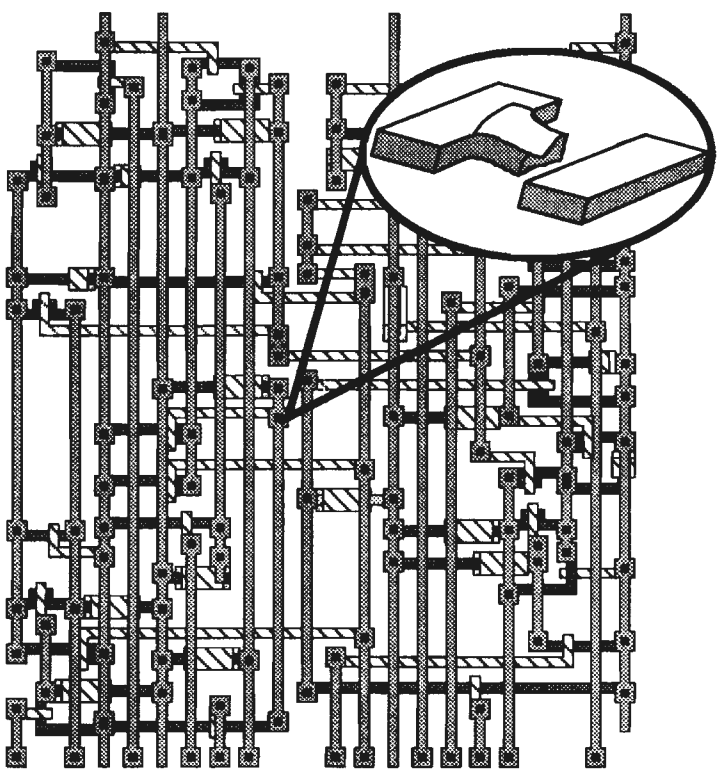
IC defect-sensitivity : theory and computational models for yield prediction

IC DEFECT-SENSITIVITY

THEORY AND COMPUTATIONAL MODELS FOR YIELD PREDICTION



José Pineda de Gyvez

IC DEFECT-SENSITIVITY

THEORY AND COMPUTATIONAL MODELS FOR YIELD PREDICTION

José de Jesús Pineda de Gyvez

IC Defect-Sensitivity

Theory and Computational Models for Yield Prediction

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof. dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op maandag 22 april 1991 om 16.00 uur.

door

José de Jesús Pineda de Gyvez

Geboren te Puebla (Mexico)

druk: wibro dissertatiedrukkerij, helmond.

Dit proefschrift is goedgekeurd door de promotoren

Prof. Dr. -Ing. J.A.G. Jess en
Prof. Dr. Ir. W. van Bokhoven

en door de copromotor

Dr. Ir. Frans Theeuwen

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Pineda de Gyvez, José de Jesús

IC defect-sensitivity: theory and computational models for yield prediction/
José de Jesús Pineda de Gyvez. - [S.l. : s.n]. - Fig., tab.
Proefschrift Eindhoven. - Met lit.opg., reg.
ISBN 90-9003969-4
SISO 663.43 UDC 621.382.049.77:681.3.06(043.3) NUGI 832
Trefw.: geïntegreerde schakelingen; computer aided design.

A Magda, Gloria y Josefina

... mis columnas morales.

Samenvatting

Een belangrijke factor die de opbrengst van correct werkende chips zelfs in volwassen IC fabricage processen bepaald is het optreden van spot defecten. Spot defecten zijn lokale verstoringen van de gelaagde silicium strukturen, veroorzaakt door stofdeeltjes, toleranties in process parameters, onzuiverheden in gebruikte materialen en vervuiling van gereedschappen en werktuigen. Spot defecten kunnen worden beschouwd als willekeurige verschijnselen met een bepaalde stochastische ruimtelijke verdeling op de plak, met een stochastische grootte en veelvuldigheid van optreden per eenheid oppervlak (defect density).

Aangezien de layout patroon resoluties op het IC steeds kleiner worden, zal het negatieve effect van dit soort defecten toenemen. Traditionele layout verificatie concentreert zich op de validatie van ontwerp regels zoals opgelegd door de technologie van het fabricage proces. Er wordt daarbij echter geen aandacht besteed aan de mate van gevoeligheid van het ontwerp ten aanzien van defecten die mogelijkerwijs kunnen optreden in een werkelijke fabricage omgeving. Hiervoor is het nodig om de zogenaamde kritieke gebieden in het ontwerp te bepalen. De kritieke gebieden zijn delen van de layout geometrie alwaar spot defecten van een gegeven afmeting aanleiding kunnen geven tot een foutief gedrag van de schakeling. Bijvoorbeeld kan een spot defect een brug vormen tussen twee naburige patronen en daarmee een deel van de schakeling kortsluiten. Een maatstaf voor de kwetsbaarheid van het ontwerp voor dit soort defecten is bepaald door de ratio van totale kritiek gebied (voor een bepaalde defect grootte) ten opzichte van de totale layout oppervlakte. Deze maatstaf noemen we "defect-sensitivity". We kunnen nu dus met behulp van de kennis omtrent de defect-sensitivity van een bepaald layout ontwerp en de bestudering van de stochastische verdeling van defecten in een bepaalde proces omgeving de totale opbrengst van het IC voorspellen.

Om een goed begrip te verkrijgen van de relaties tussen defecten, het technologisch proces, en het te ontwerpen elektrische circuit, is een formeel semantisch model opgesteld ter modellering van de door het fabricage proces geinduceerde defecten en hun invloed op foutief circuit gedrag. Deze concepten en een overzicht van opbrengst modellering worden in het eerste hoofdstuk van dit proefschrift behandeld.

Hoofdstuk 2 geeft een mathematische analyse van de geometrische eigenschappen van kritieke gebieden en een taxonomie van verschillende defect-sensitivity modellen gebaseerd op de aanwezigheid van enkele of meervoudige fouten, geinduceerd door enkele of meervoudige defecten geplaatst op een of meer lagen.

De tot nu toe bekende methoden voor het vinden van de kritieke gebieden maken gebruik van een grove benadering van de werkelijkheid en leveren dus alleen voor zeer eenvoudige layouts accurate resultaten op, terwijl voor complexe layouts de resultaten te onnauwkeurig zijn. Een nieuwe praktische methode voor het correct bepalen van de kritieke gebieden in enkelvoudige lagen wordt gepresenteerd in hoofdstuk 3; in hoofdstuk 4 wordt deze methode uitgebreid tot meervoudige lagen.

Het moet duidelijk zijn dat kwalitatief gezien de meerlaagsmethode zal leiden tot een meer nauwkeurige voorspelling van de opbrengst in vergelijking tot de eenlaagsmethode. Om dit ook kwantitatief aan te tonen, worden in hoofdstuk 5 beide methoden toegepast op een aantal voorbeeld ontwerpen.

In tegenstelling tot de algemene opvatting dat de berekening van kritieke gebieden zeer rekenintensief is, blijkt uit dit proefschrift dat er een efficiente oplossingsmethode bestaat, die dan ook kan worden toegepast in een interactieve werkomgeving. Daarnaast leveren de ontwikkelde software systemen verdere resultaten die kunnen worden gebruikt voor het analyseren van het fabricage proces, bijdragen in algemene kwaliteitsverbetering, het opstellen van tests vereenvoudigen, en mogelijkheden bieden tot foutafweging.

Summary

In mature manufacturing processes spot defects are the main detractors in the successful outcome of an IC. Their manifestation is as local disturbances of silicon layer structures mainly caused by dust particles, process variabilities, and contaminations of the fabrication equipment. Spot defects are in essence random phenomena occurring on the wafer with certain stochastic spatial distribution and also with a stochastic size and frequency per unit area (defect density).

As the IC pattern resolutions tend to shrink more and more, the effect of spot defects in the layout geometry plays a more important role in yield losses. Traditional approaches for layout verification concentrate on validating design rules imposed by the technological process. However, they do not verify the robustness of the design when it is exposed to defects in a real manufacturing environment. In order to perform the latter verification task it is necessary to capture the design's "critical areas". The so called critical areas are the places in the layout where spot defects can induce an incorrect behavior of the IC. For instance, a spot defect creating a bridge between two patterns can induce a "short circuit" in the design. A figure of merit which measures this design's vulnerability is obtained as the ratio of the critical area for a given defect size to the total layout area. This figure of merit is known as "defect-sensitivity". On the other hand, semiconductor yield is the probability of manufacturing ICs without faults. Thus, yield can be predicted by determining the defect-sensitivity of a given layout design and by studying the stochastic behavior of defects in a given manufacturing environment.

To gain an in depth knowledge about the relationship among defects, technological process, and circuit malfunction, a formal semantic model was developed to model process induced spot defects and their related

faults. These concepts and an overview of yield modeling are covered in the first chapter of the thesis. Chapter two presents a mathematical analysis of the geometrical properties of critical areas. In this chapter is also presented a taxonomy of defect-sensitivity models based on the presence of single or multiple faults, induced by single or multiple defects placed on single or multiple layers.

Current methods to find critical areas are based on simplifications to avoid the real problem. Moreover, the extraction of critical areas has been done accurately only for very simple layouts, while only approximations have been carried out for complex layouts. A new practical method to correctly find critical areas in single layers is presented in chapter three. In chapter four this method is extended to consider multiple layers.

Obviously, the multiple layer approach provides the means for a more accurate yield prediction as compared to the single layer approach. To demonstrate this fact, both methods are quantitatively compared in chapter five on a basis of several case study designs.

Contrary to the general belief that the computation of critical areas can be prohibitively expensive, the work presented in this thesis provides a solution with fast performance that can also be used in interactive applications. In addition to the extraction of critical areas, the developed systems provide further results for manufacturing process debugging, quality refinement of test vectors, and fault weighting.

Contents

Chapter 1

Defect Semantics and Yield Modeling

The variety of IC technologies increases the problem of choosing realistic fault models [37]. Traditional approaches to fault modeling assume a convenient high-level abstraction without considering the technology. Yet, faults have their origins in changes of the chemical and material compositions occurring in the IC.

Defects have very complex physical characteristics and may be significantly different from technology to technology [11, 12, 84]. The adequacy of fault modeling can be expressed in terms of defects occurring in the specific technology.

This chapter presents first a formal semantic model for IC technologies and process induced defects [59]. The model is a theoretical description of the physical properties of microelectronic processing including the necessary relationship between process induced defects and faults [45, 67]. The second main topic is an objective discussion on yield modeling, it's difficultness and development through the last 30 years.

1.1 Microelectronic Technology

A microelectronic technology, $\mathcal{T}=\{ t_k \mid k = 1,2, \ldots, N_{tech}\}$, is an ordered set of process steps which are concerned with changes in matter no more than a few microns above or below the surface of a *carrier*. This carrier is usually referred to as *wafer*. Basic process steps in the manufacturing of an IC are [1, 29]:

1. Oxide growth
2. Material deposition

3. Photoresist application
4. Mask exposure
5. Developing
6. Etching
7. Wash or strip
8. Photoresist removal
9. Implantation
10. Annealing
11. Diffusion

The goal of these steps is to transform an electrical circuit design into an operable device, e.g. the integrated circuit (IC).

Geometrically, an IC can be seen as part of a 3-D Euclidean space with "lateral" coordinates (x,y) and vertical coordinate z. In the z-direction a partition into intervals by fixing points in the z-axis, z_i, $i = 1,2,\ldots,N_{layer}$, is introduced. Those z-points define "matters" as open connected point sets as follows:

$$L_i = \{(x,y,z) \in E^3 \mid z_{i-1} < z < z_i\} \quad (1.1)$$
$$L_0 = \{(x,y,z) \in E^3 \mid -\infty < z < z_0\} \quad (1.2)$$
$$L_\infty = \{(x,y,z) \in E^3 \mid z_N < z < \infty\} \quad (1.3)$$

L_0 is actually the *substrate* with its background doping, whereas L_∞ is established by (electrically passive) *air* on top of the IC. In between is a set of layers of matters such that each different matter has unique electrical properties. By matter is meant a physical IC constituent such as thick oxide, thin oxide, metal, polysilicon, etc.. Such an arrangement of matter in *layers* is referred to as *silicon layer structure*. The set of layers is denoted as $\mathcal{L} = \{L_k \mid k = 1,2,\ldots,N_{layer}\}$, and the set of matters as $\mathcal{U} = \{u_k \mid k = 1,2,\ldots,N_{matter}\}$.

Each layer is shaped by a series of lithographic process steps such as oxidation, photo resist application, etching, etc. After the "shaping process" takes place some portions of matter disappear and some portions remain. A point set $a_k \subset L_i$ is defined to be a maximal connected *active* point set in L_i retained after the shaping process; a_k is simply called an *active pattern* in L_i. Any two such active patterns have empty intersection, that is $a_i \cap a_k = \emptyset$, $i \neq k$, because a_i and a_k are "maximal". The union of all active patterns in L_i is denoted by $\mathcal{A}_i$, that is $\mathcal{A}_i = \bigcup_k a_k$, and is called the *active region* in L_i. The complement of $\mathcal{A}_i$ in

L_i, denoted by $\bar{\mathcal{A}}_i = L_i - \mathcal{A}_i$ is called the *inactive region* in L_i. Physically, the set inactive region can be seen as the set of empty spaces, henceforth *inactive patterns*, corresponding to those layer portions that disappear after the shaping process.

Within one layer the inactive patterns may partially be occupied by some other matters. For instance, at process step t_k one may encounter active patterns of thick oxide in layer L_i. A few processing steps further, the inactive patterns of L_i may now be filled with metal. Thus, it is possible to define a set in terms of an enumerated type of distinct matters contained in L_i as $U_i \subset \mathcal{U}$. For this particular example U_i = {thick oxide, metal}. Assume that the point (x,y,η), $z_{i-1} < \eta < z_i$ is element of an inactive pattern. Fixing $z=\eta$ defines a 2-dimensional Euclidean space which in the case of $z_{i-1} < \eta < z_i$ will be denoted by $L_i(x,y)$. Then, for any point $(x,y) \in E^2$ a function $W_i(x,y) : L_i(x,y) \rightarrow U_i$ is defined, which in fact assigns a value from U_i to any pair of coordinates. This value is called the *state* of the layer at (x,y) and U_i the *stateset*.

From the above it follows that given the set of layers $\mathcal{L}$ there is an associated set of statesets $U = \{U_1, U_2, \ldots, U_{N_{layer}}\}$. Now establish the product set of all layers by letting $\boldsymbol{L} = (L_1, L_2, \ldots, L_{N_{layer}})$ be an N_{layer} dimensional vector. The product set $\boldsymbol{U} = (U_1, U_2, \ldots, U_{N_{layer}})$ is analogously defined. Logically $\boldsymbol{W} = (W_1, W_2, \ldots, W_{N_{layer}})$ becomes a vector function such that it is possible to write $\boldsymbol{W}(x,y) : \boldsymbol{L}(x,y) \rightarrow \boldsymbol{U}$.

Assume an NMOS process. Consider the silicon layer structure indicated in Fig. 1.1. Essentially five layers, L_1, L_2, L_3, L_4, L_5, and five statesets, U_1 = (FOX, DDS, DE), U_2 = (OX, THOX, ME), U_3 = (OX, POLY, ME), U_4 = (OX, ME), U_5 = (INS), are defined. For coordinate x_1 (with y fixed) one can obtain the state vector $\boldsymbol{W}(x_1)$ = (DDS, ME, ME, ME, INS). For coordinate x_2 (with y fixed) the state vector $\boldsymbol{W}(x_2)$ = (DE, THOX, POLY, OX, INS) is obtained as well.

Obviously the state characterization of silicon layer structures can be applied to identify electrical components by multivalued logical clauses. Adopting $\mathbf{x}$ as a don't care notation, consider for instance the clause $\omega(x,y)$ = (DB, THOX,POLY, $\mathbf{x}$, $\mathbf{x}$) true for some point in $(x,y) \in E^2$. This clause describes a point of an active gate area of an NMOS-transistor. The clause $\omega(x,y)$ = (DDS, ME $\vee$ OX, OX, ME $\vee$ INS, $\mathbf{x}$) indicates that (x,y) belongs to a source or drain region. These clauses are denoted as *state clauses*. A technology can be characterized by a set of state clauses

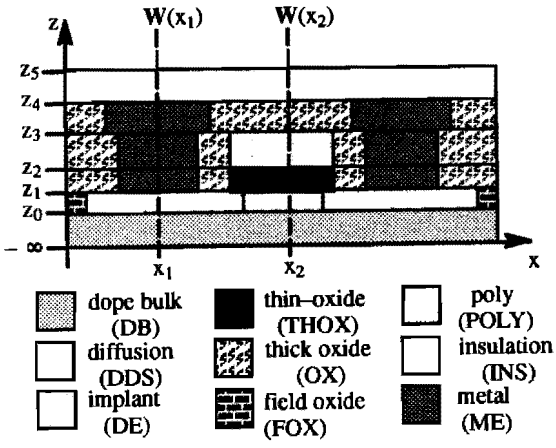


Figure 1.1. A Silicon Layer Structure

$\Sigma_{tech} = \{\omega(k) \mid k \geq 1\}$. Any of the state clauses identifies a silicon layer structure characterizing a constituent of a set of electrical components. To be able to link the silicon layer structure within a given technology Σ_{tech} to a circuit schematic, say Ψ, there must be a correspondence between the elements of Σ_{tech} and the set of constituents of the circuit schematic Ψ.

