. This model will be augmented to consider variations.

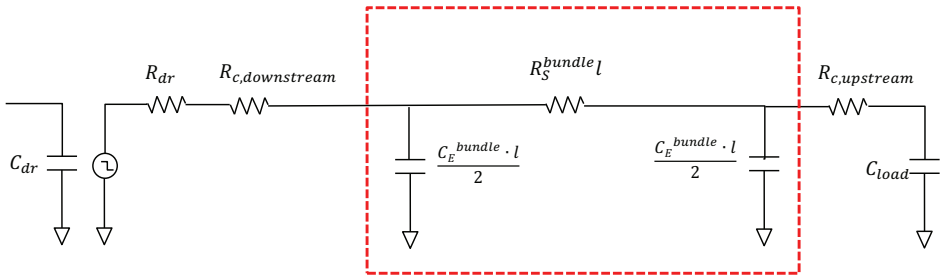



**Figure 3.2:** Simplified Equivalent $\pi$ circuit model for bundled SWCNT interconnects [3].

The resistance of bundled SWCNT interconnects consists of quantum resistance and scattering resistance. However, the resistance for global bundled SWCNT interconnects is basically equal to the scattering resistance of bundled SWCNT interconnects [3]. The scattering resistance is a function with density of the bundled SWCNTs. To consider the variations on resistance, motivated by [72] which uses first-order Taylor series expansion to approximate the gate and interconnect delays, the resistance of a wire can be expressed as follows.

$$\begin{aligned} R_v = R_S^{bundle} &= \frac{R_S l}{N_{cnt}} = \frac{R_S l}{s\delta} \\ &\approx R_{v_0} + \frac{\partial R_S^{bundle}}{\partial \delta}|_{\delta=\delta_0} \Delta\delta \\ &\approx R_{v_0} - \frac{R_S l_0}{s\delta_0^2} \Delta\delta \\ &\approx R_{v_0} - \frac{6.45 l_0}{s\delta_0^2} \Delta\delta, \end{aligned} \tag{3.4}$$

where $R_S^{bundle}$ is scattering resistance of bundled SWCNT interconnects, $R_S$ is unit scattering resistance of an isolated SWCNT, $l$ is length of an isolated SWCNT, $l_0$ is nominal length of an isolated SWCNT, $N_{cnt}$ is number of SWCNTs in the bundle, $s$ is cross section area of bundled SWCNTs, $\delta$ is density of bundled SWCNTs, and $\delta_0$ is nominal density of bundled SWCNTs.

It has been demonstrated in [22] that the density, i.e., the nanotube count, of bundled SWCNT interconnect follows normal distribution. Since the resistance $R_v$ is a linear function of the density according to Eqn. 3.4, the resistance $R_v$ also follows normal distribution. The mean value of $R_v$ is $R_{v_0}$, and the variance $\sigma^2_{R_N}$ is

$$\sigma^2_{R_N} = (\frac{6.45 l_0}{s\delta_0^2})^2 \sigma^2(\delta). \tag{3.5}$$

The capacitance of bundled SWCNT interconnects consists of quantum capacitance and electrostatic capacitance. According to [2], the effective capacitance of bundled SWCNT interconnects is nearly equal to its electrostatic capacitance and the effect

of the quantum capacitance is negligible. Therefore, the normally distributed capacitance of an isolated CNT interconnect is

$$\begin{aligned} C_v = \sum C_E &= \sum \frac{2\pi\epsilon}{cosh^{-1}y/d} \\ &\approx \sum (C_{v_0} + \frac{\partial C_E}{\partial d}|_{d=d_0}\Delta d) \\ &\approx \sum (C_{v_0} \pm \frac{y}{d_0^2(cosh^{-1}(y/d_0))^2\sqrt{((y/d_0)^2-1)}}\Delta d), \end{aligned} \tag{3.6}$$

where $C_E$ is the electrostatic capacitance of an isolated SWCNT, $C_{v_0}$ is the nominal capacitance of bundled SWCNT interconnects, $y$ is distance between an isolated SWCNT and ground, $d$ is diameter of an isolated SWCNT, and $d_0$ is nominal diameter of an isolated SWCNT. Note that the distance $y$ between the SWCNTs and the ground can be treated as constant. Similar to the above analysis, since the diameter of carbon nanotubes follows normal distribution [22], the capacitance $C_E$ also follows normal distribution. The mean value of $C_v$ is $C_{v_0}$, and the variance of $C_E$ is

$$\sigma^2_{C_N} = (\frac{y}{d_0^2(cosh^{-1}(y/d_0))^2\sqrt{((y/d_0)^2-1)}})^2\sigma^2(d). \tag{3.7}$$

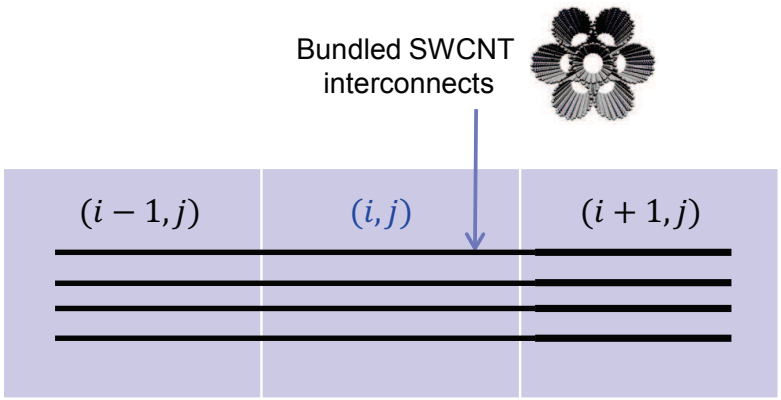



**Figure 3.3:** Spatial correlation illustration of the bundled SWCNT interconnects.

### 3.3.2 Variation Model of Resistance and Capacitance of SWCNT Interconnect Considering Unidimensional Spatial Correlation

Since CNTs grow along one dimension, after fabrication the geometric parameters (such as diameter $d$ and density $\delta$) of SWCNTs at different locations in this dimension exhibit strong correlations. In addition, the less distance between CNTs along this dimension, the more spatial correlations they have [22, 25, 36]. Motivated by [72] which models the spatial correlations for copper interconnects, one can model the unidimensional correlation on SWCNTs as follows. Refer to Figure 3.3. Along the CNT growing dimension, the circuit layout can be partitioned into a set of grids.

The resistance and capacitance of bundled SWCNT interconnects in a grid $(i, j)$ considering the spatial correlation with its neighboring grids $(i-1, j)$ and $(i+1, j)$ can be modeled as

$$\begin{aligned} R_{vs_{ij}} &= R_{vs}(\delta_{i-1,j}, \delta_{i,j}, \delta_{i+1,j}) \\ &\approx R_{vs_{ij_0}} + \frac{\partial R_{vs}}{\partial \delta_{i-1,j}} \Delta\delta_{i-1,j} + \frac{\partial R_{vs}}{\partial \delta_{i,j}} \Delta\delta_{i,j} + \frac{\partial R_{vs}}{\partial \delta_{i+1,j}} \Delta\delta_{i+1,j}, \end{aligned} \tag{3.8}$$

$$\begin{aligned} C_{vs_{ij}} &= C_{vs}(d_{i-1,j}, d_{i,j}, d_{i+1,j}) \\ &\approx C_{vs_{ij_0}} + \frac{\partial C_{vs}}{\partial d_{i-1,j}} \Delta d_{i-1,j} + \frac{\partial C_{vs}}{\partial d_{i,j}} \Delta d_{i,j} + \frac{\partial C_{vs}}{\partial d_{i+1,j}} \Delta d_{i+1,j}. \end{aligned} \tag{3.9}$$

According to the derivations of resistance and capacitance of bundled SWCNT interconnects, the resistance and capacitance of bundled SWCNT interconnects in grid $(i, j)$ considering spatial correlation can be updated as follows:

$$\begin{aligned} R_{vs_{ij}} \approx R_{vs_{ij_0}} &- \frac{6.45 l_{i-1,j,0}}{s\delta_{i-1,j,0}^2} \Delta\delta_{i-1,j} \\ &- \frac{6.45 l_{i,j,0}}{s\delta_{i,j,0}^2} \Delta\delta_{i,j} - \frac{6.45 l_{i+1,j,0}}{s\delta_{i+1,j,0}^2} \Delta\delta_{i+1,j}, \end{aligned} \tag{3.10}$$

$$\begin{aligned} C_{vs_{ij}} \approx C_{vs_{ij_0}} & \\ &\pm \frac{y}{d_{i-1,j,0}^2 (cosh^{-1}(y/d_{i-1,j,0}))^2 \sqrt{((y/d_{i-1,j,0})^2 - 1)}} \Delta d_{i-1,j}) \\ &\pm \frac{y}{d_{i,j,0}^2 (cosh^{-1}(y/d_{i,j,0}))^2 \sqrt{((y/d_{i,j,0})^2 - 1)}} \Delta d_{i,j}) \\ &\pm \frac{y}{d_{i+1,j,0}^2 (cosh^{-1}(y/d_{i+1,j,0}))^2 \sqrt{((y/d_{i+1,j,0})^2 - 1)}} \Delta d_{i+1,j}) \end{aligned} \tag{3.11}$$

The model of resistance and capacitance of bundled SWCNT interconnects has been developed considering the unidimensional variations on densities and diameters. This model will be used in the stochastic buffer insertion.

## 3.4 Unidimensional Variation Aware Importance Sampling Based Stochastic SWCNT Interconnects Buffering Algorithm

### 3.4.1 Algorithmic Flow

A new stochastic buffer insertion algorithm is developed in this work to handle the unidimensional correlation of fabrication variations on the bundled SWCNT interconnects. In the proposed algorithm, given the probabilistic distributions of the resistance and the capacitance of the bundled SWCNT interconnects, some high probability CNT parameter ranges of resistances and capacitances can be estimated. A parametric CNT buffering will be developed considering different resistances and capacitances. The buffering solutions will be evaluated using a novel importance sampling based method and the 99% timing (as defined in Section 3.2.2) will be estimated. The solution whose 99% timing satisfies the timing constraint and with the smallest buffer cost will be returned as the final solution.

The algorithmic flow is shown in Figure 3.4. It consists of three parts. The first part is to generate the high probability CNT parameter ranges of resistances and

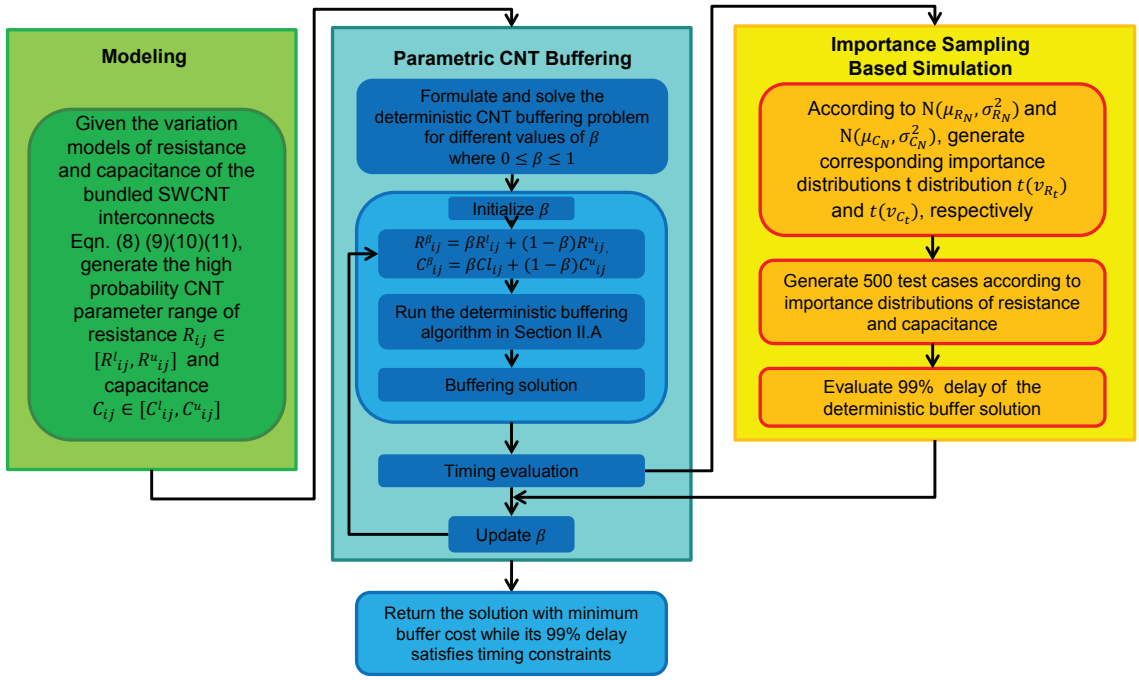



**Figure 3.4:** The algorithmic flow of the proposed unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm.

capacitances of bundled SWCNT interconnects. The resistances and capacitances are modeled as in Section 3.3.2, which considers the unidimensional spatial correlation in SWCNT fabrication variations. According to the three-sigma rule, 99.73% of the values of variables following a normal distribution $N(\mu, \sigma^2)$ lie within the range of $[\mu - 3\sigma, \mu + 3\sigma]$. Thus, with a high probability resistances and capacitances of bundled SWCNT interconnects in grid $(i, j)$ are distributed in the following ranges bounded by $[R^l_{ij}, R^u_{ij}]$, $[C^l_{ij}, C^u_{ij}]$, respectively, where

$$\begin{aligned} R^l_{ij} &= \mu_{R_N} - 3\sigma_{R_N}, \\ R^u_{ij} &= \mu_{R_N} + 3\sigma_{R_N}, \end{aligned} \tag{3.12}$$

and

$$C_{ij}^{l} = \mu_{C_N} - 3\sigma_{C_N},$$
$$C_{ij}^{u} = \mu_{C_N} + 3\sigma_{C_N}. \tag{3.13}$$

This is why the above ranges are called high probability CNT parameter ranges in this work.

The second part is a parametric CNT buffering algorithm. Motivated by [73], a parameter $\beta$ is used to model the uncertainty of the resistance and capacitance of the bundled SWCNT interconnects, as shown in Eqn. 3.14. If the lower bounds of the resistance $R_{ij}^{l}$ and capacitance $C_{ij}^{l}$ are used in the design, we call it the best case design. If the upper bounds of the resistance $R_{ij}^{u}$ and capacitance $C_{ij}^{u}$ are used in the design, we call it the worst case design. When $\beta = 1$, the resistances and capacitances are equal to the lower bounds (i.e. $R_{ij}^{l}, C_{ij}^{l}$), which is the best case design. When $\beta = 0$, the resistances and capacitances are equal to the upper bounds (i.e. $R_{ij}^{u}, C_{ij}^{u}$), which is the worst case design. Different tradeoff can be obtained through varying $\beta$ between 0 and 1. In fact, our algorithm is to find the best $\beta$ such that the 99% timing of the corresponding buffer insertion solution satisfies the timing constraint and is with minimum buffer cost. Given any $\beta$, the resistances and capacitances are deterministic values and then we run the deterministic dynamic programming based CNT buffering algorithm in [3] which is reviewed in Section 3.2.1 to compute the

corresponding buffering insertion.

$$
\begin{aligned}
R_{ij}^{\beta} &= \beta R_{ij}^{l} + (1-\beta) R_{ij}^{u}, \\
C_{ij}^{\beta} &= \beta C_{ij}^{l} + (1-\beta) C_{ij}^{u}
\end{aligned}
\tag{3.14}
$$

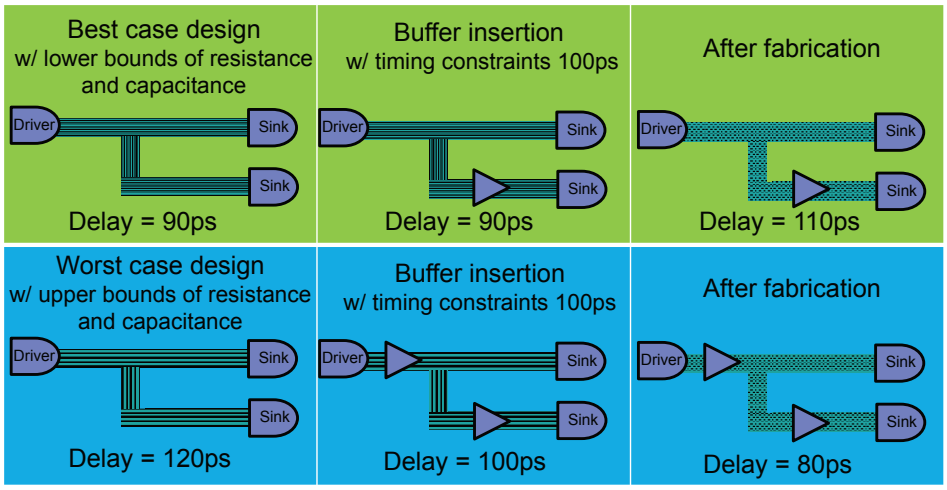



**Figure 3.5:** The illustrations of best case design and worst case design.

Different $\beta$ leads to different buffering solutions. Take the best case design and the worst case design as examples. Refer to Figure 3.5. In best case design, when the resistances and capacitances are set to the lower bounds, one just needs to insert few buffers to satisfy the timing constraints. However, such a design is too optimistic on variational impact, and the resulting buffered tree might not satisfy the timing constraint in many fabricated designs. In the worst case design, the resistances and capacitances are set to the upper bounds. Such a design is too conservative which means that there can be significant waste in buffer deployment.

The third part is to evaluate the timing of the obtained buffering solution. To estimate 99% delay (while other ratios can be easily handled) of a CNT based circuit, time consuming simulations are needed. The standard way is to perform the Monte Carlo simulations to evaluate the 99% delay of each buffering solution. For high accuracy, this typically requires a large amount (e.g., 10000 samples) of samples for each evaluation which is computationally expensive.

### 3.4.2 Importance Sampling For Timing Evaluation

An importance sampling method will be developed to accelerate the standard Monte Carlo simulation based timing evaluation. According to Section 3.3, the resistance $R_N$ follows normal distribution $N_R(\mu_{R_N}, \sigma^2_{R_N})$ and the capacitance $C_N$ follows normal distribution $N_C(\mu_{C_N}, \sigma^2_{C_N})$, with the mean values $\mu_{R_N} = R_{v0}$ and $\mu_{C_N} = C_{v0}$, and the variances $\sigma^2_{R_N}$ and $\sigma^2_{C_N}$ computed using Eqn. 3.5 and Eqn. 3.7. According to [74], the probability density of the normal distribution is

$$g(x|(\mu, \sigma^2)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{3.15}$$

where $\mu$ is the mean value of the distribution and $\sigma^2$ is the variance of the distribution.

