MODELING OF DETERMINISTIC WITHIN-DIE VARIATION IN TIMING ANALYSIS, LEAKAGE CURRENT ANALYSIS, AND DELAY FAULT DIAGNOSIS

A Thesis
Presented to
The Academic Faculty

by

Munkang Choi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
MAY 2007

MODELING OF DETERMINISTIC WITHIN-DIE VARIATION IN TIMING ANALYSIS, LEAKAGE CURRENT ANALYSIS, AND DELAY FAULT DIAGNOSIS

Approved by:

Dr. Linda S. Milor, Advisor
School of Electrical & Computer Engineering
Georgia Institute of Technology

Dr. David C. Keezer
School of Electrical & Computer Engineering
Georgia Institute of Technology

Dr. Gary S. May
School of Electrical & Computer Engineering
Georgia Institute of Technology

Dr. Thomas D. Morley
Mathematics
Georgia Institute of Technology

Dr. Abhijit Chatterjee
School of Electrical & Computer Engineering
Georgia Institute of Technology

Date Approved: [April 3rd, 2007]

To my sons,

Joseph and Joel

ACKNOWLEDGEMENTS

"Now faith is being sure of what we hope for and certain of what we do not see."

(Hebrews 11:1)

I would like to thank God for his help and guidance in my life at Georgia Institute of Technology. I would like to thank my advisor, Linda Milor for exhibiting immeasurable patience and superior guidance during the course of my graduate work. I would also like to thank the committee members of my thesis, Dr. Gary May, Dr. David Keezer, Dr. Abhijit Chatterjee, and Dr. Thomas Morley for their comments and suggestions.

I would like to thank my colleagues in the Semiconductor Test and Yield Enhancement Laboratory at Georgia Institute of Technology. I would like to thank GaTech ECE students, who took the same classes with me. I would like to thank people in Atlanta, Georgia for their kindness and big smile.

I would like to thank Advanced Micro Devices (AMD) and Semiconductor Research Corporation (SRC) for their support and fund.

Last but not least, I would like to thank to my family, my wife, my two sons, my father and mother, my father-in-law and mother-in-law, and my sister for their sacrifice, patience, and prayer.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

As semiconductor technology advances into the nano-scale era and more functional blocks are added into systems on chip (SoC), the interface between circuit design and manufacturing is becoming blurred. An increasing number of features, traditionally ignored by designers are influencing both circuit performance and yield. As a result, design tools need to incorporate new factors. One important source of circuit performance degradation comes from deterministic within-die variation from lithography imperfections and Cu interconnect chemical mechanical polishing (CMP).

To determine how these within-die variations impact circuit performance, a new analysis tool is required. Thus a methodology has been proposed to involve layout-dependent within-die variations in static timing analysis. The methodology combines a set of scripts and commercial tools to analyze a full chip. The tool has been applied to analyze delay of ISCAS85 benchmark circuits in the presence of imperfect lithography and CMP variation.

Also, this thesis presents a methodology to generate test sets to diagnose the sources of within-die variation. Specifically, a delay fault diagnosis algorithm is developed to link failing signatures to physical mechanisms and to distinguish among different sources of within-die variation. The algorithm relies on layout-dependent timing analysis, path enumeration, test pattern generation, and correlation of pass/fail signatures to diagnose lithography-caused delay faults. The effectiveness in diagnosis is evaluated for ISCAS85 benchmark circuits.

CHAPTER 1

INTRODUCTION

Scaling of semiconductor processing has enabled the production of higher performance, increasingly complex products, at lower cost. However, nano-technologies are associated with increasing numbers of unsimulated features and design-process interdependencies [1],[2]. Important sources of a discrepancy between design and manufacturing relate to deterministic within-die variation, whose physical origins are imperfections in lithography, which creates nonuniformity in printed geometries, interconnect thickness variation caused by CMP, etc. As a result, rigorous and careful design with today's tools result in designs that face heightened risk in manufacturing compared to previous technology generations.

1.1 Discrepancy between the Designed Circuit and the Manufactured Circuit

Consider, for example, variation in channel length as a function of position in the die, as shown in Fig. 1.1. Since digital designers primarily use transistors with the smallest possible channel length and attempt to optimize speed by designing circuits with many critical paths, speed is limited by those critical paths located in areas with transistors with longer channel lengths ((3, 3) in the figure). If these channel lengths are larger than those used during design, performance targets will not be achieved. This will result in reduced yield.

Manufacturers compensate for this problem by adjusting the channel length target to be a smaller value. However, reducing the channel length also risks lowering yield in part because when the channel length is too small, transistors exhibit excessive leakage current and do not turn off properly. In addition, patterning problems result, including

poly opens. Clearly, given the channel length distribution in Figure 1.1, yield is limited by those transistors with the smallest channel lengths ((1, 1) in the figure).

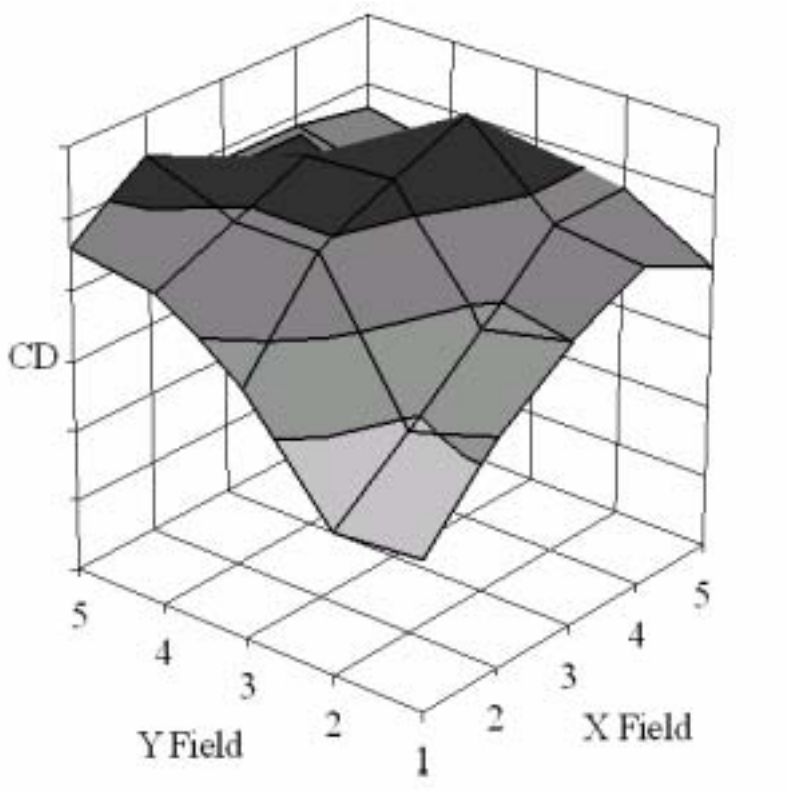


Figure 1.1. Example gate poly CD spatial map from industry, showing how the gate poly CD varies as a function of position in the reticle field.

Note that when optimizing the target transistor channel length the transistors that provide the lower limit due to excessive leakage current are different than those that provide the upper bound due to speed constraints. Moreover, if the range of channel length variation increases across a die, it becomes increasingly challenging to achieve acceptable circuit performance with acceptable yield. Therefore, it is important to reduce the range of channel length variation across a die.

1.2 Within-Die Variation

Within-die variation includes both random and systematic variation. The systematic component consists of repeatable patterns. The foundry typically collects data on such patterns by running test structures designed by process modules (lithography, etch, thin films, implant, etc.) and uses the data collected from the test structures to monitor module performance.

Systematic within-die variation may be corrected by modifying the mask. Optical proximity correction and phase shift masks compensate for the effect of neighboring features (as caused by the proximity effect) [3], [4]. The introduction of dummy features can create uniform feature densities, thereby reducing variability from factors such as flare and chemical mechanical polishing (CMP) [5], [6]. However, both approaches expand the size of the database prior to tapeout and increase mask write times and cost [7],[8]. Improved correction may also involve modifying correction as a function of chip location in the reticle and/or matching mask sets with a specific equipment set, which further increases mask write times and cost. Moreover, some sources of within die variation depend on the specific equipment set used by the manufacturer and may change over time due to lens heating, for example. Therefore, perfect correction is not possible. Consequently, it is important to determine which sources of within die variation impact circuit performance and by how much, before and after applying various approaches to correction. In addition to that, the physical cause of failures must be diagnosed to choose the proper correction strategy.

1.3 Previous Work and Their Limitations

1.3.1 Circuit Analysis with Systematic Variation

Analysis of the impact of within-die variation on circuit performance is very different than previous approaches to analyzing the impact of process variation on circuit performance. The traditional approach to analysis of the impact of process variation is worst case analysis [9]. Worst case analysis aims to provide upper and lower bounds on circuit speeds, given lot-to-lot, wafer-to-wafer, and within wafer variation in process parameters. Worst case analysis typically focuses on process variations that impact transistors, causing them to provide a range of drive currents in saturation. The drawbacks of traditional worst case analysis are summarized well in [10]. These drawbacks include the neglect of interconnect delay, which is increasing. In fact, worst case corners are typically not even determined for interconnect [11]. In addition, within-die variation is neglected. Within-die variation has been shown to systematically degrade circuit speed [12], and is increasingly significant [2],[13],[14].

The earliest attempts to include within-die variation in circuit analysis have focused on optical effects and CMP variation. Stine *et al.* [15] developed a tool that accounts for the proximity effect by using an aerial image simulator. The modified layout is re-extracted in order to generate the modified netlist, which is then used in HSPICE [17] simulations of critical paths. In [16], Chen *et al.* proposed to determine circuit speed in the presence of the proximity effect. The approach involved modifying channel lengths of transistors in HSPICE files for critical paths based on the circuit's layout and the neighboring features of each gate. HSPICE is then used to revise the critical path delays. In [12], Orshansky *et al.* improved the previous approach by

considering both the proximity effect and lens aberrations and by replacing HSPICE simulations with static timing analysis. Mehrotra *et al.* [18] modified interconnect parameters in HSPICE files, caused by thickness variation from CMP, and enabled variational analysis without repeated re-extraction of the netlist. However, this approach relies on HSPICE for critical path simulation, which limits its utility for industrial circuits.

In an alternative approach to address systematic within-die variation, Gattiker *et al.* [19] demonstrates an efficient approach for industrial designs involving static timing analysis of circuits with die-to-die and within-die variations. This approach considers just cell-level within-die variation, not transistor-level variation. The problem with cell-level analysis comes about when transistor and interconnect geometries not only vary as a function of position within a chip, but also vary as a function of neighborhood. Consider, for example, variation of channel lengths of vertically oriented transistors as a function of neighborhood, shown in Figure 1.2. It can be seen from Figure 1.2(a) (transistors in the center of the chip) that transistors that are isolated on the right and have nearby features on the left have the smallest channel lengths and transistors that are isolated on the left and have nearby features on the right have the largest channel lengths. However, variation as a function of neighborhood interacts with variation as a function of location. Specifically, transistor channel lengths as a function of neighborhood are also shown in Figure 1.2(b) for transistors located in the left-upper (1, 5) corner of the chip. Clearly, correction based on data in Figure 1.2(a) from the center of the chip will not be optimal for the transistors in Figure 1.2(b) in the left-upper corner of the chip. Variation as a

function of both neighborhood and location requires more detailed analysis at the transistor level on how cell delays vary.

Recently, many authors have looked into statistical static timing analysis [14],[20],[21]. These authors assume that within-die variation can be characterized statistically, with correlation functions, and compute an upper bound on delay. These approaches also operate at the cell level and also can't include the sources of variation in Figure 1.2.

In addition, it should be noted that within-die variation is a combination of systematic and random components. Therefore, the most accurate timing information can be achieved by simulating both the systematic and random components. The systematic component creates a shift in the mean delay, which depends on details of the chip layout and process technology. As will be demonstrated, it does not just depend on the spatial proximity of gates. The purpose of this work is simulation of this systematic component. Analysis of systematic variation should be supplemented with analysis of random variation, which may include spatial correlation. Clearly, simulation of only one of these components provides an incomplete picture of circuit timing.

This work improves on previous work on systematic within-die variation by including all of the physical models for lithography and CMP previously presented in [12],[15]-[16],[18], but enables efficient full chip timing analysis. On the other hand, unlike other full chip timing approaches, such as the approaches presented in [14],[19]-[21], transistor level detail is included in the analysis, as is required when analyzing some sources of systematic within-die variation.

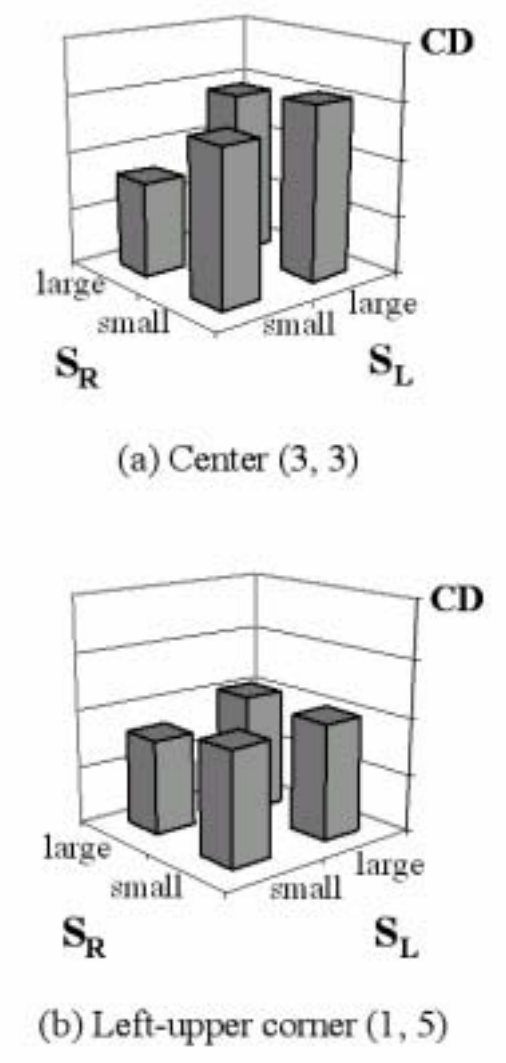


Figure 1.2. Gate poly CD as a function of neighborhood. The CD varies as a function of position in the reticle field. The figure contains two locations, the center in (a) and the left-upper corner in (b). The coordinates refer to locations in Figure 1.1.

1.3.2 Background of the Delay Fault Diagnosis

1.3.2.1 Existing Approaches to Detecting Within-Die Variation

Process variations are parametric faults, which are diagnosed with correlation analysis. Specifically, the correlation is determined between the yield for a collection of wafers and the average measurement for all test structures in the scribe line. If the correlation is high for a specific test structure, the parameter associated with the test structure is the likely cause of yield variation. For example, it may be determined that low yielding wafers are associated with high resistance vias.

If variation exhibits patterns within a wafer, correlations are performed between yield in specific wafer sectors and test structure measurements to identify the cause of variation.

The specific faults considered in this work are not easily diagnosed with correlation analysis, because faults caused by lithography result in variations within a reticle, rather than within a wafer. Typically, only a single copy of each test structure is included in each reticle. Therefore, the scribe line does not have sufficient granularity to detect such variation. Moreover, even if the scribe line were populated with sufficient test structures covering variation within a reticle, the range of sources of variation from lithography is sufficiently large that multiple copies of many test structures in many positions would be required, which is not practical. Therefore, this thesis attempts to determine if such faults can be diagnosed via product tests.

1.3.2.2 Path Delay Fault

Within-die variation from lithography causes a circuit's speed of operation to be reduced. Such variation can be thought of as a distributed manufacturing defect that cumulatively increases delays within circuit paths. Therefore, the path delay fault model [22] addresses this failure mode.

Path delay faults are sets of paths that can be sensitized by a transition or sets of transitions at the primary inputs. It has been shown that only a subset of such paths need to be tested to guarantee temporal correctness of a circuit [23], and this set is independent of component delays. The set of paths that need to be tested to guarantee temporal correctness is called the set of primitive delay faults. Several approaches have been developed to identify single and multiple primitive delay faults [24]-[26] and to generate the appropriate test sets for these faults. However, these methods are applicable to moderately sized circuits, as indicated in [24]. Sharma *et al.* [27] propose to overcome this problem by covering delay faults on robustly untestable critical paths by robustly testing their longest possible segments that are not covered by any of the testable critical paths.

Many of the paths associated with primitive delay faults have delays that are much less than the clock period under normal operation. Delay faults on such paths can only be detected during normal operation if the delay fault is large. Consequently, in order to detect smaller delay faults, it is desirable to find a set of longest paths. Several papers have been published relating to the selection of the longest critical paths [27]-[32]. Most papers have focused on selecting paths to ensure topological coverage. Specifically, in [28], [29], paths are selected to cover each gate in the circuit, i.e. the set

of longest paths through each gate is found. As a result, small delay defects associated with a gate can be detected. To ensure testability, these methods check path sensitization, but do not consider whether their tests are associated with primitive faults. Moreover, in [27]-[31], the calculated delays of the critical paths are based on discrete-valued timing models that don't take into account the signal propagation effect (signal transition slope dependency). Wang *et al.* [32] introduced the concept of path correlation in critical path selection using a statistical timing model. The statistical timing framework has the potential to properly deal with coupling noise, temperature gradients, power supply gradients, and across-chip linewidth variation [33], but determining the characterization (probability distribution functions and their correlations) of the underlying transistors and wires is a major unsolved challenge [33],[34].

1.3.2.3 Delay Fault Diagnosis

Diagnosis is the process of identifying the cause of failure. There are a variety of causes for delay faults. They include local failure mechanisms, such as resistive shorts and opens, and other mechanisms, such as crosstalk-induced delay, delay due to power supply noise, and delay due to process variations. Traditionally, delay fault diagnosis has focused on local failure mechanisms. Specifically, the goal of diagnosis is the localization of physical defects in failing circuits, in order to identify the root cause. Localization involves analyzing input vectors and output responses to determine the defect location.

Methods for localization can be classified as cause-effect and effect-cause analysis [35]. Cause-effect analysis pre-computes faulty behavior based on the assumed fault model and stores the information in a fault dictionary. The behavior of a failing

chip is compared with the fault dictionary to identify the most probable faults. Effect-cause analysis involves searching backwards from the failing outputs, deducing internal values, to identify locations of probable faults.

Underlying the diagnosis process is the fault model. A variety of fault models exist that differ from each other based on fault complexity (stuck-at vs. resistive opens or shorts), temporality (static or dynamic), and cardinality (single or multiple). Many papers have addressed diagnosis for a variety of fault models, including stuck-at faults, bridging faults, and even Byzantine defects [36], where defects result in intermediate voltage levels at a gate output and the corresponding fanout branches are associated with different logic values due to the different logic thresholds of subsequent gates.

i) Gate Delay Fault Diagnosis

Resistive shorts and opens, crosstalk, and power supply noise are among the failure mechanisms that may not cause stuck-at failures, but rather may cause single or multiple transistor delay faults. Several papers [37]-[39] have presented diagnostic methodologies to isolate delay faults associated with gates (gate delay faults and transition delay faults). Girard *et al.* [37] proposed a method to diagnose gate delay faults based on critical path tracing. The method involves logic simulation only, together with tracking of transitions for sets of patterns. If a transition results in a failing output, the gates in the paths sensitized by the transition are stored as potential sites for gate delay faults. The method accounts for potential glitches through the use of six-valued logic simulation. Wang *et al.* [38] improves the resolution of transition delay fault diagnosis through pruning impossible fault candidates using circuit timing information. Krstic *et al.* [39] goes beyond this approach by linking diagnosis to statistical timing

analysis. Specifically, in [39] the delay fault potentially associated with each gate is assumed to be probabilistic, together with the rise and fall times of the gates. Statistical timing is combined with the probabilistic fault model to construct a fault dictionary to provide probabilities of failure for all input transitions and faults. However, the delay distribution of each circuit element is assumed to be known, together with correlations among elements, and statistical characterization information for all instances is not easily available [33],[34].

ii) Path Delay Fault Diagnosis

Gate delay fault diagnosis focuses on localizing the cause of a delay fault. However, some failures may be due to small delay variations in a number of gates that accumulate to produce a delay fault. Path delay fault diagnosis addresses the isolation of such faults. Specifically, path delay fault diagnosis involves locating input-output paths in a chip that cause the delay fault. Several methods have been proposed to address this problem [40]-[43].

Diagnosis procedures generally start with a complete fault list, which is pruned by analyzing the applied tests and responses. In the case of path delay fault diagnosis, the set of potential faults is all sensitizable paths in a circuit. Therefore, the initial set of faults is exponentially large. When random tests are applied to a circuit, such tests sensitize a number of single and multiple path delay faults. Pant *et al.* [40] address this problem using an effect-cause approach where, first, the set of paths sensitized by each failing vector is determined, and, second, those paths that have been robustly tested by other passing vectors (guaranteed to be delay fault free) are removed from consideration. The result is the suspect set.

Padmanaban *et al.* [41] improved this approach by further pruning the suspect set by eliminating paths (single and multiple path delay faults) that pass validatable non-robust tests.

To further guide diagnosis Sivaraman *et al.* [42] and Krstic *et al.* [43] introduce a statistical framework. To aid in diagnosis, Sivaraman *et al.* [42] limits test patterns to those that provide single multipath robust tests. And, since each of the test vector pairs has incompletely specified inputs, the unspecified inputs are set to minimize the number of primary inputs that have transitions so that the number of side paths that get sensitized is minimized. Then, given a set of failed tests, the sets of sensitized paths are determined. For each sensitizable path, a model of process parameter variations is used together with Monte Carlo analysis to find statistical distributions of slack for each path and to weight potential sites for delay faults. Therefore, sites for likely faults are selected if the corresponding path is sensitized by a test which violates a timing constraint, and sites are more probable fault sites if tighter timing constraints are placed on the paths through them. In this way, the method in [42] provides better feedback about the location of faults than [40],[41].

Like [42], Krstic *et al.* [43] propose a similar path delay fault diagnostic framework, involving three steps. First, effect-cause analysis identifies a suspect set through logic analysis of failing patterns. Second, cause-effect analysis reduces the suspect set through statistical timing simulation in the presence of various error sources (modeling errors, single-site random size timing errors, etc.). And third, the failure mechanism is linked to potential error sources by comparing simulation results from a collection of circuit instances to the fault dictionary and voting among the faults.

The problem with these approaches is that they just pinpoint a sub-path responsible for circuit failure, and not the underlying physical cause. Hence, these algorithms must be followed with physical analysis in order to provide useful information. What is needed is physical evidence of the cause of failure so that appropriate action can be taken.

1.4 The Purpose of the Thesis

1.4.1 Within-Die Variation in the Design Analysis Flow

The purpose of this thesis is to provide a methodology to determine the impact on circuit speed of within-die variation from lithography and Cu interconnect chemical mechanical polishing (CMP). The specific sources of variation considered in this thesis include the proximity effect, lens aberrations, and flare in lithography, which impact the gate critical dimension (CD) and interconnect linewidth, and copper interconnect chemical mechanical polishing, which impacts interconnect thickness. For these sources of variation physical models have been developed and a methodology has been implemented to translate these models into static timing analysis. This enables the analysis of how these sources of systematic within-die variation degrade circuit speed. In addition, with the assistance of a tool to compute chip leakage current in the presence of systematic within die variation, the tradeoff between circuit speed and leakage current can be analyzed.

Given a method to analyze and compare the relationship between specific sources of systematic within-die variation, efforts can be made to reduce or correct the most significant factors. Specifically, if the proximity effect degrades circuit speed to a greater extent than variation from CMP, then the logical choice would be to focus greater effort

and funding on correcting the proximity effect, rather than CMP. The tool presented in this paper enables such an analysis, and consequently, it can assist in the decision-making process related to process improvement and the optimization of mask correction.

Alternatively, the analysis provided by the tool to be presented may indicate circuit performance degradation due to factors that cannot easily be corrected, such as lens aberrations that change over time and whose patterns are unique to each specific stepper in a foundry that has multiple steppers. In this case, data on these factors should be collected over time and for all stepper systems. Based on these datasets, the tool enables the designer to determine performance degradation as a function of stepper and the expected range of performance degradation over time. This allows the designer to appropriately guard-band circuit timing so that performance requirements are more likely to be achieved with first silicon without extensive over-design.

1.4.2 Linking Diagnostic Results to Physical Failure Mechanisms

In order to lead to corrective actions, diagnostic procedures must go beyond identifying the failing path to determining the physical mechanism causing the failure. To this end, several papers have proposed test pattern generation to detect crosstalk-induced delay [44]-[46], power supply noise [46],[47], and resistive open and short defects [46]. In Chen *et al.* [44], it is demonstrated that crosstalk can lead to delay faults, and test patterns are generated for a set of user-supplied single crosstalk-induced delay faults. Krstic *et al.* [45] extends this work by adding methods to select crosstalk faults based on performance sensitivity analysis. Moreover, once the paths have been selected, a genetic algorithm is used to find the test patterns.

Similarly, Krstic *et al.* [47] identifies path delay faults associated with power supply noise through performance sensitivity analysis, with a statistical dynamic timing analysis framework. Patterns are found that sensitize the faults using a genetic algorithm, which assigns unspecified primary inputs such that the power supply noise impact on the delays of signals is maximized.

Finally, in addition to crosstalk-induced and power supply noise-induced delay faults, Liou *et al.* [46] considers interconnect delays coming from resistive open and short defects. Again, a similar methodology is used to identify faults and to select test patterns to detect the faults.

This thesis is similar to these papers in that it aims to design test patterns for specific failure mechanisms that can be used to activate specific sources of delay faults. It is different because the focus is not on detecting design issues (crosstalk, power supply noise), but instead targets detection of process problems (within-die variation from lithography). Like crosstalk and power supply noise, within-die variation is not simulated during conventional design, and therefore designs are vulnerable to yield loss as a result. Moreover, process monitors in the scribe lines cannot be used for diagnosis. Hence, all of these failure mechanisms are difficult to diagnose.

This thesis differs from the above papers in that the focus is not just to detect the cause of failure but also to use the generated test patterns to provide diagnostic information on the physical causes to the chip manufacturer. Hence, the set of failures is directly linked to corrective actions.

1.5 Organization of the Thesis

This thesis is organized as follows. In Chapter 2, the origins of systematic within-die variation will be reviewed. It is also explained how various sources of systematic variation are modeled in this work. Chapter 3 describes our timing analysis methodology (layout-dependent timing analysis). Chapter 4 describes the method to estimate leakage current. The diagnosis methodology of the optical lithography faults will be addressed in Chapter 5. The applications, three analysis examples and one diagnosis example, are presented in Chapter 6. The thesis is summarized in Chapter 7.

CHAPTER 2

THE CAUSES OF SYSTEMATIC WITHIN-DIE VARIATION

Modern semiconductor manufacturing suffers from several sources of systematic within-die variation. Moreover, systematic within-die variation is increasing relative to other sources of variation (lot-to-lot, wafer-to-wafer, within wafer) in recent technology generations.

The characteristics of the MOS transistor are primarily affected by gate length, gate oxide thickness, and dopant density fluctuation. The gate oxide thickness is better managed than the other sources of variation [48]. Recently, dopant fluctuation has become an issue, as the standard deviation of the threshold voltage caused by random dopant fluctuation is inversely proportional to the square root of the gate length and width [49]. However, the focus of this work is on variation in the gate channel length caused by variation in the gate CD.

Figure 2.1 illustrates the pattern-defining flow. Each step is associated with variation. For example, the mask making process is vulnerable to problems similar to pattern definition on wafers. Gate CD variation is also impacted by the increasing mask error factor (MEF) [50]; the gate CD trim, a component of photoresist patterning, which is a function of layout patterns [51]; and the fact that during the etching process, wide trenches have abundant radicals for the passivation film on the trench side wall, while narrow ones have a limited supply of radials, also creating pattern dependency [52].

The following sections will review some of these physical factors and how they are modeled. The focus of this thesis is on optical lithography and copper CMP. However, other factors that are layout dependent can be treated similarly, since they will

also depend on feature neighborhood, geometry, and location. Therefore, the methodology can be generalized to other process features.

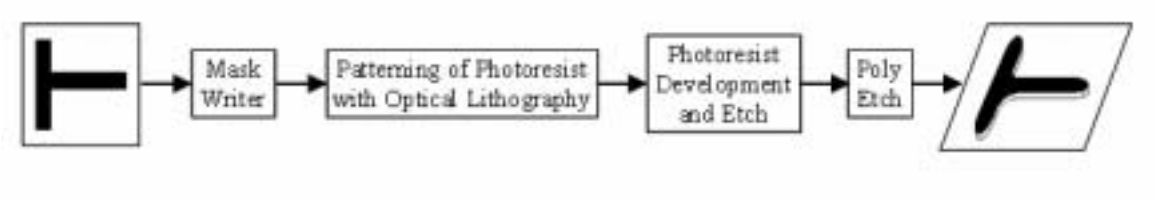


Figure 2.1. Pattern-defining flow for gate poly, from the layout to the wafer.

2.1 The Proximity Effect

The gate CD is a function of its neighborhood due to the proximity effect [4], [53]. Specifically, the proximity effect causes linewidths in dense areas to be different than linewidths in isolated areas, line-end shortening, and corner rounding. The proximity effect is caused by variations in light intensity during exposure of the photoresist, resulting from the presence of neighboring features. The light intensity as a function of distance from a printed feature is shown in Figure 2.2, for both an isolated and a dense feature. This intensity variation modifies the exposure of photoresist on gate edges, which in turn translates into systematic variation in gate CDs.

The neighborhood is accounted for by determining the distance to the nearest poly geometry on the left and on the right of each transistor gate, as in [16]. Each transistor, therefore, has two labels, the distance to the nearest poly geometry on the left and the distance to the nearest poly geometry on the right, assuming a vertical orientation. Labels for horizontal transistors correspond to distances to the nearest poly geometry above and below the feature. These two labels combine to determine the category of each gate.