Assume $\omega \in \Sigma_{tech}$. Consider a point (x,y) such that $W(x,y)=\omega$. Assume now that this point is an inner point of a closed connected maximal set R with the property that for any $(\alpha,\beta) \in R$ the state function $W(\alpha,\beta) = \omega$ holds. Then $R(\omega)$ is called a *hard-structure*. Also, for later use, let us denote by $R(\omega) \mid L_i$ the *partial hard-structure* which is obtained from the hard-structure $R(\omega)$ by restricting its argument to the layer L_i. After elimination of the z-dimension, the IC is considered as a connected rectangular subset of the 2-D Euclidean space. There may be many hard-structures $R_r(\omega)$, $r \geq 1$, on such a chip.

The state clause ω characterizes the circuit constituent $\psi \in \Psi$ (or rather the type of circuit component in question); in addition a hard-structure $R_r(\omega)$ supplies all the geometrical information that completes the description of an instance of the respective circuit constituent. In other words, any $R_r(\omega)$ corresponds to some type of "circuit element". Examples of such circuit elements are "drain region", "gate region", "via", etc., as opposed to the traditional concept of circuit elements such as resistors, capacitors, transistors, etc.

How the patterns in the layers are determined is specified by an *IC artwork*. An IC artwork represents the layout of the circuit design to be mapped into hard-structures by the processing steps. Formally, an IC artwork is a vector of *masks* $\boldsymbol{M} = (M_1, M_2, \ldots, M_{N_{mask}})$. Each mask is defined as a 2-D Euclidean space, i.e. $M_k = \{(x,y) \in E^2\}$. Let us denote $\mathcal{M} = \{M_k \mid k = 1,2, \ldots, N_{mask}\}$ as the set of masks.

Within a mask, a connected point set o is bounded by a finite set of line segments such that every extreme point of a segment is shared by exactly two edges and no subset of edges has the same property. These connected point sets, henceforth called *zones*, divide the mask in two disjoint fields, *dark* and *light*. The set *dark field*, $O = \{ o_k \mid k \geq 1\}$ contains the regions to be mapped into the silicon layer structure if the lithographic processing step uses positive image projection. On the other hand, if negative image projection is employed, the regions of the set *light field*, $\overline{O}$, are mapped into the silicon layer structure. Thus, one can create a silicon layer structure by placing the masks on top of each other, correctly aligned to establish a *mask stack* to be processed in sequence.

Very much as in the case of actual silicon layer structures, state clauses for any point (x,y) of a mask stack can be defined. Any mask of the mask stack may at any point be either dark or light establishing essentially two sets per mask. In addition let us use **x** as a don't care notion. If $\Sigma_{artwork} = \{\mu(k) \mid k \geq 1\}$ is denoted as the set of state clauses associated with the IC artwork, then with any point $(x,y) \in E^2$ there exists a boolean cube $\boldsymbol{M}(x,y)$ as a vector of "1", "0" and **x** entries, where "1" stands for dark and "0" indicates light (in the case of positive image projection). As in the case of technology state clauses, these new clauses identify constituents of circuits as well.

Let Ω be the subset of all state clauses that identifies the set of constituents of a circuit schematic. Assume some state clause $\mu \in \Omega$. Further assume a point $(x,y) \in E^2$ such that $\boldsymbol{M}(x,y) = \mu$. In general μ will be in a connected set of points where one can identify a maximal connected set $Q(\mu)$ corresponding to one of the constituents from the set Ω. Such a set $Q(\mu)$ will be called a *soft-structure*.

In order to obtain consistency among the circuit schematic, the mask stack, and the silicon layer structure, a correspondence between the set of *hard-structures* Σ_{tech}, the set of *soft structures* $\Sigma_{artwork}$ and the set of

circuit constituents Ψ must be established. On one hand, this consistency is established by proper definitions of the set Ψ and the sets Σ_{tech} and $\Sigma_{artwork}$. The relation between Σ_{tech} and $\Sigma_{artwork}$ is induced of course by the proper interpretation of the effects of the processing steps $t_k \in \mathcal{T}$.

1.2 Modeling of Process Induced Spot Defects and Faults

Defects can be classified as *local* or *global*. The latter class concerns disturbances that affect complete regions of a wafer, while the local class concerns disturbances peculiar to only an IC. Spot defects are local disturbances of the silicon layer structure caused by dust particles, process variabilities, and other contaminations of the fabrication equipment. The general assumption is that spot defects are in essence random phenomena occurring with a certain stochastic frequency and size, and a certain stochastic spatial distribution on the wafers [46]. Spot defects have often been modeled as local disturbances of one layer in the form of small round or square spots of excess or missing material [39, 40].

Not all defects are due to lithographic processing steps. Some defects arise from process variability such as incomplete step coverages. Therefore, the way in which individual process steps are executed is of critical importance to the outcome of the IC. Each of these steps has its own deviations or disturbances from the ideal process which could contribute to physical changes in the structure of the IC, and thus, create defects. Identifying the sources of defect mechanisms, i.e. extra spots of metal, missing spots of polysilicon, etc., is of importance to the success of the final product, and to yield improvement. There are many sources of defect mechanisms. Appendix 1 presents only a small summary of some sources peculiar to their processing step and to human and physical contamination [5, 31, 33, 70, 87].

The spot defect semantic model presented in this section is meant to be expressive enough to cover types of more complex character. Defects may hit any combination of layers and on any layer they may be of any shape, be it that in extremely complex shapes the model may become difficult to handle.

Often enough defects reproduce the silicon layer structure of a hard-structure yet cause a deviation of the shape of such structure. The

importance of a defect is determined by the effect that it has on the behavior of the IC, this effect is called a *fault*. A *fault model* $\mathcal{F}$ maps the set of altered hard-structures, R_{def} onto the fault class F, $\mathcal{F}: R_{def} \rightarrow F$. The range of $\mathcal{F}$ is the set of fault types. Those fault types are equivalence classes of faults. They include an empty class referring to a fault free state.

The empty fault class has as its domain the set of hard-structures with the property that the IC functional behavior is unaltered. This class of hard-structures is approximately insensitive towards a given shape deviation. Therefore the respective kind of defects is denoted as the kind of *benevolent defects*. On the other hand, those defects placed in hard-structures which are assigned to nonempty fault classes are denoted as *catastrophic defects*.

Defect mechanisms for spot defects in hard-structures can be classified as *protrusions, intrusions*, and *isolated spots*. A *protrusion defect*, d^p in some layer L_i is an undesired active pattern defined as a connected set of points (x,y,z) such that at least one such point intersects some active pattern(s) in L_i. An *intrusion defect* d^i in some structure layer L_i is an undesired inactive pattern defined as a connected set of points intersecting some active pattern(s) of L_i. An *isolated spot defect* d^s in some structure layer L_i is a connected set of points such that no point of d^s intersects an active pattern of L_i. Fig. 1.2 illustrates these defect mechanisms.

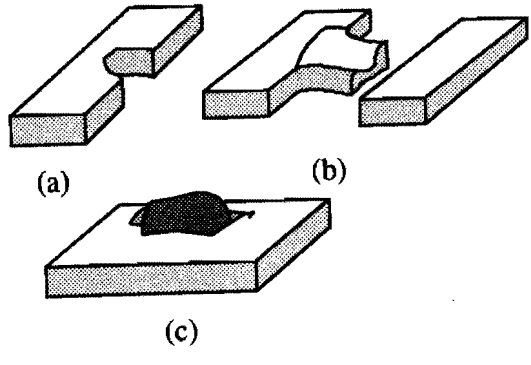


Figure 1.2. Types of defect mechanisms. (a) Intrusion defect. (b) Protrusion defect. (c) Isolated defects.

Protrusion and intrusion defects generated during process step t_i change the shape of active patterns in the same layer where they occur, and may also have impact on some active patterns at different layers processed at some alternate processing step, t_j , $j \neq i$. Isolated spot defects do not have effect on active patterns in the layer where they originate. Rather they may have effect on some active patterns in other layers. In fact, an isolated spot defect changes the state clause at its location. Thus, a parasitic or even undefined circuit constituent may be generated.

The notation of section 1.1 allows to define defect mechanisms as an additional set of multivalued state clauses. It may be necessary to extend the set of values for the various coordinates of Ω. For instance, an isolated spot defect in the thin oxide layer of an NMOS transistor active gate area most likely will imply the presence of polysilicon in the respective layer which in a correct structure would not appear. The defect-free state clause, in a 4-layer structure, may be $\omega = (\mathbf{x}, \text{THOX}, \text{POLY}, \text{OX})$ and the presence of the isolated spot would be indicated by some clause, say, $\omega_{defect} = (\mathbf{x}, \text{POLY}, \mathbf{x}, \mathbf{x})$. The shape of the defect would be captured by using the concept of a hard-structure.

To consider another example, assume the state clause for the presence of just metal in a 4-layer structure to be $\omega = (\mathbf{x}, \mathbf{x}, \mathbf{x}, \text{ME})$. Consider two hard-structures $R_1(\omega)$ and $R_2(\omega)$ identifying two different wiring trees in the metal layer. The state clause for a metal spot defect δ is of course equal to ω. Assume some hard-structure $D(\delta)$ actually modeling the defect. Then if $R_1(\omega) \cap D(\delta) \neq \emptyset \ \wedge \ R_2(\omega) \cap D(\delta) = \emptyset$, $D(\delta)$ acts as a protrusion of $R_1(\omega)$ if $D(\delta) - (R_1(\omega) \cap D(\delta)) \neq \emptyset$, otherwise δ is not a defect because it is actually contained in the metal wire. If however $R_1(\omega) \cap D(\delta) \neq \emptyset \ \wedge \ R_2(\omega) \cap D(\delta) \neq \emptyset$, then the whole point set $R_1(\omega) \cup D(\delta) \cup R_2(\omega)$ shows a bridge between the two hard-structures.

1.3 Approaches to Yield Modeling

Silicon foundries usually handle circuit design and process development as separate domains [41]. As a result, two different orientations in VLSI did evolve, a system orientation and a process orientation. System engineers visualize the VLSI design as a more or less geometrical activity. Process engineers see the process enhancements as the only means of improvements of performance.

Therefore, yield improvement has to strengthen the weaknesses of both parties and to create a "communication channel" between them.

Yield improvement and estimation is not a task that can be achieved through a "formula", or from one day to another. It is rather an education that is acquired through previous experiences by avoiding mistakes committed in the past, or simply by improving the quality of previous processes and/or designs. Hence, the "yield improvement tools" should help to visualize possible defects in order to take appropriate corrective actions, and should also predict possible faults in order to improve designs.

The essential needs of industry may be reflected in the following questions:

1. What will be the technological problems of a new product given the current processing conditions and manufacturing technology ?
2. What will be the wafer yield of the product ?
3. What can be done to enhance the robustness of the design, and thus the product yield ?

Question one is based on the continuous use of Product Yield Monitors (PYM). The objective of using these monitors is to replicate as close as possible the geometrical features of the actual products in order to foresee possible defects in the real production of the IC [7]. Question two is essential for yield management since a bad yield has a significant impact on the IC cost. Finally, question three is the implications of the previous ones, that is, to give a feedback about the problems to the process engineers and the designers, and to generate solutions in order to increase the product yield.

Naturally, with the advent of new technologies and smaller resolution features manufacturing yield is becoming a more important issue [37, 45, 82]. This is especially the case when the IC is a new product for which a design oriented yield modeling approach becomes a necessity [42]. This approach considers the IC not as a "black box", but as the union of geometrical features that are likely to be affected by defect mechanisms. Yet, several tasks of yield modeling cannot be covered by just studying the IC geometry. These tasks are the statistical characterization of the environmental conditions prevailing in the manufacturing line, such as the characterization of defect density

variations and defect size distributions to obtain area utilization factors. Furthermore, increasing the levels of semiconductor integration brings to attention topics such as the changes of the yield associated with individual process steps like etching, metallization, etc., or the spatial distribution of local deformations in the IC.

Based on the rationale presented above, yield modeling can be split into two approaches: 1) A *macro-model* that deals with wafers as an entity, such as to discover defect density variations, clustering, and random and systematic sources of yield loss, among others [6, 34, 47, 55, 68, 69]. 2) A *micro-model* dealing with the IC as an entity. In this approach, yield prediction is based on a characterization of the design's sensitivity to spot defects. The work presented in this thesis is devoted to the micro-approach.

Yield modeling has evolved from simple analytical formulae to complex full simulations [9, 36, 79]. In this long trajectory, evolution is seen from empirical formulae, to formulae based on statistical data, to simple CAD based analyses, to full Monte Carlo simulations. Each improvement is seen in the way that the models cope with more and more features to improve their accuracy. These features are shown in Table 1.1.

Table 1.1. Features in yield models

Feature	Type
1	IC area
2	Defect density
3	Spatial distribution of defects on wafers
4	Defect size distributions
5	Global disturbances on the wafers
6	Layout information
7	Technological process information

Next to the desired accuracy, modern approaches try to provide more information, not only to predict yield, but also to infer the reasons of yield loss [38, 73]. Yield models can be characterized according to their *fidelity, complexity,* and *dimensionality* [36]. By fidelity is meant the accuracy in predicting yield compared to actual data. Complexity is the accuracy in describing the physical phenomena causing yield losses.

Dimensionality is the number of features used in the model. These three characteristics are closely related. Obviously, as the dimensionality increases, more complex phenomena are taken into account, a fact that is reflected in a better fidelity of the model.

Though models with a higher dimensionality have the best fidelity, simpler models can be used as well. If one is interested in observing the effect of defects in the behavior of the IC such that efficient procedures for testing can be developed, or if there is a need to infer about the reasons of yield loss, then obviously the choice is a model with a high dimensionality heavily relying on CAD procedures. Yet if the interest lies in having a "number" that can indicate the current and future yield trends of a product, a simpler analytical model suffices. Notice however, that in order to apply an analytical model efficiently, and also to obtain realistic results, it is necessary to consider the features previously mentioned. The claim here is that analytical formulae have to rely on CAD approaches. Therefore, the previous features can be grouped in such a way that both CAD and analytical approaches can be combined and at the same time be independent of each other. For instance, analytical formulae can consider features 1 to 5 while CAD approaches features 5, 6 and 7. This leads to the following partitioning of tasks

i) CAD approaches dealing with geometrical features of layouts and with recipes of the manufacturing process,

ii) Analytical approaches dealing with the statistical characterization of defects.

CAD approaches should then tackle tasks as the localization of "critical areas" in the layout where defects can have a catastrophic impact on the functional behavior of the design, and also tasks as simulations of processing steps, i.e. etching, alignment, etc. On the other hand, analytical approaches should concentrate on statistical characterizations of defect density variations, spatial distributions of defects, quality control of processing steps, etc.

1.3.1 Brief overview of historical yield models

The commencement of yield modeling dates back to 1960 [9, 79] when a binomial model, and subsequent derivations, were proposed to evaluate the yield of transistors [26, 65, 91]. This model can be expressed as

$$\text{Yield} = (1 - \frac{A_{IC}}{A_{wafer}})^{N_{def}} \tag{1.4}$$

where N_{def} is the number of defects in an area A_{IC}, and A_{wafer} is the total area. The weakness of this model is twofold. First the defect density per wafer is assumed to be constant over a set of wafers, a situation which is rarely observed in real manufacturing environments [18,24,25,77,92]. And second the spatial distribution of defects over some wafer is considered to be uniform, whereas in real production lines clustering of defects is observed especially at the edges of the wafer [17,75,78,80,81].

Since the number of defects per unit area, or defect density, varies from wafer to wafer, from IC to IC, and even from run to run, a new model was proposed: [48]

$$\text{Yield} = \int_0^\infty e^{-A_{IC}D} f(D)dD \tag{1.5}$$

where D is a defect density, and f(D) a probability distribution function representing the variation in defect densities. This defect density distribution is known as a "compounder". The goal of the compounder is to describe as close as possible the defect density variations among and within wafers. Several types of compounders can be used, Murphy employed a uniform and a triangular probability distribution function, yet the predicted yield was still pessimistic. Seeds [66] conjectured that high yields observed for blocks of two, four, etc., chips in a wafer were caused by large populations of low defect densities and small populations of high defect densities, therefore he used an exponential defect density distribution which results in the following yield expression

$$\text{Yield} = \frac{1}{1 + A_{IC}\bar{D}} \tag{1.6}$$

where $\bar{D}$ is the average defect density. Despite the curves of the previous distribution functions could be shaped according to the mean and standard deviation of the distribution of defects, the yield model does not allow any adjustment for statistical analysis. Although the formula is appealing because of its simplicity the predicted yield is optimistic [79].

In 1972 an Erlang distribution was proposed as the compounder [51]. The reasons for using this distribution were that it can capture the characteristics of the exponential, bell shaped, and the delta distributions, which are good estimates for f(D), by adjusting only one parameter. Also the use of an equivalent average defect density fits actual data well. The parameter in this yield model must be an integer related to the number of process steps. However, in a technology with many process steps the parameter does not longer help to fit data well. Rather than using an Erlang distribution, Stapper [72] used a gamma distribution which results in the following model known as the "negative binomial yield model"

$$\text{Yield} = Y_0 \frac{1}{(1 + \frac{A_{IC}\bar{D}}{\alpha})^{\alpha}} \tag{1.7}$$

where Y_0 denotes the gross cluster yield. This parameter is required because of localized defect clusters which are usually observed in semiconductor processes [73]. The entity α is a parameter related to the coefficient of variation of the gamma distribution. The coefficient of variation is a rational number greater than zero usually associated with the clustering of defects. Furthermore, it can be realized that the gamma distribution can emulate the distributions mentioned above by just selecting appropriate values of α, and also that α is independent of the number of process steps.

Obviously not the entire area of the IC is sensitive to defects, and even more, the sensitive area varies from design to design. By sensitive, or critical, area is meant the area such that if the center of defect is placed there an incorrect IC behavior occurs. Therefore it is necessary to take into account the *defect-sensitivity* of the design in order to have more accurate results for yield prediction [14, 15]. The defect-sensitivity is defined as the ratio of the total critical area to the total IC area. This figure of merit is obtained as follows

$$\text{Sensitivity}(\delta) = \frac{A_{crit}(\delta)}{A_{IC}} \tag{1.8}$$

where δ is the defect size, and A_{crit} represents the total critical area.

