Electron-Scattering-Aware Mixed-Signal IC Placement with Reinforcement Learning for EBL Technologies

By
©Mohammad Hajijafari

A dissertation submitted to the School of Graduate Studies
in partial fulfillment of the requirements for the degree of

Master of Engineering

Faculty of Engineering & Applied Science
Memorial University of Newfoundland

Supervisory Committee

Dr. Lihong Zhang

May 2022

St. John's, Newfoundland

Abstract

As the technology node shrinks to the nanometer regime, the demand for new lithography methods with high resolution and low cost is increasing. Electron beam lithography (EBL) is one of the promising next-generation lithography (NGL) technologies that can tackle both challenges compared to the traditional lithography methods. Electron scattering, which causes pattern distortion in layout, is one of the main challenges for industry to widely adopt EBL in technologies below 22nm. Two major effects generated by electron scattering in EBL process are proximity effect and fogging effect. This thesis proposes a reinforcement-learning (RL) placement method that trains a neural network as an agent to effectively control the fogging and proximity effects in the EBL technologies. To speed up our method compared to other popular placement approaches (e.g., absolute coordinate based analytical placement, simulated annealing (SA) based placement, advantage actor critic (A2C) based placement), we benefit from the following innovations: using topological floorplan representation for manipulating our layouts during placement, and deploying an RL trained agent that can intelligently take actions. To more effectively tackle mixed-signal ICs, our method focuses on the sensitive analog devices, which are better protected from potential variations due to the fogging and proximity effects of other digital/analog portions. The experimental results show that our proposed placer is able to efficiently decrease the variation of the fogging and proximity effects among sensitive

devices in the analog portion up to 89.26% and 95.22% respectively, while it is 15, 4.3, and 5 times faster than the analytical RL-based placement, SA-based and A2C-based placement methods, respectively. In summary, our proposed approach has three main contributions: 1) to the best of our knowledge, our work is the first study that considers the fogging and proximity effects in analog portion of mixed-signal ICs, 2) we apply deep Q-network (DQN) based placement to handle the fogging and proximity effects that improves the quality and speed of placement by intelligently choosing actions, and 3) we introduce a new RL placer in this study, which is based on a topological representation scheme resulting in much smaller configuration space and in turn faster placement operation.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Lihong Zhang for his continuous support in my master's study, research, motivation, enthusiasm, immense knowledge and his assistance in writing papers and thesis. This work was supported by the Department of Electrical and Computer Engineering at Memorial University of Newfoundland.

To the loving memory of my father

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Next-generation lithography	**NGL**
Electronic Design Automation	**EDA**
Electron Beam Lithography	**EBL**
Critical Dimension	**CD**
Next Generation Lithography	**NGL**
Backward Proximity Effect	**BPE**
Forward Proximity Effect	**FPE**
Point Spread Function	**PSF**
Reinforcement Learning	**RL**
Deep Neural Network	**DNN**
Deep Q-Network	**DQN**
Simulated Annealing	**SA**
Advantage Actor Critic	**A2C**
Successive-Approximation-Register	**SAR**
Analog-to-Digital Converter	**ADC**
Integrated Circuit	**IC**

Chapter 1

Introduction

1.1. Lithography in the Semiconductor Manufacturing Process

As the modern semiconductor manufacturing process has to fabricate new technologies below 22nm, one of the designers' concerns is the high cost of masks in the traditional optical lithography, which is approximately four times more than that in the 45nm technology [1]. The lithography, which transfers the pattern to a thin layer (e.g., metal layer or polysilicon layer) on the substrate, is one of the key operations in semiconductor fabrication. Therefore, any error or variation in this procedure results in a defect or even a malfunction in the manufactured electronic device.

Based on the data shown in Fig. 1 as a demonstration of the Moore's law, transistor density approximately doubles every two years. To make sure that this trend is still followed by industry, Intel introduced and reported a 10-nm technology node in 2017 that can increase the transistor density to about 100 million transistors per square millimetre [2]. This number is 2.7 times more than their previous announcement two years before, and shows that Moore's law is still correct and trustable. In addition, this improvement in

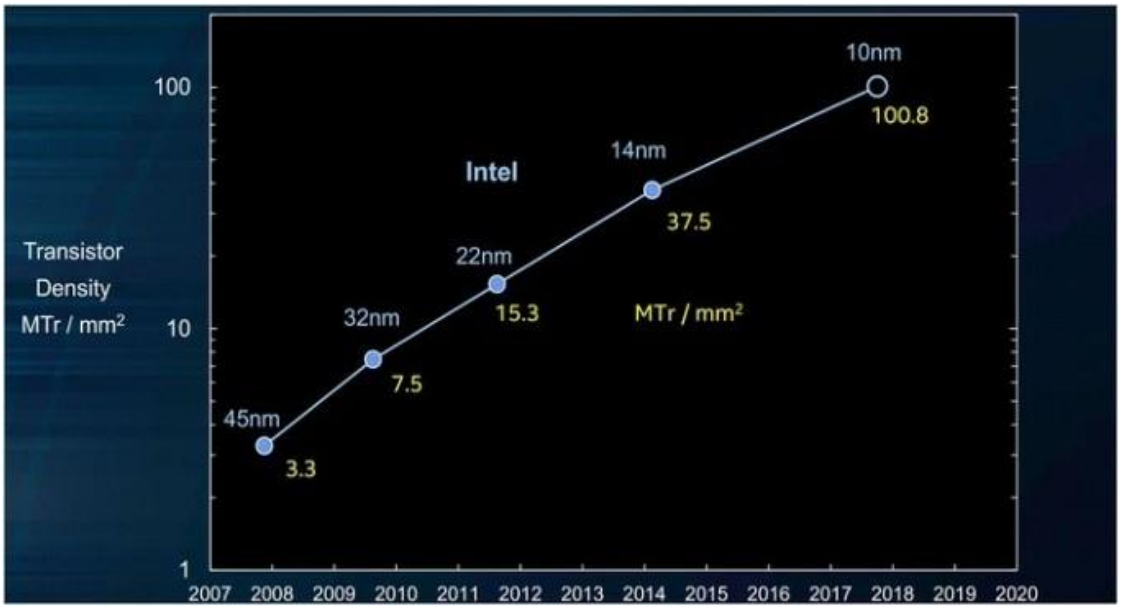


Fig. 1. Transistor density trend [3].

transistor density means the size of the transistor must keep shrinking, and subsequently the patterning resolution requires to be further increased. Furthermore, the patterning process should meet the requirements of speed, cost and power, and address the challenges of the modern manufacturing procedure, which in turn makes the lithography process more complicated, complex and daunting. Therefore, the lithography method must be upgraded in order to let the technology nodes continue shrinking and improving.

The traditional photolithography, which uses 193 nm wavelength, has been employed as one of the most popular techniques in the semiconductor fabrication industry for decades. However, as the technology node has shrunk, this method was not able to meet the demands and reach beyond its resolution limits. So, the resolution enhancement technologies (RET) method was introduced to solve this problem. The RET invented to increase the capability of photolithography includes phase shift mask, optical proximity correction (OPC),

immersion, modified or off-axis illumination (OAI), and multiple patterning [4]. One of the most popular RETs that has been suitable and applicable to semiconductor mass production in the industry is multiple patterning. The multi-patterning method (e.g., double patterning and triple patterning) enables the traditional photolithography to go beyond its resolution limits and have been used for many years in the industry.

Currently, the 22 nm and 14 nm technology-node electronic devices are fabricated by multi-patterning immersion ArF (argon fluoride) lithography. However, as this method is used to manufacture devices with smaller feature size, major concerns and challenges appear. Using more mask to increase the resolution in the new technologies results in much higher mask cost and tighter overlay control [5]. So, utilizing multi-patterning as a lithography method in high volume manufacturing leads to a more expensive and sophisticated electronic device fabrication. As a matter of fact, one earlier investigation from Global Semiconductor Alliance in 2007 already showed that the major concern of designers in semiconductor device fabrication for moving from one technology node to a new one is the higher mask cost [1]. Traditional photolithography and its RET methods have reached their limitation and are no longer effective; so, a new lithography method is required to address such industry demands.

Therefore, next-generation lithography (NGL) methods have been introduced to overcome these problems for further technology scaling. NGL methods mainly include electron-beam lithography (EBL), extreme ultraviolet lithography (EUVL), nanoimprint lithography (NIL), directed self-assembly (DSA), and focused ion beam lithography

(FIBL). Electron beam lithography (EBL) is one of the best among them, with higher accuracy in comparison to light-based technologies. EBL is a maskless lithography technology that prints feature patterns directly on the wafer with high resolution down to a few nanometers [6]. In EBL technology, in order to create a feature, electron beams are first emitted from the electron source. After going through a set of lenses and apertures, they directly write the pattern onto the resist-coated substrate. However, as EBL is a high-resolution pattern formation method, some undesired exposures, which occur during the lithography, would result in a feature length change [7]. This critical dimension (CD) error is caused by electron scattering during the lithography, which results in a performance degradation or even malfunction. The two major concerns in EBL induced by this process variation are fogging effect and proximity effect. Therefore, these two challenges should be resolved to let the EBL be widely applied in the semiconductor fabrication industry. So, in this thesis we are motivated to concentrate on the impact of fogging effect and proximity effect on sensitive analog devices especially in the context of the surrounding digital portion in the mixed-signal ICs. In other words, we intend to minimize the impact of fogging effect and proximity effect on the performance of analog circuits.

1.2. Structure of the Thesis

The rest of the thesis is organized as follows. Chapter 2 reviews the basics and previous works related to the fogging and proximity effects. Moreover, different analog placement methods, including topological-representation-based, absolute-coordinates-based, and AI-based analog placement methods, are discussed. In Chapter 3, the energy distribution for

the fogging and proximity effects, and our variation models for controlling the electron scattering effects in the EBL technologies are proposed. Chapter 4 presents a B*-tree based placement methodology to control the fogging and proximity effects while still satisfying the other critical analog constraints. In Chapter 5, the experimental results of four different placement methods are discussed. In Chapter 6 we draw conclusions and point out future work for this research.

1.3. Summary

In this chapter, the basics and issues of the traditional lithography process in semiconductor manufacturing industry were discussed. Based on the Moore's law, the transistor density doubles every two years. Therefore, the resolution of the lithography methods must be improved continuously to keep up with this trend. We reviewed the traditional photolithography and its limitations for the sub-22nm technology nodes. In addition, the solutions called RETs were presented to improve the resolution of photolithography. Finally, the next generation lithography (NGL) methods were introduced as effective methods for new technology nodes in semiconductor fabrication.

Chapter 2 Fundamentals and Literature Review

As transistors are becoming increasingly smaller to form electronic integrated circuits (ICs) in a more compact area, the demands for high-resolution lithography methods have been increased. The traditional photolithography methods and their resolution enhancements technologies have reached their limitations and are not as effective and efficient for the technology nodes below 22nm. Therefore, the next generation lithography methods, which include extreme ultraviolet lithography (EUVL), nanoimprint lithography (NIL), directed self-assembly (DSA), and electron beam lithography (EBL), have been introduced.

In this chapter, we discuss in detail the process, advantages, and disadvantages of these four NGL methods. EBL is one of the most popular NGLs that directly print the patterns on the wafer with a high-resolution down to a few nanometres. However, the scattering of electrons in EBL is the most challenging concern of designers, which leads to two effects: fogging effect, and proximity effect. In the following, we review the studies and publications on these two effects and the solutions to minimizing them. One of the most effective techniques to address these two effects is to minimize them during the chip placement stage. So in the last section of this chapter, we will present three different

placement approaches in the literature including: topological-representation-based placement, absolute-coordinates-based placement, and AI-based placement.

2.1. Next Generation Lithography (NGL) Methods

Lithography is the process of printing a pattern on the wafer, and lithography improvement is a major contribution to the continuation of the Moore's law in the modern time. Photolithography has been the most popular lithography technique in semiconductor industry for several decades. This approach utilizes an optical-mechanical system including a set of lenses, masks, and light sources. To manufacture different circuits with specifically defined resolution, a special light source is applied. It might be krypton fluoride (KrF) excimer, ultraviolet, or argon fluoride (ArF) laser. The resolution of the photolithography system is calculated by the following equation [8]:

$$Resolution = k_1 \frac{\lambda}{NA} , \tag{1}$$

where k_1 is a constant depending on the lithography process. Moreover, λ and NA are the light source wavelength and numerical aperture, respectively. In the common lithography process, k_1 and NA are in the range of 0.5 to 0.8 and 0.5 to 0.6, respectively. In addition, λ depends on the light source, while an argon fluoride (ArF) laser has the minimum wavelength of 193 nm. In the new semiconductor technologies, the associated feature size keeps shrinking nowadays. Thus, it always demands a lithography process with better resolution, which can print the patterns on the wafer with higher accuracy.

Based on the equation above, to improve the resolution we can decrease both k_1 and λ or increase NA. Using imaging resist material can reduce k_1 to less than 0.5. Moreover, the common way of resolution improvement is to use light sources with less wavelength. So, the argon fluoride (ArF) was applied as the light source, which has the minimum wavelength of 193 nm compared to others. Moreover, complicated computer modeling has been applied to design an innovative optical system with increased NA lenses for new lithography processes [9]. In addition, the semiconductor industry has applied various resolution enhancement technologies (RETs) to increase the resolution of photolithography, among which multi-patterning is one of the most popular and effective approaches. However, photolithography, which has already reached its limit, is no longer economically effective for manufacturing sub-22 nm technologies. Therefore, the next generation lithography methods have been introduced to overcome this challenge. The most popular NGLs include extreme ultraviolet lithography (EUVL), nanoimprint lithography (NIL), directed self-assembly (DSA), and electron beam lithography (EBL).

The EUV lithography technology, which has high resolution used for sub-7nm technology nodes, applies ultraviolet radiation for printing the patterns on wafer [10]. This method utilizes photons with wavelength of 13.5 nm, which are generated by a plasma source. As Fig. 2 shows, the EUV radiation is emitted by a CO_2 laser and then collected by an optical tool. Next, the light is controlled and projected to wafer by a set of optical elements called illuminator, reticle stage, and projection optics. EUVL has significantly reduced the semiconductor device manufacturing cost and increased the reliability of this process because of high resolution and throughput. However, since the EUV light is

absorbed by all tools and elements, such as collector, illuminator and projection optics, in the lithography system they must utilize the high-performance molybdenum–silicon multilayer mirrors. In addition, the entire optical path must be housed in a near-vacuum environment. Therefore, these two considerations in EUVL raise the complexity and startup cost of the lithography process.

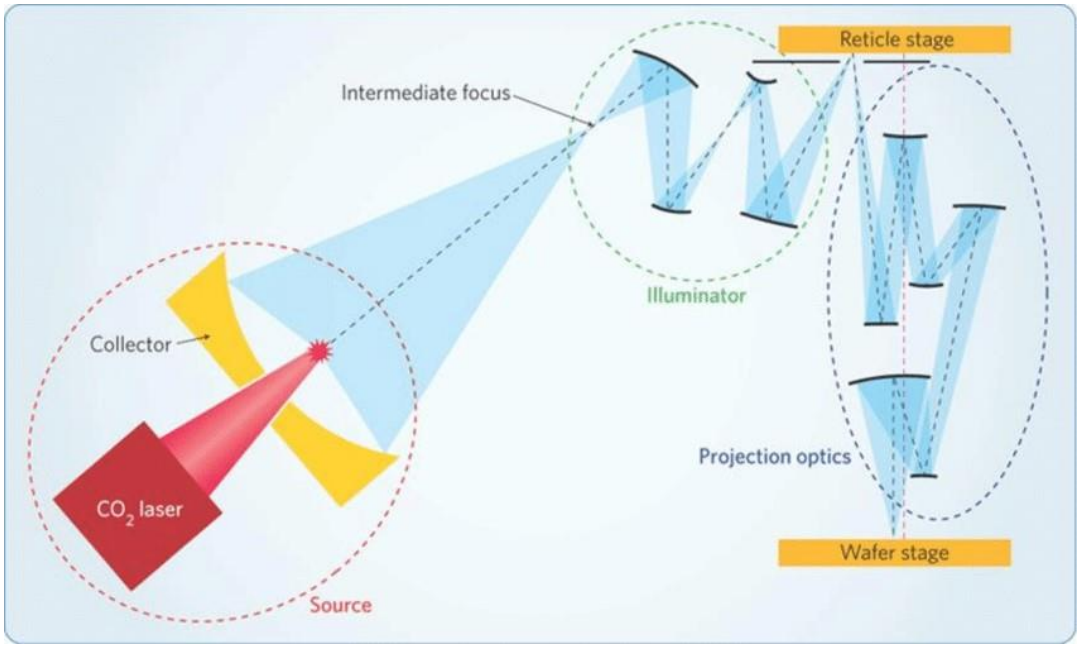


Fig. 2. Schematic of the EUVL process.

Nanoimprint lithography (NIL) is a simple and high resolution (down to sub-3 nm) technology whose resolution is independent of area size being printed. Therefore, NIL has high throughput and accuracy that result in a low-cost mass production of semiconductor devices. This lithography approach is divided into four major categories: thermal NIL, UV-NIL, laser-assisted NIL and electrochemical nanoimprints [11]. In NIL, the mold is used to extract the pattern on the resist or substrate by utilizing high pressure heating, UV

exposure, laser radiation, and electrical voltage in thermal NIL, UV-NIL, laser-assisted NIL and electrochemical nanoimprints, respectively. The NIL techniques have significant and valuable advantages in nanomanufacturing; however, the fabrication process of the mold for each specific topological pattern is time-consuming, which inevitably increases the cost of fabrication.

The directed self-assembly (DSA) technique is one of the most promising NGLs that takes advantage of block copolymer materials to print the fine pattern shapes on the wafer with high resolution down to sub-10 nm. This method appeals to semiconductor device manufacturers thanks to its potential for efficient and cost-effective mass production of electronic nanometer devices. The DSA lithography process has two types: chemoepitaxy and graphoepitaxy, which apply chemical stripes or physical trenches to direct and define the block copolymer structures, respectively [12]. DSA has many advantages, such as low cost, high throughput and resolution, which make it as an excellent candidate for lithography in semiconductor industry. However, this technique has shown weaknesses in critical dimension (CD) control and line edge roughness (LER) compared to other NGLs. Moreover, it needs significantly long processing time and is designed for limited number of pattern shapes in comparison with the alternative NGLs.

The last popular NGL technique is EBL, which utilizes electron beams to print the fine and accurate features on the wafer. This approach is fully described in the following section.

2.2. Electron Beam Lithography

Various NGL approaches have been introduced to support technology node improvement and feature size shrinking. One of the most well-known and effective NGLs is electron beam lithography (EBL). EBL utilizes electron beams to directly print the pattern on the substrate. This maskless lithography method has high resolution and significantly decreases the manufacturing cost. The direct writing in EBL enables excellent controlling in lithography process that leads to fine and accurate patterns with high resolution down to 5 nanometers [13].

Fig. 3 illustrates the EBL system and the process of printing a pattern on the substrate by using this method. In this system, the electron gun is a source of electrons that generates, accelerates, concentrates and emits electron beams to resist-coated substrate. The electrons are generated by electron emitters or cathodes. Then, the electrons are accelerated by an electrostatic field that increases the kinetic energy of electrons to a specific level defined by the fabrication technology. The high-energy electron beam is controlled and focused by a set of instruments including lenses, deflectors and blanking plate. After passing through the guidance system, the e-beam is exposed onto the resist as one pixel and this sequence of tasks are repeated until the full pattern is printed on the substrate. In the next step, the projection is followed by etching and deposition to create the fine pattern on the substrate.

The direct writing and eliminating the need for mask in lithography make the EBL as one of the best candidates for lithography in high-volume manufacturing of semiconductor devices. However, there are two major problems that increase the risk of using this method

in industry. The first one is the low throughput resulting in a low speed of manufacturing; so, it is not economically viable for mass production of electronic devices. The second challenge is the effects induced by electron scattering including fogging and proximity effects.

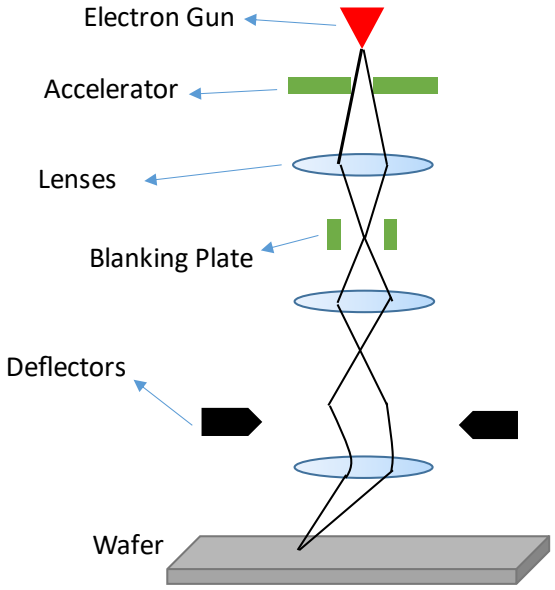


Fig. 3. Electron beam lithography process.