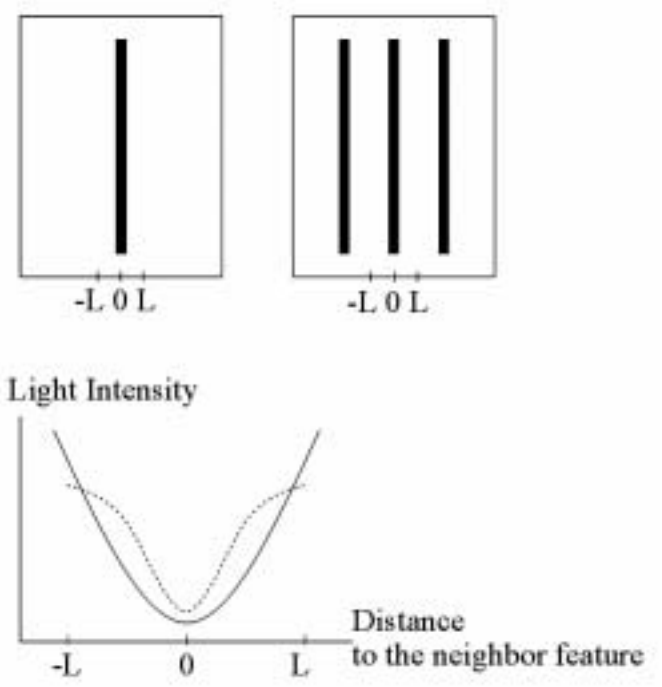


Figure 2.2. The optical proximity effect [4]. (The solid line is for the isolated line and the dashed line for the dense line.)

The script has been implemented with *Mentor Graphics Calibre* [54]. This two-distance model is an approximation. It is possible that the distances may be different at different points within a transistor, as discussed in [16]. Transistors may also be impacted by the second closest feature for the newest technologies. And, line edge roughness (LER) ensures non-uniform spacing among transistors [55]. These kinds of non-uniformity can be handled by averaging distances, prior to categorization, or under the assumption that they modify the channel dopant distribution [56], weighted averaging may be performed.

In the examples, the distances to the left and to the right are labeled as n1 to n5, where n1 is the minimum poly spacing Smin. The largest distance is n5, which corresponds to all

distances greater than 2.5Smin. These distance categories have been chosen arbitrarily, but they conform to common distances seen in a layout, i.e. minimum poly spacing, minimum poly spacing if the space contains a contact, etc. The distance categories are illustrated in Figure 2.3. Since we have five distance categories in each direction (left and right), all possible combinations of distance categories result in a total of 25 categories for vertical transistors and 25 categories for horizontal transistors.

A similar classification has been used for segments in local and global interconnect.

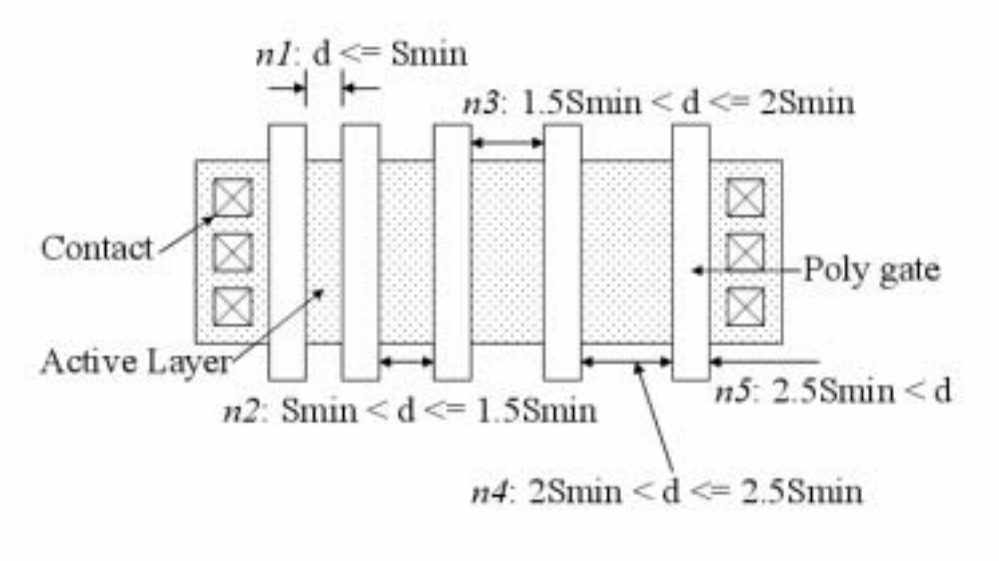


Figure 2.3. Distance categories for poly gates. Smin is the minimum space design rule between two poly lines with no contacts in between them.

The proximity effect between cell instances was ignored in [16]. However, for advanced technologies, it may no longer be appropriate to ignore inter-cell effects.

Therefore, we have included the inter-cell proximity effect by adding a label to the cell instance name, as in Figure 2.4. In the figure the nand3_2x cell instance has four poly gates close to the cell edges. Based on the placement in the layout, the distances from these poly gates to poly patterns in adjacent cell instances correspond to categories n3, n4, n5, and n5. As a result, the cell instance is relabeled as nand3_2x_3455. The adjacent cell instances are determined by the placement report, where the distances of all poly patterns to cell edges are pre-characterized and incorporated in the cell instance name. Therefore, determining the gate categories requires a look up in the placement report and analysis of the poly distance profile of adjacent cell instances.

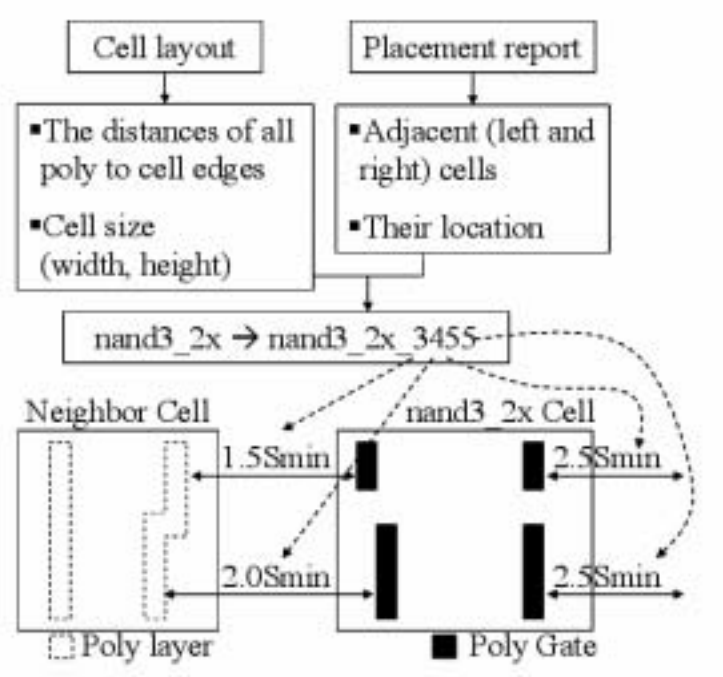


Figure 2.4. How neighboring cells are taken into account in the analysis of the proximity effect by attaching labels to the cell instance name. (It is assumed that rows of cells are placed with matching transistor orientations.)

2.2 Lens Aberrations

Lenses have imperfections, which can be described by aberrations, as shown in Figure 2.5. These aberrations create optical path differences (OPD) for each ray through the lens. OPDs can be decomposed into spherical aberrations, astigmatism, Coma, etc. [57], [58]. Data on lens aberrations is typically collected by fabricating arrays of transistors or resistors with varying neighborhoods in different positions within the reticle on test chips. Because aberrations depend on the lens system and settings used in lithography, data is collected for each system to characterize variability and to optimize settings.

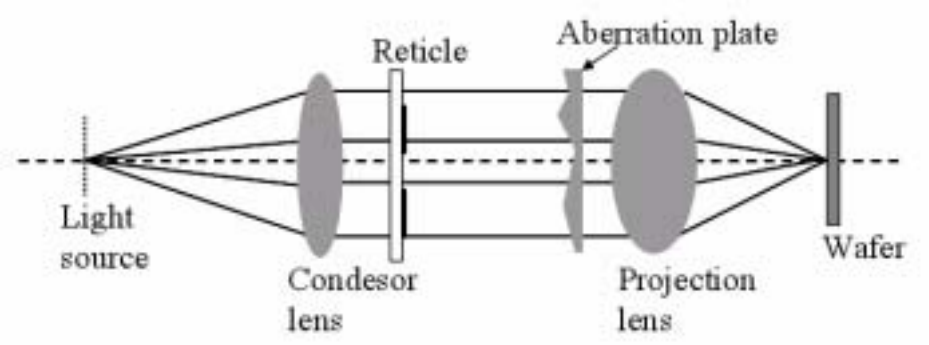


Figure 2.5. Modeling of lens aberrations typically involves distortions introduced by an aberration plate which affects the distance from the light source to the wafer [57].

Accounting for lens aberrations involves determining the location of the pattern in the layout and its neighborhood [12]. The placement and routing tool (in this thesis, *Cadence Silicon Ensemble* [59]) gives information about the location of the cells and global interconnect.

In order to analyze the impact of lens aberrations, the layout is partitioned by a grid. The location of each cell is looked up with respect to the grid, and a label is attached to the cell name indicating its location, as shown in Figure 2.6. Because lens aberrations and the proximity effect interact, gate CDs are a function of their location *and* neighborhood. Since gate CD variation for different proximity categories has different location-dependencies, each proximity effect category has its own CD map [12]. Thus the CD of any gate poly in the layout is a function of the location tag of the cell name and the tagged gate neighborhood in the HSPICE file.

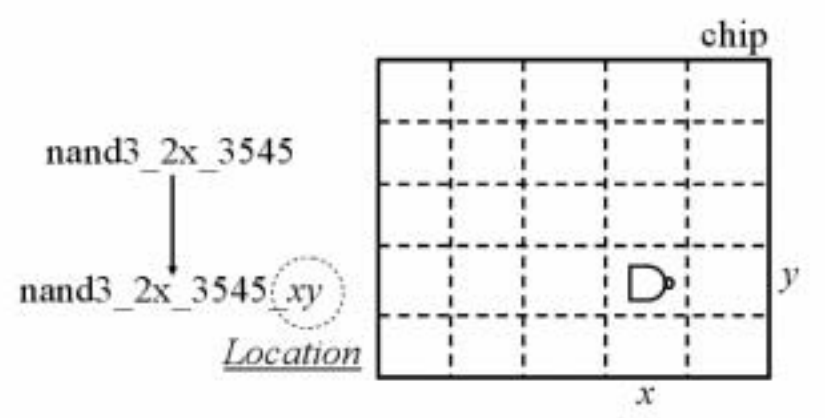


Figure 2.6. Cell instance names are modified according to location in order to analyze the impact of lens aberrations.

Coma is a lens aberration that depends on both the neighborhood and location [57]. We have focused on Coma because Coma becomes severe when making use of resolution enhancement techniques such as phase shift masks (PSM) and off-axis illumination (OAI) [60]−[62].

Analyzing Coma requires distinguishing between features to the left and features to the right of a specific pattern, since patterns with asymmetric categories are printed on the wafer differently as shown in Figure 2.7.

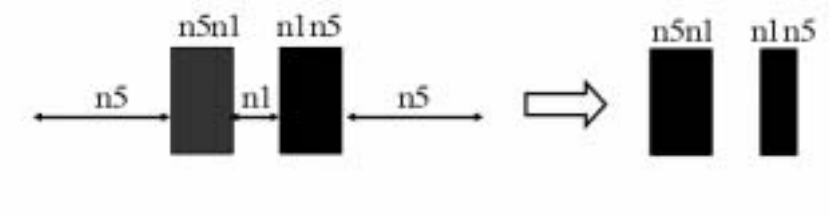


Figure 2.7. The original patterns with equal widths will have printed patterns with different widths due to Coma. Therefore, modeling Coma requires labels that differentiate between features to the right and to the left of vertically oriented transistors, such as the labels n5n1 vs. n1n5 in the figure.

If a cell instance is flipped during the placement and routing step, the labels of all gates need to be reversed, i.e. distances to the left become distances to the right, and vise versa. We indicate this by adding a label to the cell instance name indicating if the cell instance has been flipped (f) or not (n).

2.3 Flare

Flare results from the unwanted scattering and reflections of the optical system, as in Figure 2.8 [63]−[65]. Flare causes CD variation since more stray light scatters under the dark regions on the mask as shown in Figure 2.8 [63], [64]. Local flare depends on the density of chrome in the mask [64], [65]. Therefore, in the tool the gate CD and

interconnect linewidth are determined by computing the percent chrome of the mask in the neighborhood of the pattern. The range of the neighborhood is currently not well understood, and therefore, in the tool it is a user input.

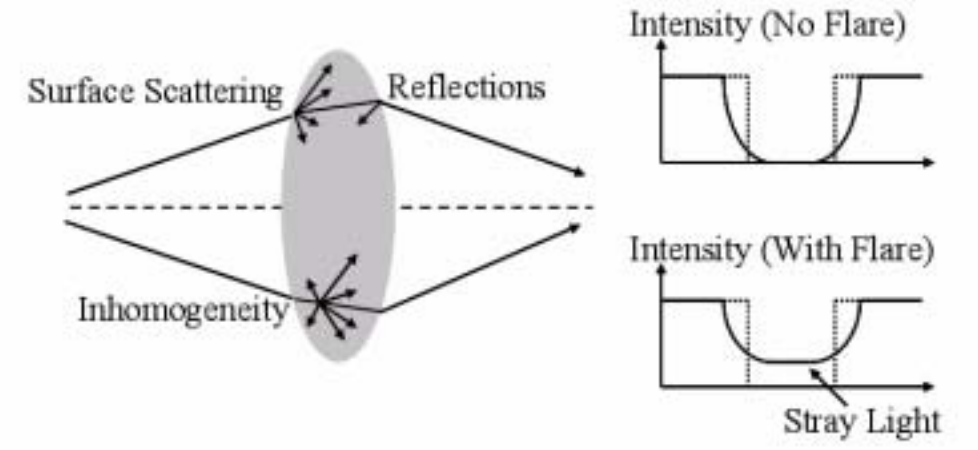


Figure 2.8. Flare [63] is caused by surface scattering, inhomogeneity, and reflections in the optical lithography system, as shown here. The result is stray light that exposes photoresist. If a sector of the mask is very dense, having many patterns, stray light will influence the CD of the printed geometries.

To obtain the pattern density, the chip is divided into sectors. As the pattern objects in the GDS file are scanned sequentially, it is decided in which sector and which layer the pattern is located. As patterns are added to a sector in a layer, the density of that sector in that layer increases. In this way, we obtain the pattern density by reading the GDS file once. The pattern density of each sector is recorded in the pattern density file, and the cell name is tagged by the sector, as was done for lens aberrations. The procedure is shown in Figure 2.9. In this thesis, effective pattern density is calculated using the

square weighting function, as in [66].

In order to determine the CD accounting for flare, the sector tag of the cell name is used as an index to look up the appropriate pattern density and the corresponding CD values.

```
Allocate array of sectors for all layers.
Initialize array to zero.

For all patterns in the GDS file {
    Fracture the pattern into rectangle sub-patterns.
    For each sub-pattern {
        Determine its sector and layer.
        Calculate the area of the sub-pattern.
        Increase the entry in the array for the sector and
        layer by the area.
    }
}
```

Figure 2.9. Procedure for pattern density calculation.

2.4 Chemical Mechanical Polishing

It is well known that Copper CMP has pattern dependent problems, such as metal dishing and dielectric erosion [67], [68].

Copper CMP consists of three intrinsic stages: bulk copper removal, barrier metal removal, and overpolishing, as shown in Figure 2.10 [68]. The bulk copper removal rate is proportional to the pressure of the polishing pad. The bulk copper removal rate is dependent on layout patterns, since the initial copper topography is conformal to the

underlying patterns. The result is a thinner copper layer covering areas with dense patterns. Then, although areas with a thicker copper layer are exposed to higher pressure, this does not completely compensate for differences in the initial topography. As a result, copper polishing reaches the barrier metal first in areas that start with a thinner copper layer. Bulk copper and barrier metal have different material characteristics. This modulates the removal rate once the barrier layer has been reached and causes barrier removal to be pattern-dependent. Finally, overpolishing is required to avoid shorts between adjacent metals. It is also pattern-dependent due to the material differences between dielectric and copper. In summary, the initial topography, combined with variations in the material removal rates, cause variation of the metal line thickness, which is dependent on the line width, pattern density, and line spacing.

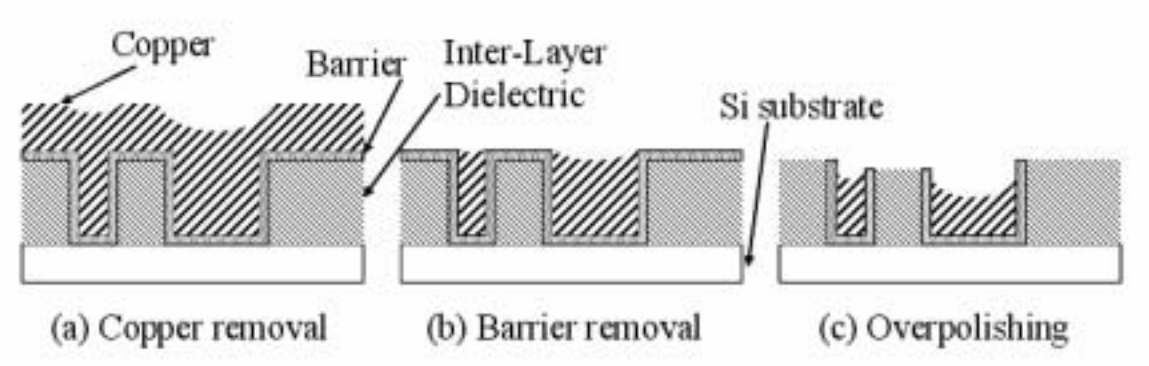


Figure 2.10. Copper CMP flow. It involves patterning of the dielectric, copper deposition, copper removal via polishing, barrier removal, and overpolishing to ensure that the barrier is removed throughout the wafer surface.

In the methodology, dishing in Figure 2.11(a) causes interconnect thinning as the linewidth increases. Erosion leads to thin interconnect in dense areas, as shown in Figure 2.11(b). Normally, dense areas have thinner interconnect than isolated areas. However, if the density of the underlying layer is high, the thickness is less impacted, as shown in Figure 2.11(c). It is reported that the pattern interaction distance is 25μm [69], so a rectangle window of 50μm is assumed in the examples for the calculation of the effective pattern density.

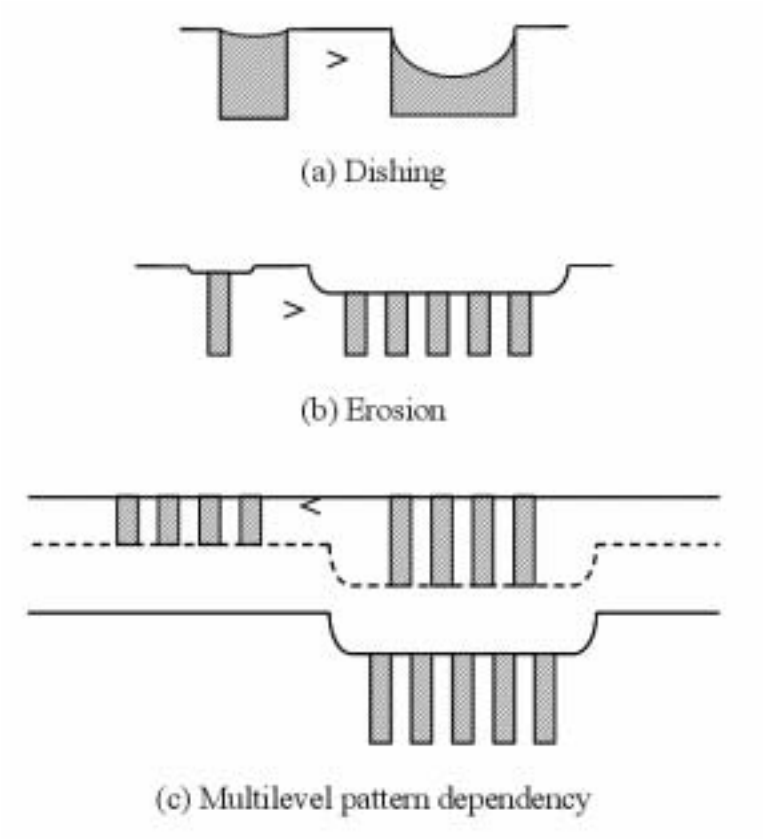


Figure 2.11. Causes of metal thickness variation caused by CMP. (a) Dishing refers to thinning of wide lines. (b) Erosion refers to thinning of lines in dense areas of the mask. (c) Because interconnect involves many layers, erosion on one layer propagates to higher layers, increasing or decreasing erosion in subsequent layers.

2.5 Summary of Lithographic and CMP Variation Data Collection Methods

Variation can be classified into lot-to-lot, wafer-to-wafer, within-wafer, and intra-die variation, as shown in Figure 2.12.

To manage the lot-to-lot and wafer-to-wafer variation, data are sampled from each lot or each wafer and they are monitored by conventional statistical methods (statistical process control). A typical control chart is shown in Figure 2.13.

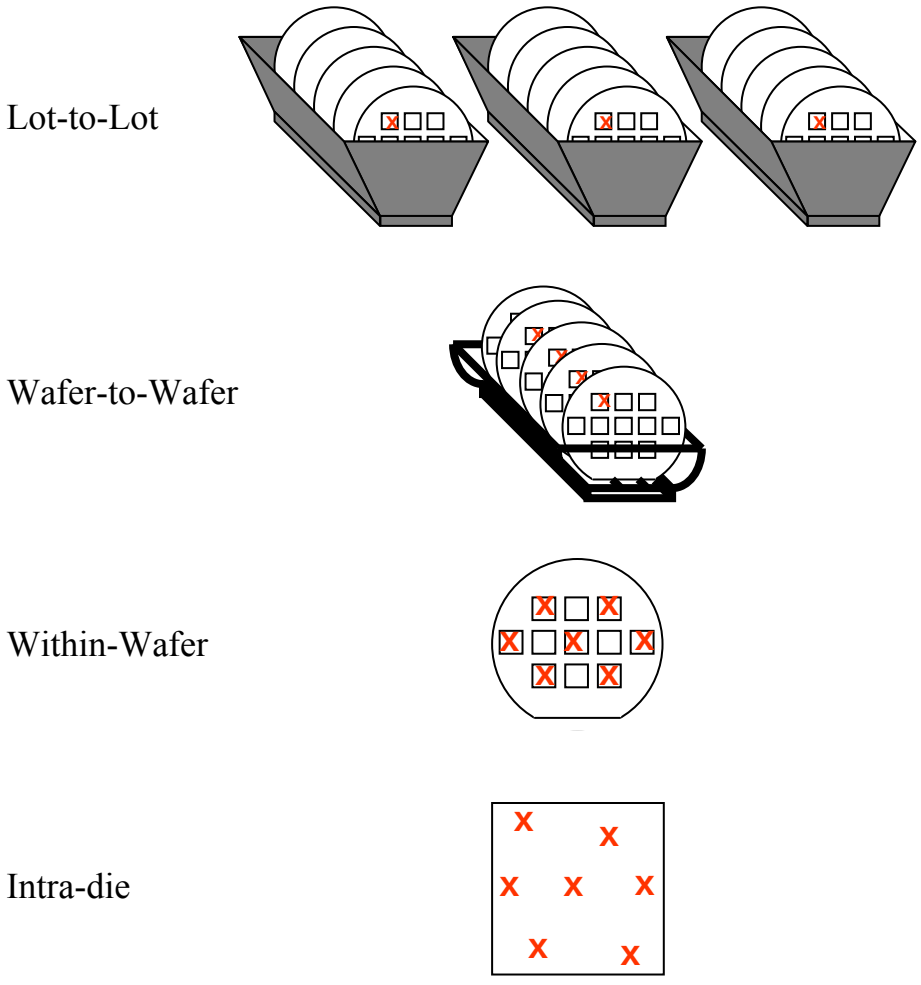


Figure 2.12. Variations on different scales.

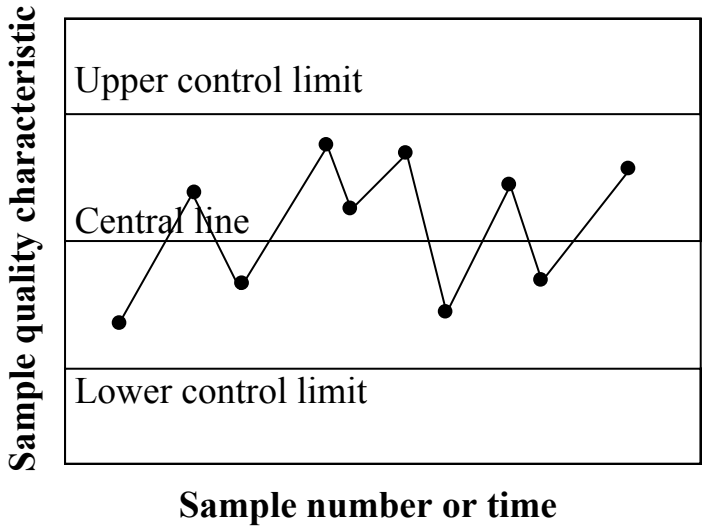


Figure 2.13. A typical control chart.

Within-wafer and intra-die variations are related in complicated ways. Thus the within wafer variation cannot be obtained by simply collecting one data point from each die. Within-wafer and intra-die variations should be carefully decomposed. A decomposition method of the within-wafer and intra-die variations is suggested in [70].

In that work, several methods are evaluated to estimate the within-wafer variation, which include the down-sampled moving average estimator (DSMA), the meshed spline method (MSM), the linear regression coupled with a physically based cross validation approach, or a linear combination of these estimators. Die-level variation is extracted by the FFT-based method using the raw data minus the within-wafer variation. After subtracting the within-wafer and die-level components from the raw data, a simple spline-based method or an FFT-based method are used to estimate the wafer-die interactions. Finally, the residuals are left over. The decomposition flow is shown in Fig. 2.14.

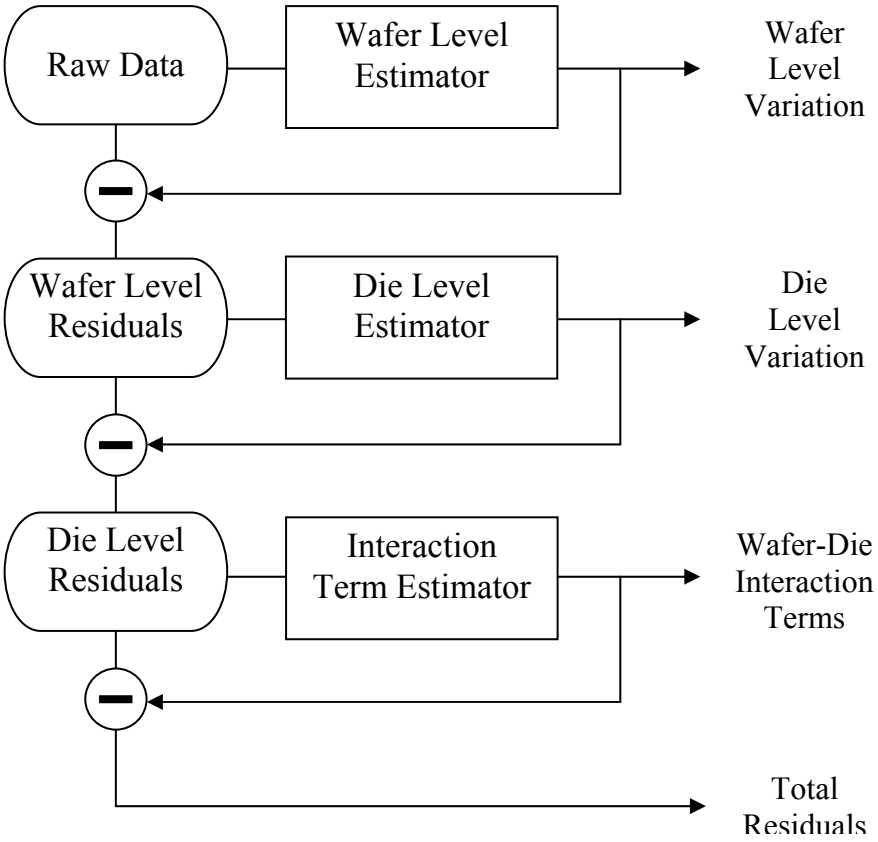


Figure 2.14. Variation decomposition flow.

2.5.1 Gate Poly CD Data Collection Methods and Test Structures

The gate poly CD is usually measured by scanning electron microscopy (SEM). However, many measurements are required to obtain the within-die variation. In addition, the SEM has issues with sample charging and differences between top and down SEM measurements, as shown in Figure 2.15 [4]. Therefore, several electrical measurements have been proposed. First, the CD is measured by the poly gate resistance, but this method averages top and down CDs and includes the variation of the poly doping profile. Recently, to collect the within-die variation efficiently, a memory array structure is suggested, as shown in Figure 2.16. The modified SRAM array is used in [71], [72],

where the drain current of each nMOSFET and pMOSFET in an SRAM cell is measured and the gate CD is extracted. Masuda *et al.* [73], [74] designed the measurement array unit (MAU), including nMOSFETs, pMOSFETs, ring oscillators, and interconnect parasitics. This approach also measures the drain current of the transistors and the voltage drop on interconnect test patterns. From these measurements, gate CDs and interconnect geometries are extracted. These active electrical metrologies include gate oxide thickness and channel doping profile variations.