Since the probability of occurrence of defects of different sizes is not constant, the probability ϕ that the IC will have a fault depends upon a

defect size distribution as it is indicated in the following formula, [16]

$$\phi = \int_{min}^{max} \text{Sensitivity}(\delta)\, D_{size}(\delta)\, d\delta \tag{1.9}$$

Here min and max are the smallest and largest defect size respectively, and $D_{size}(\delta)$ is the defect size distribution.

The semiconductor yield is the probability of manufacturing devices without faults. If the IC probability of failure is known, yield can be predicted as well. By incorporating eq. 1.9 in eq. 1.7 a new model which has a better fidelity is obtained. This model is expressed as

$$Y = Y_0 \frac{1}{(1 + \frac{A_{IC}\bar{D}\phi}{\alpha})^{\alpha}} \tag{1.10}$$

Obviously the fabrication of ICs requires a long sequence of process steps, each having the potential to introduce different types of defects. Therefore, the fidelity of the previous models can be improved by relating the yield to the number of defect mechanisms found in each processing step [28, 49]. For instance, formula 1.10 can be transformed into

$$\text{Yield} = \prod_{i=1}^{i=n} Y_{0i} \frac{1}{(1 + \frac{A_{IC}\bar{D}_i\phi_i}{\alpha_i})^{\alpha_i}} \tag{1.11}$$

where the index i distinguishes the distinct defect mechanisms.

The impact of CAD shows up in calculating the defect-sensitivity of designs as this is a very laborious task to be performed by hand. Several existing approaches to yield modeling are based on CAD extractions of critical areas combined with some analytical formulae [32, 60]. However, these systems are based on extractions of single layer critical areas. The extractions neglect relationships between layers, a fact which can result in inaccuracies of the predicted yield. One way of avoiding this shortcoming is by performing Monte Carlo yield simulations [89, 90]. In essence, Monte Carlo yield simulations place defects on the IC and then analyze their effect. For each iteration of the Monte Carlo loop a defect is positioned on the layout, then the defective IC circuit is extracted and compared to the defect free circuit to determine possible IC functional faults. As this is a very expensive

procedure, an analytical approach that makes use of a simplified concept of multi-layer critical areas was developed [8]. This approach comprises a hierarchical model for defect statistics describing defect characteristics at different layout levels. It employs strict analytical methods to find the failure probability of simple layout patterns which in turn are used to calculate the failure probability of complex layout patterns in a hierarchical style.

Chapter 2

Computational Models for Defect Sensitivity

For a long time layout verification has been confined mainly to validate the design rules imposed by the fabrication process. However, as processes mature and advance to smaller resolution features other forms of layout verification become imperative. In the past, external contaminants that could lead to defective layouts were not relevant and in most cases not taken into account. Nowadays their significance is crucial to the successful manufacturing of the chip even though there is a precise control of the line features. One such form of layout verification is to predict the robustness of the artwork in real manufacturing environments by carrying out a design's defect-sensitivity analysis.

IC sensitivity to spot defects is studied by extracting the "critical areas" from layouts. Roughly speaking, *a critical point in the layout is a point such that if a spot defect is centered there a malfunction in the respective circuit arises*. Critical areas are open connected sets of critical points. [30, 74, 76] They naturally depend on the layout geometry and on the defect size involved. Thus, the defect-sensitivity of a design is obtained as the ratio of the total critical area to the total layout area.

The original concept of critical areas appeared in the late 60's at the facilities of IBM Yorktown where initial attempts to evaluate spot defects in FET memories were done [79]. It was not until 1983 that this concept was presented in the literature [74], and in the same year that a simple geometrical method to extract critical areas from complex layouts appeared [43].

This chapter is aimed at presenting a formal theory to compute critical areas [56]. The theory is based on a new concept of "susceptible sites" which then is used to compute the critical areas for a whole set of points in a given domain of defect sizes. An overview establishing a taxonomy for defect-sensitivity models is presented as well.

2.1 Taxonomy of Defect-Sensitivity Models

Very much as in the case of Flynn's taxonomy of computer architectures [19] one can classify the computational models of defect's effects on an IC. The classification is based on the effect of defects in patterns of established silicon layer structures. These effects are denoted as *failure primitives*. Furthermore, these failure primitives are grouped into two classes, *intersections* and *covers*. The intersections comprise defects affecting patterns of the same layer where they originated from while the covers concern defects affecting patterns of other layers than those of their layer of origin, see Fig. 2.1. Among the *intersection class* the failure primitives are the following

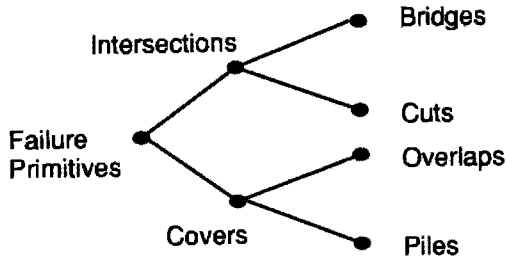


Figure 2.1. Taxonomy of failure primitives

- **Bridges**. Two or more active patterns, $a_k \subset L_i$, $k = 1,2,3,..$, joined by one protrusion defect in the same layer.
- **Cuts**. One or more patterns, $a_k \subset L_i$, $k = 1,2,...$, cut by an intrusion defect in the same layer.

The *cover class* comprises

- **Overlaps**. One or more patterns, $a_k \subset L_i$, $k = 1,2,...$, intersected by a spot defect, protrusion or isolated type, of some other layer different from L_i.

- **Piles.** An undesired layer in a silicon layer structure caused by an isolated spot defect. For instance, given the state clause $\mu = (L_1, L_2, L_3, \ldots, L_n)$, the state clause $\mu' = (L_1, L_x, L_2, L_3, \ldots, L_n)$ represents the undesired layer between L_1 and L_2.

These failure primitives can further be abstracted to faults at the transistor level, gate level, and system level [10, 13, 20, 86].

Flynn's taxonomy classifies computer architectures according to the presence of single or multiple streams of instructions and data. Analogously, this taxonomy classifies models of defect sensitivity according to the presence of single or multiple failure primitives, induced by single or multiple spot defects, on single or multiple layers. This classification yields two categories of defect susceptibility in silicon layers, namely, a *sequential class* and a *concurrent class*. The sequential class deals with defects causing only one type of failure primitive, while the concurrent class deals with defects causing more than one distinct failure primitive type at a time. For instance, a spot defect can introduce a bridge and a cut simultaneously.

With the two classes it is also possible to group defects according to their *nature*. The nature of a defect is determined by two features, namely the number of defects simultanously occurring and the number of layers involved. This classification yields four categories, see Fig. 2.2, namely:

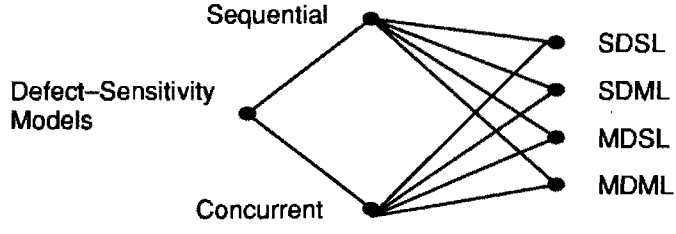


Figure 2.2. Taxonomy of defect-sensitivity models

SDSL (Single Defect Single Layer) In these models one defect at a time is modeled in one layer of the silicon layer structure. The models ignore the electrical significance of relationships among layers, as is the case with transistors, vias, etc. Due to this restriction, the only failure primitives that can be meaningfully applied in this category

are bridges and cuts.

SDML (Single Defect Multiple Layers) In this category fall the models that consider the isolated effect of one defect on the entire silicon layer structure. For instance in a diffusion-metal via, intrusion defects of metal and diffusion, and isolated defects of extra oxide in the hole of the via cause the same electrical fault which is a break in the respective node. This way of modeling allows to compute critical areas for silicon layer structures rather than for simple patterns.

MDSL (Multiple Defect Single Layer) The simultaneous effects of one or more defects, placed on one layer and possibly occurring at the same place, are modeled through the whole silicon layer structure. These models consider the concurrent effect of defects that originate in the same place. As an example assume a protrusion and an intrusion defect in some layer L_i both occurring at the same place such that no electrical fault is formed.

MDML (Multiple Defects Multiple Layers) The effect of one or more defects, simultaneously occurring in more than one layer and possibly occurring at the same place, is modeled through the entire silicon layer structure. These models consider the concurrent effects of defects in more than one layer. Consider for instance a spot of missing thick oxide and a spot of missing polysilicon, in a poly-metal crossing, both occurring at the same place, such that their net effect is canceled.

The critical areas that will be developed through the rest of this chapter are for sequential models.

2.2 Theoretical Foundation of Critical Areas

It is difficult to model the exact shape of defects since in reality they are rough-edged splotches. However, through this dissertation defects will be modeled as square shaped objects. This approximation is sufficiently correct, and furthermore it can be proved that it implies very simple and fast algorithms.

Before going into details, some definitions which will be used frequently through this section will be introduced first. Let $(pos(\alpha) ; l(\alpha) , r(\alpha))$, $(pos(\beta) ; l(\beta) , r(\beta))$, be two horizontal line segments α, and β, with ordinate pos having l and r as their left and right abscissae. Both line

segments, α and β, are *comparable* at abscissa x if there exists a vertical line that intersects them. The relation *above at* x is defined as: α above β at x if α and β are comparable at x and pos(α) > pos(β) [62]. Analogously one says that β is *below* α. Two more functions are defined, x(p) and y(p), which return the x and y coordinates of a point p, respectively.

The explanations to follow are for horizontal line segments and are restricted only to paraxial layouts.

2.2.1 Susceptible sites

Susceptible sites are subsets of active and inactive patterns which provide indications of defect susceptibility of sections of silicon layer structures. Susceptible sites are always related to one separately defined defect mechanism. To consider an example, assume three non intersecting parallel active patterns; let two of these patterns be polysilicon and also let them be adjacent to each other; let the third active pattern be metal. Then, the inactive pattern between the two polysilicon patterns is a susceptible site for protrusion defects of polysilicon, but the inactive pattern between the metal and polysilicon patterns is not a susceptible site for protrusion defects of metal, nor is it for protrusion defects of polysilicon. Analogously, the polysilicon active patterns are susceptible sites for intrusion defects of polysilicon but not for defects of metal.

Let us formalize now the construction of susceptible sites. Assume three closed connected point sets A, B, and C as shown in Fig. 2.3a.

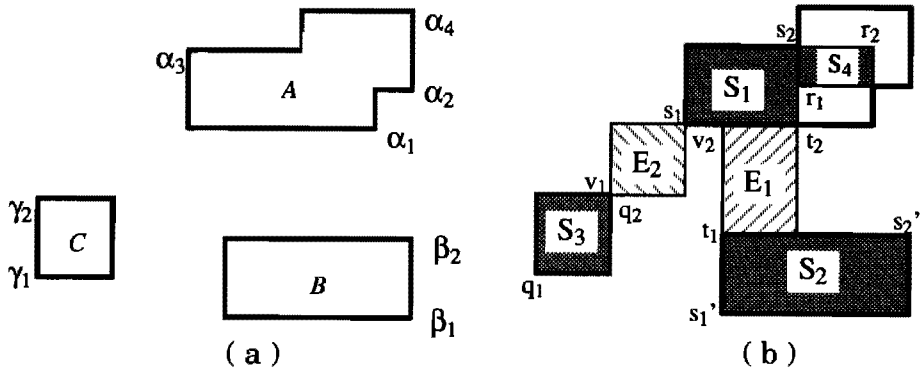


Figure 2.3. (a) Three connected point sets. (b) Susceptible sites.

Let α_1 , α_2 , α_3 , and α_4 be four horizontal line segments of A. Let also β_1 and β_2 , and γ_1 and γ_2 be the horizontal line segments of B and C, respectively. Take any two comparable line segments of the same closed connected point set such as α_1 and α_3 . Then, in Fig. 2.3b the open rectangle S_1 with corner points (s_1 , s_2) is called a *vertical internal lateral susceptible site*, where

$$\begin{cases} s_1 = (\max(l(\alpha_3), l(\alpha_1)), \min(pos(\alpha_3), pos(\alpha_1))) \\ s_2 = (\min(r(\alpha_3), r(\alpha_1)), \max(pos(\alpha_3), pos(\alpha_1))) \end{cases} \quad (2.1)$$

Take now any two non-comparable line segments of the same connected point set such that the interior of the point set lies above one of the line segments and below the other one, i.e. α_3 and α_2 . Then in Fig. 2.3b the open rectangle $S_4(\alpha_3, \alpha_2)$ with corner points (r_1 , r_2) is called an *internal corner susceptible site*, where

$$\begin{cases} r_1 = (\min(r(\alpha_3), r(\alpha_2)), \min(pos(\alpha_3), pos(\alpha_2))) \\ r_2 = (\max(l(\alpha_3), l(\alpha_2)), \max(pos(\alpha_3), pos(\alpha_2))) \end{cases} \quad (2.2)$$

Similarly, two kinds of susceptible sites between different connected point sets are identified. However these sites are defined as a function of their associated internal ones. Take any two internal lateral susceptible sites such that their line segments are comparable, for instance $S_1 = (s_1, s_2)$ and $S_2 = (s_1', s_2')$ in Fig. 2.3b. Then, in the same figure, the open rectangle $E_1(S_1, S_2)$ with corner points (t_1 , t_2) is denoted as a *vertical external lateral susceptible site*, where

$$\begin{cases} t_1 = (\max(x(s_1), x(s_1')), \min(y(s_2), y(s_2'))) \\ t_2 = (\min(x(s_2), x(s_2')), \max(y(s_1), y(s_1'))) \end{cases} \quad (2.3)$$

External corner susceptible sites are established from any two internal lateral sites belonging to different closed connected point sets such that no line segment of one internal susceptible site is comparable to any line segment of the other internal susceptible site. Consider the internal susceptible sites $S_1 = (s_1, s_2)$ and $S_3 = (q_1, q_2)$ of Fig. 2.3b, then the *external corner susceptible site* $E_2(S_1, S_3)$, with corner points (v1,v2) is formed as follows

$$\begin{cases} v1 = (\min(x(s_2), x(q_2)), \min(y(s_2), y(q_2))) \\ v2 = (\max(x(s_1), x(q_1)), \max(y(s_1), y(q_1))) \end{cases} \quad (2.4)$$

The *magnitude* of a vertical susceptible site $S = (s_1, s_2)$ is defined as

$$Mag(S) = y(s_2) - y(s_1) \tag{2.5}$$

2.2.2 Critical regions and areas

A *critical region* is an open connected point set constructed for a defect (of size δ), such that if the center of that defect is placed anywhere on this region, the defect is catastrophic. Critical regions are directly constructed from susceptible sites. Naturally, they are a function of the defect size and of the defect mechanism.

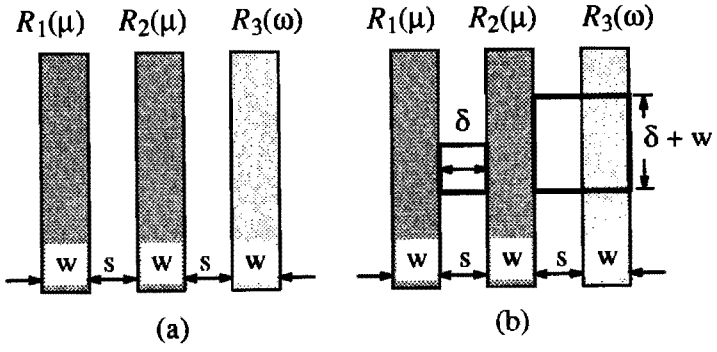


Figure 2.4. (a) Three hard-structures. (b) Different defect sizes to form a bridge and a parasitic transistor

Since the electrical significance of layer interrelationships is of crucial importance to determine whether a hard-structure is catastrophically affected by spot defects, a *failure criterion*, specific to the hard-structure, must be introduced. The failure criterion is a bound defined as a rational number. It determines the size of the intersection between defect and pattern typical for some defect type. A defect is catastrophic if the size exceeds the bound. To consider an example, assume the state clauses μ and ω for the presence of poly and diffusion, in a 4-layer structure. Let these clauses be μ = (SUBSTRATE,OXIDE,POLY,x) and ω = (DIFFUSION,OXIDE,OXIDE,x), respectively. Consider three hard-structures $R_1(\mu)$, $R_2(\mu)$, and $R_3(\omega)$ identifying two wires of poly and one of diffusion, each one of them of width w and spaced a distance s apart of each other, as depicted in Fig. 2.4(a). Consider now the presence of a protrusion defect in the poly layer. While a minimum defect size δ can introduce a bridge between $R_1(\mu)$ and $R_2(\mu)$, the same defect size has no effect between $R_2(\mu)$ and $R_3(\omega)$. However, if the defect size were at least δ + w a parasitic transistor could have been formed, see Fig. 2.4(b). In

this last situation w represents the failure criterion of $R_3(\omega)$ due to protrusion defects originating in the poly layer. Summarizing, a failure criterion depends on the defect mechanism, on the state clause, and on the geometrical situation of the hard-structure involved in the defect mechanism.

