Yield Analysis and Optimization

Puneet Gupta
Blaze DFM Inc.,
Sunnyvale, CA, USA
puneet@blaze-dfm.com

Evanthia Papadopoulou
IBM TJ Watson Research Center
Yorktown Heights, NY, USA
evanthia@watson.ibm.com

In this chapter, we are going to discuss yield loss mechanisms, yield analysis and common physical design methods to improve yield. Yield is defined as the ratio of the number of products that can be sold to the number of products that can be manufactured. Estimated typical cost of modern 300mm or 12inch wafer 0.13 μm process fabrication plant is $2-4 billion. Typical number of processing steps for a modern integrated circuit is more than 150. Typical production cycle-time is over 6 weeks. Individual wafers cost multiple thousands of dollars. Given such huge investments, consistent high yield is necessary for faster time to profit.

1 Introduction

Total yield for an integrated circuit Y_{total} can be expressed as follows [3].

$$Y_{total} = Y_{line} \times Y_{batch} \tag{1}$$

Here Y_{line} denotes line yield or wafer yield which is the fraction of wafers which survive through the manufacturing line. Y_{batch} is the fraction of integrated circuits which on each wafer which are fully functional at the end of the line. Steep yield ramp means quicker path to high batch yield and hence volume production. Earlier volume production means higher profitability for the semiconductor manufacturer in today's market with time-to-market pressures.

Y_{batch} can be further classified based on either type of defect or of failure. Failure-type taxonomy is as follows.

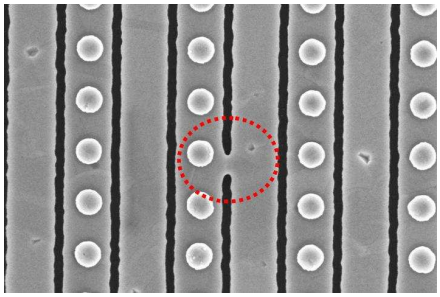

Figure 1: An SEM (Scanning Electron Microscope) picture showing a bridging fault on Metal 3. Note the row of vias on each metal line.

- *Catastrophic Yield Loss.* These are functional failures such as open or short circuits which cause the part to not work at all. Extra or missing material particle defects are the primary causes for such failures. A nice picture of a bridging fault is shown in Figure 1 drawn from [63]. Critical area analysis is used to predict this type of yield loss and is discussed later in this chapter.

- *Parametric Yield Loss.* Here the chip is functionally correct but it fails to meet some power or performance criteria. Parametric failures are caused by variation in one or set of circuit parameters, such that their specific distribution in a design makes it fall out of specifications. For example, parts may function at certain VDD, but not over whole required range. Another example source of parametric yield loss is leakage in deep sub-micron technologies [28]. Parametric failures may be caused by process variations. Several kinds of integrated circuits are *speed-binned* (i.e. grouped by performance). A common example of such class of designs is microprocessors wherein lower performance parts are priced lower. The other class is typical ASICs which cannot be sold if the performance is below a certain threshold (for example due to compliance with standards). In the latter case, there can be significant performance-limited yield loss which is why such circuits are designed with a large guardband. In the former case too, there can be significant dollar value loss even if there is little yield loss.

Additionally, there is also testing-related yield loss as no testing process can detect all possible faults (and potential faults). Such yield-loss is related to defect level (e.g., see [39]) and field returns (e.g. see [57]). We will not include such yield-losses in our discussion as

they are not physical design related. Another aspect of field-returns is long-term reliability of designs (e.g. see [21]). Reliability is typically treated as a separate topic and we would discuss yield-loss only in its most common definition: number of bad parts at the end of manufacturing line.

Defect types can be classified as follows [1].

- *Random Defects.* These are randomly distributed faults such as particle contamination.
- *Systematic Defects.* These kind of defects are predictable. Example sources include CMP (Chemical Mechanical Polishing) and photoresist pattern collapse.

It is important to understand that both random and systematic defects can cause parametric or catastrophic yield loss. For example, lithographic variation which is typically systematic and pattern dependent can cause catastrophic line-end shortening leading gate (polysilicon over diffusion) not forming and hence a functional failure. A less drastic rendition of lithographic variation is gate-length variation causing gates on critical paths to speed up too much leading to hold-time violations under certain voltage and temperature conditions. Various defect types and failure modes are shown in Figure 2. Systematic mechanism limited yield loss is projected to be the dominant source of yield loss in current and future technology generations [3].

Decision at the IC manufacturing site of which parts are not working and should be discarded is an important one. Though this is more a discussion of testing and testability, a very brief introduction is essential to understand final yield measurement at the foundry. For a more detailed discussion, see [4]. Tests are usually classified as delay tests (intended usually to test the parametric failures) and functional tests (intended usually to test for catastrophic failures). Two common examples of test are FMAX testing and IDDQ testing. FMAX tests essentially keep increasing the clock frequency till a failure is detected. This is to determine the maximum frequency of operation of the circuit. IDDQ tests measure the quiescent current in the power supply after bringing the circuit to a known state. Such tests can help detect (for example) bridging faults.

[1] Similar taxonomy is typically used for process variations as well. The terms defects and variations are used interchangeably in literature. One common distinction between the two terms is the exclusion of the particle defects from variations.

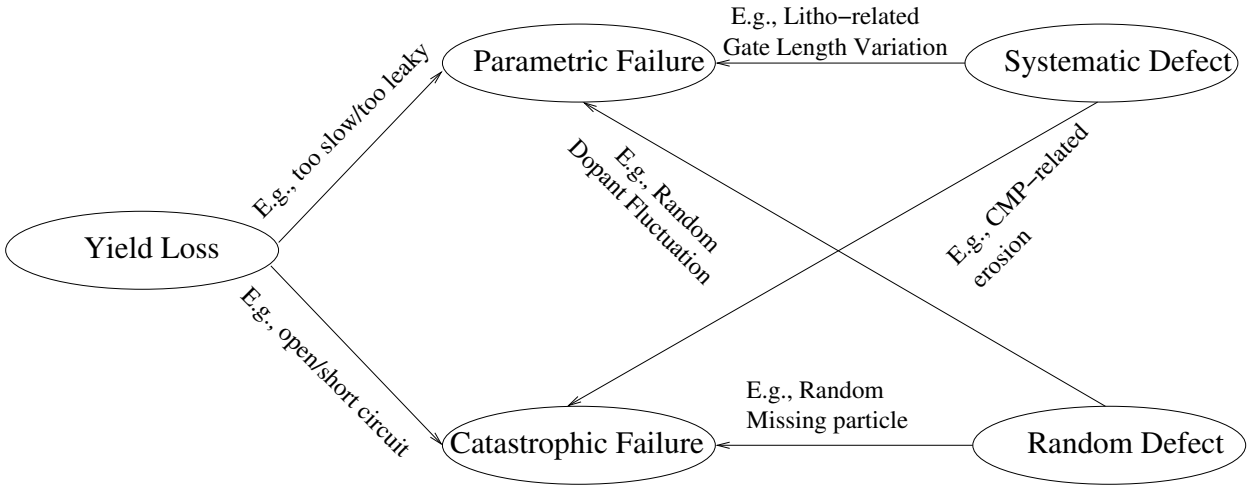

Figure 2: Sources and types of yield loss. Note that either type of failure can be caused by either type of defect.

A necessary component of the yield improvement and process ramp-up process is root cause analysis of failures. Failure analysis attempts to determine both the failure mechanism and the underlying cause. Modern failure analysis labs have several advanced techniques at their disposal. For example, with focused ion beam (FIB), existing circuit lines can be cut and new lines inserted for mechanical or electron beam probing. In some cases transmission electron microscope (TEM) may be used to provide atomic resolution images of structures.