In the recent years, many researchers and laboratories have researched on EBL challenges. To speed up the EBL process and increase the throughput, a new method of parallel writing, which is called multiple e-beam direct write (MEBDW) lithography, was introduced [14]. In this technique, a bunch of parallel electron beams are emitted and projected to the wafer to directly print a pattern on the substrate, which has much higher speed than the single e-beam writing. So, now the low throughput and high lithography cost of EBL can be solved by using MEBDW in the industry. For the second challenge,

several promising studies and research work have been conducted to overcome the fogging and proximity effects in EBL. In the following we first review these works and then presents the concept of our study for these two effects.

2.3. Fogging and Proximity Effects

Fogging and proximity effects are two major concerns from the designers in sub 22 nm technology nodes that apply EBL as a lithography process. As Fig. 4(a) shows, a portion of electrons normally emitted from the primary electron beam scatters into resist and substrate after exposing the resist. These scattered ones may scatter again and generate back-scattered electrons. Such back-scattered electrons, if passing through substrate and resist, hit the bottom of the objective lens. They will produce the next-generation electrons called re-scattered electrons [15]. The effect induced by scattered and back-scattered electrons is called *proximity effect*, while the re-scattered electrons cause *fogging effect*. To be exact, there are two kinds of proximity effects. The first one called *forward proximity effect* is caused by scattered electrons (also called *forward-scattered electrons*), which have low angle deviation after exposing the resist and substrate. The back-scattered electrons (also called *backward-scattered electrons* in some papers and studies), which have a wider angle scattering and deviation, produce *backward-proximity effect*. These scattered electrons cause a non-desired exposure on resist resulting in a layout pattern distortion, which may cause critical dimension (CD) error up to several nanometers [16], as shown in Fig. 4(b). The CD error depends on the type of the resist applied in the lithography process. The electron scattering effects (i.e., the fogging and proximity effects) decrease the feature

size in the case of positive resist, while they increase it when the negative resist is applied. In both cases, these two effects change the layout pattern and subsequently the feature geometry; so, they may result in circuit performance degradation or even malfunction if they are not properly taken into account.

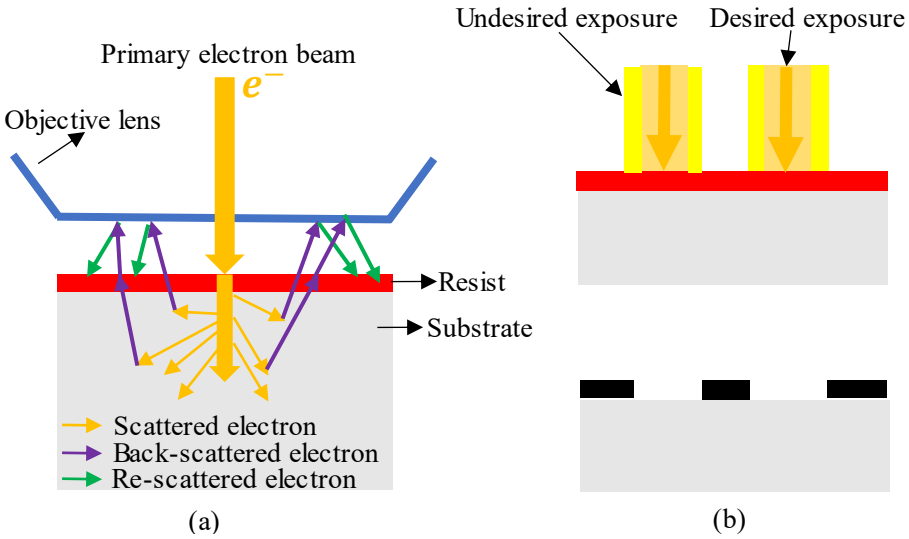


Fig. 4. (a)Fogging and proximity effects induced by electron scattering. (b) Critical dimension error induced by electron scattering.

When the re-scattered electrons hit the bottom of the lens, they expose the resist and substrate to show the fogging effect in a long distance from the primary e-beam, which belongs to the category of long-range effects. In contrast, the scattered and back-scattered electrons have low angle deviation that leads to short range proximity effect. In addition, the re-scattered electrons have several collisions with other particles at the moment of scattering. Subsequently they lose a great deal of kinetic energy in each collision, which results in the low intensity feature of the fogging effect. On the other hand, the scattered and back-scattered electrons have much fewer touches with other particles and thus lose

less energy, which leads to the high intensity feature of the proximity effect. Fig. 5 illustrates the energy distribution of the proximity effect and fogging effect in Fig. 5(a) and Fig. 5(b), respectively. As shown, in contrast to the proximity effect being a short-range and high intensity effect, the fogging effect is a long range and low intensity effect. Fig. 5(c) shows the energy distribution induced by the summation of both effects.

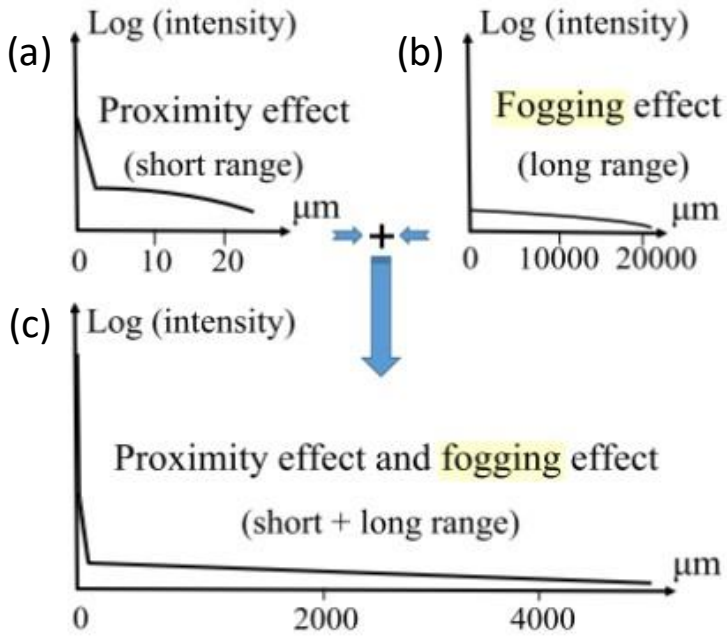


Fig. 5. Range and intensity of (a) proximity effect, (b) fogging effect, and (c) both effects [15].

2.4. Previous Works

2.4.1. Control of Fogging and Proximity Effects

Some studies have been conducted on the minimization of the fogging and proximity effects for integrated circuits (IC). There are two major methodologies in the literature to minimize the fogging and proximity effects: 1) minimizing fogging and proximity effects

during the manufacturing process and 2) in the layout design stage. For the first category, Pease [17] introduced two approaches. To avoid scattering, the author suggested to use very thin substrate that minimizes the coefficient of electron back-scattering. In this technique, thin resists and very low abbreviation lenses are used in the EBL system to reduce scattering of electrons after exposing the resist and lens. Thus, this approach enables the designers to manufacture devices with feature sizes less than 10 nm. The second approach is to employ ion beam lithography, which uses ions instead of electrons. However reasonable these methods sound, in practice they are still not efficient and ready enough to be directly used for mass production in semiconductor industry.

In another promising research work, Hudek *et al.* [15] applied the dose modulation method to control the critical dimension unity (CDU) of the layout. The optimized parameter for dose modulation was determined by convolution of pattern density and fogging effect function. However, this work could only control CDU within 8nm, and it required an exposure system that was able to work with a sufficient number of dose steps. In [18], Figueiro *et al.* presented a method to compensate the fogging effect based on pattern density. They verified and calibrated the fogging effect model based on several test patterns and then applied dose and/or geometric modulation to make up the fogging effect. While they used density map instead of actual pattern to reduce the computation cost, runtime of this method is still significantly large.

In [19], Shimomura *et al.* used a plate containing electron holes in an electron beam system, which could absorb back-scattered electrons. The authors could successfully

reduce the critical dimension (CD) error down to 8nm with this method. Nevertheless, they were still not able to completely remove but just reduce the fogging effect in the EBL process. Morimoto *et al.* in [20] used a method to characterize and quantify the fogging effect. They measured the current flow generated by fogging electrons in the electrodes installed on the objective lens and specimen. Their experiments showed that the fogging effect is dependent on the objective lens materials. They indicated that using carbon for the objective lens could minimize the fogging effect in comparison to aluminum, copper or stainless steel.

When it comes to proximity effect minimization during the manufacturing process, Kamikubo *et al.* in [21] introduced a novel formula for proximity effect minimization through dose correction. They used the expansion series to approximate the optimum correction dose. In comparison to the conventional formulas, they reduced the correction error to less than 1% and minimized the pattern dimensional errors down to ± 4 nm. Fig. 6 shows the comparison between their work and the conventional dose correction formula. To tackle the proximity effect in the fabrication process, Seo and Suh [22] took advantage of a thin film of silicon dioxide (SiO_2), which could minimize the electron scattering after exposure to resist and substrate. They added a thin layer of SiO_2 on a bare substrate and then coat the surface with the resist. Their results showed that increasing the thickness of the SiO_2 layer led to a decrease of the proximity effect and subsequently less CD error. However, adding the SiO_2 layer and incrementing its thickness require extra steps in lithography to remove that layer, which might significantly raise the manufacturing cost and complexity. Fig. 7 presents the complete schematic of their scheme.

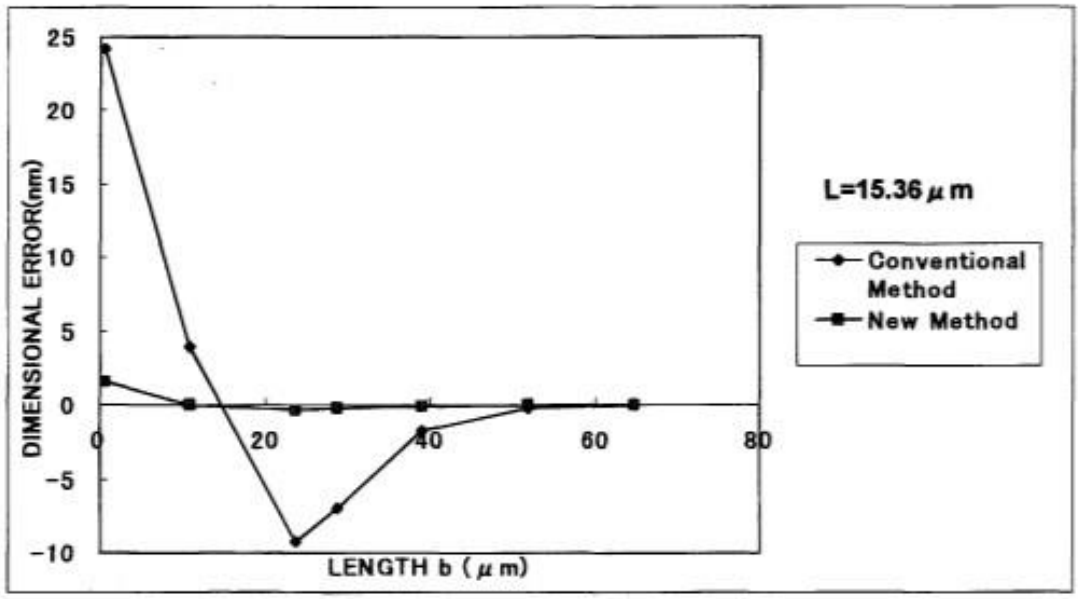


Fig. 6. Dimensional errors after dose correction by [21] and conventional methods.

As for the second category of minimizing the fogging and proximity effects in the layout design stage, Chen *et al.* [23] tackled the fogging effect by using an analytical placement method, which is one of the first studies considering the fogging effect in the chip design process. They derived the fogging source model and then applied fast Gauss transform with Hermite expansion to estimate the fogging effect map of the target layout. Subsequently, this estimation scheme speeds up the fogging effect evaluation by approximately 30 times faster than the convolutional-based methods. Then they proposed an analytical placement method to minimize the fogging effect variation during global placement and improve placement quality through fogging-effect-based legalization and detailed placement. In comparison to their previous work [24], they improved the wirelength quality and running time, while minimizing the fogging effect variation by 35%. In addition to these two papers,

these authors introduced an innovative placement method in [25] to control both the fogging effect and proximity effect simultaneously in the digital layout design. They formulated the optimization problem, then applied a new algorithm called proximal group alternating direction method of multipliers (ADMM) to solve the problem, which can control the chip density and minimize the wirelength, fogging effect and proximity effect. Their approach was 1.65 times faster and could reduce the fogging effect and proximity effect by 13.4% and 21.4% lower than the previous work, respectively.

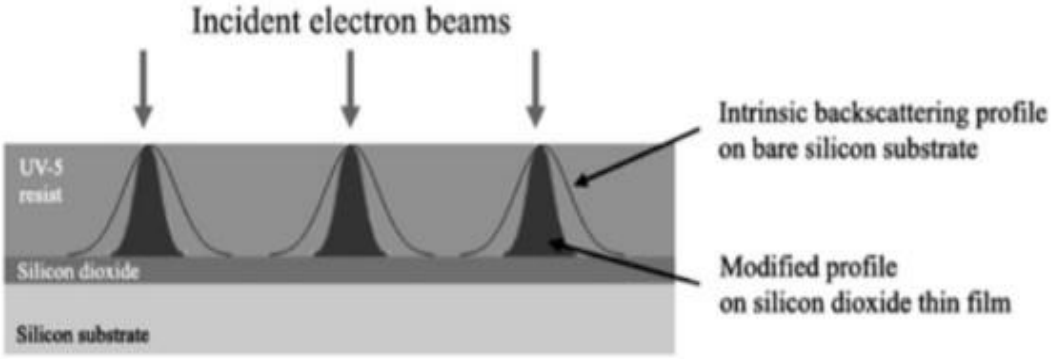


Fig. 7. Schematic of back-scattering profile for method used in [22].

2.4.2. Analog Placement Methods

It is observed that all the previous studies on the placement methods to minimize the fogging and proximity effects only focus on digital circuits. In this thesis, we are motivated to develop an electron-scattering-aware placement method for the analog part of a mixed-signal IC. In the literature, there are different categories of analog placement approaches that use optimization algorithms to minimize the cost function by properly placing the devices in different positions. By nature, an analog placement method has to embed the following two indispensable factors: representation and search engine. The representation

refers to a way how the placement method describes the location of modules in a layout. It could be based on exact location or relative location. The search engine stands for an optimization scheme that the placement method uses to drive the evolution to an optimal state. We can classify those existing placement approaches into two categories: 1) in terms of representation and 2) in terms of search engine.

2.4.2.1. Classification Criterion of Representation

The analog placement methods apply different representation techniques to define the location of modules within the given layout. The representation methods can be divided into two main categories: topological and absolute-coordinate-based representations.

The topological-representation-based placement (i.e., using topological floorplan representation for the layouts) is one common analog placement approach that often applies simulated annealing (SA) to minimize the cost function [26]. The most significant feature of the topological representations is their relative description of module locations in a layout. This type of placement schemes effectively considers the geometrical constraints, although the SA engine might be slow in solving some complicated placement problems, such as electron-scattering-effect-inclusive placement, because of random perturbations.

There are two types of floorplans: slicing and non-slicing. Different representation methods have been introduced to describe the floorplan under each category. In the slicing floorplan, the slicing structure is achieved by repetitively cutting rectangles horizontally or vertically and dividing them to smaller rectangles, which are the representatives of the

modules in a layout. However, almost all real layouts cannot be sliced as mentioned; so, they are non-slicing floorplans per se. The most popular topological representation methods used for non-slicing floorplans include: sequence pair (SP) [27], O-tree [28], B*-tree [29], and transitive closure graph (TCG) [30].

SP utilizes an ordered pair of module sequences (a, b) to represent the placement of modules in the given layout. The time complexity of packing in this approach is $O(m^2)$, where m is the number of modules. When it comes to TCG, this representation scheme applies the transitive closure graph to represent the geometric relationships among modules. A transitive closure of the directed acyclic graph G is shown as $\acute{G} = (V, \acute{E})$, where V and $\acute{E}$ are the sets of nodes and edges that connect the nodes, respectively. In this method, the floorplan is represented by two graphs: horizontal transitive closure graph C_h and a vertical transitive closure graph C_v, which describe the horizontal and vertical geometrical relationship between the modules, respectively. A unique transitive closure graph $\acute{G}$ can be derived from a given layout placement. Each module a_i is represented by a node n_i with width (height) as its weight in C_h (C_v). To create the TCG, if module a_i is on the left side of a_j, we construct the directed edge (n_i, n_j), which connects n_i to n_j by a directed array in C_h. In the same manner, the directed edge (n_i, n_j) is created in C_v from node n_i to n_j if a_i is on the top of a_j. Fig. 8 illustrates a floorplan and its corresponding transitive closure graph.

O-tree and B*-tree are two kinds of topological representations, which benefit from the tree structures to show the module position relationship. These two methods are faster

and more efficient compared to the previous approaches. Moreover, B*-tree is even faster and required 60% less memory than O-tree. Given a layout, to create the corresponding tree, the first step is to generate an admissible placement of the layout. A placement is admissible if it is compact and no module inside can move to the left or bottom side. And then starting from the most bottom left module as the root of the B*-tree, the left child node is the right module and the right child node of the root is its top module.

In all of topological-representation-based placement methods, the simulated annealing (SA) algorithm is most frequently applied to find the global optimum [31]. This optimization method emulates the annealing process in metallurgy and tries to approach the global minimum. This optimization method has faster convergence to global minimum compared to the brute-force methods or general heuristic methods, while it is able to avoid trapping in local minimums compared to other optimization methods (such as the conjugate gradient method). However, it applies random perturbation to optimize the objective function, which results in longer computation time.

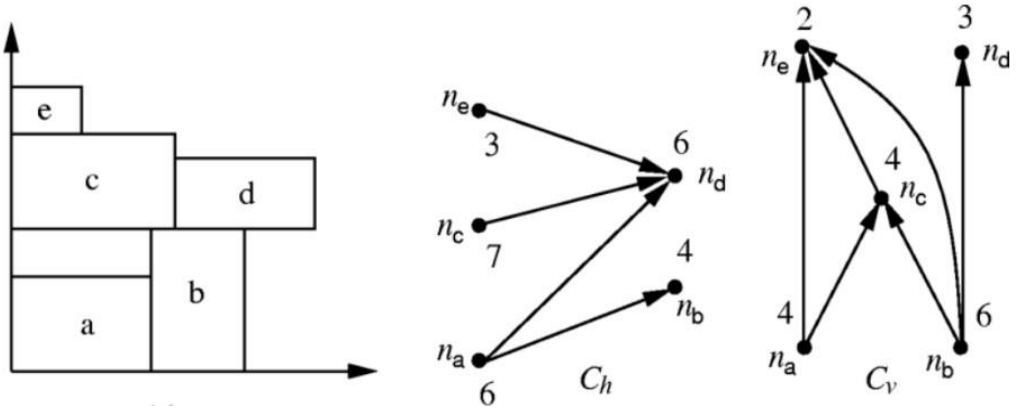


Fig. 8. The layout and its corresponding TCG.

Recently, different B*-tree-based analog placement approaches have been introduced to handle geometric constraints. In [26], the authors introduced QB-tree representation model to reduce the time complexity while considering different constraints. QB-tree is a hierarchical method that combines B*-tree and quadtree. To construct the QB-tree, they first repeatedly divide area to four sub-regions with a vertical and horizontal line crossing along with pre-placed modules' boundaries. This process of subdividing will continue until a maximum of one preplaced module remains. These sub-regions are represented by child nodes in QB-tree, whose parents are the larger sub-region where they are located. Then the B*-trees representing the modules in each sub-region are constructed. The root of B*-tree is connected to the random leaf node of quadtree, which means those modules are placed in the corresponding sub-region. By using the QB-tree, this work could achieve linear time complexity for module packing. Therefore, this method gained better placement quality, faster running time, and more constraints to be handled. However, subdividing of layout may lead to cutting modules and put them in different sub-regions; so, handling this issue is challenging in QB-tree placement.

In terms of representation, the second group of placement methods is based on absolute coordinates, which define the location of each module by its exact coordinates in the process of optimization [32]. This kind of placement methods has significantly large configuration space including infeasible placements. Therefore, the legalization and detailed placement, which can slow down the placement process and significantly increase the runtime, are required to eliminate the overlaps of modules and improve the placement quality. The absolute coordinate-based placement method proposed in [33] formulates the

placement problem as an exact and accurate mathematical modeling process in order to apply an optimization algorithm to minimize the cost function and find the optimum location of modules in the layout. In particular, the authors utilized the nonlinear analytical analog placement to handle layout dependent effects (LDE) including well proximity, length of oxide diffusion, and oxide-to-oxide spacing, which change the electrical characteristics of transistors and lead to performance degradation. In a global placement phase, they minimized the cost function by the nonlinear conjugate gradient technique to tackle LDEs, symmetry constraints, wirelength, and overlaps of modules. A new legalization called ILP-based method was introduced to determine the finger number of modules, since the non-integral finger number is not valid in analog placement. In detailed placement phase, the aim of placer is completely removing overlaps. However, after justification to minimize the overlaps, the other constraints should be considered in optimization. Therefore, the legalization and detailed placement parts significantly increase the running time.