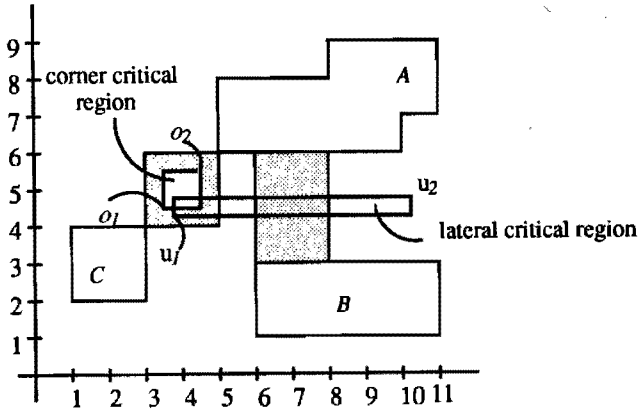


Figure 2.5. Creation of critical regions from susceptible sites. The corner critical region is created for a defect size of 3 units, the lateral critical region for a defect size of 3.5 units.

Let us first formalize the notion of *critical regions*. Assume now a lateral susceptible site, either internal or external. Take for instance the external susceptible site $E_1 (S_1 , S_2) = (t_1 , t_2)$ of Fig. 2.3b. Assume now that we have a defect of size δ. Then, a *vertical lateral critical region* $C_{lat}(\delta)$ is established only if

$$y(t_2) - (\frac{\delta}{2} + \Phi_{S_1}) \leq y(t_1) + (\frac{\delta}{2} + \Phi_{S_2}) \tag{2.6}$$

where Φ_{S_1} and Φ_{S_2} are the failure criteria of the hard-structures associated to the susceptible site $E_1 (S_1 , S_2)$. If the susceptible site were of an internal type, then $\Phi_{S_1} = \Phi_{S_2}$ because there is only one susceptible site. $C_{lat}(\delta)$ is in fact an open rectangle with corner points (u_1, u_2), see Fig. 2.5, given as

$$\begin{aligned} u_1 &= (x(t_1) - (\frac{\delta}{2} + f),\ y(t_2) - (\frac{\delta}{2} + \Phi_{S_1})) \\ u_2 &= (x(t_2) + (\frac{\delta}{2} + f),\ y(t_1) + (\frac{\delta}{2} + \Phi_{S_2})) \end{aligned} \tag{2.7}$$

where f is an additional horizontal extension of the critical region that is dependent on the failure criterion. This extension is denoted as the "end-effect" of the critical region. It appears in cases when the defect's edge can be located some distance f away of the pattern, without having to physically intersect it, and still be catastrophic.

Corner critical regions are obtained from corner susceptible sites. As with lateral regions, whether the site is external or internal is of no relevance. Take for instance $E_2(S_1, S_3) = (v_1, v_2)$ in Fig. 2.3b. The corner critical region is established if the following condition is satisfied

$$\begin{aligned} &(y(v_2) - (\frac{\delta}{2} + \Phi_{S_1}) \le y(v_1) + (\frac{\delta}{2} + \Phi_{S_3})) \wedge \\ &(x(v_2) - (\frac{\delta}{2} + \Phi_{S_1}) \le x(v_1) + (\frac{\delta}{2} + \Phi_{S_3})) \end{aligned} \tag{2.8}$$

where Φ_{S_1} and Φ_{S_3} are the failure criteria of the associated hard-structure(s) of E_2. Also, if E_2 were an internal susceptible site, $\Phi_{S_1} = \Phi_{S_3}$. Then, the open rectangle $C_{cor}(\delta)$, of Fig. 2.5, with corner points (o_1, o_2) defines the corner critical region, where

$$\begin{aligned} o_1 &= (x(v_2) - (\frac{\delta}{2} + \Phi_{S_3}), y(v_2) - (\frac{\delta}{2} + \Phi_{S_3})) \\ o_2 &= (x(v_1) + (\frac{\delta}{2} + \Phi_{S_1}), y(v_1) + (\frac{\delta}{2} + \Phi_{S_1})) \end{aligned} \tag{2.9}$$

The area enclosed in a critical region is called the *critical area*. Given n lateral critical regions and m corner critical regions, the total critical area for a defect size δ is obtained as

$$\text{Critical Area} = \text{area}\Big(\bigcup_{1 \le i \le n} C_{lat}(\delta)_i \cup \bigcup_{1 \le j \le m} C_{cor}(\delta)_j \Big) \tag{2.10}$$

where area(A) is a function computing the area of a given closed point set A according to the Euclidean metric.

2.3 Geometrical Proof of the Construction of Critical Regions

It has been shown that critical regions can be found geometrically [43]. Under this approach critical regions for bridges are found by expanding each pattern by an amount equal to half the defect size, and then by checking if the expansions intersect. If so, then the (amount of) intersection corresponds to the critical region between the patterns. In

the case of critical regions for cuts every pattern is shrunk by half the defect size, and a critical region is established only, when on shrinking, parallel edges of the same region pass over each other. This solution is formally postulated, for failure criteria equal to zero, by:

Algorithm 2.1. Geometrical construction of critical regions

---

foreach ($L_i \in \mathcal{L}$) {
  $n = |\mathcal{A}_i|$, $\mathcal{A}_i \subset L_i$
  foreach ($a_k \in \mathcal{A}_i$) {
    Bisect all the angles of a_k. *Move the edges outwards by a distance 0.5* δ *while keeping all vertices in the original angle bisectors. Denote this new point set as* a_k^{bridge}.

    Bisect all the angles of a_k. *Move the vertical (horizontal) edges inwards and the horizontal (vertical) edges outwards by a distance* 0.5δ *while keeping all vertices in the original angle bisectors. Denote this new point set as* a_k^{cut}.
  }
  /* *form critical regions* */

$$C_{bridge}(\delta) = \bigcup_{1 \le k < j \le n} (a_k^{bridge} \cap a_j^{bridge})$$

$$C_{cut}(\delta) = \bigcup_{1 \le k \le n} (a_k^{cut})$$

  /* *compute critical areas* */

  $\text{Critical Area}_{bridges} = \text{area}(C_{bridge}(\delta))$
  $\text{Critical Area}_{cuts} = \text{area}(C_{cut}(\delta))$.
}

The above solution has for bridges a quadratic time complexity in terms of n, and thus it is computationally prohibitive for very large layouts. Also, the shrinkage-expansion operations imply that a layout extraction has to be executed for each defect size. An important shortcoming of this method is that critical regions are directly derived from the point sets rather than from the susceptible sites. This action hinders the use of failure criteria because for any susceptible site a separate value for Φ can be defined. That is, there may be different values of Φ for different portions of some pointset a_k. Hence, for the same defect size some critical regions cannot be established for defects inducing failure primitives such as covers and piles using the shrinkage-expansion technique.

Since susceptible sites are subsets of patterns and critical regions are derived directly from susceptible sites, only a single layout extraction is needed for any span of defect sizes. Therefore, the problem is reduced to determine subsets constituting the critical regions from the susceptible sites for any given defect size. Furthermore, by using susceptible sites failure primitives such as piles and covers can be analyzed as well.

The following four theorems prove that all the critical regions obtained from algorithm 2.1 can be obtained as well using the concept of susceptible sites for a failure criterion equal to zero. Theorems 2.1 and 2.2 prove that all the points contained in any corner critical region, C_{cor}, and in any lateral critical region, C_{lat}, are contained in the point set C_{bridge} of algorithm 2.1. Similarly, theorem 2.3 demonstrates that all the points of any a_k^{bridge} are included in one or more C_{lat} and C_{cor}. Finally, theorem 2.4 shows that all the critical regions obtained from susceptible sites are all the critical regions obtained from algorithm 2.1.

Theorem 2.1 If for given a defect size δ, $C_{lat}(\delta)$ is a critical region obtained from an external lateral susceptible site, then $C_{lat}(\delta) \subset C_{bridge}(\delta)$.

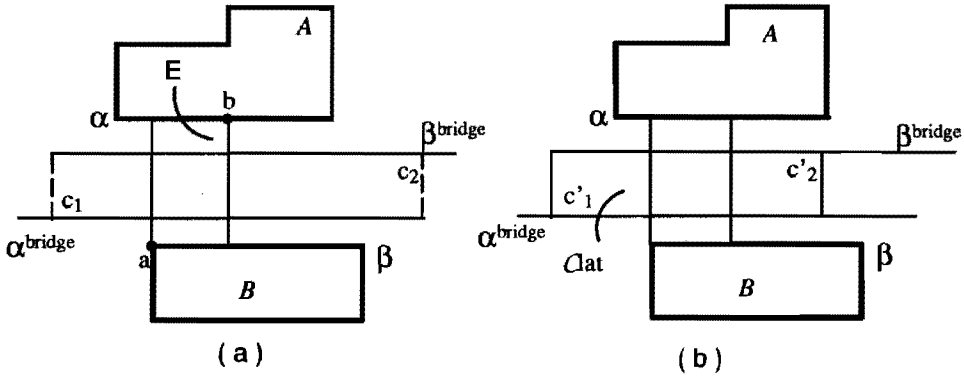


Figure 2.6. Theorem 2.1. (a) Susceptible Sites. (b) Extracted Critical Regions

Proof: Without loss of generality assume two connected point sets A and B, and one external lateral susceptible site E = (a,b), as depicted in Fig. 2.6. Let α and β be two line segments in the boundary of A

and B, respectively, such that α is above β. Let α^{bridge} and β^{bridge} be two expanded line segments, of A and B, respectively. From algorithm 2.1, if $(pos(\alpha^{bridge}) \leq pos(\beta^{bridge})$ then the rectangular open section, $C_{\alpha\beta}$, between α^{bridge} and β^{bridge} has corner points (c_1,c_2) given as $c_1 = (\max(l(\alpha), l(\beta)) - \frac{\delta}{2}, pos(\alpha) - \frac{\delta}{2})$, $c_2 = (\min(r(\alpha), r(\beta)) + \frac{\delta}{2}, pos(\beta) + \frac{\delta}{2})$. From the definition of critical regions, $C_{lat}(\delta) = (c'_1,c'_2)$ is directly derived from E according to eq. 2.7. It can be observed that $c_1 = c'_1$, and that $x(c'_2) > x(c_2) \wedge y(c_2) = y(c'_2)$, from which it follows that $C_{lat}(\delta) \subset C_{\alpha\beta}$. Since $C_{\alpha\beta} \subset C_{bridge}(\delta)$ it follows that $C_{lat}(\delta) \subset C_{bridge}(\delta)$.
□

Theorem 2.2 If for a given defect size δ, $C_{cor}(\delta)$ is a critical region obtained from a corner susceptible site, then $C_{cor}(\delta) \subset C_{bridge}(\delta)$.

Proof: Let α and β be the horizontal line segments, of any two point sets A and B, respectively. Assume that $r(\beta) < l(\alpha)$ and $pos(\alpha) > pos(\beta)$. Let us denote α^{bridge} and β^{bridge} as the corresponding expanded line segments, according to algorithm 2.1, for a defect of size δ. If $pos(\alpha^{bridge}) < pos(\beta^{bridge}) \wedge r(\beta^{bridge}) > l(\alpha^{bridge})$, then the rectangular open section $C_{\alpha\beta}$ between $\alpha^{bridge} \wedge \beta^{bridge}$ has corner points (d_1,d_2) given as $d_1 = (l(\alpha) - \frac{\delta}{2}, pos(\alpha) - \frac{\delta}{2})$, $d_2 = (r(\beta) + \frac{\delta}{2}, pos(\beta) + \frac{\delta}{2})$. From the definition of corner critical regions it can be noticed that (d_1,d_2) is equivalent to the corner points given in eq. 2.9. Since $C_{\alpha\beta} \subset C_{bridge}(\delta)$ it follows that $C_{cor}(\delta) \subset C_{bridge}(\delta)$.
□

Theorem 2.3 If for a given defect size δ, an open rectangular section $C_{\alpha\beta}$ is established from any two α^{bridge} and β^{bridge}, according to algorithm 2.1, then there exist one or more lateral critical regions $C_{lat}(\delta)_i$ and corner critical regions $C_{cor}(\delta)_j$ such that $\bigcup_i C_{lat}(\delta)_i \cup \bigcup_j C_{cor}(\delta)_j = C_{\alpha\beta}$.

Proof: Without loss of generality assume three closed connected point sets A, B, and C as shown in Fig. 2.7. Let α, β, and γ, be three maximal line segments in the boundary of A, B, and C, respectively. Let also α be above β, $pos(\alpha) > pos(\gamma) \wedge pos(\gamma) = pos(\beta)$, and $r(\gamma) < l(\beta)$. Let $E_1 = (a,b)$, $E_2 = (a',b')$, and $E_3 = (a'',b'')$ be two lateral and one corner susceptible sites obtained from the point sets, respectively. Let the point set $C_{\alpha\beta}$

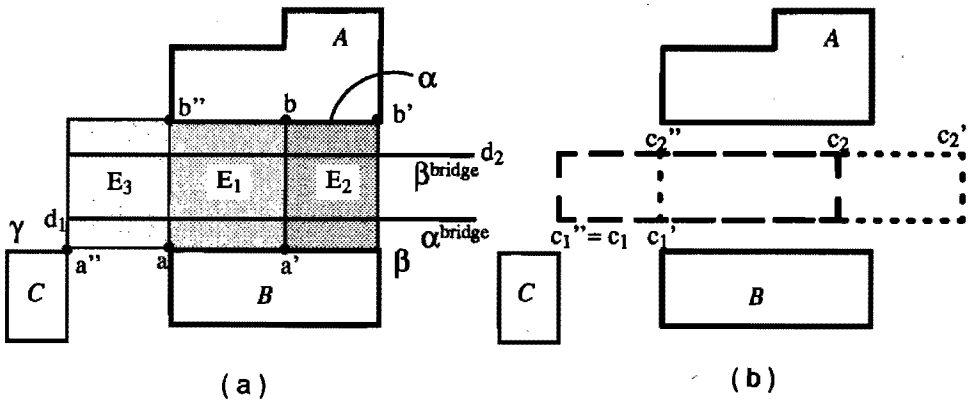


Figure 2.7. Theorem 2.3. (a) Three connected point sets. (b) Extracted Critical Regions

be expressed by the corner points $d_1 = (l(\alpha) - \frac{\delta}{2}, pos(\alpha) - \frac{\delta}{2})$ and $d_2 = (r(\alpha) + \frac{\delta}{2}, pos(\beta) + \frac{\delta}{2})$. Now, the corner critical region $C_{cor}(\delta)$, obtained from E_3, has corner points $c''_1 = (x(b'') - \frac{\delta}{2}, y(b'') - \frac{\delta}{2})$ and $c''_2 = (x(a'') + \frac{\delta}{2}, y(a'') + \frac{\delta}{2})$. Similarly, the lateral critical regions $C_1(\delta)$ and $C_2(\delta)$, obtained from E_1 and E_2 have corner points $c_1 = (x(a) - \frac{\delta}{2}, y(b) - \frac{\delta}{2})$ and $c_2 = (x(b) + \frac{\delta}{2}, y(a) + \frac{\delta}{2})$, and $c'_1 = (x(a') - \frac{\delta}{2}, y(b') - \frac{\delta}{2})$ and $c'_2 = (x(b') + \frac{\delta}{2}, y(a') + \frac{\delta}{2})$, respectively. It can be seen that $y(c_2) = y(c'_2) = y(c''_2) = y(d_2) \wedge y(c_1) = y(c'_1) = y(c''_1) = y(d_1)$ and that $x(c''_1) \le x(c_1) \le x(c''_2)$, $x(c_1) \le x(c'_1) \le x(c_2)$, $x(c'_2) > x(c_2)$. Since $x(c''_1) = x(d_1) \wedge x(c'_2) = x(d_2)$, the theorem holds.
□

Theorem 2.4 Let n be the number of critical regions $C_{lat_i}(\delta)$ and m be the number of critical regions $C_{cor_j}(\delta)$ obtained from lateral and corner susceptible sites for bridges, respectively, then $(\bigcup_{1 \le i \le n} C_{lat}(\delta)_i \cup \bigcup_{1 \le j \le m} C_{cor}(\delta)_j) = C_{bridge}(\delta)$.

Proof: The theorem holds from the proofs of theorems 1, 2, and 3.
□

The following two theorems examine some properties of critical regions. These theorems essentially demonstrate that some critical regions are enclosed in others. Theorem 2.5 demonstrates that if a corner susceptible site is completely contained by another corner susceptible site, then the critical region obtained from the contained susceptible site encloses the critical region of the external one. Theorem 2.6 demonstrates the same case but for lateral susceptible sites.

Theorem 2.5 Let E_1 and E_2 be two *corner susceptible sites*, and $C_1(\delta)$ and $C_2(\delta)$ their corresponding critical regions for a defect size δ, respectively. If $E_1 \subset E_2$ then $C_2(\delta) \subset C_1(\delta)$.