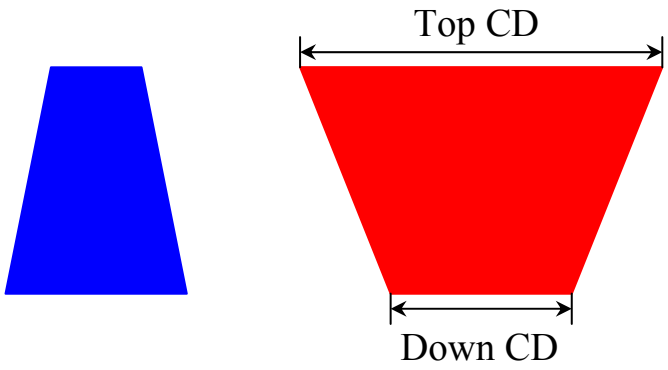


Figure 2.15. Two profiles have the same down CD, but the measured CD of the right profile is larger when using a SEM.

The optical proximity effect and lens aberrations interact, so the gate CD for each proximity category varies differently as a function of lens aberrations, as demonstrated in [75]. Thus, Orshansky *et al.* [75] suggested that the test pattern for each category should be placed in every region of the chip (5 x 5 regions), and CDs for all categories were measured as a function of the location in the reticle, as shown in Figure 2.17.

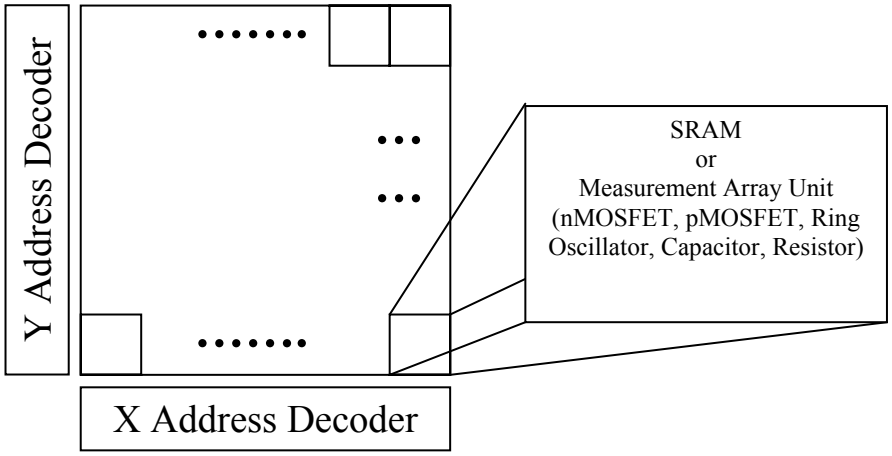


Figure 2.16. Active electrical metrology.

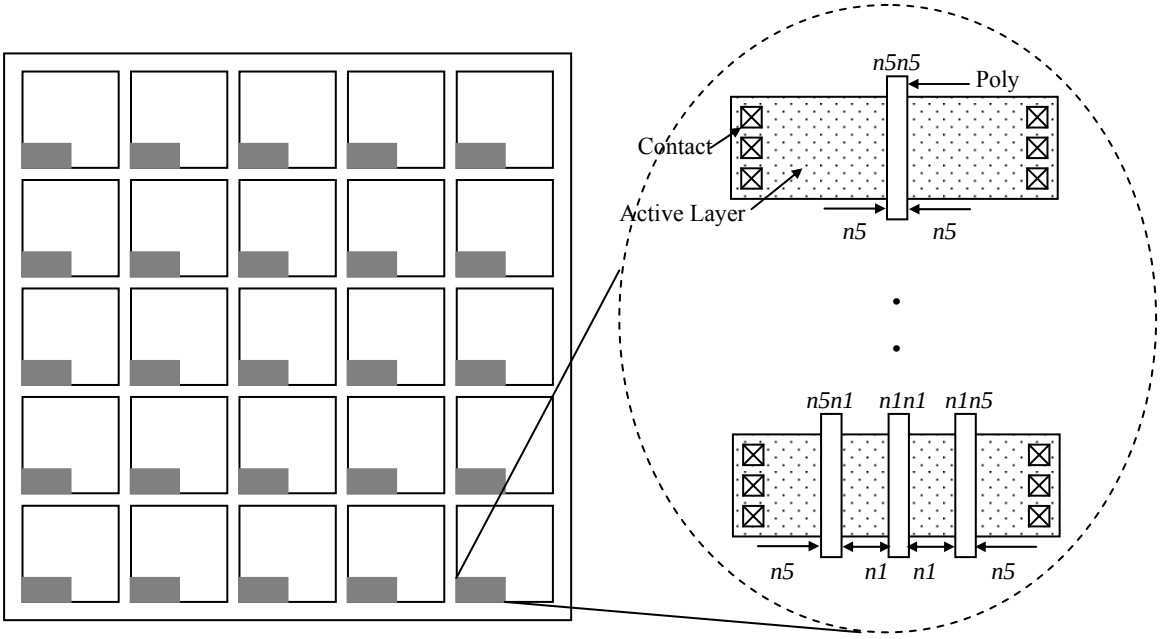


Figure 2.17. Test patterns for proximity categories are located in the different regions of the reticle.

Flare in optical lithography is affected by the pattern density in the neighborhood. So a test structure with the different pattern densities is proposed in [64]. Test patterns are located in regions with different pattern densities, as illustrated in Figure 2.18. To consider the effect of flare on circuit performance, the pattern density is calculated for a fixed unit area. The value for unit area is not known *a priori.* It must be calculated to determine the distance of interaction between mask patterns and variations in printed geometries. This distance is measured with a test structure shown in Figure 2.19.

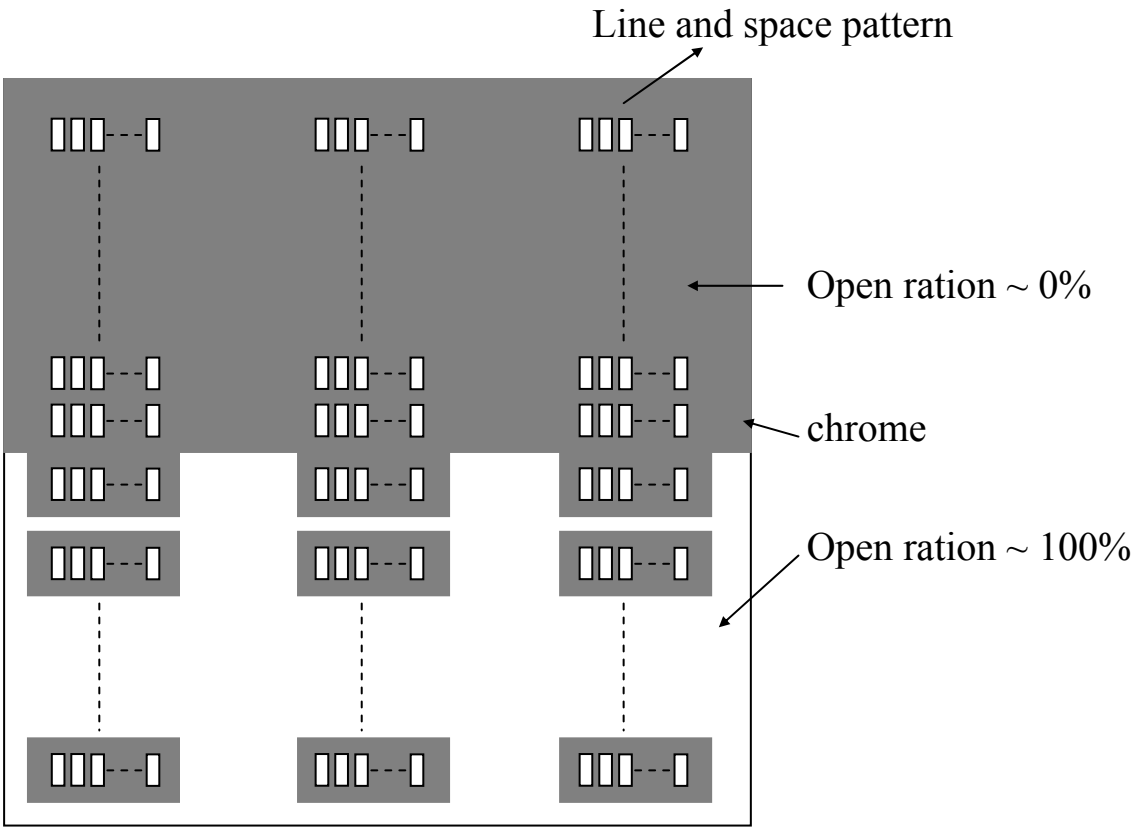


Figure 2.18. Test patterns for flare effect.

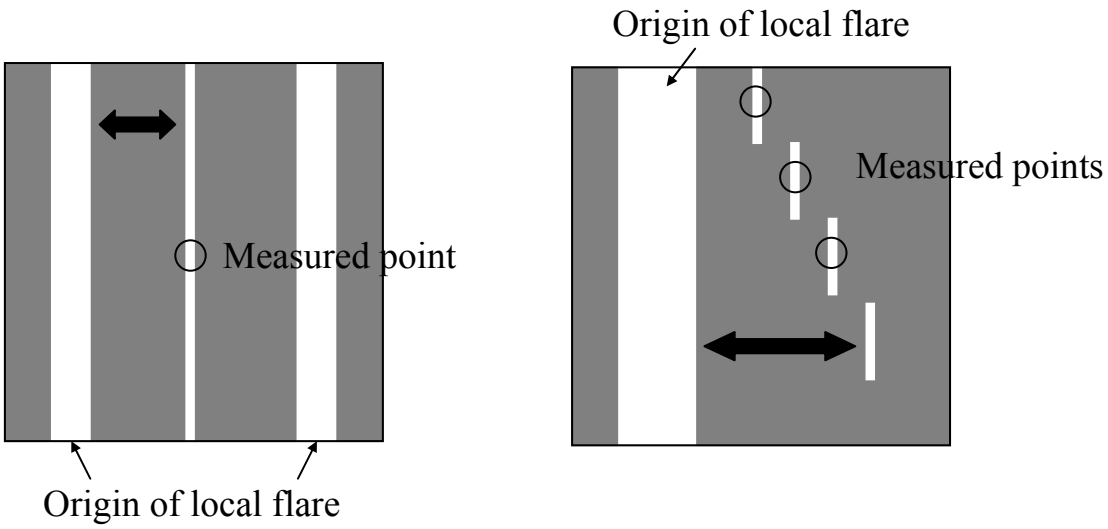


Figure 2.19. The range of the flare effect.

2.5.2 Metal Line Thickness Measurements and Test Patterns

The thickness of copper metal lines can be measured by scanning the surface with a profilometry scan. The electrical measurement of the copper thickness is also desirable for efficient data collection and for reasons similar to the need for gate CD measurements. The metal thickness is extracted from the resistance of metal lines. Figure 2.20 shows a test structure where the isolated line is used for a copper dishing measurement, one of the sources of variation in chemical mechanical polishing (CMP). The array region in Figure 2.20 is used to measure oxide erosion. Park *et al.* [67] proposed test masks to collect the thickness variation as a function of the density, area, and pitch, as shown Figure 2.20. A test pattern for the multilevel dependency is shown in Figure 2.21. In [68], the under-layer effect is measured by varying the pattern density in the under-lying layer.

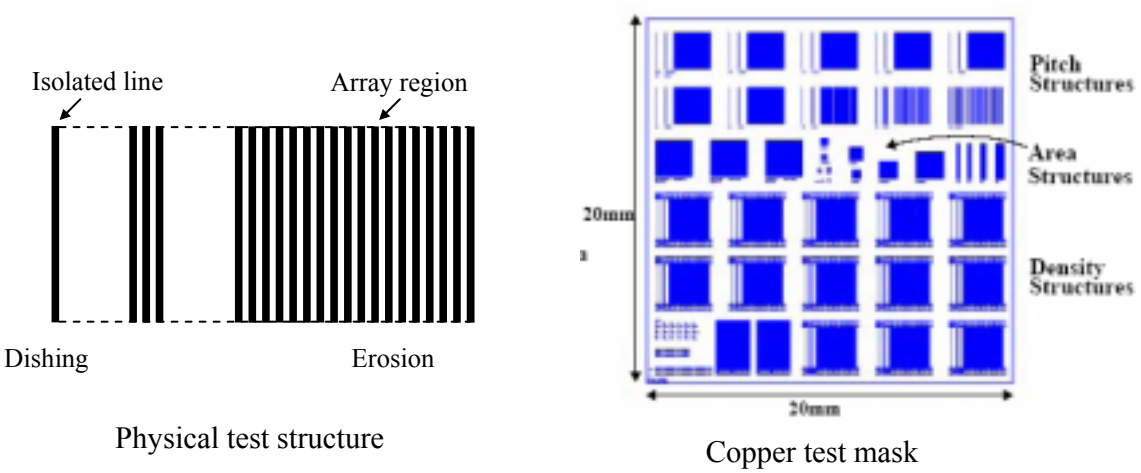


Figure 2.20. Test structure for CMP variations (Dishing and erosion).

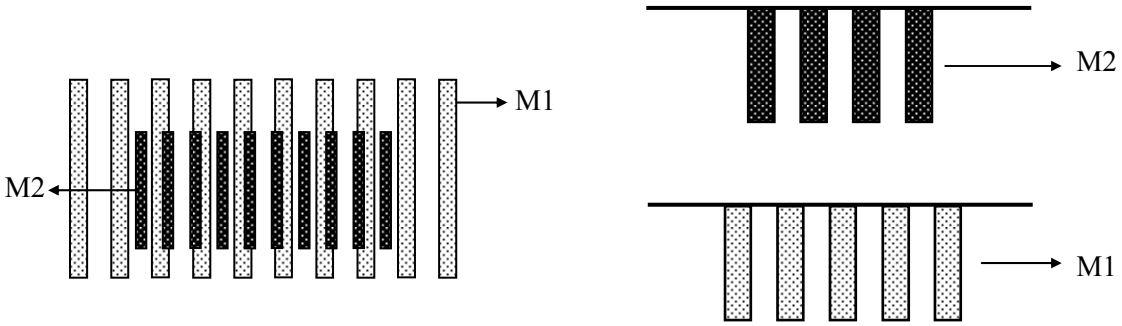


Figure 2.21. Multilevel dependency test patterns.

CHAPTER 3

LAYOUT-DEPENDENT TIMING ANALYSIS METHODOLOGY

In this chapter, it will be explained how within-die variations, described in the previous chapter, are modeled in the timing simulation flow. Also, the computational cost of the methodology will be discussed.

3.1 Timing Simulation Flow

The inputs to the methodology include the layout, the gate cell netlist of the circuit, and the list of critical paths and near critical paths. The set of critical paths depends on processing conditions and will change as a function of the sources of within-die variations described in this thesis. Therefore, it is important to consider a large set of potential critical paths.

A critical path is composed of gate cells and interconnect, as shown in Figure 3.1(a). The conventional timing analysis tool has three classes: Class GATE_TABLE, GATE_INSTANCE, and NODE_INSTANCE. Class GATE_TABLE provides the technology information, such as input pin capacitance, delay and leakage tables, of each gate cell. Class GATE_INSTANCE describes the topology of the circuit (connections of instances of the gate cells). Class NODE_INSTANCE has the parasitic information of the metal interconnect. Objects of class GATE_TABLE are created by reading the technology library, and objects of class GATE_INSTANCE are generated from the hardware description language (HDL) file, which is verilogHDL in this thesis. For example, if a circuit has 2000 gates, which are classified into 30 types, then 2000 objects of class GATE_INSTANCE and 30 objects of class GATE_TABLE are required for

timing analysis. The parasitic information is back-annotated to objects of class NODE_INSTANCE from the layout. The components that make up conventional timing analysis are shown in Figure 3.2.

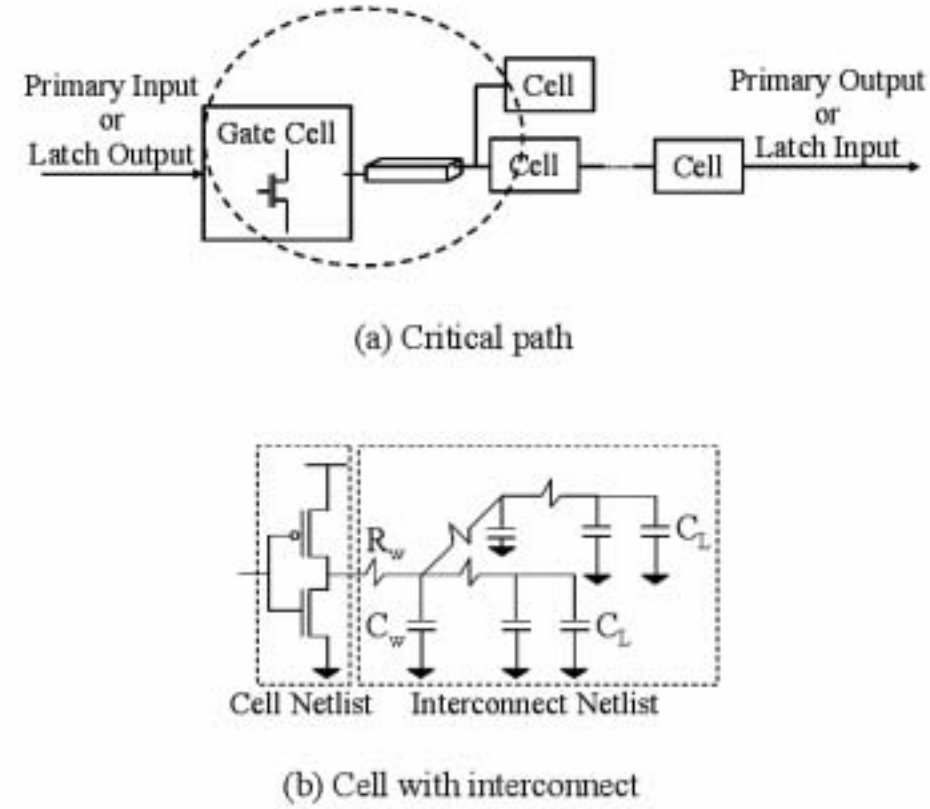


Figure 3.1. Critical path structure for timing analysis. Critical paths contain cells and interconnect, as shown. They are modeled by netlists, which are partitioned into cell and interconnect components.

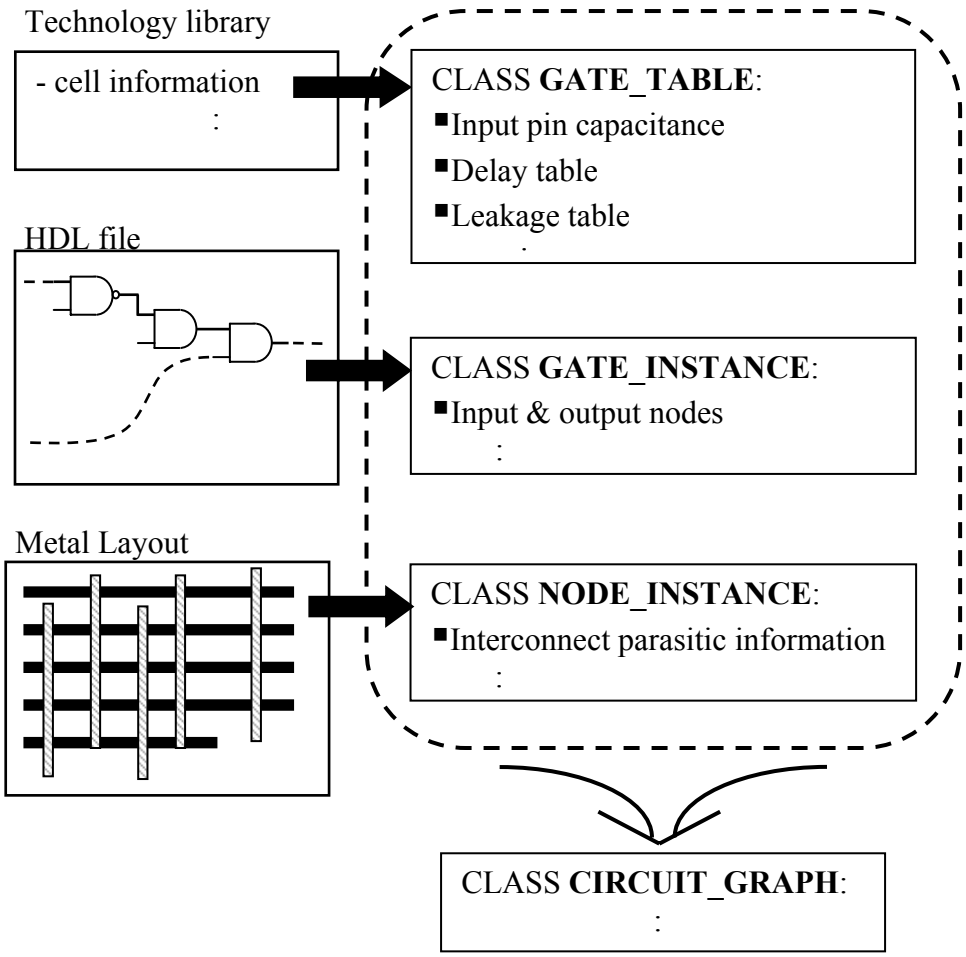


Figure 3.2. Inside the conventional timing analysis flow.

Within die variations caused by lithography modify the gate CDs and the interconnect linewidth based on neighboring patterns in the layout (proximity effect), the location in the layout (lens aberrations), and the density of features on the mask (flare). Additionally, CMP variation changes the interconnect thickness as a function of the line width and space, and the pattern density.

In order to account for all of these factors, the layout-dependent timing analysis flow is established. The layout-dependent timing analysis flow adds an array GATE_Tr

in class GATE_TABLE, pattern density tables in class CIRCUIT_GRAPH, and new variables to contain the location of the gate in class GATE_INSTANCE, as shown in Figure 3.3. Class GATE_Tr contains the proximity effect information, gate length, gate width, and transistor pin connections in the gate cell, as shown in Figure 3.4.

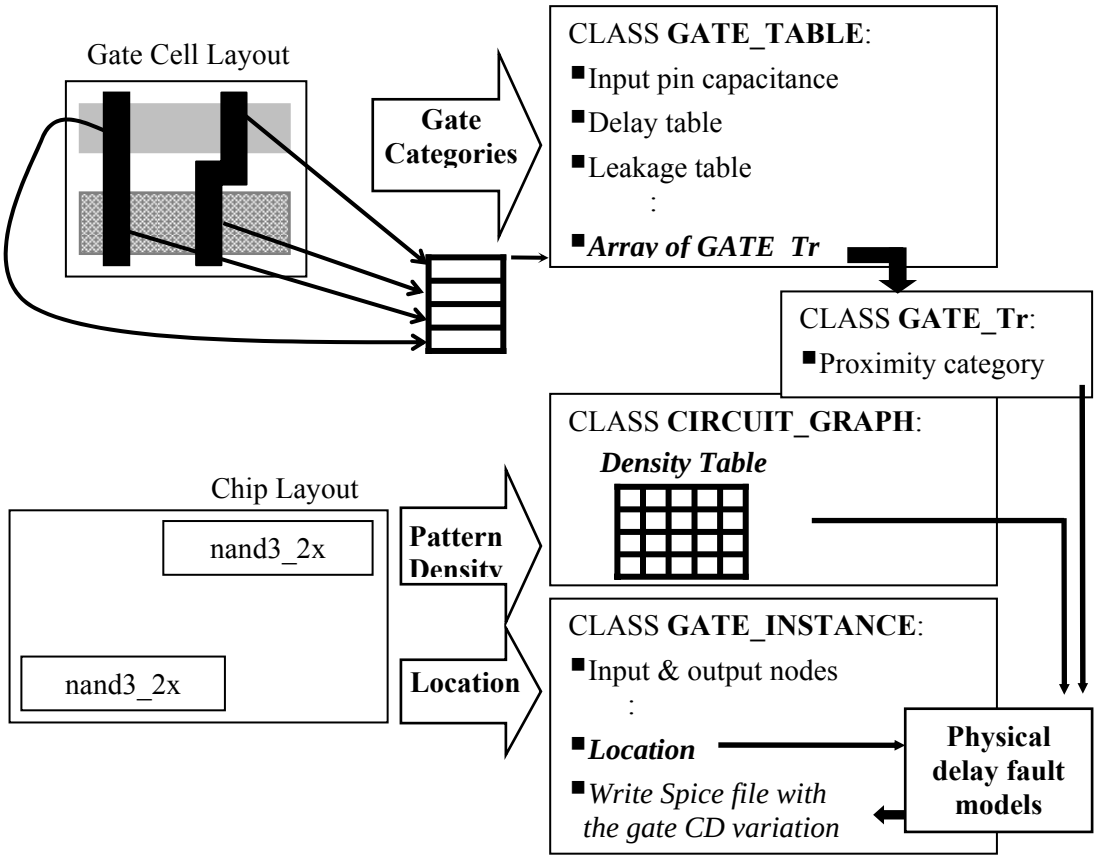


Figure 3.3. Inside layout-dependent timing analysis flow.

The goal of the layout-dependent timing analysis is updated critical path delays, which involves updating delays of cell instances. The delays of cell instances are a function of the CDs of poly gates within the instances, which in turn depend on layout features. Therefore layout data is extracted and fed into the timing analyzer, together with data on variations as a function of layout features (proximity effect, Coma, lens

aberrations, flare). Based on this information, it is then straightforward to generate a new gate cell netlist just by writing the modified gate length and the other variables in GATE_Tr into class GATE_TABLE. The link between detailed transistor data in GATE_Tr and physical cell characteristics in GATE_TABLE requires delay re-characterization of the gate cell. This can be determined through various methods, including using Hspice simulation, analytic gate cell delay models [76], and efficient dynamic simulation [77]. This work has used *Avant! Hspice* [17].

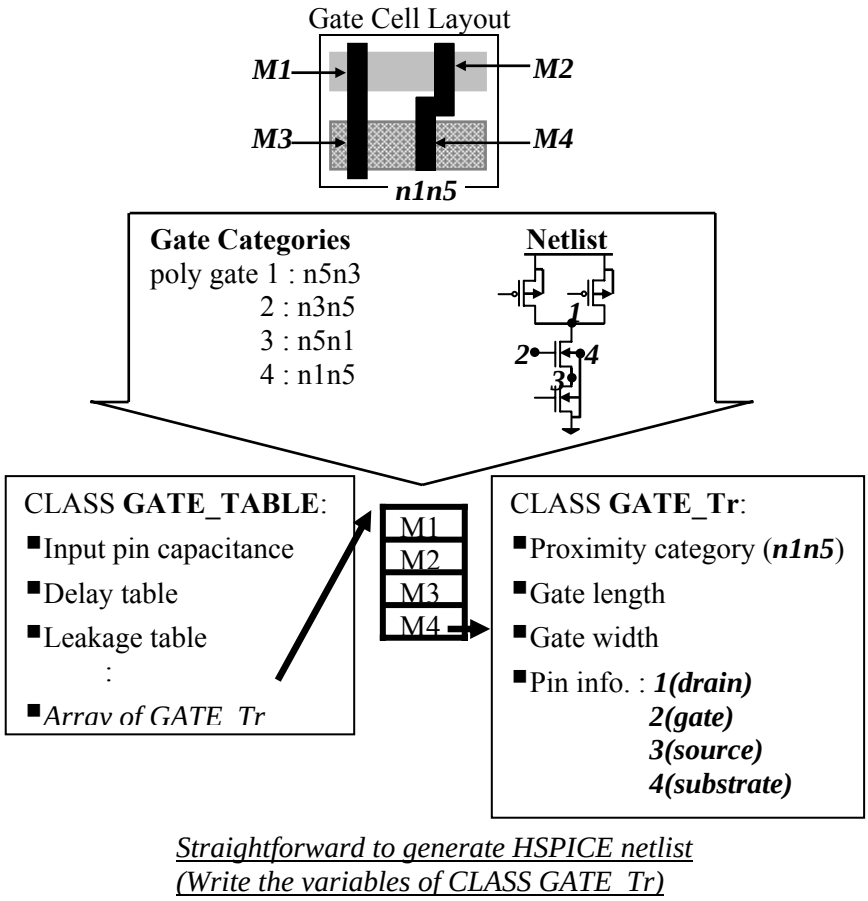


Figure 3.4. Generation of the modified gate cell netlist, which includes neighborhood information for each transistor.

In this thesis, interconnect variation is considered, too. The interconnect netlist, connected to the output of the cell as shown in Figure 3.1(b), has parasitic resistance and capacitance which are affected by the same sources of systematic variation as transistors. The parasitic extraction flow and the approach to updating the values of parasitics will be discussed in Section 3.2.

After merging the interconnect netlist into the cell netlist, the delays of the cells are calculated. However, it should be noted that all points aren't needed in the delay table, because this thesis targets analysis, not synthesis. Thus each critical path or near critical path can be processed sequentially. When processing each critical path, each cell on the critical path is simulated sequentially using *Avant! Hspice* [17]. Because the delays are a function of the transition time of input signals and the loading capacitance [78], the transition time is handed over to the next cell in the path. This process continues until the delay of the last cell on the critical path is calculated. The delays are then summed up to determine the delay of the critical path. In this way, path delay calculations are performed for all paths in the near-critical path set of the chip. The flow chart of the algorithm is shown in Figure 3.5.