Inline process monitoring is another common way to make sure that the fabrication line is running fine. It is also used for characterization purposes (e.g. to characterize process variation). The most common way to accomplish this is to place and measure simple test structures such as ring oscillators in the scribe-line area of the wafer (i.e. the empty area on the wafer between functional chips). Such measurements are done by wafer-level probing and do not require dicing and packaging of the structures. In addition, scanning electron microscope (SEM) measurements of critical dimension (CD)[2].

2 Sources of Yield Loss

As mentioned earlier in the chapter, yield loss can be due to systematic as well as random defects. Contamination related spot defects are discussed later in the chapter. In this section

[2]CD is the commonly used term for the smallest (and hence the most critical) linewidth in the design.

we focus our attention to variations. There are several ways to classify variations depending on the axis:

- Process vs. Environmental. Variation occurring during circuit operation (e.g. temperature, power supply, etc) are environmental in nature while those occurring during the manufacturing process (e.g. mask misalignment, stepper focus, etc) are physical. We will focus only on process variations.

- Systematic vs. Random. As discussed earlier systematic variations (e.g. metal dishing, lithographic proximity effects, etc) can be modeled and predicted while random variations (e.g. material variations, dopant fluctuations, etc) are inherently unpredictable.

- Inter-die vs. Intra-die. Depending on the spatial scale of the variation, it can be classified as die-to-die (e.g. material variations) or within-die (e.g. layout pattern dependent lithographic variation). Inter-die variations correspond to variation of a parameter value across nominally identical die. Such variations may be die-to-die, wafer-to-wafer or even lot-to-lot. Inter-die variations are typically accounted for in design, by shift in the mean of a parameter value. Intra-die variations on the other hand correspond to parameter fluctuations across nominally identical circuit elements such as transistors. Intra-die perturbations are usually accounted in design by guardbanding and prevention. Variation compensation in design is further discussed in the next section.

An interesting point to note here is the level of abstraction for sources of variation. For logic designers, variation may be caused by cell delay or transistor delay changes. Such modeling is evident, for example, in most statistical timing analysis tools (e.g. [51, 5, 72]. For circuit designers, the level of abstraction may go down to (say) transistor gate-length variation which leads to cell or transistor delay variation. Going further down, a lithographer may attribute critical dimension (CD) variation to focus variation which may be further blamed on wafer flatness imperfections.

Variation in process conditions can manifest itself as dimensional variations or material variations. Dimensional variations include the following.

- Lateral dimension variation. Across chip linewidth variation or ACLV is one of the biggest contributors to parametric variation. In this category important causes of parametric and functional failure are gate-length variation, line-end pullback and contact or via overlap. Lithography and etch processes are the biggest culprits for such variations. Such variations are largely systematic and layout pattern dependent.[3] With scaling geometries, even small variations in dimensions can be detrimental to circuit performance. For example line edge roughness (LER) is projected to be a big concern for 32nm device performance [2, 43].

- Topography variation. Dielectric erosion and metal dishing caused by chemical mechanical polishing (CMP) processes is one of the biggest contributors to interconnect failures. In front-end of the line (FEOL), imperfect STI (Shallow Trench Isolation) CMP process is an example cause of topographic variation. Topographic variation not only results in interconnect resistance and capacitance variation but by virtue of acting as defocus for lithographic manufacturing of subsequent layers resulting in linewidth variation [32].

Several processing steps during the manufacture of deep sub-micron integrated circuits can result in material parameter perturbations. Besides material purity variations, such variations can be caused for example by perturbations in implantation or deposition processes. An important example of material variation is discrete dopant fluctuation. Random placement of atoms at discrete location in the channel can cause V_{th} variation. With number of dopant atoms going down to few hundred in sub-100nm devices, random dopant fluctuation is becoming an important source of variation.

The result of these physical variations is variation in circuit metrics like performance and power. International Technology Roadmap for Semiconductors (ITRS) projects as much as 15% slow-down in design signoff delay by the year 2014. Leakage and leakage variability is an even bigger problem due to exponential dependence of leakage power on physical dimensions like gate-oxide thickness and gate length as well material properties like dopant concentration. 30X variation in leakage in microprocessor has been noted by [13]. According

[3]Lateral dimension variation is typically mitigated on the manufacturing side by resolution enhancement techniques (RETs) such optical proximity correction (OPC).

to ITRS projections, containing V_{th} variability to within 58%, circuit performance variability to within 57% and circuit power variability to within 59% is a "red-brick" (i.e. no known solutions). On the BEOL (Back End of the Line) side, varying electrical parameters include via resistance as well as wire resistance and capacitance.

In this section, we have barely touched upon various sources of yield loss. A very good discussion of process variations can be found in [12].

3 Yield Analysis

The yield of a VLSI chip depends on its parametric as well as functional sensitivity to the various kinds of defects discussed earlier. Yield prediction requires modeling of various complicated physical and statistical phenomena. The yield analysis problem can be decomposed into analysis of (1) parametric and (2) catastrophic failures. Yield analysis of catastrophic failures is discussed at length in Section 3.2. A very brief introduction to parametric yield analysis is presented next.

3.1 Parametric Yield Analysis

Analysis of chip failures and consequent yield loss is an active area of research and there is little consensus on yield metrics and calculation methods in this regime. In recent years, statistical timing analysis methods which help predict parametric yield loss due to timing failures have received a lot of attention [71, 72, 17, 5, 51]. Other statistical methods have focused on power-limited yield as well [61, 64]. Several other methods that concentrate on the systematic component of variation have also been proposed [18, 52, 29, 79]. Statistical analysis methods can be characterized either as *performance-space* (directly modeling distributions of gate or interconnect delays) or *parameter space* (modeling distributions of sources of performance variations such as gate length, threshold voltage with performance variables modeled as functions of basic parameters). *Block-based* analysis tools propagate these variability distributions through circuit timing graph [4] to calculate arrival time and required time distributions and consequent slack distributions at all circuit nodes. *Path-based* methods work on a set of critical paths instead of the full design and as a result are

[4]The key operations in such propagation are sum, min and max of random variables.

better equipped to handle arbitrary distributions and correlations using Monte Carlo simulations. Correlations: spatial, logical or otherwise play an important role in such statistical timing analysis. From a foundry perspective, it is very difficult to characterize the process to identify all sources of variation and their magnitude, compute correlations between these sources and also find out the spatial scale to which they extend. To add to the complexity, most of these sources of variation have very systematic interactions with layout and cannot be easily split into inter- and intra-die components. Nevertheless, with the magnitude and sources of variability increasing, statistical power and performance analysis coupled with accurate modeling of systematic variations will lead to parametric yield analysis to be part of standard design sign-off.

3.2 Random defect yield modeling and critical area computation

A number of models for the prediction of yield of a semiconductor device due to random manufacturing defects have been proposed over the years. The common focus of all models is a measure called *critical area* that represents the sensitivity of a VLSI design to random defects during the manufacturing process.

A majority of random defects is introduced into the IC layer by the lithography process. These are spot defects caused by various contamination particles. Spot defects may result in circuit failure depending on their size and location. They are classified into *extra-material* defects (also referred to as bridges or protrusion defects) and *missing material* defects (also called voids, notches or intrusion defects). Extra-material defects result in shorts between different conducting regions. Missing-material defects result in open circuits. Missing material defects that result in broken (open) conducting paths or destroyed contacting regions are called *opens* or *breaks*. Missing material defects on contact and via layers that destroy contacts and vias are called *via blocks*. Another class of defects, known as pinholes, occur in dielectric insulators. Pinholes are tiny defects that may cause shorts if located in the overlap region between patterns at different photolithographic levels (see e.g. [66]). Shorts, opens (breaks), and via-blocks are the main types of random manufacturing defects resulting in circuit failure.