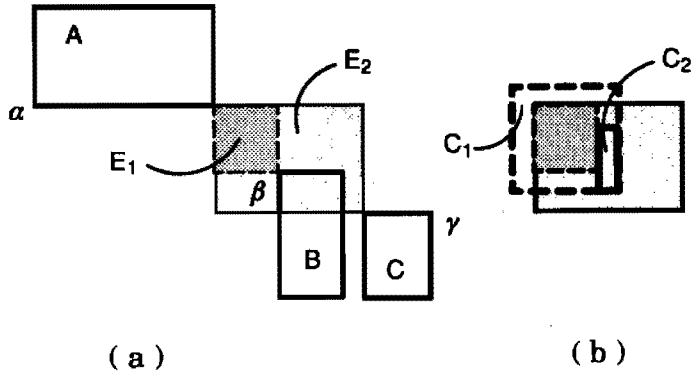


Figure 2.8. Theorem 2.5. (a) Susceptible Sites. (b) Extracted Critical Regions

Proof: Assume three closed connected point sets A, B, and C, as depicted in Fig. 2.8. Let α, β, and γ be three maximal horizontal line segments of A, B, and C, respectively, such that the interior of A is above α, the interior of B is below β and the interior of C is below γ. Assume also that α and β, and α and γ are diagonally neighboring in the following order $r(\alpha) < l(\beta) < r(\beta) < l(\gamma) < r(\gamma)$, and $pos(\alpha) > pos(\beta) \geq pos(\gamma)$. Let $E_1(A,B)$ and $E_2(A,C)$, be the corresponding corner susceptible sites given by the corner points (q_1,q_2) and (r_1,r_2), respectively, expressed as: $q_1 = (r(\alpha), pos(\beta))$, $q_2 = (l(\beta), pos(\alpha))$, $r_1 = (r(\alpha), pos(\gamma))$, $r_2 = (l(\gamma), pos(\alpha))$. If a defect of size $\delta \geq \max(|l(\gamma) - r(\alpha)|, |pos(\alpha) - pos(\gamma)|)$ is placed in such a form that it affects the three connected point sets simultaneously, then the critical regions $C_1(\delta)$ with corner points (u_1,u_2), and $C_2(\delta)$ with corner points (v_1,v_2),

that are obtained from $E_1(A,B)$ and $E_2(A,B)$, respectively, are given by $u_1 = (l(\beta) - \frac{\delta}{2}, pos(\alpha) - \frac{\delta}{2})$, $u_2 = (r(\alpha) + \frac{\delta}{2}, pos(\beta) + \frac{\delta}{2})$, $v_1 = (l(\gamma) - \frac{\delta}{2}, pos(\alpha) - \frac{\delta}{2})$, $v_2 = (r(\alpha) + \frac{\delta}{2}, pos(\gamma) + \frac{\delta}{2})$. It can be seen that the abscissa of u_1 is smaller than the one of v_1, and also that the coordinates of $u_2 \geq v_2$, from which it follows that $C_2(\delta) \subset C_1(\delta)$.
□

Theorem 2.6 Let A, B and C be three closed connected point sets, such that A and B, as well as B and C, are vertical neighbors, as depicted in Fig. 2.9. Let E(A,B), E(B,C), and E(A,C) be their corresponding susceptible sites, respectively. Let also $E'(A,B) \subset E(A,B)$, $E'(B,C) \subset E(B,C)$, and $E'(A,C) \subset E(A,C)$ be the largest three rectangular open point sets with the characteristic that they share the same left and right coordinates. Assume that a defect of size δ is placed. Let $C_{AB}(\delta)$, $C_{BC}(\delta)$, and $C_{AC}(\delta)$ be the corresponding subsets of critical regions of $C'_{AB}(\delta)$, $C'_{BC}(\delta)$, and $C'_{AC}(\delta)$, respectively, obtained in the same manner as indicated in the definition of critical regions. Then $C_{AC}(\delta) \subset (C_{AB}(\delta) \cup C_{BC}(\delta))$.

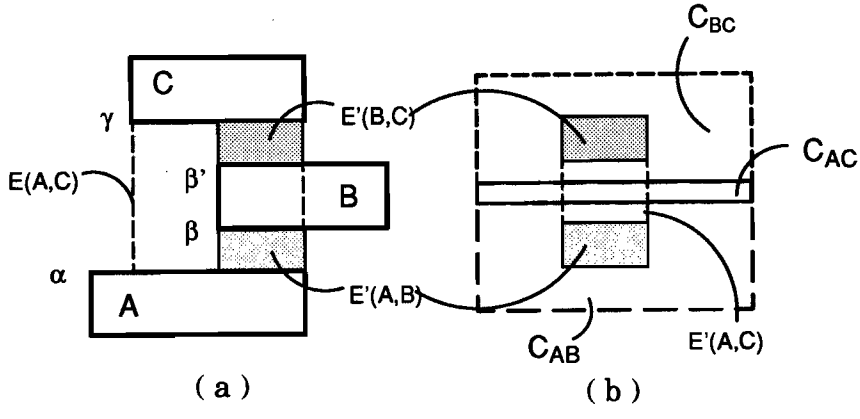


Figure 2.9. Theorem 2.6. (a) Susceptible Sites. (b) Extracted Critical Regions

Proof: Let α, β, β',and γ be four maximal horizontal line segments of A, B, and C, respectively, such that the interior of A is below α, the interior of B is above β and below β', and the interior of C is above γ. Suppose also that the four line segments are comparable at a closed interval [a,b] of the domain of points of the abscissae. Assume that

the following order is imposed $pos(\alpha) < pos(\beta) < pos(\beta') < pos(\gamma)$. Let (q_1,q_2), (r_1,r_2), and (s_1,s_2) be the corner points of the lateral susceptible sites E'(A,B), E'(B,C), and E'(A,C), respectively, expressed as $q_1 = (a\ , pos(\alpha))$, $q_2 = (b\ , pos(\beta))$, $r_1 = (a\ , pos(\beta'))$, $r_2 = (b\ , pos(\gamma))$, $s_1 = (a\ , pos(\alpha))$, $s_2 = (b\ , pos(\gamma))$. Assume that a defect of size $\delta \geq |pos(\gamma) - pos(\alpha)|$ is placed in such a form that it affects the three connected point sets simultaneously, then the critical regions $C_{AB}(\delta)$ with corner points (u_1,u_2), $C_{BC}(\delta)$ with corner points (v_1,v_2), and $C_{AC}(\delta)$ with corner points (w_1,w_2), obtained from the susceptible sites E'(A,B), E'(B,C), E'(A,C), respectively, are given by $u_1 = (a - \frac{\delta}{2}\ , pos(\beta) - \frac{\delta}{2})$, $u_2 = (b + \frac{\delta}{2}\ , pos(\alpha) + \frac{\delta}{2})$, $v_1 = (a - \frac{\delta}{2}\ , pos(\gamma) - \frac{\delta}{2})$, $v_2 = (b + \frac{\delta}{2}\ , pos(\beta') + \frac{\delta}{2})$, $w_1 = (a - \frac{\delta}{2}\ , pos(\gamma) - \frac{\delta}{2})$, $w_2 = (b + \frac{\delta}{2}\ , pos(\alpha) + \frac{\delta}{2})$. We can notice that $w_1 = v_1$ and $w_2 = u_2$; since the bottom coordinates of u_1 and u_2 are smaller than v_1 and v_2, respectively, it follows that $C_{AC}(\delta) \subset (C_{AB}(\delta) \cup C_{BC}(\delta))$. This property of the critical regions is known as the *proximity effect* [76].

□

Chapter 3

Single Defect Single Layer (SDSL) Model

A novel method to asses deterministically the sensitivity of layouts to spot defects is presented in this chapter [56,57]. The models for catastrophic faults considered are unintended bridges and unintended cuts related to patterns in one layer. The classical prototype of this construction,in the case of bridges, consists of three steps (see also chapter 2): (1) Extend all patterns by half the defect size; (2) Compute all the mutual intersections of the extended patterns; (3) Compute the area of the union of all intersections. Applying the scanline principle and assuming N line segments of the original mask patterns leads to an algorithm with asymptotic complexity $N^2 \log N^2$, a bound which is sharp in particular for large defect sizes. Our approach, based on the new concept of "susceptible sites" reduces this complexity to $N \log N$. Moreover, only two scans are necessary to extract all "susceptible sites" which then are used to compute the "critical areas" for a whole set of points in a domain of defect sizes. Under a UNIX-C environment an implementation has been created which actually exhibits the theoretically predicted gain in speed. Complex layouts can be analyzed under interactive operating conditions on standard workstations.

Prior to this method, critical area extractions have been based on approximations, on layout simplifications, or on Monte Carlo statistical simulations. Due to the complexity of the layouts, several approaches derive a "layout image" to avoid the real extraction and computation of the critical areas. This is the case with the concepts of "virtual layouts" [39] and "equivalent layouts" [8]. These layout images are composed of parallel lines which represent statistically the width, length, and spacing of all the patterns of the actual layout. The approach cannot

point out the critical regions in the real layout, and moreover the area computed is only an approximation of the exact value.

Other methodologies find some set of patterns that is likely to fail due to a predefined defect size, however the area of those patterns, or the area contained in the space between them is not the correct critical area [21,22]. Usually the results are pessimistic estimates of the exact critical area. On the other hand, suggested approaches to extract the critical areas in complex layouts are based on a statistical Monte Carlo simulation [64]. The existing analytical methods [8] are restricted only to simple and regular layouts.

In addition to the algorithms describing the construction of critical areas for SDSL models, section 3.6 presents a defect-sensitivity analysis of three different layout styles implemented for a same combinational function. It is discussed how defect-sensitivity results can be interpreted and how different manufacturing conditions affect yield.

3.1 Theory of Critical Regions for SDSL Models

In SDSL models critical areas are extracted per IC layer. Therefore, the failure primitives considered are only two, namely:

— the *bridge* (joining patterns unintendedly)

— the *cut* (breaking patterns unintendedly)

3.1.1 Single-layer susceptible sites

Suppose now that we deal with two non-intersecting active patterns $a \subset L$ and $b \subset L$, for $L \in \mathcal{L}$.

Let *a* and *b* be as in Fig. 3.1a Then, in Fig. 3.1b, S_1 , S_2 , S_3 , S_4, S_5 and S_6 represent *internal vertical susceptible sites for cuts*, and S_7 and S_8 represent *internal corner susceptible sites for cuts*. Similarly, $E_1 = (S_1 , S_2)$, $E_2 = (S_3 , S_2)$, $E_3(S_4,S_6)$, and $E_4 = (S_4 , S_5)$ in Fig. 3.1b, represent *external vertical susceptible sites for bridges* in L if there are no other points of active patterns in L intersecting them. $E_5 = (S_3 ,S_6)$ is an *external corner susceptible site for bridges in* L.

To be able to take into account the failure criterion for patterns of some layer L_i, a *sensitivity factor*, σ^i, is needed. This sensitivity factor

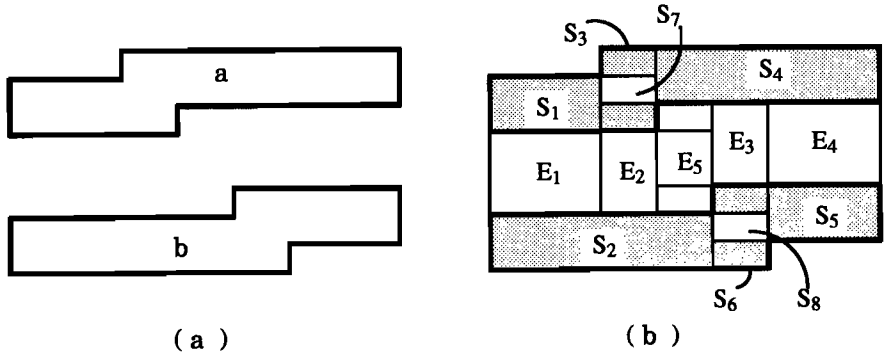


Figure 3.1. (a) Two active patterns identifying two different wiring trees. (b) Susceptible sites for bridges and cuts.

determines the geometrical situations in which a defect can be catastrophic. For the case of bridges it determines the percentage of the magnitude of an external susceptible site that has to be intersected before the two patterns are considered bridged by some protrusion defect. For the case of cuts it determines the percentage of the magnitude of the internal susceptible site that has to be covered before the pattern is considered cut by some intrusion defect.

3.1.2 Critical regions for bridges

A function can be written now, in terms of susceptible sites, defining the SDSL *geometrical failure criterion for protrusion defects* as

$$\Phi_{bridge}(E,\sigma^i) = \left\{ \sigma^i \, Mag(E), \quad 0 \le \sigma^i \le 1 \right. \tag{3.1}$$

where σ^i is the sensitivity factor of layer L_i. E is the external susceptible site where the defect occurs.

The failure criterion can physically be interpreted as follows. When $\sigma^i = 0$ a minimum catastrophic defect size has to be equal to Mag(E). This is the case for defects making a physical bridge between any two patterns. For $0 < \sigma^i \le 1$, a minimum catastrophic defect size has to be $Mag(E)(1 - \sigma^i)$. This case arises when a protrusion defect causes an electrical bridge between any two patterns without having to physically intersect them.

Making Φ_{S_1} and Φ_{S_2} equal to Φ_{bridge} in eqs. (2.6) and (2.7) the *vertical lateral critical region for bridges* in L_i between patterns *a* and *b* at lateral susceptible sites S_1, S_2 and E_1, is established if eq.(2.6) is satisfied, and it is found according to eq.(2.7).

3.1.3 Critical regions for cuts

The function describing the SDSL *failure criterion for intrusion defects in* L_i is expressed as:

$$\Phi_{cut}(S,\sigma^i) = \{\sigma^i \, Mag(S), \quad 0 \le \sigma^i \le 1 \tag{3.2}$$

where σ^i is the *sensitivity factor* of layer L_i, and S is the associated internal susceptible site where the defect takes place.

The physical meaning of this failure criterion is as follows. When $\sigma^i > 0$, defects smaller than Mag(S) are catastrophic. This situation appears when the remaining conducting area around the intrusion defect is so small that whenever the current flows through it, the area will be blown and the pattern will be cut anyway. When $\sigma^i = 0$, defects are catastrophic when their size is at least Mag(S).

Once more, substituting Φ_{S_1} and Φ_{S_2} by Φ_{cut} in eqs. (2.6) and (2.7), the *lateral critical region for cuts* in *a* is established if eq. (2.6) is satisfied, and it is found according to eq. (2.7). *Corner critical regions for intrusion defects* can be derived from eq. (2.9) if eq. (2.8) is satisfied.

3.2 Computation of Critical Areas for SDSL Models

The steps involved in the computation of critical areas are outlined now and described later in more detail. As one has access only to the layout and not to the IC, and since there is a one to one correspondence between hard-structures and soft-structures, the strategy is based on a verification performed on the layout. The basic steps of this strategy are mainly a layout extraction to obtain the susceptible sites, the creation of critical regions from the susceptible sites, and the computation of the critical areas themselves.

Step 0. For some mask M_j form new point sets by computing the union of the dark fields. Decompose them into line connected segments and attach to each line segment an identification

Step 1. Sweep the layout horizontally and vertically to extract, from the pre-processed mask, all the susceptible sites for bridges and cuts. Store them in so called "susceptibility structures", one for each kind of fault.

Step 2. For every defect size defined in the range of sizes traverse the "susceptibility structures" . The coordinates of areas sensitive to bridges, or cuts, are obtained by shrinking the abscissae (for sites obtained from the horizontal sweep), or the ordinates (for sites obtained from the vertical sweep), or both abscissae and ordinates (for corner sites) of the related susceptible sites. See Fig. 3.2.

Step 3. For every defect size compute the total critical area per type of fault as the union of the individual critical regions found in step 2.

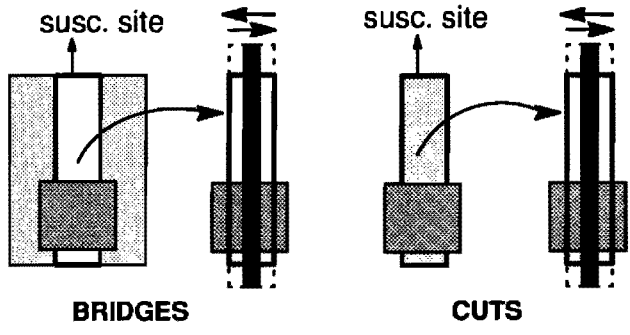


Figure 3.2. The critical regions for both bridges and cuts are found by shrinking the susceptibles sites.

3.3 Extraction of SDSL Susceptible Sites

All line segments are classified as belonging to two types. The horizontal (vertical) line segments are of type *BEGIN* when the interior of their connected point set is above (to the right) of the line segment. Similarly, the line segments are of type *END* when the interior of the connected point set is below or to the left of the line segment. Each vertical line segment is specified by its x-coordinate and the y-values of the lower and upper endpoints. Each horizontal line segment is similarly specified by its y-coordinate and the x-values of its left and

similarly specified by its y-coordinate and the x-values of its left and right endpoints. Horizontal and vertical line segments are stored in two different data structures.

The extraction of susceptible sites is based on the principle of the scanline algorithm [3, 4]. Moreover, two orthogonal layout sweeps are performed: a bottom-up and a right-left sweep that cover all the susceptible sites parallel to the scanline. The bottom-up sweep, or VERTICAL sweep, scans the data structure with the horizontal line segments. The right-left sweep, or HORIZONTAL sweep, scans the one with the vertical segments. As the algorithms for finding susceptible sites for bridges and cuts are very similar only the explanation for bridges follows. The algorithm is given in Algorithm 3.1.

Suppose now that the vertical sweep is being performed. Let $P = \{p_1, \cdots, p_{N_{lines}}\}$ be the set of horizontal line segments of some M_j, T an initially empty set used to maintain the scanline order, and L an initially empty auxiliary set, all of them lexicographically sorted by x and y coordinates. Let S be a set to store the susceptible sites found. The main loop of the algorithm sweeps a scanline through the set P and stores every swept *END* line segment in the set T. Whenever a *BEGIN* line segment is encountered, its comparable sections with respect to the line segments of T are obtained first. For every comparable section a new susceptible site is made and stored in the set S only if the identification number of the *BEGIN* and *END* line segments are different. Based on theorem 2.5, the nearest line segments, of T, to the left (predecessor) and to the right (successor) of the endpoints of the *BEGIN* line segment are obtained as well. If these lines exist the corresponding left and right corner susceptible sites are created.

Lateral susceptible sites are labeled as VERTICAL, or HORIZONTAL, depending on the sweep in which they are found. Corner susceptible sites are labeled as CORNER. The labeling is necessary because for all three types of susceptible sites the critical regions are computed differently.

The set T is updated in such a form that only the comparable sections of the intersected line segments are deleted. Theorem 2.6 guarantees that these sections are no longer necessary to create critical regions with other line segments ahead of the scanline position.