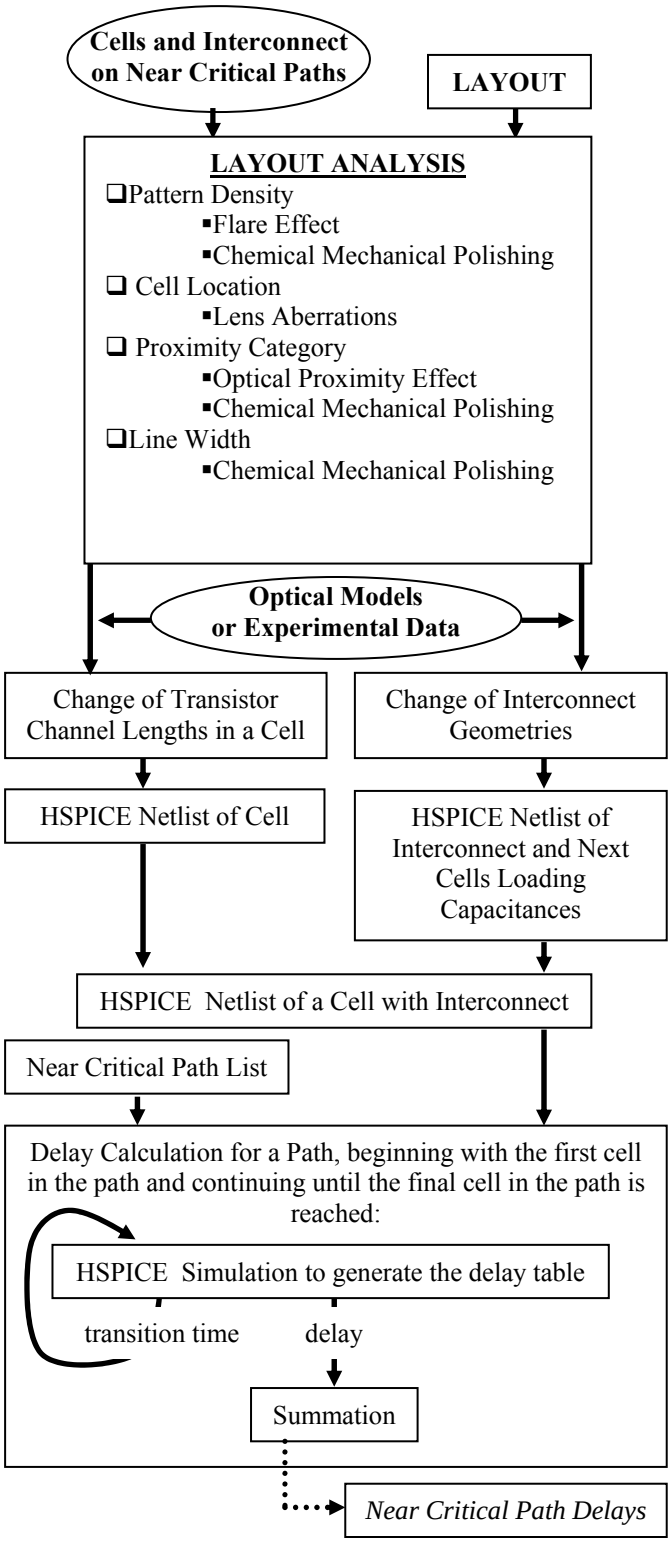


Figure 3.5. Flow chart for updating the delay of critical paths.

3.2 Interconnect RC Extraction

Interconnect parasitic extraction is needed to incorporate interconnect variation in the flow. The conventional extraction methodology of interconnect parasitics in final timing verification is to precharacterize the representative interconnect structures, store them in the pattern library, match interconnect structures to entries in the library, and extract interconnect capacitance [79]. However, this methodology requires significant simulation during pre-characterization by a three-dimensional field solver. Thus this methodology is not compatible with static timing analysis.

Cong *et al.* [80] proposed a 2 ½-D capacitance extraction methodology, which is simple but accurate. When Cong's methodology is combined with analytical capacitance models, it becomes more efficient. Also, the information gathered for capacitance extraction is directly used when considering systematic within-die variation. Therefore, capacitance can be updated to include systematic variation and changed during extraction on the basis of Cong's methodology [80] and Wong's analytical capacitance models [81].

According to Cong's five foundations, when the object metal is located in i-th layer, the (i+2)-th and (i-2)-th layers are assumed to be ground planes, and only the electric fields to the closest neighboring metals in i-th layer and the overlapped and underlapped metals in the (i+1)-th and (i-1)-th layers are considered. The interconnect structure is represented as in Figure 3.6. The top two metal layers are modeled as in Figure 3.6(a) and the other layers are modeled as in Figure 3.6(b). The area-fringe and coupling capacitances are shown in Figures 3.6(a) and 3.6(b), and the crossover capacitance is shown in Figure 3.6(c). Sim *et al.* [82] proposed the 3-D fringing component and the concept of the effective width, but the methodology has not included these factors for simplicity.

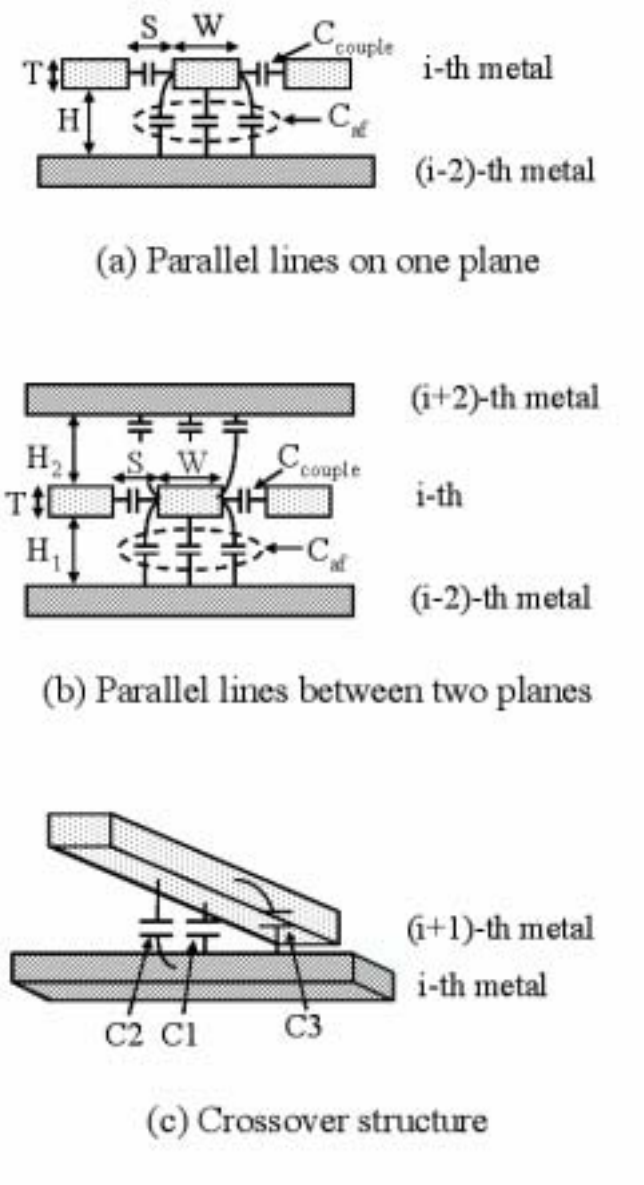


Figure 3.6. Interconnect structures for the capacitance calculation. Extracted capacitances include area capacitances, Caf, to the $(i+2)^{nd}$ and $(i-2)^{nd}$ layers, coupling capacitance, Ccouple, to other features in the i^{th} layer, and overlap capacitances, C1, C2, and C3 to the $(i+1)^{st}$ and $(i-1)^{st}$ layers.

Clearly, interconnect parasitic extraction based on analytical models inherently includes location and neighborhood information. Therefore, the only additional data needed for the analysis of systematic variation is the pattern density, which is calculated in the same way as transistor pattern density.

Accounting for variation due to density requires updating the linewidths of all interconnect lines, based on the pattern density of its sector. Analysis of the neighborhood requires more detailed analysis. Specifically, the analysis of the impact of imperfections in lithography on interconnect geometry is achieved by partitioning the metal line based on the neighboring metal patterns as shown in Figure 3.7. As shown in the figure, a single segment may be partitioned into multiple segments as a function of the neighboring geometries. Each of these segments may have different CDs after updating.

It should be noted that only the CDs of the interconnect networks in the critical or near critical paths are analyzed, even though the CD variation of the neighboring metal segments belonging to different networks influences the coupling capacitance. Thus it is assumed that the neighboring metal CD will match that of the interconnect segment being analyzed in order to save analysis time.

The extraction flow is summarized in Figure 3.8. It involves updating the interconnect HSPICE netlists based on location, neighborhood, and density information.

Inductive effects are increasing [83]. The inductance of a line may be insensitive to geometric variation due to the loop characteristics, but the inductive effects change the operating point (typical delay). Thus, inductive effects change the delay sensitivity to the gate length and the interconnect geometries. One method to include inductive effects is

the partial electrical equivalent circuit (PEEC), which is an accurate way to determine interconnect inductance [84]. However, its computational cost is not compatible with circuit synthesis. Nevertheless, efficient algorithms continue to be reported [84], including the quasi-TEM mode approximation [82] and the effective loop inductance [85]. However, these approaches are not mature. Instead, designers focus on methods to suppress inductive effects [83]. As a result, this thesis currently does not consider the inductance of metal lines, although inductance may be included in the future.

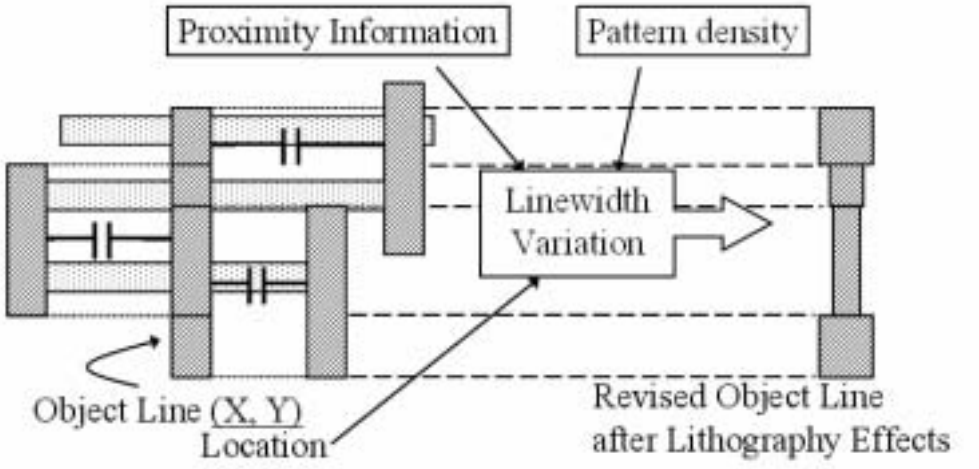


Figure 3.7. Metal linewidth variation from the proximity effect is modeled by partitioning a line into segments according to distances to the next feature on the right and left. Based on the distances, the location in the reticle, and pattern density, the line segments are resized, as shown.

```
Collect the information on all interconnect networks in critical
paths from the routing report.

For all interconnect networks in critical paths {
   Fracture the network into rectangles.
   Construct a tree structure where each rectangle
    becomes an edge.
      A tree keeps all nodes.
      Each node has a list of edges.
      Each edge has source and destination nodes.
      A source node of a tree is the output of the cell.
      Leaf nodes of a tree are the inputs (gates) of the
      next cells and the outputs (sources or drains) of
      other cells.

   For each edge {
      Gather the information about the adjacent metals
      and the overlap and underlap metals.
      Tag segments that are in the same network.
      (Capacitance between two metals in the same
       network is ignored.)
      Divide the edge into sub-edges depending on the
      adjacent metals.

      For each sub-edge {
         Change the linewidth and thickness
         according to data on the proximity effect,
          lens aberrations, flare, CMP, etc.
         Calculate the area, coupling, and overlap capacitances.
         Calculate the resistance.
         Add sub-edge capacitances and resistances to
         the capacitance and resistance of the edge.
      }
   }
   Build the RC network and write the interconnect RC
   HSPICE netlist.
}
```

Figure 3.8. Interconnect RC extraction flow.

3.3 Computational Cost

The analysis of the within-die variation will be performed after physical synthesis. The sources of computational cost of the methodology are summarized in Table I.

Table 3.1: Computational challenges.

Item	Bottleneck	Cost
Pattern density	- Management of a large GDS file	O(N)
Location & Neighboring cells	- Time to search for cells - Management of a large GDS file	O(Ncp·N)
Proximity categorization	- Enlargement of the cell library due to different	O(Ncl)
Interconnect RC extraction	- Time to search for neighboring patterns and over & under-lapping patterns - Management of a large GDS file	O(Ncp·N)
Cell HSPICE simulation	- Increase in near-critical paths	O(Ncp)

N: Total number of transistors in a chip, Ncp: Total number of cells in near-critical paths, Ncl: Number of cells in the cell library

According to the procedure in Figure 2.9, the computational cost of obtaining the pattern density map is proportional to the GDS file size. (It is assumed that the size of the GDS file is proportional to the total number of transistors in a chip (N).)

The algorithm also requires a search to determine neighboring patterns for all gates in critical paths. Some of these adjacent gates will be in neighboring cells. Therefore, it is not only necessary to search for the locations of cells on the critical paths, but it is also necessary to search for the adjacent cells. The cost of looking for one cell is

O(N), assuming that total number of cells in a chip is proportional to the total number of transistors. The cost is multiplied by total number of cells in near-critical paths (Ncp).

The computational cost of categorization for the proximity effect is proportional to the size of the cell library (Ncl), since each cell GDS file is treated independently. Treatment of cell GDS files is straightforward, and the number of cells in the cell library does not increase rapidly as a function of technology generation.

Interconnect RC extraction is major bottleneck. However, parasitic analysis is always performed at the stage of physical synthesis. The extracted interconnect RC netlist can be easily saved during conventional parasitic analysis, together with the data needed for the analysis.

Cell HSPICE simulation time is proportional to the number of cells in near-critical paths.

Clearly, the overall computational complexity is O(Ncp·N), where Ncp depends on the chip architecture, not the chip size. This is the major source of computational cost since modern architectures rely heavily on pipelining which requires that the path delays of each stage are equal. For example, the misprediction branch pipeline of the Pentium 4 has 20 stages while the Pentium 3 has 10 [86]. This results in a significant increase in the number of near-critical paths. Therefore, it should be noted that for such circuits the number of near-critical paths will be very large. In this case, it would be better to treat all cells in the layout, in which case, the computational complexity will be O(N·N), rather than O(Ncp·N).

CHAPTER 4

LEAKAGE CURRENT ESTIMATION

Leakage currents of MOS transistors in Figure 4.1 consist of the subthreshold leakage, source/drain junction leakage, gate oxide tunneling current, gate-induced drain leakage, etc. [87]. The subthreshold current of the gate cell is determined by the off-transistors in the off-state pull-down or pull-up networks and the gate tunneling current in the on-transistors [48]. The subthreshold leakage is exponentially affected by gate length variation through DIBL (Drain Induced Barrier Lowering) [87], [88]. On the other hand, the gate tunneling current is linearly affected by gate length variation and exponentially affected by oxide thickness variation because of electron tunneling between the gate and the channel [48], [89]. Research on high-k gate dielectric materials is being actively pursued. In high-k materials the tunneling current is suppressed. Therefore, the subthreshold leakage current is the focus of this thesis.

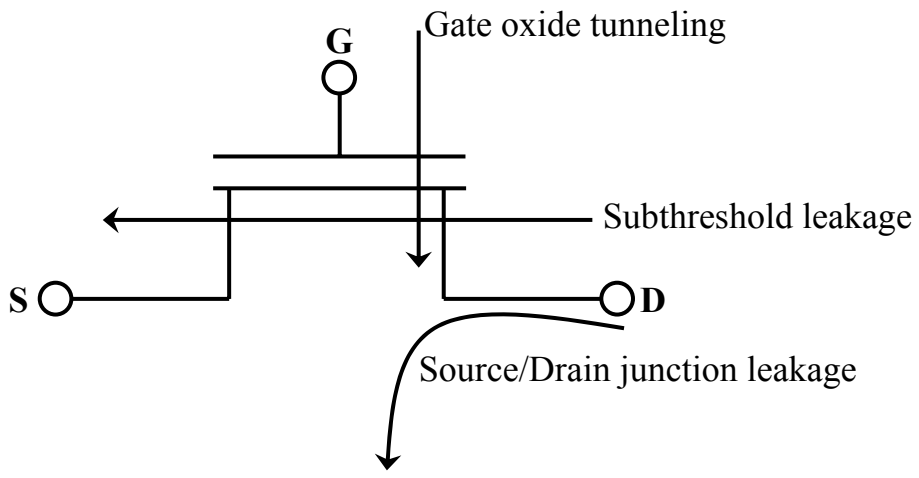
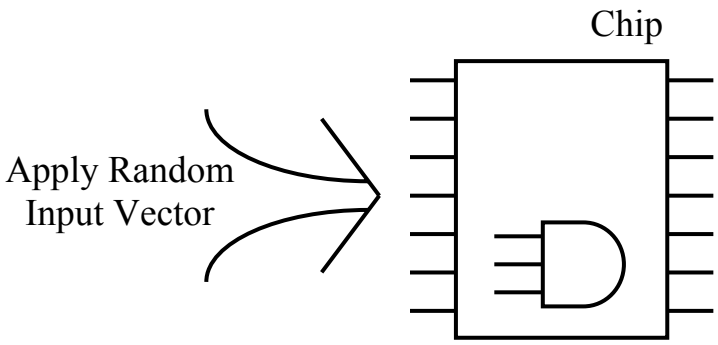


Figure 4.1. MOS transistor leakage current mechanisms.

The leakage current of each gate cell instance varies as a function of the transistor biases, which is determined by the primary inputs (PIs) to the chip. So, chip leakage current is normally estimated by applying random input vectors to PIs, collecting the input vector statistics of each gate cell, characterizing the leakage current of each gate cell as a function of the input vectors, and summing up the leakage current of each gate. The normal chip leakage current simulation flow is illustrated in Figure 4.2. However, this method requires a large amount of effort for gate cell leakage characterization and thousands of logic simulations in order to obtain the input vector statistics of each gate cell. Thus, to simplify the analysis, the subthreshold leakage current of each gate was estimated by averaging the leakage current with all gates with high and low biases, as shown in Figure 4.3. The simple approach reduces the leakage current simulation time of the gate cell instances by over a factor of four.

The simulation flow for chip leakage current estimation, shown in Figure 4.4, is as follows. The poly gate length of the cell was varied accounting for each source of within-die variation from lithography, as in the timing simulation flow using the layout dependent models of Chapter 2. The cell leakage current is obtained by HSPICE simulation when all inputs are low and high, and then the leakage currents of the two cases are averaged. Finally, the summation of leakage currents of all cells in a chip provides the total leakage current of the chip. When the normal method is compared with the simple method, shown in Figure 4.5, the simple method underestimates the leakage current. But, we focus on variation in this thesis. Since the comparison shows a similar trend for each source of within-die variation, the simple method is a reasonable approximation in the following applications.



– Establish the following table for each gate cell instance .

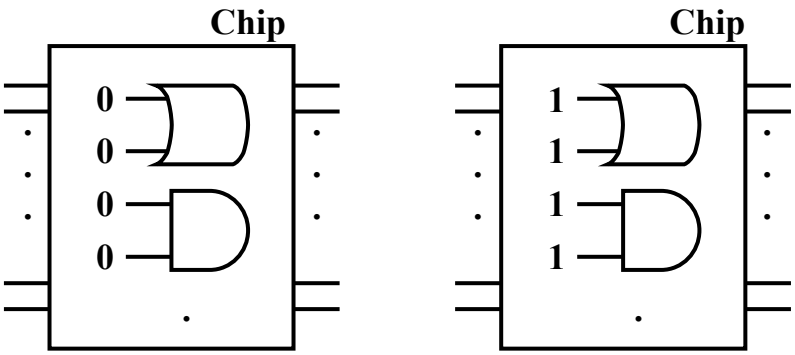
Input pattern	Occurrence [%]	Leakage current [A]
000	O_0	I_0
001	O_1	I_1
.		
.		
111	O_7	I_7

– Multiply the probability of occurrence of each input pattern and the leakage current and sum the results to obtain the leakage current of a gate cell instance.

→ Gate cell instance leakage current = $\sum O_i \cdot I_i$

– Chip leakage current = $\sum I_{instance}$

Figure 4.2. Conventional chip leakage current estimation process.



(a) Gate input = low (b) Gate input = high

Figure 4.3. Gate input states for leakage current estimation. The total leakage current is estimated as the average of the case with inputs set low and the case with inputs set high.

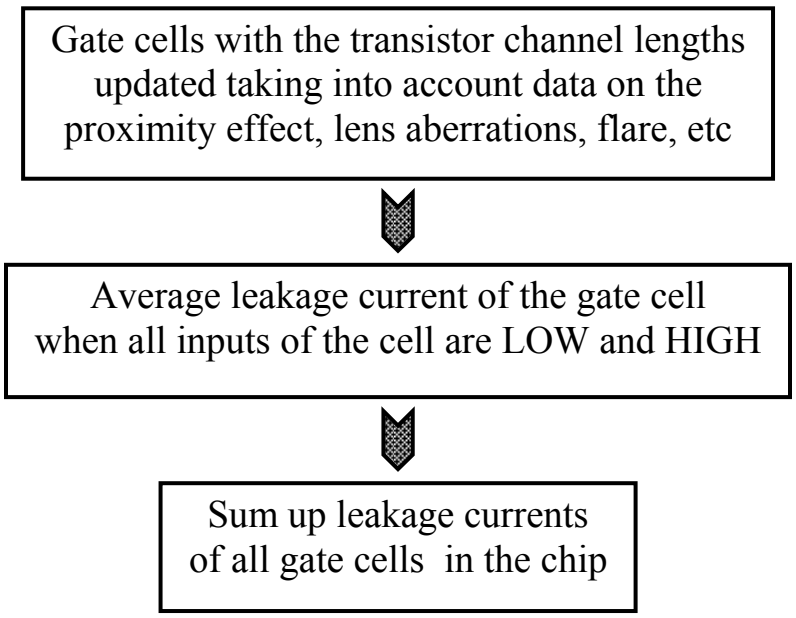


Figure 4.4. Leakage current simulation flow.

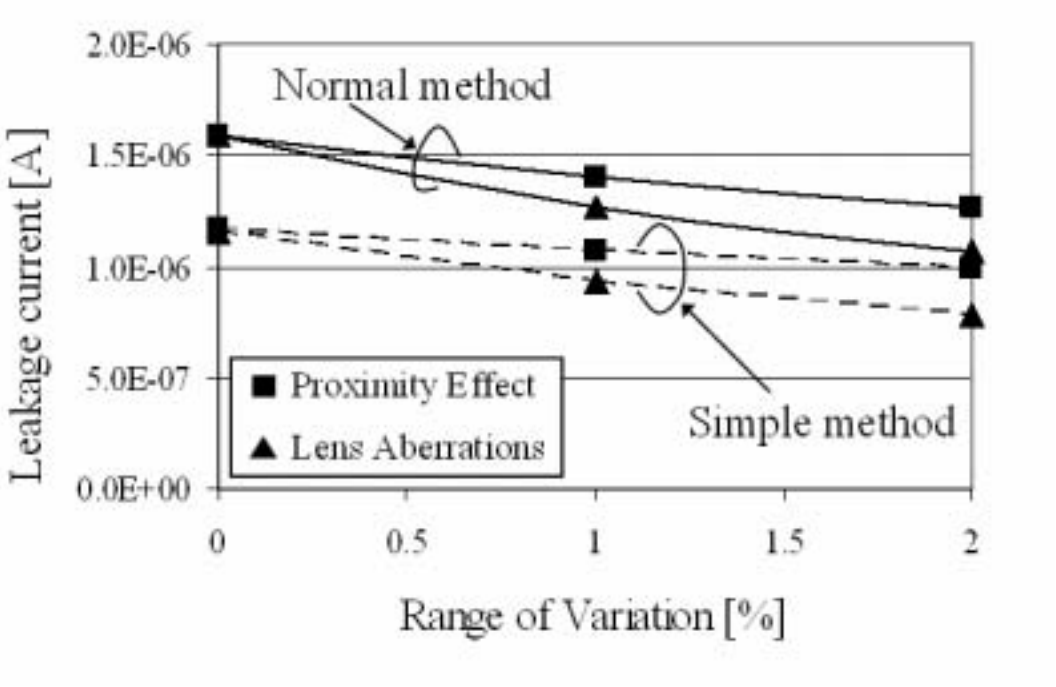


Figure 4.5. Comparison between the normal and simple chip leakage current estimation method. It shows a similar sensitivity to variation for each optical effect.

CHAPTER 5

LITHOGRAPHY IMPACTED DELAY FAULT DIAGNOSIS

The delay fault diagnosis methodology begins by extracting the most significant critical paths through efficient path enumeration of the fault-free circuit. Path enumeration takes into account the transition time dependency and uses a depth first search algorithm, which is improved by pruning the search space. Next, test patterns are generated for the critical paths using a commercial tool, which in this case is *Synopsys Design Compiler* [90]. Using the resulting test patterns, faults are activated, and dynamic timing analysis is performed by applying the extracted test patterns to obtain delays. The delays are compared with a fixed threshold to determine if the test pattern generates a pass or fail. Pass/fail data is collected for all test patterns and all faults to construct the fault dictionary. Observed pass/fail patterns are compared with those in the dictionary through correlation to link the observed signature to a physical mechanism.

5.1 Path Enumeration

One disadvantage of the path delay fault model is that practical circuits have a very large number of paths. One of the ISCAS ’85 benchmark circuits [91], c6288, has 10^{20} paths. Thus we focus on a set of most significant critical paths, under the assumption that other paths are unlikely to affect circuit speed. This may not be the case for delay faults caused by resistive defects, which can cause a large delay in a localized area. It is acceptable for faults caused by within-die variation because such faults cause many small deviations which when combined together result in faults. Thus we reduce the computational effort required to handle a huge number of paths by restricting paths to ones with delays over a specified threshold.

Several papers [28]-[31] provide algorithms to enumerate the longest paths. In [30], [31], the K longest paths are enumerated in the circuit while pruning the search space with the maximum possible delay. In [28], [29], on the other hand, the longest paths are selected such that they cover each gate in the circuit. In these approaches the search space is pruned by the maximum delay constraint, and the paths are checked to ensure sensitizability in order to eliminate any false paths. In all of these approaches the maximum delay to the sink is used for pruning in order to enhance efficiency. However, if the delay constraint is, say, 90% of the maximum circuit delay, these algorithms may have problems with memory management, since the number of paths satisfying this constraint can be extremely large. In [31], in order to limit the number of paths that need to be stored, the algorithm aims to extract the K longest paths. To this end, each node in the circuit graph is associated with a K array, storing the K longest paths to that node. This, however, could be a very large array and could still lead to memory problems.

This work improves the depth first search (DFS) algorithm by pruning the search space through backward signal propagation, as used in static timing analysis [92]. As in [92], it includes the signal propagation effect, which is important for timing analysis, as will be explained in Section 5.3.

5.2 Some Definitions

Combinational circuits have a plurality of inputs and outputs. Combinational circuits may be represented as graphs, where gates are associated with edges and interconnect is associated with the nodes. Therefore, the signals flow from the input nodes to the output nodes through the gates (edges). By convention, adding a source node s and a sink node f to the graph makes the handling of the boundary conditions

easier. Therefore, we modify the circuit graph accordingly. In addition, here are useful definitions relating to the circuit graph.

Definition 1: A timing graph is defined as a directed graph having one source and one sink node: $G = \{N, E, n_s, n_f\}$, where $N = \{n_1, n_2, \ldots, n_k\}$ is a set of nodes, $E = \{e_1, e_2, \ldots, e_l\}$ is a set of edges, $n_s \in N$ is a source node, and $n_f \in N$ is a sink node. Each edge $e \in E$ is simply an ordered pair $e = (n_i, n_j)$ of nodes, where $n_i, n_j \in N$.

Definition 2: A path P of a timing graph $G = \{N, E, n_s, n_f\}$ is a sequence of its nodes $P = (n_a, n_b, \ldots, n_z)$ such that each pair of adjacent nodes n_i and n_j has an edge $e_{ij} = (n_i, n_j)$.

Definition 3: The path delay d_P of path P is defined as $\sum_{e_{ij} \in P} d_{ij}(s_i)$, where $d_{ij}(s_i)$ is a delay of an edge e_{ij} on path P with input transition time s_i, and the summation is over all edges belonging to path P. The edge delays are also a function of the loading capacitance, but we have not denoted loading capacitance as an argument since it is fixed for a specific network.

In order to understand the sequence of operations through a path, a graph is divided into layers. The layers begin at the inputs and end at the outputs. Most static timing analysis algorithms progress from the inputs sequentially through the layers of the graph until reaching the outputs. Backward signal propagation, on the other hand, starts at the outputs and progresses through the layers to the inputs.

Definition 4: A node, n_i, is connected to a set of adjacent nodes, n_j, on successive layers by edges, e_{ij}. The set of nodes connected to node, n_i, on successive layers is called $SUCC(i)$.

5.3 Signal Propagation Effect

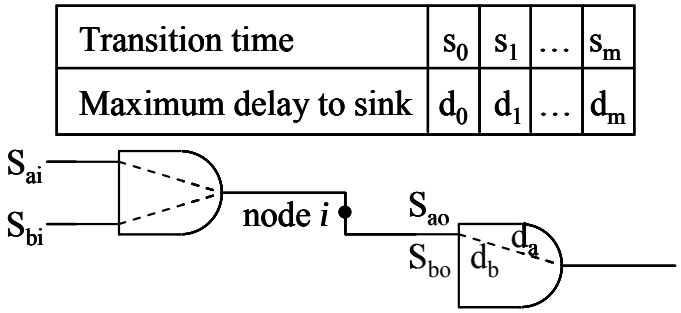
Previous path enumeration algorithms [28]-[31] assume a fixed edge delay (delay between each of the inputs and the output of the gate). The gate cell delay is actually a function of the transition time of the input signal and the loading capacitance. In fact, when two signals arrive at an input, the one that arrives first can determine the longest path if the transition time (slope) is longer. This is why we need to take into account the signal propagation effect.