The absolute-coordinates-based placement methods are straightforward, accurate, and efficient for simple layouts. However, it is hard to achieve acceptable performance in a placement that requires to consider complex constraints during placement, especially when calculating or manipulating those constraints is time-consuming (such as fogging and proximity effects) for large circuits. This is mainly due to the extremely large configuration space that absolute coordinates may incur when being used as a representation method for describing layout. In practice, using exact coordinates of modules' location in the placement optimization process not only leads to large configuration space, but also brings

about the messy overlaps among modules that require complex post-processing operations for legal final placement output.

2.4.2.2. Classification Criterion of Search Engine

The existing placement approaches in the literature can be also classified by their search engines applied for optimization. They can be categorized into three types: heuristic-based (e.g., SA, genetic algorithm (GA), turbo search, particle swamp optimization, etc.), mathematical-programming-based (e.g., conjugate gradient), and artificial-intelligence-based methods.

Simulated annealing (SA) is a heuristic-based optimization algorithm, which is widely used in analog layout placement to find the global optimum. Lu *et al.* [34] proposed a new topological-representation-based placement method utilizing SA search engine to address different geometric constraints in FinFET technology. They used WB-tree to represent the placement of modules. The WB-tree representation method combines window mesh data structure and corner stitching compliant B*-tree (CB-tree) [35] to handle both traditional and FinFET related constraints. To construct the WB-tree, they first subdivide the given layout into identical windows, whose size is defined by the users. To create the window mesh, the adjacent nodes that correspond to window regions are connected to each other. Then, the CB-trees are generated for all modules and the root of each tree is randomly connected to one node in window mesh. Next, they use an SA algorithm to perturb the modules for reaching the best placement solution. WB-tree based placement is able to handle a variety type of geometrical constraints in a reasonable runtime. However, this

method applies corner stitching as a data structuring techniques that significantly increases the required memory space for this placement approach.

The conjugate gradient (CG) is one of the most popular mathematical-programming-based placement techniques, which is used as a search engine for analog placement optimization. In [70], the authors proposed a high-quality mixed-size analytical placer considering preplaced blocks and density constraints. They applied conjugate gradient (CG) to minimize the cost function in global placement stage. The cost function is comprised of density and wirelength, supported by a novel log-sum-exp wirelength model. Two major contributions in this study, which result in a significant speedup in the placement process, include using dynamic step-size control for CG and utilizing look-ahead legalization technique during the global placement. The experimental results of this work show better performance in terms of faster runtime and better wirelength minimization.

Recently nonlinear optimization approaches have been used for layout placement in VLSI and showed significantly high performance. In [36], the authors introduced a novel nonlinear placer called elfPlace to optimize the wirelength and density constraint. This placer applies augmented Lagrangian method (ALM) for minimizing the cost function. As results show, their approach is better than four state of the art placers in wirelength minimization with shorter running time. In addition, elfPlace is able to optimize the routability and pin density, which noticeably improve the placement quality.

Artificial intelligence (AI), which has been applied to many sub-areas of electronic design automation (EDA), was promoted to develop new methods for solving analog

placement problems. Gusmao *et al.* in [37] introduced a semi-supervised artificial neural network for analog IC placement. They used 125,574 different floorplans as the data set to train the artificial neural network. In the training process, they used mean squared error as the loss function for the first 100 epochs to minimize the cost function considering area, current flow, overlaps, and symmetry constraints. For the training in the next 1000 epochs, they applied a novel topological-loss function (TLF), which considers the geometric constraints. By utilizing this approach, the optimal placement of a layout is predicted in a few milliseconds. However, this method needs a huge number of data sets for training, which brings a barrier to general usage since providing such training data for the placements considering constraints is not always accessible.

A new machine-learning-based placement was introduced in [38], which applies a transfer learning approach to automate and speed up the chip placement. Using transfer learning enables the placer to learn from the previous experience of placements, which leads to more accurate and faster placement for unseen circuit structures. These authors defined the placement optimization as a reinforcement learning problem and trained an agent to intelligently place the modules in the layout canvas to minimize the power and wirelength, and improve the performance. Their approach was able to find the optimal placement for an unseen circuit after 6 hours of training. However, this placer improves through the learning of previous placements; so, it can not be efficient for primary placements and it takes long time to achieve high performance quality.

Most recently, Ahmadi and Zhang [39] proposed an AI-based analog placement method that uses deep Q-learning to train an agent, and intelligently chooses the best action for each module. Their method is 77 times faster than the traditional analytical placement approaches while reaching approximately the same electrical performance and chip area. Nevertheless, the number of possible states in their environment of the reinforcement learning approach is quite high, which may slow down the placement process in more sophisticated layout designs.

2.5. Our Contributions

Different from the digital circuit domain with the studies of the fogging and proximity effects already undertaken, so far we have not seen similar research in the context of mixed-signal or analog ICs. Therefore, we are mainly aimed to focus on the fogging and proximity effect control on the analog part within a mixed-signal circuit in this thesis. For this purpose, we have developed a reinforcement learning (RL) placement method using the topological representation. Such an artificial intelligence method cannot only facilitate the automation in the placement evaluation and action selection process without a need for human's involvement, but also boost the search efficiency thanks to using a topological representation (i.e., B*-tree in our implementation).

In our reinforcement learning placer, the RL environment is a B*-tree representing the module positions in a layout, while the RL agent is a deep neural network (DNN) with reward calculated by our specified cost function. Below we list the main contributions of our research conducted in this thesis work:

- We closely consider the impact of the fogging and proximity effects on analog portion of mixed-signal circuits.
- We propose an RL-based placement method to minimize the variations of the fogging and proximity effects, which liberates the designers from their mandatory expertise or rough estimation.
- We introduce a new RL placer in this study, which is based on a topological representation scheme resulting in much faster placement operation.

2.6. Summary

In this chapter, we reviewed the fundamentals of the next generation lithography methods and their challenges. In addition, we explained the fogging and proximity effects, which are two major concerns in the electron beam lithography. To review the previous works, the studies on the fogging and proximity effects were discussed in detail. Moreover, the previous works on different placement methods were reviewed in term of two classifications: representation methods and search engines. Absolute coordinate-based and topological representation were discussed for the first category, while simulated annealing, conjugate gradient and AI-based placement approaches were reviewed as the search engine category. In the next chapter, we will introduce the variation modelling of the fogging and proximity effects based on their energy distribution functions.

Chapter 3
Modelling of the Fogging and Proximity Effects in Mixed-Signal ICs

3.1. Introduction

Since EBL as a next-generation lithography technology has been used to manufacture devices with sub-10nm feature size at lower cost, new effects like the fogging and proximity effects are commonly existent due to process variations. In the EBL technology, in order to create a feature, electron beams are first emitted from an electron source. After going through a set of lenses and apertures, they directly write the pattern onto resist-coated substrate. Then this exposure step is followed by etching and deposition process to make the desired feature patterns [40]. However, as EBL is a high-resolution pattern formation method, some undesired exposures, which occur during the lithography, would result in a feature dimension change [7]. Those undesired exposures, which are caused by electron scattering phenomenon that can deposit energy in resist, lead to pattern distortion and critical dimension (CD) errors.

As described in Chapter 2, the group of scattered and back-scattered electrons produces the proximity effect while the re-scattered electrons result in the fogging effect.

These two effects cause uncertainty and errors in electronic device fabrication. Thus, they limit the capability of EBL to be used as a dominant lithography method in high volume manufacturing of semiconductor industry. Therefore, the researchers in academic laboratories and semiconductor fabrication companies have been researching into these two challenges to find a solution to minimizing or completely eliminating them.

At the first step, the characteristics of the fogging and proximity effects should be studied to extract their mathematical models for correction and minimization of these effects by using any solutions. Based on previous studies, it is learned that the energy distribution model of the fogging and proximity effects is a Gaussian function with maximum intensity at the center of primary e-beam. Then, the analytical model of the fogging and proximity effects is extracted to be used in our specially designed placement algorithm in order to minimize the variation of these effects. Such a model can compute and estimate the variations of the fogging and proximity effects among the modules in sensitive devices of the analog circuits. In this way, we will seek to protect them from the fogging and proximity effects induced by other modules in both digital and analog parts of mixed-signal ICs.

The remaining of this chapter is organized as follows. The energy distributions of the fogging effect and proximity effect are described in Sections 3.2 and 3.3, respectively. Section 3.4 discusses the fogging and proximity effects in the general mixed-signal ICs. In Section 5, our variation modelling of the fogging and proximity effects is introduced. Finally, Section 6 draws a conclusion of this chapter.

3.2. Energy Distribution of the Fogging Effect

Some previous works studied the fogging effect modeling. Based on simulations and experiments that had been done, the energy distribution of the fogging effect was modeled as a Gaussian point spread function (PSF) for a given EBL process. There are some software programs and experiments to characterize the behavior of electrons when they emit from an electron gun and are exposed to resist and substrate there. Fig. 9 demonstrates the result of one simulation of electron beam lithography on CHARIOT [41], which is one popular Monte Carlo software. The figure obviously shows the scattering of electrons and their deviations after being exposed to resist and substrate. The blue lines are primary and scattered electrons, while the red ones are back-scattered electrons. As written on the bottom of the figure, the back-scattering coefficient is 0.08. Based on the simulation, the number of the scattered electrons resulting in the fogging effect is much less than the ones leading to the proximity effect, which actually causes the low intensity of the fogging effect in comparison to the proximity effect. However, the fogging effect is featured by its long range nature and the accumulative amount of this effect from all modules in the chip on one target point is still considerable and should be controlled in a better way.

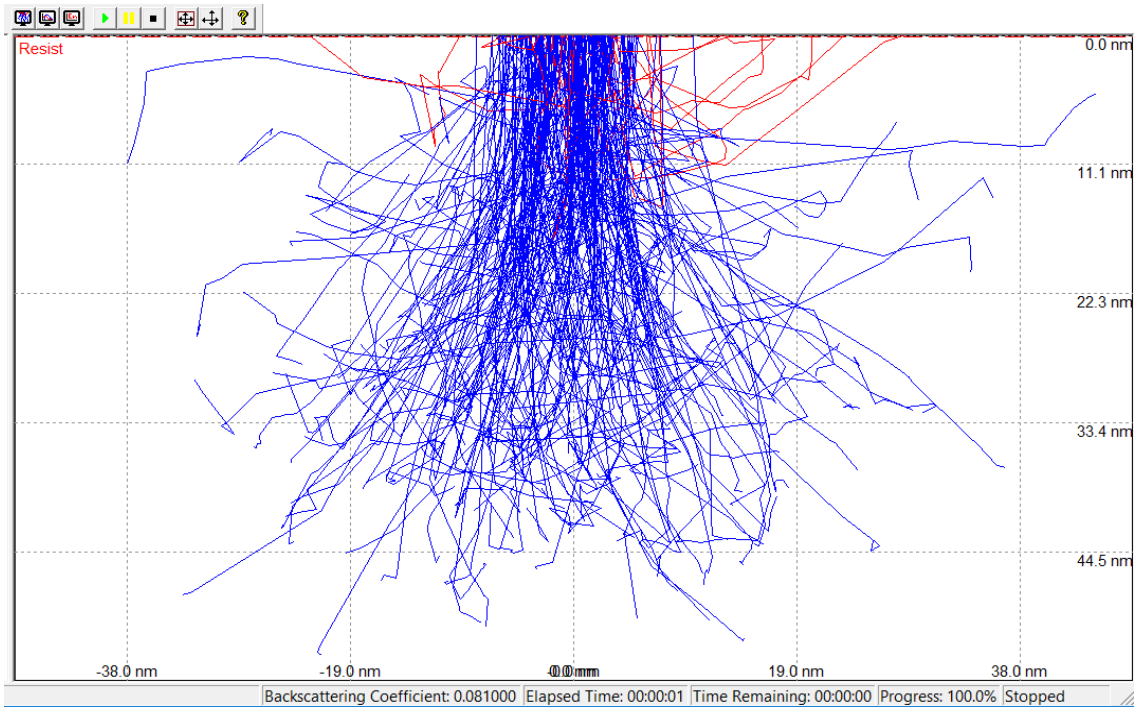


Fig. 9. Simulation of electron beam lithography by CHARIOT.

To model the energy distribution of the fogging effect during the simulation, the charging and absorbed energy in resist caused by the re-scattered electrons should be evaluated and modeled. However, there is a challenge in the simulation of the fogging effect that is caused by the long-range aspect of the effect. The software or experiment setup must be able to simulate the electron trajectories, which are in a long distance from the primary e-beam. The updated CHARIOT software is able to consider and evaluate the back-scattered electrons scattering far away from the center of beam, scattering of the electrons returning from the bottom of the lens, and paths of the re-scattered electrons after being exposed and scattering into the resist.

The simulation above was done for a 2,500 square micrometer resist, which had 150 nm thickness. The e-beam dose and voltage was 28 $\mu C/Cm^2$ and 50 kV, respectively [42].

The charging and absorbed energy in the resist was evaluated for different working distances and resist depths. As Fig. 10 shows, the energy absorption in the regions near the surface of the resist is much higher than the places near the bottom of the resist, which has the coordinate of 0 on the resist depth axis. In contrast, the simulation results for the primary and back-scattered electrons illustrate even absorbed energy for different depths. It clearly suggests that the re-scattered electrons have much less energy than the forward and backward-scattered electrons; so, this observation demonstrates the low intensity of the fogging effect in EBL. In addition to intensity of the fogging effect, the simulation results for four different working distances (defined as the distance between the lens and resist) show that as the working distance decreases, the absorbed energy range of the fogging effect grows approximately in a linear style.

The main purpose of those simulations is to model the energy distribution of the fogging effect. In Fig. 11(a), the absorbed energy in the resist, which is caused by the primary and re-scattered electrons in the EBL process, is shown. The maximum intensity that occurs in the center of the target point is caused by the primary exposure of electrons into the resist. However, Fig. 11(b) only demonstrates the energy absorption from the re-scattered electrons, which can be approximated as a Gaussian form (shown in a red line). To accurately estimate the Gaussian parameter, several simulations and experiments were done for different working distances. Finally the Gaussian function of the fogging effect energy distribution is modelled as follows [43]:

$$E_{fog}(x,y) = \frac{a}{w\sqrt{2\pi}} e^{-\frac{\sqrt{x^2+y^2}}{2w^2}}, \tag{2}$$

where E is the energy absorption in the resist at the point (x, y), a is a constant, w is the fogging effect range, and $\sqrt{x^2 + y^2}$ represents the distance of the target point from the center of the primary electron beam.

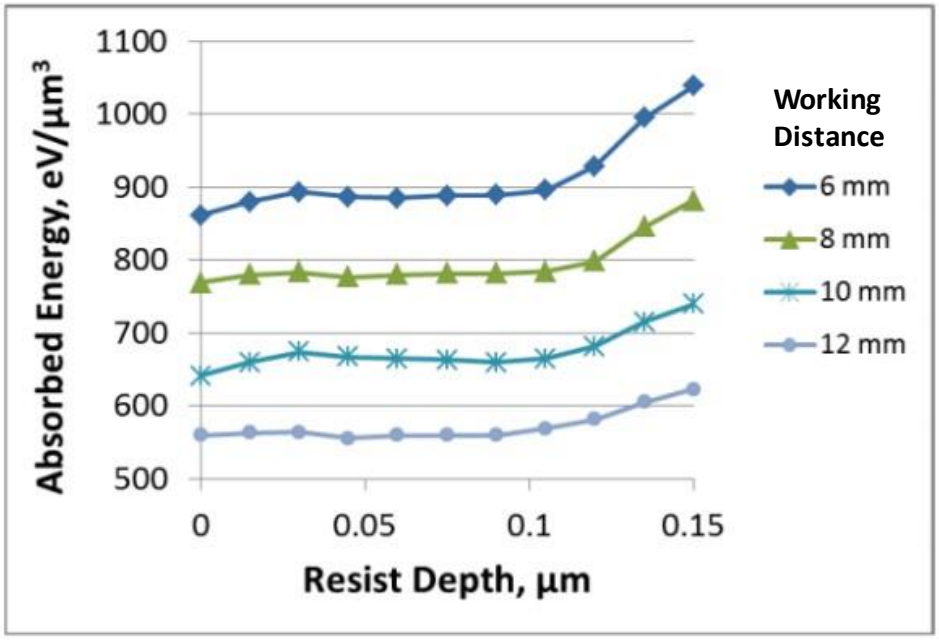
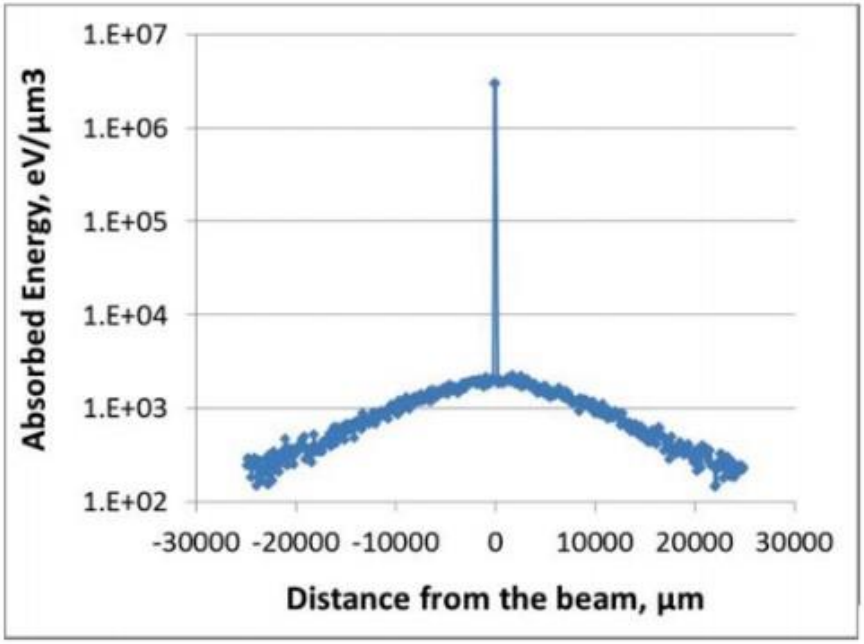


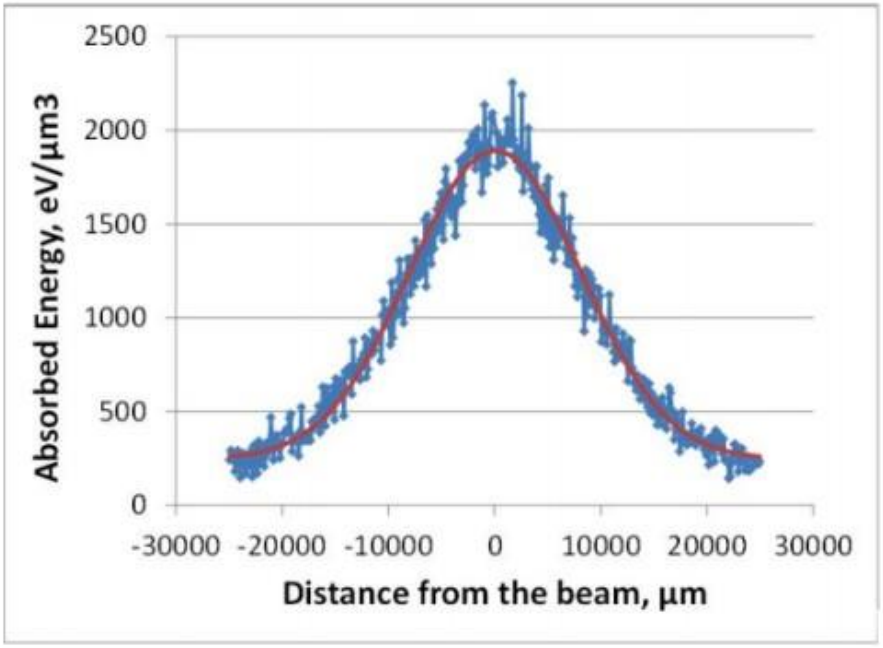
Fig. 10. Energy absorption in different resist depths for different working distances [42].

The energy distribution model of the fogging effect, which is used in most of the previous works, is the Gaussian point spread function (PSF). It is extracted from Eq. (2) and expressed below [7]:

$$f_{fog} = \frac{v_F}{{\gamma_F}^2} e^{-\frac{r}{{\gamma_F}^2}}, \tag{3}$$



(a)



(b)

Fig. 11. The energy absorption in the resist. (a) the total absorbed energy (b) the absorbed energy caused by the re-scattered electrons [43].

where γ_F and v_F are the range and weight of the fogging effect, respectively, and r is the distance of target point from the center of the primary electron beam. Fig. 12 is the PSF of the fogging effect simulated in MATLAB. From the extracted information of the simulation results, it is estimated that the absorbed energy in the resist by undesired exposure of the re-scattered electrons is approximately 5% of the total energy absorption, which is reasonably significant.