The yield of a chip considering random manufacturing defects is computed as

$$Y = \Pi_{i=1}^{m} Y_i$$

where Y_i is the random defect yield associated with the ith step of the manufacturing process (see e.g. [73, 47, 26]). For convenience the subscript is omitted and Y is referred as the yield of a single processing step. There is a number of models to compute random defect yield such as Seed's, the Poisson, the negative binomial, and Murphy's model (see e.g. [25]). The main difference between the various yield models is in the choice of statistics that are assumed to govern the spatial distribution of defects. For example, choosing negative binomial statistics results in the wide spread negative binomial yield model which is given by the following equation for a single processing step:

$$Y = \left(1 + \frac{dA_c}{\alpha}\right)^{-\alpha}$$

where d denotes the average number of defects per unit of area, α is a clustering parameter, and A_c denotes the *critical area*. All yield models, independent of the statistics used, result in yield equations that are functional forms of the same quantity termed critical area.

Critical area is a measure reflecting the sensitivity of the design to random manufacturing defects and is defined as follows

$$A_c = \int_0^\infty A(r)D(r)dr$$

where $A(r)$ denotes the area in which the center of a defect of radius r must fall in order to cause circuit failure and $D(r)$ is the density function of the defect size. $A(r)$ is referred to as the *critical area for defect size* r. The total critical area integral A_c for all defect sizes is also referred to as the *weighted critical area*. The defect density function has been estimated as follows[73, 27, 66]:

$$D(r) = \begin{cases} cr^q/r_0^{q+1}, & 0 \le r \le r_0 \\ cr_0^{p-1}/r^p, & r_0 \le r \le \infty \end{cases} \tag{2}$$

where p, q are real numbers (typically $p = 3, q = 1$), $c = (q+1)(p-1)/(q+p)$, and r_0 is some minimum optically resolvable size. Figure 3 illustrates $A(r)$, the critical area for shorts for the given defect of size r. Note that the illustrated defect causes a short if and only if its center falls anywhere within the shaded area $A(r)$. The extraction of critical area requires

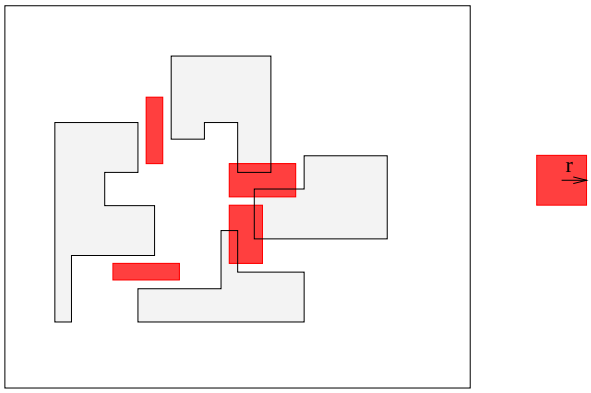

Figure 3: Critical area $A(r)$ for a given defect of size r.

further the computation of the total critical area integral for all defect sizes given the defect size distribution $D(r)$.

The extraction of critical area for various types of faults poses the major computational bottleneck in VLSI random yield prediction. In the following we review the main computational paradigms proposed in the literature for the extraction of critical area and focus on their algorithmic aspects. Pinhole defects are ignored because extracting their critical area is straightforward. Pinhole defects are modeled as points of no area and their critical area is simply the area of overlap between patterns at different photolithographic levels (see e.g. [66]). We first give some mathematical insight on various defect models.

3.2.1 Defect models

Random manufacturing defects are typically modeled as circular disks with a size distribution. In particular, a defect of size r is modeled as a circle of radius (or diagonal) r. The radius r is a random variable with a known probability density function $D(r)$ as given above. When calculating critical area it has been a normal practice to approximate the circular defect by shapes that are computationally easier to deal with such as squares or regular k-gons for an even k, usually $k = 8$. Figure 4(b)-(d) depicts defect shapes commonly used in practice instead of the ordinary circular defect depicted in Figure 4(a). This common practice has a mathematical interpretation that can facilitate the derivation of error bounds.

Modeling defects by any convex shape corresponds to measuring distance for critical area under a corresponding convex distance function. For example the circular defect corresponds

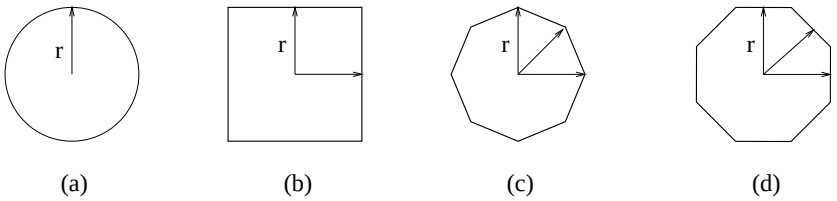

Figure 4: Common defect shapes of size r.

to measuring distance in the ordinary Euclidean way, where the Euclidean distance between two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ is $d_e(p, q) = \sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}$. The square defect model corresponds to computing distances in the L_∞ metric, where the L_∞ distance between two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ is $d_\infty(p, q) = \max\{|x_p - x_q|, |y_p - y_q|\}$. Computing distances in the L_1 (Manhattan) metric, where $d_1(p, q) = |x_p - x_q| + |y_p - y_q|$, corresponds to a square diamond defect (a square rotated by 45°). The *k-gon distance* between two points p, q is the *size* of the smallest k-gon P touching p and q, where the size can be naturally defined either in terms of the *diameter*[5] or the *width*[6] of P. Depending on whether the diameter or the width of the k-gon is used to define size, the k-gon metric can be regarded as a generalization of the L_1 or the L_∞ metric respectively in $k/2$ directions. For example the distance functions implied by Figure 4(c) and Figure 4(d) can be defined as generalizations of the L_1 and the L_∞ metric respectively. Critical area computed under these metrics can serve as an upper bound to the Euclidean critical area of circular defects e.g., the L_∞ metric or the metric of Figure 4(d), or as a lower bound e.g., the L_1 metric and the metric of Figure 4(c). A worst case bound of 2 for critical area between square and circular defects is given in [53]. Often in practice the purpose of critical area computation is the prediction of relative yield. That is, predict how a new design will yield in comparison to existing designs by comparing their critical areas. In this respect, the consistency of the defect model is far more important than the actual numerical values. In the following we review the main computational paradigms that have been proposed for the extraction of critical area.

[5]The radius of a regular k-gon P is the distance from the center of P to any of its vertices. The diameter is twice the radius.

[6]The width of a regular k-gon, for an even k, is the distance between two parallel edges of P.

3.2.2 Statistical methods

There have been two types of statistical methods proposed for critical area estimation at the chip level: *Monte Carlo simulation* [75, 67] and *layout sampling* [6]. Monte Carlo simulation has been the oldest and the most widely implemented technique. The method is as follows: Draw a large number of defects with their radii distributed according to $D(r)$, check for each defect if it causes a fault, and divide the number of defects causing faults with the total number of defects. The main advantage of the Monte Carlo method is simplicity and the ease to conceptualize the procedure necessary to detect almost any type of failure mechanism. A tool can potentially be constructed upon most Design Rule Checking (DRC) Platforms. The method is computationally intensive. A naive implementation could result in prohibitively long runtimes. However, adaptive sampling and adaptive integration techniques can greatly benefit the basic method. Distributed processing can significantly improve performance further as reported in [74]. Despite potential inefficiencies the Monte Carlo simulation method is widely accepted and it provides a good standard for comparison.

Layout sampling, in combination with a deterministic method to compute critical area over a layout window, can give an alternative statistical technique to estimate the critical area of an entire chip. Layout random sampling in combination with standard *shape shifting* techniques to compute critical area were introduced in [6]. The method is as follows: Randomly sample the layout to obtain random sample windows, compute critical area in every sample using a deterministic method, combine results on sample windows to obtain a critical area estimate for the entire chip. Stratified sampling can increase the accuracy of the prediction by dividing the layout area into a number of regions (strata) for which critical area is estimated using sampling techniques. The performance as well as the accuracy of the sampling methodology depends heavily on the method to generate samples as well as on the method to compute critical area in a selected sample. Details of the method as implemented in the EYES system combining stratified random sampling and shape shifting techniques are given in [6].