The loading capacitance is fixed for a specific network. Therefore, the transition time is only dependent on the path being investigated. In fact, if two signals are propagating to a single input, *i,* of a gate, s_{ai} and s_{bi}, the edge delays through the gate, d_{ai} and d_{bi}, respectively, are different as shown in Figure 5.1. The result is distinct delays between node *i* and the sink. To accurately find the critical paths, we then need to take into account all signal transition times within a path. If we do this by brute force, the complexity of the problem of finding the K critical paths becomes exponential.

The estimation of critical path delay in static timing analysis suffers from a similar problem. In [92], this problem is addressed by backward signal propagation to determine the slack at each node. Based on the initial computations of slack, this approach maintains a table of the maximum delay to the sink as a function of the transition time at each node, as shown in Figure 5.1. As illustrated in Figure 5.1, each node in the graph stores a table. This table indicates the delay to the sink as a function of transition time.

The methodology borrows backward signal propagation to overcome the drawback of previous path enumeration approaches. Specifically, we start at the outputs

and progress towards the inputs by creating a table for each preceding node until we reach the inputs. To create the table, we consider a specific transition time at node, n_i. We then look up the resulting delay for the successive edge, e_{ij}. We also look up the delay to the sink for the successive node, n_j, which has been computed previously. We add these delays together, and take the maximum among all successive nodes, to create the entries in the table for the given transition time. This process is repeated for each transition time. Figure 5.2 shows the backward signal propagation algorithm.



– Depending on which signal (a or b) is propagated, edge delay (d_a, d_b) are different. That leads to distinct delays of node *i* to sink.

Figure 5.1. Transition time dependency of the maximum delay from node *i* to the sink (outputs), where s_{ai} and s_{bi} are distinct signals arriving at node *i*.

The table at each node stores a set of discrete values of input transition times and delays to the sink. We have assumed six representative transition times at each node, as in [92], although this choice is arbitrary. Because we obtain the delay by interpolating the delay table, more transition times make the delay estimate more accurate, but more transition times also increase computation time and consume more memory.

Initialize the delay table of the sink node n_f as a function of the transition time.

{Backward propagation}
Visit a node layer in reverse topological order.
For each node *i* in the same order
{
For each edge $e_{ij} = (n_i, n_j)$, where .
{
Propagate the delay table from n_j through e_{ij}
using $D_{ij}(s) = D_j(s_{ij}(s)) + d_e(s)$,
where $s_{ij}(s)$ = transition time at n_j,
$D_j(s_{ij}(s))$ = maximum delay of n_j to sink,
$d_e(s)$ = edge e_{ij} delay.
}
Establish the delay table of n_i
using $D_i(s) = \max(D_{ij}(s), D_{ik}(s), \ldots)$.
}

Figure 5.2. Backward delay_table_to_sink propagation.

5.4 Improved Depth First Search (DFS) Algorithm with Search Space Pruning

As the path enumeration flow has the table of the maximum delay to the outputs as a function of the transition time at each node, the DFS algorithm to determine all critical paths with a delay greater than a threshold can be easily upgraded. While proceeding with DFS, the algorithm starts at the inputs and estimates the maximum delay for all paths emanating from each node. At each node, if the delay is less than the threshold delay, searching for paths emanating from that node is terminated, as shown in Figure 5.3. All complete paths that reach a primary output are saved on the hard disk and are not kept in memory. Figure 5.4 shows the pseudo code of the pruned DFS path

enumeration algorithm.

Once the paths are determined, they are converted to a test set using a commercial ATPG tool, which determines the primary input signals that can sensitize the path and the resulting values at the primary outputs.

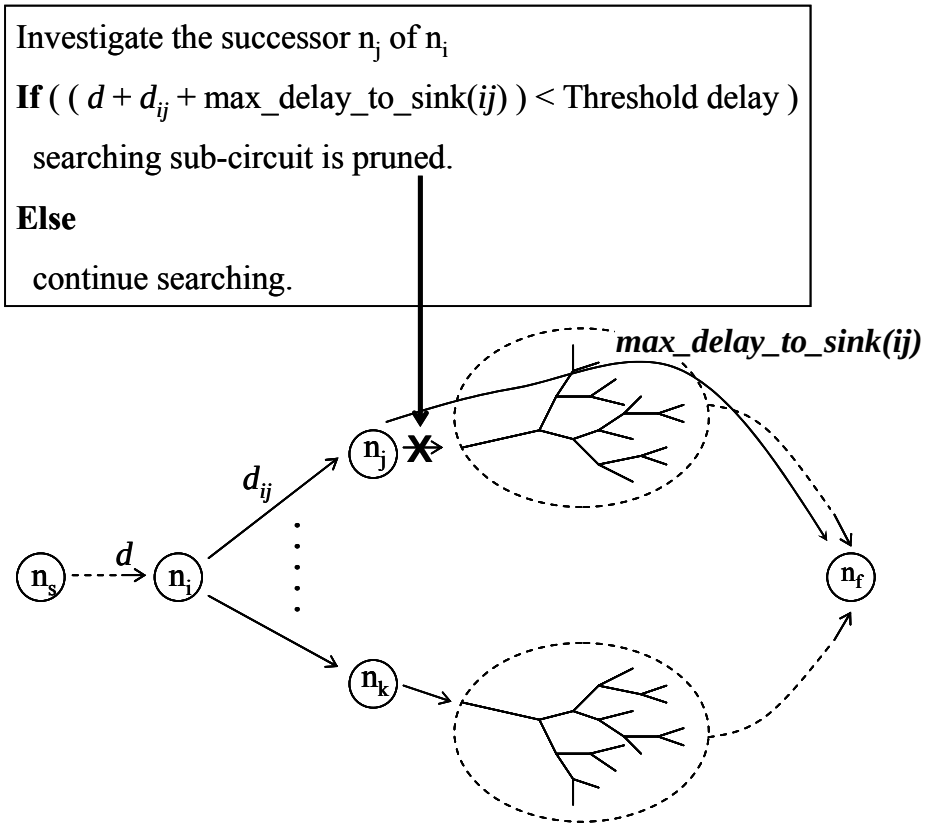


Figure 5.3. Pruning in the DFS path enumeration algorithm. The maximum delay to a sink (output) is stored at each node. If this delay is below a threshold at a specific node, enumeration of paths that involve branches beyond that node is terminated.

```
T = threshold delay, looking for paths over T in G.

{Create the source node s and the sink node f.}
{Compute the maximum delays to sink.}
{Sort the successors of each node.}

{Path enumeraton using pruned DFS algorithm}
 prunedDFS (s);

 prunedDFS (Node i)
 {
  If ( i = sink node )
    save path information;
    return;

  For each  j ∈ SUCC(i)
  {
   If ( ( d + d_ij + max_delay_to_sink(j) ) < T )
     Return;
   Else
     save arrival time ( d + d_ij ) and transition time;
     prunedDFS (j);
  }
 }
```

Figure 5.4. Pruned DFS path enumeration algorithm pseudo code.

5.5 Dynamic Timing Analysis for Fault Simulation

Path enumeration has not taken into account details about the signal train. Therefore, we use dynamic timing analysis to verify temporal correctness of the selected test patterns and to perform fault simulation.

In dynamic timing analysis the signal is propagated through the gates in topological order, just as with static timing analysis. The signal train is input into each

gate. The signal train consists of the arrival time train, the transition time train, and the logic state train, as shown in Figure 5.5.

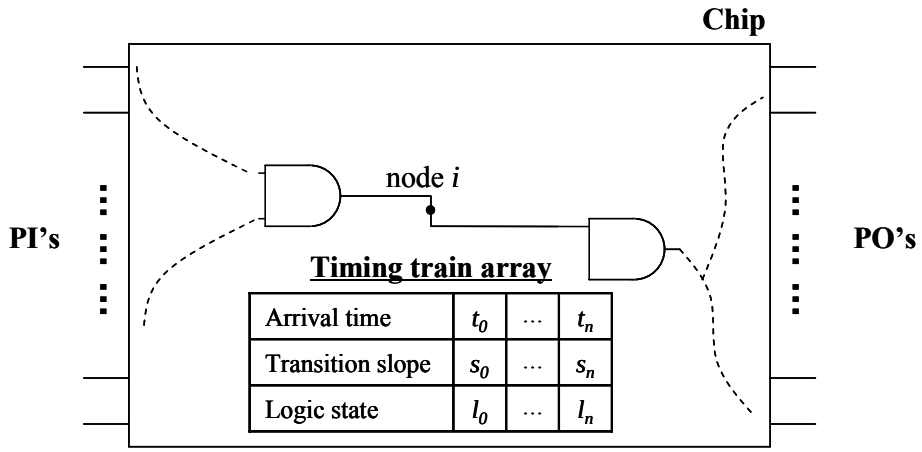


Figure 5.5. A node has signal train for dynamic timing analysis.

The input signals to the gate are propagated to the output, as in Figure 5.6. First, the output arrival time and logic state trains are obtained with zero gate delay. Second, for each transition of the output, it is determined which input changes the output, and the gate delay and transition times caused by each input transition are obtained from the gate cell delay table. Finally, the output arrival time train is updated by adding the delay, and the transition time train at the output is established.

Fault simulation relies on dynamic timing analysis. To simulate a fault, the revised delay tables that reflect faulty behaviors are generated. The dynamic timing analyzer then inputs the revised faulty delay tables and test patterns to determine the delay associated with each test pattern.

The implementation of the dynamic timing analysis does not consider signal coupling between interconnect lines (crosstalk) [93], data dependent delays [94]-[96], and the closeness dependency of input transitions.

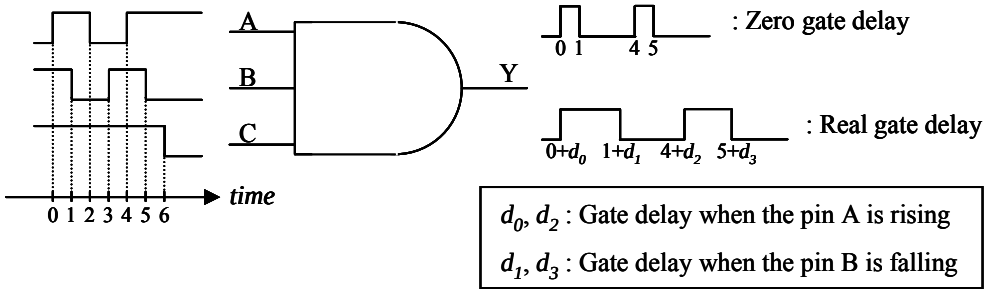


Figure 5.6. Dynamic signal propagation with a signal train.

5.6 Fault Diagnosis

The patterns selected for test application have delays greater than a threshold. For our examples, this threshold was set to 90% of the maximum delay. Therefore, for the 90% threshold, all selected test patterns, *t*, have delay $d_t > 0.9 \cdot d_{\max}$. Then, if the clock frequency is increased to $f = 1/0.9d_{\max}$, all of the selected patterns, with delay greater than 90% of the maximum delay will fail for fault-free circuits. Even in the presence of die-to-die variation, because die-to-die parameter variation is almost 100% correlated, all of the selected patterns will fail. However, if the circuit contains a fault caused by within-die variation, it is possible that some of the patterns will pass. The patterns that pass are associated with specific faults, and each fault is associated with a pass/fail signature for the applied test patterns. Consequently, the test methodology involves

determining the maximum delay, $d_{\max}$, for each circuit instance, applying test patterns at $f = 1/0.9d_{\max}$, and determining the passing patterns, where $d_{\max}$ is a function of process parameters and the fault being considered. Faults are detectable if there is at least one passing pattern. Faults are diagnosed by correlating the pass/fail signatures, i.e. if the pass/fail patterns match those in the dictionary.

CHAPTER 6

APPLICATIONS

Four applications have been considered. The first application is to use the proposed methodology to compare the impact of different sources of degradation in lithography when the leakage current of the chip is fixed. Second, model-based correction versus a simpler correction scheme is analyzed. Third, it is investigated how systematic variations in interconnect affects circuit performance, and the tool is used to analyze the impact of an underlying layer on interconnect thickness variation. Fourth, the effectiveness of the diagnosis methodology is investigated.

6.1 Impact of Optical Effects with Fixed Leakage Current

In this analysis, four sources of systematic variation are considered in lithography separately (the proximity effect, Coma, lens aberrations, and flare).

In each experiment, the minimum gate CD and the range of CD variation (maximum – minimum) were varied. Hence, each experiment had two input variables. The minimum CD was varied by ±10%, where the levels of -10% and +10% were coded as -1 and +1 respectively in all equations, as shown in Table 6.1(a). Similarly, the ranges of the sources of systematic variation in lithography were set to vary from 0 to 10 percent. These variables were also coded as varying from –1 to +1, as shown in Table 6.1(b). For the proximity effect, the range was varied by up to 10%, by causing the CD for dense patterns (n1n1) to be 10% larger than the CD for isolated patterns (n5n5). The sizes of intermediate patterns were interpolated, as shown in Table 6.2. Coma was assumed to cause transistors (n5n1) with dense features on the right and isolated features

on the left to have up to 10% larger CDs than transistors (n1n5) with dense features on the left and isolated features on the right, as shown in Table 6.3. Lens aberrations were assumed to cause transistors on the right side of the chip to have up to 10% larger CDs than transistors on the left side, as shown in Figure 6.1. (A simplified model of lens aberrations is used for demonstration purposes. More realistic patterns can be found in [12].) Finally flare was assumed to cause transistors in dense areas to be up to 10% larger than transistors in isolated areas, as shown in Figure 6.2.

Table 6.1: Coded variation.

(a) Minimum CD

Coded	-1	0	1
Original [%]	-10	0	10

(b) Range of variation

Coded	-1	0	1
Original [%]	0	5	10

Table 6.2: Proximity effect in the example (x max. impact [%]).

Category	**n1**	**n2**	**n3**	**n4**	**n5**
n1	1.000	0.875	0.750	0.625	0.500
n2	0.875	0.750	0.625	0.500	0.375
n3	0.750	0.625	0.500	0.375	0.250
n4	0.625	0.500	0.375	0.250	0.125
n5	0.500	0.375	0.250	0.125	0.000

Table 6.3: Coma effect in the example (x max. impact [%]).

Left \ Right	**n1**	**n2**	**n3**	**n4**	**n5**
n1	0.500	0.375	0.250	0.125	0.000
n2	0.625	0.500	0.375	0.250	0.125
n3	0.750	0.625	0.500	0.375	0.250
n4	0.875	0.750	0.625	0.500	0.375
n5	1.000	0.875	0.750	0.625	0.500

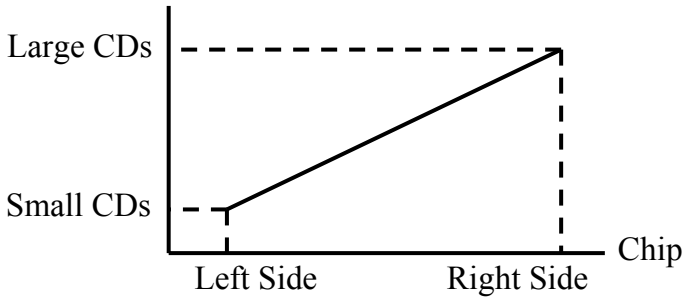


Figure 6.1. Impact on CD of lens aberrations in the example. The CD increases linearly from the left side of the chip to the right side of the chip.

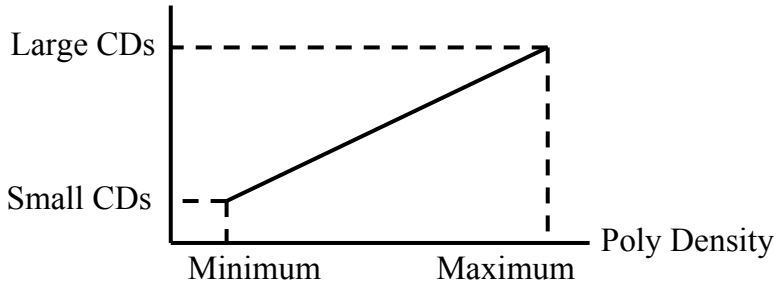


Figure 6.2. Impact on CD of flare in the example.

The methodology was applied to c7552 (an ISCAS '85 benchmark circuit [91]: ALU). Figure 6.3 shows the delay of the critical paths as a function of the minimum CD and the range of CD variation due to each source of within-die variation. As can be seen from the graphs, delay is reduced with smaller minimum CD and perfect lithography (lower left corner). Moreover, compensating for lens aberrations and flare is more effective in improving circuit speed.

A linear model

$$\eta - \eta 0 = \beta 1 \cdot X1 + \beta 2 \cdot (X2+1) \qquad (1)$$

was fit to the simulation results (X1: minimum CD, X2: range of CD variation). $\eta 0$, the intercept on the η (delay) axis, and the least squares values of $\beta 1$ and $\beta 2$ for each source of within-die variation are shown in Table 6.4, while ANOVA tables for the fits are shown in Table 6.5.

Because process engineers attempt to optimize circuit speed with a constraint on yield, in order to properly calibrate the models, models were also constructed for total leakage current. This is because it is believed that leakage is a major cause of yield loss, since transistors with high levels of leakage current do not function properly and circuits with leaky transistors dissipate large amounts of power, violating specifications on power. Therefore, in this example the circuit speed is optimized with a constraint on leakage current. Specifically, it has been assumed that the maximum allowed leakage current is 1uA. (This is a very small module.)

Figure 6.4 shows chip leakage currents as a function of the minimum CD and the range of CD variation. As can be seen from the graphs, leakage is largest with smaller minimum CD and perfect lithography (lower left corner).

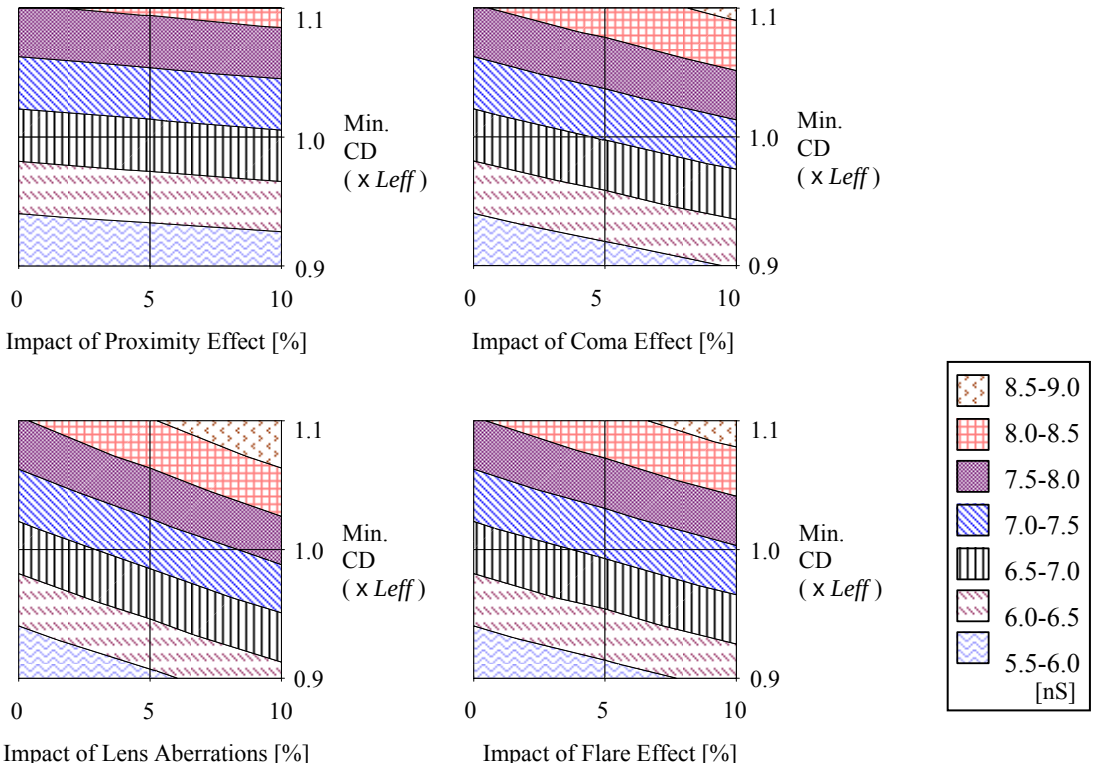


Figure 6.3. Delay as a function of minimum CD and the range of variation for the proximity effect, Coma, lens aberrations, and flare (*Leff* = 0.35μm).

Table 6.4: Least squares values for delay variation.

Coefficients / Litho. impacts	η0	β1	β2
Proximity effect	6.73	1.25 ± 0.010	0.10 ± 0.006
Coma effect	6.73	1.27 ± 0.022	0.30 ± 0.014
Lens aberrations	6.73	1.28 ± 0.035	0.47 ± 0.022
Flare effect	6.73	1.27 ± 0.028	0.37 ± 0.018

95% confidence interval

Table 6.5: ANOVA tables for the circuit delay linear models.

(a) ANOVA table for the delay model impacted by the proximity effect.

Source	Sum of squares	DOF	Mean square	F ratio
Model	9.601397	2	4.800698	48,793.91
Residual	0.000689	7	0.000098	Pr(F) < 0.001
Total	9.602034	9		

(b) ANOVA table for the delay model impacted by the Coma effect.

Source	Sum of squares	DOF	Mean square	F ratio
Model	11.07087	2	5.535435	10,203.39
Residual	0.003798	7	0.000543	Pr(F) < 0.001
Total	11.07495	9		

(c) ANOVA table for the delay model impacted by the lens aberrations.

Source	Sum of squares	DOF	Mean square	F ratio
Model	13.22423	2	6.612116	5,146.00
Residual	0.008994	7	0.001285	Pr(F) < 0.001
Total	13.23332	9		

(d) ANOVA table for the delay model impacted by the flare effect.

Source	Sum of squares	DOF	Mean square	F ratio
Model	11.882	2	5.940997	6,988.55
Residual	0.005951	7	0.00085	Pr(F) < 0.001
Total	11.88752	9		

The leakage current is expected to be an exponential function. Therefore, the logarithm was taken of the leakage current data before fitting the model

$$LN(\eta) = \beta 0 + \beta 1 \cdot X1 + \beta 2 \cdot X2 + \beta 12 \cdot X1 \cdot X2 + \beta 11 \cdot X1 \cdot X1 + \beta 22 \cdot X2 \cdot X2 \quad (2)$$

to the simulation results. The least square values of β0, β1, β2, β12, β11, and β22 are shown in Table 6.6, while ANOVA tables for the fits are shown in Table 6.7.

For this example, with fixed leakage current, Figure 6.5(a) shows the relationship between delay and each source of systematic variation. It is interesting to note that delay is sensitive to lens aberrations and flare after we control for leakage current, but not the other effects. The reason is that the contour line for speed has a steeper slope than that for leakage current when considering lens aberrations, as shown in Figure 6.5(b), but contour lines of speed and leakage have the same slope in the other cases. For lens aberrations, speed is more strongly affected by gates in the critical paths that are in the worst optical impacted region, while leakage current is less sensitive because it is equally influenced by as many gates in the best region.

The results, however, do not imply that optical proximity correction is ineffective. Optical proximity correction and phase shift masks are required to ensure printability, since they reduce corner rounding, gate pullback, etc.

The sensitivity of the results was checked by reversing large and small CD transistors for the proximity effect, Coma, and lens aberrations. In particular, the data in Table 6.2 on the proximity effect was reversed by assuming that dense transistors have the smallest CDs, rather than the largest CDs, and vice versa for isolated transistors. Similarly, the data in Table 6.3 on Coma was reversed. Identical results were obtained, i.e. no sensitivity of delay to these parameters, after controlling for leakage current.

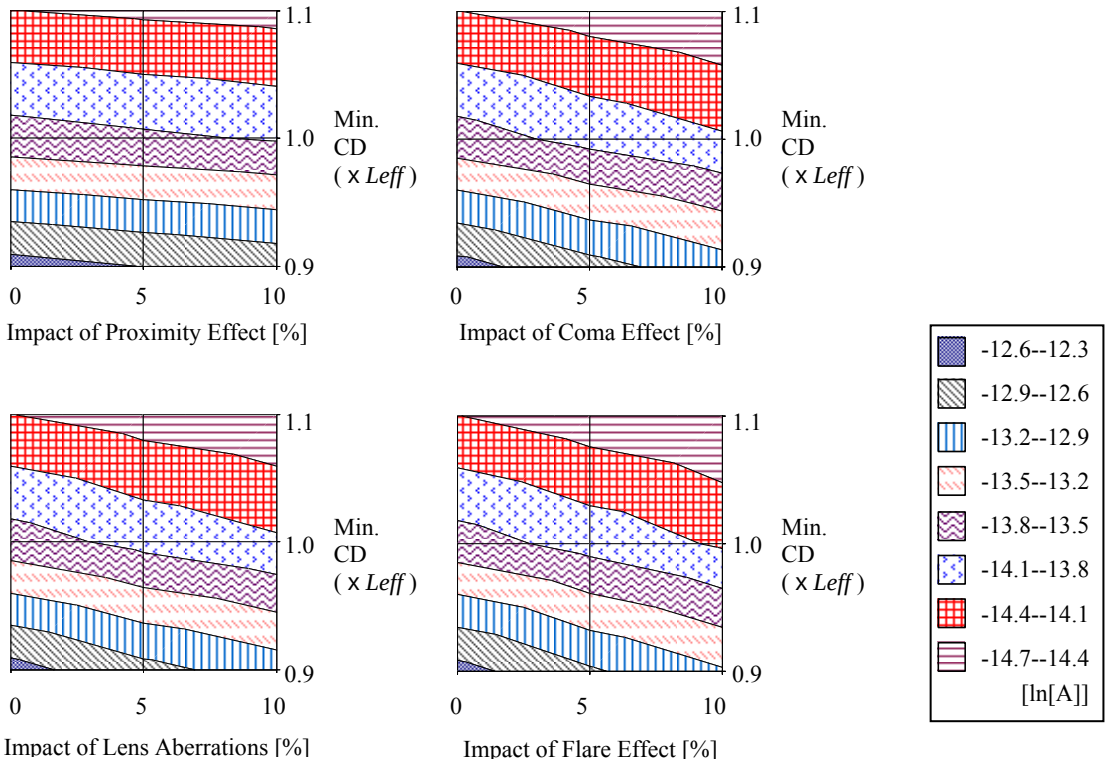


Figure 6.4. Leakage current as a function of the minimum CD and the range of variation for the proximity effect, Coma, lens aberrations, and flare (Natural log scale, *Leff* = 0.35µm).

Table 6.6: Least squares values for leakage current variation.

Coefficients / Litho. impacts	β0	β1	β2	β12	β11	β22
Proximity effect	**-13.75 ± 0.002**	**-0.92 ± 0.001**	**-0.08 ± 0.001**	**0.03 ± 0.002**	**0.23 ± 0.002**	**0.01 ± 0.002**
Coma effect	**-13.88 ± 0.016**	**-0.87 ± 0.009**	**-0.20 ± 0.009**	**0.08 ± 0.010**	**0.22 ± 0.015**	**0.02 ± 0.015**
Lens aberrations	**-13.89 ± 0.017**	**-0.87 ± 0.009**	**-0.20 ± 0.009**	**0.08 ± 0.011**	**0.22 ± 0.016**	**0.03 ± 0.016**
Flare effect	**-13.92 ± 0.013**	**-0.85 ± 0.007**	**-0.24 ± 0.007**	**0.10 ± 0.009**	**0.21 ± 0.013**	**0.02 ± 0.013**

95% confidence interval

Table 6.7: ANOVA tables for the leakage current quadratic models.

(a) ANOVA table for the leakage model impacted by the proximity effect.

Source	Sum of squares	DOF	Mean square	F ratio
Model	1,669.86	6	278.3096	272,523,347.66
Residual	0.000003	3	0.000001	Pr(F) < 0.001
Total	1,669.86	9		

(b) ANOVA table for the leakage model impacted by the Coma effect.

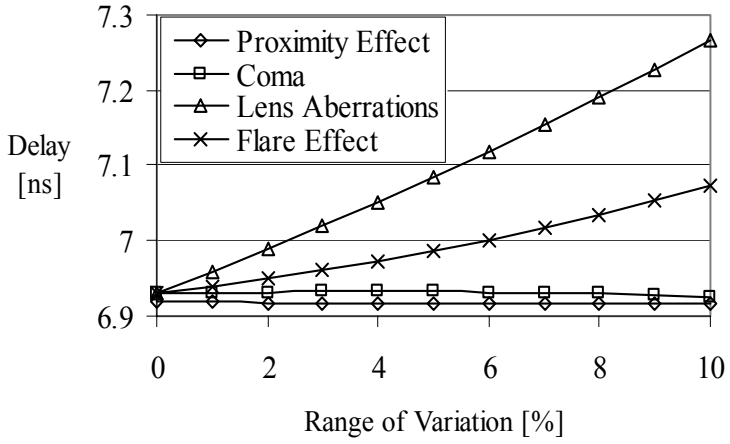
Source	Sum of squares	DOF	Mean square	F ratio
Model	1,701.37	6	283.562	6,515,797.63
Residual	0.000131	3	0.000044	Pr(F) < 0.001
Total	1,701.37	9		

(c) ANOVA table for the leakage model impacted by the lens aberrations.

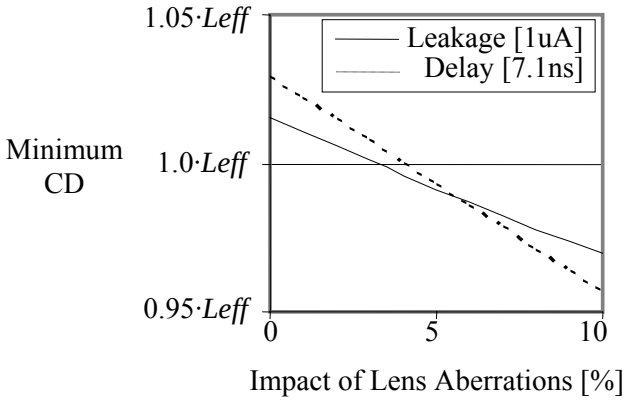
Source	Sum of squares	DOF	Mean square	F ratio
Model	1,701.27	6	283.5453	5,692,893.47
Residual	0.000149	3	0.00005	Pr(F) < 0.001
Total	1,701.27	9		

(d) ANOVA table for the leakage model impacted by the flare effect.