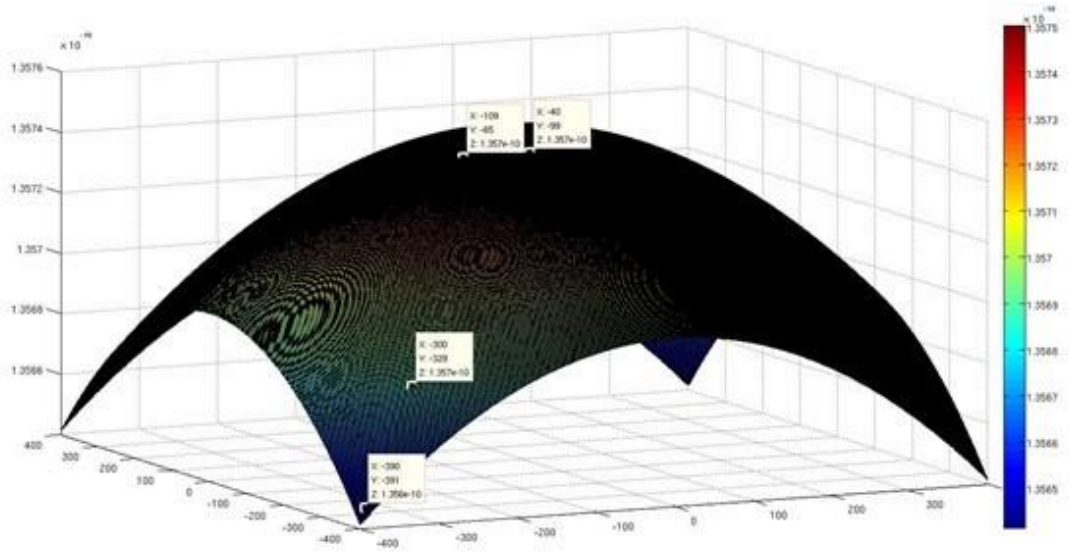


Fig. 12. The PSF simulation of the fogging effect in MATLAB.

3.3. Energy Distribution of the Proximity Effect

The EBL technology applies e-beams that can be focused down to 0.5 nm diameter [44]. So, it is assumed that the resolution of the electron beam lithography can reach a few nanometers. However, this is completely theoretical since there are always some process variations that may limit the resolution of EBL, especially in the new technology node where the feature size is significantly small and the pattern density is considerably high.

One of such major process variations is the proximity effect, which is caused by scattered and back-scattered electrons. As such, the proximity effect suffers from two kinds of scattering:

- Forward scattering: electrons are transmitted from the electron gun as a focused e-beam in a diameter of about 0.5 nm. However, they experience a little deviation before reaching the resist surface and cause a wider e-beam, which is due to the two phenomena of lens aberrations and electrostatic interactions between electric charges called Coulomb interaction. In addition, the electrons diverge at more angle after reaching and passing through resist and substrate by collision with other particles in the resist and substrate. The range of this scattering is approximately from 2 nm to 40 nm, which is related to lithography process characteristics such as lens aberration, e-beam dose, and the energy accelerating the electrons at the beginning of transmission from the source. This kind of electron deviation, which leads to a sever critical dimension (CD) error, is called *forward scattering*.
- Backward scattering: The electrons entering the resist and substrate are attracted or forced back when getting close to an atom, which changes the trajectory of that electron. Some of these scattered electrons will go up toward the resist if they have enough kinetic energy. Therefore, they expose the resist from the bottom side with a distance of a few micrometers from the primary e-beam exposure point. Moreover, there are interactions between the e-beam electrons and the electrons inside the substrate material. If the collision

between these two kinds of electrons provides enough energy, it causes the second-generation electrons, which are called secondary electrons. These electrons also can rise up and hit the bottom of the resist if they have enough energy. These two kinds of electron scatterings in EBL are called *back-scattering*.

These two kinds of scatterings would cause a noticeable variation in the EBL technology by changing the feature size or generally speaking CD error. The electrons expose the resist and deposit an undesired energy inside the resist, which leads to non-uniform CD variation cross the chip after the etching process. Fig. 13 illustrates the energy distribution caused by the forward and backward scatterings in the resist; where E, D, and T are electron energy, e-beam diameter, and thickness of polymethyl methacrylate (PMMA) resist. As shown, the back-scattered electrons lead to a long range effect called *backward proximity effect* (BPE), while the forward-scattered electrons have lower angle deviation and subsequently result in a shorter range effect called *forward proximity effect* (FPE). The FPE has much higher intensity than BPE while the range of BPE is about double compared to that of FPE.

As the proximity effect is originated from two sources, its energy distribution is the sum of those short and long range effects, which is expressed in Eq. (4):

$$E(x,y) = ShortRange(x,y) + LongRange(x,y)\ . \tag{4}$$

To apply the proximity effect constraints in our placement approach, there is a need for a model that can evaluate and estimate the proximity effect. From Fig. 13 and Fig. 14, it can be assumed that the proximity effect can be modeled by two Gaussian functions. The

most straightforward and accurate analytical model that approximates energy deposition function due to the proximity effect is defined as [45]:

$$E_{pro}(r) = c_f e^{-\frac{r^2}{B_f{}^2}} + c_b e^{-\frac{r^2}{B_b{}^2}}, \tag{5}$$

where c_f and c_b are two constants, r is the target point distance from the center of the primary e-beam, and B_f and B_b are the range parameters of the forward and backward proximity effects, respectively.

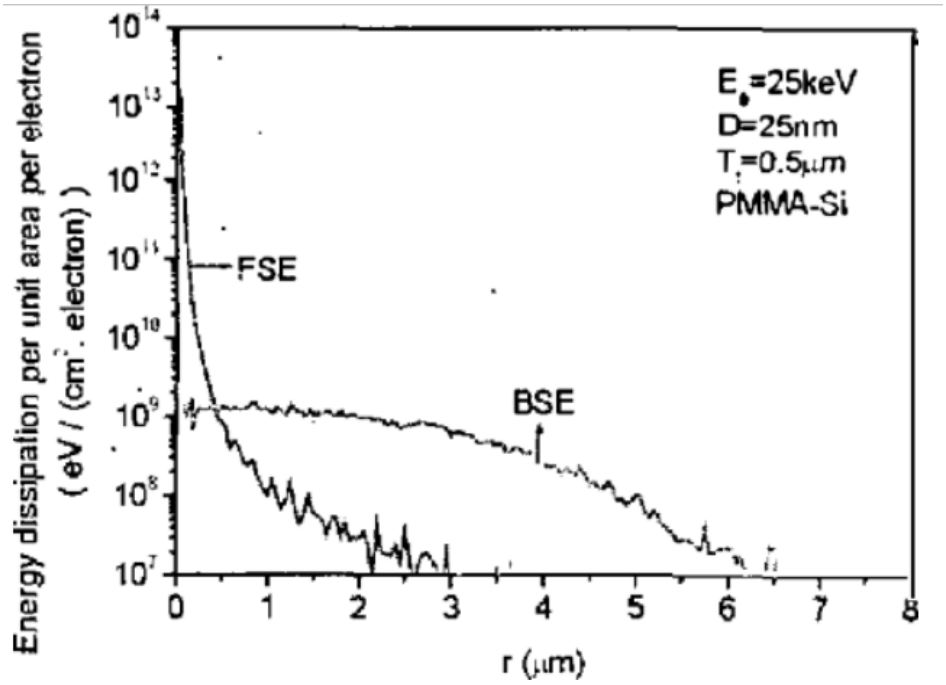


Fig. 13. Energy distributions by forward scattered electrons (FSE) and backward scattered electrons (BSE) [46].

However, some new studies utilized a PSF approximation function as a mathematical model to evaluate energy distribution related to the proximity effect. Based on different experiments and simulations by using some software programs, such as Monte Carlo, the point spread function (PSF) function of the proximity effect is given by [47]

$$f_{pro} = \frac{1}{\pi(1+\eta)}\left(\frac{1}{{B_f}^2}e^{-\frac{r}{{B_f}^2}} + \frac{\eta}{{B_b}^2}e^{-\frac{r}{{B_b}^2}}\right), \quad (6)$$

where η is the ratio of back-scattered energy to forward-scattered one, and B_f and B_b are the range of FPE and BPE with the amounts of 0.06 μm and 30 μm, respectively. Here Eq. (6) consists of two parts, the first one is due to PSF of the forward proximity effect and the second part is the contribution from the backward proximity effect.

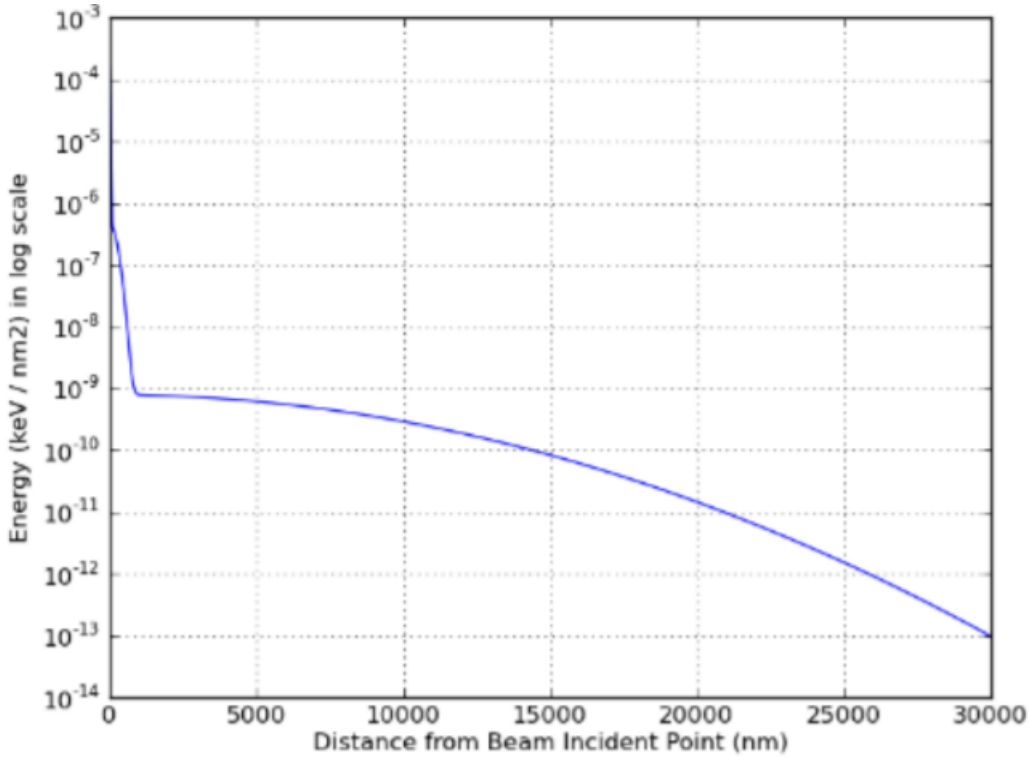


Fig. 14. The energy deposition distribution due to the proximity effect [48].

3.4. Fogging and Proximity Effects in Mixed-Signal ICs

The mixed-signal integrated circuit is an IC that consists of both analog and digital portions fabricated on a single chip [49]. Nowadays these ICs are actually used in almost

all the electronic devices around us, such as smartphones, digital cameras, house appliances and healthcare digital instruments. In this regard, virtually a majority of modern digital devices need to enclose mixed-signal ICs because these devices need an interface to connect to the real world, where the signals are analog by nature. Therefore, it is indispensable to convert the analog signals to digital ones at the input end and vice versa at the output end. Obviously any manufacturing error and undesired variation in fabrication process may lead to characteristic deviation or malfunction of electronic components (e.g., transistors), and then in turn performance degradation or even failure of such mixed-signal ICs.

In analog portion of the mixed-signal circuits, there are always some sensitive devices (e.g., differential pairs or matched current mirrors), in which any variation from the nominal values may lead to severe device characteristic deviation and in turn circuit performance degradation. In addition, some manufacturing process variations result in transistor parameter change. For example, one of the most important transistor parameter is the threshold voltage, which defines the characteristics and impacts on the performance of transistors. The variation of this parameter due to uncertainty in fabrication processes is given by

$$\sigma_{V_{th}} = \left(\frac{\sqrt[4]{4q^3\varepsilon_{Si}\emptyset_B}}{2}\right)\left(\frac{T_{ox}}{\varepsilon_{ox}}\right)\left(\frac{\sqrt[4]{N_{ch}}}{\sqrt{W_{gate}L_{gate}}}\right), \quad (7)$$

where q, ε_{Si}, $\emptyset_B$, T_{ox}, ε_{ox}, N_{ch} and W_{gate} are electron charge, silicon permittivity, bulk potential, gate oxide thickness, gate oxide permittivity, carrier density, and gate width, respectively. As expressed in the equation above, the threshold voltage largely depends on

L_{gate}, which is the length of the gate, among others. Based on the previous explanation about the fogging and proximity effects, these effects cause CD error that subsequently changes the transistor critical geometry sizes (such as gate length) non-uniformly in the chip. Therefore, it can be inferred that the proximity and fogging effects would surely cause variations in the performance of the mixed-signal ICs. Some countermeasure schemes have to be resorted to if one intends to protect the mixed-signal ICs from such scattering effects in the EBL technology.

To address the challenges due to the fogging and proximity effects in the placement of the analog portion within a given mixed-signal circuit, we should analyze these effects separately in the context of analog performance. The effective range of the proximity effect normally is less than 30 μm with high intensity [7]. In the mixed-signal circuits, the analog portion is often separated from the digital part to avoid signal/noise interference. Therefore, for the sake of simplicity, we can assume that the digital part would cause insignificant proximity effect on analog portion within a mixed-signal IC. That is to say, the proximity effect is considered as a local effect within the analog domain itself. So, in this work we will only consider the proximity effect of the modules from the analog portion in our mixed-signal placement method.

However, the situation of the fogging effect is different. Although the intensity of the fogging effect is much less than that of the proximity effect, the fogging effect will become noticeable when the total e-beam dose exposed to the wafer is considerable. As a matter of fact, this is true in the industrial EBL manufacturing process, where the mixed-signal ICs

are fabricated together on the wafer. Due to this very reason, all of the previous studies on the fogging effect were focused on digital circuits. Similarly, when we consider an analog circuit as part of a mixed-signal IC, there are a huge number of fogging effect sources from both digital and analog blocks, which make this effect non-negligible [50]. Moreover, the fogging effect is a long range effect that can go up to one millimeter. Thus, we need to take into account the fogging effect originated from the digital blocks (e.g., standard cells) onto the modules in the analog portion of a mixed-signal IC. To examine and demonstrate the statements above, one experiment was done in [50], which applied EBL in a specimen chamber with a scanning electron microscope (JSM-35CF) and utilized a Faraday cup to measure the current induced by electron beams. Based on the experimental results, the exposure intensity distribution caused by the proximity and fogging effects could be calculated by using the equation below:

$$EID(r) = \frac{1.2 \times 10^{-2}.r^{-2.43} \times 10^{-31}.r^{-7}}{1.2 \times 10^{-2}.r^{-2.43} + 10^{-31}.r^{-7}} + 4 \times 10^{13}.exp\left[-\left(\frac{r}{2.3 \times 10^{-6}}\right)^2\right] \quad (8)$$

$$+ \frac{4\times10^{6}.r^{-0.25} \times 1.7\times 10^{-1}.r^{-4}}{4\times10^{6}.r^{-0.25} + 1.7\times 10^{-1}.r^{-4}},$$

where r is the distance from the electron primary beam exposure. Based on Eq. (8), for one electron beam emission the fogging effect has considerably low intensity compared to the proximity effect, which suggests that we might ignore it. However, another experiment was done to measure the fogging effect on a 100 μm^2 area placed in the center of 100 mm^2 area, which is shown in Fig. 15. The entire space was irradiated by 100 $\mu C/cm^2$ dose electron beams except for the area in the center where we want to evaluate the fogging effect. The experimental results showed that the absorbed energy induced by the fogging

effect from the large area, which was considerable compared to the amount induced by the proximity effect, might have huge contribution to the CD error. Therefore, we draw a conclusion that we have to consider both the fogging effect and the proximity effect in our placement method to ensure the performance integrity of the mixed-signal circuits fabricated in the EBL technology.

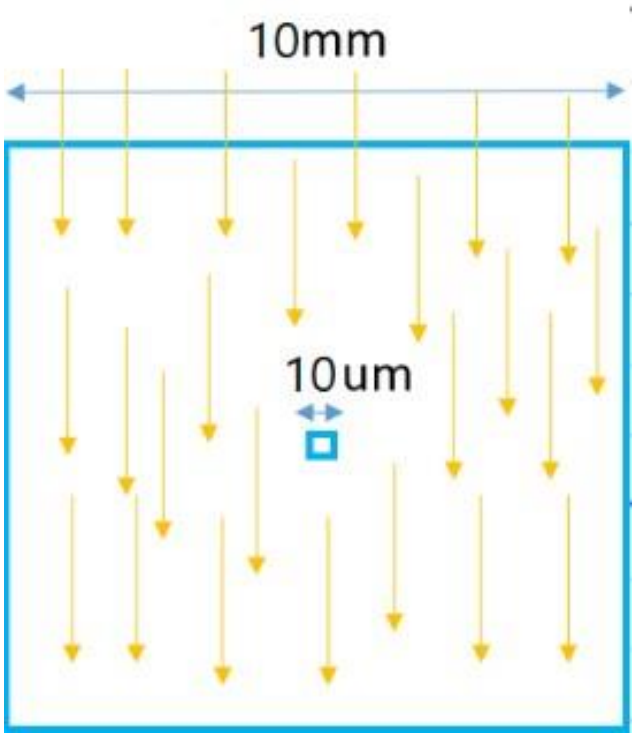


Fig. 15. The schematic of experiment in [50].

3.5. Our Variation Modelling of the Fogging and Proximity Effects

One of the major concerns in the fogging and proximity effects studies is the critical dimension (CD) error caused by undesired exposure of scattered, back-scattered, and re-scattered electrons. Moreover, analog circuits always contain sensitive devices, such as

symmetric transistors in differential pairs or matched current mirrors [51], which we want to protect from any process variations and then in turn characteristic differences. Therefore, in this work we aim to minimize the critical dimension variations between sensitive transistors induced by the fogging and proximity effects.

The pattern density map $I_d(x,y)$ is convolved with the PSF of the fogging effect or proximity effect in order to evaluate the fogging effect or proximity effect over a layout, respectively. Thus, the fogging effect map $I_f(x,y)$ and proximity effect map $I_p(x,y)$ are computed with accurate approximation as follows:

$$I_f(x,y) = I_d(x,y) \otimes f_{fog}(x,y) \,, \tag{9}$$

$$I_p(x,y) = I_d(x,y) \otimes f_{pro}(x,y) \,, \tag{10}$$

where f_{fog} and f_{pro} are PSF of fogging and proximity effects from Eq. (3) and (6), respectively. This computation is very time-consuming and takes large memory due to the convolution operation. To solve this problem, previous works used fast Gauss transform to estimate fogging effect over a layout with a highly accurate approximation [23] [24]. Based on this technique, the fogging effect can be calculated for every target point on the layout by summing up Gaussian distributions centered at the sources of the fogging effect,

$$G_{t_i}^{S}(s) = \sum_{j=1}^{N_s} q_j e^{-\frac{|t_i - s_j|^2}{\delta}}, \tag{11}$$

where δ is a positive constant, and t_i is a target point from the set of targets, $T = \{t_1,\ t_2, \dots, t_{N_t}\}$ on the layout where we want to calculate the fogging effect. In addition, s_i is a source of the fogging effect from the set of $S = \{s_1,\ s_2, \dots, s_{N_s}\}$; q_j is the weight of the fogging effect source, which is the pattern density of the source. Thus, a larger pattern area

would lead to a larger weight of the source. In addition, $|t_i - s_j|$ is the distance between the target and source points. In the same way, the forward and backward proximity effects can be calculated by [25]

$$G_{t_i}^{K}(k) = \sum_{l=1}^{N_k} m_l e^{-\frac{|t_i - k_l|^2}{\delta}}, \tag{12}$$

$$G_{t_i}^{C}(c) = \sum_{u=1}^{N_c} z_u e^{-\frac{|t_i - c_u|^2}{\delta}}, \tag{13}$$

where k_l and c_u are the sources of forward and backward proximity effects from the set of $K = \{k_1,\ k_2, \dots, k_{N_k}\}$ and $C = \{c_1,\ c_2, \dots, c_{N_c}\}$, respectively. In addition, m_l and z_u are the weights of the forward and backward proximity effects, respectively. Accordingly, as Eqs. (11), (12) and (13) indicate, the fogging and proximity effects on a target point depends on two factors: 1) the distance from other modules, and 2) the pattern density among them.