Therefore, the probability density of normal distribution of resistance is

$$g(R_N|(\mu_{R_N}, \sigma^2_{R_N})) = \frac{1}{\sigma_{R_N}\sqrt{2\pi}} e^{-\frac{(R_N-\mu_{R_N})^2}{2\sigma^2_{R_N}}}. \tag{3.16}$$

Similarly, the probability density of normal distribution of capacitance is

$$g(C_N|(\mu_{C_N}, \sigma^2_{C_N})) = \frac{1}{\sigma_{C_N}\sqrt{2\pi}} e^{-\frac{(C_N-\mu_{C_N})^2}{2\sigma^2_{C_N}}}. \tag{3.17}$$

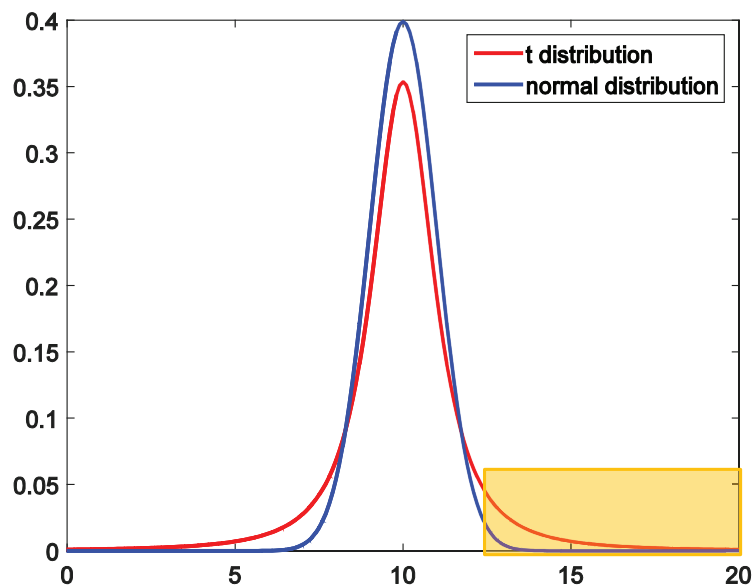



**Figure 3.6:** The comparison between a normal distribution and a $t$ distribution.

The idea of importance sampling is to generate samples according to a new distribution instead of the original distribution such that much fewer samples are needed in the new distribution so as to achieve the similar simulation accuracy [75]. The new

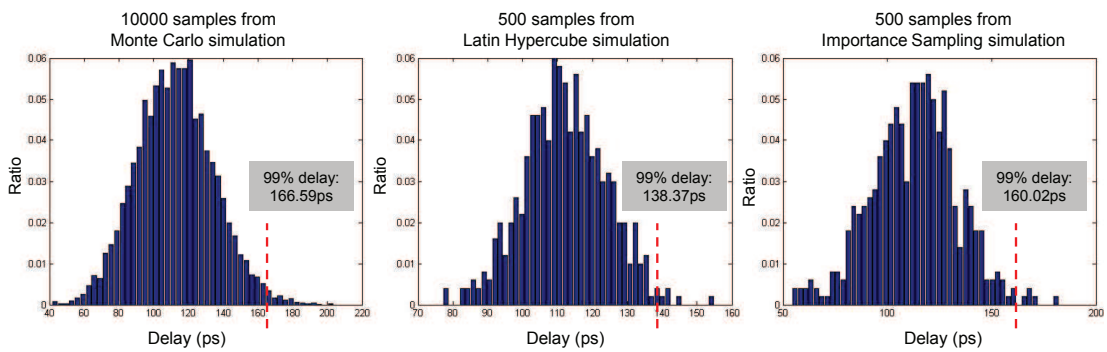



**Figure 3.7:** The comparison of 99% delay obtained from the standard Monte Carlo simulation with 10000 samples, Latin Hypercube sampling based simulation with 500 samples, and Importance Sampling based simulation with 500 samples.

distribution is called importance distribution. According to [76], $t$ distribution is a good choice as the importance distribution for normal distribution. The probability density function for $t$ distribution is

$$f(x|v) = \frac{\int_0^\infty x^{\frac{v+1}{2}-1} e^{-x} dx}{\sqrt{v\pi} \int_0^\infty x^{\frac{v}{2}-1} e^{-x} dx} (1 + \frac{x^2}{v})^{-\frac{v+1}{2}}, \tag{3.18}$$

where $v$ is the number of degrees of freedom in $t$ distribution. The variance of $t$ distribution is equal to $v/(v-2)$. The probability density of $t$ distribution of resistance is

$$f(R_t|v_{R_t}) = \frac{\int_0^\infty R_t^{\frac{v_{R_t}+1}{2}-1} e^{-R_t} dR_t}{\sqrt{v_{R_t}\pi} \int_0^\infty R_t^{\frac{v_{R_t}}{2}-1} e^{-R_t} dR_t} (1 + \frac{R_t^2}{v_{R_t}})^{-\frac{v_{R_t}+1}{2}}, \tag{3.19}$$

where $v_{R_t} = \frac{2\sigma_{R_N}^2}{\sigma_{R_N}^2 - 1}$.

Similarly, the probability density of $t$ distribution of capacitance is

$$f(C_t|v_{C_t}) = \frac{\int_0^\infty C_t^{\frac{v_{C_t}+1}{2}-1} e^{-C_t} dC_t}{\sqrt{v_{C_t}\pi} \int_0^\infty C_t^{\frac{v_{C_t}}{2}-1} e^{-C_t} dC_t} (1 + \frac{C_t^2}{v_{C_t}})^{-\frac{v_{C_t}+1}{2}}, \tag{3.20}$$

where $v_{C_t} = \frac{2\sigma_{C_N}^2}{\sigma_{C_N}^2 - 1}$.

Note that $t$ distribution has zero mean value. One can simply shift it to handle non-zero mean case. For this, one first generates samples $R'_t, C'_t$ using $t$ distribution, and then add $R_{v0}, C_{v0}$ to obtain the shifted samples $R_{v0} + R'_t, C_{v0} + C'_t$, which is denoted as $R_t, C_t$. Refer to Figure 3.6 for the comparison between normal distribution and $t$ distribution. $t$ distribution has a heavier tail, and thus the possibility to generate extreme cases is improved which means that the total number of samples generated using $t$ distribution can be significantly reduced. According to our experiences, $t$ distribution only needs 500 samples in simulation such that the computed 99% timing can well approximate the value obtained from the standard Monte Carlo simulation using 10000 samples.

Given the $k^{th}$ sample of wire segment $w$ generated from $t$ distribution, let $R_{t_k}^w$ denote the resistance and $C_{t_k}^w$ denote the capacitance. The delay of the wire segment $w$ in the $k^{th}$ sample generated from $t$ distribution is evaluated as $d_{t_k}^w$. According to the importance sampling, the delay of $t$ distribution needs to be transformed to that of

$$d^w_{n_k} = \frac{R^w_{t_k} C^w_{t_k}}{2} \cdot \frac{\frac{1}{\sigma_{R_N}\sqrt{2\pi}} e^{-\frac{(R^w_{t_k}-R_{v0})^2}{2\sigma^2_{R_N}}}}{\frac{\int_0^\infty (R^w_{t_k}-R_{v0})^{\frac{v_{R_t}+1}{2}-1} e^{-(R^w_{t_k}-R_{v0})} dR^w_{t_k}}{\sqrt{v_{R_t}\pi}\int_0^\infty (R^w_{t_k}-R_{v0})^{\frac{v_{R_t}}{2}-1} e^{-(R^w_{t_k}-R_{v0})} dR^w_{t_k}} (1+\frac{(R^w_{t_k}-R_{v0})^2}{v_{R_t}})^{-\frac{v_{R_t}+1}{2}}} \cdot \frac{\frac{1}{\sigma_{C_N}\sqrt{2\pi}} e^{-\frac{(C^w_{t_k}-C_{v0})^2}{2\sigma^2_{C_N}}}}{\frac{\int_0^\infty (C^w_{t_k}-C_{v0})^{\frac{v_{C_t}+1}{2}-1} e^{-(C^w_{t_k}-C_{v0})} dC^w_{t_k}}{\sqrt{v_{C_t}\pi}\int_0^\infty (C^w_{t_k}-C_{v0})^{\frac{v_{C_t}}{2}-1} e^{-(C^w_{t_k}-C_{v0})} dC^w_{t_k}} (1+\frac{(C^w_{t_k}-C_{v0})^2}{v_{C_t}})^{-\frac{v_{C_t}+1}{2}}} \tag{3.22}$$

normal distribution denoted as $d^w_{n_k}$, which can be computed as follows.

$$d^w_{n_k} = d^w_{t_k} \cdot \frac{g(R^w_{t_k}) \cdot g(C^w_{t_k})}{f(R^w_{t_k} - R_{v0}) f(C^w_{t_k} - C_{v0})}, \tag{3.21}$$

where $g(\cdot)$ is the probability density function of resistance and capacitance following normal distribution (Eqn. 3.16 and Eqn. 3.17), $f(\cdot)$ is the probability density function of resistance and capacitance following $t$ distribution (Eqn. 3.19 and Eqn. 3.20), and $d^w_{t_k} = \frac{R^w_{t_k} C^w_{t_k}}{2}$. Combining Eqn. 3.16, Eqn. 3.17, Eqn. 3.19 and Eqn. 3.20, Eqn. 3.21 can be derived to Eqn. 3.22. Using first order approximation, Eqn. 3.22 can be simplified to Eqn. 3.23. The factor $\frac{\Gamma(\frac{v_{R_t}}{2},-R_{v0})\Gamma(\frac{v_{C_t}}{2},-C_{v0})}{\Gamma(\frac{v_{R_t}+1}{2},-R_{v0})\Gamma(\frac{v_{C_t}+1}{2},-C_{v0})}$ is approximately equal to 1 if $v_{R_t} >> 1$ and $v_{C_t} >> 1$. The delay of the circuit $d_{n_k}$ for $k^{th}$ sample can be calculated using additions and multiplications. The set of timing values $D_n = \{d_{n_1}, d_{n_2}, ..., d_{n_K}\}$ are then sorted and 99% timing can be identified which is returned as the 99% delay.

Note that the Latin Hypercube (LH) sampling is also a popular method for improving

$$
\begin{aligned}
d^w_{n_k} &\approx \frac{\partial d^w_{n_k}}{\partial R^w_{t_k}}|_{R^w_{t_k}=R_{v0}}\Delta R^w_{t_k} + \frac{\partial d^w_{n_k}}{\partial C^w_{t_k}}|_{C^w_{t_k}=C_{v0}}\Delta C^w_{t_k} \\
&\approx \frac{v_{R_t} v_{C_t} C^w_{t_k} e^{-\frac{(C^w_{t_k}-C_{v0})^2}{2\sigma^2_{C_N}}} (1+\frac{(C^w_{t_k}-C_{v0})^2}{v_{C_t}})^{\frac{1+v_{C_t}}{2}} \Gamma(\frac{v_{R_t}}{2}, -R_{v0})\Gamma(\frac{v_{C_t}}{2}, -C_{v0})}{8\sigma_{R_N}\sigma_{C_N}\Gamma(\frac{v_{R_t}+1}{2}, -R_{v0})\Gamma(\frac{v_{C_t}+1}{2}, -C_{v0})}\Delta R^w_{t_k} \\
&+ \frac{v_{R_t} v_{C_t} R^w_{t_k} e^{-\frac{(R^w_{t_k}-R_{v0})^2}{2\sigma^2_{R_N}}} (1+\frac{(R^w_{t_k}-R_{v0})^2}{v_{R_t}})^{\frac{1+v_{R_t}}{2}} \Gamma(\frac{v_{R_t}}{2}, -R_{v0})\Gamma(\frac{v_{C_t}}{2}, -C_{v0})}{8\sigma_{R_N}\sigma_{C_N}\Gamma(\frac{v_{R_t}+1}{2}, -R_{v0})\Gamma(\frac{v_{C_t}+1}{2}, -C_{v0})}\Delta C^w_{t_k} \\
&\approx \frac{v_{R_t} v_{C_t} \Gamma(\frac{v_{R_t}}{2}, -R_{v0})\Gamma(\frac{v_{C_t}}{2}, -C_{v0})}{8\sigma_{R_N}\sigma_{C_N}\Gamma(\frac{v_{R_t}+1}{2}, -R_{v0})\Gamma(\frac{v_{C_t}+1}{2}, -C_{v0})} \cdot \Big( (C^w_{t_k} e^{-\frac{(C^w_{t_k}-C_{v0})^2}{2\sigma^2_{C_N}}} (1+\frac{(C^w_{t_k}-C_{v0})^2}{v_{C_t}})^{\frac{1+v_{C_t}}{2}})\Delta R^w_{t_k} \\
&+ R^w_{t_k} e^{-\frac{(R^w_{t_k}-R_{v0})^2}{2\sigma^2_{R_N}}} (1+\frac{(R^w_{t_k}-R_{v0})^2}{v_{R_t}})^{\frac{1+v_{R_t}}{2}} \Delta C^w_{t_k} \Big) \\
&\approx \frac{v_{R_t} v_{C_t}}{8\sigma_{R_N}\sigma_{C_N}} \cdot \Big( (C^w_{t_k} e^{-\frac{(C^w_{t_k}-C_{v0})^2}{2\sigma^2_{C_N}}} (1+\frac{(C^w_{t_k}-C_{v0})^2}{v_{C_t}})^{\frac{1+v_{C_t}}{2}})\Delta R^w_{t_k} \\
&+ R^w_{t_k} e^{-\frac{(R^w_{t_k}-R_{v0})^2}{2\sigma^2_{R_N}}} (1+\frac{(R^w_{t_k}-R_{v0})^2}{v_{R_t}})^{\frac{1+v_{R_t}}{2}} \Delta C^w_{t_k} \Big), where\ \Gamma(a,b) = \int_b^\infty x^{a-1}e^{-x}dx
\end{aligned}
\tag{3.23}
$$

the standard Monte Carlo simulation in terms of the computational efficiency. LH sampling, which is first proposed in [77], stratifies the input probability distributions. LH sampling divides the cumulative probability curve into equal probability intervals and takes a random value from each interval of the input distribution. However, the 99% delay are distributed along the right tails of the simulation space, which are rare events. Using the importance distribution, the tail probability is enlarged and less samples are needed to calculate the 99% delay. To verity the assumption, we design a case study and compare the LH sampling method with the $t$ distribution based importance sampling. According to Figure 3.7, 99% delay obtained from 10000 samples from normal distribution and 500 samples from $t$ distribution are close to each other. The 99% delay obtained from 500 samples of Latin Hypercube simulation is not as accurate as the importance sampling simulation. Thus, importance sampling

is chosen to estimate the timing.

## 3.5 Experimental Results

### 3.5.1 Experimental Setup

The proposed unidimensional variation aware importance sampling based stochastic SWCNT interconnect buffer insertion algorithm is implemented using C language and tested on a computer with 3.40GHz Intel Pentium CPU and 3GB memory. The test cases in [3] are scaled to 16nm technology node. In this work, the buffer area is used to measure the buffer cost. Our experiments are performed to 50 global nets extracted from an industrial ASIC chip in an old technology. Due to the lack of industrial nets in 16nm technology, wirelengths are scaled. According to the ITRS [68], the unit resistance and capacitance of CNT and copper are shown in Table 3.1. The results of fabrication variation aware CNT buffering are compared with copper buffering. Three sets of testcases are used in the experiments. According to [21, 23, 78], the variations of the resistance and capacitance of bundled SWCNT interconnects are set to be 5%, 10% and 20%, respectively. Since the impact of fabrication variations of global copper interconnects on the timing could be negligible according to [79], in our comparison no variations on copper interconnects are assumed.

**Table 3.1**
Unit resistance and capacitance (for $1\mu m$) of global interconnects with copper and bundled SWCNTs at 16nm technology node.

| Properties | Cu | CNT |
|---|---|---|
| Unit resistance ($\Omega/\mu m$) | 5.38 | 2.86 |
| Unit capacitance ($fF/\mu m$) | 0.16 | 0.16 |

### 3.5.2 Experimental Results

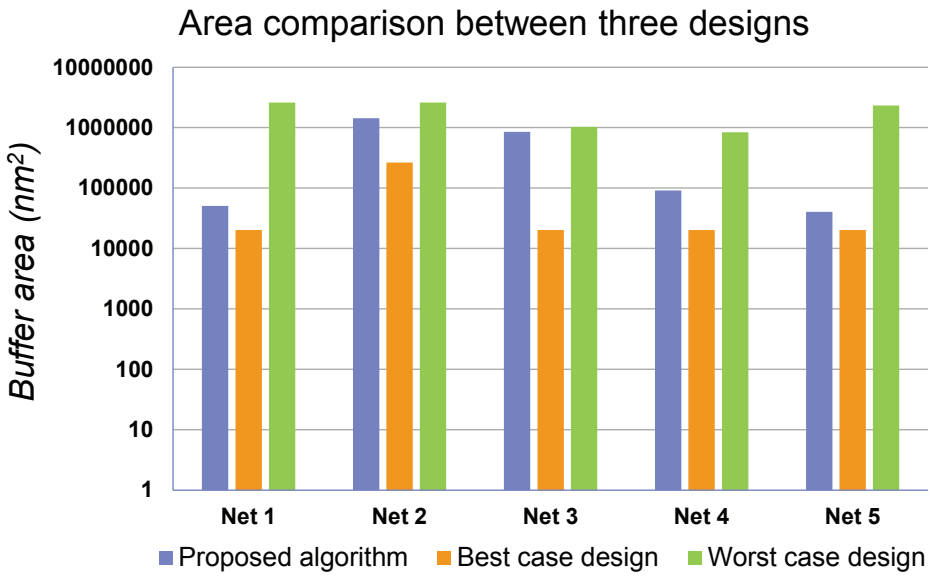
**Table 3.2**
Average results for timing constrained minimum cost buffering on 50 nets comparing with copper buffering. No variations on copper interconnects and 5%, 10%, 20% variations on resistances and capacitances of bundled SWCNT interconnects are considered, respectively.

| Test cases | Buf Area ($nm^2$) | Buf Area Ratio | 99% Delay (ps) |
|---|---|---|---|
| CNT (5%) | 33800.78 | 0.660 | 908.32 |
| CNT (10%) | 32369.92 | 0.632 | 904.99 |
| CNT (20%) | 33583.79 | 0.656 | 901.35 |
| Cu | 51184.90 | 1 | 912.54 |