3.2.3 Deterministic iterative methods

Critical area estimation methods in this category iteratively compute $A(r)$, the critical area for a specific defect size r, for several different values or r. Those values are then interpolated with the defect density function $D(r)$ to approximate the total critical area integral. The majority of these methods are based on *shape shifting* techniques (see e.g. [9, 16, 38]). For shorts the typical shape shifting method to compute $A(r)$ can be described as follows:

1. Expand each geometry shape by $r/2$.
2. Compute the region of overlap among two or more shapes in different nets.
3. Let the area of the resulting region be $A(r)$.

The main advantage of the shape-shifting methods is the ease of implementation using widely available shape-manipulation functions through most DRC platforms. The disadvantage is that step 2 can be expensive demonstrating quadratic behavior even for smart scanline type of implementations. The reason is that the number of intersections between the expanded polygons, denoted as I, can be $\Omega(n^2)$, where n is the number of edges of the original shapes, especially for medium or large defect radii r. Even in the case where $O(N \log N)$ type of efficient scanline algorithms are used to compute the area of overlap (see e.g. [9, 16]), N has a quadratic flavor because $N = \Omega(I + n)$ and $I = \Omega(n^2)$. As a result, shape-shifting methods work well for small to medium values of r, however they break down when trying to compute the entire critical area integral for a sizable layout because of the quadratic time required to compute $A(r)$ for medium or large values of r. The layout hierarchy can substantially speed up the computation of $A(r)$ (see e.g.[50]) for regular designs. Hierarchical processing however becomes less useful as the value of the defect radius r increases providing no benefit beyond a certain threshold.

For Manhattan layouts (i.e., layouts consisting of axis parallel shapes only) and square defects there are additional more efficient methods in this category that require an initial decomposition of shapes into rectangles [80, 23]. [23] is a scanline approach that first computes *susceptible sites* (rectangular regions defined by the original layout shapes that provide defect susceptibility information) then manipulates (shrink/expand) those susceptible sites to compute critical regions for a given defect size r, and finally computes $A(r)$ as the total

area of those critical regions (each critical region is a rectangle). The method in [80] can be regarded as a reverse shape shifting method that first computes $A(r_{max})$ for the maximum defect size r_{max} and then iteratively determines $A(r)$ for smaller radii r. The method can be summarized as follows: 1) Compute all Maximum-Critical-Area rectangles (called Max-CARs) by expanding all layer rectangles by r_{max} and determining all overlapping pairs of rectangles such that the two rectangles belong to different nets. Max-CARs are collected into buckets, one bucket per net pair. 2) Given a defect radius r, the Critical Area rectangles for r (called CARs) in each bucket are readily available by shrinking the Max-Cars by $\Delta r = r_{max} - r$. 3) Let $A(r)$ be the total area of all computed CARs for defect of radius r. Computing the Max-Cars of step 1 is a rather expensive operation performed only once. The area of N rectangles can be efficiently computed in $O(N \log N)$ time using interval trees (see e.g. [22]) and thus, once the Max-CARs are available, $A(r)$ can be computed efficiently for a number of radii r. The number N however, of Max-CARs or $CARs$ can be large i.e., $\Omega(n^2)$, where n is the number of layout edges. Clearly the number of CARs reduces as the defect radius r decreases and thus performance depends on the size of r_{max}.

Most of the investigation on critical area extraction in this category has been done for shorts. Opens have been studied less and are often approximated as a dual problem, substituting shape-expansion by shape-shrinking enhanced with other shape manipulation operations to derive critical regions. The reader is referred to [58] for details on such a treatment of opens based on DRC operations. Via-blocks are also treated in [58] for a simplified definition where a defect is considered a fault if it simply overlaps any portion of a contact or via. For a discussion on the sensitivity of via chains to metal opens and via-opens see [24].

3.2.4 The Voronoi (non-iterative) deterministic method

This method, originally proposed in [55, 53], computes the entire critical area integral in $O(n \log n)$ time, where n is the number of layout edges, in a single pass of the layout. It is based on the following concept: Divide the area of the layout into small regions such that critical area within each region is easy to compute. The total critical area integral can be derived as the summation of all partial critical areas obtained within each region. Assuming that within each layout region the critical area integral can be computed accurately, the total critical area of the layout can be easily extracted once the appropriate layout subdivision

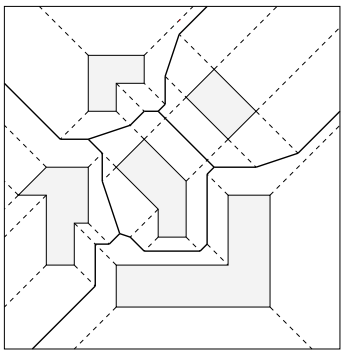

Figure 5: The L_∞ Voronoi diagram of polygons.

is available. The well known concept of a *Voronoi diagram* can help obtain the appropriate layout subdivision needed for each type of fault.

The Voronoi diagram of a set of polygonal sites is a partitioning of the plane into regions, one for each site, called *Voronoi regions*, such that the Voronoi region of a site s is the locus of points closer to s than to any other site. The Voronoi region of s is denoted as $reg(s)$ where s is the *owner* of $reg(s)$. The boundary that borders two Voronoi cells is called a *Voronoi edge*, and consists of portions of *bisectors* between the owners of the neighboring cells. The bisector of two polygonal objects (such as points, or segments) is the locus of points equidistant from the two objects. The point where three or more Voronoi edges meet is called a *Voronoi vertex*. Figure 5 illustrates the Voronoi diagram of polygons under the L_∞ distance metric. The Voronoi diagram can be regarded as an encoding of nearest neighbor information. The combinatorial complexity of the Voronoi diagram is linear in the number of the original sites.

The *critical radius* of a layout point t, denoted $r_c(t)$, is the radius of the smallest defect centered at t causing a circuit fault. Given a layer of interest C, and a fault type, the Voronoi method subdivides C into regions such that for any point t the critical radius is easy to compute. In particular, $r_c(t)$ is given by the distance of t from the layout element owning the region where t belongs. In the L_∞ metric (similarly for L_1 and the octagon metric) $r_c(t)$ becomes a simple linear function allowing for simple critical area integration. In the following we indicate the Voronoi diagram for shorts and refer the reader to [53, 54] for the case of opens and via-blocks. The L_∞ metric is assumed throughout the section. The concepts are easily extendible to the octagon metric with some penalty in the complexity of

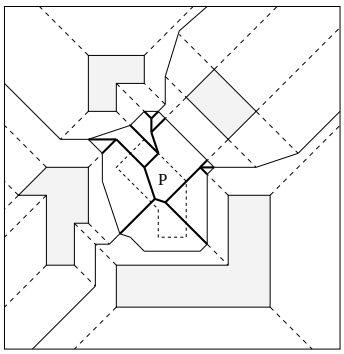

Figure 6: The 2nd order L_∞ Voronoi diagram in $reg(P)$.

the Voronoi diagram construction (see e.g. [19] for k-gons). For circular defects no analytical formulation for critical area integration is known.