Source	Sum of squares	DOF	Mean square	F ratio
Model	1,709.16	6	284.8602	8,825,791.91
Residual	0.000097	3	0.000032	Pr(F) < 0.001
Total	1,709.16	9		



(a) Leakage current = 1uA



(b) Contour lines (*Leff* = 0.35μm)

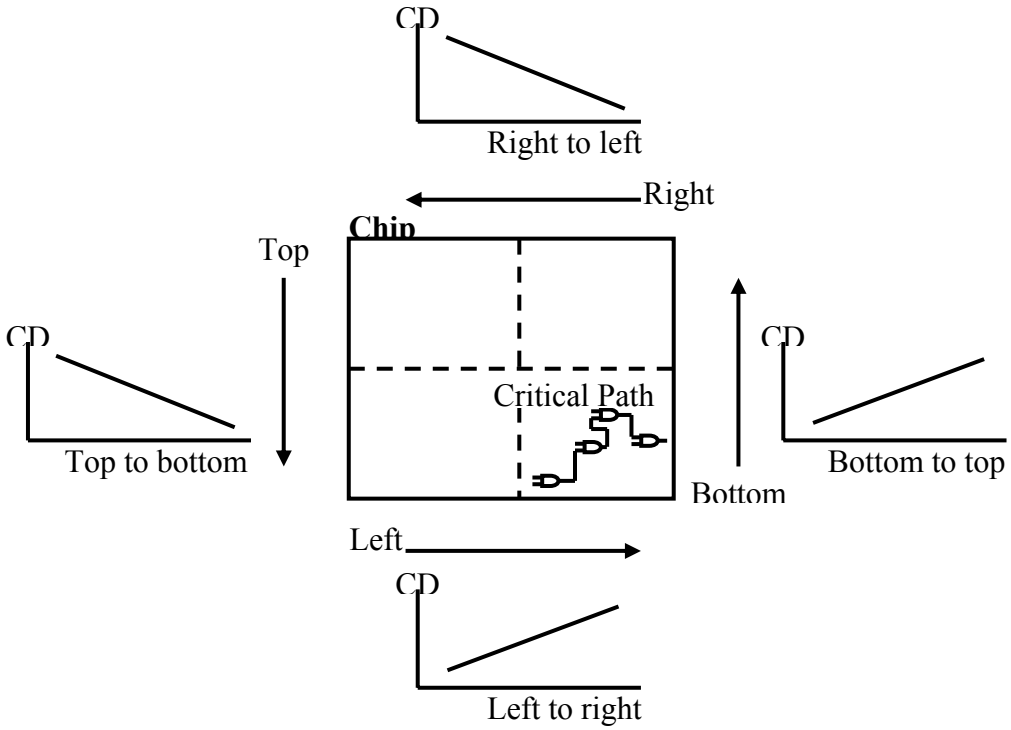
Figure 6.5. Delay sensitivity to the impact of the proximity effect, Coma, lens aberrations, and flare. As lens aberrations increase, the minimum CD is reduced to maintain constant delay and leakage current. The lower slope for the leakage current contour in (b) indicates that leakage current is less sensitive to lens aberrations. As a result, delay increases with increasing lens aberrations when the leakage current is constant, as shown in (a).

On the other hand, the results show that delay is sensitive to CD gradients caused by lens aberrations. To determine if the orientation of the gradient impacted the results, four CD gradients were simulated: left to right, right to left, top to bottom, and bottom to top in Figure 6.6(a). The results are shown in Figure 6.6(b). The graph indicates that delay is very sensitive to the direction of the CD gradient. In the example, gradients towards the top and left reduced delay, while gradients towards the bottom and right increased delay. The results are explained by the location of the critical paths in the circuit, which are in the bottom right corner of the chip in Figure 6.6(a). As a result, CD gradients with smaller CDs in this area resulted in faster chips.

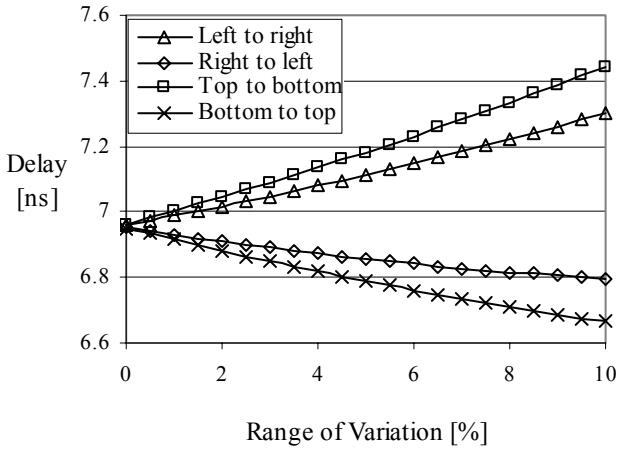
Similar results can be expected for large circuits, for the most part. In particular, large circuits are likely to have critical paths in all sectors of the chip. Sources of within-die variation that affect all transistors in a sector equally, such as lens aberrations and flare, are likely to more significantly degrade speed, compared to other factors that are averaged within critical paths, such as the proximity effect. However, unlike in this small example, it's expected that the increase in delay due to lens aberrations will be insensitive to the direction of the CD gradient, since larger circuits are less likely to have critical paths in only one sector of the die.

In this application, it is assumed that each optical effect doesn't interact with each other. To clarify this, full factorial experiments are performed. Results from a 2^4 design employed in the lithography impacted circuit delay are shown in Table 6.8. The effects calculated from the results are shown in Table 6.9. The normal probability plot of effect estimates from the circuit delay experiment is shown Figure 6.7. From the normal probability plot, the main effects of A, B, C, and D are found to be significant. In other

words, interactions between different optical effects are negligible. The analysis of variance summarized in Table 6.10 confirms these findings. It is also concluded that the different lithography effect interactions in the circuit leakage current are negligible from the 2^4 factorial design results in Table 6.11, the calculated effects in Table 6.12, the normal probability plot in Figure 6.8, and the ANOVA in Table 6.13.



(a) CD gradient



(b) Leakage current = 1uA

Figure 6.6. Delay sensitivity to all CD gradients from lens aberrations.

Table 6.8: Circuit delay data from a 2^4 factorial design.

Simulation number	A	B	C	D	Delay [ns]	Variable	-	+
1	-	-	-	-	6.7568	**A** Proximity effect (%)	0	10
2	-	-	-	+	7.4964	**B** Coma effect (%)	0	10
3	-	-	+	-	7.6888	**C** Lens aberrations (%)	0	10
4	-	-	+	+	8.4855	**D** Flare effect (%)	0	10
5	-	+	-	-	7.3595			
6	-	+	-	+	8.1378			
7	-	+	+	-	8.3383			
8	-	+	+	+	9.1782			
9	+	-	-	-	6.9656			
10	+	-	-	+	7.7249			
11	+	-	+	-	7.9170			
12	+	-	+	+	8.7359			
13	+	+	-	-	7.5821			
14	+	+	-	+	8.3816			
15	+	+	+	-	8.5820			
16	+	+	+	+	9.4454			

Table 6.9: Estimated effects from a 2^4 factorial design, circuit delay.

Effects	Estimated effects
Average	8.048485
A	0.236684
B	0.654247
C	0.995807
D	0.799474
AB	0.007667
AC	0.010744
AD	0.010830
BC	0.024935
BD	0.020828
CD	0.030267
ABC	0.000393
ABD	0.000360
ACD	0.000615
BCD	0.001094
ABCD	-0.000027

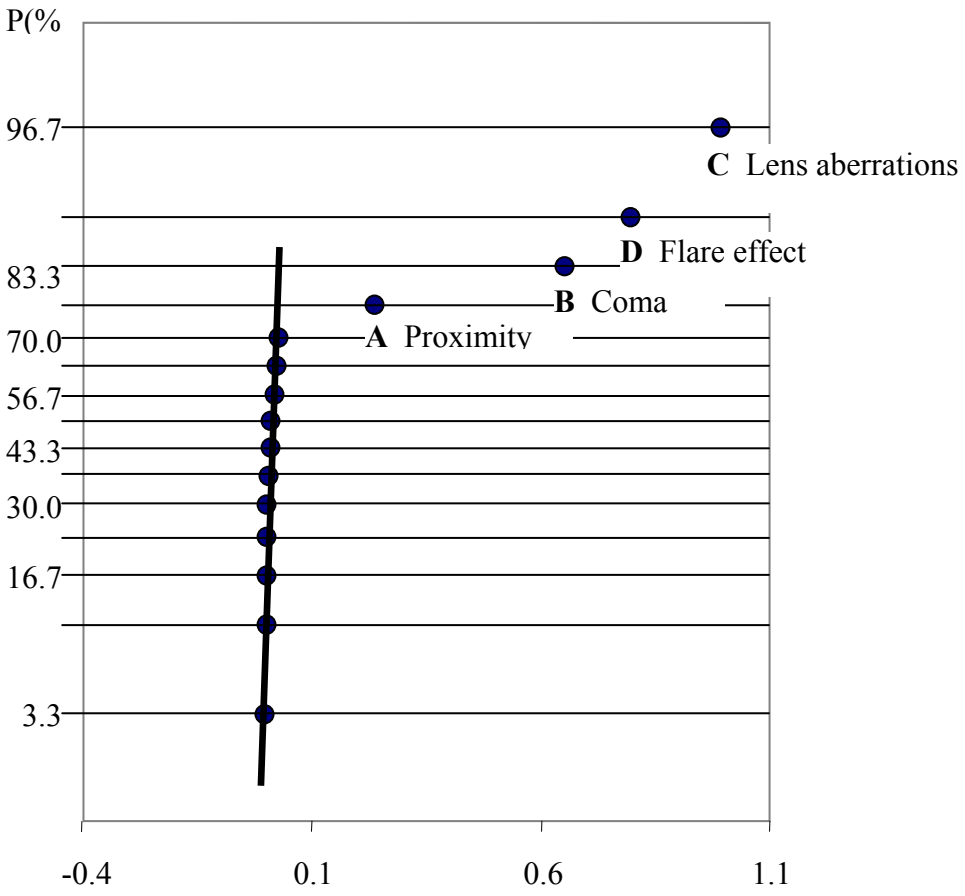


Figure 6.7. Normal plot of optical effects, circuit delay.

Table 6.10: ANOVA table for the circuit delay full factorial experiment.

Source	Sum of squares	Degrees of freedom	Mean square	F ratio	Probability (F)
A	0.003501	1	0.003501	272.1	< 0.00001
B	0.026752	1	0.026752	2078.8	< 0.00001
C	0.061977	1	0.061977	4815.9	< 0.00001
D	0.039947	1	0.039947	3104.1	< 0.00001
Error	0.000142	11	0.000013		
Total	0.132320	15			

All two-, three-, and four-factor interactions are supposed negligible.

Table 6.11: Circuit leakage current data from a 2^4 factorial design.

Simulation number	A	B	C	D	Leakage current [ns]	Natural log of leakage [ln[A]]	Variable	-	+
1	-	-	-	-	1.16E-06	-5.9372	**A** Proximity effect (%)	0	10
2	-	-	-	+	6.12E-07	-6.2130	**B** Coma effect (%)	0	10
3	-	-	+	-	7.86E-07	-6.1045	**C** Lens aberrations (%)	0	10
4	-	-	+	+	4.70E-07	-6.3279	**D** Flare effect (%)	0	10
5	-	+	-	-	7.77E-07	-6.1094			
6	-	+	-	+	4.68E-07	-6.3300			
7	-	+	+	-	5.69E-07	-6.2447			
8	-	+	+	+	3.83E-07	-6.4172			
9	+	-	-	-	9.88E-07	-6.0051			
10	+	-	-	+	5.49E-07	-6.2605			
11	+	-	+	-	6.87E-07	-6.1632			
12	+	-	+	+	4.31E-07	-6.3660			
13	+	+	-	-	6.81E-07	-6.1671			
14	+	+	-	+	4.29E-07	-6.3674			
15	+	+	+	-	5.10E-07	-6.2925			
16	+	+	+	+	3.58E-07	-6.4460			

Table 6.12: Estimated effects from a 2^4 factorial design, circuit leakage current.

Effects	Estimated effects
Average	8.048485
A	-0.047981
B	-0.124606
C	-0.121511
D	-0.213037
AB	0.005064
AC	0.004624
AD	0.010007
BC	0.014905
BD	0.026332
CD	0.025004
ABC	-0.000003
ABD	-0.000205
ACD	-0.000145
BCD	-0.001270
ABCD	-0.000182

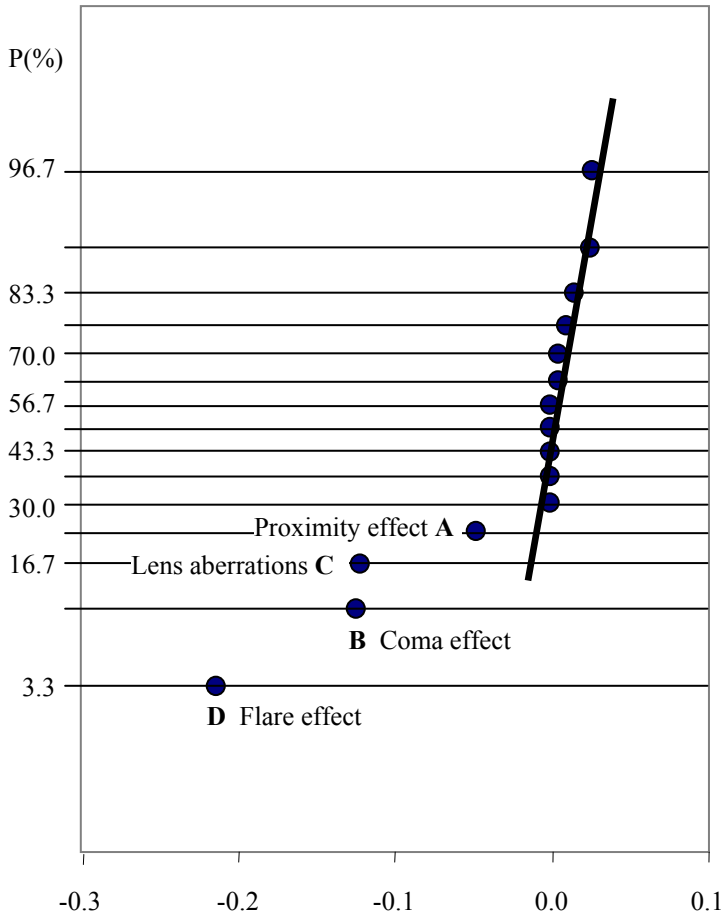


Figure 6.8. Normal plot of optical effects, circuit leakage current.

Table 6.13: ANOVA table for the circuit leakage current full factorial experiment.

Source	Sum of squares	Degrees of freedom	Mean square	F ratio	Probability (F)
A	0.000144	1	0.000144	15.0	0.0026
B	0.000970	1	0.000970	101.1	< 0.00001
C	0.000923	1	0.000923	96.1	< 0.00001
D	0.002837	1	0.002837	295.5	< 0.00001
Error	0.000106	11	0.000010		
Total	0.004979	15			

All two-, three-, and four-factor interactions are supposed negligible.

6.2 Model-Based Proximity Correction

Resolution enhancement techniques result in enlargement of the layout which requires a larger database prior to tapeout and longer mask write times for advanced technologies [7],[8]. Simpler correction schemes may reduce mask complexity. In this example, two approaches to proximity correction are applied to c7552, and the effect on delay is analyzed. One approach applies model-based correction, which is modeled as perfect correction based on the aerial image. The other approach involves correcting only categories n1n1, n3n3, and n5n5 and interpolating corrections for all other categories. It is assumed that both approaches maintain the same chip leakage current by adjusting the target CD as in the previous example.

Lithography simulation, calibrated to data from an advanced process, was used to determine the impact of the proximity effect on CD. The results are shown in Table 6.14, where CD is measured in the middle of the channel. Perfect model-based correction suppresses CD variation in Table 6.14 to zero. Table 6.15 illustrates the simpler correction scheme, where the correction amounts for the n1n1, n3n3, and n5n5 categories are determined, and other categories are interpolated from these categories. Table 6.16 presents the resulting CD variations after simple correction, where it can be seen that n1n1, n3n3, and n5n5 are corrected perfectly. However, as can be seen in Table 6.16, the variations result in shorter channel lengths for most of the other categories. To keep the chip leakage current at the same level as the model-based correction scheme, the typical gate length is increased by 5.75% in the simpler correction scheme. This quantity was obtained by iterating the gate CD target until the leakage current for the two correction schemes is equal. Final CD variations are shown in Table 6.17.

Table 6.14: Simulated CD variation (%).

(CD is measured in the middle of the channel.)

Category	**n1**	**n2**	**n3**	**n4**	**n5**
n1	26.0	4.0	14.5	14.5	17.5
n2	4.0	-17.0	-16.0	-15.0	-10.0
n3	14.5	-16.0	-13.0	-9.0	-4.0
n4	14.5	-15.0	-9.0	-5.0	-2.0
n5	17.5	-10.0	-4.0	-2.0	-1.0

Table 6.15: Simpler proximity correction scheme (%).

Category	**n1**	**n2**	**n3**	**n4**	**n5**
n1	-26.0	-16.25	-6.5	-9.5	-12.5
n2	-16.25	-6.5	3.25	0.25	-2.75
n3	-6.5	3.25	13.0	10.0	7.0
n4	-9.5	0.25	10.0	7.0	4.0
n5	-12.5	-2.75	7.0	4.0	1.0

Table 6.16: CD variation (%) after simpler correction.

Category	**n1**	**n2**	**n3**	**n4**	**n5**
n1	0.0	-12.25	8.0	5.0	5.0
n2	-12.25	-23.5	-12.75	-14.75	-12.75
n3	8.0	-12.75	0.0	1.0	3.0
n4	5.0	-14.75	1.0	2.0	2.0
n5	5.0	-12.75	3.0	2.0	0.0

Table 6.17: CD variation (%) after adjusting the chip leakage current.

Category	**n1**	**n2**	**n3**	**n4**	**n5**
n1	5.75	-7.20	14.21	11.04	11.04
n2	-7.20	-19.10	-7.73	-9.85	-7.73
n3	14.21	-7.73	5.75	6.81	8.92
n4	11.04	-9.85	6.81	7.87	7.87
n5	11.04	-7.73	8.92	7.87	5.75

Figure 6.9 shows the delay results for static timing analysis of 10 critical paths. The results inform that the simpler optical proximity correction increases the chip delay by 5.5%. Therefore, the proposed tool informs that reducing mask complexity comes at the expense of a 5.5% increase in delay. The reduction of the mask size can be deduced from the reference [97]. Figure 6.10 shows the mask file size and CD error as a function of a grid spacing used by the mask-writing tool. In this example, the CD range is 10% after the simpler OPC and the chip leakage current adjustment. As the minimum feature size is 160nm in the reference, the 10% range of CD error is 16nm. That corresponds to the grid size 7nm. Thus it is speculated that the mask size is decreased from at least 6.5 multiplier factors to 4.

These results may look inconsistent with the insensitivity to the proximity effect in Figure 6.5(a). The reason is that two different sets of data describing CD variation were used. In the second example, more transistors in the critical paths contain the worst optically impacted categories. For example, the critical paths have most gates in category n4n5. This category is less impacted than others in Table 6.2, but not in Table 6.17. Therefore, in this example, variations in CDs within critical paths are not averaged. This is a good example to show how the tool can be used to evaluate correction for different

technologies and fabrication lines for specific circuits.

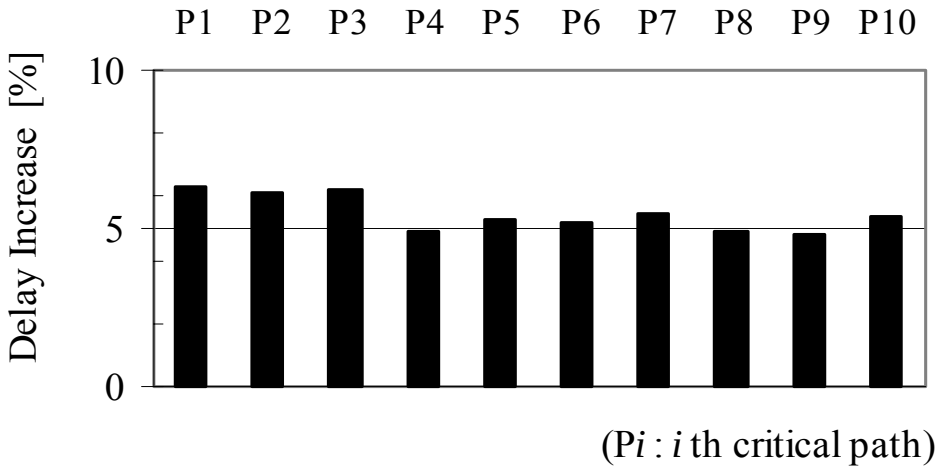
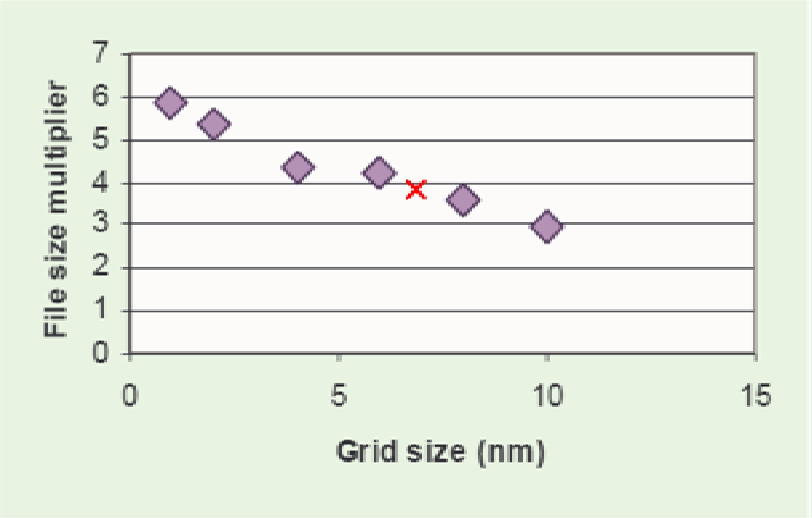
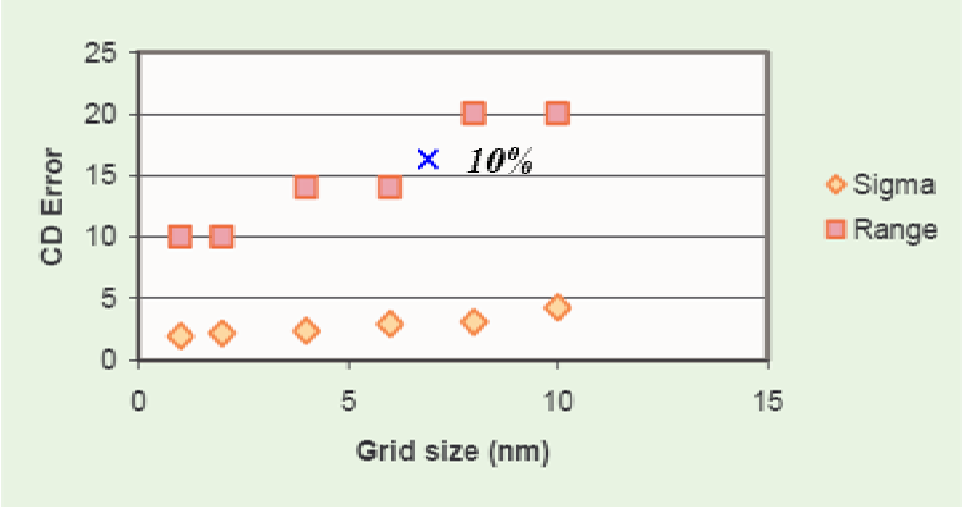


Figure 6.9. Delay impact comparing model-based proximity correction and simpler optical proximity correction for 10 critical paths, P1-P10.



(a) File size as a function of grid.



(b) CD error as a function of grid.

Figure 6.10. Mask size vs. CD error. (In this example, the CD range is 10% after the simpler OPC and the chip leakage current adjustment. The minimum feature size is 160nm in the reference.[97])

6.3 Impact of Systematic Within-Die Variations on Interconnect

In this section, the sensitivity is investigated of delay to interconnect linewidth variation due to within-die variation caused by lithography (the proximity effect, Coma, lens aberrations, flare) and interconnect thickness variation caused by CMP. The interconnect parameters used in this study are shown in Table 6.18.

Table 6.18: Interconnect parameters.

Parameter	Valu	Units
Dielectric Constant	3.5	
Metal (Cu) Resistivity	2.2	μΩ-cm
Metal 1,2,3 Wiring Pitch	0.25	μm
Metal 4 Wiring Pitch	0.6	μm
Metal 1,2,3,4 Aspect Ratio	1.5	

The proximity effect, Coma, lens aberrations, and flare are assumed to vary as in the example in Section 6.1. According to [69], CMP variation causes metal in the densest areas to be thinner and causes metal in the coarsest areas to be thicker, as shown in Figure 6.11. However, metal thickness is proportional to the density of the underlying layer of metal. Therefore, the metal with the coarsest underlying layer will be thinner, as in Figure 6.12.

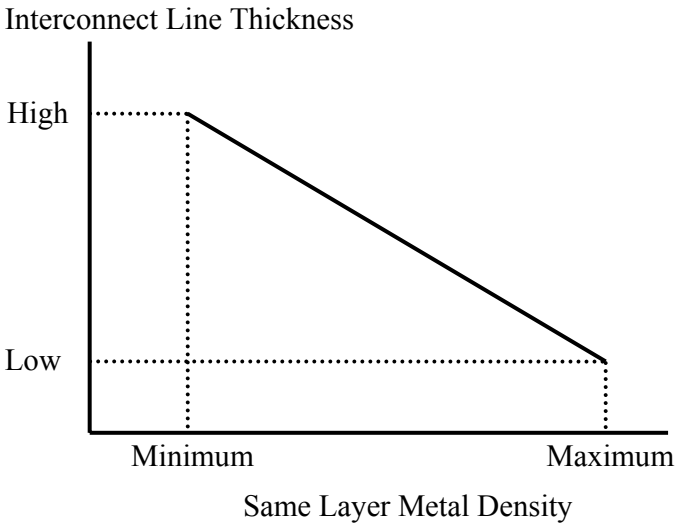


Figure 6.11. Interconnect thickness variation caused by CMP as a function of same layer pattern density in the example. (As the same layer density increases, erosion in Figure 2.11(b) reduces interconnect wire height.)

Interconnect variation was analyzed by turning off transistor CD variation. Because the transistor CDs did not vary, no compensation was made to maintain constant leakage current. The maximum impact on delay is one tenth of the range of feature variation, as shown in Figure 6.13. This is much smaller than the impact of within-die variation on transistor CDs, even though interconnect parasitics increase the path delay by over 100%. The reason is that increases in CDs reduce interconnect resistance and increase interconnect capacitance. Decreasing CDs increase resistance and reduce capacitance. Consequently, such variations are partially cancelled, as shown in Table 6.19.

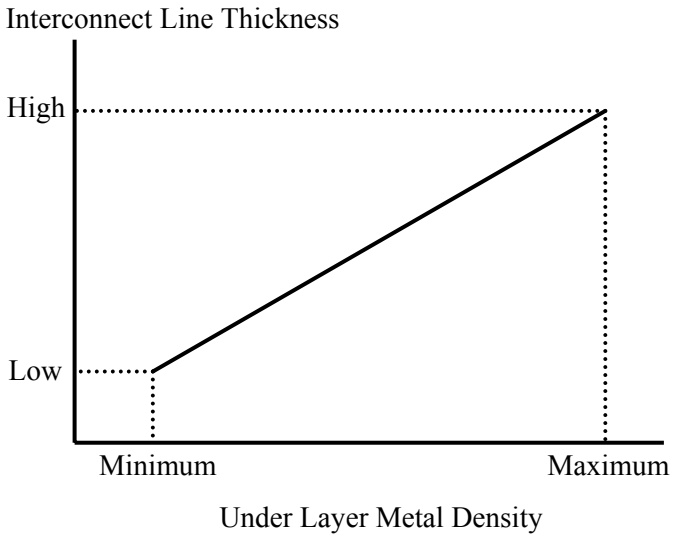


Figure 6.12. Interconnect thickness variation caused by CMP as a function of under layer pattern density in the example. (As the under layer density increases, the multilevel pattern dependency in Figure 2.11(c) leads to unchanged interconnect line height.)

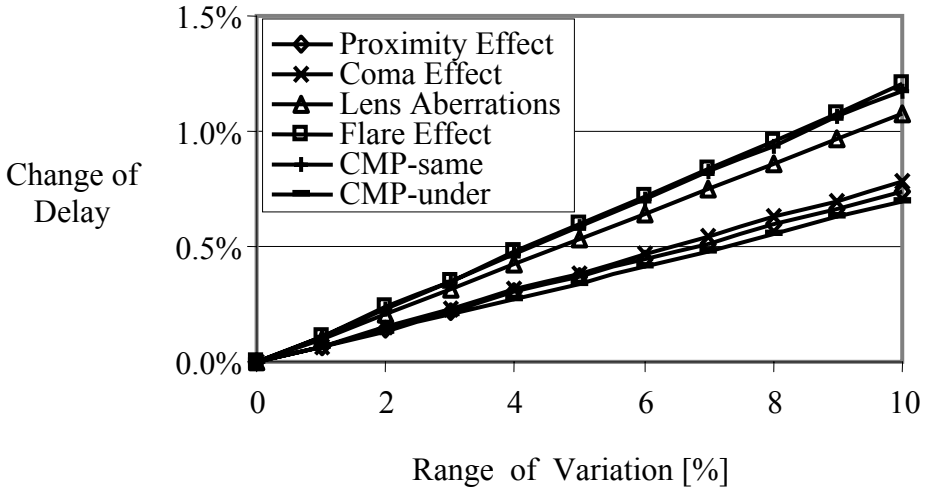


Figure 6.13. Delay sensitivity including only interconnect variation.