Assume in an analog circuit we have a set of transistor pairs, $P = \{p_1,\ p_2, \dots, p_N\}$, with different sensitivities and that we want to minimize the variations of the fogging and proximity effects between two transistors in each pair. Therefore, this mathematical minimization formulation can contribute to less CD variation for each device pair, leading to less device characteristic deviation and in turn better circuit performance. The variation function of the fogging effect for each device pair can be expressed as:

$$F_{p_i} = \left| \sum_{j=1}^{N_S} q_j e^{-\frac{|t_{p_i 1} - s_j|^2}{\delta}} - \sum_{j=1}^{N_S} q_j e^{-\frac{|t_{p_i 2} - s_j|^2}{\delta}} \right|, \tag{14}$$

where $\{s_i\}$ in our placement methodology is all the devices of both analog and digital portions in mixed-signal circuits that cause the fogging effect on the given sensitive device. Moreover, p_i1 and p_i2 are two transistors in the sensitive pair i. Similarly, the variation functions of the forward and backward proximity effects can be calculated by using the following equations:

$$FP_{p_i} = \left| \sum_{l=1}^{N_k} m_l e^{-\frac{|t_{p_i1} - k_l|}{\delta}} - \sum_{l=1}^{N_k} m_l e^{-\frac{|t_{p_i2} - k_l|}{\delta}} \right|, \tag{15}$$

$$BP_{p_i} = \left| \sum_{u=1}^{N_c} z_u e^{-\frac{|t_{p_i1} - c_u|}{\delta}} - \sum_{u=1}^{N_c} z_u e^{-\frac{|t_{p_i2} - c_u|}{\delta}} \right|, \tag{16}$$

where $\{k_l\}$ and $\{c_u\}$ include only the modules in the analog portion of the given mixed-signal circuits, which induce the forward and backward proximity effects on the sensitive module p_{i1}, respectively.

Fig. 16 shows a pair of transistor modules (printed as blue rectangles) as two target points along with six other modules (displayed as red triangles) as fogging and proximity effects sources. To determine the variations of the fogging and proximity effects between these two target modules, we sum up the fogging and proximity effects induced by each of the seven other modules. The variation of the fogging and proximity effects is the difference between those two sum-ups. Therefore, the matched modules (e.g., symmetric ones) need to be placed as close as possible if we want to minimize the variation. However, in a circuit with many sensitive device pairs or matched module tuples, it is not always

possible to equivalently keep them close to each other to a great extent. So a better priority, which is the key to this very problem, should definitely go to the more sensitive modules. Therefore, our placement algorithm must be able to deal with this priority based on the sensitivity of modules in the analog circuits. As Eqs. (11), (12) and (13) show, two important factors define the fogging and proximity effects in each target point: the distance of the target point with other sources and the pattern density of the other sources, in other words, module locality. There are two ways to minimize the variations of the fogging and proximity effects between two modules in a sensitive pair: the first one is to place these modules in a shorter distance and as close as possible, while the second one is to locate them in a similar environment for better locality.

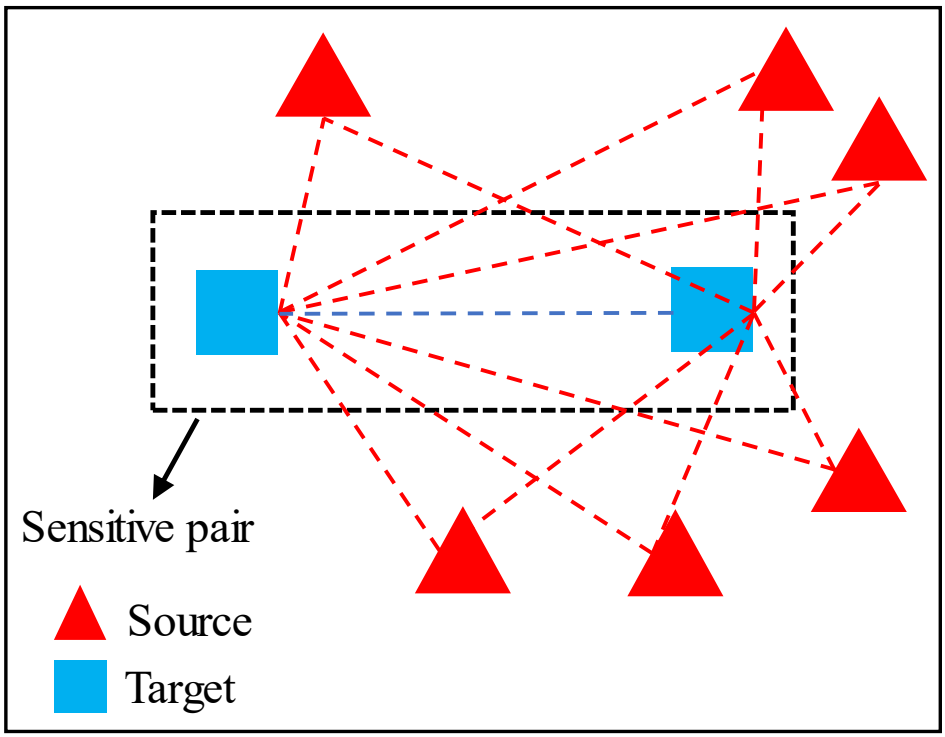


Fig. 16. Fogging and proximity effects induced on two target point (two modules in a sensitive pair) with six other modules as the sources of fogging effect.

3.6. Summary

In this chapter, the energy distribution functions for the fogging and proximity effects were reviewed and studied in the context of placement design for mixed-signal ICs. Then, we modeled those functions as a mathematical form called point spread function. Moreover, the fogging and proximity effects were discussed as a major concern in mixed-signal ICs fabricated in the EBL technology due to process variation. Finally, we proposed our modeling scheme for the variations of the fogging and proximity effects in the placement handling for mixed-signal ICs.

Chapter 4
B*-tree Based Placement with Reinforcement Learning

4.1. Introduction

As artificial intelligence (AI) is becoming more popular in various industrial applications, new AI-based placement methods have been introduced into electronic design automation (EDA). Such placers in the integrated circuit (IC) layout design are more effective and much faster than the traditional methods, and this advantage becomes more significant in complex and time-consuming placements. The sophisticated placement schemes need to consider different constraints, which may take long time and large memory size in computation [52]. Such new constraints include the fogging and proximity effects, which have been deemed as one of the designers' major concerns in the EBL technology, which is among the most promising next-generation lithography technologies good for feature sizes below 22nm [6].

Although a few previous works studied the fogging and proximity effects in the digital circuits, to the best of our knowledge, none of the existing research in the literature has considered the fogging and proximity effects in the design of mixed-signal or analog ICs.

In this study, we mainly aim to identify and evaluate the fogging and proximity effects on the analog portion of a mixed-signal circuit. To minimize the variations of such effects, we have developed a reinforcement learning (RL) placement method based on topological representation. Using an RL-based placer as an artificial intelligence (AI) method can help us eliminate the need for any special expertise from the designers. In addition, using a topological representation scheme (i.e., B*-tree in our implementation [53]) leads to significantly less states in the RL environment, which we believe is the major reason why our work is more effective and much faster than the previous works. In our reinforcement learning placer, the RL environment includes a B*-tree representing the module positions in a layout, while the RL agent is a deep neural network (DNN) with reward calculated by our specialized cost function.

4.2. Topological-Based Placement Method for Control of the Fogging and Proximity Effects

Given a set of modules with defined dimensions, the task of placement is to properly locate these modules in the chip area without any overlap (if not specified) and strive to reach the defined objectives while meeting any defined constraints. The common placement objectives include minimization of the occupied area and minimization of the interconnect wirelength, while the normal constraints include geometric constraints and electrical performance constraints. This process, which is also called *rectangular packing problem*, plays a critical role in IC physical design [54].

There are two kinds of representations indicating the status of layout placement: 1) absolute-coordinates-based representation and 2) topological representation. The absolute-coordinates-based approach has been popular and widely used in different studies. However, this method, which applies the exact coordinates of each module in the placement operation, may result in a large configuration space. This is because the total number of the possible module coordinates is close to infinity if no resolution of the floating-point numbers for presenting coordinates is applied. Moreover, for some constraints (e.g., symmetry constraints), it is not easy to design efficient and valid operations if using the absolute coordinate representation in the placement methods [55]. In addition, to avoid overlaps among the modules in the placement methods, an extra step called *detailed placement* has to be resorted to in the absolute-coordinates-based placement methods, which would significantly increase the complexity and execution time of the placement process. Therefore, it is challenging to utilize this type of representation in the layout placement operations to deal with the constraint control of the fogging effect and proximity effect among others, since it is too time-consuming to deliver effective and efficient solutions.

When it comes to the topological representation, it is a relative placement that indicates the topological relationship among the modules. Over the years, this representation has become more popular than the absolute-coordinates-based representation. Interestingly, most of the existing placement methods that use the topological representation selected to utilize simulated annealing as their optimization engine. In general, using the topological representation can improve the placement quality

and runtime efficiency, since it results in smaller configuration spaces and then in turn better search opportunities [34]. Moreover, recently different topological representation methods have been applied to tackle geometric constraints, such as symmetry [56] and proximity [35]. One of the most adopted topological representations is B*-tree, which is based on an ordered binary tree used for non-slicing floorplans [29].

4.2.1. B*-tree

In our placement method, B*-tree is used as a representation method indicating the states in our RL environment. B*-tree is one of the fast, efficient, and simple topological representation methods. So we first review the B*-tree data structure for placement representation below. Given a set of modules $B = \{b_1, b_2, \dots, b_{N_b}\}$, the B*-tree is an ordered binary tree representation of the compacted placement of N_b modules. In compacted placement, no module is able to move further towards the bottom or the left side. To build a B*-tree, starting from the bottom left corner module, which is the root of the B*-tree, the right child of the parent node (n_i) (representing module b_i) corresponds to the first module on top of that with the same horizontal coordinate. In addition, the left child of n_i represents the lowest adjacent module on the right side of module b_i.

To calculate the reward from the defined cost function in the RL environment, we need to calculate the bottom left coordinates (x_i, y_i) of each module b_i represented by node n_i. Given a B*-tree corresponding to a layout placement, the coordinates of modules can be calculated through a packing operation. Starting from the left subtree of the root node, with (0,0) bottom-left coordinates, for the right child n_j of n_i, $x_j = x_i$; for the left child n_k of

n_i, $x_k = x_i + w_i$, where w_i is the width of the module b_i. In addition, the y coordinates of modules can be calculated by using contour structure [28]. After the left subtree of the root, we can go through the right subtree to calculate the coordinates of the remaining modules. Fig. 17 illustrates the placement of a layout and its corresponding B*-tree representation.

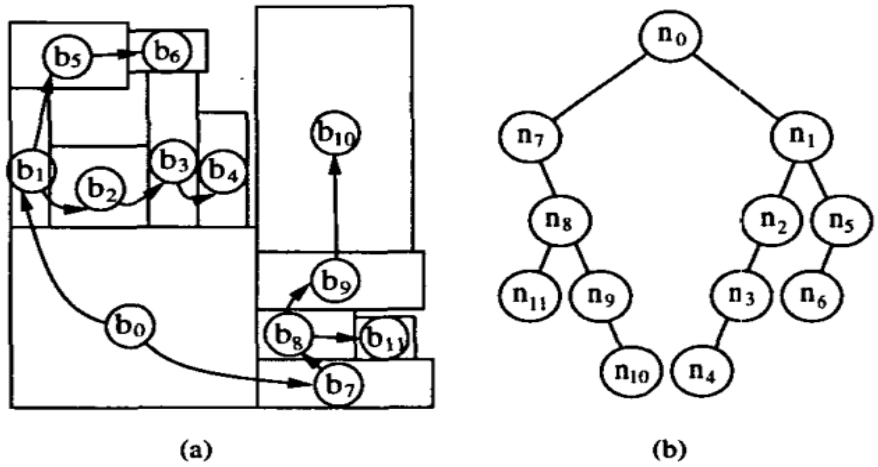


Fig. 17. (a)The placement of a layout and (b) its corresponding B*-tree.

4.2.2. Symmetry-Aware Placement

In analog circuit design, there are typically some modules required to be placed symmetrically. Considering symmetry constraints in analog placement can noticeably reduce sensitivity to process variation, thermal gradient, and other mismatch effects (e.g., parasitics) [57]. Since topological representations are flexible and efficient to handle geometrical constraints, recently they have been widely applied to handle the placement of symmetric modules. In the previous studies, different symmetry-aware placements have been proposed, such as sequence pair (SP) [58] and transitive closure graph (TCG) [59]. In [53], a new placement technique is used to address the symmetry constraints. It uses B*-

tree as the representation medium and simulated annealing as the search engine for the placement optimization.

In symmetry-aware placement, one of the major challenges is to put symmetric devices as close as possible to each other. Variations of the fogging and proximity effects and electrical parameters of symmetric devices depend on the distance between two devices in the symmetry pair. Equations (14), (15) and (16) indicate such dependency for the fogging and proximity effects. Moreover, the error of electrical parameter P for a transistor caused by mismatch can be calculated by the following equation [60]:

$$\sigma^2(\Delta P) = \frac{A_P^2}{WL} + S_P^2 D_x^2 \ , \tag{17}$$

where A_P and S_P are the constant factors, W and L are width and length of the transistor, respectively. As shown in the equation above, the mismatch error (σ) is directly dependent on D_x, which is the distance between two modules. Therefore, to have a more reliable circuit with minimized mismatch, the symmetric modules should be kept close to each other. Thus, we can put these symmetric devices in the closest proximity as a group, which is called *symmetric island*. After creating the symmetric islands, we consider each of them as a module in the placement run. In such a process, the B*-tree data structure corresponding to the layout placement is constructed and manipulated, and the optimization algorithm is used to minimize the cost function, which is defined to evaluate the quality of placements.

4.2.3. Placement Algorithm

Given a set of modules with specified constraints as inputs, the goal of our placement process is to identify the optimal position of modules that can minimize the area, wirelength, variations of the fogging and proximity effects while still handling the other analog constraints (e.g., symmetry constraints). Fig. 18 depicts the flowchart of the B*-tree-based placement scheme. In the first step called initialization phase, a B*-tree data structure is constructed based on the input file including the circuit netlist and analog constraints. Then, the optimization engine, which is simulated annealing (SA) in our implementation, begins to search for the minimum of the cost function. In more detail, it receives the initialized B*-tree, and does some perturbation to derive another B*-tree. Next, the new positions of the modules are computed by a packing operation from the new B*-tree data structure, which realizes a conversion from B*-tree to layout placement. Now, we can use module coordinates to calculate the cost function of the placement. In the next stage, if the predefined termination conditions are not met, another iteration begins to build a new placement after perturbing the current B*-tree by SA. This process will be repetitively done until the termination conditions are satisfied so that the placement optimization is completed. Therefore, at the end of the placement process, we can derive the optimal positions of all the modules, which can lead to minimum total area, wirelength, variations of the fogging and proximity effects while still meeting the required analog constraints (e.g., symmetry constraints).

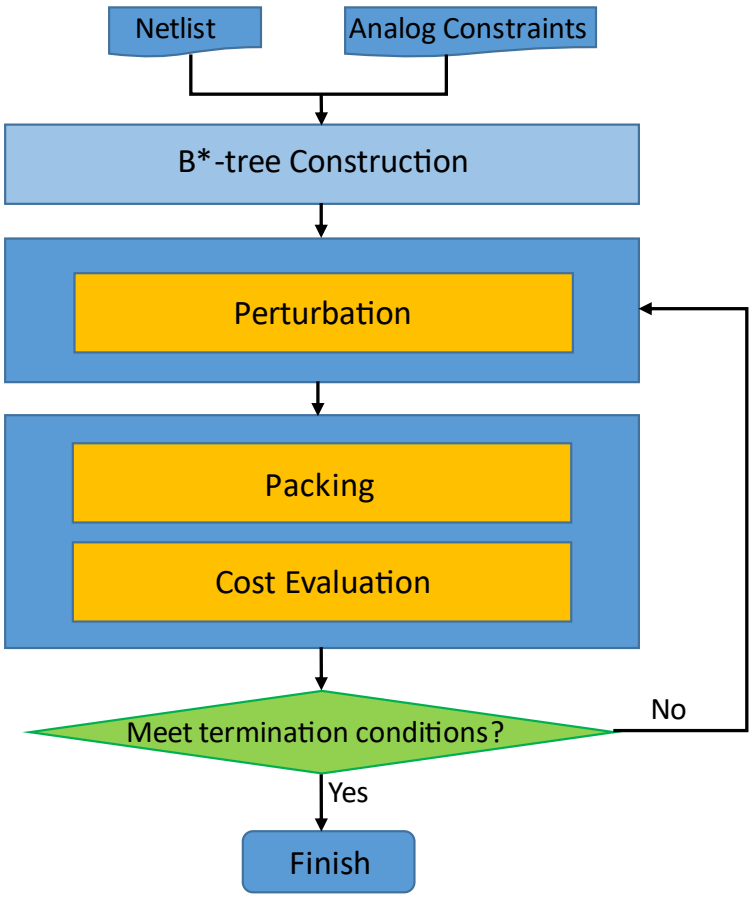


Fig. 18. Flowchart of B*-tree-based placement algorithm.

Simulated annealing is one of the most effective methods for finding the global optimum, which is inspired by the common cooling process of a metal. In SA, a random perturbation is used to reach a new configuration, then validity of the new placement is evaluated based on cost variation ΔC. If the cost is reduced ($\Delta C < 0$), it means the new placement is acceptable and can be considered as the initial placement for the next iteration. However, when the cost is increased ($\Delta C > 0$), the new configuration still may be accepted. Unlike other optimization methods, SA accepts either uphill and downhill movements,

which can help avoid getting stuck in local minimums. The uphill movement is accepted by a probability, which is calculated based on current temperature T and ΔC. The probability of the uphill movement will be calculated by $P = e^{-\frac{\Delta C}{T}}$, and then compared with a random number R, which is between 0 and 1. If the $R < P$, the uphill movement is accepted. This process will be repeated for perturbations of each M number of movements in temperature T. The temperature is then decreased by a user-defined rate α until the cost variation is less than a small number specified by the users, which is the termination point of minimization.

Perturbation is the process of generating a new structure of the B*-tree. In this procedure, three operations can be taken:

- Operation 1: Rotate a module
- Operation 2: Move a node to another position in the B*-tree
- Operation 3: Swap two nodes

To take these operations, two main actions are defined, including insertion and deletion. For example, *Operation 2* requires one insertion and one deletion action while the *Operation 3* needs two of each action. Moreover, in *Operation 1*, just the X and Y coordinates of the corresponding module will be exchanged.

To evaluate the performance of the SA-based placement algorithm, a cost function is defined. The cost function consists of area, wirelength, variations of the fogging and proximity effects, as shown in the following equation:

$$Cost\ Function = \alpha A + \beta W + \sum_{i=1}^{N} \gamma_i F_{p_i} + \sum_{i=1}^{N} \delta_i \left(BP_{p_i} + FP_{p_i} \right) , \qquad (18)$$

where α, β, γ and δ are the user-defined weights for area, wirelength, variations of the fogging and proximity effects, respectively. F_{p_i}, BP_{p_i} and FP_{p_i} are the variations of the fogging, backward proximity and forward proximity effects between two symmetric transistors in pair p_i.

4.3. Reinforcement Learning

Reinforcement learning (RL) is one of the most popular machine learning approaches, which trains a fully autonomous agent to identify an optimal policy to seek for a solution for decision-making problems in a variety of applications, such as EDA [61]. It is comprised of two basic components, RL agent and RL environment. The RL agent learns to decide intelligently and automatically through a learning process from experiences and action results given by the RL environment.

4.3.1. Fundamentals of RL

For better understanding of RL, the primary definitions are explained below:

- Agent: An RL agent is responsible for taking the best actions. In other words, the agent is the algorithm applied in RL for decision making. In our study, the RL agent is the algorithm used to do the layout placement, which meets the desired specifications, such as minimum area, minimum wirelength, and minimum variations of the fogging and proximity effects.

- Action (A): A is the set of possible movements that the agent can take. It is considered as the way that the RL agent interacts with the RL environment. In the layout placement approach, the action is one possible movement of one electronic device (e.g., transistor) in the given layout.
- State: The state shows the current representation and situation of the RL environment, which is the execution result of the action taken by the RL agent. In addition, it can be defined as the current position of the RL agent in our problem environment. For example, one state in our study represents the current layout placement, which reflects the module status after execution of the modules' movements.
- Reward: To train the RL agent, it is required to let it know if the action taken is a good decision or not as a feedback. The reward is the value that shows the result of the action taken by the RL agent. It might be functioning as a punishment or encouragement. In the domain of our layout placement problem, the reward is calculated based on the cost function as defined in Eq. (18).
- Environment: The RL environment is the world and in particular the ambient surrounding which the RL agent interacts with to take actions. It means that the RL environment receives an action from the RL agent to be executed, then gives the corresponding reward of that action as well as the newly formed state back to the RL agent. In our layout placement application, the RL environment is the chip layout setting that represents the placement status of all the modules inside.