Voronoi diagram for shorts A short at a layout point t is a defect centered at t overlapping with at least two shapes in two different nets. Let P to be the polygon *nearest* to t. The critical radius of t is determined by the *2nd nearest* polygon to t, say Q, such that Q is in a different net than P, and $r_c(t) = d(t, Q)$. Thus, 2nd nearest neighbor information is needed which can easily be obtained by the *2nd order Voronoi diagram* on the layer of interest defined as follows: For every polygon P partition the interior of $reg(P)$ by the Voronoi diagram of all polygons other than P. In Figure 6, the thick lines illustrate the *2nd order* subdivision of $reg(P)$, where P is shown in dotted lines. Note that only Voronoi neighbors of $reg(P)$ can contribute to the 2nd order subdivision of $reg(P)$. More formally, given a layer C, the 2nd order Voronoi region of an element $s \in C - P$ within the Voronoi cell of P is defined as $reg_P(s) = \{x \mid d(s, x) \leq d(t, x), \forall t \in C - P\}$. For any point $t \in reg_P(s)$, $r_c(t) = d(t, s)$. To avoid counting shorts between disjoint polygons of the same net, any neighboring Voronoi regions of the same net can be united prior to the 2nd order Voronoi computation.

Critical Area Integration Let's assume that the appropriate Voronoi subdivision of a layer for a fault type is available. Each Voronoi region can be partitioned into simple subregions such as rectangles and triangles (assuming the L_∞, L_1 or octagon metric), where the critical area integral can be computed analytically given the defect size distribution $D(r)$. Once analytic formulas are established for each type of simple region, the total critical area

integral can be derived as a simple summation of those formulas. As formulas are analytic there is no integration error. In [55, 56] analytic formulas were derived for the widely used defect size distribution $D(r) = 1/r^3$ assuming the L_∞ metric and were shown to simplify into terms derived directly from Voronoi edges. As a result critical area extraction becomes trivial once the appropriate Voronoi diagram is computed. In case $A(r)$, the critical area for a given specific defect size r, is also needed it can be easily derived in linear time from the same Voronoi subdivision of the layout.

Scanline construction of the Voronoi diagram The Voronoi diagram of a layout can be constructed by a scanline approach as described in [55, 56] for the L_∞ metric. The main advantage of the scanline construction is the low memory requirement for critical area computation. For critical area extraction there is never any need to keep the Voronoi diagram of the entire layout in memory. Instead only a portion of the Voronoi diagram near the scanline is maintained. As soon as the Voronoi cell of a polygon or a net is computed, 2nd order computation and critical area computation within that cell can be performed and the Voronoi cell can be immediately discarded. As a result, the layout can remain in a compact hierarchical form while the scanline incrementally flattens keeping only a small neighborhood of the design flat at a time near the scanline. The time complexity of the scanline algorithm to compute the L_∞ Voronoi diagram is $O(n \log n)$, where n is the number of input layout edges i.e., the size of the layout. The 2nd order Voronoi diagram within the Voronoi cell of a polygon is computed in the same way maintaining the same time complexity. Critical area integration is then easily done in linear time. Thus the entire critical area integral can be computed accurately in one scanline pass of the layout in $O(n \log n)$ time.

Results on the wide use of the Voronoi method to compute critical area and predict yield by IBM Microelectronics are given in [48].

3.2.5 Other non-iterative approaches

The grid method of [73] assumes a fine grid over the layout and uses it to perform critical area integration. The grid resolution can provide a trade off between accuracy and speed. The method computes the critical radius for every grid point and uses this information to compute the critical area integral. The approach is appropriate for an interactive tool and

can be sped up as shown in [55].

FedEx [65] is a fault extractor for shorts. That is, instead of computing critical area, it extracts a list of all two node intra-layer bridges (shorts). It also computes approximate weighted critical area for each bridge, and provides approximate fault locations. As pointed out in [65] FedEx trades accuracy for speed and memory. It assumes Manhattan layouts. FedEx starts with a hierarchical design description, incrementally flattens the layout and writes bridging faults out in a flat manner. For circuit and fault extraction uses a scan line algorithm that first converts polygons into rectangles. Memory consumption is relatively small as only a moving window of geometry is kept, i.e., approximately $O(\sqrt{n})$, where n is the size of the layout (number of rectangles). Bridge fault sites are written flat to the output file. There are several performance similarities between FedEx and the Voronoi method. Both methods start with a hierarchical design using a scanline that only locally sees the layout geometry flat. Memory consumption is relative small as only a neighborhood of the design near the scanline is kept in memory. The 1st order Voronoi diagram of the layout geometry also provides information on same layer two node bridges as obtained by FedEx. FedEx outputs fast an approximate critical area for each bridge and the Voronoi method uses the 2nd order Voronoi diagram to obtain an accurate same layer critical area number maintaining an $O(n \log n)$ worst case performance.

4 Methods for Yield Optimization

Aggressive technology scaling had made process variation control from purely manufacturing perspective very tough. Design-related yield losses have been projected to increase [77] which implies greater cooperation between physical design and process communities is necessary. Yield optimization methods work with the "measure, model and mitigate" flow. Measurements are usually done by targeted test structures which are measured on silicon for physical parameters like linewidth and thickness as well as electrical parameters like sheet resistance and transistor saturation current. A good publication to keep track of for those interested in test-structure design and measurement is ICMTS [1]. Models of process extracted from such test-structure measurements are usually abstracted to simpler models or a set of rules for physical design and verification tools to use. In this section, we will briefly discuss the

evolution of yield optimization physical design techniques.

4.1 Critical Area and Catastrophic Yield Optimization Methods

BEOL yield and manufacturability optimization is a complicated task. Methods for yield improvement vary ranging from critical-area based wire spreading, metal fill, and the development of new rules and optimization for routers. We start with a review of available methods for wire spreading and critical area reduction.

Methods for critical area reduction fall into two broad categories: methods that alter the topology of the layout by attempting critical area optimization at the routing phase and methods used as a postprocessing step that keep the layout topology fixed while attempting to alleviate congestion and increase wire spacing. The two categories can be regarded complementary and both can be incorporated into the design cycle.

In the first category the most representative method is [36], where a general routing cost function is described that takes into account critical area in conjunction with traditional routing objectives. The cost function combines most types of major defects i.e., shorts, opens, number of vias, and pinhole defects. Results verify that taking critical area into account at the routing phase can result in effective critical area reduction and therefore effective optimization for yield. In [44], channel routing is modified to reduce critical area between wire segments. [44] also minimizes the number of vias as their presence increases manufacturability complexity and degrades the yield.

The methods in the second category attempt to redistribute spacing between adjacent wires without changing the layout topology. They are usually based on compaction techniques using the following observation: In a VLSI layout distances between shapes can vary as long as the minimum value imposed by the design rules is met. Slack between two shapes is defined by the difference of the current distance between the two shapes and the minimum distance required by the design rules. Carefully redistributing the slacks can result in a layout with a better yield. Several slack redistribution techniques have been proposed, see [20, 11, 35, 14]. In their majority they are based on principles of layout compaction and are formulated as a one-dimensional layout optimization problem. They start with a constraint graph representation of the layout and perform layout modification for yield in one direction

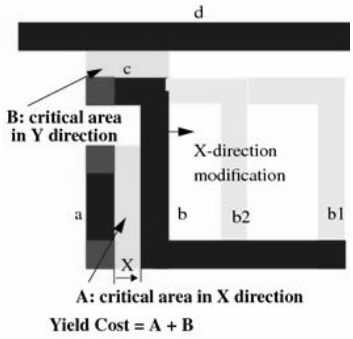

Figure 7: Movement of wire b in x-direction decreases critical area in x-direction but increases critical area in y-direction.

at the time, using in majority a one-dimensional yield objective function. The main drawback of a one-dimensional yield objective function is that, although it optimizes for critical area in one direction, it fails to take into consideration a potential critical area increase in the orthogonal direction. Figure 7, reproduced from [35], illustrates one such situation where movement of a layout element in one direction decreases critical area in one direction but increases critical area in the orthogonal direction. To address this problem [35] combines the one-dimensional movement for slack redistribution with a two-dimensional yield objective.