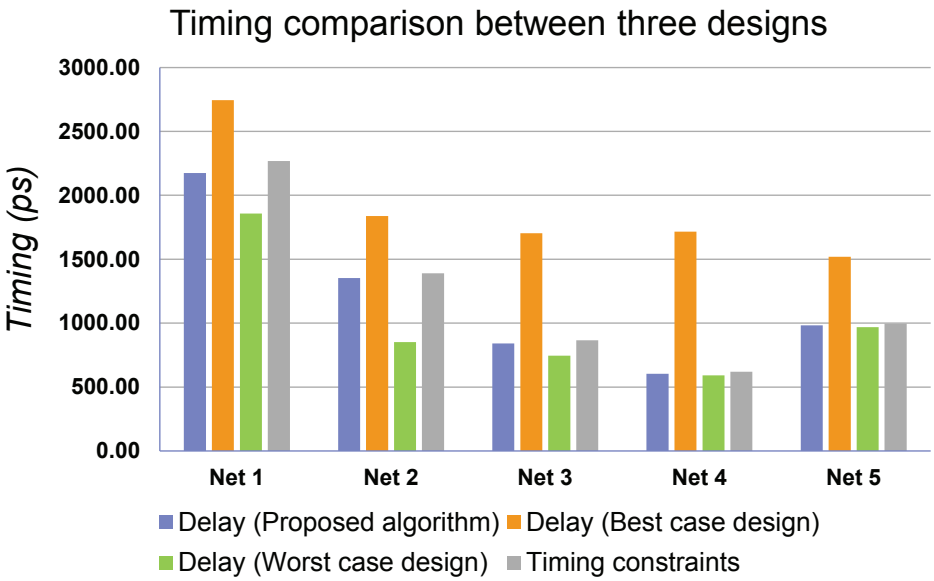
**Table 3.3**
Stochastic buffering results on 5 representative nets comparing with the best case design and the worst case design with 10% variations on resistances and capacitances of bundled SWCNT interconnects.

| Test cases | | Net 1 | Net 2 | Net 3 | Net 4 | Net 5 |
|---|---|---|---|---|---|---|
| The proposed algorithm | Buf Area ($nm^2$) | 50578.00 | 1426493.20 | 850000.80 | 91040.40 | 40462.40 |
| | Delay (ps) | 2174.82 | 1352.28 | 841.44 | 603.61 | 982.84 |
| Best case design | Buf Area ($nm^2$) | 20231.20 | 263392.80 | 20231.20 | 20231.20 | 20231.20 |
| | Delay (ps) | 2743.54 | 1837.82 | 1703.79 | 1715.34 | 1518.52 |
| Worst case design | Buf Area ($nm^2$) | 2599709.20 | 2599709.20 | 1032033.20 | 839885.20 | 2316520.80 |
| | Delay (ps) | 1857.37 | 851.64 | 745.89 | 591.98 | 968.82 |

The proposed unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm is compared with copper buffering. The comparison results of 50 nets are shown in Table 3.2. The variations of resistance and capacitance of the bundled SWCNT interconnects are set to be 5%, 10% and 20%,




(a) Buffer area comparison




(b) Timing comparison

**Figure 3.8:** Buffer area and timing comparison between the proposed design and the best case design and the worst case design.

respectively. In the experimental results, the delay of CNT based design refers to the 99% delay obtained from importance sampling based simulations, while the delay of

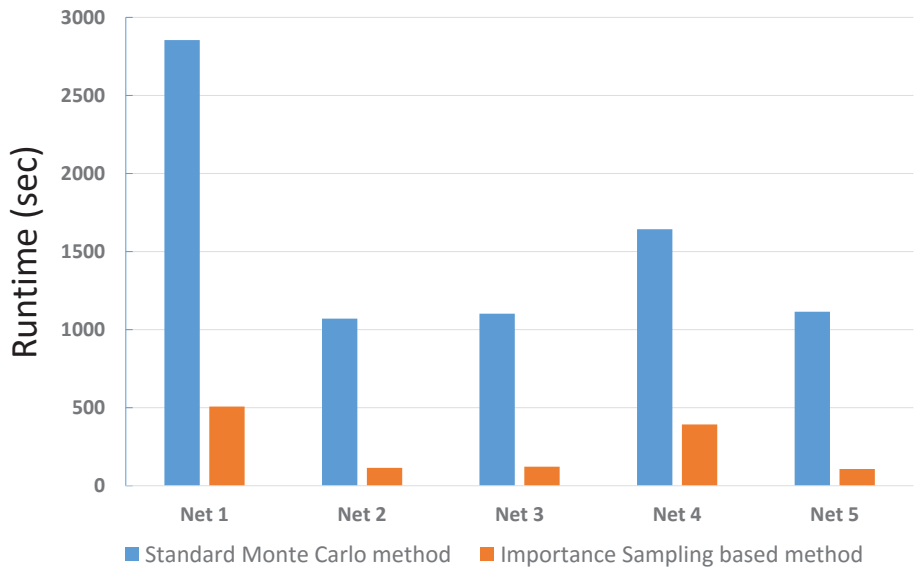



**Figure 3.9:** Runtime comparison between the standard Monte Carlo method and the importance sampling based method.

copper design refers to the nominal delay without considering variations since global copper interconnect variations are not important according to [79]. Thus, our comparison would be actually in favor of copper based design. Nevertheless, one can observe that, with 5% variation, the fabrication variation aware CNT buffering reduces the buffer area by 34.0% on average compared to the copper buffering, while the 99% delays of their solutions are still better than the nominal delays of copper buffering solutions. With 10% variation, the fabrication variation aware CNT buffering reduces the buffer area by 36.8% on average, and with 20% variation, the fabrication variation aware CNT buffering reduces the buffer area by 34.4%, respectively.

To study the impact of variations to the buffered CNT based designs, the proposed algorithm is also compared with the best case design and the worst case design. The

results for five representative nets are shown in Table 3.3, Figure 3.8, and Figure 3.9. We make the following observations.

- † Recall that in the best case design, the resistances and capacitances of the bundled SWCNT interconnects are always set to the lower bounds. One observes that best case design can achieve the smallest buffer areas. However, the 99% delays do not satisfy the timing constraints, which means that many fabricated designs cannot meet timing targets and thus these solutions are useless. Note that sometimes buffer area of the best case design is zero, which means no buffer is inserted. It is due to that the best case design is too optimistic on the variational impact to interconnects.

- † In the worst case design, the resistances and capacitances of the bundled SWCNT interconnects are always set to the upper bounds. According to the experimental results, the 99% delay can always satisfy the timing constraints. However, the worst case design is too conservative and significant amount of buffers are wasted.

- † Comparing to the best case design and the worst case design, the 99% delays of the proposed algorithm always satisfy the timing constraints while the buffer area is much less than the worst case design. The buffer cost reduction is very significant.

- † To evaluate the efficiency of our technique, the comparison of runtime between

the proposed importance sampling based technique (using 500 samples) and the standard Monte Carlo method (using 10000 samples) is conducted. Compared with the standard Monte Carlo method, the runtime of the proposed algorithm is on average reduced by 84.03%.

## 3.6 Summary

In this work, the models of the resistance and capacitance of the bundled SWCNT interconnects are analyzed considering unidimensional spatial correlation. A unidimensional variation aware importance sampling based stochastic SWCNT interconnects buffering algorithm is then developed. The experimental results demonstrate that our algorithm on average saves more than 30% buffer area over copper buffering while satisfying timing constraints. In addition, our proposed stochastic SWCNT interconnects buffering algorithm achieves much better performance than the best case design and the worst case design in terms of timing and buffer cost. To the best of our knowledge, this is the first work on the bundled SWCNT interconnect optimizations considering variations.

# Chapter 4

# Lorenz Chaotic System Based Carbon Nanotubes Physical Unclonable Functions[1]

## 4.1 Introduction

Physical unclonable function (PUF) is an emerging technology for security applications, such as true random number generation, secure key generation, low-cost authentication, etc [37, 38]. Most conventional encryption methodologies rely on the

[1]The material contained in this chapter is submitted to a journal publication.

secure keys stored in flash or non-volatile memory, and they are vulnerable to physical attacks. As an alternative approach, PUF exploits the hardware fabrication variations and generates unpredictable secure information in a storage-less fashion. For example, Gassend et al. propose an arbiter PUF through leveraging the timing difference on the two identically designed paths due to fabrication variations [39]. The input signal of the PUF is called the challenge and the output signal is called the response.

Given a PUF design, we can have many fabricated chips. The only differences among each fabricated chip are from the fabrication variations, which is not predictable nor clonable. The carbon nanotubes are promising candidates for highly secure PUF design thanks to their significant fabrication variations [23]. Chemical vapor deposition (CVD) is the most popular method for CNT fabrication, in which the pressure and temperature of the environment have significant effects on the fabricated features such as diameters and densities of CNTs. When CNTs are used as FETs and interconnects for reliable designs, the fabrication variations are not desired [21, 25, 40]. However, these inherent imperfections make the CNT based circuits natural candidates for building highly secure PUFs. Several carbon nanotube PUFs (CNT PUFs) are designed in the previous works such as [1, 41, 42, 43], which demonstrate significant advantages such as low cost and significant randomness. A secure empirical mode decomposition projection based CNT PUF design is developed in [41]. The other prominent one is discussed in [1] where self-assembled CNTs are used to design a

random bit generation approach for low-cost and hard-to-forge security applications. In Figure 4.1, individual CNTs are placed between two layers which can be randomly connected or disconnected [1].

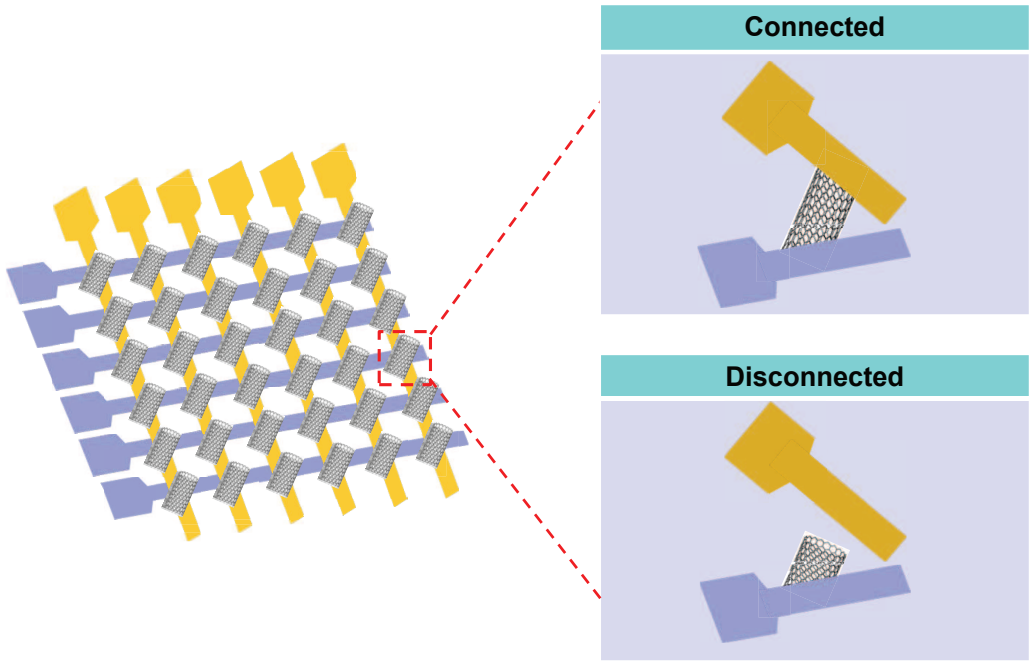



**Figure 4.1:** The illustration of 2D CNT bitarray crossbar structure (adapted from [1]).

Although CNT PUF designs have many advantages, they are still vulnerable to machine learning modeling attacks, where the attacker does not need to access the PUF hardware physically. The attacker collects a large amount of challenge-response pairs as the training data. Machine learning modeling attack methods are then used to model the PUF. Subsequently, the attacker can use the model to predict the responses given new challenges. Deep learning (DL) and evolution strategy (ES) are the most prominent machine learning methods used for modeling attacks [44]. It is

demonstrated in some works [45, 46, 47] that machine learning modeling attacks can achieve high prediction rate, e.g., 99.9%.

This motivates [48, 49, 50, 51] to design PUFs resistant to modeling attacks. In [48], a secure physically-embedded data encryption architecture is proposed by replacing conventional weak arbiter PUF with a specific strong PUF proposed in [49]. However, it is not easy to build that specific strong PUF proposed in [49]. A circuit that relies on non-linear current mirrors is designed to generate modeling resistant PUF in [50]. The current sources are assumed to be ideal which is impractical. In [51], the authors propose a lockdown technique in the PUF based system by adding server authentication. It could effectively prevent the attacker to collect many challenge-response pairs. However, the lockdown technique is of low efficiency.

In this paper, our objective is to design a CNT PUF which is resistant to machine learning modeling attacks. First, one needs to know how the modeling attack methods work. For most PUFs, similar challenges could generate similar responses [52, 53, 54]. Therefore, one possible method for preventing modeling attacks is to magnify the differences among responses of similar challenges. Since Lorenz chaotic system yields widely diverging outputs given similar inputs, it motivates us to develop a novel CNT PUF design by leveraging Lorenz chaotic system. To the best of our knowledge, this is the first such work in CNT PUF design. To demonstrate the effectiveness of our proposed Lorenz chaotic system based CNT PUF, various machine learning

attacks are preformed, including Support Vector Machine (SVM), Deep Learning (DL), Logistic Regression (LR) and Evolution Strategies (ES). The experimental results demonstrate that the proposed Lorenz chaotic system based CNT PUF is robust to these attacks. The main contribution of this work is summarized as follows.

- † In this paper, a novel CNT PUF design is developed by leveraging Lorenz chaotic system. Lorenz chaotic system magnifies the differences among responses of similar challenges, which makes the proposed PUF design resistant to modeling attacks.
- † To demonstrate the security performance of the proposed PUF, various machine learning methods are used on the proposed PUF, including SVM, DL, LR and ES.
- † The experimental results demonstrate that the machine learning modeling attack methods can achieve as high as 100% bit-wise prediction rates on the CNT PUF without Lorenz chaotic system, while they can only obtain less than 55% bit-wise prediction rates on the proposed Lorenz chaotic system based CNT PUF. To the best of our knowledge, this is the first work to leverage Lorenz chaotic system to CNT PUF.
- † The significant security performance of the proposed PUF is mainly contributed by Lorenz chaotic system. However, if one uses Lorenz chaotic system only in the design, the parameters need to be induced by fabrication variations, which

could be complicated. Therefore, one needs to combine CNT PUF and Lorenz chaotic system as discussed in this paper.

The rest of this paper is organized as follows. Lorenz chaotic system is overviewed in Section 4.2. Lorenz chaotic system based CNT PUF is proposed in Section 4.3. The machine learning modeling attack methods are discussed in Section 4.4. The experimental results and analysis are presented in Section 4.5. A summary of this paper is given in Section 4.6.

## 4.2 Preliminaries

### 4.2.1 Lorenz Chaotic System

Chaos theory is used to study the behavior of dynamic system that are highly sensitive to initial conditions, which is referred as the butterfly effect. Small differences in initial conditions yield widely diverging outcomes in a Lorenz chaotic system. Therefore, Lorenz chaotic system has desirable features for encryption which been studied in some previous works [80, 81]. In addition, besides the original Lorenz chaotic system, some other chaotic systems are proposed in the literature, such as the chaotic Chen

system [82], Rössler system [83] and three-dimensional conservative quadratic systems [84] [85]. The chaotic Chen system is a dual of the Lorenz system. Since Lorenz chaotic is the classic and most widely used, it is considered in this paper.

The standard Lorenz chaotic system is shown as below:

$$\begin{aligned} x' &= -\sigma x + \sigma y \\ y' &= -xz + \gamma x - y \\ z' &= xy - \beta z, \end{aligned} \tag{4.1}$$

where $x$, $y$, and $z$ are the input variables, $\sigma$, $\gamma$, and $\beta$ are system parameters.

### 4.2.2 Discrete Lorenz Chaotic System

Based on the standard Lorenz chaotic system, the discrete Lorenz chaotic system can be derived as follows [86]:

$$
\begin{aligned}
x_{i+1} &= \sigma(y_i - x_i) + x_i \\
y_{i+1} &= -x_i z_i + \gamma x_i \\
z_{i+1} &= x_i y_i - \beta z_i + z_i,
\end{aligned}
\tag{4.2}
$$

where $x_i$, $y_i$, and $z_i$ are the input variables, $\sigma$, $\gamma$, and $\beta$ are system parameters. Given initial inputs $x_0$, $y_0$, and $z_0$, one can iterate the discrete Lorenz chaotic system $n$ times and generate the output values $x_n$, $y_n$, and $z_n$.

Refer to Figure 4.2. There are two sets of $x$ values and the only difference between the two sets is the initial value of $x_0$. One is with $x_0 = 1$ and the other is with $x = 1.0001$, and all other parameters are the same. It can be observed that the values of $x$ over iterations are quite different between the two sets. In other words, the output is very sensitive to the initial inputs and other parameters. Therefore, the discrete Lorenz chaotic system can be used to design modeling attack resistant PUFs.

Refer to Figure 4.3. Another example with binary input values is designed as follows. The inputs of the discrete Lorenz chaotic system are a set of 16-bit strings which are used as $x_0$ and shown in Figure 4.3 (a), and all other values are set to certain numbers. The discrete Lorenz chaotic system is performed for 10 iterations, the outputs are obtained and shown in Figure 4.3 (b). Next, we flip the least significant bit of the inputs to obtain a new set as shown in Figure 4.3 (c). For example, if the least significant value is 1 in Figure 4.3 (a), we set it to 0 in Figure 4.3 (c).