Fig. 19 depicts the flowchart of the RL method. As illustrated, action A_t taken by the RL agent at time step t is given to the RL environment. Next, the RL environment computes the reward of that action and sends reward r_{t+1} and new sate s_{t+1} to the RL agent. This process is repeated over and over again until the RL agent is well trained by the experiences received from the RL environment and is able to intelligently choose the best actions.

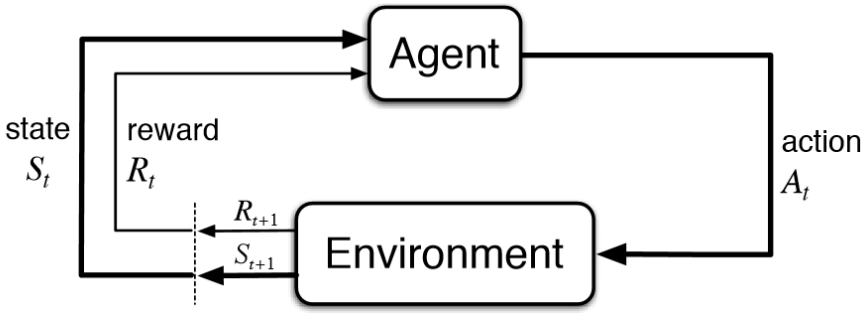


Fig. 19. Flowchart of the RL method

4.3.2. Deep Q-Network

Machine learning (ML) techniques are used in the variety of applications to solve the real-life problems. Conventional ML methods were not able to take and process the raw form of natural data. That is to say, to train an agent or learn a subsystem for solving a specific problem, the designers had to put a lot of effort to transfer the raw data (such as an image, an audio file, or a text) to a recognizable form of features, which the agent or the learning system can accept and realize as a valid input [62]. Therefore, new innovative machine learning methods called *representation learning* were introduced, which enables the learning system to receive raw input data and recognize them as a suitable form for the learning process.

One of the most popular representation learning methods is deep neural network (DNN) or generally called deep learning, which uses several hidden layers for high level data abstraction. DNN consists of an input layer, an output layer, and multiple hidden layers including units called nodes, which can be connected. As Fig. 20 shows, the nodes of a layer can be connected to the nodes in the adjacent layers, and each connection has a weight. The summation of weighted inputs is calculated at each node and then an activation function is applied to transform the weighted summation to create the output of that node. The output is fed to the connected nodes in the next adjacent layer as an input. For example, in Fig. 20, the output of node A_1 is calculated by applying activation function σ to the weighted summation of X_1, X_2 and X_3, which are connected to A_1 as input. So the output of A_1 is $\sigma\left(\sum_{i=1}^{3} W_i^{(A_1)} X_i\right)$, where the $W_i^{(A_1)}$ is the weight of connection between X_i and A_1.The final output of the last layer is the result or called *solution* of the given problem, which includes Y_1 and Y_2 for the one as illustrated in Fig. 20. However, this result is not the correct answer yet, because the neural network is required to be trained in order to provide better and more accurate solutions [63]. For the learning process, the loss function, which is the difference between the output of neural network and the desired result, needs to be calculated. The network tries to update the weights based on the loss function in order to achieve a model, which is able to predict the solutions with high accuracy.

Since DNNs significantly improved their performance after the researchers' effort in the past years, they have been utilized in a variety of academic and industrial applications. Moreover, DNNs also made contributions to reinforcement learning to solve the problems

with big input data faster and more accurately. Thus, this leads to a new area in machine learning called *deep reinforcement learning* (DRL). One of the most popular and successful DRL approaches is deep Q-network (DQN), which was introduced in 2015 [64]. Moreover, many studies and research works have used DNN through the RL and the integration of these two methods goes back to 1996 [65] [66].

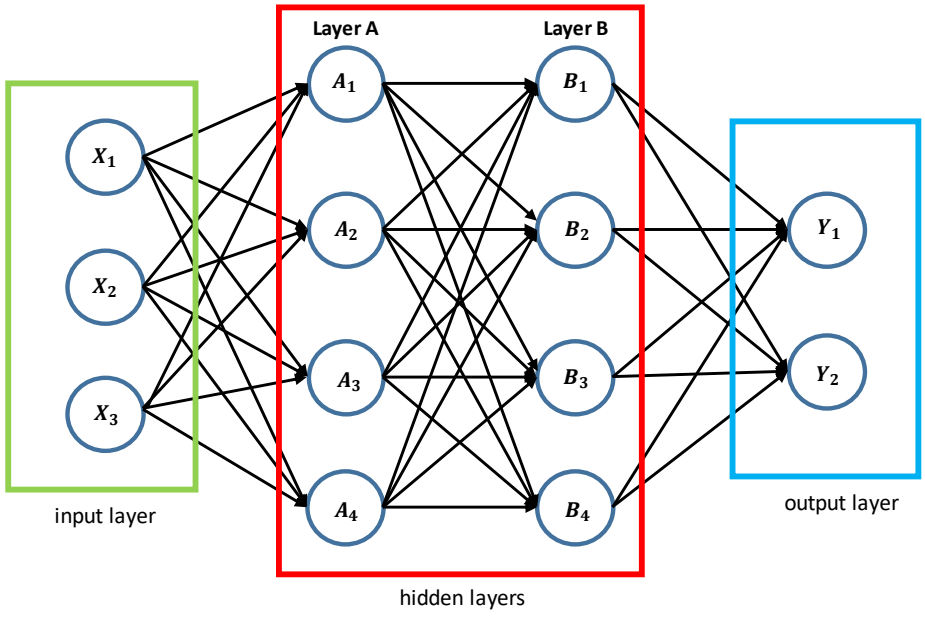


Fig. 20. Schematic of a DNN

The main purpose in reinforcement learning is to train an agent that takes the best actions to maximize the discounted accumulative reward, which is also called *return* (R_t) as calculated below:

$$R_t = \sum_{t=0}^{\infty} \gamma^t \times r_t \ , \tag{19}$$

where γ and r_t are the discount factor and reward at time step t, respectively. The discount factor, which is normally between 0 and 1, is applied to define how much an agent should consider the uncertain rewards in distant future rather than immediate rewards. The ideal form in RL is to have an optimal action-value function (called *Q function*) that takes the best action in the present state resulting in the maximization of the return. By having the optimal Q function (Q^*), we can create a policy (π) that intelligently chooses the best actions in each state, which is defined as below:

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q^*(s,a) \ , \tag{20}$$

where s stands for current state and *a* refers to current action taken by the RL agent.

However, we do not have access to the optimal Q function, whose attainment is not possible in reinforcement learning. But in [64], the DQN was introduced to apply deep neural network as a method to approximate the optimal action-value function for current state s and action a, which is defined as follows:

$$Q^*(s,a) = max\ E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a] \ , \tag{21}$$

where γ is the discount factor, and r_{t+n} is the future reward at time step n . Therefore, the DNN is considered as an intelligent agent that predicts the Q-values. However, in the beginning the neural network estimation is not good and it is required to be trained. Thus, an error function is introduced to DNN to allow the network to update the weights repeatedly until it can approximate the Q-values with high accuracy. The error function is

the difference between the optimal Q-value and the predicted one, which is the output of DNN.

4.4. Our Reinforcement-Learning-Based Placement Methodology

In this work, we develop a reinforcement learning placement method based on a topological representation (i.e., B*-tree in our implementation). In our reinforcement learning placer, the RL environment can accept a B*-tree data structure representing the module positions in a layout, while the RL agent is a deep neural network (DNN) with reward calculated by our specified cost function. Given a netlist, we locate the modules in a way that can lead to minimization of area, wirelength, and variation of the fogging and proximity effects among sensitive analog modules in the mixed-signal ICs.

As Fig. 21 shows, in the operating process of our placement method, there is an initialization as the first step where an initial B*-tree representation of modules is formed based on the netlist of the circuit. Then, the nodes in the tree structure representing modules in the layout are moved in a controlled and organized way to minimize the cost function, resulting in a layout with the desired performance. In most of the previous placement algorithms, such as conjugate gradient or simulated annealing based methods, a greedy or random movement is used. In contrast, in this work an RL agent is trained to intelligently choose the best action for each node in B*-tree based on the prior generated states and their cost feedback. In the next step, the image of the layout is fed to the RL agent, which is a

DNN. Such an RL agent is trained to map the image of layout to proper actions that lead to the best node movement in the B*-tree structure.

The neural network generates Q-values for each possible action of modules. The RL environment takes the action with the highest Q-value and executes it, resulting in a new RL state. Now, the cost function of the new RL state is calculated and fed to the RL agent. The biases and weights in the layers of DNN are updated based on the cost function, which is the reward for our RL agent. So the new Q-values are generated on the network's output based on the reward and new RL state on the input of DNN. This process continues to train the RL agent to choose the best action efficiently and intelligently for each module in the B*-tree data structure.

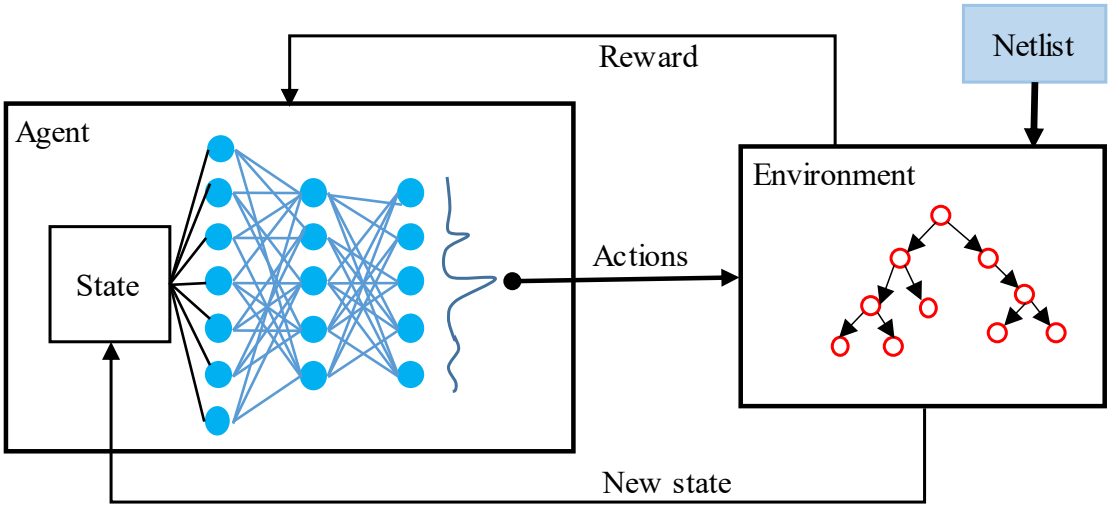


Fig. 21. RL-based placement with B*-tree

4.4.1. RL Agent

In reinforcement learning, the purpose for the RL agent is to learn an optimal policy that maximizes the accumulative immediate rewards. The RL agent takes the best actions

based on the benefits and punishments received from previous actions. The RL agent in our work is a DNN with three hidden layers. The first two are conventional layers with Max pooling for downsampling, reducing the size of inputs by removing unnecessary data. On the output of the last layer, there are Q-values for each possible action of each module in the circuit. We define four actions for each module: swap, rotate, move, and no action. Therefore, the last layer is a dense layer with a size of four times the total number of modules. Thus, given a state to the neural network as an input, the RL agent tries to choose the optimal Q-value for each action. The optimal Q-value is calculated based on the following equation:

$$Q(s_t, a_t) = r_t + \gamma \times \max_{\forall a \in A_i} Q(s_{t+1}, a) \ , \tag{22}$$

where r_t is the instant reward of the previous action, γ is the discount factor. $\max_{\forall a \in A_i} Q(s_{t+1}, a)$ is the maximum expected future reward, given a new state and all possible actions for each module (A_i) at the new state. The weights of the network are initially set by random numbers. To train our DNN, we use mean squared error. So in each iteration, the loss function is calculated, and the weights are updated to minimize that. The loss function of our DNN is:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(Q_i - \acute{Q}_i)^2 \ , \tag{23}$$

where n is the number of output nodes, and Q and $\acute{Q}$ are the optimal and predicted Q values, respectively.

4.4.2. RL Environment

In the reinforcement learning method, the RL environment is responsible for taking action from the RL agent, executing it, calculating the reward, and then feeding the reward and new state back to the RL agent. This interaction between the RL agent and the RL environment leads to successful learning. In our placement method, the states are possible configurations of the B*-tree data structure. The RL environment executes the action by changing position of any nodes in the B*-tree structure, and then a new state is derived. For calculating the reward, the RL environment needs to do the module packing operation to have the exact coordinates of each module. Then the reward is generated based on the cost function of the new state and new placement. The cost function in our work is defined as:

$$\varphi = \alpha A + \beta W + \sum_{i=1}^{N} \lambda_i F_{p_i} + \sum_{i=1}^{N} \theta_i \left(BP_{p_i} + FP_{p_i} \right) , \tag{24}$$

where α and β are the user-specified factors, A is the chip area, and W is the wirelength. For considering the fogging effect in our cost function, we use F_{p_i} from Eq. (14), i.e., the fogging effect variation between two modules in the sensitive device pair i. The weight of the fogging effect variation in pair i is specified by λ_i, which is related to the sensitivity level of the pair. The proximity effect variation is considered by two components BP_{p_i} and FP_{p_i} from Eqs. (15) and (16). θ_i specifies the weight of proximity effect variation for pair i. We take into account the sum of the fogging and proximity effects variations of all N sensitive pairs. In addition to those, the regular analog constraints (e.g., symmetry

constraints and boundary constraints) are similarly considered as those already published in the literature [67].

To generate the reward of a taken action and give it back to the RL agent for the training purpose, we define the rewards or punishments based on the cost function as follows:

$$r_t = \begin{cases} +300 & \varphi < C_{th} \\ +5 & C_{th} < \varphi < 1.3 \times C_{th} \\ +2 & 1.3 \times C_{th} < \varphi < 1.5 \times C_{th} \\ -1 & else \end{cases}, \tag{25}$$

where C_{th} is the cost threshold value defined by the users.

4.5. Summary

This chapter discussed our reinforcement-learning-based placement methodology that uses the B*-tree data structure as one topological representation. First of all, the fundamentals and basics of reinforcement learning, deep Q-learning, B*tree, and SA were explained. Then, we presented our placement approach that uses deep reinforcement learning as the optimization mechanism. In this placer, we utilized B*-tree data structure as the topological representation in the RL environment and applied the deep Q-network as the RL agent. Beyond the common objectives such as the chip area and wirelength, the fogging and proximity effects were taken into account in the cost function, which could be in turn reflected in the reward of the RL training process. Eventually they enabled the RL agent to intelligently choose the best actions in a significantly shorter time.

Chapter 5
Experimental Results

5.1. Introduction

We implemented our scattering-effect-aware placement algorithm in Python by using Tensorflow [68] and Adam optimizer [69]. We evaluated our scattering-effect-aware placement method on three operational amplifier circuits within mixed-signal ICs in 18nm FinFet technology. As mentioned before, we are considering the placement for the analog part of a mixed-signal circuit. To take into account the fogging and proximity effect sources from the digital part modules, we assume a 1.5mm×1.5mm square that can be divided into the same standard cells with defined pattern density. In this section, we will make a comparison among four placement methods for minimization of the fogging effect and proximity effect as well as achieving the other optimization objectives with various analog constraints satisfied. The first method is the analytical RL placement method [39], which uses absolute coordinate representation and deep Q-learning. The second one is a simulated annealing (SA) based analog placement method using B*-tree as a topological floorplan representation [53]. The third placement method is based on advantage actor critic (A2C),

which is a new reinforcement learning approach. And the last one is our proposed method using RL as the optimization engine with the B*-tree topological representation.

5.2. Evaluation of the Fogging and Proximity Effects

As mentioned in Chapter 3, two of the main concerns for the designers in the EBL technologies are the fogging and proximity effects. However, the fogging and proximity effects have not been considered in the analog designs in the previous studies. So, in this part we first do the experiment to evaluate the fogging and proximity effects in an analog circuit, which is part of a mixed-signal IC. We will analyze the fogging and proximity effect on the comparator part of a successive-approximation-register analog-to-digital converters (SAR-ADC). Based on [16], the gate length change caused by the fogging effect is about 10% of the critical dimension (CD) of the used technology process. Fig. 22 shows the schematic of the comparator circuit with two inputs on $V1$ and $V2$, and two outputs (OUTN and OUTP). We want to simulate fogging effect by changing the gate length by 10% in transistor $M1$ that is considered as CD error induced by fogging effect.

When the voltage of $V2$ (or $V1$) is more than of $V1$ (or $V2$), the positive output called OUTP turns high (low). Therefore, at the time when the output voltage changes from high to low or vice versa, the difference between two input voltages must be zero and any error in this voltage difference leads to bit error in binary output of SAR-ADC. Fig. 23 illustrates the simulation result considering CD error variation in the symmetric differential pair caused by the fogging effect. The gate length change is 1.8 nm in this simulation, which occurs in transistor $M1$. The voltage error is specified in the figure, which is about 13.2

mV. To find out the result of this error on the output of the analog to digital converter, we can use the equation below:

$$\Delta V = \frac{V_{FSR}}{2^n}, \tag{26}$$

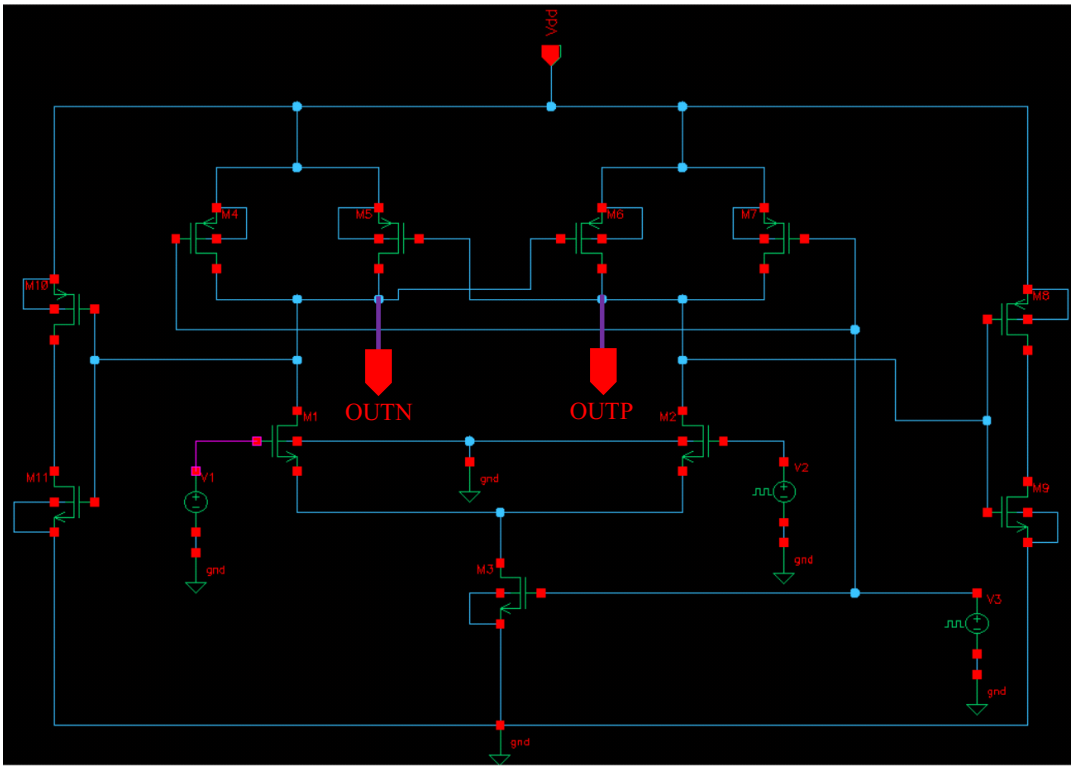


Fig. 22. Schematic of comparator circuit in Virtuoso

where ΔV is the smallest voltage change that result in change of digital output of an n-bit ADC, which is also called the least significant bit (LSB) voltage. V_{FSR} is the full scale input voltage range. Therefore based on Eq. (26), the fogging effect can result in at least two bits error on the output of the 10-bit SAR-ADC with 2 V of input voltage range. So, it can be seen that the fogging effect has a significant impact on the performance of a mixed-signal IC. In addition, when it comes to proximity effects, it has much higher intensity than the

fogging effect while the range of that effect is much shorter. Thus, it can be inferred that the proximity effect would also result in a noticeable performance error of analog portion within a mixed-signal IC. If considering 10% of CD error induced by the proximity effect, the output of the SAR-ADC can also have at least two bits of error as the impact of the proximity effect if not being properly controlled.

Fig. 23. The simulation result considering the fogging effect in the comparator circuit. The red, green, and blue graphs are OUTP, V2, and V1 voltages, respectively.