The first compaction based algorithm to improve yield was given in [20]. A heuristic algorithm increases the spacing of layout objects through a series of spacing iterations in one direction. Only objects off the critical path are allowed to move maintaining the original layout area. The defect sensitivity of open-circuit type faults is reduced by increasing the width of certain non-critical elements in the layout. In [11] the slack redistribution problem in one direction was transformed into a network flow problem which can be solved via the fast wire length minimization algorithm of [70]. The layout is represented by a constraint graph where a node corresponds to a layout object and an edge links the nodes of two adjacent layout objects. The cost of each graph edge is an estimate of the fault probability between the two corresponding objects, expressed as a function of the length of the graph edge, that can be approximated by a convex piece-wise linear cost function. Another one-dimensional compaction based formulation is given in [14] where first the critical area rectangles for one chosen defect size are computed. The standard compaction area optimization objective is en-

hanced with the additional terms of minimizing the critical area rectangles which are reduced into functions of original layout variables. In this manner the original compaction algorithm is upgraded with yield enhancement without introducing extra variables or constraints.

A non-compaction based approach in this category is based on post-route optimization using a rubber-band wiring model [69]. The layout is given in a *rubber-band sketch (RBS)* form which represents every layer of interconnect as a set of flexible rubber-bands with elastic properties. Wire spreading is achieved by estimating the critical area of the RBS and exploiting the flexibility of the rubber band behavior while maintaining wire connectivity. For more information see [69]. Heuristic layout changes to improve yield are described in [8] through the use of a set of local rules for contacts, metal and polysilicon layers. A system that allows the user to first evaluate layout modifications by applying them to samples only of the chip layout, rather than the entire layout, is described in [7]. The results from those samples can be used to define the modifications to be applied to the whole chip.

An effective way to reduce open faults is the introduction of redundant interconnects. Using redundant interconnects the potential for open faults reduces at the cost of increasing the potential for shorts. By trading off the two overall design reliability can increase. The problem was formulated in [40] as a variant of the classic 2-edge connectivity augmentation problem taking into account a wirelength increase budget, Steiner points and routing obstacles. Formulation is as follows: *Manhattan Routing Tree Augmentation (MRTA)* Problem: Given a rectilinear feasible routing region FRR, a rectilinear Steiner routing tree T within FRR, and a wirelength budget W, find a set of augmenting paths A within the FRR such that the total length of augmenting paths is at most W, and the total length of edges of T that are non-bridges in $G = T \cup A$ is maximum. An exact algorithm based on an integer programming formulation, and a greedy heuristic algorithm that iteratively adds an augmenting path between vertices were given in [40]. Experimental results show that the greedy augmentation method achieves significant increase in reliability, as measured by the percentage of biconnected tree edges, with only small increase in wirelength.

In addition to reducing the potential for opens, redundant interconnects have also been proposed in clock networks to overcome the clock skew variation problem. In [60] cross links are inserted to a regular clock tree converting it to a non-tree with lower skew variability

and only a small increase in wire length.

Redundant via insertion provides another effective way of increasing design reliability and yield. Vias have an inherently low reliability (e.g due to stress related via voids) and thus redundant via insertion is a good solution to reduce the yield loss by via failure. Typically redundant via insertion is done post-routing on a "wherever space is available" basis but considering redundant vias in detailed routing also has been proposed [78]. Note that an increased number of vias could have a negative impact in terms of routing area and may reduce critical area for via blocks at the cost of increasing the critical area for shorts. Overall however making appropriate tradeoffs design reliability can increase considerably.

Antenna fixes is another topic for improving design reliability. Since VLSI layers are formed one at a time during fabrication, "dangling" metal1 routes (e.g. nets not yet fully fabricated) connected to the poly gate can cause stray charge deposition on gate damaging it. Methods to correct such situations include inserting *jumpers* in routes such that the maximum dangling route length is limited (see for example [68]). Diffusions diodes can also be inserted to provide a discharge path if space is available.

4.2 Design Rules

Abstraction of manufacturing constraints into a set geometric of constraints or design rules for the layout designers to follow have traditionally been foundry's main method to ensure high probability of correct fabrication of integrated circuits. Typical design rules are constraints on width, spacing or pattern density. Origins of design rules lie various manufacturing steps such as lithography, etch, implant, CMP, etc. Other factors influencing design rule values include preserving scaling, area overhead, layout migratability [7] and ability of design tools and flows to handle them.

Manufacturability implications of technology scaling have led to three major trends in design rules:

- More complicated rule sets. The sheer number of design rules has been growing at a rapid pace with every technology generation. More process constraints have required

[7] Automatic migration of layouts from one technology generation to next is an important concern, especially for custom layouts.

new kinds of rules [41, 59]. This has made physical verification, routing as well as custom layout very difficult and time consuming tasks.

- Restrictive design rules. To cope with sub-100nm manufacturability concerns where manufacturing equipment is not keeping pace with feature scaling, radically restraining layout options has been proposed as a viable option [46, 45]. One common restriction is to enforce regularity in layout which aids printability. An example of such a rule is allowing only one or two pitched on the polysilicon layer.

- DFM rules. Most 90nm and 65nm design rule manuals include a separate set of non-minimum design rules. These design rules if obeyed by the layout, enhance its manufacturability. For example, the minimum metal-via enclosure can be 20nm while the corresponding DFM rule can be 30nm. The increased enclosure can reduce chances of loss of contact between metal route and via at the cost of increased routing area.

Though design rules have served the industry well in the past as the abstraction layer, inadequacy and sub-optimality of such yes/no rules has led to a slow but steady adoption of model-based checking methods [59].

4.3 Corner-Based Design Analysis

Traditionally, static timing and power analysis tools have relied on two or more *corners* of process, voltage and temperature or PVT. We are not going to discuss operating variations such as voltage fluctuations and temperature gradients here. Timing corners are typically specified as slow (S), typical (T) or fast (F). Thus, SS represents a process corner with slow PFET and slow NFET behavior. The common performance analysis process corners are (TT, SS, FF, SF, FS). Usually hold time violations are checked at the FF corner and setup time violations are checked at the SS corner. Similarly, interconnect parasitics can also have typical, minimum and maximum values. The rationale for corner-based analyses lies in the fact that ensuring correct operation of the design at the PVT extrema ensures correct operation throughout the process and operation range. This assumption though not strictly correct, usually holds well in practice. Corner-based analysis enables pessimistic but deterministic analysis and optimization of designs. Most modern physical design algorithms rely

on corner based design being acceptable. Sub-100nm process issues (especially variability) have led to the following trends in corner-based design analysis and optimization.

- More corners. As more complicated process effects emerge and as a result of non-monotone dependence of delay on many of the process parameters, the number of PVT corners at which a design needs to be signed off is increasing.
- On Chip Variation (OCV) analysis. To model within-die variation in static timing tools implicitly analyze clock paths and data paths at separate corners [76]. For example, for setup time analysis, the launching clock path may be analyzed at a slow corner while the capturing clock is analyzed at a fast corner and the data path is analyzed at the slow corner. This in essence tries to model the worst-case impact of on chip variation. Additional techniques such as common path pessimism removal (CPPR) which figures out the shared logic between launching and capturing paths to avoid pushing them to different corners, are used to reduce the inherent pessimism in OCV analysis.

Though the runtime overhead of ever-increasing number of corners, the excess pessimism in corner-based analysis and fear of missing some corners in a high process-variability regime has led to an increasing interest in statistical analysis tools, corner-based design deterministic design optimization still remains mainstay of commercial parametric yield optimization.

4.4 Futures of Parametric Yield Optimization

As mentioned earlier, explicit parametric yield analysis and optimization is a relatively new field of research. Several interesting published works in the past few years have attempted to deal with the problem of manufacturing variability.