Table 6.19: Interconnect parasitic RC variation when the flare effect is turned on, and with a range of variation of 10%. The calculation involves summing the change in resistance and capacitance of all interconnect segments in a critical path and dividing by total resistance and capacitance in the path.

Parameter	ΔR/R [%]	ΔC/C [%]
Average	6.7	-1.8

Also, to further understand the impact of the underlying layer, the following specific data in [69] has been used on erosion:

$$T \sim (1 - 0.2 \cdot D), \quad (3)$$

and the effect of the underlying layer on erosion:

$$T \sim (0.7 + 0.3 \cdot uD), \quad (4)$$

where T is the metal thickness, D is the metal pattern density, and uD is the underlying layer metal pattern density. Both effects (the pattern density and the underlying layer pattern density) are considered by multiplying the above two relationships.

Delays of 10 critical paths of c7552 were calculated with and without the underlying layer effect. The result in Fig. 6.14 shows that delays increase by 70% on average when the underlying layer is involved. This confirms the importance of modeling multilevel pattern dependencies.

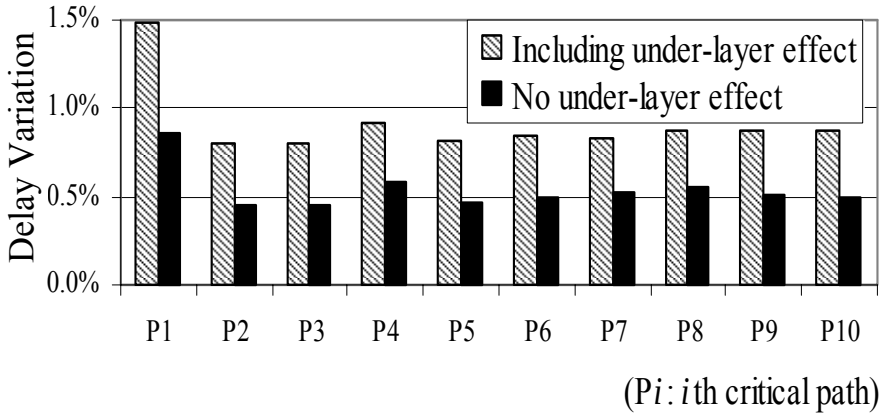


Figure 6.14. Delay sensitivity when considering the CMP effect, with and without models of the underlying layer.

6.4 Experimental Results of Lithography Fault Diagnosis

The diagnosis methodology described in Chapter 5 has been implemented in Java on a Solaris 2.8 running on a Sun E-450 Server with four CPUs and 4GB RAM. Experiments have been performed on ISCAS85 circuits [91]. Table 6.20 shows the execution time for critical path extraction and test pattern simulation. It also shows the number of critical paths, true paths, test vectors, and the maximum delay for each circuit under fault-free conditions.

In the table, the DFS time refers to the time required to enumerate all paths with a delay above 90% of the maximum circuit delay, except for c2670, where a 70% threshold is used. The pruned DFS time is the execution time when pruning is used, as described in Section 5.4. For c2670, the threshold was set to 70% to obtain a reasonable number of true paths. We can see that the execution time for the pruned DFS algorithm is 10 times faster than DFS on average.

Table 6.20: Execution times and experimental results.

Circuit	DFS Time [sec]	Pruned DFS Time [sec]	Paths*	True Paths**	Test Vectors	Maximum Dynamic Delay [nsec]	Test Pattern Simulation Time [sec]
c432	16.5	8.2	16,598	40	40	5.85	42.8
c499	2.4	0.7	1340	181	62	2.81	25.8
c880	3.8	0.2	112	95	63	3.72	55.6
c1355	976.1	229.8	229,660	4441	332	3.71	148.8
c1908	185.0	25.9	25,232	1025	942	4.61	1581.6
c2670	395.7	379.0	383,839	18	18	5.15	50.7
c3540	7417.4	233.9	159,305	225	225	6.58	483.3
c5315	486.2	79.8	71,410	1874	1874	6.04	7948.6
c7552	248.4	10.9	11,095	139	139	4.98	835.7

Circuits c432 and c2670 did not get much reduction in execution time due to pruning. In the case of c2670, 56% of the total paths have delays over 70% of the maximum circuit delay. In the case of c432, 20% of the total paths have delays over 90% of the maximum circuit delay. In other words, both of these circuits have a large number of long paths. Besides these two circuits, the other circuits have many short paths. Therefore, the search space was reduced to below 20% of the paths in these circuits, and pruning was effective.

Table 6.20 also lists the number of "true paths". It's well known that ISCAS '85 circuits are poorly testable, except for c880 [98], where almost all paths are true paths.

For each true path, test vectors are simultaneously extracted. The number of test vectors is shown in Table 6.20. Finally, fault simulations are performed for each test pattern, using dynamic timing simulation. The dynamic fault simulation time required to generate the entire fault dictionary is shown in the last column of Table 6.20.

The optical lithography faults tested in the experiment are shown in Table 6.21. In order to generate the fault dictionary for pass/fail signatures a maximum variation of 5%, 10%, and 15% of the gate length is assumed for each fault.

Table 6.21: Physical origins of faults considered.

Physical Origin Code	Optical Effects
0	Optical proximity effect (large n1n1, small n5n5)
1	Coma (large n5n1, small n1n5)
2	Lens aberrations (Left->Right)
3	Optical proximity effect (Reverse trend)
4	Coma (Reverse trend)
5	Lens aberrations (Right->Left)
6	Lens aberrations (Bottom->Top)
7	Lens aberrations (Top->Bottom)

Before looking at diagnosability of the causes of within-die variation, it is needed to look at detectability. To be detectable, at least one pattern has to pass tests applied at frequency $f = 1/0.9d_{\max}$. The results are shown in Figure 6.15. This figure indicates that detectability increases for larger ranges of variation, i.e. 15% variation is more easily detectable than 5% variation. Moreover, some circuits displayed poor detectability, while within-die variation faults in others appear to be detectable. In addition, it appears that larger circuits show improved detectability compared to smaller circuits, indicating potentially improved results for large circuits. Two other circuits, c2670 and c3540, are also large circuits, but path delays of these circuits are not distributed near the threshold ($0.9 \cdot d_{\max}$) region compared with c1908, c5315, and c7552, as shown in Figure 6.16. Thus, these circuits are less likely to be affected by the faults.

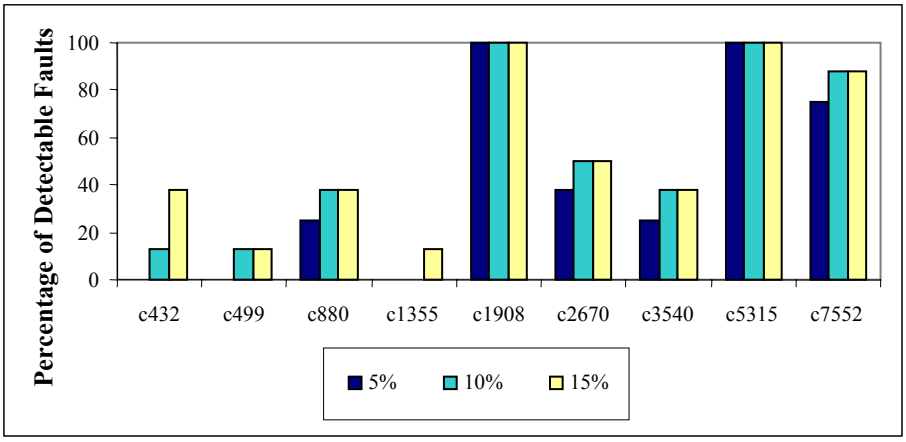
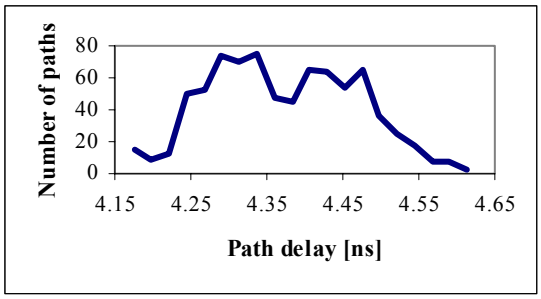


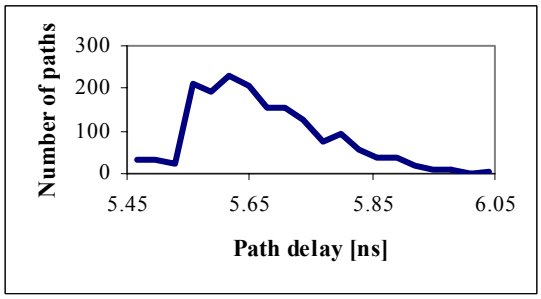
Figure 6.15. Percentage of detectable faults as a function of range of within-die variation (5%, 10%, 15%).

Diagnosis requires distinguishing among different faults. In order to do this, the correlation between the pass/fail patterns associated with the fault to be diagnosed must be significantly higher than correlations with other faults in the dictionary. First, the dictionary was constructed containing faults associated with a 10% range of variation. Testing of each circuit instance was performed at frequency $f = 1/0.9d_{\max}$, where $d_{\max}$ is the maximum operating frequency of the faulty circuit instance. The patterns in the dictionary are denoted p_{ik}, where *i* is the physical origin code and $k = 10\%$, the range of variation of the fault. In order to evaluate diagnosability, it is supposed that a circuit contains a fault, *j,* where *j* is the physical origin code, of size *l,* where *l* is the range of variation, and the pass/fail pattern, p_{jl}, is determined for tests applied at frequency $f = 1/0.9d_{\max}$, where $d_{\max}$ is the maximum operating frequency of the faulty circuit instance. Correlations are then performed with all patterns in the dictionary, i.e. $\rho(p_{jl}, p_{ik})$ is computed. The fault is correctly diagnosed if $\rho(p_{jl}, p_{ik}) > \max_{i \neq j} \rho(p_{jl}, p_{ik})$,

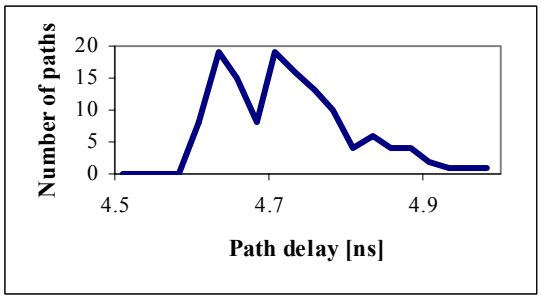
where $k = 10\%$ and l is an arbitrary range of variation. Otherwise, it is said that a fault is mis-diagnosed.



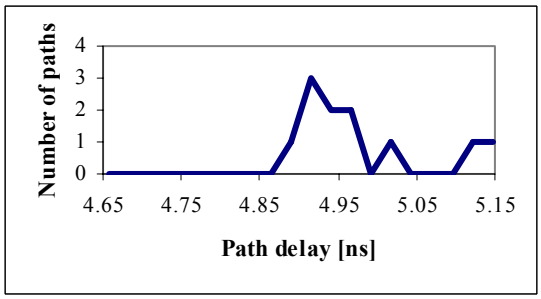
(a) c1908



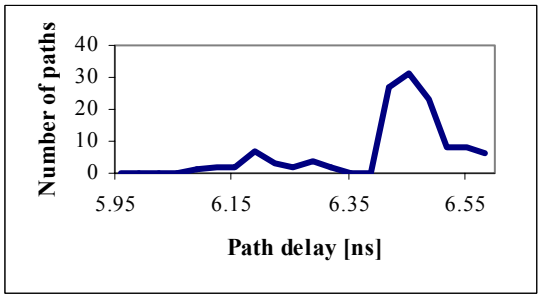
(b) c5315



(c) c7552



(d) c2670



(e) c3540

Figure 6.16. Delay distribution for some ISCAS '85 circuits (delay range: $0.9 \cdot d_{max}$ ~ d_{max}). Most path delays of c2670 and c3540 are crowded closer to d_{max}. More path delays of c1908 and c5315 are distributed near the $0.9 \cdot d_{max}$ region than for c7552.

In the examples, three circuits are considered, where faults with a 10% range of variation are mostly detectable: c1908, c5315, and c7552. Figures 6.17, 6.18, and 6.19 compare $\rho(p_{jl}, p_{ik})$ and $\max_{i \neq j} \rho(p_{jl}, p_{ik})$ for c1908, c5315, and c7552, respectively. These figures indicate that faults with a 10% and 15% range of variation are diagnosable for c1908 and c5315. Diagnosability decreases for a 5% range of variation. The test vector set for c7552 is small than for c5315 and c7552, and the feasible path set of c7552, affected by the faults, is also smaller, considering the path delay distributions in Figure 6.16. Hence, the diagnosability of c7552 is poor.

In order to attempt to improve diagnosability for smaller ranges of variation, adding pass/fail patterns associated with a 5% range of variation to the dictionary was considered. As a result, the dictionary contains patterns, p_{ik}, where i is the physical origin code and k = 5%, 10%, and 15%, the range of variation of the fault. Then, the pattern, p_{jl}, is determined for an arbitrary fault with physical origin code j and range of variation l, and it is checked to see if the fault is correctly diagnosed by determining if $\max_{k=5\%,10\%} \rho(p_{jl}, p_{ik}) > \max_{i \neq j, k=5\%,10\%} \rho(p_{jl}, p_{ik})$ where k = 5% and 10% and l is an arbitrary range of variation. Figures 6.20, 6.21, and 6.22 compare $\max_{k=5\%,10\%} \rho(p_{jl}, p_{ik})$ and $\max_{i \neq j, k=5\%,10\%} \rho(p_{jl}, p_{ik})$ for c1908, c5315, and c7552, respectively. In all circuits, the diagnosability of the 5% range of variation of the fault improves due to the pass/fail patterns in the 5% range of variation in the dictionary. However, since the 5% pass/fail patterns increase the detectabilities of other 15% faults, the diagnosability of a few 15% faults deteriorated.

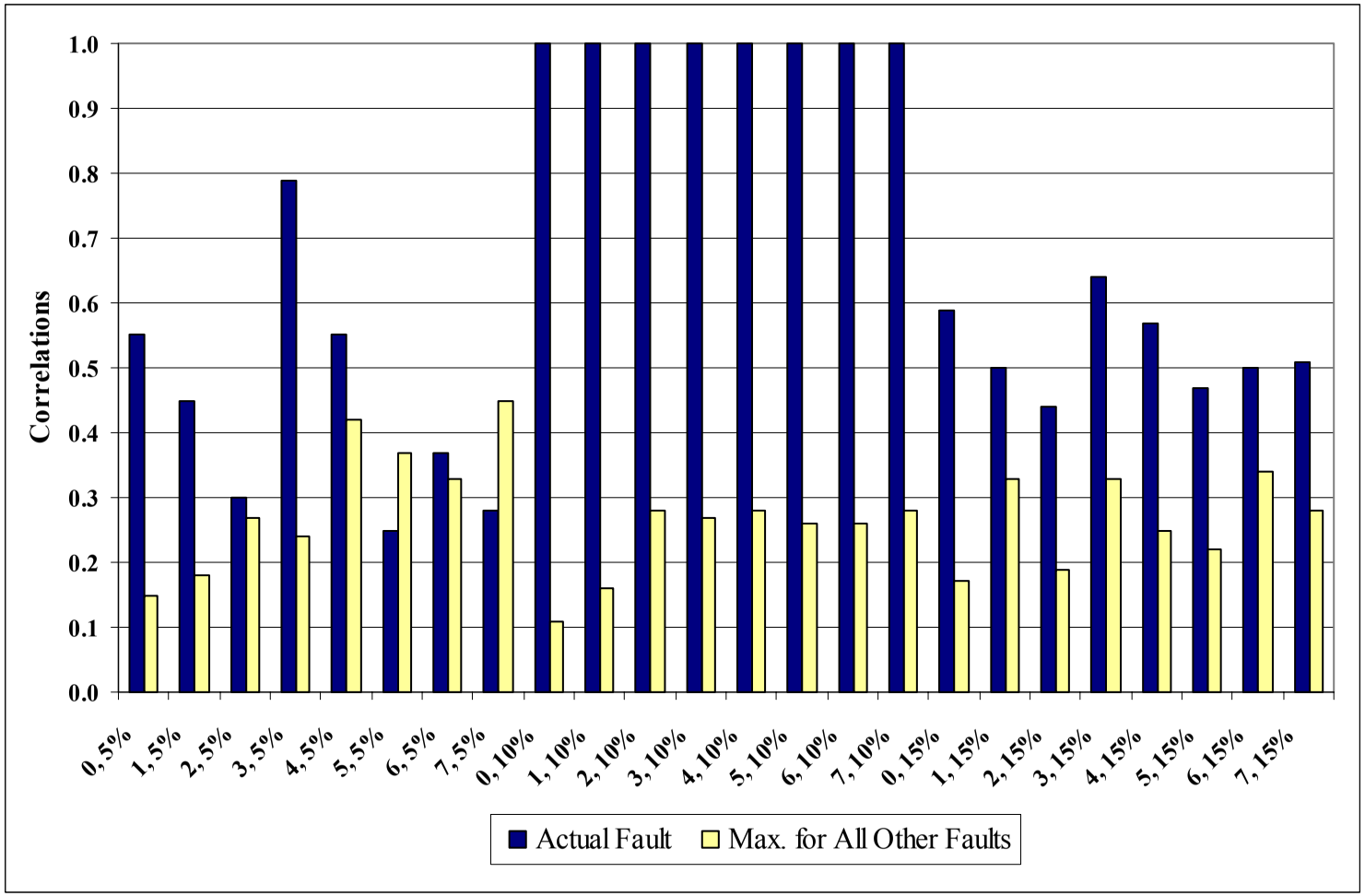


Figure 6.17. Correlations between pass/fail patterns for faults as a function of range of within-die variation (5%, 10%, 15%) for c1908. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the range of variation is 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

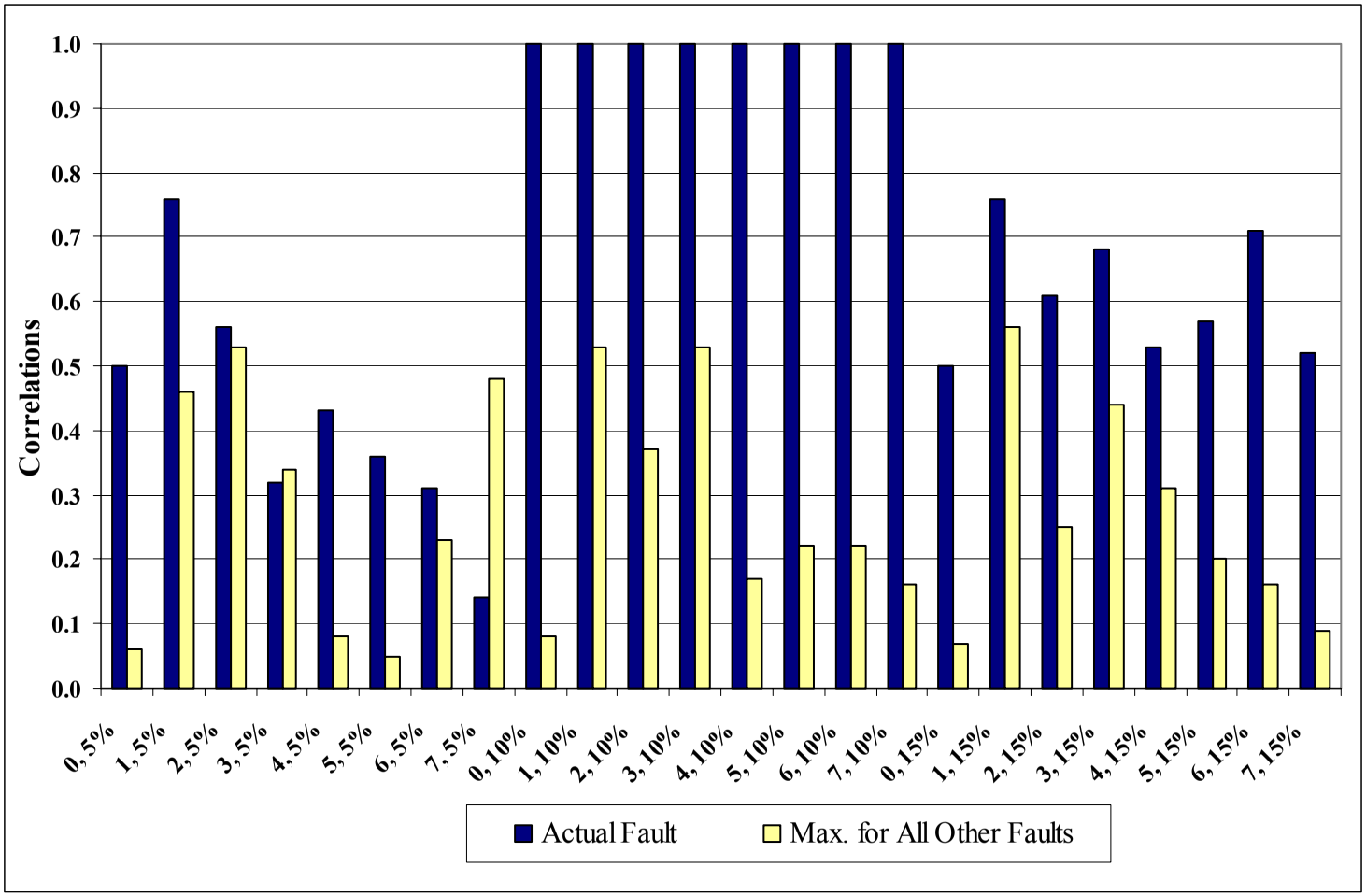


Figure 6.18. Correlations between pass/fail patterns for faults as a function of range of within-die variation (5%, 10%, 15%) for c5315. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the range of variation is 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

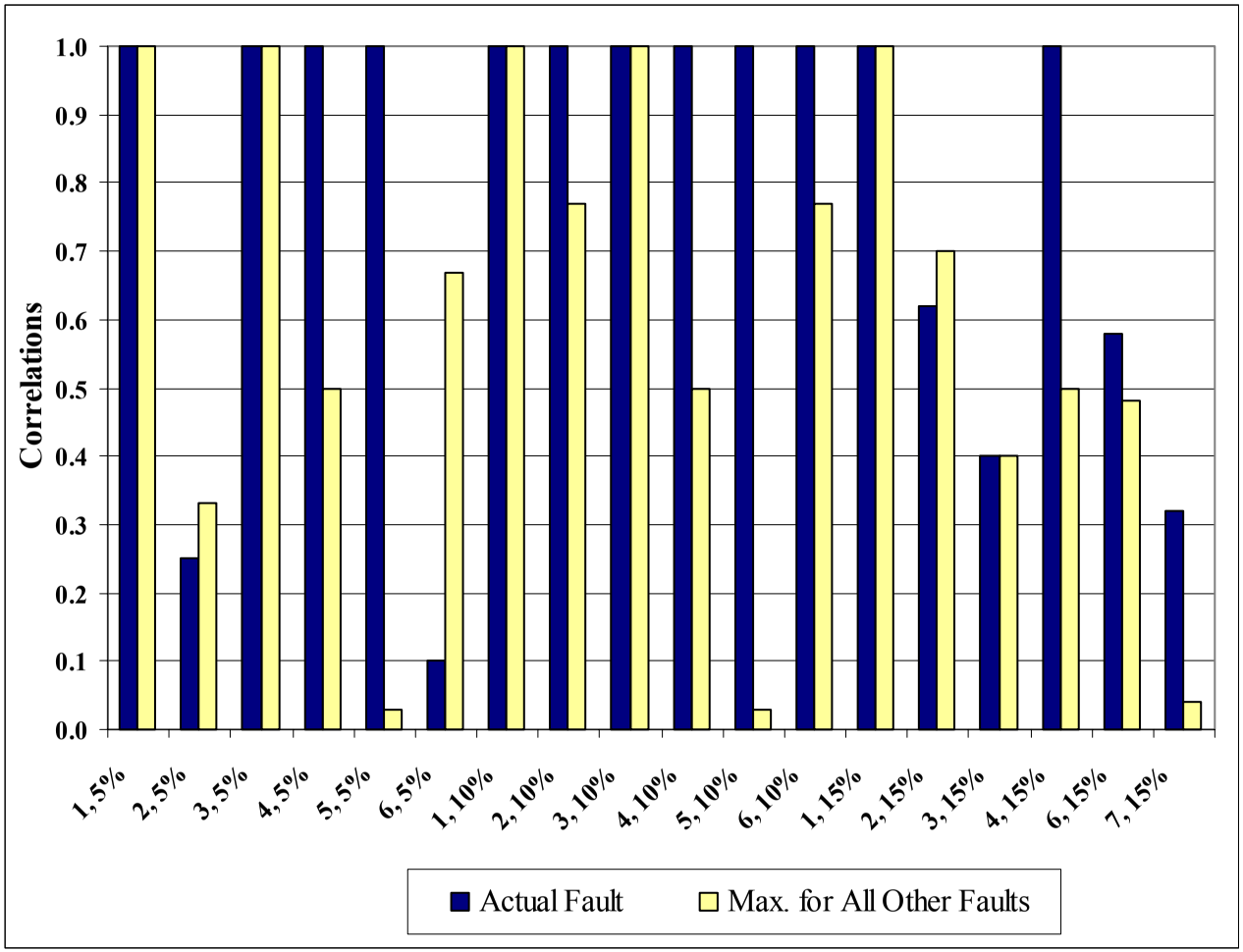


Figure 6.19. Correlations between pass/fail patterns for detectable faults as a function of range of within-die variation (5%, 10%, 15%) for c7552. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the range of variation is 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

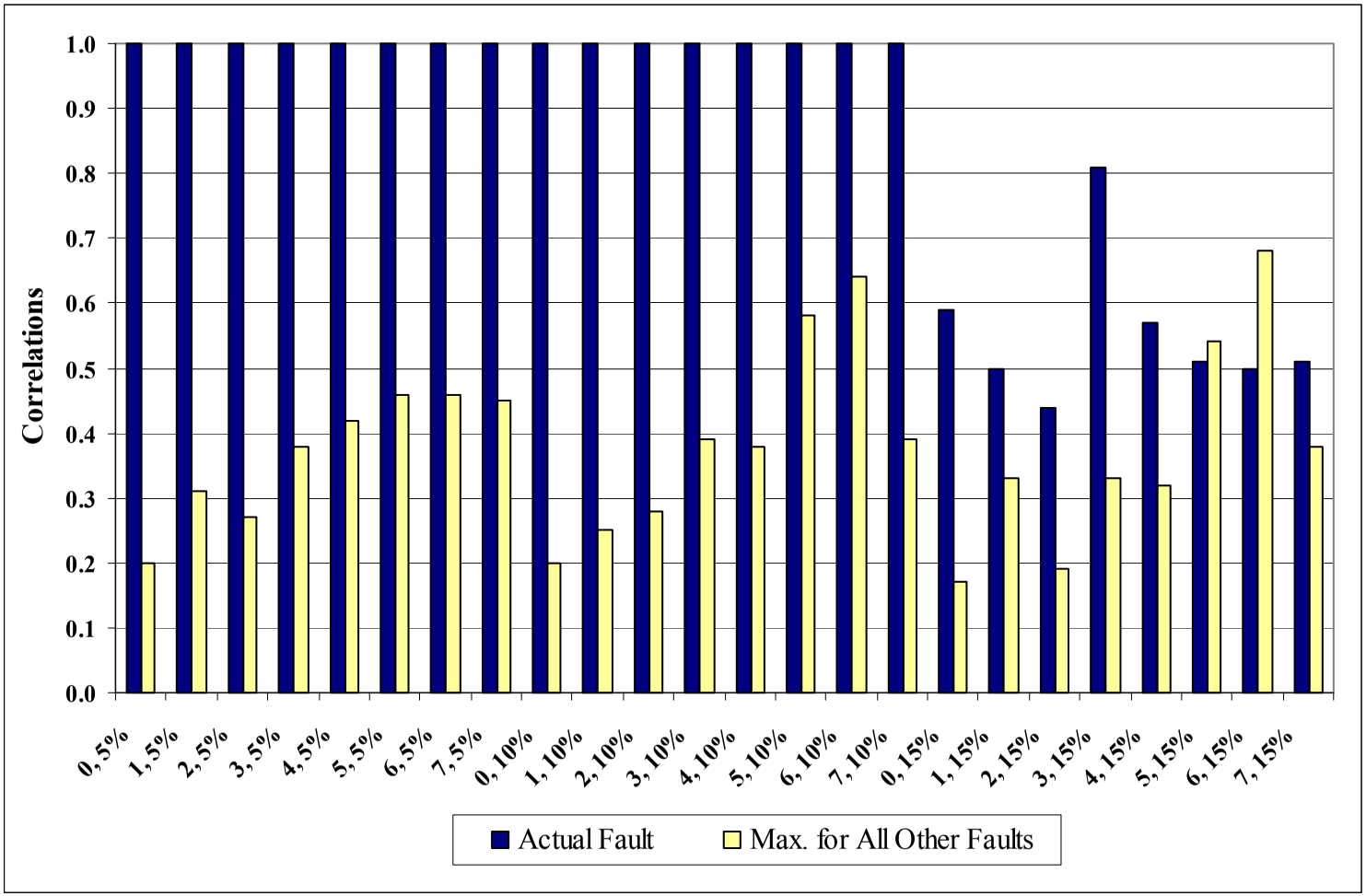


Figure 6.20. Correlations between pass/fail patterns for faults as a function of range of within-die variation (5%, 10%, 15%) for c1908. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the ranges of variations are 5% and 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