5.3. Sensitivity Analysis

The fogging and proximity effects cause different gate length errors in different symmetric pairs, which may degrade the CD uniformity in the chip. Moreover, the error in circuit performance vary for the mismatch of different symmetric pairs. This means one symmetric pair has its own sensitivity level, which might be different from that of the other

symmetric pair [70]. To demonstrate this point, we did a simulation for a folded-Cascode operational amplifier (opamp) to show dependency of circuit performance on gate length errors from different transistors. Fig. 24 and Fig. 25 depict the schematic of the circuit and simulation results in the Cadence Virtuoso environment, respectively.

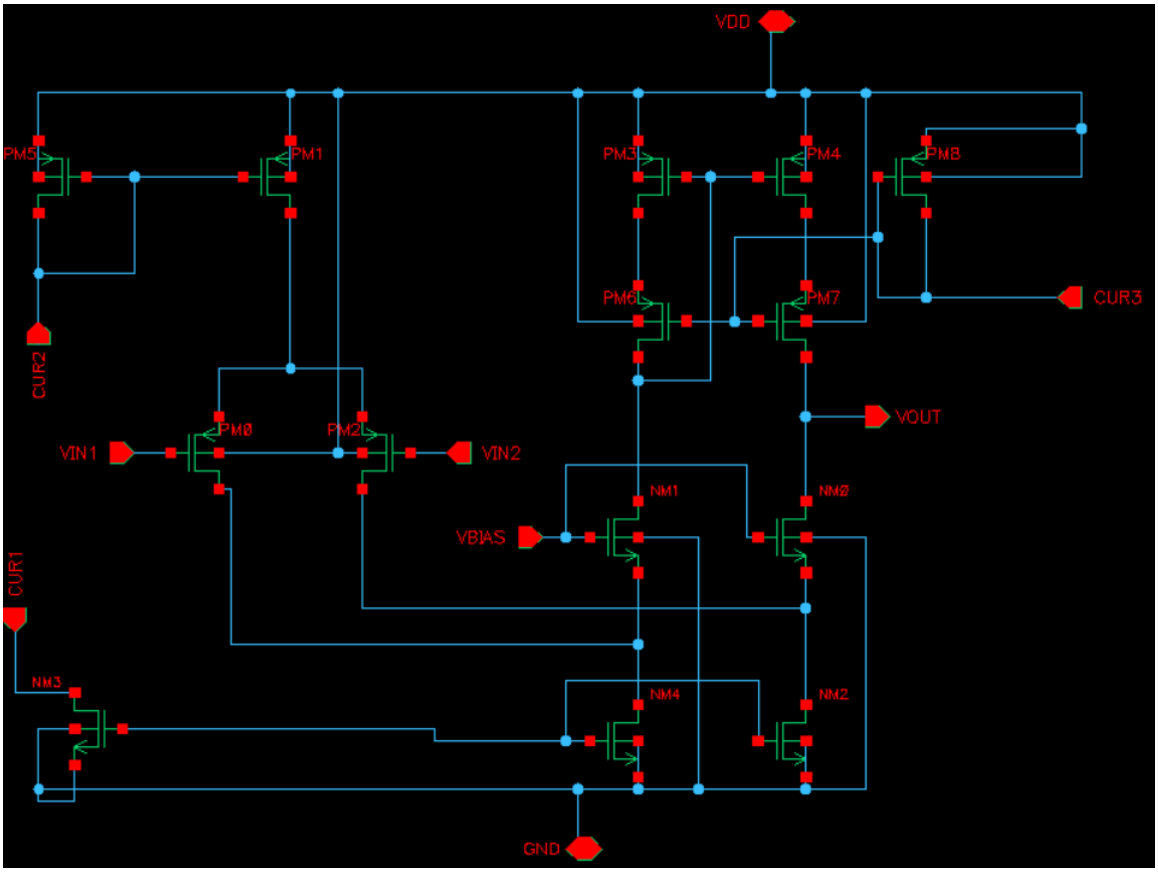


Fig. 24. Schematic of the folded-Cascode opamp

The gains of the circuit are shown in Fig. 25 for different settings, in each of which only one transistor has its size changed. The gate length error is considered to be $1nm$ that happens to just one of the two transistors in the symmetric pair, whose name is marked in the figure. Based on the results, the dc gain changes dramatically when there is a mismatch

between transistors PM0 and PM2; it may cause up to approximately 45 dB error for just $1nm$ gate length error in PM0. Therefore, we have to implement our placement algorithm to prioritize the variation minimization due to process imperfection for different symmetric pairs based on their sensitivity analysis.

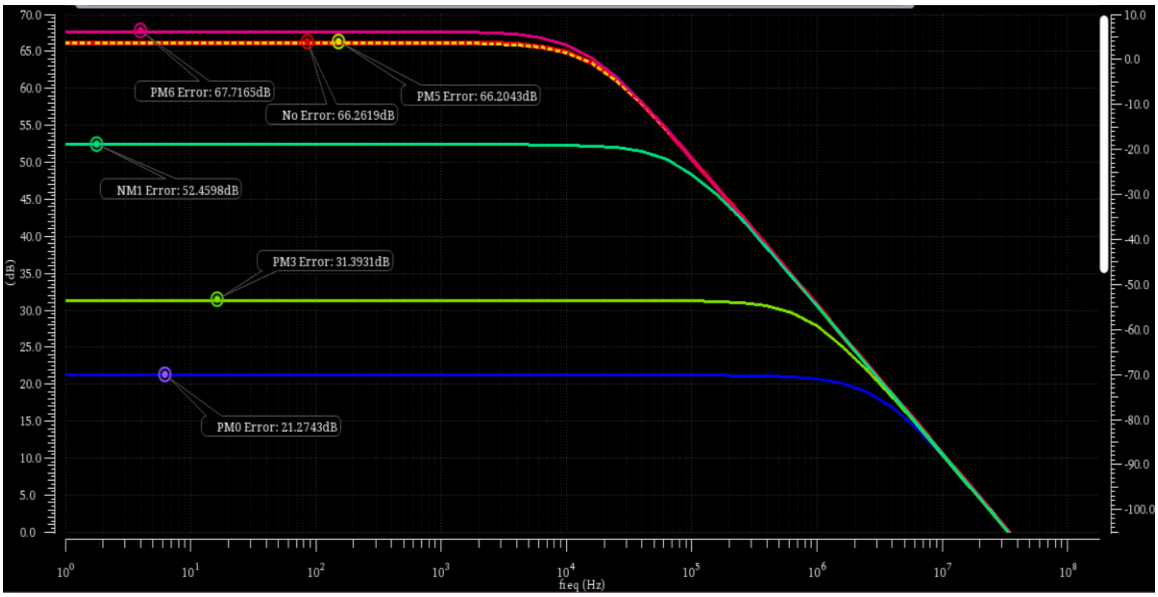


Fig. 25. Simulation results showing gain change based on gate length error from different transistors

5.4. Running and Training Time

In Table 1, the running and training time of the four methods are compared for the two-stage operational amplifier (opamp) circuit. It shows the average time needed for the analytical placement trained model, SA-based placer, A2C and our proposed RL placement method to minimize the cost function by using the user-defined threshold cost value. We considered 700 as the threshold cost value and used 50 episodes to train the models in our reinforcement learning technique. The optimal threshold cost value is achieved through the

experiments and simulations. Based on Table 1, the average running time for our proposed model in this work is 7.29 seconds, which is about 224 times faster than the analytical RL placement method, while the training time is 96 minutes which is approximately 15 times faster. The main reason for this improvement is the number of possible states in the RL environment that defines the search space. For the placement method in [39], the chip is considered as a square with 40 grid lines on each side, dividing the chip into 1,600 sub-grids. The transistors are able to be only placed on the bottom left corner. Hence, if we have eight transistors and 1,600 possible placement positions for each one, there are approximately 4.2×10^{25} states in the RL environment. To calculate the possible states in our method which is equivalent to the number of structurally different B*-trees in our RL environment, we use the following equation:

$$S(n) = C_n \times n! \times 4^n \,, \tag{27}$$

where n is the number of nodes in the B*-tree and C_n is the Catalan number of n. The C_n represents the different configuration of a B*-tree with n nodes [71]; in addition, the $n!$ shows the different states generated by all possible allocations of n modules to n nodes. Moreover, the 4^n in the equation above represents the states induced by rotating of modules in clockwise directions with 0, 90, 180, or 270 degrees. So based on Eq. (27), there are 3.7×10^{12} possible states as the representation of different layouts. The comparison between possible states in the two methods shows that the absolute-coordinate-based placement method in [39] has about 8.8×10^{12} times more possible states. As a result, the search space of our model is much smaller than that of the analytical approach,

Table 1. Average running time and training time for analytical RL placement, SA, A2C, and our work for the two-stage opamp circuit

Technique	Average Running Time (sec.)	Training Time (min.)
Analytical RL Placement	1633.8	1446
SA	24875.6	-
A2C	38.57	435
This work	7.29	96

which contributes to faster training and running time. Moreover, the modules in a common layout are preferably located close to each other, which makes a compact placement, if no special constraints are applicable. Therefore, a big portion of the possible states in the absolute-coordinate-based placement method in [39], which does not bring much benefit to the electrical and geometric performance, is not necessary by nature.

When it comes to compare the efficiency of our proposed approach with the SA-based placement method, our placer is 4.3 times faster. In the SA-based placement method, we use random perturbations rather than a trained RL agent, which is able to intelligently perturb the modules. Based on our experimental results, the SA algorithm calculated the cost function for 6.6×10^{13} times during the optimization, where the fogging and proximity effects was evaluated in each iteration. However, the maximum states in our proposed method is 3.7×10^{12}. Therefore, the SA-based method calculated the fogging

and proximity effects for the given placement at least 18 times more than our proposed RL method, which causes its slower operation and poorer performance.

To compare the efficiency of the A2C-based placement method and our work, the running time and training time in our method are 5.3 and 4.5 times shorter. The A2C method applies two deep neural networks called actor and critic to implement the reinforcement learning algorithm [72]. So utilizing two neural networks, compared to our DQN method that has only one neural work, increases the complexity of the A2C method. This complexity gets even worse in the placement process considering the fogging and proximity effects that requires intensive computation.

Table 2 presents the average running and training time of the four placement approaches for the folded-Cascode opamp circuit. Based on the results, the running and training time of the four placement methods increases significantly, if the size of the circuit gets bigger. However, our method shows slightly less growth in training time and rises to 117 min.

Table 2. Average running time and training time for analytical RL placement, SA, A2C, and our work for the folded-Cascode opamp circuit

Technique	Average Running Time (sec.)	Training Time (min.)
Analytical RL Placement	2546.3	2721
SA	46720	-
A2C	961.4	2270
This work	34.3	117

The training time of A2C, SA, and analytical RL placement takes too long for fully symmetric opamp and high gain opamp circuits since they are large circuits. Without any configuration to their original algorithms, the training time could take longer than four days. Therefore, we changed the configuration settings of those three approaches to reduce the training time. Table 3 and 4 summaries the running and training time of the four placement approaches for fully symmetric opamp and high gain opamp circuits, respectively.

Table 3. Average running time and training time for analytical RL placement, SA, A2C, and our work for the fully symmetric opamp circuit

Technique	Average Running Time (sec.)	Training Time (min.)
Analytical RL Placement	640.5	215
SA	21620	-
A2C	584.71	198
This work	259.06	545

Table 4. Average running time and training time for analytical RL placement, SA, A2C, and our work for the high gain opamp circuit

Technique	Average Running Time (sec.)	Training Time (min.)
Analytical RL Placement	961.47	328
SA	26590	-
A2C	750.37	245
This work	310	643

5.5. Performance Comparison

To evaluate our proposed placement method for minimizing variation of the fogging and proximity effects, wirelength, and area, we did experiments on the two-stage opamp with 8 transistors, folded-Cascode opamp with 14 transistors, and a fully symmetric opamp with 22 transistors. Table 2 shows the results of the analytical placement scheme [34], the SA-based placement method [46], A2C-based placement method [73], and our proposed B*-tree RL placement method for the two-stage opamp. The total area and wirlength obtained by the analytical method are 4.5 and 3.9 times more than ours, respectively. However, the fogging effect variation minimizations of both methods are roughly the same, while the proximity effect variation minimization and DC gain from our method are approximately 7 % and 1 dB better than those obtained from the analytical RL method. In this experiment, our method is much faster with better area, wirelength, and proximity effect handling and a slightly better level of fogging effect minimization. To compute the DC gain in the simulation, the pattern distortion towards gate length change is taken into account. We consider that the fogging effect or the proximity effect has a linear relationship with the gate length change. And in the worst case, the gate length change is assumed to be 10% of the critical dimension (CD) by following the work reported in [16].

As Table 3 shows, our proposed method has significantly better performance than the SA-based placement method with 26 dB, 10%, and 35% better DC gain, fogging and proximity effect variations respectively, while our placement method outperforms the SA-

based placement method in both area and wirelength optimization. Moreover, the A2C-based placement approach shows weaker performance compared to ours in minimization of the fogging and proximity effect variations, which results in 10.7 dB less DC gain.

Table 5. Comparison of the placement results among analytical RL, SA-based, A2C-based, and our proposed B*-tree-based RL placement methods for the two-stage opamp

Technique	Analytical RL [39]	SA [53]	A2C [73]	Our work
Area (nm^2)	572.2	150.5	581.3	126
Wirelength (nm)	354.5	104.7	364.96	90.11
Fogging effect minimization (%)	88.91	79.22	87.45	89.26
Proximity effect minimization (%)	88.25	60.2	83.01	95.22
Gain (dB)	64.43	39.4	54.72	65.49

Table 4 lists the simulation results of the four abovementioned placement approaches for a folded-Cascode opamp. Our proposed method shows good performance in minimizing the fogging effect, which is 16.7%, 11.2% and 30.55% better than the analytical RL, A2C and SA-based methods, respectively. In addition, the proximity effects minimization in our method is 84.08% that is much better than the other comparison placement approaches. When it comes to performance improvement, our method has 44.16 dB DC gain, which is 4.96 dB, 5.37 dB and 7.62 dB better than analytical RL, SA and A2C-based placement methods, respectively.

Table 6. Comparisons of the placement results among analytical RL, SA-based, A2C-based and our proposed B*-tree-based RL placement methods for the folded-Cascode opamp

Technique	Analytical RL [39]	SA [53]	A2C [73]	Our work
Area (nm^2)	576	254.4	627	304
Wirelength (nm)	695.32	179.71	620.67	247.38
Fogging effect minimization (%)	71.8	77.3	57.95	88.5
Proximity effect minimization (%)	81.9	76.4	58	84.08
Gain (dB)	39.2	38.79	36.54	44.16

Table 5 lists the experimental results for the fully symmetric opamp circuits (including 22 transistors) as illustrated in Fig. 26. As shown in the table, our proposed placement approach has better fogging and proximity effects control compared to analytical RL method with 46.7% and 45.7% better minimization, respectively. In addition, the area and wirelength in our method are 3.5 and 3.7 times less than of those in analytical placement, while our work shows about 30 dB more in terms of DC gain. Moreover, our method has better performance compared to the A2C-based placement approach with 8.2 dB more DC gain, 21% less fogging effect variation, and 11% less proximity effect variation.

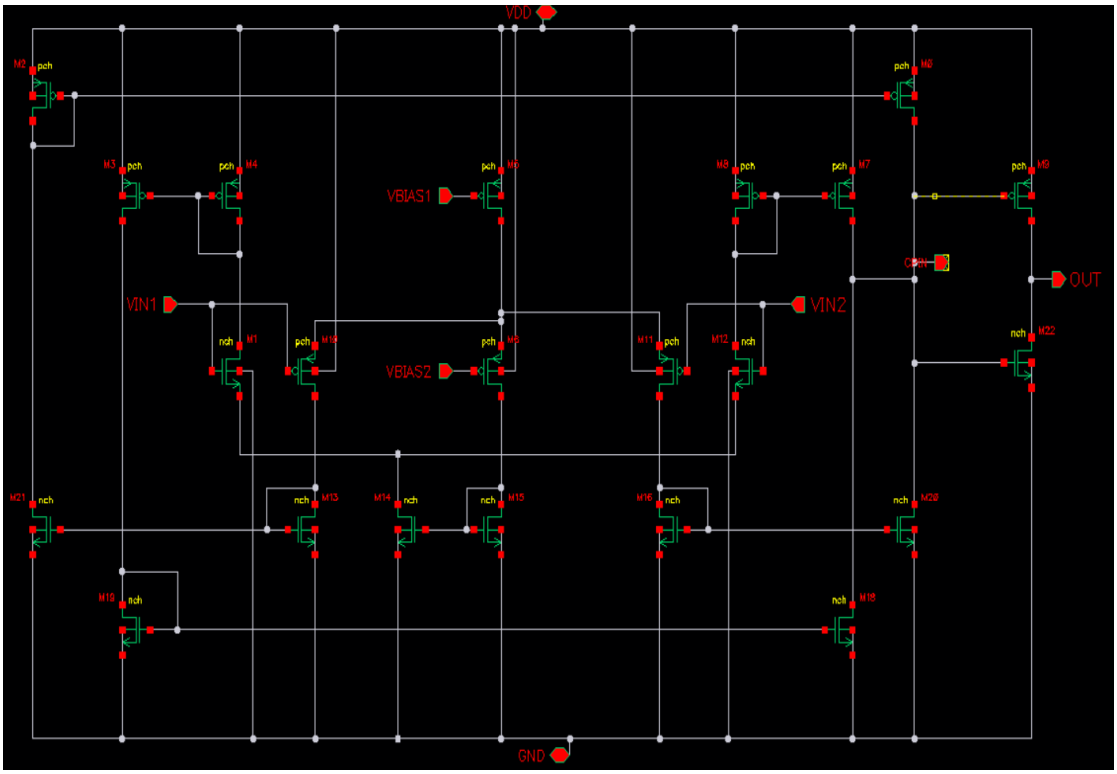


Fig. 26. Schematic of the fully symmetric opamp with 22 transistors

Table 7. Comparisons of the placement results among analytical RL, SA-based, A2C-based and our proposed B*-tree-based RL placement methods for the fully symmetric opamp circuit illustrated in Fig. 26

Technique	Analytical RL [39]	SA [53]	A2C [73]	Our work
Area (nm^2)	2482	650.7	2518.5	704
Wirelength (nm)	1242.88	263.7	1259.49	333.38
Fogging effect minimization (%)	40.84	61.5	66.6	87.55
Proximity effect minimization (%)	24.5	40.3	59.2	70.25
Gain (dB)	36.18	48.23	58.6	66.8

Table 6 illustrates the experimental results for the high gain opamp circuits (including 22 transistors) with 10 symmetric pairs as illustrated in Fig. 27. Based on the information in the table, our proposed method shows satisfying performance with approximately 85% and 77 % minimization of fogging and proximity effects, respectively. Therefore, the gain achieved by our placement method with 64.6 is much better than other three approaches.