4.4.1 Methods for Systematic Variability

There are several pattern-dependent process effects which are systematic in nature. These can be compensated for during physical design to aid manufacturability and hence improve yield. The biggest contributors in this bucket are CMP and photolithography. Metal filling and slotting techniques for CMP are discussed elsewhere in the book. Traditionally, design rules have been the method to optimize for systematic variation. Recently more

explicit mitigation of impact of systematic variation on circuit power and performance has been studied. For instance some methods have tried to reduce CD variability by avoiding lithography-induced forbidden pitches during detailed placement [30] or detailed routing [62, 49]. Making circuit more robust to focus variations has been studied in [31, 42].

4.4.2 Statistical Optimization

Just as statistical analyses, statistical physical design is an active area of research with very little in terms of well-accepted methods of optimization. Deterministic physical design tends to generate a wall of slack. As the number of uncorrelated critical paths increase in a design, any of them can "pop up" to being critical and hence be the determinant of circuit delay. As a result, a higher wall of slack can mean a slower circuit delay distribution. Intentional "under-optimization" by assigning a penalty to paths that are close to critical has been suggested as a simple technique to overcome this issue [10]. Another approach in same vein assigns a delay penalty to every gate proportional to its delay variability [15] and uses standard static timing analysis in optimization. Other approaches explicitly rely on a statistical timing engine in a statistical sensitivity [33, 34] or nonlinear programming based optimization [37]. The biggest challenge in statistical physical design besides computational complexity is accurate modeling of physical reality. For example, ignoring parametric or spatial correlations (i.e. assuming independence or perfect correlation between performance or process random variables) can undo any benefit from statistical optimization.

5 Conclusions

In this chapter we have touched upon various sources of manufacturing yield loss in modern sub-micron processes. We have briefly described methods of yield calculation and optimization with emphasis on well-known methods related to random-defect driven yield loss. We have also discussed the emerging area of parametric yield analysis and optimization in physical design.

References

[1] Ieee international conference on microelectronic test structures.

[2] International technology roadmap for semiconductors: Front end processes. `http://public.itrs.net`, 2005.

[3] International technology roadmap for semiconductors: Yield enhancement. `http://public.itrs.net`, 2005.

[4] M. Abramovici, M.A. Breuer, and A.D. Friedman. *Digital Systems Testing and Testable Design*. John Wiley & Sons, 1994.

[5] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *IEEE/ACM ICCAD*, pages 900–907, 2003.

[6] G. A. Allan. Yield prediction by sampling IC layout. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(3):359–371, 2000.

[7] G. A. Allan. Targeted layout modifications for semiconductor yield/reliability enhancement. *IEEE Trans. on Semiconductor Manufacturing*, 17(4):573–581, Nov. 2004.

[8] G. A. Allan and A. J. Walton. A yield improvement technique for IC layout using local design rules. *IEEE Trans. on Computer-Aided Design*, 11(11):1355–1360, Nov. 1992.

[9] G. A. Allan and A.J. Walton. Efficient extra material critical area algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(10):1480–1486, 1999.

[10] X. Bai, C. Visweswariah, P.N. Strenski, and D.J. Hathaway. Uncertainty-aware circuit optimization. In *IEEE/ACM Design Automation Conference*, pages 58–63, 2002.

[11] C. Bamji and E. Malavasi. Enhanced network flow algorithm for yield optimization. In *Proc. 33rd annual Conference on Design Automation*, pages 746–751, 1996.

[12] D. Boning and S. Nassif. *Design of High Performance Microprocessor Circuits*, chapter Models of Process Variations in Device and Interconnect, pages 98–116. Wiley-IEEE Press, 2000.

[13] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *IEEE/ACM Design Automation Conference*, pages 338–342, 2003.

[14] Y. Bourai and C. J. R. Shi. Layout compaction for yield optimization via critical area minimization. In *Proc. Design Automation and Test in Europe*, pages 122–125, 2000.

[15] S. Boyd, S.-J. Kim, D. Patil, and M. Horowitz. A heuristic method for statistical digital circuit sizing. In *SPIE International Symposium on Microlithography*, 2006.

[16] I. Bubel, W. Maly, T. Wass, P. Nag, H. Hartmann, D. Schmitt-Landsiedel, and S. Griep. AFFCA: A tool for critical area analysis with circular defects and lithography deformed layout. In *Proc. IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems 1995*, pages 10–18, 1995.

[17] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using single pert like traversal. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 621–625, 2003.

[18] L. Chen, L. Milor, C. Ouyang, W. Maly, and Y. Peng. Analysis of the impact of proximity correction algorithms on circuit performance. *IEEE Transactions Semiconductor Manufacturing*, 12(3):313–322, 1999.

[19] Zhenming Chen, E. Papadopoulou, and Jinhui Xu. Robustness of k-gon Voronoi diagram construction. *Information Processing Letters*, 97(4):138–145, 2006.

[20] V. K. R. Chiluvuri and I. Koren. Layout synthesis techniques for yield enhancement. *IEEE Trans. on Semiconductor Manufacturing*, 8(2):178–187, May 1995.

[21] C. Constantinescu. Trends and challenges in vlsi circuit reliability. *IEEE Micro*, 23(4):14–19, 2003.

[22] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, 1997.

[23] J. Pineda de Gyvez and C. Di. IC defect sensitivity for footprint-type spot defects. *IEEE Trans. on Computer-Aided Design*, 11(5):638–658, 1992.

[24] D. K. de Vries and P. L. C. Simon. Calibration of open interconnect yield models. In *Proc. 18th IEEE Intermational Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 26–33, 2003.

[25] A. V. Ferris-Prabhu. Role of defect size distribution in yield modeling. *IEEE Transactions on Electron Devices*, 32(9):1727–1736, September 1985.

[26] A. V. Ferris-Prabhu. *Introduction to Semiconductor Device Yield Modeling.* Norwood, MA: Artech House, 1992.

[27] A.V. Ferris-Prabhu. Defect size variations and their effect on the critical area of VLSI devices. *IEEE J. of Solid State Circuits*, 20(4):878–880, Aug. 1985.

[28] Richard Goering. 90-, 65-nm yields prey to leakage. EE Times `http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=172303036`, October 24 2005.

[29] P. Gupta and F.-L. Heng. Toward a systematic-variation aware timing methodology. In *IEEE/ACM Design Automation Conference*, pages 321–326, 2004.

[30] P. Gupta, A. B. Kahng, and C.-H. Park. Enhanced resist and etch cd control by design perturbation. In *25th SPIE BACUS Symposium on Photomask Technology and Management*, 2005.

[31] P. Gupta, A.B. Kahng, Y. Kim, and D. Sylvester. Self-compensating design for focus variation. In *IEEE/ACM Design Automation Conference*, pages 365–368, 2005.

[32] P. Gupta, A.B. Kahng, C.-H. Park, K. Samadi, and X. Xu. Wafer topography-aware optical proximity correction. *IEEE Transactions on Computer-Aided Design*, to appear.

[33] M.R. Guthaus, N. Venkateswaran, C. Visweswariah, and V. Zolotov. Gate sizing using incremental parameterized statistical timing. In *IEEE/ACM ICCAD*, pages 1029–1036, 2005.

[34] M. Hashimoto and H. Onodera. A performance optimization method by gate sizing using statistical static timing analysis. In *IEEE International Sympoium on Physical Design*, 2000.

[35] F. L. Heng and Z. Chen. Vlsi yield enhancement through layout modification. Technical Report ??, IBM, ??

[36] E.P. Huijbregts, H. Xue, and J. A. G. Jess. Routing for reliable manufacturing. *IEEE Trans. on Semiconductor Manufacturing*, pages 188–194, 1995.

[37] E.T.A.F. Jacobs and M.R.C.M. Berkelaar. Gate sizing using a statistical delay model. In *IEEE/ACM Design Automation and Test in Europe*, 2000.