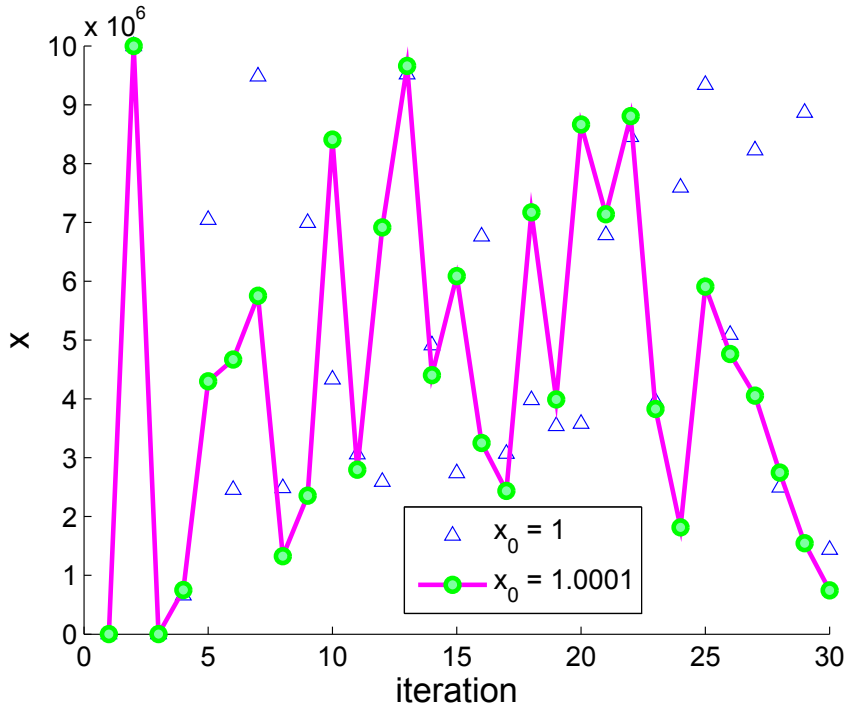
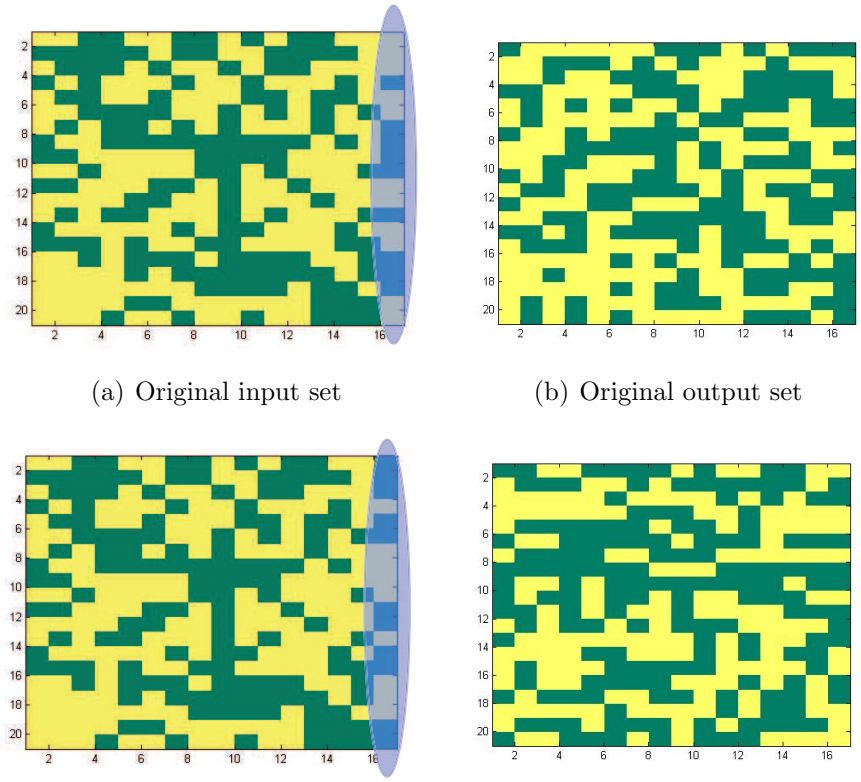



**Figure 4.2:** The value of x over iterations with two slightly different initial inputs.

Subsequently, the outputs are generated, which are shown in Figure 4.3 (d). It is clearly shown that the slight changes in the initial inputs could result in significant changes in the outputs of discrete Lorenz chaotic system.



(a) Original input set (b) Original output set

(c) Updated input set (d) Updated output set

**Figure 4.3:** The comparison between two sets with similar inputs (The least significant bits of the two set of inputs are complementary).

## 4.3 Lorenz Chaotic System Based CNT PUF

### 4.3.1 CNT Crossbar Structure

An unclonable electronic random structure is designed to generate two-dimensional (2D) random bit arrays in [1]. Single carbon nanotubes are used as switches in the 2D structure. Refer to Figure 4.4. In [1], the authors use the inherent CNT fabrication imperfections to construct an unclonable electronic random structure at low cost

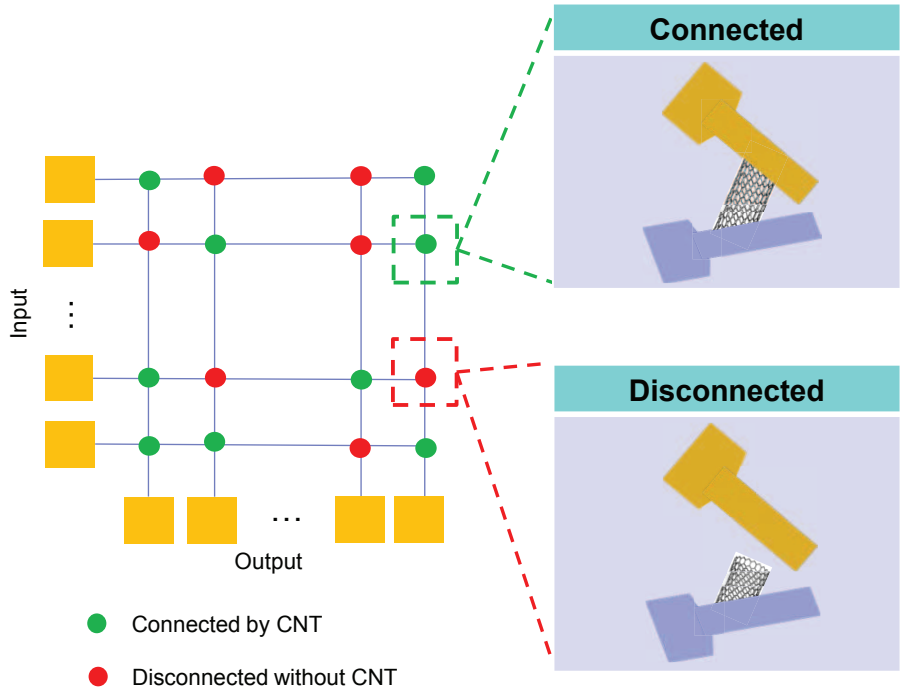



**Figure 4.4:** The schematic of 2D CNT bitarray crossbar structure (adapted from [1]).

from carbon nanotubes. The intersections of the 2D grid are not connected in the first stage. The single CNTs are then grown. If a single CNT connects two wires of the intersection, it is then connected. Otherwise, the intersection is not connected. The connected intersection represents bit 1 and the non-connected intersection represents bit 0. Thus, a random 2D bitarray is generated.

### 4.3.2 The Standard CNT PUF

The fabrication process of carbon nanotubes induce large variations on various features such as diameters, densities and alignments of carbon nanotubes [21, 25]. These

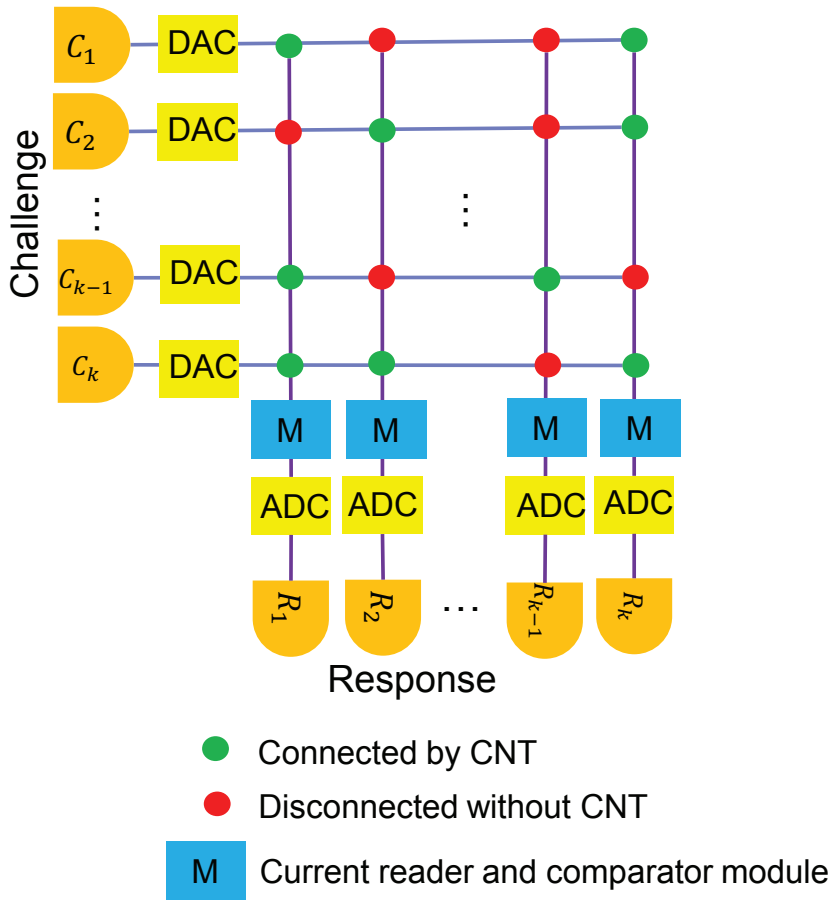



**Figure 4.5:** The CNT crossbar structure based PUF.

variations will affect the electrical performances of carbon nanotube based circuits resulting in large performance variations. These variations can be explored in building highly secure PUFs. In the literature, there are works studying the variation aware CNT based circuit designs including those focused on carbon nanotube field effect transistor (CNFET) [28, 29, 33, 87] and those on bundled single-walled carbon nanotubes (SWCNT) interconnects [21, 35]. In particular, the presence of metallic CNTs in CNFET has been explored to design a Carbon Nanotube PUF in [42] which achieves better reliability against environmental variations.

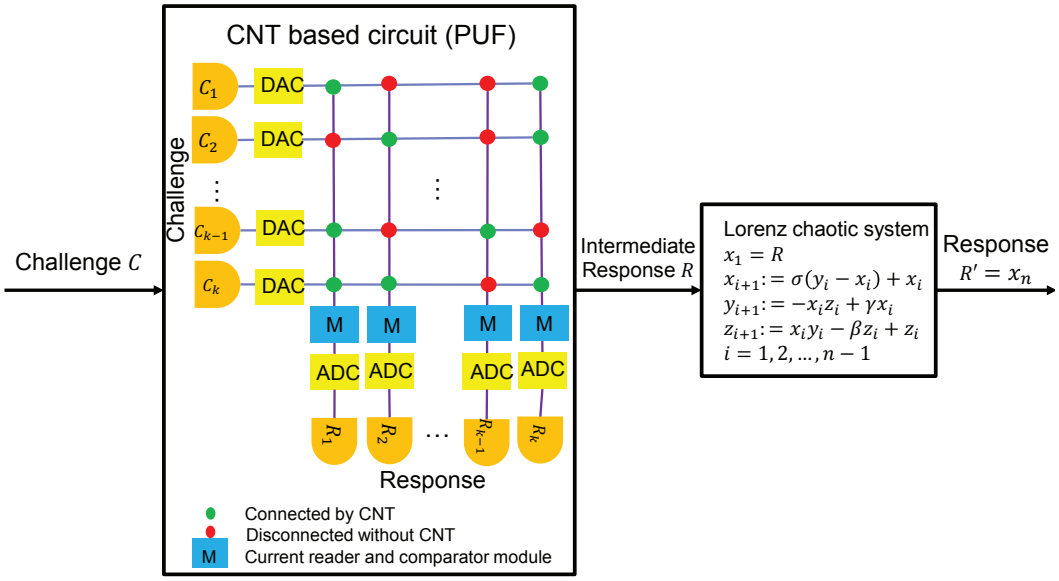



**Figure 4.6:** The proposed Lorenz chaotic system based CNT PUF.

Motivated by the 2D CNT crossbar bitarray design in [1], we design a CNT crossbar structure based PUF by adding the digital-to-analog converters (DAC) at the input, and current measurement module (e.g., [88]), current comparator (e.g., [89]) and analog-to-digital converters (ADC) at the output. Refer to Figure 4.5. The input is a set of binary values as the challenges and is converted to analog signals fed to the CNT PUF. The current of the output analog signal is measured and convert to a value by comparing with a user-defined value. The output is then converted to binary values 1 or 0 as the response.

The advantage of this PUF is from the significant randomness of CNT growing. It is extremely hard to clone the functions and it is resistant to physical attacks. For example, the micro-probing method could easily break down the CNT PUF and the information is destroyed. However, it still could be vulnerable to machine learning

modeling attacks. According to the experimental results shown in Section 4.4, the machine learning modeling attacks can achieve the prediction rate as high as 99% to 100%. This motivates us to design a machine learning modeling attack resistant CNT PUF which is presented in the following section.

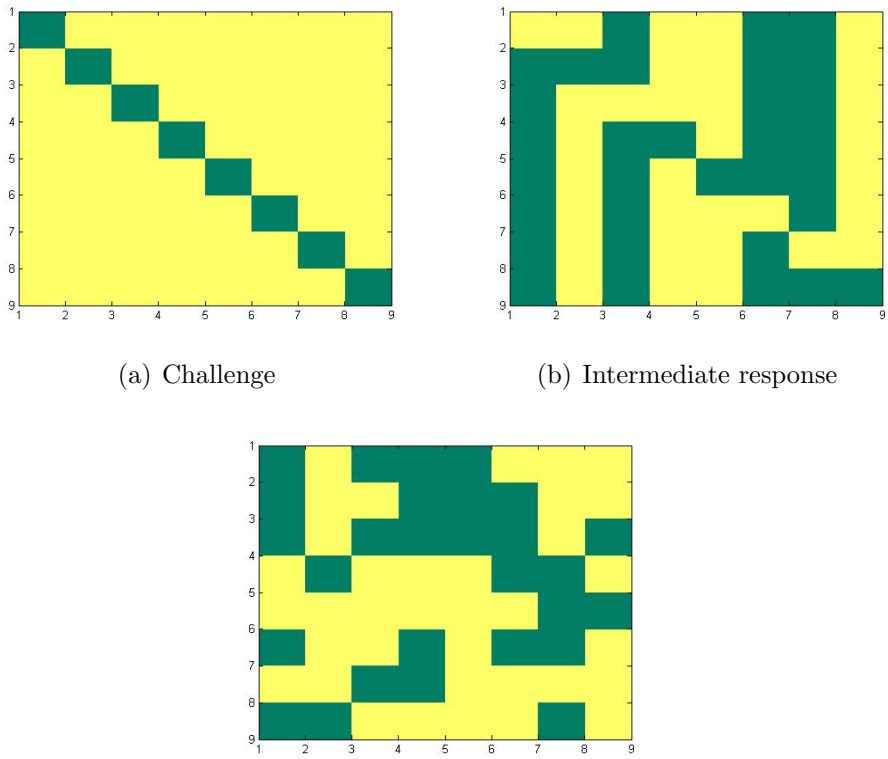
### 4.3.3 Lorenz Chaotic System Based CNT PUF

The proposed Lorenz chaotic system based CNT PUF is illustrated in Figure 4.6. Let $C$ denote the challenge which is the input of the CNT PUF. Let $R$ denote the intermediate response which is the output of the CNT PUF and the input of Lorenz chaotic system. Let $R'$ denote the response which is the output of Lorenz chaotic system. There are two components in the proposed PUF. The first component is the CNT PUF as illustrated in Section 4.3.2. The second component is the discrete Lorenz chaotic system which post-processes the intermediate response. Since discrete Lorenz chaotic system is very sensitive to the small differences of the input $R$, it can generate diverging output $R'$ given similar $R$. Thus, the final responses of similar challenges would not share similarity any more. Comparing to the CNT PUF only, the proposed Lorenz chaotic system based CNT PUF decreases the correlation between the challenges and responses of the CNT PUF, which makes it resistant to machine learning modeling attacks.

The parameters $\{\sigma, \gamma, \beta, y_1, z_1\}$ used in discrete Lorenz chaotic system can be stored in the circuit. However, these parameters could be vulnerable to physical attacks. Thus, to improve the security of the proposed design, the parameters of discrete Lorenz chaotic system are set based on the intermediate response $R$, which can be not revealed by physical attacks. Refer to Eqn. 4.3. Each parameter of discrete Lorenz chaotic system is a function of $R$. For example, the value of $y_1$ can be $0.1R$. After obtaining all the parameters, the response of the whole PUF $R'$ can be calculated using discrete Lorenz chaotic system.

$$\begin{aligned} \sigma &= f_\sigma(R) \\ \gamma &= f_\gamma(R) \\ \beta &= f_\beta(R) \\ x_1 &= R \\ y_1 &= f_y(R) \\ z_1 &= f_z(R). \end{aligned} \tag{4.3}$$

The illustrations of challenge and response of the CNT PUF without Lorenz chaotic system (Figure 4.5) and the proposed Lorenz chaotic system based CNT PUF (Figure 4.6) are shown in Figure 4.7 and Figure 4.8, respectively. Figure 4.7 (a) shows a set of 8-bit challenges and each row represents one challenge $C$. Figure 4.7 (b) shows the responses of the CNT PUF and each row represents one intermediate response $R$.



(a) Challenge

(b) Intermediate response

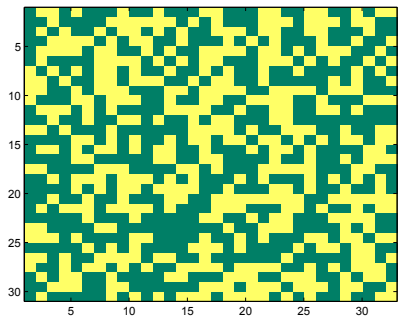
(c) Response

**Figure 4.7:** Illustration of 8-bit challenge, intermediate response and response of the proposed PUF.

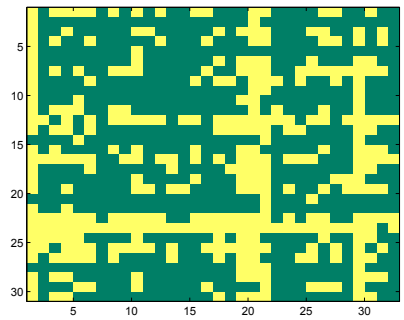
The set of final responses $R'$ of the proposed CNT PUF is shown in Figure 4.7 (c). It is observed that the proposed PUF can generate more diverse responses than the CNT PUF. The metric of similarity is defined by the average of Hamming Distance (HD) between each response with the other responses. Suppose that there are $M$ responses. The HD between one response $R^m$ with the other responses is calculated as follows:

$$HD^m = \sum_h |R^m - R^h|, \tag{4.4}$$

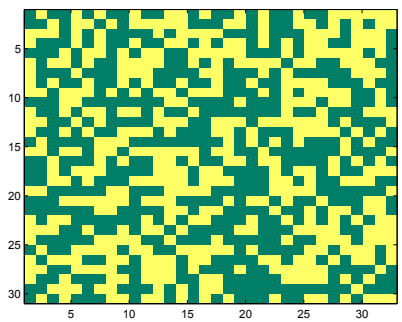
where $h = 1, 2, ..., M$ but $h \neq m$. Thus, the metric of similarity is calculated as




(a) Challenge




(b) Intermediate response




(c) Response

**Figure 4.8:** Illustration of 32-bit challenge, intermediate response and response of the proposed PUF.

follows:

$$Sim = \frac{1}{M} \sum_{m=1}^{M} HD^m, \tag{4.5}$$

The similarity of responses in Figure 4.7 (b) is 2 and that of responses in Figure 4.7 (c) is 4.375. The smaller similarity means the responses are more similar to each other. Therefore, the proposed PUF with Lorenz chaotic system can generate more diverse responses than the CNT PUF without Lorenz chaotic system.