Algorithm 3.1. Creation of susceptible sites for bridges

```
make_site( left , right , bott , top ,sweep)
creates a susceptible site with the given coordinates and labels it according
to "sweep".

line( left , right ,pos)
creates a new line segment

foreach ( p_i ∈ P) {
  if (type of p_i ≡ ℰ𝒩𝒟)
    T = T ∪ { p_i }
  else {                    /* find line segments that are comparable and put them in L */
    L = { α_i ∈ T | id of p_i ≠ id of α_i ∧ (l( α_i ) ≤ l( p_i ) ≤ r( α_i ) ∨ l( α_i ) ≤ r( p_i ) ≤ r( α_i )) }
    foreach ( α_i ∈ L) {                    /* make lateral susceptible sites */
      s ← make_site(max(l( α_i ),l( p_i )),min(r( α_i ),r( p_i )),pos( α_i ),pos( p_i ),sweep))
      S = S ∪ {s}
      L = L - { α_i }
      T = T - { α_i }                       /* update T */
      if (l( α_i ) < l( p_i )) {
        v ← line(l( α_i ),l( p_i ),pos( α_i ))
        T = T ∪ {v}
      }
      if (r( α_i ) > r( p_i )) {
        v ← line(r( p_i ),r( α_i ),pos( α_i ))
        T = T ∪ {v}
      }
    }
                        /* find line segments to the right of p_i and put them in L */
    L = { α_i ∈ T | (l( α_i ) > r( p_i ) ∧ pos( α_i ) < pos( p_i )) ∧ (∀ pos( α_i ) > pos(α_j)))}
                                                                    j≤i
    foreach ( α_i ∈ L) {                    /* make corner susceptible sites */
      s ← make_site(r( p_i ),l( α_i ),pos( α_i ),pos( p_i ),CORNER))
      S = S + {s}
      L = L - { α_i }
    }
  }
}
```

Fig. 3.3 illustrates in a sequence of captions how the external susceptible sites are formed. Each caption shows the actions taken at each scanline position of Fig. 3.3a. At scanline position 1 (Fig. 3.3b) the $\mathcal{ENDD}$ line segment is installed in T. At position 2 (Fig. 3.3c), the installed line segment is split, the comparable section is deleted from T, and a susceptible site is formed. At position 3 (Fig. 3.3d) the new $\mathcal{ENDD}$

line segment is installed in T. At position 4 (Fig. 3.3e) new susceptible sites are made, and all comparable sections are deleted from T.

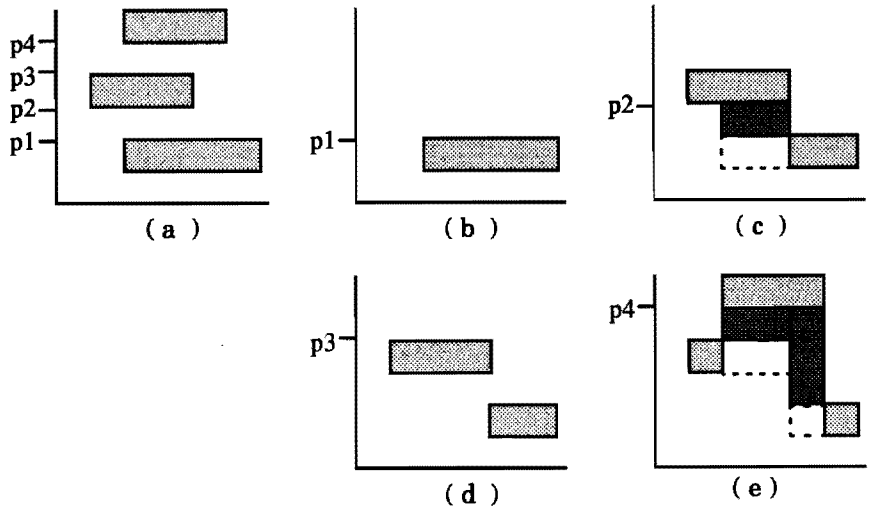


Figure 3.3. Creation of susceptible sites. *BEGIN* line segments are at positions p2 and p4, *END* line segments are at positions p1 and p3.

The extraction of susceptible sites for breaks is essentially the same except that in the algorithm the *BEGIN* line segments are the ones that are stored in the set T, and instead of processing line segments of different identification numbers, the line segments must have the same id's.

3.4 Computation of SDSL Critical Areas

The task concerning the computation of critical areas is split in two phases. Namely, creating the critical regions for each defect size, and then computing the critical area.

The algorithm developed to find the critical regions is the same for bridges and for cuts. However the sets used to store the critical regions must be independent, one for each type of fault. Let $S = \{s_1, \cdots, s_{N_{sites}}\}$ be the set of the susceptible sites in ascending lexicographical x and y order. The main loop of the algorithm traverses the set S and computes the critical regions one by one. Every line segment of a constructed critical region is saved in the set $C[N_{defect}]$ indexed by the defect size.

Algorithm 3.2. Creation of critical regions

```
top (A), bott (A)
return the ordinate of the right and left corner points of a rectangle A, respectively

right (A), left (A)
return the abscissae of the right and left corner points of a rectangle A, respectively

new_line(left,right,pos)  creates a new line segment

foreach (defect_size) {
  Δ = defect_size/2 + Φ                    /* Φ is the failure criterion */
  foreach ( s_i ∈ S) {
    switch ( s_i .label) {
    HORIZONTAL:                            /* shrink the abscissae */
      if ( right ( s_i ) - Δ ≤ left ( s_i ) + Δ) {
        c ← new_line( right ( s_i ) - Δ, left ( s_i ) + Δ,bott( s_i ) - Δ)
        C[defect_size] = C[defect_size] ∪ {c}
        c ← new_line( right ( s_i ) - Δ, left ( s_i ) + Δ,top( s_i ) + Δ)
        C[defect_size] = C[defect_size] ∪ {c}
      }
      break
    VERTICAL:                              /* shrink the ordinates */
      if (top( s_i ) - Δ ≤ bott( s_i ) + Δ) }
        c ← new_line( left ( s_i ) - Δ, right ( s_i ) + Δ,top( s_i ) - Δ)
        C[defect_size] = C[defect_size] ∪ {c}
        c ← new_line( left ( s_i ) - Δ, right ( s_i ) + Δ,bott( s_i ) + Δ)
        C[defect_size] = C[defect_size] ∪ {c}
      }
      break
    CORNER:                                /* shrink both abscissae and ordinates */
      if ( right ( s_i ) - Δ ≤ left ( s_i ) + Δ ∧ top( s_i ) - Δ ≤ bott( s_i ) + Δ) {
        c ← new_line( right ( s_i ) - Δ, left ( s_i ) + Δ,top( s_i ) - Δ)
        C[defect_size] = C[defect_size] ∪ {c}
        c ← new_line( right ( s_i ) - Δ, left ( s_i ) + Δ,bott( s_i ) + Δ)
        C[defect_size] = C[defect_size] ∪ {c}
      }
      break
    }
  }
}
```

This procedure is repeated until the range of defect sizes is exhausted. Worth noticing is that each defect size has its unique set of critical regions in the layout, and that in fact the critical regions compose a

"critical mask" specific for the defect size. The algorithm is shown in Algorithm 3.2.

It can be seen from the algorithm that the critical regions for any defect size are found straightforwardly from the susceptible sites. And also, no matter how large the defect size is the critical regions are extracted in a time proportional to the number of susceptible sites.

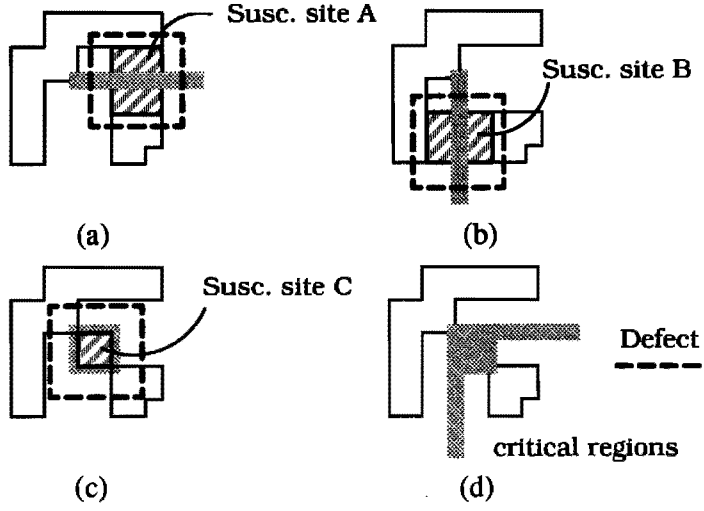


Figure 3.4. Creating critical regions for bridges from their susceptible sites. (a) Susceptible sites from the vertical sweep. (b) Susceptible sites from the horizontal sweep. (c) Corner susceptible site. (d) Critical regions formed.

The example of Fig. 3.4 will help to visualize the creation of critical regions. Let us consider the case of critical regions for bridges. Assume two L shaped conductors, running parallel to each other, with space s between them and that a defect of size $\delta > s$ is placed among them. The susceptible sites for bridges are identified as A and B and C. Susceptible site A was obtained in the vertical sweep, see Fig. 3.4a, thus their ordinates are shrunk. Susceptible site B was obtained in the horizontal sweep, see Fig. 3.4b, hence the abscissae are shrunk. Susceptible site C is a corner susceptible site therefore both abscissae and ordinates are shrunk, see Fig. 3.4.c. The resulting critical regions are shown in Fig. 3.4d.

Preparata [62] presented an algorithm to find the area of the union of a set of rectangles. This is the algorithm used to compute the critical

areas. As the algorithm is well known, it only will be sketched briefly. The implementation runs a bottom-up scanline in the critical mask. Let $C = \{c_1, \cdots, c_{N_{regions}}\}$ be the set of horizontal line segments of the critical regions lexicographically sorted by y and x coordinates. At any instance of c_i the total area is updated by evaluating the area of the union of the rectangle's sections that lie in the plane strip $pos(c_i)$ and $pos(c_{i-1})$. The length of the intercept of the scanline in the strip with the union of the critical regions' sections is determined by means of a segment tree. See. Fig. 3.5 for an illustration.

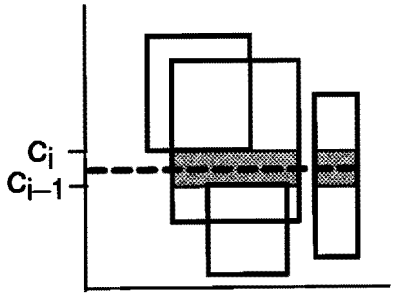


Figure 3.5. Computation of the area of the union of a set of rectangles

3.5 Complexity Analysis

Table 3.1 presents the time complexity analysis of each of the steps of this method. The nomenclature used is N for the number of line segments of the preprocessed layout, and k for the number of susceptible sites. The number of susceptible sites is dependent on the style of the IC artwork, in the worst case $k \rightarrow N$. Let #d be the number of defect sizes, then the total time complexity to find the critical areas for a range of defect sizes, assuming worst case, is $O(N + N\log N + \#d(N(1 + \log N))$. No results of similar tools have been presented in the literature as to make comparisons [64]. A hypothetical case in which the critical areas are found according to Algorithm 2.1 of chapter 2 will be assumed. The steps will be, 1) growing the patterns, 2) finding overlaps of the grown patterns, and 3) computing the areas. Table 3.2 shows the ideal time complexity analysis. In this case $\kappa(\delta)$ represents the number of overlaps, as a function of the defect size δ, that were created in the "pattern

Table 3.1. Time complexity analysis of the proposed method

Step	Operation	complexity
1)	Susceptible sites	k + NlogN
2)	Critical regions	k
3)	Critical areas	klogk

expansion" process of step 1. Notice that as the defect size increases $\kappa(\delta) \rightarrow N^2$ since most patterns will be intersecting each other. For this algorithm, the total time complexity to find the critical areas for a range of defect sizes, assuming also a worst case, is $O(\#d(N(1 + \log N) + N^2(1 + \log N^2)))$.

Table 3.2. Time complexity analysis for the ideal case

Step	Operation	complexity
1)	Pattern expansion	N
2)	Overlap search	$N\log N + \kappa(\delta)$
3)	Critical areas	$\kappa(\delta)\log(\kappa(\delta))$

Comparing both complexity analyses one can see that the new method is NlogN and depends only on the geometry of the layout. Algorithm 2.1 is dependent on both the defect size and the layout geometry. For very large defect sizes, this method is superior. Furthermore, the NlogN complexity and defect size independency of this novel method provides an approach for interactive applications.

3.6 Design Defect-Sensitivity and its Impact on Yield

The previous algorithms were implemented in a system aimed at layout yield analysis [52]. Based on this system, an experiment for defect-sensitivity analysis was carried out by implementing a combinational function in three different layout styles, namely a Programmable Logic Array (PLA), a Transistor Matrix (TM) [23], and a Standard Cells place and route approach (STD) [85], see Fig. 3.6. The layouts were designed for an NMOS technology of 6μm of minimum resolution features. Table 3.3 shows the total area and dimensions of each layout in (μm units).

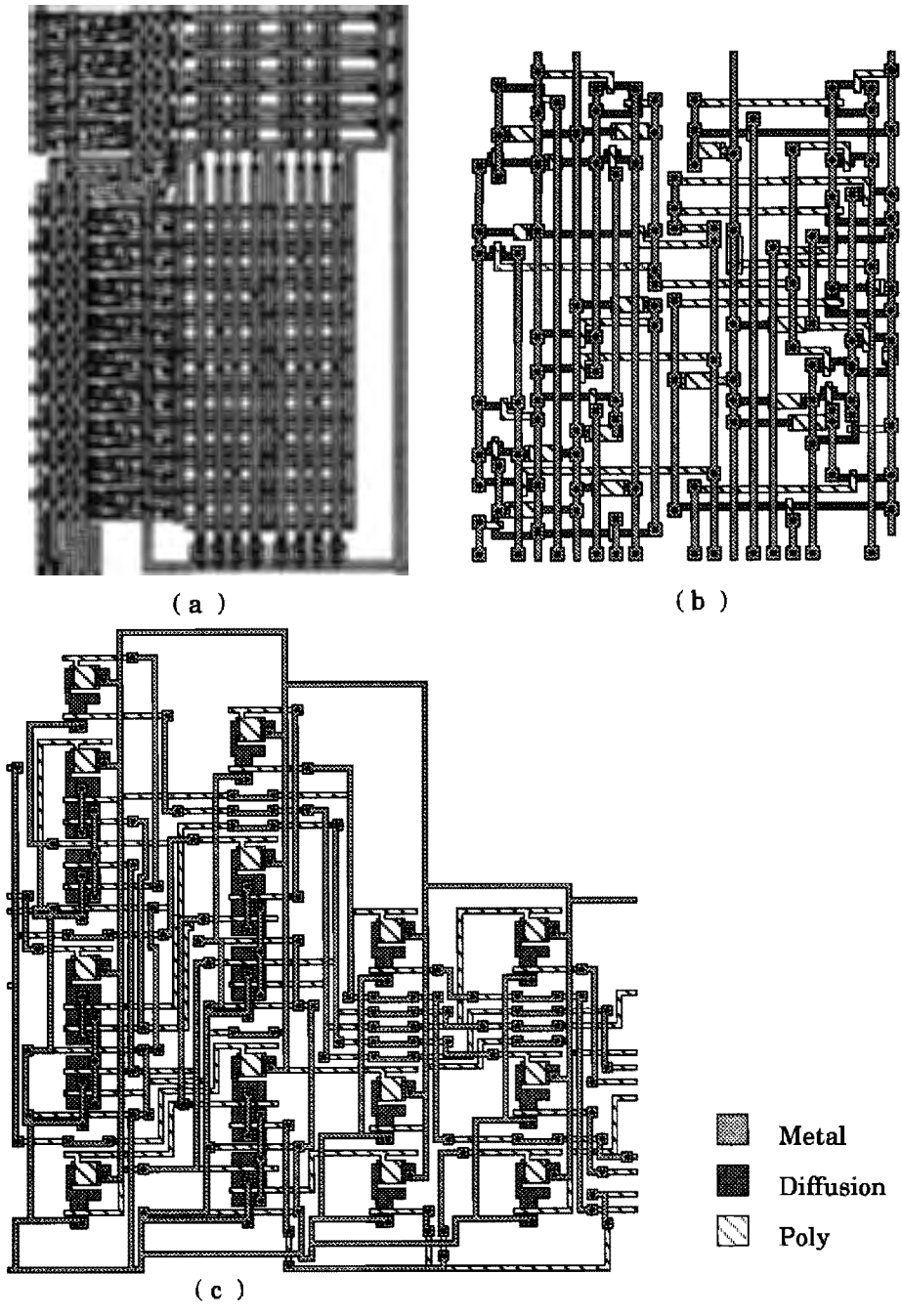


Figure 3.6. (a) PLA. (b) TM. (c) STD

Table 3.3. Layout dimensions

Layout	Vertical Dim. (μ)	Horizontal Dim. (μ)	Area (μ^2)
PLA	767	503	385801
TM	390	453	176670
STD	771	756	582876

To give the reader a feeling of how the critical regions are displayed Fig. 3.7 shows the poly layer of the TM and its critical regions for bridges and cuts for defect sizes of 30μm and 12μm respectively.