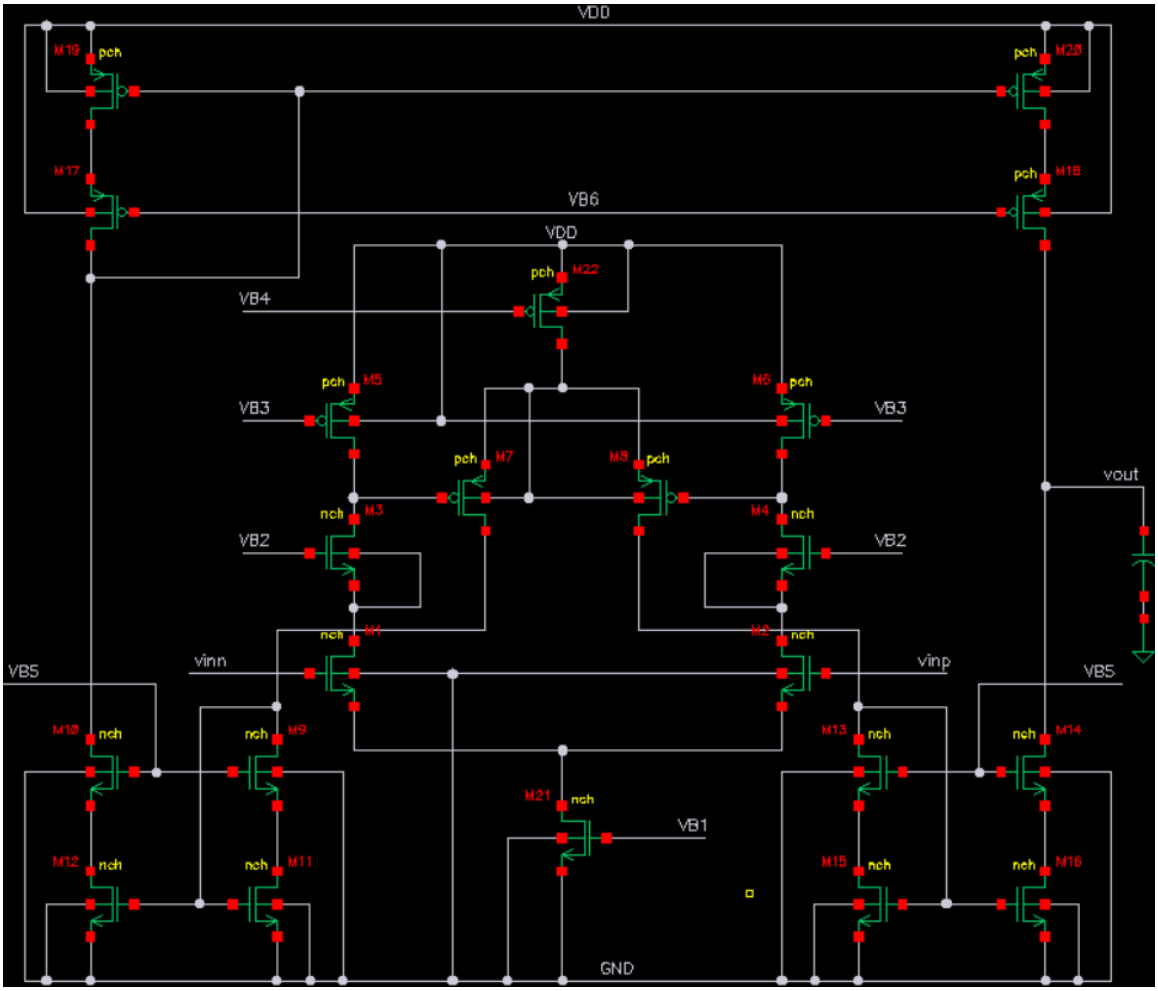


Fig. 27. Schematic of the high gain opamp with 22 transistors

Table 8. Comparisons of the placement results among analytical RL, SA-based, A2C-based and our proposed B*-tree-based RL placement methods for the high gain opamp circuit illustrated in Fig. 27

Technique	Analytical RL [39]	SA [53]	A2C [73]	Our work
Area (nm^2)	2520	610	2516	990
Wirelength (nm)	1137.1	302.7	1257.82	449.64
Fogging effect minimization (%)	43.02	50.8	52.65	84.9
Proximity effect minimization (%)	24.37	44.58	40.16	77.09
Gain (dB)	25.44	40.3	42.29	64.6

5.6. Summary

This chapter presented and discussed the results of our experiments. In the first part, the fogging and proximity effects were evaluated in the context of mixed-signal ICs and the errors induced by those effects were discussed with regard to circuit performance. Then in terms of running and training time, we compared our proposed methodology with the other state-of-the-art placement methods, including simulated-annealing-based, absolute-coordinate-based, and advantage-actor-critic-based placement methods. In addition, the performance of those approaches were presented and discussed for three different opamp circuits. In general, our proposed B*-tree RL-based placement method showed its better

performance on minimizing the variations of the fogging and proximity effects in significantly shorter time.

Chapter 6 Conclusions and Future Work

6.1. Conclusions

In this thesis, we introduced a reinforcement-learning-based placement method to minimize the variations of the fogging and proximity effects in the analog part of mixed-signal ICs. In our method, we used deep Q-learning to train our RL agent so that it is able to intelligently choose the actions in less runtime. Instead of the popular absolute coordinate representation, we applied the topological representation in our placer that can contribute to smaller search space.

Chapter 1 presented the principles and issues of traditional lithography methods. The traditional lithography approaches experience critical resolution issues for fabricating semiconductor devices in sub-22 nm technologies. In this chapter, the RET methods were discussed, which are applied to improve the resolution in photolithography. Finally, the next generation lithography (NGL) methods were introduced as effective methods for new technology nodes in semiconductor fabrication.

In Chapter 2, the next-generation lithography techniques and their challenges were discussed. Then, we presented the basics and fundamental concepts of the fogging and proximity effects as two major challenges in electron beam lithography. In addition, we reviewed the previous studies about the fogging and proximity effects as well as the

methods for controlling them. In addition, the previous works on different placement methods were reviewed in terms of two classifications: representation methods and search engines.

Moreover, the energy distribution and variation modeling of the fogging and proximity effects were proposed in Chapter 3. We reviewed and discussed the extraction of the mathematical modelling for the fogging and proximity effects called point spread function (PSF), which is achieved by energy distribution models of the effects. Finally, we proposed our modeling scheme for the variations of the fogging and proximity effects in the placement handling for mixed-signal ICs.

Chapter 4 presented our B*-tree based RL placement method considering the fogging and proximity effects. First, we introduced the topological-based placement method, which is applied to control the variations of the fogging and proximity effects in the mixed-signal ICs. Then, we provided the fundamentals of reinforcement learning and deep Q-network. Finally, our innovative placement approach was proposed to apply deep reinforcement learning to handle the fogging and proximity effects in addition to wirelength and area. In this placer, we utilized B*-tree data structure as the topological representation in the RL environment and applied the deep Q-network as the RL agent.

In Chapter 5, we presented the experimental results to show the significant electrical performance errors induced by the fogging and proximity effects in mixed-signal ICs. In addition, the running time, training time, and performance of our proposed method were compared with the other three state-of-the-art placement methods, including simulated

annealing (SA) based, absolute coordinate based, and advantage-actor-critic (A2C) based placement methods. The experimental results showed that our placer could efficiently minimize the variations of the fogging and proximity effects for the sensitive devices resulting in more desirable circuit performance, besides meeting chip area, wirelength and other common analog placement constraints. In addition, our proposed approach is 15, 4.3, and 5 times faster than the state-of-the-art analytical RL-based, simulated-annealing-based and A2C-based placement methods, respectively. Generally speaking, our proposed reinforcement learning method utilizes B*-tree and deep Q-network as the RL environment and the RL agent, respectively, which leads to noticeably speedup of our placer. Moreover, as the first study we considered the control of the fogging and proximity effects in the analog portion of mixed-signal ICs.

6.2. Future Work

As feature size is getting increasingly smaller in new semiconductor device fabrication technologies using EBL as the lithography method, the importance of the fogging and proximity effects increases. In analog placement, considering different constraints, such as the fogging and proximity effects, it is required to preserve the circuit electrical performance. However, computing the fogging and proximity effects is quite time-consuming. Therefore, using efficient placement approaches with low running time would be crucial. Our proposed RL method has showed significant improvement in training and running time. However, our developed placement algorithm can be only used for a specific circuit; that is to say, for any unseen circuits, the machine learning model has to be re-

trained from scratch. Therefore, utilizing the transfer learning technique might be able to contribute considerable benefit to this study. Such a technique can help transfer the trained model to the next training task so that it can decrease the training time for the optimization of the unseen circuits. In addition, as the placer will be trained for more unseen circuits, the accuracy and performance of the placer will be improved leading to placements with better quality.

In addition to the fogging and proximity effects, there are some other major effects, which are induced by process variation and may lead to critical CD errors. One of the known long-range effects is develop loading, which results in a CD error two times more than fogging effect [16]. The process of development is applied in the lithography to remove the undesired resist after the exposure. However, the development inhibition occurs in high dense areas and results in a remaining undesired resist. This defect in development process is called *develop loading*, which is related to pattern density and has a long-range effect. Since develop loading induces significant CD errors, considering that in the placement can lead to circuit performance improvement. This is especially interesting for the analog integrated circuits, whose electrical performance is strongly dependent on the transistor sizes. Such CD errors, which used to be negligible for the circuits in the old technologies, are considerable for the smaller transistors (especially the gate length) in the advanced nanometer technologies. Therefore, as one of the promising future work, a placer should be developed to tackle the fogging, proximity and develop loading effects simultaneously within reasonable execution time for the analog portion in mixed-signal ICs.

References

[1] A. Fujimura, "Beyond light: The growing importance of e-beam," in *Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2009.

[2] Rachel Courtland, "Intel Now Packs 100 Million Transistors in Each Square Millimeter," 2017.

[3] Bohr, Mark, "Moores law leadership," intel newsroom 28, 2017.

[4] Hasan, R.M.M., Luo, X, "Promising Lithography Techniques for Next-Generation Logic Devices," *Nanomanuf Metrol,* vol. 1, pp. 67-81, 2018.

[5] Mulkens, Jan, et al., "Overlay and edge placement control strategies for the 7nm node using EUV and ArF lithography," in *Extreme Ultraviolet (EUV) Lithography VI. Vol. 9422. SPIE*, 2015.

[6] A. Fujimura, "Design for e-beam: Getting the best wafers without the exploding mask costs," in *Proc. of International Symposium on Quality Electronic Design (ISQED)*, March 2010.

[7] Y. Chang, Ru-Gun Liu and S. Fang, "EUV and e-beam manufacturability: Challenges and solutions," in *Proc. 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015.

[8] L. R. Harriott, "Limits of lithography," in *Proceedings of the IEEE, vol. 89, no. 3, pp. 366-374*, March 2001.

[9] Soichi Owa and Hiroyuki Nagasaka , "Advantage and feasibility of immersion lithography," *Journal of Micro/Nanolithography, MEMS, and MOEMS,* vol. 3, no. 1, 2004.

[10] Wagner, Christian, and Noreen Harned., "Lithography gets extreme," *Nature Photonics,* vol. 4, no. 1, pp. 24-26, 2010.

[11] Sreenivasan, S., "Nanoimprint lithography steppers for volume fabrication of leading-edge semiconductor integrated circuits," *Microsyst Nanoeng,* vol. 3, 2017.

[12] H. Yi, Y. Bao, J. Zhang et al., "Flexible control of block copolymer directed self-assembly using small topographical templates: potential lithography solution for integrated circuit contact hole patterning," *Advanced Materials,* vol. 14, pp. 3107-3114, 2012.

[13] Arjmandi, N., Lagae, L. and Borghs, G., "Enhanced resolution of poly (methyl methacrylate) electron resist by thermal processing," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena,* vol. 27, no. 4, pp. 1915-1918, 2009.

[14] Lin, Burn. (2012). Future of Multiple-E-Beam Direct-Write Systems. Journal of Micro/Nanolithography, MEMS, and MOEMS. 8323. 3011-. 10.1117/1.JMM.11.3.033011. , "Future of multiple-e-beam direct-write systems," *Journal of Micro/Nanolithography, MEMS, and MOEMS,* vol. 8323, pp. 3011-, September 2012.

[15] P. Hudek, U. Denker, D. Beyer, N. Belic, and H. Eisenmann, "Fogging effect correction method in high-resolution electron beam lithography," *Microelectronic Engineering,* vol. 84, pp. 814-817, 2007.

[16] J.-ichi Kon, Y. Kojima, Y. Takahashi, T. Maruyama, and S. Sugatani, "Characterization of fogging and develop-loading effects in electron-beam direct-writing technology," *Japanese Journal of Applied Physics,* p. 51, 2012.

[17] R. F. W. Pease, "Electron beam lithography," *Contemporary Phys,* vol. 22, pp. 265-290, 1981.

[18] T. Figueiro et al., "Extreme long range process effects characterization and compensation," in *Proc. SPIE, vol. 8886*, Oct. 2013.

[19] N. Shimomura, M. Ogasawara, J. Takamatsu, S. Yoshitake, K. Ooki, N. Nakayamada, F. Okabe, and T. Tojo, "Reduction of fogging effect caused by scattered electrons in an electron beam system," in *Proc. SPIE 3748, Photomask and X-Ray Mask Technology VI*, 1999.

[20] K. Morimoto, Y. Ito, R. Gauvin, and M. Kotera, "Measurement of fogging electrons present in scanning electron microscope specimen chamber," *Japanese Journal of Applied Physics,* vol. 59, 2020.

[21] Kamikubo, Abe, T., Oogi, S., Anze, H., Shimizu, M., Itoh, M., Nakasugi, T., Takigawa, T., Iijima, T., Hattori, Y., & Tojo, T., "Proximity effect correction for electron beam lithography: Highly accurate correction method," *Journal of Applied Physics,* vol. 36, pp. 7546-7551, 1997.

[22] C.-H. Seo and K.-Y. Suh, "Reduction of proximity effect in electron beam lithography by deposition of a thin film of silicon dioxide," *Korean Journal of Chemical Engineering,* vol. 25, pp. 373-376, 2008.

[23] J. Chen, Y. -W. Chang and Y. -C. Huang, "Analytical Placement Considering the Electron-Beam Fogging Effect,," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 40, pp. 560-573, 2021.

[24] Y.-C. Huang and Y.-W. Chang. , "Fogging effect aware placement in electron beam lithography," in *Proceedings of IEEE/ACM Design Automation Conference (DAC)*, 2017.

[25] J. Chen, L. Yang, Z. Peng, W. Zhu and Y. Chang, "Novel Proximal Group ADMM for Placement Considering Fogging and Proximity Effects," in *IEEE/ACM International Conference on ComputerAided Design (ICCAD*, 2018.

[26] I. Wu, H. Ou and Y. Chang, "QB-trees: Towards an optimal topological representation and its applications to analog layout designs," in *53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2016.

[27] F. Balasa and K. Lampaert, "Module placement for analog layout using the sequence-pair representation," in *Proceedings Design Automation Conference*, 1999.

[28] P.-N. Guo, C.-K. Cheng and T. Yoshimura, "An O-tree Representation of Non-slicing Floorplan and Its Applications," in *Proc. of DAC*, 1999.

[29] Y.-C. Chang, Y.- W. Chang, G.-M. Wu and S.- W. Wu, "B*-trees: A New Representation for Non-slicing Floorplans," in *Proc. of DAC*, 2000.

[30] Jai-Ming Lin and Yao-Wen Chang, "TCG: A transitive closure graph-based representation for general floorplans," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 13, no. 2, pp. 288-292, 2005.

[31] Kirkpatrick, Scott & Gelatt, C. & Vecchi, M.., "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680, 1983.

[32] K. T. J. L. I. W. a. Y. C. H. Ou, "Layout-Dependent Effects-Aware Analytical Analog Placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 35, pp. 1243-1254, 2016.

[33] H. -C. Ou, K. -H. Tseng, J. -Y. Liu, I. -P. Wu and Y. -W. Chang, "Layout-Dependent Effects-Aware Analytical Analog Placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 35, no. 8, pp. 1243-1254, 2016.

[34] Y. -S. Lu, Y. -H. Chang and Y. -W. Chang, "WB-Trees: A Meshed Tree Representation for FinFET Analog Layout Designs," in *55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018.

[35] H.-F. Tsao et al., "A Corner Stitching Compliant B*-tree Representation and Its Applications to Analog Placement," in *Proc. of ICCAD*, 2011.

[36] W. Li, Y. Lin and D. Z. Pan, "elfPlace: Electrostatics-based Placement for Large-Scale Heterogeneous FPGAs," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.

[37] A. Gusmão, F. Passos, R. Póvoa, N. Horta, N. Lourenço and R. Martins, "Semi-Supervised Artificial Neural Networks towards Analog IC Placement Recommender," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[38] Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhori, E., Wang, S., Lee, Y.J., Johnson, E., Pathak, O., Bae, S. and Nazi, A., "Chip placement with deep reinforcement learning," in *arXiv preprint arXiv:2004.10746*, 2020.

[39] M. Ahmadi and L. Zhang, "Analog Layout Placement for FinFET Technology Using Reinforcement Learning," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[40] A. Tseng, Kuan Chen, C. D. Chen and K. J., "Electron beam lithography in nanoscale fabrication: recent development," *IEEE Transactions on Electronics Packaging Manufacturing,* vol. 26, pp. 141-149, 2003.

[41] Babin, Sergey & Borisov, Sergey & Ivanchikov, A. & Ruzavin, I., "CHARIOT: software tool for modeling SEM signal and e-beam lithography," *Physics Procedia,* vol. 1, pp. 305-313, 2008.

[42] S. Babin, S. Borisov, and E. Patyukova., "Dose variation and charging due to fogging in electron beam lithography: simulations using CHARIOT Monte Carlo software.," in *Proc. of SPIE*, 2013.

[43] S. Babin, S. Borisov, V. Militsin, and E. Patyukova, "Simulation and correction of resist charging due to fogging in electron-beam lithography," in *Proc. of SPIE*, 2013.

[44] A. N. Broers, "Resolution limits for electron-beam lithography," *IBM Journal of Research and Development,* vol. 32, pp. 502-513, 1988.

[45] Parikh, & Kyser, D. F., "Energy deposition functions in electron resist films on substrates," *Journal of Applied Physics,* vol. 50, no. 2, pp. 1104-1111, 1979.

[46] S. Rizvi, Handbook of Photomask Manufacturing Technology, CRC Press, 2005.

[47] Hauptmann, Marc & Choi, K.-H & Jaschinsky, P & Hohle, Christoph & Kretz, J & Eng, Lukas., "Determination of proximity effect parameters by means of CD-linearity in sub 100 nm electron beam lithography," *Microelectronic Engineering,* vol. 86, 2009.

[48] S. Landis, Lithography, John Wiley & Sons, Incorporated, 2010.

[49] Bin Zhao, "RF/analog/mixed-signal IC for wireless communications: challenges and opportunities," in *Asia-Pacific Radio Science Conference*, 2004.

[50] M. Kotera and T. Maekawa, "Simulation of fogging electrons in electron beam lithography," *Japanese Journal of Applied Physics,* vol. 48, no. 6, 2009.

[51] L. Zhang, R. Raut, L. Wang, and Y. Jiang, "Analog Module Placement Realizing Symmetry Constraints Based on a Radiation Decoder," in *Proc. IEEE International Midwest Symposium on Circuits and Systems*, 2004.

[52] L. Zhang, "VLSI circuit layout," in *Wiley Encyclopedia of Computer Science and Engineering*, Hoboken, 2009.

[53] P. Lin, Y. Chang and S. Lin, "Analog Placement Based on Symmetry-Island Formulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 28, no. 6, pp. 791-80, 2009.

[54] Lihong Zhang, C. . -J. R. Shi and Yingtao Jiang, "Symmetry-aware placement with transitive closure graphs for analog layout design," in *Asia and South Pacific Design Automation Conference*, 2008.

[55] L. He, Z. Zhao, Y. Chen and L. Zhang, "Advanced Transitive-Closure-Graph-Based Placement Representation for Analog Layout Design," in *27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020.

[56] J.-M. Lin, G.-M. Wu, Y.-W. Chang and J.-H. Chuang, "Placement with Symmetry Constraints for Analog Layout Design Using TCG-S," in *Proc. of ASPDAC*, 2005.

[57] Rui He and L. Zhang, "Symmetry-aware TCG-based placement design under complex multi-group constraints for analog circuit layouts," in *15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010.

[58] F. Balasa and K. Lampaert, "Symmetry within the sequence-pair representation in the context of placement for analog design," *IEEE TCAD,* vol. 19, pp. 721-731, 2000.

[59] L. Zhang, Y. Zhang, Y. Jiang, and C.-J. R. Shi, "Symmetry-Aware Placement Algorithm using Transitive Closure Graph Representation for Analog Integrated Circuits," *International Journal of Circuit Theory and Application,* vol. 38, no. 3, pp. 221-241, 2010.

[60] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits,* vol. 24, no. 5, pp. 1433-1439, 1989.

[61] Sutton, R. S. and Barto, A. G., "Reinforcement Learning: An Introduction," in *MIT Press*, 1998.

[62] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1798-1828, 2013.

[63] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access,* vol. 7, pp. 53040-53065, 2019.

[64] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabi, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[65] Bertsekas, D. P. and Tsitsiklis, J. N., Neuro-Dynamic Programming, Athena Scientific, 1996.

[66] Li, Yuxi., "Deep reinforcement learning: An overview," in *arXiv preprint arXiv:1701.07274*, 2017.

[67] L. Zhang and U. Kleine, "A novel analog layout synthesis tool," in *Proc. IEEE International Symposium on Circuits and Systems*, 2004.

[68] Abadi, M. et al., "TensorFlow: a system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016.

[69] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR,* vol. abs/1412.6980, 2015.

[70] J. Chen, G. Shi, A. Tai and F. Lee, "A size sensitivity method for interactive MOS circuit sizing," in *10th IEEE International NEWCAS Conference*, 2012.

[71] Črepinšek, Matej & Mernik, Luka, "An efficient representation for solving Catalan number related problems," *International Journal of Pure and Applied Mathematics,* vol. 56, pp. 589-604., 2009.

[72] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, et al., "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016.

[73] S. Han, O. W. Bhatti and M. Swaminathan, "Reinforcement Learning for the Optimization of Decoupling Capacitors in Power Delivery Networks," in *IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, 2021.

Appendix: Published Papers

[1] M. Hajijafari, M. Ahmadi, Z. Zhao, L. Zhang " Fogging-Effect-Aware Mixed-Signal IC Placement with Reinforcement Learning," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), January 2022, (Accepted).

[2] M. Hajijafari, M. Ahmadi, Z. Zhao, L. Zhang " Reinforcement-Learning-based Mixed-Signal IC Placement for Fogging Effect Control," in Proc. IEEE International Symposium on Quality Electronic Design (ISQED), December 2021, (Accepted).