Another example is shown in Figure 4.8. A set of 32-bit challenges are randomly

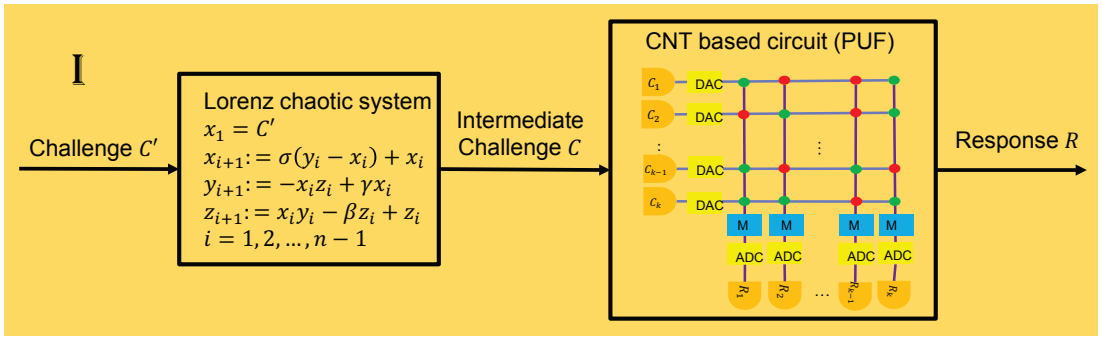
generated as shown in Figure 4.8 (a). The intermediate response $R$ and final response $R'$ are shown in Figure 4.8 (b) and (c), respectively. It can be observed that the patterns of $R'$ seems more random than that of $R$, which is desirable. The metric of randomness is defined by the average of the ratio of 0 or 1 in the response whichever is larger. Suppose that there are $M$ responses. The randomness is calculated as follows:

$$Ran = \frac{1}{M}\sum_{m=1}^{M} max\{p_0^m, p_1^m\} \times 100\%, \tag{4.6}$$

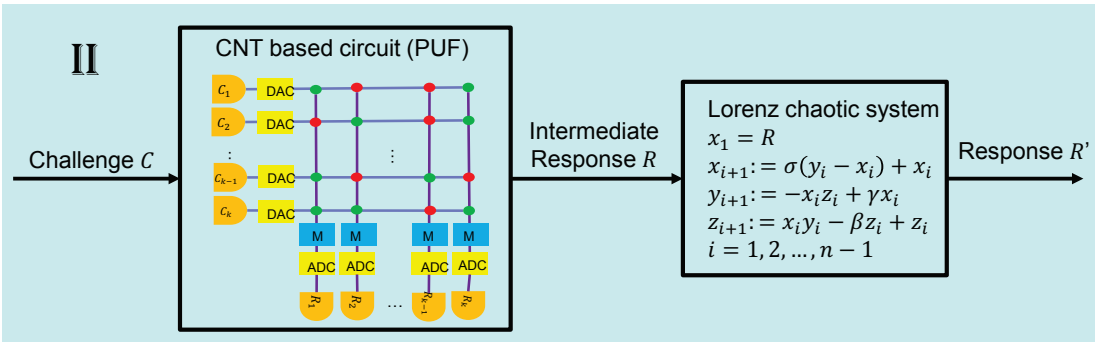
where $p_0^m$ is the ratio of 0 in response $R_m$ and $p_1^m$ is the ratio of 1 in response $R_m$. The ideal randomness is 50% and the worst is 100%. The randomness of responses in Figure 4.8 (b) is 73.23% and the randomness of responses in Figure 4.8 (c) is 56.77%. Therefore, the proposed PUF with Lorenz chaotic system can generate more random responses than the CNT PUF without Lorenz chaotic system given random challenges.

### 4.3.4 Two Possible Design Styles

One might wonder whether the two components in the proposed PUF, as shown in Figure 4.6, can be switched. Refer to Figure 4.9. The structures of two possible design styles are illustrated. In Design I, Lorenz chaotic system is placed before CNT PUF as shown in Figure 4.9 (a). Let $C'$ denote the challenge which is the input of




(a) Design I




(b) Design II

**Figure 4.9:** Structures of two possible designs (Design I: Lorenz chaotic system is placed before CNT PUF; Design II: Lorenz chaotic system is placed after CNT PUF. For consistency, input of CNT PUF is always denoted by $C$ and output of CNT PUF is always denoted by $R$).

Lorenz chaotic system. Let $C$ denote the intermediate challenge which is the output of Lorenz chaotic system and the input of the CNT PUF. Let $R$ denote the response which is the output of the CNT PUF. In Design II, Lorenz chaotic system is placed after CNT PUF as shown in Figure 4.9 (b). Let $C$ denote the challenge which is the input of the CNT PUF. Let $R$ denote the intermediate response which is the output of the CNT PUF and the input of Lorenz chaotic system. Let $R'$ denote the response which is the output of Lorenz chaotic system. To compare them, let us analyze two cases: (1) Lorenz chaotic system is not vulnerable to attacks and (2) Lorenz chaotic

system is vulnerable. The values of $(C', R)$ in Design I and $(C, R')$ in Design II are available to the attacker. There is an assumption that CNT PUF can be hacked if the challenges $C$ and responses $R$ are known, which is demonstrated in Section 4.5.

† Let us consider the first case that Lorenz chaotic system is not vulnerable to any attacks which means that the parameters of Lorenz chaotic system cannot be revealed. In Design I, $C$ cannot be calculated and CNT PUF cannot be hacked. Thus Design I is resistant to machine learning modeling attacks. In Design II, $R$ cannot be calculated and CNT PUF cannot be hacked. Thus, Design II is also resistant to machine learning modeling attacks. In this case, Design I and Design II have same performance against modeling attack methods. However, one could not ensure that there exist no effective attacking methods for Lorenz chaotic system.

† Suppose that Lorenz chaotic system is vulnerable to attacks which means that the parameters can be revealed and are available to the attacker. In Design I, given $C'$, $C$ can be calculated using the parameters of Lorenz chaotic system. In this scenario, $C$ and $R$ of CNT PUF are known to the attacker and the CNT PUF can be modeled using machine learning methods, such as logistic regression which achieves 100% accuracy (as demonstrated in Section 4.5). Therefore, Design I is vulnerable to machine learning modeling attacks. In Design II, given the response $R'$ and the parameters of Lorenz chaotic system, it could

still be extremely hard to solve Lorenz chaotic system and obtain $R$ due to the nature of Lorenz chaotic system. In this scenario, $C$ is known but $R$ is unknown, thus CNT PUF cannot be hacked. Therefore, Design II is resistant to machine learning modeling attacks and it is chosen in this paper.

One might also consider that whether one can only use Lorenz chaotic system itself in the design without CNT PUF. In this case, the parameters of Lorenz chaotic system need to be induced from fabrication variations, which could be complicated. Therefore, one needs to combine CNT PUF followed by Lorenz chaotic system as discussed before.

### 4.3.5 Hierarchical CNT PUF

In some security design applications, large number of bits are needed, such as 256-bit and 512-bit. Therefore, a hierarchical idea is explored and the 64-bit PUF is illustrated in Figure 4.10 by cascading two 32-bit PUFs. First the 64-bit challenge is divided to two 32-bit challenges which are the inputs of two 32-bit Lorenz chaotic system CNT PUFs. The responses of the two 32-bit PUFs are then combined to generate the final 64-bit response. Similarly, 128-bit PUF can be generated by two 64-bit PUFs. Using the hierarchical structure, PUFs with large number of bits becomes possible.

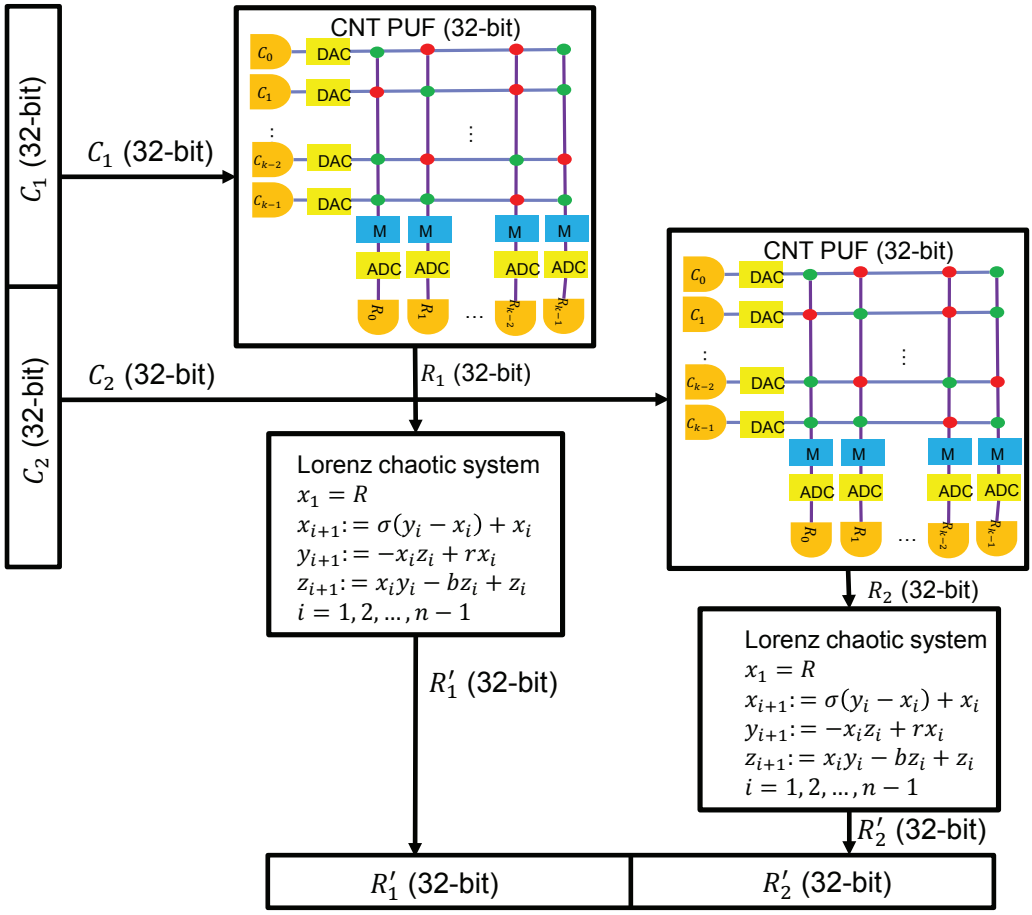



**Figure 4.10:** The hierarchical structure Lorenz chaotic system based CNT PUF.

## 4.4 Machine Learning Modeling Attack Methods

To demonstrate the security performance of the proposed CNT PUF against machine learning modeling attacks, various methods are preformed to the proposed Lorenz chaotic system based CNT PUF and the CNT PUF without Lorenz chaotic system. In the following, the two PUFs are denoted by the CNT PUF w/ Lorenz and the CNT PUF w/o Lorenz. The training data are $M$ challenge-response pairs for both

designs. Machine learning modeling methods are performed to estimate the models of the CNT PUF w/o Lorenz and the CNT PUF w/ Lorenz. Subsequently, the models will be used to predict the responses given the new challenges in the test data. In this paper, SVM, LR, DL and ES machine learning methods are used to evaluate the two PUF designs. These machine learning methods are widely used to attack PUFs in the literature [46, 90, 91, 92].

The security performance of the PUF design is measured by the bit-wise prediction rate of the response. For example, there are 1,000 challenge-response pairs in the test data, and the length of the response is 32 bits. Thus, the total bits of the responses is $1,000 \times 32 = 32,000$. Suppose that 30,000 bits are estimated correctly, then the bit-wise prediction rate is calculated by $\frac{30,000}{32,000} = 93.75\%$. Four machine learning methods are considered in this paper.

### 4.4.1 Support Vector Machine

Support Vector Machine (SVM) has been widely used to attack PUFs. In [90], SVM is used to model an arbiter PUF, and the results show that SVM reaches over 90% prediction rate using 20,000 challenge-response pairs. In this paper, SVM is used to evaluate the security performance of the CNT PUF w/o Lorenz and the CNT PUF w/ Lorenz. Denote the challenge-response pairs by $\{(C^1, R^1), (C^2, R^2), ..., (C^M, R^M))\}$ in

the training data and let $m$ denote the index of the challenge-response pairs. Each challenge is denoted by $C^m = \{C_1^m, C_2^m, ...C_n^m, ..., C_N^m\}$ and each response is denoted by $R^m = \{R_1^m, R_2^m, ...R_n^m, ..., R_N^m\}$, where $n$ is the index of the bit in the challenge and response, and $N$ is the total number of bits. For example $C_2^1$ is the second bit in the first challenge $C^1$.

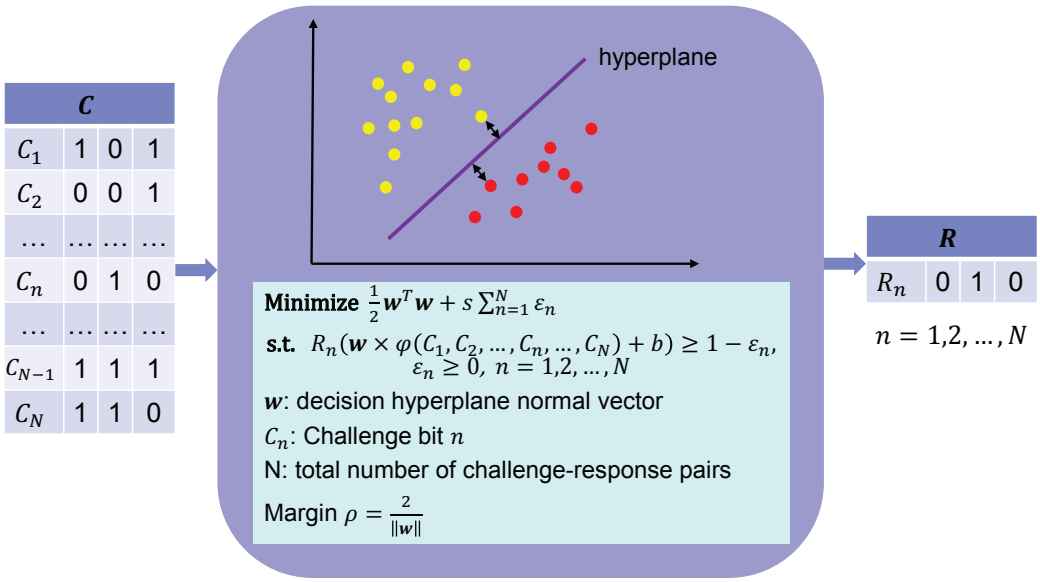



**Figure 4.11:** The SVM model between one response bit $R_n$ and all challenge bits.

Refer to Figure 4.11. An SVM model is estimated for each response bit $R_n$ considering all challenge bits $\{C_1, C_2, ..., C_n, ..., C_N\}$. The response bits can be classified to two classes, where $R_n = 0$ and $R_n = 1$. Each challenge $\{C_1, C_2, ..., C_n, ..., C_N\}$ and can be mapped to a high-dimensional feature space. SVM method finds a separating hyperplane that maximizes gap between the classes. The SVM model solves a quadratic programming problem as follows [44]:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\omega^T\omega + s\sum_{n=1}^{N}\epsilon_n \\
\text{subject to} \quad & R_n(\omega \times \varphi(C_n) + b) \geq 1 - \epsilon_n, \\
& \epsilon_n \geq 0, \\
& n = 1, 2, ..., N,
\end{aligned}
\tag{4.7}
$$

where $\omega$ is a normal vector of the hyperplane, $b$ is a bias, $\varphi(C_n)$ is the non-linear mapping function, $\epsilon_n$ is the error in the misclassification and $s$ is a regularization constant. This is a convex quadratic programming optimization problem. The computed SVM model is applied to the testing data and the predicted responses can be estimated. Subsequently, the bit-wise prediction rate can be calculated comparing the predicted responses with the true responses.

### 4.4.2 Logistic Regression

Logistic Regression (LR) is a most widely used machine learning modeling attack method for PUFs. In [46], LR is used to attack an arbiter PUF and an XOR arbiter PUF where the results show that LR can reach as high as 99% prediction rate for 64-bit PUFs using around 10,000 challenge-response pairs. Refer to Figure 4.12. A logistic regression model is computed between one response bit $R_n$ and all challenge bits $\{C_1, C_2, ...C_n, ..., C_N\}$. The logistic function is as follows [44]:

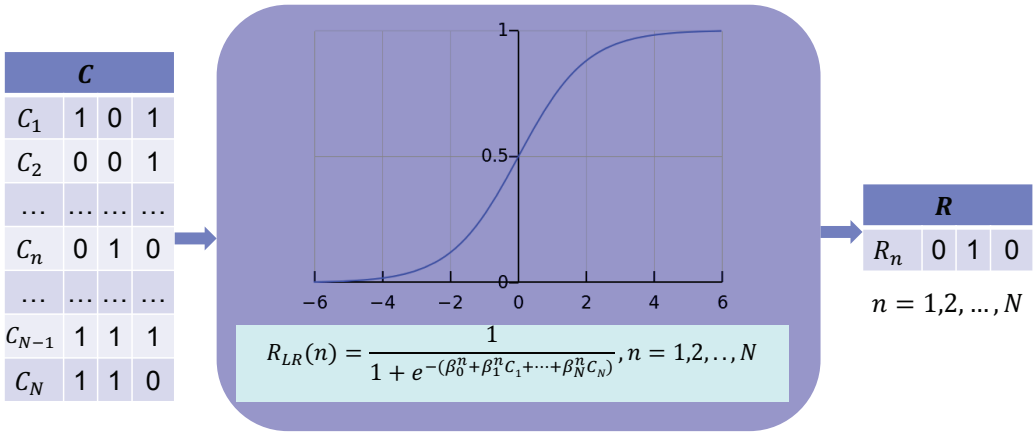



**Figure 4.12:** The logistic regression model between one response bit $R_n$ and all challenge bits.

$$R_{LR}(n) = \frac{1}{1 + e^{-(\beta_0^n + \beta_1^n \cdot C_1 + \beta_2^n \cdot C_2 + \cdots + \beta_N^n \cdot C_N)}}, \tag{4.8}$$

where $\beta_0^n, \beta_1^n, ..., \beta_N^n$ are the parameters of the LR model for $R_n$. The computed LR models $R_{LR}(n)$ where $n = 1, 2, ..., N$ are then applied to the challenges in the test data and the estimated responses can be obtained. If $R_{LR}(n) >= 0.5$, the predicted response bit is set to 1 and 0 otherwise. Comparing the predicted responses with the true responses, the bit-wise prediction rate can be calculated.