[38] A. L. Jee and F.J. Ferguson. CARAFE: An inductive fault analysis tool for CMOS VLSI circuit. In *Proc. IEEE VLSI Test Symposium*, pages 92–98, 1992.

[39] W.-B Jone and K.S. Tsai. Confidence analysis for defect-level estimation of vlsi random testing. *ACM Transactions on Design Automation of Electronic Systems*, 3(3):389–407, July 1998.

[40] A. B. Kahng, B. Liu, and I. I. Mandoiu. Non-tree routing for reliability and yield improvement. *IEEE Trans. on Comp. Aided Design of Integrated Circuits and Systems*, 23(1):148 – 156, 2004.

[41] A.B. Kahng. Research directions for coevolution of rules and routers , april 2003, pp. . In *ACM/IEEE Intl. Symposium on Physical Design*, pages 122–125, 2003.

[42] A.B. Kahng, S. Muddu, and P. Sharma. Defocus-aware leakage estimation and control. In *IEEE International Symposium on Low Power Electronics and Design*, pages 263–268, 2005.

[43] S.-D. Kim, H. Wada, and J.C.S. Woo. Tcad-based statistical analysis and modeling of gate line-edge roughness effect on nanoscale mos transistor performance and scaling. *IEEE Transactions on Semiconductor Manufacturing*, 17(2):192–200, 2004.

[44] S. Y. Kuo. YOR: A yield optimizing routing algorithm by minimizing critical areas and vias. *IEEE Trans. on Comp. Aided Design of Integrated Circuits and Systems*, 12(9):1303–1311, 1993.

[45] M. Lavin, F.-L. Heng, and G. Northrup. Backend cad flows for restrictive design rules. In *IEEE/ACM International Conference on Computer-Aided Design*, page 739746, 2004.

[46] L. Liebmann, G. Northrop, J. Culp, L. Sigal, A. Barish, and C. Fonseca. Layout optimization at the pinnacle of optical lithography. In *SPIE*, volume 5042, pages 1–14, 2003.

[47] W. Maly. Computer-aided design for VLSI circuit manufacturability. *Proc. IEEE*, pages 356–392, February 1990.

[48] D. N. Maynard and J. D. Hibbeler. Measurement and reduction of critical area using Voronoi diagrams. In *Advanced Semiconductor Manufacturing IEEE Conference and Workshop*, 2005.

[49] J. Mitra, P. Yu, and D.Z. Pan. Radar: Ret-aware detailed routing using fast lithography simulations. In *42nd ACM/IEEE Design Automation Conference*, 2005.

[50] P. K. Nag and W. Maly. Hierarchical extarction of critical area for shorts in very large ICs. In *Proc. IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 19–27, 1995.

[51] M. Orshansky and K. Keutzer. A general probabilistic framework for worst-case timing analysis. In *IEEE/ACM Design Automation Conference*, pages 556–561, 2002.

[52] 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 Transactions on Semiconductor Manufacturing*, 17(1):2–11, 2004.

[53] E. Papadopoulou. Critical area computation for missing material defects in VLSI circuits. *IEEE Transactions on Computer-Aided Design*, 20(5):583–597, 2001.

[54] E. Papadopoulou. The Hausdorff Voronoi diagram of point clusters in the plane. *Algorithmica*, 40:63–82, 2004.

[55] E. Papadopoulou and D. T. Lee. Critical area computation via voronoi diagrams. *IEEE Transactions on Computer-Aided Design*, 1999.

[56] E. Papadopoulou and D.T. Lee. The l_∞ Voronoi diagram of segments and VLSI applications. *International Journal of Computational Geometry and Applications*, 11(5):503–528, 2001.

[57] Stephen Pateras, Jaffer Hussain, and Thomas Martis. Reducing leakage-induced field returns. Whitepaper Logicvision Inc., 2005.

[58] W. A. Pleskacz, C. H. Ouyang, and W. Maly. Extraction of critical areas for opens in large VLSI circuits. *IEEE Trans. on Computer-Aided Design*, 18(2):151–162, 1999.

[59] P. Rabkin. Dfm for advanced technology nodes: Fabless view. Future Fab International. Issue 20. `http://www.future-fab.com`, 20056.

[60] A. Rajaram, J. Hu, and R. Mahapatra. Reducing clock skew variability via cross links. In *Proc. ACM/IEEE Design Automation Conference*, pages 18–23, 2004.

[61] R.R. Rao, A. Devgan, D. Blaauw, and D. Sylvester. Modeling and analysis of parametric yield under power and performance constraints. *IEEE Design & Test*, pages 376–384, July-August 2005.

[62] S.C. Shi, A.K. Wong, and T.-S. Ng. Forbidden-area avoidance with spacing technique for layout optimization. In *SPIE Design and Process Integration for Microelectronic Manufacturing II*, volume 5379, pages 67–75, 2004.

[63] Z.G. Song, S.P. Neo, S.K. Loh, and C.K. Oh. Root cause analyses of metal bridging for copper damascene process. In *International Symposium for Testing and Failure Analysis*, 2005.

[64] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical Analysis and Optimization for VLSI*. Springe, 2005.

[65] Z. Stanojevic and D. M. H. Walker. FedEx – A fast bridging fault extractor. In *Proc. IEEE International Test Conference*, pages 696–703, 2001.

[66] C. H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Develop.*, 27(6):549–557, November 1983.

[67] C. H. Stapper. Modeling of defects in integrated circuit photolithographic patterns. *IBM J. Res. Develop.*, 28(4):461–475, 1984.

[68] B.-Y. Su, Y.-W. Chang, and J. Hu. An optimal jumper insertion algorithm for antenna avoidance/fixing on general routing trees with obstacles. In *IEEE International Symposium on Physical Design*, 2006.

[69] J.Z. Su and W. Dai. Post route optimization for improved yield using a rubber-band wiring model. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pages 700–706, 1997.

[70] R. Varadarajan and G. Lakhani. A wire length minimization algorithm for circuit layout compaction. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 1987.

[71] C. Visweswariah. Death, taxes and failing chips. In *IEEE/ACM Design Automation Conference*, pages 343–347, 2003.

[72] C. Visweswariah, K. Ravindran, K. Kalafala, S.G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *IEEE/ACM Design Automation Conference*, pages 331–336, 2004.

[73] I. A. Wagner and I. Koren. An interactive VLSI CAD tool for yield estimation. *IEEE Trans. on Semiconductor Manufacturing*, 8(2):130–138, 1995.

[74] D. M. H. Walker and D. S. Nydick. DVLASIC: Catastrophic yield simulator in a distributed processing environment. *IEEE Trans. on Computer-Aided Design*, 9(6):655–664, 1990.

[75] H. Walker and S. W. Director. VLASIC: A yield simulator for integrated circuits. *IEEE Trans. on Computer-Aided Design*, 5(4):541–556, 1986.

[76] M. Weber. My head hurts, my timing stinks, and i dont love on-chip variation. In *SNUG, Boston. http://www.siliconlogic.com/pdf/OCVstinks_MattWeber_SLE.pdf*, 2002.

[77] K. Wu, D. Thon, and P. Mayor. Collaborative dfm critical for enabling nanometer design. FSA Fabless Forum `http://www.fsa.org/publications/forum/article.asp?article=0503/wu`, March 2005.

[78] G. Xu, L. Huang, D.Z. Pan, and M. D.-F. Wong. Redundant-via enhanced maze routing for yield improvement. In *IEEE Asia South Pacific Design Automation Conference*, 2005.

[79] J. Yang, L. Capodieci, and D. Sylvester. Advanced timing analysis based on post-opc extraction of critical dimensions. In *IEEE/ACM Design Automation Conference*, pages 359–364, 2005.

[80] S. T. Zachariah and S. Chakravarty. Algorithm to extract two-node bridges. *IEEE Transactions on VLSI Systems*, 11(4):741–744, 2003.