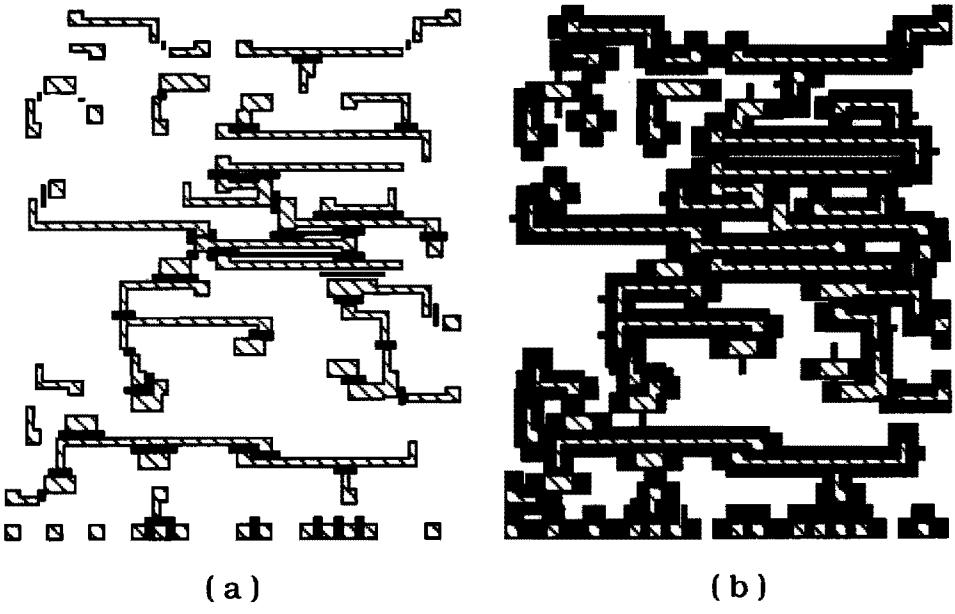


Figure 3.7. Critical regions are shown in black.(a)Bridges (b)Cuts.

3.6.1 Sensitivity analysis

Each layout was analyzed for defects in the range from 1 to 100μm. The sensitivities for bridges and cuts of each mask are shown in Fig. 3.8. The following can be observed from the figures:

+ ***Metal Mask (BRIDGES).*** The TM is the best compromise for small defects in the range from 6 to more or less 15μm. This is mainly because it is the smallest layout and thus the total critical area is less than the other ones. However as the defect size increases, the sensitivity rises with a steep slope due to the fact that the metal mask is laid out in a very regular manner, and because the spacing between lines is small. It can be seen that the PLA follows the same trend but with a less steep slope, mainly due to a relaxed spacing between wires and also due to a smaller number of adjacent lines which reduces the chances of a bridge among them. On the other hand the STD is the best compromise for very large defects. The layout style is very relaxed, the four channels of metal wires are laid out very distant from each other, and also notice the right top corner of the layout where there is a big unutilized space.

+ ***Poly Mask (BRIDGES).*** Once more the TM is the best for small defects ranging from 6 to 25μm. It can also be seen that even for large defects the mask is quite tolerant. The reason it appears as a non-regular mask is because wires are used mainly to form transistors, and as interconnectors only when it is necessary. The PLA shows a very regular pattern and thus the sensitivity rises with a steep slope. Notice that its sensitivity is almost twice as much compared to TM. The crossing in the curves is because the TM's area is smaller. The STD exhibits a quasi regular pattern especially in the feed-throughs. This aspect is worthwhile noticing because one might think that because of the large empty spaces it would be more tolerant to small defects.

+ ***Diffusion Mask (BRIDGES).*** The STD layout is in a column style. Almost all the patterns of a column are joined together, and the space between columns is big. This is why the sensitivity is extremely low. On the other hand, the style for the PLA is very regular, but with large distances between lines. The TM is not regular, the compactness of the layout however puts the lines closer. It is interesting to notice these two aspects since the two effects seem to be equivalent from the sensitivity point of view. The PLA appears to have a higher sensitivity because of the input and output buffers which are not included in the TM. The slope is very slanted because even when the lines exhibit regularity they are very

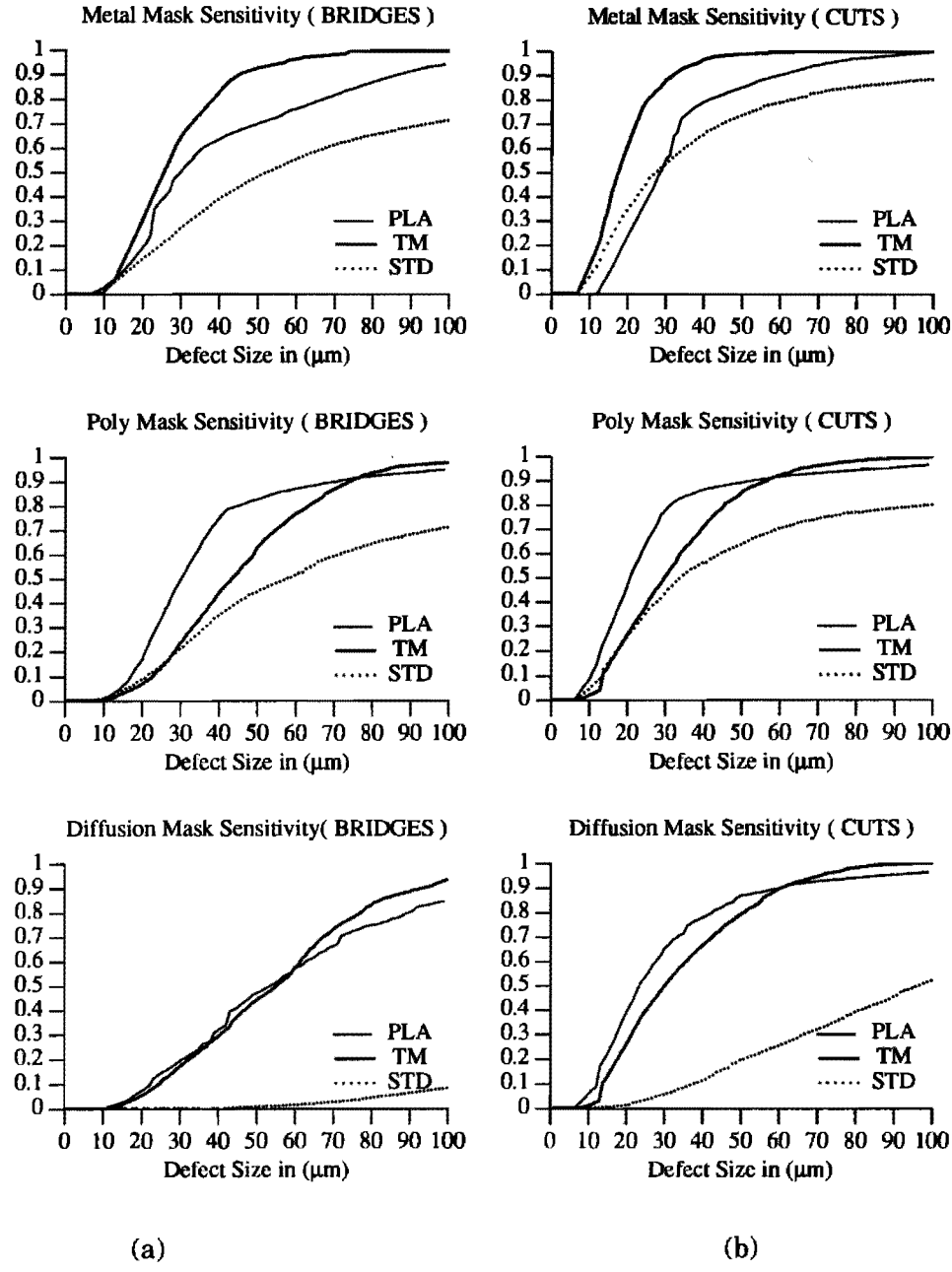


Figure 3.8. (a) Layout Sensitivity to bridges. (b) Layout Sensitivity to cuts.

interconnected, i.e. many U shapes, hence catastrophic bridges are less likely to happen.

+ ***Metal Mask (CUTS).*** In this case the PLA is the best compromise because its lines are wider than in the other two layouts. Notice the crossing in the curves of the PLA and the STD. This is because the channel's lines in the STD are widely separated which reduces the chances of a defect cutting more than one line at the same time, whereas in the PLA even when the lines are wider they are closer to each other. The TM simply cannot tolerate large defects.

+ ***Poly Mask (CUTS).*** The TM is the best compromise thanks to the scarce appearance of the lines. The abrupt jump for defects in the range of 12 to 15μm is a reflection of many square patterns in which the critical area is twice as much, to be precise in the form of a cross as shown in Fig. 3.7. The PLA shows again a steep slope due to the regularity in its layout. The STD proves to be the best for large defects due to the empty spaces.

+ ***Diffusion Mask (CUTS).*** In the STD approach the use of gates makes the ratio of the transistors bigger in order to compensate delays. This results in wider patterns which are less likely to be cut by small defects, a fact which is reflected in its sensitivity. The PLA's and TM's sensitivities are very similar due to the reasons explained above.

3.6.2 Yield analysis

The sensitivity analysis gives an insight of the way the masks are laid out. It reveals the endurance of the masks for different defect sizes. However, the probability of occurrence of each defect size is not the same. Furthermore, in practice the probability that a defect of a very large size occurs is almost neglible and hence the sensitivity analysis cannot reflect what would happen to the layout in a manufacturing environment, whereas the layout probability of failure (POF) does. To this end three different defect size distributions for the metal, diffusion, and poly masks were created to observe their impact on the layout, see Fig. 3.9. The distributions were characterized according to the model presented in [16]. The distribution for the diffusion mask was made to have a long tail for large defect sizes, the distribution for the poly mask was forced to peak at defects larger than the minimum resolution of

6μm, and finally, the one for the metal mask was tuned to represent a mature process.

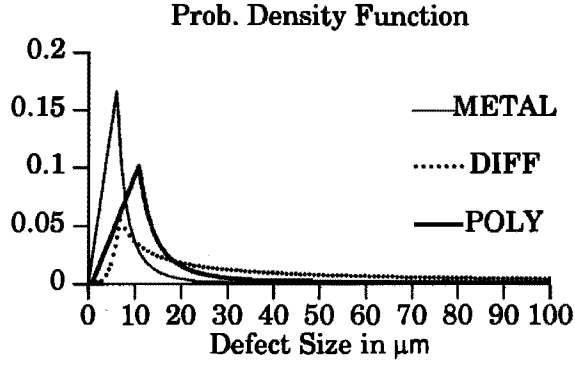


Figure 3.9. Defect Size Distribution

The layout probability of failure for bridges and cuts is shown in Fig. 3.10.

+ ***Metal Mask (BRIDGES).*** Something that is worthwhile noticing here is the similarity in the POFs of the PLA and STD. It is customary to think that the larger the layout the more likely it is to fail. In fact the curves show the contrary. The explanation is that the PLA uses long lines in a very uniform pattern even when they are used to connect just one transistor. This style increases the risk of catching unnecessary defects along the lines. The STD uses the lines also to interconnect, however they are not in a regular style and furthermore there are many empty spaces among them. The right shift in the curve of the TM, with respect to the other two, confirms its safetiness for small defects although for large defects it is quite unreliable.

+ ***Poly Mask (BRIDGES).*** A remarkable aspect is that the occurrence of defects in the size range from 15 to 30μm seems not to affect drastically the TM despite the defect size distribution peaks at 10μm. Once more, the regularity in the PLA is its major drawback. The STD with its empty spaces is more reliable to large defect sizes.

+ ***Metal Mask (CUTS).*** Since the probability of having large defects is very small the PLA proves to be the safest design due to

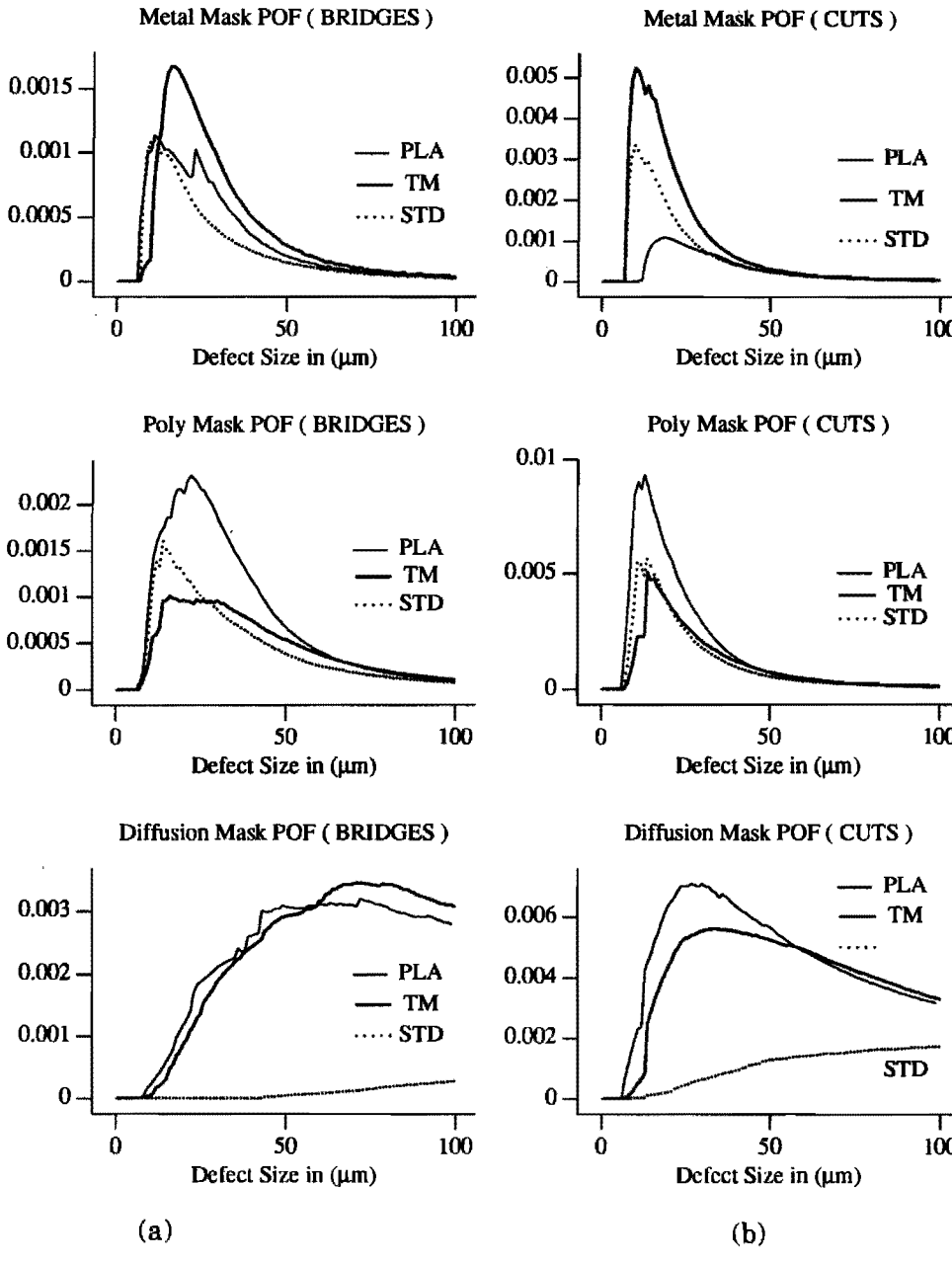


Figure 3.10. Layout Probability of Failure (POF).(a) Bridges. (b) Cuts.

its wide lines. The STD and the TM are very similar due to the fact that their lines are narrow (6μm). The TM shows a higher POF because the lines are laid out uniformly and because of the compactness of the layout.

+ ***Poly Mask (CUTS).*** The sparseness in the lines of the TM and the quasi-regular lines of the STD results in two layout styles which have more or less the same probability of failure, although for small defect sizes the TM is better. The PLA has once again the highest POF.

+ ***Diffusion Mask (BRIDGES & CUTS).*** We can see here the dramatic differences in layout style reliabilities when they are not correctly "tuned" for a defect environment in a manufacturing line. Notice that even though the probability of occurrence of large defects is big the STD style remains surprisingly low whereas for the PLA and the TM is very high.

The curves of the defect size distributions are normalized in such a way that the integral over all defect sizes for any curve yields a value "1". That means that POF curves are based on the assumption that a defect of any size and any type occurs on the layout with probability "1". The next step is to investigate the POF situation for different values of defect densities. The probability of failure was integrated for defect sizes from 1μm to 100μm. The simulated yield was modeled according to the yield model of eq. 1.10 for defect densities varying from 1 to 10000 defects/cm^2, with no gross yield losses, and a clustering parameter, α, equal to 0.9. The results obtained are shown in Fig. 3.11 and the curves speak for themselves.

The total layout yield, presented in Fig. 3.12, is computed according to eq. 1.11. One remarkable conclusion is that the largest layout is not always the most likely to fail, as it is depicted in the figure. On the other hand, regular layouts, like the PLA are not always very reliable. This aspect should be taken into account since PLAs are frequently used in controllers and these modules play an important role in many designs.