### 4.4.3 Deep Learning

Deep Learning (DL) is a popular machine learning method and deep belief networks (DBN) is one of popular DL methods. DBN has an input layer, multiple hidden layers and an output layer, which consist of artificial neurons imitating biological neurons.

In [91], DL is used to attack an arbiter PUF, where the results show that DL reaches prediction rate of 58% for 50,000 challenge-response pairs. In this paper, DBN is used to model the PUFs where DBN is composed of multiple layers of restricted Boltzmann machines (RBM).

#### 4.4.3.1 Restricted Boltzmann Machines

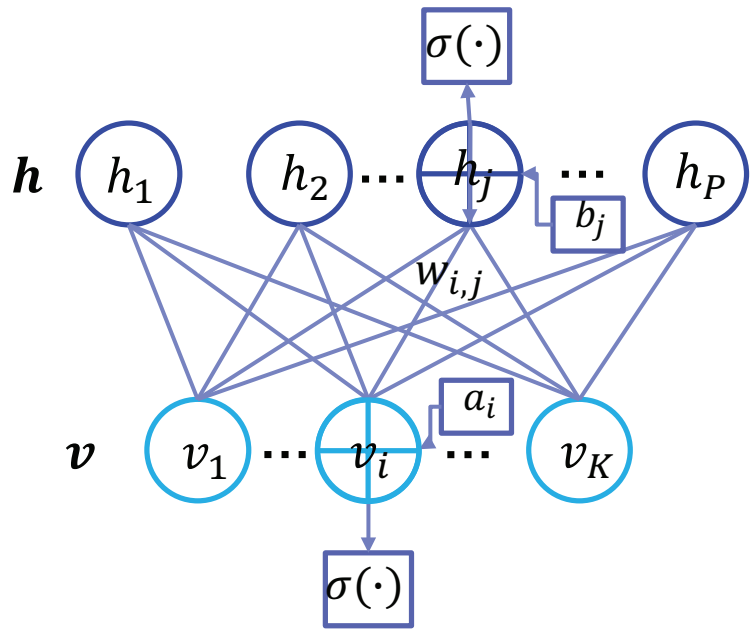



**Figure 4.13:** The illustration of an RBM with hidden units and visible units.

A restricted Boltzmann machines (RBM) is a generative model including hidden units

and visible units. Refer to Figure 4.13. The energy function of the state $\{v, h\}$ is [93]

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j, \tag{4.9}$$

where $a_i$ is bias for $v_i$, $b_j$ is bias for $h_j$, $w_{i,j}$ is weight on the edge linking between $v_i$ and $h_j$. The joint probability distribution is defined as

$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_v \sum_h e^{-E(v,h)}}. \tag{4.10}$$

The probability of hidden unit $h_j$ setting to 1 is

$$P(h_j = 1|v) = \frac{1}{1 + e^{-(\sum_i w_{i,j} v_i + b_j)}}. \tag{4.11}$$

The probability of visible unit $v_i$ setting to 1 is

$$P(v_i = 1|h) = \frac{1}{1 + e^{-(\sum_j w_{i,j} h_j + a_i)}}. \tag{4.12}$$

Given training data, the RBM model parameters $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{w}\}$ can be estimated using a fast algorithm contrastive divergence proposed in [94].

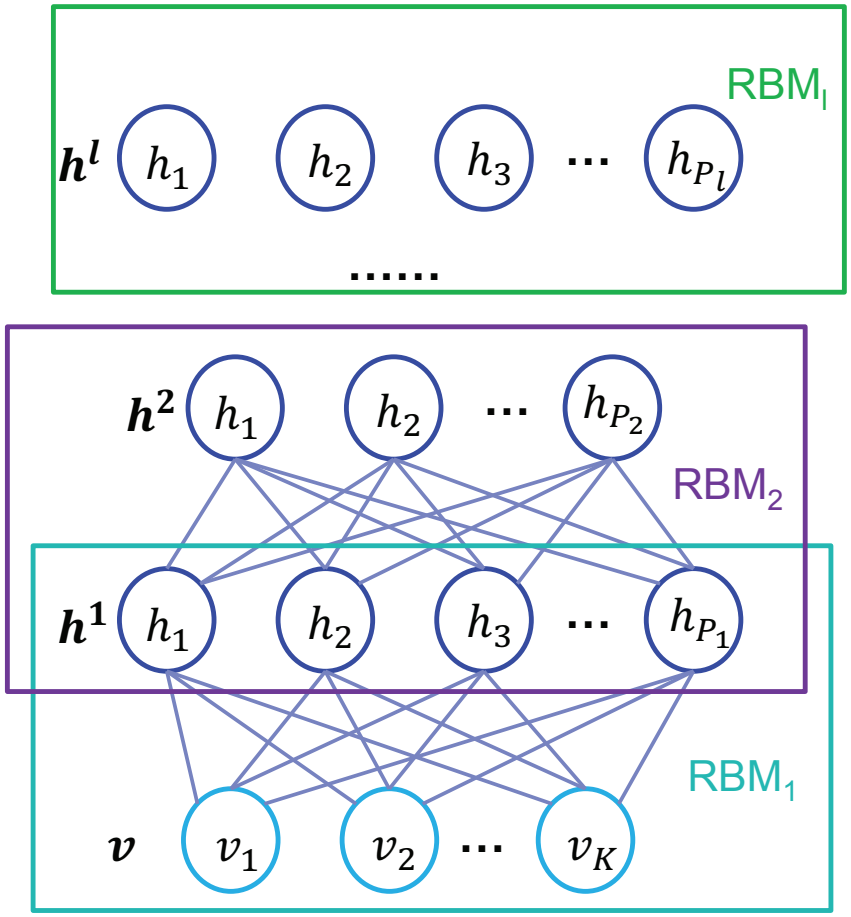



**Figure 4.14:** The illustration of a DBN model.

#### 4.4.3.2 Deep Belief Networks

A deep belief network (DBN) is a probabilistic generative model with many hidden layers, where each hidden layer corresponds to an RBM [95]. Refer to Figure 4.14. There are $l$ hidden layers in the DBN model. The joint probability distribution over

visible units and hidden units is

$$P(v, h^1, ..., h^l) = P(v|h^1)P(h^1|h^2)\dots P(h^{l-1}|h^l). \tag{4.13}$$

The dependency between two adjacent hidden layers is computed as below:

$$P(h_j^{l-1} = 1|h^l) = \frac{1}{1 + e^{-(\sum_i w_{i,j}^l h_i^l + b_j^l)}}, \tag{4.14}$$

where $\boldsymbol{b^l}$ is bias vector and $\boldsymbol{w^l}$ is weight matrix. A heuristic learning algorithm given in [95] is used to train the DBN sequentially through computing RBM in a layer-by-layer fashion.

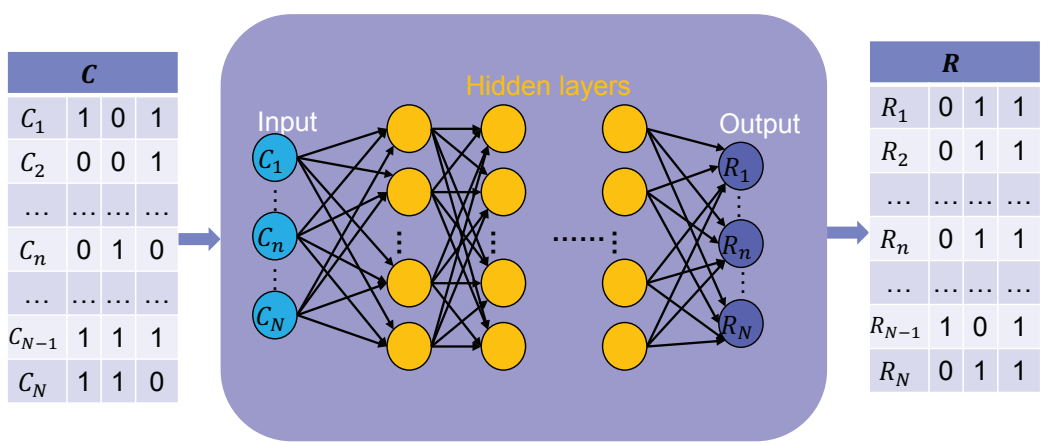



**Figure 4.15:** The DBN model between all response bits and all challenge bits.

Refer to Figure 4.15. In our case, the challenge $\{C_1, C_2, ..., C_N\}$ can be used as the set of visible units in the first RBM layer, and the values of output layer units are

set to be the response $\{R_1, R_2, ..., R_N\}$. The computed DBN model is then applied to the testing data to estimate responses. Subsequently, the bit-wise prediction rate can be obtained comparing the estimated responses with the true responses.

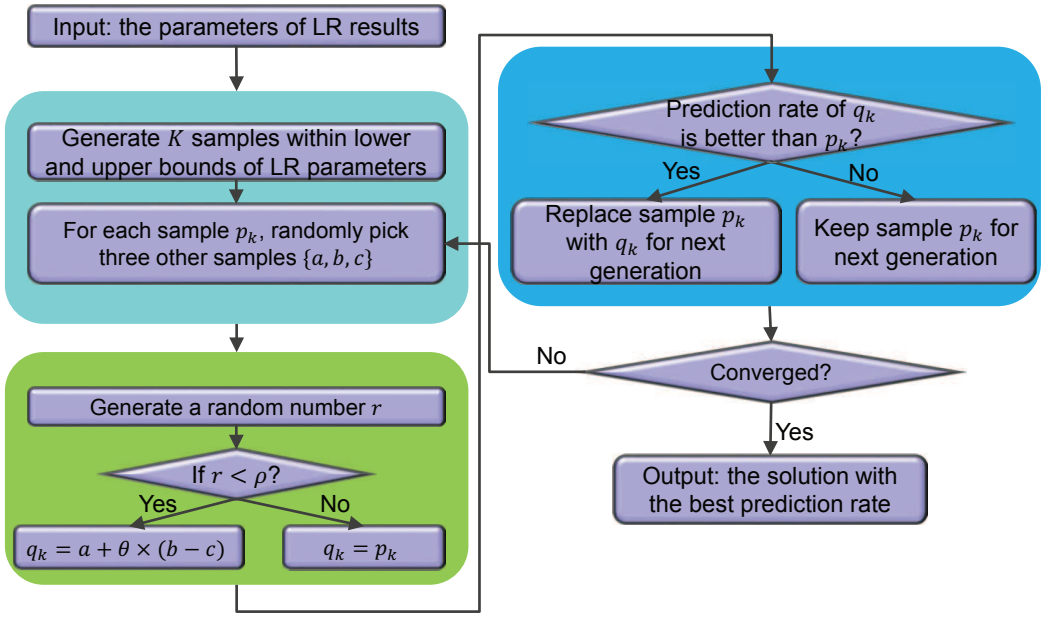



**Figure 4.16:** The Logistic Regression integrated Evolution Strategies method.

### 4.4.4 Evolution Strategy

In [92], the authors propose an evolution strategy (ES) based machine learning method to attack arbiter PUFs. In this paper, the ES method is developed through exploring LR results. A PUF model instance can be represented using LR parameters in Eqn. 4.8. The main idea of the ES machine learning method is to generate

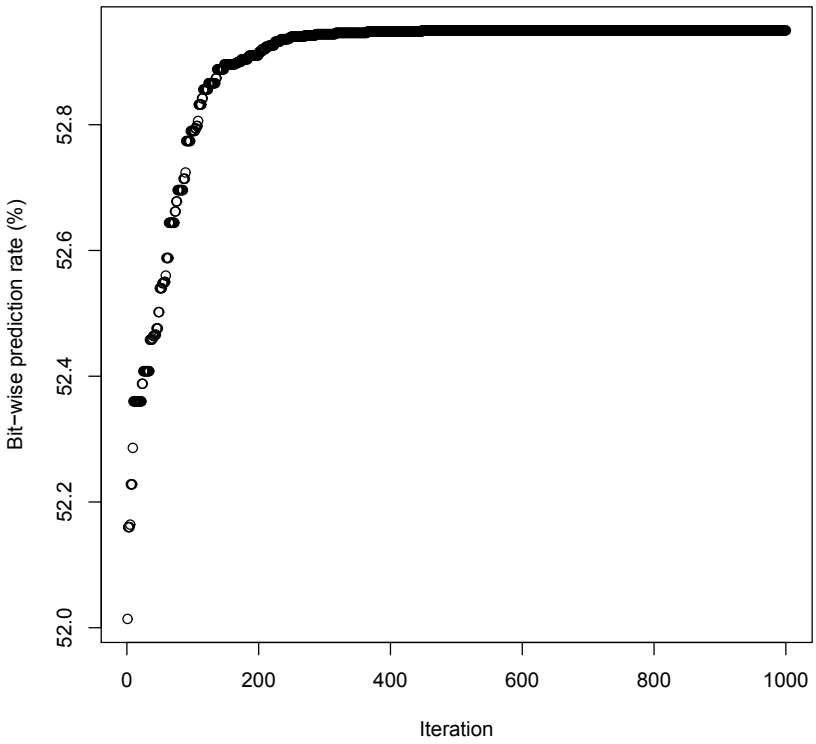



**Figure 4.17:** The bit-wise prediction rate over 1000 iterations of ES method.

random PUF instances and pick the ones which best model the real PUF. In other words, the PUF instances which provide highest prediction rates are survived and kept as ancestors for the next generation. In the next generation, descendants of PUF instances are generated using that of the ancestors together with some random mutations. The descendants of PUF instances which provide highest prediction rates are survived and kept as ancestors for the next generation. This process is repeatedly performed until convergence.

In this paper, a popular ES method, differential evolution is used to evaluate the

security performance of the proposed CNT PUF. The algorithmic flow is shown in Figure 4.16. The inputs are the parameters of the LR model $\{\beta_0, \beta_1, ..., \beta_N\}$. First, a set of samples are generated within the given lower and upper bound of the LR parameters, where a sample is defined as a candidate of parameters $\{\beta_0, \beta_1, ..., \beta_N\}$. The lower bound and upper bound are as follows:

$$\begin{aligned} \beta^l &= \{\beta_0 - \alpha_0|\beta_0|, \beta_1 - \alpha_1|\beta_1|, ..., \beta_N - \alpha_N|\beta_N|\}, \\ \beta^u &= \{\beta_0 + \alpha_0|\beta_0|, \beta_1 + \alpha_1|\beta_1|, ..., \beta_N + \alpha_N|\beta_N|\} \end{aligned} \tag{4.15}$$

For example, if $\beta_0 = 10$ and $\alpha_0 = 0.2$, the range of $\beta_0$ is $[8, 12]$.

Let $K$ denote the number of samples and $k$ denote the index of a sample. The sample update policy works as follows. A control parameter $\rho$ is used to control whether the sample $p_k$ is updated. A random number is first generated. If it is smaller than $\rho$, the new sample $q_k$ is calculated as follows:

$$q_k = a + \theta \times (b - c), \tag{4.16}$$

where $\theta$ is a user defined differential weight and $a, b, c$ are three other distinct samples. Using the new sample $q_k$, the prediction rate can be calculated. If the prediction rate using $q_k$ is better than $p_k$, $p_k$ will then be replaced by $q_k$ in the next generation. Otherwise, $p_k$ will be kept in the next generation. If the random generated number is larger than $\rho$, the sample $p_k$ will be kept in the next generation. The above procedure

is repeated until convergence.

Refer to Figure 4.17. ES method is performed for 1,000 iterations. The bit-wise prediction rate of ES method increases from 51.69% to 52.95%, where 51.69% is the prediction rate of LR method. It can be observed that the bit-wise prediction rate is non-decreasing over iterations. The reason is that in each generation, the better samples (at least the same samples) will be propagated to the next generation.

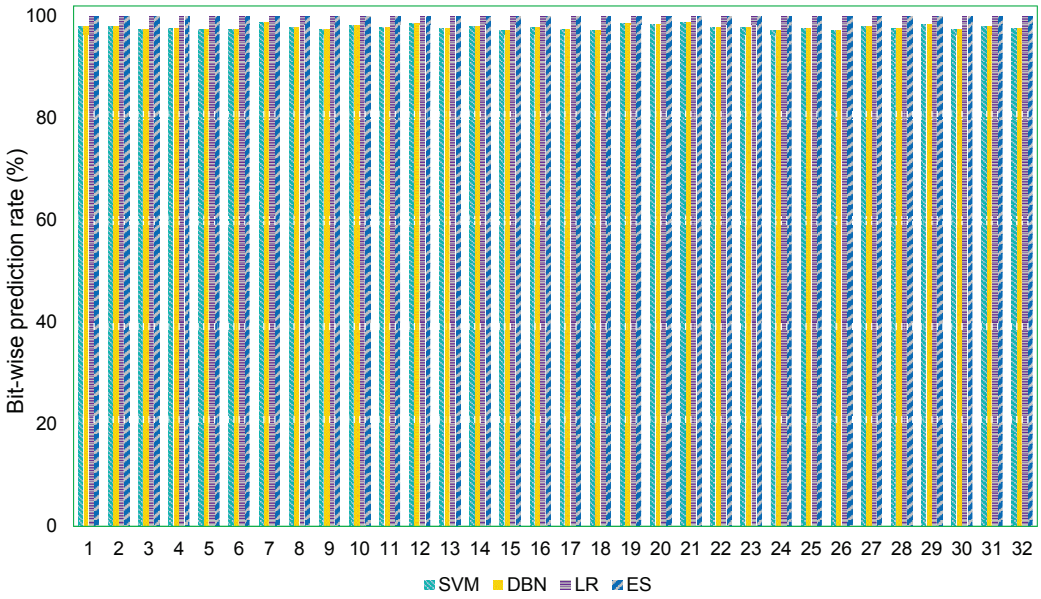



**Figure 4.18:** The bit-wise prediction rates of each bit on the CNT PUF w/o Lorenz.

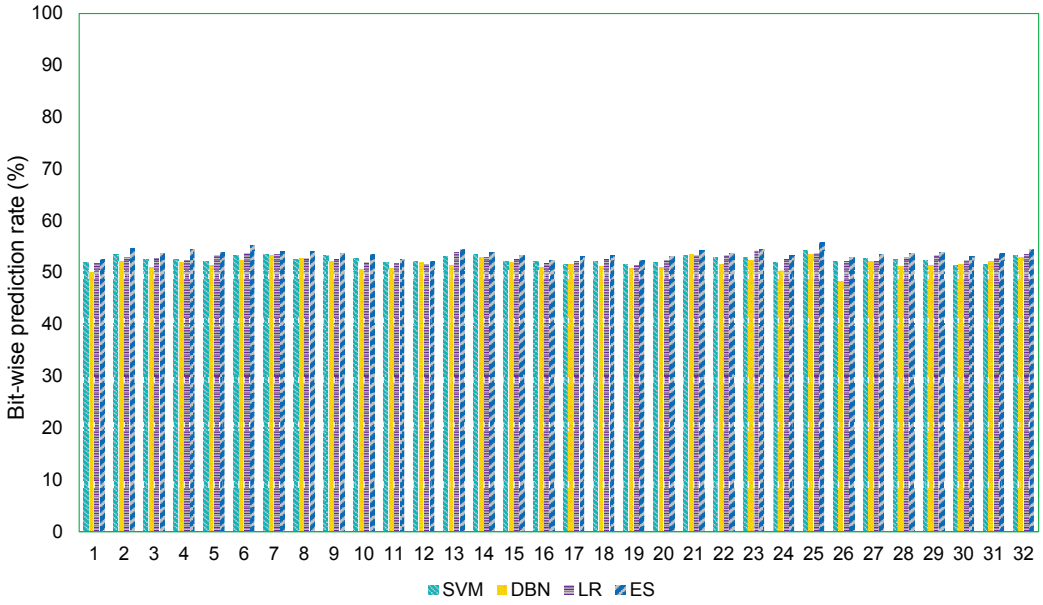
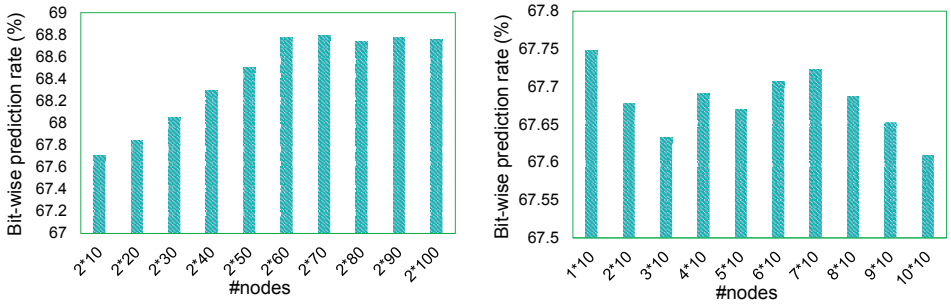