Figure 6.21. Correlations between pass/fail patterns for faults as a function of range of within-die variation (5%, 10%, 15%) for c5315. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the ranges of variations are 5% and 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

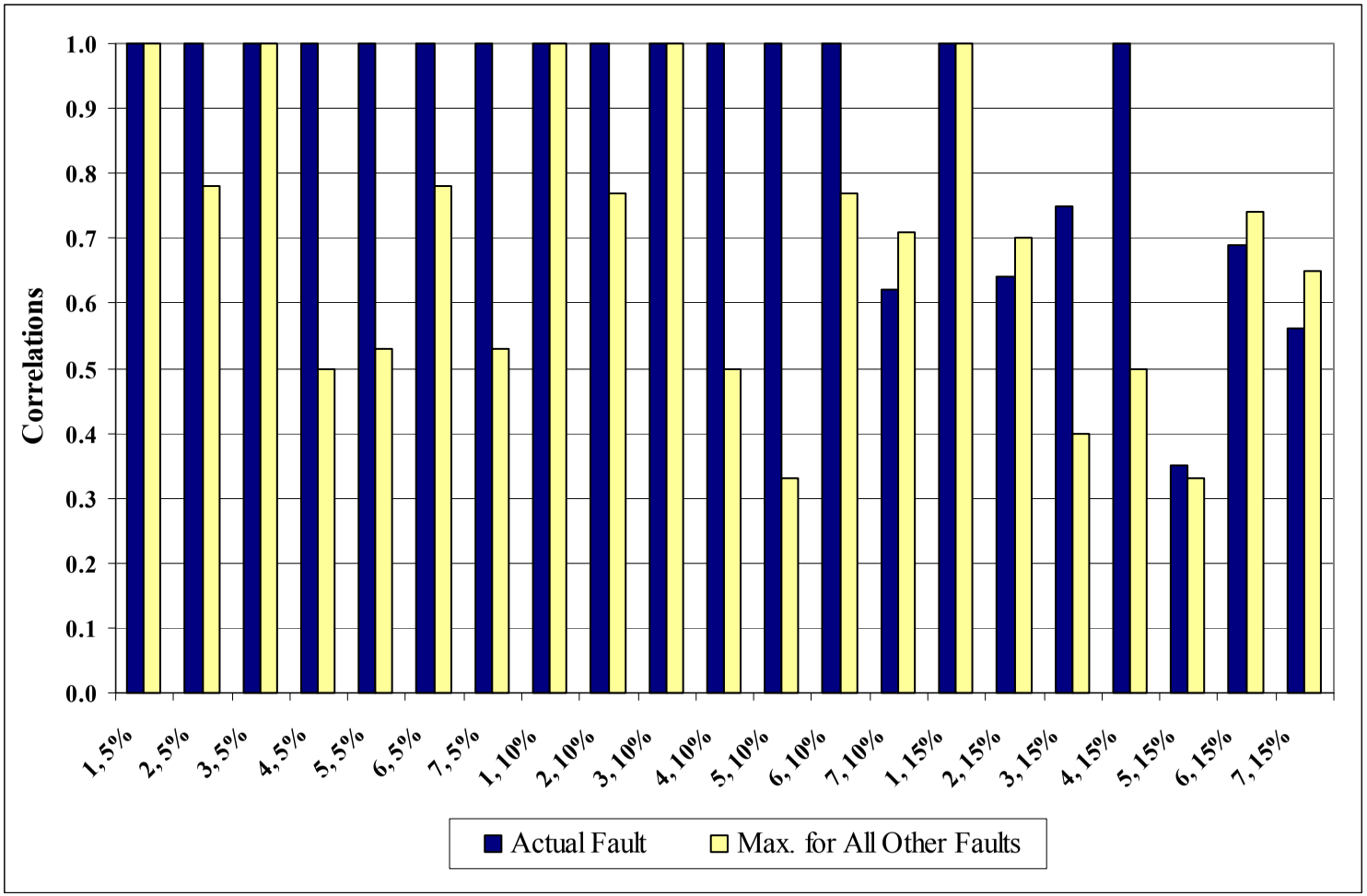


Figure 6.22. Correlations between pass/fail patterns for detectable faults as a function of range of within-die variation (5%, 10%, 15%) for c7552. The labels indicate the physical origin code and the range of variation of the fault. Each pair compares the correlation of the actual fault in the dictionary (where the ranges of variations are 5% and 10%) and the maximum correlations between pass/fail patterns for all other faults in the dictionary (excluding the actual fault).

CHAPTER 7

CONCLUSIONS

7.1 Summary of the Thesis

In this thesis, a methodology has been established to consider systematic and deterministic variation from the proximity effect, lens aberrations, flare, and CMP in circuit analysis, and a methodology has been presented to diagnose the physical origin of path delay faults caused by known imperfections in optical lithography. This methodology can be expanded to analyze and diagnose mask error [58], incorporate other geometric effects, such as micro-loading [52], [99], and reduce lithographic correction work.

7.1.1 Analysis Methodology considering Systematic Within-Die Variation from Optical Lithography

The methodology involves labeling the cell instance with information needed to account for lens aberrations and flare. Neighborhood information relating to the proximity effect on individual transistors is reflected in HSPICE files of cells by tags on the transistor names. Neighborhood information relating to the impact of the proximity effect on interconnect involves updating parasitic capacitance and resistance. The implementation of parasitic extraction is based on Cong's interconnect RC extraction methodology and Wong's analytical capacitance models. Cell transistor HSPICE netlists with the modified gate poly geometries are combined with interconnect RC HSPICE netlists with modified metal geometries to generate revised cell HSPICE netlists, which

are used to determine the revised cell delays, where delay is a function of the transition time of the previous stages and the loading capacitances of the next stages.

The simulation tool has been demonstrated with three applications. First, the relationship among speed, leakage current, the minimum CD, and the sources of systematic variation in lithography was determined for an example circuit. It was revealed that delay is affected by lens aberrations and flare after the chip leakage current is adjusted, but not by the other effects. Intuitively, local variation from factors, such as the proximity effect and Coma, is averaged within each critical path, while variation due to global factors, such as flare and lens aberrations is less likely to be averaged, unless the circuit contains many long global critical paths traversing the chip.

Second, the analysis methodology has been made use of in order to compare "perfect" model-based proximity correction with simpler optical proximity correction algorithms. The results showed that simpler optical proximity correction could alleviate mask complexity at the cost of a 5.5% increase in delay in one example.

Third, systematic variation in interconnect geometries has also been studied. The fact that variations in interconnect resistance and capacitance partially cancel each other in determining delay has resulted in a much smaller impact on circuit delay than gate poly variation. The methodology has been then employed in order to examine the under-layer effect of CMP. It was found that delay variation, caused by the under-layer effect, i.e. thickness variations caused by pattern density variations of underlying layers, is comparable to that resulting from pattern density variations of a single layer. Consequently, it is important to model multilevel pattern dependencies in CMP.

These applications have demonstrated that the proposed analysis methodology can reduce manufacturing risk associated with newly emerging imperfections in nano-scale semiconductor manufacturing.

7.1.2 Physical Origin Diagnosis Methodology of Systematic Within-Die Variation caused by Lithography Imperfection

The diagnosis methodology of the path delay fault physical origin, caused by known imperfections in optical lithography, involves layout-dependent timing analysis, taking into account deterministic within-die variation (gate length variation as a function of the local neighborhood, location in the reticle, and pattern density). Critical paths are extracted by pruned DFS and used to establish test patterns and pass/fail patterns associated with each fault. The results are stored in a dictionary. Observed faults are matched with simulated pass/fail patterns to diagnose the physical origin of within-die variation and to help properly allocate mask correction effort.

The detectability of faults increases for the larger faults and the larger circuits. In addition, the detectability depends on the path delay distribution of the circuit. The diagnosabilities for three ISCAS ’85 circuits, c1908, c5315, and c7552 are investigated. If the dictionary has the pass/fail patterns associated with several ranges of variation, diagnosability is improved.

7.2 Future Work

7.2.1 Improvement of the proposed methodology

The major bottleneck of the methodology is the pre-characterization of the gate cell delay impacted by the systematic variation. To reduce the computational cost of the pre-characterization, analytic gate cell delay models [76] or efficient dynamic simulation [77] can be candidates to improve computational cost.

The diagnosis methodology employs the path enumeration and the test pattern generation separately. They can be combined by upgrading the algorithms proposed in [28], [29], and the test pattern generation for each path can be performed simultaneously.

7.2.2 Approach to the full-custom circuits

The proposed methodology in this thesis is based on the standard gate cell design flow. If the timing verification uses HSPICE [17], it's straightforward to apply the methodology to full-custom circuits. However, as the standard gate cell design flow uses static timing analysis, full-custom design requires fast timing analysis for a quick filter. Thus it will be investigated how to consider within-die variation in fast timing analysis for full-custom circuits, for instance, in symbolic DC formulations [100].

7.2.3 New layout-dependent phenomena

Several process-originated mechanisms in Chapter 2, affecting CD variation, should be researched.

Recently, transistor performance is greatly improved by strain engineering [101]. Local strain is introduced by an eSiGe source/drain [102], dual stress liner (DSL) [103], and stress memorization techniques [104]. Mobility enhancement by strain is strongly

affected by the layout in the neighborhood of a device [102], [105]. Therefore, the layout-dependent strain effect on the circuit performance will be analyzed as future work.

REFERENCES

[1] W. Maly, *et al.*, "Design-manufacturing interface: Part I – Vision," *Proc. Design, Automation, and Test in Europe*, 1998, pp. 550-556.

[2] S. R. Nassif, "Modeling and forecasting of manufacturing variations," *Int. Workshop on Statistical Metrology*, 2000, pp. 2-10.

[3] A. K. Wong, "Microlithography: trends, challenges, solutions, and their impact on design," *IEEE Micro*, vol. 23, pp. 12-21, Mar./Apr. 2003.

[4] H. J. Levinson, *Principles of lithography*. SPIE PRESS, 2001.

[5] Y. Chen, *et al.*, "Area fill synthesis for uniform layout density," *IEEE Trans. Computer-Aided Design,* vol. 21, pp. 1132-1147, Oct. 2002.

[6] B. E. Stine, *et al.*, "The physical and electrical effects of metal-fill patterning practices for oxide chemical-mechanical polishing processes," *IEEE Trans. Electron Devices,* vol. 45, pp. 665-679, Mar. 1998.

[7] Y. Borodovsky, *et al.*, "Lithography strategy for 65nm node," *Proc. SPIE,* pp.1-14, 2002.

[8] P. Gupta, *et al.*, "A cost-driven lithographic correction methodology based on off-the-shelf sizing tools," *Proc. IEEE/ACM Design Automation Conf.,* 2003, pp. 16-21.

[9] S. R. Nassif, A. J. Strojwas, and S. W. Director, "A methodology for worst-case analysis of integrated circuits," *IEEE Trans. Computer-Aided Design,* vol. CAD-5, pp. 104-113, Jan. 1986.

[10] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," *Proc. IEEE/ACM Design Automation Conf.,* 2002, pp. 556-561.

[11] Y. Liu, L. T. Pileggi, and A. J. Strojwas, "Model order-reduction of RC(L) interconnect including variational analysis," *Proc. IEEE/ACM Design Automation Conf.,* 1999, pp. 201-206.

[12] M. Orshansky, *et al.*, "Impact of spatial intra-chip gate length variability on the performance of high speed digital circuits," *IEEE Trans. Computer-Aided Design,* vol. 21, pp. 544-553, May 2002.

[13] X-Y. Li, *et al.*, "An effective method of characterization poly gate CD variation and its impact on product performance and yield," *Proc. Int. Symp. Semiconductor Manufacturing,* 2003, pp. 259-262.

[14] D. G. Chinnery and K. Keutzer, "Closing the gap between ASIC and custom: An ASIC perspective," *Proc. IEEE/ACM Design Automation Conf.,* 2000, pp. 637-642.

[15] B.E. Stine, *et al.,* "Simulating the impact of pattern-dependent poly-CD variation on circuit performance," *IEEE Trans. Semiconductor Manufacturing,* vol. 11, pp. 552-556, Nov. 1998.

[16] L. Chen, *et al.*, "Analysis of the impact of proximity correction algorithms on circuit performance," *IEEE Trans. Semiconductor Manufacturing,* vol. 12, pp. 313-322, Aug. 1999.

[17] Star-Hspice Manual, Avant!, 1998.

[18] V. Mehrotra, *et al.,* "A methodology for modeling the effects of systematic within-die interconnect and device variation on circuit performance," *Proc. IEEE/ACM Design Automation Conf.,* 2000, pp. 172-175.

[19] A. Gattiker, *et al.,* "Static timing analysis based circuit-limited-yield estimation," *Proc. ISCAS,* 2002, pp. 81-84.

[20] A. Agarwal, V. Zolotov, and D. T. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Trans. Computer-Aided Design,* vol. 22, pp. 1243-1260, Sep. 2003.

[21] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," *Proc. ICCAD,* 2003, pp. 900-907.

[22] G. L. Smith, "Modeling for delay faults based upon paths," *Proc. Int. Test Conf.,* 1985, pp. 342-349.

[23] W. Ke and P.R. Menon, "Synthesis of delay-verifiable combinational circuits," *IEEE Trans. Computers,* vol. 44, no. 2, pp. 213-222, Feb. 1995.

[24] R.C. Tekumalla and P.R. Menon, "Identification of primitive faults in combinational and sequential circuits," *IEEE Trans. Computer-Aided Design,* vol. 20, pp. 1426-1442, Dec. 2001.

[25] M. Sivaraman and A.J. Strojwas, "Primitive path delay faults: Identification and their use in timing analysis," *IEEE Trans. Computer-Aided Design,* vol. 19, pp. 1347-1362, Nov. 2000.

[26] A. Krstic, K.-T. Cheng, and S.T. Chakradhar, "Primitive delay faults: Identification, testing, and design for testability," *IEEE Trans. Computer-Aided Design,* vol. 18, pp. 669-684, June 1999.

[27] M. Sharma and J.H. Patel, "Testing of critical paths for delay faults," *Proc. Int. Test Conf.,* 2001, pp. 634-641.

[28] M. Sharma and J. H. Patel, "Finding a small set of longest testable paths that cover every gate" *Proc. Int. Test Conf.*, 2002, pp. 974-982.

[29] W. Qiu and D.M.H. Walker, "An efficient algorithm for finding the K longest testable paths through each gate in a combination circuit," *Proc. Int. Test Conf.*, 2003, pp. 592-601.

[30] S.H.C. Yen, D.H.C. Du, and S. Ghanta, "Efficient algorithms for extracting the K most critical paths in timing analysis," *Proc. Design Automation Conf.,* 1989, pp. 649-654.

[31] S. Kundu, "An incremental algorithm for indentification of longest (shortest) paths," *Integration the VLSI Journal*, pp. 25-31, 1994.

[32] L.-C. Wang, J.-J. Liou, and K.T. Cheng, "Critical path selection for delay fault testing based upon a statistical timing model," *IEEE Trans. Computer-Aided Design,* vol. 23, pp. 1550-1565, Nov. 2004.

[33] C. Visweswariah, "Death, taxes and failing chips," *Proc. Design Automation Conf.,* 2003, pp. 343-347.

[34] S.R. Nassif, D. Boning, and N. Hakim, "The care and feeding of your statistical static timer," *Proc. Int. Conf. on Computer-Aided Design,* 2004, pp. 138-139.

[35] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design,* IEEE Press, 1990, ch. 12.

[36] X. Wen *et al.,* "On per-test fault diagnosis using X-fault model," *Proc. Int. Conf. on Computer-Aided Design,* 2004, pp. 633-640.

[37] P. Girard, C. Landrault, and S. Pravossoudovitch, "A novel approach to delay-fault diagnosis," *Proc. Design Automation Conf.,* 1992, pp. 357-360.

[38] Z. Wang, *et al.,* "Delay-fault diagnosis using timing information," *IEEE trans. Computer-Aided Design,* vol. 24, pp. 1315-1325, Sept. 2005.

[39] A. Krstic *et al.,* "Delay defect diagnosis based upon a statistical timing model – the first step," *IEE Proc. – Computers and Digital Techniques,* vol. 150, pp. 346-354, Sept. 2003.

[40] P. Pant, *et al.*, "Path delay fault diagnosis in combinational circuits with implicit fault enumeration," *IEEE Trans. Computer-Aided Design,* vol. 20, pp. 1226-1235, Oct. 2001.

[41] S. Padmanaban and S. Tragoudas, "An implicit path-delay fault diagnosis methodology," *IEEE Trans. Computer-Aided Design,* vol. 22, pp. 1399-1408, Mar. 2003.

[42] M. Sivaraman and A. J. Strojwas, "Path delay fault diagnosis and coverage-A metric and an estimation technique," *IEEE Trans. Computer-Aided Design,* vol. 20, pp. 440-457, Mar. 2001.

[43] A. Krstic *et al.,* "Diagnosis-based post-silicon timing validation suing statistical tools and methodologies," *Proc. Int. Test Conf.,* 1999, pp. 558-567.

[44] W.-Y. Chen, S.K. Gupta, and M.A. Breuer, "Test generation for crosstalk-induced delay in integrated circuits," *Proc. Int. Test Conf.,* 1999, pp. 191-200.

[45] A. Krstic, *et al.,* "Delay testing considering crosstalk-induced effects," *Proc. Int. Test Conf.,* 2001, pp. 558-567.

[46] J.-J. Liou, *et al.*, "Modeling, testing, and analysis for delay defects and noise effects in deep submicron devices," *IEEE Trans. Computer-Aided Design,* vol. 22, pp. 756-769, Jun. 2003.

[47] A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Pattern generation for delay testing and dynamic timing analysis considering power-supply noise effects," *IEEE Trans. Computer-Aided Design,* vol. 20, pp. 416-425, Mar. 2001.

[48] D. Lee, D. Blaauw, and D. Sylvester, "Gate oxide leakage current analysis and reduction for VLSI circuits," *IEEE Trans. Very Large Scale Integration Systems*, vol. 12, pp. 155-166, Feb. 2004.

[49] H. Yang, *et al.*,, "Current mismatch due to local dopant fluctuations in MOSFET channel," *IEEE Trans. Electron Devices*, vol. 50, pp. 2248-2254, Nov. 2003.

[50] H. Zhang, J. P. Cain, and C. J. Spanos, "Compact formulation of mask error factor for critical dimension control in optical lithography," *Proc. SPIE,* vol. 4689, pp. 462-465, 2002.

[51] M. Nagase, *et al.*, "Accurate gate CD control for 130nm CMOS technology node," *Proc. Int. Symp. Semiconductor Manufacturing,* 2003, pp. 183-186.

[52] Y. Granik, "Correction for etch proximity: new models and applications," *Proc. SPIE,* vol. 4346, pp. 98-112, 2001.

[53] L. Van den hove, K. Lonse, and R. Pforr, "Optical lithography techniques for 0.25um and below: CD control issues," *Proc. Int. Symp. on VLSI Technology, Systems, and Applications,* 1995, pp. 24-30.

[54] Calibre Verification User's Manual, Mentor Graphics, 2002.

[55] T. Linton, *et al.,* "Determination of the line edge roughness specification for 34nm devices," *Proc. IEDM,* 2002, pp. 303-306.

[56] M. Hane, T. Ikezawa, and T. Ezaki, "Coupled atomistic 3D process/device simulation considering both line-edge roughness and random-discrete-dopant effects," *Proc. Int. Conf. Simulation of Semiconductor Process and Devices*, 2003, pp. 99-102.

[57] T.A. Brunner, "Impact of lens aberrations on optical lithography," *IBM J. of Research and Development,* vol. 41, pp. 57-67, Jan./March 1997.

[58] J. P. Cain, H. Zhang, and C. J. Spanos, "Optimum sampling for characterization of systematic variation in photolithography," *Proc. SPIE*, vol. 4689, 2002, pp. 430-442.

[59] Silicon Ensemble Reference Manual, Cadence, 1998

[60] B. W. Smith, "Variations to the influence of lens aberration invoked with PSM and OAI," *Proc. SPIE,* vol. 3679, 1999, pp. 330-346.

[61] B. W. Smith and J. S. Petersen, "Influences of off-axis illumination on optical lens aberration," *J. Vac. Sci. Technol. B*, pp. 3405-3410, Nov/Dec 1998

[62] R. L. Kostelak, E. L. Raab, and S. Vaidya, "Impact of lens aberrations on phase-shifting masks," *J. Vac. Sci. Technol. B*, pp. 3793-3798, Nov/Dec 1994

[63] C. A. Mack, "Measuring and modeling flare in optical lithography," *Proc. SPIE,* vol. 5040, 2003, pp. 151-161.

[64] T. M. Jeong, *et al.*, "Measurement of the flare and in-field line width variation due to the flare," *Proc. SPIE,* vol. 4691, 2002, pp. 1465-1473.

[65] M. Osawa, *et al.*, "Correction for local flare effects approximated with double Gaussian profile in ArF lithography," *J. Vac. Sci. Technol. B*, pp. 2806-2809, Nov/Dec 2003

[66] D. O. Ouma, *et al.*, "Characterization and modeling of oxide chemical-mechanical polishing using Planarization length and pattern density concepts," *IEEE Trans. Semiconductor Manufacturing,* vol. 15, pp. 232-244, May 2002.

[67] T. Park, T. Tugbawa, and D. Boning, "Overview of methods for characterization of pattern dependencies in copper CMP," *Proc. Chemical Mechanical Polish for ULSI Multilevel Interconnection Conf.,* 2000, pp. 196-205.

[68] T. Tugbawa, *et al.*, "Modeling of pattern dependencies in abrasive-free copper chemical mechanical polishing processes," *Proc. VLSI Multilevel Interconnect Conf.,* 2001, pp. 113-122.

[69] S. Lakshminarayanan, P. J. Wright, and J. Pallinti, "Electrical characterization of the copper CMP process and derivation of metal layout rules," *IEEE Trans. Semiconductor Manufacturing,* vol. 16, pp. 668-676, Nov. 2003.

[70] B. E. Stine, D. S. Boning, and J. E. Chung, "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE Trans. Semiconductor Manufacturing,* vol. 10, pp. 24-41, Feb. 1997.

[71] X. Ouyang, C. N. Berglund, and R. F. W. Pease, "High-throughput mapping of short-range spatial variations using active electrical metrology," *IEEE Trans. Semiconductor Manufacturing*, vol. 15, pp. 108-117, Feb. 2002.

[72] X. Ouyang, T. Deeter, C. N. Berglund, R. F. W. Pease, J. Lee, and M. A. McCord, "High-throughput high-density mapping and spectrum analysis of transistor gate length variations in SRAM circuits," *IEEE Trans. Semiconductor Manufacturing*, vol. 14, pp. 318-329, Nov. 2001.

[73] S. Ohkawa, M. Aoki, and H. Masuda, "Analysis and characterization of device variations in an LSI chip using an integrated device matrix array," *IEEE Trans. Semiconductor Manufacturing*, vol. 17, pp. 155-165, May 2004.

[74] M. Yamamoto, H. Endo, and H. Masuda, "Development of a large-scale TEG for evaluation and analysis of yield and variation," *IEEE Trans. Semiconductor Manufacturing*, vol. 17, pp. 111-122, May 2004.

[75] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate CD variability, its impact on circuit performance, and spatial mask-level correction," *IEEE Trans. Semiconductor Manufacturing*, vol. 17, pp. 2-11, Feb. 2004.

[76] A. Chatzigeorgiou, S. Nikolaidis, and I. Tsoukalas, "A modeling technique for CMOS gates," *IEEE Trans. Computer-Aided Design,* vol. 18, pp. 557-575, May 1999.

[77] Y.-H. Shih, Y. Leblebici, and S. M. Kang, "ILLIADS: A fast timing and reliability simulator for digital MOS circuits," *IEEE Trans. Computer-Aided Design,* vol. 12, pp. 1387-1402, Sep. 1993.

[78] M. A. Horowitz, "Timing models for MOS circuits," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, Jan. 1984.

[79] W. H. Kao, *et al.*, "Parasitic extraction: Current state of the art and future trends," *Proc. of the IEEE*, vol. 89, pp. 729-739, May 2001.

[80] J. Cong, *et al.*, "Analysis and justification of a simple, practical 2 ½-D capacitance extraction methodology," *Proc. IEEE/ACM Design Automation Conf.*, 1997, pp. 627-632.

[81] S-C. Wong, *et al.*, "An empirical three-dimensional crossover capacitance model for multilevel interconnect VLSI circuits," *IEEE Trans. Semiconductor Manufacturing*, vol. 13, pp. 219-227, May 2000.

[82] S-P. Sim, *et al.*, "A unified RLC model for high-speed on-chip interconnects," *IEEE Trans. Electron Devices*, vol. 50, pp. 1501-1510, Jun. 2003.

[83] Y. Massoud, *et al.*, "Managing on-chip inductive effects," *IEEE Trans. Very Large Scale Integration Systems*, vol. 10, pp. 789-798, Dec. 2002.

[84] K. Gala, *et al.*, "Inductance model and analysis methodology for high-speed on-chip interconnect," *IEEE Trans. Very Large Scale Integration Systems*, vol. 10, pp. 730-745, Dec. 2002.

[85] Y. Cao, *et al.*, "Effective on-chip inductance modeling for multiple signal lines and application to repeater insertion," *IEEE Trans. Very Large Scale Integration Systems*, vol. 10, pp. 799-805, Dec. 2002.

[86] G. Hinton, *et al.*, "The microarchitecture of the Pentium 4 processor," *Intel Technology Journal,* pp. 1-13, Q1 2001.

[87] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE,* vol. 91, pp. 305-327, Feb. 2003.

[88] R. Rao, *et al.*, "Statistical analysis of subthreshold leakage current for VLSI circuits," *IEEE Trans. Very Large Scale Integration Systems*, vol. 12, pp. 131-139, Feb. 2004.

[89] S. Mukhopadhyay, *et al.*, "Gate leakage reduction for scaled devices using transistor stacking," *IEEE Trans. Very Large Scale Integration Systems*, vol. 11, pp. 716-730, Aug. 2003.

[90] Design Compiler Reference Manual, Synopsys, 2000.

[91] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinatorial benchmark circuits," *Proc. Int. Symp. Circuits and Systems*, 1985, pp. 695–698.

[92] D. Lee, V. Zolotov, and D. Blaauw, "Static timing analysis using backward signal propagation," *Proc. Design Automation Conf.,* 2004, pp. 664-669.

[93] W.-Y. Chen, S.K. Guota, and M.A.Breuer, "Analytical models for crosstalk excitation and propagation in VLSI circuits," *IEEE Trans. Computer-Aided Design,* vol. 21, pp. 1117-1131, Oct. 2002.

[94] S.-Z. Sun, D.H.C. Du, and H.-C. Chen, "Efficient timing analysis for CMOS circuits considering data dependent delays," *IEEE Trans. Computer-Aided Design,* vol. 17, pp. 546-552, June 1998.

[95] A. Pierzynska and S. Pilarski, "Pitfalls in delay fault testing," *IEEE Trans. Computer-Aided Design,* vol. 16, pp. 321-329, March 1997.

[96] C.T. Gray, *et al.,* "Circuit delay calculation considering data dependent delays," *Integration, the VLSI Journal,* pp. 1-23, 1994.

[97] M. L. Rieger, *et al.*, "OPC strategies to minimize mask cost and writing time," *Proc. SPIE,* vol. 4562, 2002, pp. 154-160.

[98] K. Fuchs, F. Fink, and M.H. Schulz, "DYNAMITE: An efficient automatic test pattern generation system for path delay faults," *IEEE Trans. Computer-Aided Design,* vol. 10, pp. 1323-1335, Oct. 1991.

[99] G. S. Hwang and K. P. Giapis, "The influence of electron temperature on pattern-dependent charging during etching in high-density plasma," *Journal of Applied Physics,* vol. 81, pp. 3433-3439, 1997.

[100] H.-Y. Song, *et al.,* "Timing analysis for full-custom circuits using symbolic DC formulations," *IEEE Trans. Computer-Aided Design,* vol. 25, pp. 1815-1830, 2006.

[101] S. E. Thomson, *et al.,* "A 90-nm logic technology featuring strained-silicon," *IEEE Trans. Electron Devices,* vol. 51, pp. 1790-1797, 2004.

[102] G. Eneman, *et al.,* "Layout impact on the performance of a locally strained PMOSFET," *Proc. Int. Symp. on VLSI Technology,* 2005, pp. 22-23.

[103] X. Chen, *et al.,* "Stress proximity technique for performance improvement with dual stress liner at 45nm technology and beyond," *Proc. Int. Symp. on VLSI Technology,* 2006, pp. 60-61.

[104] C. Ortolland, *et al.,* "Stress memorization technique (SMT) optimization for 45nm CMOS," *Proc. Int. Symp. on VLSI Technology,* 2006, pp. 78-79.

[105] V. Moroz, *et al.,* "Stress-aware design methodology," *Proc. ISQED,* 2006, pp. 701-706.

VITA

Munkang Choi received B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, in 1992 and 1995, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering at Georgia Institute of Technology, Atlanta, GA. From 1998 through 2001, he was with Samsung Electronics Co., Ltd., Kyungki-Do, Korea, where he engaged in process and transistor engineering using TCAD. Now, he is with TCAD DFM Group, Synopsys, Mountain View, CA, developing SEISMOS (stress proximity analysis tool). His research interests include design for manufacturability for nano-scale CMOS generations, strain engineering, path delay fault diagnosis, and CMOS gate compact modeling.