**Figure 4.19:** The bit-wise prediction rates of each bit on the CNT PUF w/ Lorenz.




(a) DBN with two hidden layers (b) DBN with more hidden layers

**Figure 4.20:** The bit-wise prediction rates of DBN with different parameters.

## 4.5 Experimental Results

### 4.5.1 Experimental Setup

The Lorenz chaotic system based CNT PUF (CNT PUF w/ Lorenz) and the CNT PUF only (CNT PUF w/o Lorenz) are simulated on a computer with 3.40GHz Intel Pentium CPU and 3GB memory. Two sets of experiments are designed. One is with 32-bit challenges and responses and the other is with 64-bit challenges and responses. Similar to existing works of machine learning modeling attacks on PUF [46], the number of training challenge-response pairs is set to 50,000. The number of testing challenge-response pairs is also set to 50,000. Support Vector Machine (SVM), Deep Belief Network (DBN), Logistic Regression (LR) and Evolution Strategies (ES) machine learning methods are implemented using R language and tested on a computer with 3.40GHz Intel Pentium CPU and 3GB memory. The R program packages used for SVM and DBN can be downloaded from [96, 97]. According to our observations, the maximum number of iterations in ES methods is empirically set to be 1000, which is enough for convergence. In addition, cross validation is considered in the implementations.

**Table 4.1**
Bit-wise prediction rates of Support Vector Machine (SVM), Deep Belief Network (DBN), Logistic Regression (LR) and Evolution Strategies (ES) for CNT PUF w/o Lorenz chaotic system and the CNT PUF w/ Lorenz chaotic system using 50,000 32-bit and 64-bit challenge-response pairs training data.

| | 32-bit | | 64-bit | |
|---|---|---|---|---|
| | CNT PUF w/o Lorenz | CNT PUF w/ Lorenz | CNT PUF w/o Lorenz | CNT PUF w/ Lorenz |
| SVM | 92.59% | 52.55% | 92.15% | 53.11% |
| DBN | 97.80% | 51.66% | 95.00% | 52.97% |
| LR | 100.00% | 52.68% | 100.00% | 53.45% |
| ES | 100.00% | 53.65% | 100.00% | 54.46% |

**Table 4.2**
Runtime of SVM, DBN, LR and ES for 64-bit CNT PUF w/o Lorenz chaotic system and CNT PUF w/ Lorenz chaotic system using 50,000 challenge-response pairs training data.

| | CNT PUF w/o Lorenz | | CNT PUF w/ Lorenz | |
|---|---|---|---|---|
| | Prediction rate | Runtime (s) | Prediction rate | Runtime (s) |
| SVM | 92.15% | 7813.98 | 53.11% | 6889.22 |
| DBN | 95.00% | 45.98 | 52.97% | 41.21 |
| LR | 100.00% | 1289.36 | 53.45% | 249.15 |
| ES | 100.00% | 1289.36 | 54.46% | 21135.12 |

### 4.5.2 Experimental Results

In this paper, two criteria are used to evaluate the security performance of the PUF designs. One is the bit-wise prediction rate, which represents the accuracy rate in term of bits. For example, there are 1,000 challenge-response pairs in the test data, and the length of the response is 32 bits. The total bits to predict is $1,000 \times 32 = 32,000$ bits.

**Table 4.3**
String-wise prediction rates of SVM, DBN, LR and ES for CNT PUF w/o Lorenz chaotic system and the CNT PUF w/ Lorenz chaotic system using 50,000 challenge-response pairs training data.

| | #allowed error bit | 32-bit | | 64-bit | |
|---|---|---|---|---|---|
| | | CNT PUF w/o Lorenz | CNT PUF w/ Lorenz | CNT PUF w/o Lorenz | CNT PUF w/ Lorenz |
| SVM | 0bit | 18.49% | 0.01% | 4.10% | 0.00% |
| | 1bit | 42.74% | 0.05% | 13.66% | 0.00% |
| DBN | 0bit | 52.35% | 0.00% | 8.98% | 0.01% |
| | 1bit | 83.35% | 0.00% | 27.90% | 0.04% |
| LR | 0bit | 100.00% | 0.00% | 100.00% | 0.05% |
| | 1bit | 100.00% | 0.00% | 100.00% | 0.10% |
| ES | 0bit | 100.00% | 0.00% | 100.00% | 0.00% |
| | 1bit | 100.00% | 0.00% | 100.00% | 0.02% |

Suppose that there are 30,000 bits are predicted correctly, the bit-wise prediction rate is calculated by $\frac{30,000}{32,000} = 93.75\%$. The other criteria is the string-wise prediction rate, which represents the accuracy rate in term of strings. For example, there are 1,000 challenge-response pairs in the test data, and there are 900 responses are predicted correctly. The string-wise prediction rate is $\frac{900}{1,000} = 90\%$. The number of allowed error bit is also considered. It represents the maximum number of allowed incorrect bits in the response. For example, the response is 32 bits and the number of allowed error bits is 2. It means that if the predicted response contains less than or equal to 2 incorrectly estimated bits, it is still treated as correct prediction. Clearly, 0 allowed error bit means the standard string-wise prediction rate.

The bit-wise prediction rates of two PUFs designs over 32 bits of the four popular machine learning modeling methods are shown in Table 4.1. The bit-wise prediction

rates of each bit for the PUF w/o Lorenz and the PUF w/ Lorenz are shown in Figure 4.18 and Figure 4.19. The string-wise prediction rates of two PUFs designs over 32 bits of the four popular machine learning modeling methods are shown in Table 4.3. We make the following observations.

- † Refer to Table 4.1. For 32-bit CNT PUF w/o Lorenz, all four methods obtain high bit-wise prediction rates. In particular, LR and ES achieve 100.00% bit-wise prediction rates. Therefore, the CNT PUF w/o Lorenz is vulnerable to machine learning modeling attacks.

- † Refer to Table 4.3. The string-wise prediction rate of SVM is the lowest, and that of LR and ES are 100.00%, which is the highest. DBN achieves higher string-wise prediction rates than SVM but lower than that of LR and ES. Therefore, LR and ES have the best performance for attacking 32-bit CNT PUF w/o Lorenz. Again, the CNT PUF w/o Lorenz is vulnerable to machine learning modeling attacks.

- † Refer to Table 4.1. For 32-bit CNT PUF w/ Lorenz, all four methods obtain very low bit-wise prediction rates, around 50%. Since in theory, the random guess method could obtain 50% bit-wise prediction rates. Therefore, the CNT PUF w/ Lorenz is resistant to machine learning modeling attacks.

- † Refer to Table 4.3. The string-wise prediction rates of all four methods are nearly zero. Again, the CNT PUF w/ Lorenz is resistant to machine learning

modeling attacks.

† Refer to Figure 4.18, the four methods obtain high bit-wise prediction rates for each bit in the 32-bit CNT PUF w/o Lorenz. Refer to Figure 4.19, the four methods obtain low bit-wise prediction rates for each bit in the 32-bit CNT PUF w/ Lorenz.

† The above conclusions are reasonable under our models and our parameters. It does not extend to other PUFs and there might be other better machine learning methods and other better parameters which could obtain better performance.

The bit-wise prediction rates of two PUFs designs over 64 bits of the four popular machine learning modeling methods are shown in Table 4.1 and the runtime information in shown in Table 4.2. The string-wise prediction rates of two PUFs designs over 64 bits of the four popular machine learning modeling methods are shown in Table 4.3. We make the following observations.

† Refer to Table 4.1. For 64-bit CNT PUF w/o Lorenz, all four methods obtain high bit-wise prediction rates. In particularly, LR and ES achieve 100% bit-wise prediction rates. Therefore, the CNT PUF w/o Lorenz is vulnerable to machine learning modeling attacks.

† Refer to Table 4.3. The string-wise prediction rates of SVM and DBN are very low, and that of LR and ES are 100%, which is the highest. Therefore, LR

and ES have the best performance for attacking 64-bit CNT PUF w/o Lorenz. Again, the CNT PUF w/o Lorenz is vulnerable to machine learning modeling attacks.

† Refer to Table 4.1 and Table 4.3. For 64-bit CNT PUF w/ Lorenz, all four methods obtain very low bit-wise prediction rates, around 70%. The string-wise prediction rates of all four methods are also very low. Therefore, the CNT PUF w/ Lorenz is resistant to machine learning modeling attacks.

† Refer to Table 4.2. DBN is the most efficient and the reason is that there is only one DBN model is used to estimate the PUF. All other methods use 64 models to estimate the PUF, therefore they are less efficient than DBN.

† Smaller prediction rates mean that it is harder to predict the correct response. For example, the bit-wise prediction rate is 95.00% under DBN model for CNT PUF w/o Lorenz. It means that $64 \times 95.00\% = 60.8 \approx 60$ bits can be predicted correctly. Then the attacker needs to guess the other 4 bits, where there exist $2^4 = 16$ possibilities. Consider another example the bit-wise prediction rate is 52.97% under DBN model for CNT PUF w/ Lorenz. It means that $64 \times 52.97\% = 33.9 \approx 33$ bits can be predicted correctly. Then the attacker needs to guess the other 31 bits, where there exist $2^{31} = 2,147,483,648$ possibilities.

Two testcases are designed to study the performance of DBN. Refer to Figure 4.20 (a). There are two hidden layers and the number of nodes within each layer ranges

from 10 to 100. It can be observed that, within certain number of nodes, the bit-wise prediction rate is increasing with more nodes in each layer. However, the bit-wise prediction rates cannot increase after some certain number of nodes. Refer to Figure 4.20 (b). There are 10 nodes in each hidden layer and the number of hidden layers ranges from 1 to 10. It can be observed that, more hidden layers do not provide higher prediction rates.

We also study the security of Lorenz chaotic system. 50,000 training data are generated for 32-bit PUF and 64-bit PUF, including challenge $C$, intermediate response $R$ and response $R'$ (Figure 4.6). LR is performed to model the CNT PUF only $(C, R)$, Lorenz chaotic system only $(R, R')$, and Lorenz chaotic system based CNT PUF with $(C, R')$, where $(C, R)$ are the input and output of CNT PUF, $(R, R')$ are the input and output of Lorenz chaotic system, and $(C, R')$ are the input and output of the proposed Lorenz chaotic system based CNT PUF. The prediction rates are shown in Table 4.4. The bit-wise prediction rate for CNT PUF only is 100%, which means that CNT PUF is vulnerable to machine learning modeling attacks. The bit-wise prediction rates for Lorenz chaotic system only and Lorenz chaotic system based CNT PUF are both very low ranging from 50% to 70%. Therefore, the high security performance of the proposed Lorenz chaotic system based CNT PUF is mainly contributed by Lorenz chaotic system.

**Table 4.4**
Bit-wise prediction rates of LR for the proposed PUF considering challenge $C$, intermediate response $R$ and response $R'$ using 50,000 32-bit and 64-bit challenge-response pairs training data.

| 32-bit | | | 64-bit | | |
|---|---|---|---|---|---|
| $(C, R)$ CNT PUF only | $(R, R')$ Lorenz only | $(C, R')$ CNT PUF + Lorenz | $(C, R)$ CNT PUF only | $(R, R')$ Lorenz only | $(C, R')$ CNT PUF + Lorenz |
| 100.00% | 53.96% | 52.68% | 100.00% | 54.58% | 53.45% |

## 4.6 Summary

PUFs exploit the hardware fabrication variations to generate secure keys on the fly. Carbon nanotube based circuits are natural candidates for building highly secure PUFs due to significant fabrication variations. However, existing PUFs are reported to be vulnerable to machine learning modeling attacks. In this paper, Lorenz chaotic system is leveraged to CNT PUF through magnifying the differences among responses of similar challenges. It is demonstrated that the proposed Lorenz chaotic system based CNT PUF is resistant to machine learning modeling attacks, including SVM, DBN, LR and ES. The experimental results demonstrate that the machine learning modeling attack methods can achieve as high as 100% bit-wise prediction rates of the CNT PUF without Lorenz chaotic system, while can only obtain less than 55% bit-wise prediction rates of the proposed Lorenz chaotic system based CNT PUF. Therefore, our proposed PUF is resistant to machine learning modeling attacks.

# Chapter 5

# Conclusion

Carbon nanotube interconnects have become a promising replacement material for copper interconnects thanks to their superior conductivity. A timing driven buffer insertion technique is proposed for carbon nanotube interconnects. In the experimental results, it demonstrates that with the same timing constraint, CNT buffering can save over 50% buffer area compared to copper buffering. In addition, CNT buffering can effectively reduce the delay by up to 32% without considering cost. However, due to the imperfection of fabricating long straight carbon nanotubes (CNT), there exist significant variations on the critical CNT geometric parameters such as the diameter and density, which will affect the circuit performance. On the other hand, the prevailing CNT fabrication uses Chemical Vapor Deposition, where the unidimensional spatial correlation manifests strongly. A unidimensional variation aware importance

sampling based stochastic CNT interconnects buffering algorithm is then developed. The simulation results demonstrate that the proposed algorithm on average saves more than 30% buffer area over copper buffering while satisfying timing constraints. In addition, our proposed stochastic Experimental interconnects buffering algorithm achieves much better performance than the best case design and the worst case design in terms of timing and buffer cost.

Although the fabrication variations of carbon nanotubes are not desired for the circuit designs targeting performance optimization and reliability, these inherent imperfections make the CNT based circuits natural candidates for building highly secure physical unclonable function (PUF). A novel CNT PUF design through leveraging Lorenz chaotic system is proposed, which is resistant to machine learning modeling attacks. Support Vector Machine (SVM), Deep Learning (DL), Logistic Regression (LR) and Evolution Strategies (ES) machine learning modeling attack methods are used to evaluate the security performance of the proposed Lorenz chaotic system integrated CNT PUF. The experimental results demonstrate that the machine learning modeling attack methods can achieve as high as 100% bit-wise prediction rates on the CNT PUF without Lorenz chaotic system, while only obtain less than 55% bit-wise prediction rates on the proposed Lorenz chaotic system based CNT PUF, respectively.

# References


[1] Hu, Z.; Comeras, J. M. M. L.; Park, H.; Tang, J.; Afzali, A.; Tulevski, G. S.; Hannon, J. B.; Liehr, M.; Han, S.-J. *Nature nanotechnology* **2016**, *11*(6), 559–565.

[2] Srivastava, N.; Li, H.; Kreupl, F.; Banerjee, K. July **2009**, *8*(4), 542–559.

[3] Liu, L.; Zhou, Y.; Hu, S. In *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pages 362–367, 2014.

[4] Wei, B.; Vajtai, R.; Ajayan, P. *Applied Physics Letters* **2001**, *79*(8), 1172–1174.

[5] Radosavljević, M.; Lefebvre, J.; Johnson, A. *Physical Review B* **2001**, *64*(24), 241307.

[6] van Ginneken, L. P. P. P. In *IEEE International Symposium on Circuits and Systems*, pages 865–868 vol.2, 1990.

[7] Lillis, J.; Cheng, C.-K.; Lin, T. T. Y. Mar **1996**, *31*(3), 437–447.

[8] Shi, W.; Li, Z. June **2005**, *24*(6), 879–891.

[9] Hu, S.; Alpert, C. J.; Hu, J.; Karandikar, S. K.; Li, Z.; Shi, W.; Sze, C. N. Nov **2007**, *26*(11), 2009–2022.

[10] Hu, S.; Li, Z.; Alpert, C. J. In *2009 46th ACM/IEEE Design Automation Conference*, pages 424–429, 2009.

[11] Xu, C.; Li, H.; Banerjee, K. In *Proceedings of IEEE International Electron Devices Meeting*, pages 1–4, 2008.

[12] Kreup, F.; Graham, A. P.; Liebau, M.; Duesberg, G. S.; Seidel, R.; Unger, E. In *Proceedings of IEEE International Electron Devices Meeting (IEDM)*, pages 683–686, 2004.

[13] Lee, S. W.; Lee, D. S.; Morjan, R. E.; Jhang, S. H.; Sveningsson, M.; Nerushev, O.; Park, Y. W.; Campbell, E. E. *Nano Letters* **2004**, *4*(10), 2027–2030.

[14] Sukirno.; Bisri, S. Z.; Hasanah, L.; Mursal.; Usman, I.; Suryamas, A. B.; Edison, T. A. In *Proceedings of IEEE International Conference on Semiconductor Electronics*, pages 155–159, 2006.

[15] Wu, J.; Eastman, M.; Gutu, T.; Wyse, M.; Jiao, J.; Kim, S.-M.; Mann, M.; Zhang, Y.; Teo, K. B. K. *Applied Physics Letters* **2007**, *91*(17).

[16] Lee, S. H.; Min, B.; Park, S. I.; Lee, K. C.; Lee, S. S. In *Proceedings of IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 268–271, 2010.

[17] Chikkadi, K.; Haluska, M.; Hierold, C.; Roman, C. In *Proceedings of IEEE International Conference on Microelectronic Test Structures (ICMTS)*, pages 173–177, 2013.

[18] Nieuwoudt, A.; Massoud, Y. Aug **2008**, *55*(8), 2097–2110.

[19] Srivastava, A.; Xu, Y.; Sharma, A. K. *journal of nanophotonics* **2010**, *4*(1), 041690–041690.

[20] Close, G. F.; Wong, H.-S. P. *IEEE Transactions on Nanotechnology* **2008**, *7*(5), 596–600.

[21] Naeemi, A.; Meindl, J. D. Jan **2007**, *54*(1), 26–37.

[22] Zhang, J.; Patil, N.; Hazeghi, A.; Mitra, S. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 71–76, 2009.

[23] Zhang, J.; Patil, N.; Lin, A.; Wong, H. S. P.; Mitra, S. In *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1159–1164, 2010.

[24] Patil, N.; Lin, A.; Myers, E. R.; Ryu, K.; Badmaev, A.; Zhou, C.; Wong, H. S. P.; Mitra, S. July **2009**, *8*(4), 498–504.

[25] Zhang, J.; Patil, N. P.; Hazeghi, A.; Wong, H. S. P.; Mitra, S. Aug **2011**, *30*(8), 1103–1113.

[26] Appenzeller, J.; Knoch, J.; Martel, R.; Derycke, V.; Wind, S. J.; Avouris, P. Dec **2002**, *1*(4), 184–189.

[27] Patil, N.; Deng, J.; Mitra, S.; Wong, H. S. P. Jan **2009**, *8*(1), 37–45.

[28] Zhang, J.; Patil, N. P.; Mitra, S. Sept **2009**, *28*(9), 1307–1320.

[29] Ali, M.; Ahmed, M.; Chrzanowska-Jeske, M.; Morris, J. In *Proceedings of IEEE International Conference on Nanotechnology (IEEE-NANO)*, pages 1218–1221, 2015.

[30] Cho, G.; Lombardi, F.; Kim, Y. B. In *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pages 289–296, 2010.

[31] Cho, G.; Lombardi, F. In *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 419–425, 2011.

[32] Shahi, A. A. M.; Zarkesh-Ha, P. In *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 140–145, 2012.

[33] Zhang, J.; Bobba, S.; Patil, N.; Lin, A.; Wong, H. S. P.; Micheli, G. D.; Mitra, S. In *Proceedings of ACM/IEEE Design Automation Conference (DAC)*, pages 889–892, 2010.

[34] Wang, C.; Jiang, L.; Hu, S.; Li, T.; Liang, X.; Jing, N.; Qian, W. In *Proceedings of IEEE International System-on-Chip Conference (SOCC)*, pages 362–367, 2015.

[35] Sun, P.; Luo, R. In *Proceedings of IEEE International Symposium on Electromagnetic Compatibility*, pages 103–107, 2009.

[36] Lin, A.; Patil, N.; Wei, H.; Mitra, S.; Wong, H. S. P. Dec **2009**, *56*(12), 2969–2978.

[37] Herder, C.; Yu, M. D.; Koushanfar, F.; Devadas, S. Aug **2014**, *102*(8), 1126–1141.

[38] Cao, Y.; Zhang, L.; Chang, C. H.; Chen, S. July **2015**, *34*(7), 1143–1147.

[39] Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 148–160, New York, NY, USA, 2002. ACM.

[40] Liu, L.; Zhou, Y.; Hu, S. *accpected to IEEE Transactions on Emerging Topics in Computing.*

[41] Liu, Y.; Liu, L.; Zhou, Y.; Hu, S. In *2016 IEEE INFOCOM Cyber-Physical System Security Workshops*, pages 176–180, 2016.

[42] Konigsmark, S. T. C.; Hwang, L. K.; Chen, D.; Wong, M. D. F. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 73–78, 2014.

[43] Adames, I. A. B.; Das, J.; Bhanja, S. In *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, pages 317–322, 2016.

[44] Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. *Machine learning: An artificial intelligence approach;* Springer Science & Business Media, 2013.

[45] Rhrmair, U.; Slter, J. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6, 2014.

[46] Rhrmair, U.; Slter, J.; Sehnke, F.; Xu, X.; Mahmoud, A.; Stoyanova, V.; Dror, G.; Schmidhuber, J.; Burleson, W.; Devadas, S. Nov **2013**, *8*(11), 1876–1891.

[47] Sahoo, D. P.; Nguyen, P. H.; Mukhopadhyay, D.; Chakraborty, R. S. Aug **2015**, *34*(8), 1334–1343.

[48] Du, C.; Bai, G. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 967–972, 2014.

[49] Rührmair, U.; Sölter, J.; Sehnke, F. *IACR Cryptology ePrint Archive* **2009**, *2009*, 277.

[50] Kumar, R.; Burleson, W. In *Proceedings of IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 38–43, 2014.

[51] Yu, M. D.; Hiller, M.; Delvaux, J.; Sowell, R.; Devadas, S.; Verbauwhede, I. *IEEE Transactions on Multi-Scale Computing Systems* **2016**, *PP*(99), 1–1.

[52] Lee, J. W.; Lim, D.; Gassend, B.; Suh, G. E.; Van Dijk, M.; Devadas, S. In *Proceedings of the IEEE VLSI Circuits Symposium*, pages 176–179, 2004.

[53] Majzoobi, M.; Koushanfar, F.; Potkonjak, M. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 670–673. IEEE, 2008.

[54] Suh, G. E.; Devadas, S. In *Proceedings of the annual Design Automation Conference*, pages 9–14. ACM, 2007.

[55] Alpert, C.; Devgan, A. In *Proceedings of ACM/IEEE Design Automation Conference (DAC)*, pages 588–593, 1997.

[56] Gokturk, H. S. In *Nanotechnology, 2005. 5th IEEE Conference on*, pages 677–680. IEEE, 2005.

[57] Scheffer, L. K. In *Proceedings of the 44th annual Design Automation Conference*, pages 576–581. ACM, 2007.

[58] Srivastava, N.; Banerjee, K. *JOM Journal of the Minerals, Metals and Materials Society* **2004**, *56*(10), 30–31.

[59] Raychowdhury, A.; Roy, K. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 237–240. IEEE Computer Society, 2004.

[60] Xu, Y.; Srivastava, A. *International Journal of Circuit Theory and Applications* **2010**, *38*(6), 559–575.

[61] Datta, S. *Nanotechnology* **2004**, *15*(7), S433.

[62] Srivastava, N.; Banerjee, K. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 383–390. IEEE Computer Society, 2005.

[63] Burke, P. J. *IEEE Transactions on Nanotechnology* **2002**, *99*(3), 129–144.

[64] Nabors, K.; White, J. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **1991**, *10*(11), 1447–1459.

[65] 45nm technology. http://www.nangate.com.

[66] Ngspice. http://ngspice.sourceforge.net/.

[67] Buffer capacitance. https://www.pagiamtzis.com/articles/how-to-find-input-capacitance-using spice.

[68] International technology roadmap for semiconductors. http://www.itrs2.net/itrs reports.html. **2007**.

[69] Aitken, R.; Yeric, G.; Cline, B.; Sinha, S.; Shifren, L.; Iqbal, I.; Chandra, V. In *Proceedings of the 2014 on International symposium on physical design*, pages 65–68. ACM, 2014.

[70] International technology roadmap for semiconductors. **2011**.

[71] Rabaey, Jan M., A. P. C.; Nikolic, B. *Digital Integrated Circuits: A Design Perspective, second edition;* Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2003.

[72] Chang, H.; Sapatnekar, S. S. Sept **2005**, *24*(9), 1467–1482.

[73] Wei, T.; Chen, X.; Hu, S. Oct **2011**, *30*(10), 1569–1573.

[74] Rudas, T. *Probability theory;* SAGE Publications, Inc: Thousand Oaks, CA, 2004.

[75] Sun, S.; Feng, Y.; Dong, C.; Li, X. Dec **2012**, *31*(12), 1831–1844.

[76] Owen, A. B. *Monte Carlo theory, methods and examples;* 2013.

[77] Mckay, M.; Beckman, R.; Conover, W. *Technometrics* **1979**, *21*(2), 239–245.

[78] Liu, X.; Pichler, T.; Knupfer, M.; Golden, M. S.; Fink, J.; Kataura, H.; Achiba, Y. Jul **2002**, *66*.

[79] Verma, K.; Singh, R.; Kaushik, B.; Majumder, M. K. In *Proceedings of IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 599–604, 2011.

[80] Findik, O.; Kahramanli, Ş. In *International Conference on Advances in Information Technology*, pages 27–33. Springer, 2010.

[81] Li, W.; Zhang, Q.; Ding, Q. In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pages 262–266, 2015.

[82] Chen, G.; Ueta, T. *International Journal of Bifurcation and chaos* **1999**, *9*(07), 1465–1466.

[83] Rössler, O. E. *Physics Letters A* **1976**, *57*(5), 397–398.

[84] Yang, X.-S.; Chen, G. *Far East J. Dyn. Syst* **2002**, *4*, 27–38.

[85] Lü, J.; Chen, G.; Cheng, D. *International Journal of Bifurcation and Chaos* **2004**, *14*(05), 1507–1537.

[86] Song, W.; Liang, J. *International Journal of Pure and Applied Mathematics* **2013**, *83*(1), 101–110.

[87] Wei, H.; Patil, N.; Zhang, J.; Lin, A.; Chen, H.-Y.; Wong, H.-S.; Mitra, S. In *Proceedings of International Symposium on VLSI Technology (VLSIT)*, pages 237–238, 2010.

[88] Frick, V.; Hebrard, L.; Poure, P.; Anstotz, F.; Braun, F. Dec **2003**, *3*(6), 752–760.

[89] Solis, J. E. M.; Navarro, M. G.; Mejia, I.; Lozano, R. Z. G.; Rojas, F. L.; Ocampo-Hidalgo, J.; del Toro, H. B. Spring **2016**, *39*(2), 127–131.

[90] Fukushima, K.; Souissiy, Y.; Hidano, S.; Nguyeny, R.; Dangery, J. L.; Guilleyy, S.; Nakano, Y.; Kiyomoto, S.; Sauvage, L. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 201–207, 2016.

[91] Ikezaki, Y.; Nozaki, Y.; Yoshikawa, M. In *2016 IEEE 5th Global Conference on Consumer Electronics*, pages 1–2, 2016.

[92] Becker, G. T. Aug **2015**, *34*(8), 1295–1307.

[93] Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory Technical report, DTIC Document, **1986**.

[94] Hinton, G. E. *Neural computation* **2002**, *14*(8), 1771–1800.

[95] Hinton, G. E.; Osindero, S.; Teh, Y.-W. *Neural computation* **2006**, *18*(7), 1527–1554.

[96] https://cran.r-project.org/web/packages/e1071/index.html. SVM, R.

[97] https://cran.r-project.org/web/packages/deepnet/index.html. Deepnet, R.

[98] Vijayakumar, A.; Kundu, S. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 653–658, 2015.

[99] Tucker, W. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* **1999**, *328*(12), 1197–1202.

[100] Zalivaka, S. S.; Zhang, L.; Klybik, V. P.; Ivaniuk, A. A.; Chang, C.-H. In *Secure System Design and Trustable Computing;* 2016; pages 39–81.

# Appendix A

# List of Notations

The notations are listed as follows.

$d$-diameter of an SWCNT

$d_0$-nominal diameter of an isolated SWCNT

$d_{i,j}$-diameter of an SWCNT in grid $(i, j)$ $l$-length of an isolated SWCNT

$l_0$-nominal length of an isolated SWCNT

$s$-cross section area of bundled SWCNTs

$\delta$-density of bundled SWCNTs

$\delta_0$-nominal density of bundled SWCNTs

$\delta_{i,j}$-density of bundled SWCNTs in grid $(i, j)$

$N_{cnt}$-number of SWCNTs in the bundle

$y$-distance between an isolated SWCNT and ground

$\epsilon$-permittivity

$R_v$-resistance of bundled SWCNT interconnects

$R_{v_0}$-nominal resistance of bundled SWCNT interconnects

$R_{vs_{ij}}$-resistance of bundled SWCNT interconnects in grid $(i, j)$

$R_S$-unit scattering resistance of an isolated SWCNT

$R_S^{bundle}$-scattering resistance of bundled SWCNTs

$C_v$-capacitance of bundled SWCNT interconnects

$C_{v_0}$-nominal capacitance of bundled SWCNT interconnects

$C_{vs_{ij}}$-capacitance of bundled SWCNT interconnects in grid $(i, j)$

$C_E$-electrostatic capacitance of an isolated SWCNT

$\beta$-parameter to model the uncertainty of the resistance and capacitance

$R_{ij}^{\beta}$-resistance of bundled SWCNT interconnects in grid $(i, j)$ for a given $\beta$

$C_{ij}^{\beta}$-capacitance of bundled SWCNT interconnects in grid $(i, j)$ for a given $\beta$

$R_{ij}^{l}$-lower bound of resistance of SWCNT interconnects in grid $(i, j)$

$R_{ij}^{u}$-upper bound of resistance of SWCNT interconnects in grid $(i, j)$

$C_{ij}^{l}$-lower bound of capacitance of SWCNT interconnects in grid $(i, j)$

$C_{ij}^{u}$-upper bound of capacitance of SWCNT interconnects in grid $(i, j)$

$\mu_{R_N}$-mean value of normal distribution of resistance

$\sigma_{R_N}^2$-variance of normal distribution of resistance

$\mu_{C_N}$-mean value of normal distribution of capacitance

$\sigma_{C_N}^2$-variance of normal distribution of capacitance

$v_{R_t}$-number of degrees of freedom of $t$ distribution of resistance

$v_{C_t}$-number of degrees of freedom of $t$ distribution of capacitance

$d^w_{n_k}$-delay of wire segment $w$ of $k^{th}$ sample under normal distribution

$d^w_{t_k}$-delay of wire segment $w$ of $k^{th}$ sample under $t$ distribution

# Appendix B

# Biographical Sketch

Miss. Lin Liu received her B.Sc. degree of Electronic Information Engineering from University of Science and Technology of China, Hefei, P.R. China in 2011. She is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA. Her research interests include carbon nanotube (CNT) based physical design, fabrication variation aware CNT circuit performance optimization, CNT based hardware security, and smart home system and scheduling.

She was a Firmware Engineer intern at Broadcom Corporation, Santa Clara, CA in Summer 2013, and Software Engineer intern at Facebook, Inc., Menlo Park, CA in Summer 2015. She was also an intern at Netease Game, Guangdong, China in

Summer 2014 and Summer 2016.

The following are the technical contributions by Miss. Lin Liu during the period of her candidacy between 2011 and 2017:

**Journal Articles**

[1] **Lin Liu**, Yuchen Zhou, and Shiyan Hu, "Stochastic Buffering For Bundled SWCNT Interconnects Considering Unidimensional Fabrication Variation", accepted to *IEEE Transactions on Emerging Topics in Computing*.

[2] **Lin Liu**, Yang Liu, Lizhe Wang, Albert Zomaya, and Shiyan Hu, "Economical and Balanced Energy Usage in The Smart Home Infrastructure: A Tutorial and New Results", *IEEE Transactions on Emerging Topics in Computing*, Vol. 3, No. 4, pp. 556-570, December 2015.

[3] **Lin Liu**, Xin Yang, Han Huang and Shiyan Hu, "Smart Home Scheduling For Cost Reduction and Its Implementation on FPGA", *Journal of Circuits, Systems and Computers (JCSC)*, Vol. 24, No. 4, pp. 1-15, April 2015.

[4] Jia Wang, **Lin Liu**, Yuchen Zhou, and Shiyan Hu, "Buffering Carbon Nanotube Interconnects Considering Inductive Effects", accepted to *Journal of Circuits, Systems and Computers (JCSC)*.

[5] Xiaodao Chen, Lizhe Wang, Albert Zomaya, Shiyan Hu, **Lin Liu**, "Cloud Computing For VLSI Floorplanning Considering Peak Temperature Reduction", *IEEE Transactions on Emerging Topics in Computing*, Vol. 3, No. 4, pp. 534-543, December 2015.

**Book Chapters**

[6] **Lin Liu**, Yuchen Zhou and Shiyan Hu, "Timing Driven Buffer Insertion for Carbon Nanotube Interconnects", in the *Nano-CMOS and Post-CMOS Electronics: Device and Modelling*, Saraju P. Mohanty and Ashok Srivastava, ed., IET, 2015. (invited)

**Conference Papers**

[7] Yang Liu, **Lin Liu**, Yuchen Zhou, and Shiyan Hu, "Leveraging Carbon Nanotube Technologies in Developing Physically Unclonable Function for Cyber-Physical System Authentication", in *Proceedings of IEEE INFOCOM Cyber-Physical System Security Workshop*, 2016.

[8] **Lin Liu**, Yuchen Zhou and Shiyan Hu, "Buffering Carbon Nanotube Interconnects for Timing Optimization", in *Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2014. (invited)

[9] **Lin Liu**, Yuchen Zhou, Yang Liu and Shiyan Hu, "Dynamic Programming Based Game Theoretic Algorithm for Economical Multi-User Smart Home Scheduling", in *Proceedings of IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2014. (invited)

[10] Jia Wang, Xiaodao Chen, **Lin Liu** and Shiyan Hu, "Fast Approximation For Peak Power Driven Voltage Partitioning in Almost Linear Time", in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012.

# Appendix C

# Letters of Permission

## C.1 Permission Letters for Chapter 2 and Figure 3.2

## C.2 Permission Letters for Chapter 3

Copyright Clearance Center RightsLink®

Home | Create Account | Help

IEEE Requesting permission to reuse content from an IEEE publication

| | |
|---|---|
| **Title:** | Buffering Single-Walled Carbon Nanotubes Bundle Interconnects for Timing Optimization |
| **Conference Proceedings:** | VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on |
| **Author:** | Lin Liu |
| **Publisher:** | IEEE |
| **Date:** | July 2014 |



LOGIN

**If you're a copyright.com user,** you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

## Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK ••• CLOSE WINDOW

**Figure C.1:** The copyright permission for Chapter 2 and Figure 3.2.

Copyright Clearance Center RightsLink

Home | Create Account | Help

IEEE Requesting permission to reuse content from an IEEE publication

**Title:** Stochastic Buffering For Bundled SWCNT Interconnects Considering Unidimensional Fabrication Variation

**Author:** Lin Liu

**Publication:** IEEE Transactions on Emerging Topics in Computing

**Publisher:** IEEE

**Date:** Dec 31, 1969

Copyright © 1969, IEEE

LOGIN

**If you're a copyright.com user,** you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

## Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK ||| CLOSE WINDOW



**Figure C.2:** The copyright permission for Chapter 3.