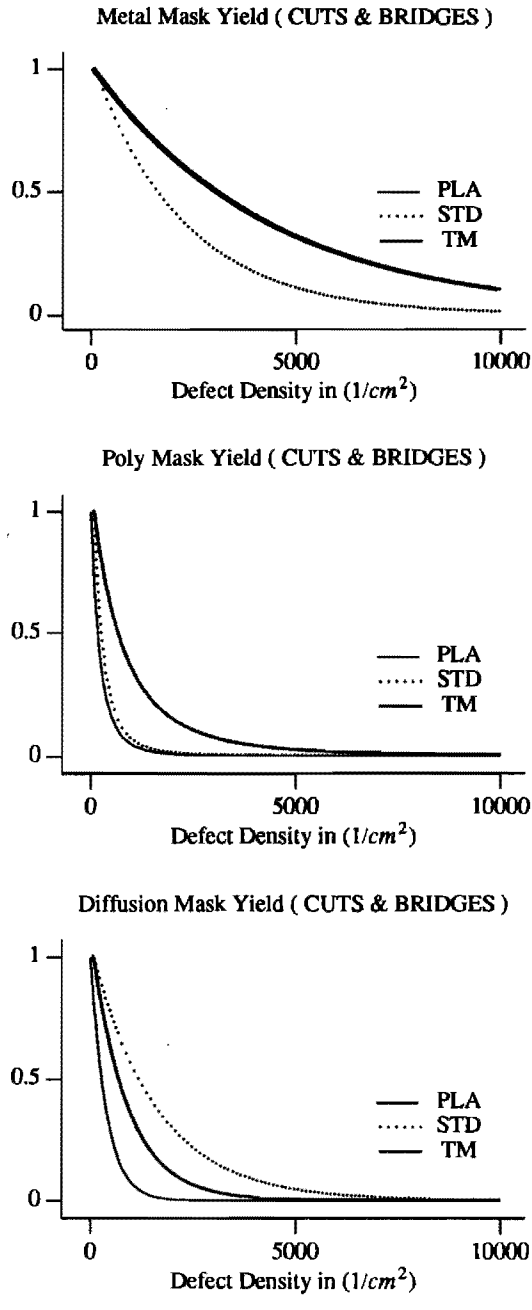


Figure 3.11. Mask yield

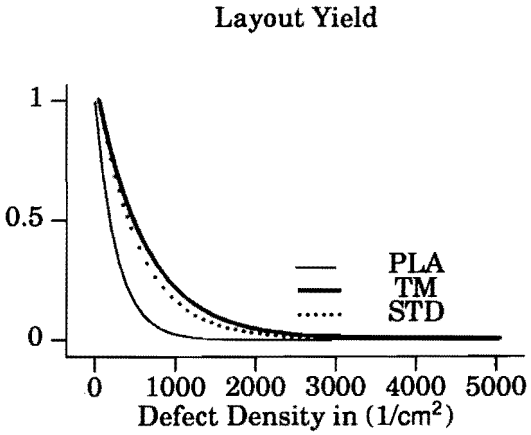


Figure 3.12. Final Layout yield

Chapter 4

Single Defect Multiple Layer (SDML) Model

SDSL models neglect any interrelationships between different masks, as is the case with transistors, and also neglect those defects which do not cause either a short or a break. Yet if the defect falls in the poly-diffusion area of a transistor it can be fatal even if it does not totally break the geometrical pattern. It is thus not sufficient to extract single-layer critical areas if either an accurate yield prediction or a realistic layout to fault extraction are desired.

In this chapter a method to construct multilayer critical areas deterministically is presented [54]. These critical areas are established on the theoretical basis of defect semantics and on the concept of "susceptible sites" presented in chapters one and two, respectively. Based on these foundations, a system comprising several algorithms was developed. In principle, these algorithms maintain simultaneously as many scanlines as the number of layers in such a way that it is possible to keep track of the vertical and horizontal effects of defects. The extracted areas are a function of the geometrical patterns in the layers, of their electrical significance, of their relationship to patterns in the same or other layers, and of course of the defect size. Moreover, the approach is technology and defect independent.

The chapter is organized as follows: the first three sections present the general theory to model spot defects in multiple layers; sections four to ten show a strategy to extract the critical regions, and section eleven presents some results.

4.1 Critical Regions for Protrusion Defects

Consider now two state clauses $\omega,\mu \in \Sigma_{tech}$. Assume three hard-structures $R_1(\omega)$, $R_2(\omega)$, and $R(\mu)$, mutually nonintersecting, and also that two protrusion defects, d_j^p and d_k^p, originated in layers L_j and L_k, respectively, are present. Suppose now that we deal with the active patterns $a \subset L_i$ originating from the partial hard-structure $R_1(\omega) \mid L_i$, $b \subset L_j$ originating from $R(\mu) \mid L_j$, and $c \subset L_k$ originating from $R_2(\omega) \mid L_k$. Assume also that the protrusion defect, d_j^p, is capable of affecting the functional behavior of $R_1(\omega)$ and $R(\mu)$ at *a* and *b*, respectively, and that the size of such defect is δ. See Fig. 4.1 for an illustration of these hard-structures. It is important to mention that the layer structure of any defect d^p is not explicitly given or even immaterial. The only thing that matters is the size of the defect on a particular layer and a condition concerning its pattern overlap in terms of the failure criterion.

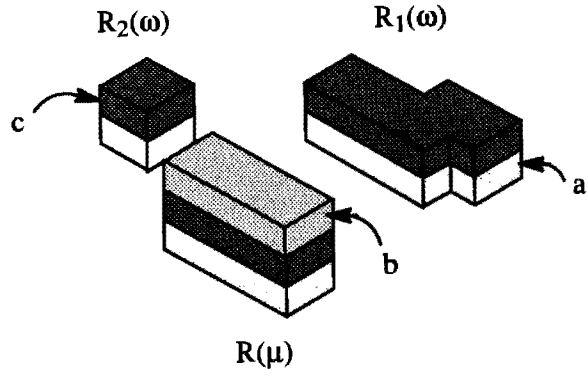


Figure 4.1. Three mutually nonintersecting hard-structures.

Let patterns *a*, *b*, and *c* be represented as in Fig. 4.2a. E_1 , in Fig. 4.2b, represents a *multilayer vertical susceptible site in* L_i *for protrusion defects in* L_j. In a similar way, E_2 is a *multilayer corner susceptible site for protrusion defects in* L_k.

Before going into details, let us first study the geometrical conditions under which a protrusion defect of some pattern a_i can be catastrophic on some pattern a_j. These are the following ones: *i)* The protrusion of a_i has to span over pattern a_j. This case is typically exemplified by a protrusion of say polysilicon spanning over a diffusion pattern such that a parasitic transistor is formed. *ii)* The protrusion of a_i has to

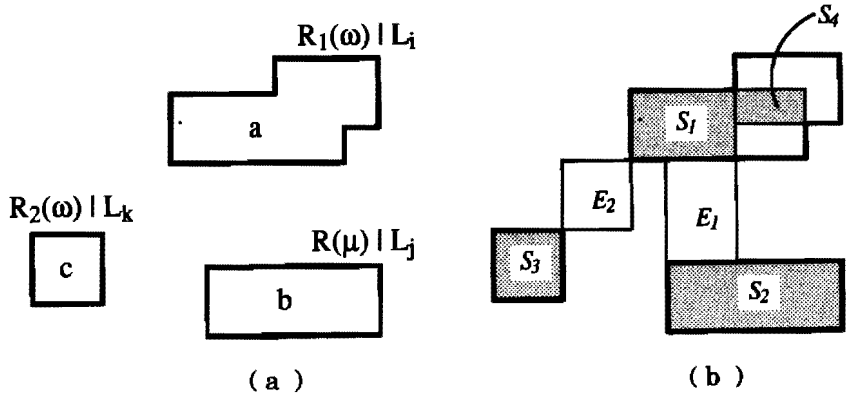


Figure 4.2. A multilayer situation is depicted in which three active patterns belonging to three different layers are characterized by two kinds of hard-structures. (a) Pattern characterization. $a \subset L_i$ by $R_1(\omega) \mid L_i$, $b \subset L_j$ by $R(\mu) \mid L_j$, and $c \subset L_k$ by $R_2(\omega) \mid L_k$. (b) Multilayer susceptible sites.

intersect at least the boundary of pattern a_j. This is the typical case for protrusion defects making a physical bridge between two or more patterns, and *iii)* The defect comes closer than a certain distance to pattern a_j. This case arises due to electrical phenomena such as cross-talk between the patterns.

To be able to consider the situations mentioned above, the concept of sensitivity factors for single layers, presented in section 3.1.2, is extended to multiple layers. Let σ_μ^{ij} be a *multilayer sensitivity factor*. σ_μ^{ij} determines defect conditions for a state clause μ provided that a defect originating in L_i affects a pattern of μ at a layer L_j . In other words, σ_μ^{ij} determines how much area on a_j has to be covered by protrusion in a_i, or how close to a_j the defect must be located. Furthermore, σ_μ^{ij} is a function of the layers involved in the defect mechanism and of the particular state clause of the hard-structure that is affected. In general, for a hard-structure consisting of say n layers, n^n different values of σ_μ^{ij} can be specified. Each of these *multilayer sensitivity factors* determines the conditions of protrusion defects in every layer affecting each one of the layers of the hard-structure. A function can now be written, in terms of susceptible sites, defining the *geometrical failure criterion for protrusion defects* as

$$\Phi_{prot}(E,S,\sigma_\mu^{ij}) = \begin{cases} \sigma_\mu^{ij}\ \mathrm{Mag}(E) & 0 \le \sigma_\mu^{ij} \le 1 \\ \sigma_\mu^{ij}\ \mathrm{Mag}(S) & -1 \le \sigma_\mu^{ij} < 0 \end{cases} \tag{4.1}$$

where σ_{μ}^{ij} is the sensitivity factor of layer L_j given the state clause μ and provided L_j is affected by protrusion defects originating in L_i. S and E are the internal and external susceptible sites where the defect occurs. In this function, negative values of σ_{μ}^{ij} cover the above mentioned case *i*, and positive values cover cases *ii* and *iii*.

Substituting Φ_{S_1} and Φ_{S_2} in eqs. (2.6) and (2.7) by the corresponding Φ_{prot} of *a* and *b*, the *vertical lateral critical region for protrusion defects* in L_j between patterns *a* and *b* at lateral susceptible sites S_1 , S_2 and E_1 , is established if eq.(2.6) is satisfied, and it is found according to eq.(2.7).

Similarly, assume that the protrusion defect, d_k^p, is capable of affecting $R_1(\omega)$ and $R_2(\omega)$ at *a* and *c*, respectively, and that the size of such defect is Δ. Then the *corner critical region for protrusion defects* in layer L_k between patterns *a* and *c*, at the corner susceptible site E_2 , and lateral susceptible sites S_1 , S_3 , is also found by substituting Φ_{S_1} and Φ_{S_3} by the corresponding Φ_{prot} in eq. (2.9) if the conditions of eq. (2.8) are satisfied.

The critical regions for those two cases are illustrated in Fig. 4.3 for $\sigma_{\mu}^{ij} = 0.3$, $\sigma_{\omega}^{ij} = -1$, $\sigma_{\omega}^{ik} = 0.0$, $\sigma_{\omega}^{kk} = 0.0$, $\delta = 4.5$ units, and $\Delta = 2.5$ units.

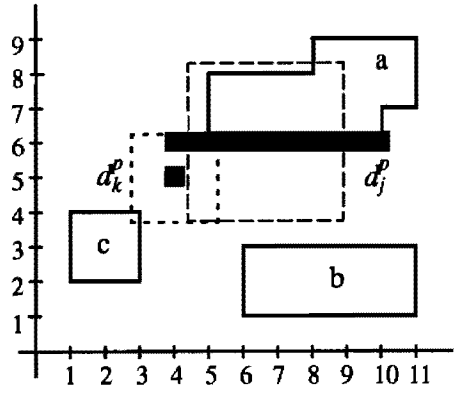


Figure 4.3. Multilayer critical regions for protrusion defects

When creating critical regions, besides the pattern extension proportional to $\frac{\delta}{2}$, an additional extension, proportional to the failure criterion of the patterns has to be considered. For the ease of modeling and simplification of the algorithms for computing critical areas, the

shape of this section is always approximated as a rectangle with length equal to $\text{Ramp}(\min(\Phi_{prot}, \Phi_{prot}'))$ and width equal to $\delta - s + \Phi_{prot} + \Phi_{prot}'$, where Ramp is the standard "ramp function" defined as

$$\text{Ramp}(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases} \tag{4.2}$$

and Φ_{prot} and Φ_{prot}' are the failure criteria of any two patterns a and b, respectively. See Appendix 2 for a detailed derivation of this section of the critical region and its error relative to the total critical region of the bridge.

4.2 Critical Regions for Isolated Spot Defects

Isolated spots are a special case in which a missing or an extra piece of material is present in the layer but it only affects layers other than the one of origin and even more, their effect is only vertical. A typical example is an isolated spot of missing thick oxide in a polysilicon-metal crossing inducing a bridge between both conductors. Another example is an isolated extra spot of polysilicon, completely covering a diffusion pattern, transforming the "diffusion conductor" into a parasitic transistor.

Consider now a state clause $\mu \in \Sigma_{tech}$. Assume an active pattern $a \subset L_i$ originating from the partial hard-structure $R(\mu) \mid L_i$. Let the pattern be depicted by the connected point set *a* of Fig. 4.2a. Assume now that an isolated spot defect, d^s, in L_j, $i \neq j$, affects *a*, and that the size of such defect is δ.

In this case the internal susceptible site S_1, in Fig. 4.2b, is a *multilayer vertical lateral susceptible site in* L_i *for isolated spot defects* in L_j if in the open rectangle S_1 no other points of active patterns of the layer where the isolated spot has its origin are allowed. This condition prevents from establishing redundant susceptible sites as could be the case of an isolated spot defect of polysilicon over a polysilicon-metal crossing.

The function describing the *failure criterion for isolated spot defects* in L_j is defined as follows

$$\Phi_{spot}(S, \sigma_\mu^{ij}) = \sigma_\mu^{ij}\, \text{Mag}(S), \quad 0 \le \sigma_\mu^{ij} \le 1 \tag{4.3}$$

where σ_{μ}^{ij} is the *multilayer sensitivity factor* of layer L_i related to the state clause μ and given that L_i is affected by isolated-spot defects originated in L_j. S is the associated internal susceptible site where the defects takes place.

The function is interpreted as follows. For $0 < \sigma_{\mu}^{ij} < 1$ a catastrophic defect size can be less than the magnitude of the susceptible site. As an illustration of this case consider a spot defect of the implant layer, in an NMOS technology, entirely spanning over the gate area of an enhancement transistor such that the transistor is turned into a depletion one. In the extreme case of $\sigma_{\mu}^{ij} = 1$ any defect at least intersecting the boundary of the susceptible site can be catastrophic. This situation is exemplified by pinholes in the thin oxide of a transistor's gate area. For $\sigma_{\mu}^{ij} = 0$ the defect size has to be at least Mag(S) in order to be catastrophic.

Substituting Φ_{S_1} and Φ_{S_2} by Φ_{spot} in eqs. (2.6) and (2.7), the *lateral critical region for isolated-spot defects* of *a* is established if eq. (2.6) is satisfied, and it is found according to eq. (2.7). *Corner critical regions for isolated-spot defects* can be derived from eq. (2.9) if eq. (2.8) is satisfied.

A lateral and a corner critical regions for isolated-spot defects are illustrated in Fig. 4.4 for $\sigma_{\mu}^{ij} = 0$ and $\delta = 2$ units.

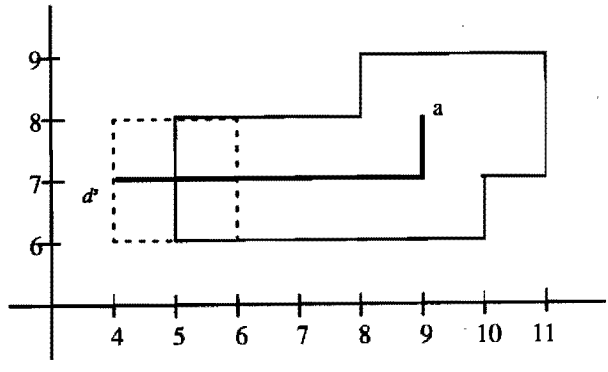


Figure 4.4. Multilayer critical regions for isolated spot defects

Isolated spot defects do not need any additional extensions due to the end effects because a defect centered ahead of the critical region will never physically intersect the pattern.

4.3 Critical Regions for Intrusion Defects

Unlike isolated spots and protrusion defects, intrusion defects only affect patterns in their layer of origin. Consider a state clause $\mu \in \Sigma_{tech}$. Assume an active pattern $a \subset L_i$ originating from a partial hard-structure $R(\mu) \mid L_i$. Let the pattern be depicted by the connected point set a in Fig. 4.2a. Assume now that an intrusion defect, d^i, affecting a is present, and that the size of such defect is δ.

In this case S_1 , in Fig. 4.2b, represents *vertical internal susceptible site for intrusion defects* in L_i. In a similar way, S4 is defined as a *corner internal susceptible site for intrusion defects.*

The *failure criterion for intrusion defects* is defined as

$$\Phi_{intr}(S,\sigma_\mu^{ii}) = \sigma_\mu^{ii}\ Mag(S), \quad 0 \le \sigma_\mu^{ii} \le 1 \tag{4.4}$$

where σ_μ^{ii} is the *multilayer sensitivity factor* of layer L_i related to the state clause μ and provided that the defects originate in the same layer L_i. S is the associated internal susceptible site where the defect takes place.

The physical meaning of this failure criterion is analogous to the one of isolated-spot defects. When $\sigma_\mu^{ii} > 0$, defects smaller than Mag(S), provided that S is associated to μ, can be catastrophic. This situation appears when the remaining conducting area, after that the intrusion defect takes place, is so small that whenever the current flows through it, the area will be blown and the pattern will be cut anyway. When $\sigma_\mu^{ii} = 0$, defects are lethal when their size is at least Mag(S).

Substituting Φ_{S_1} and Φ_{S_2} by Φ_{intr} in eqs. (2.6) and (2.7), the *lateral critical region for intrusion defects* of a is established if eq. (2.6) is satisfied, and it is found according to eq. (2.7). *Corner critical regions for intrusion defects* can be derived from eq. (2.9) if eq. (2.8) is satisfied.

4.4 Description of the System for SDML Critical Areas

Fig. 4.5 shows the overall framework for computing multilayer critical areas.

The four stages are represented by ovals, the input data to each stage is described by rectangles pointing to the ovals and the output data by

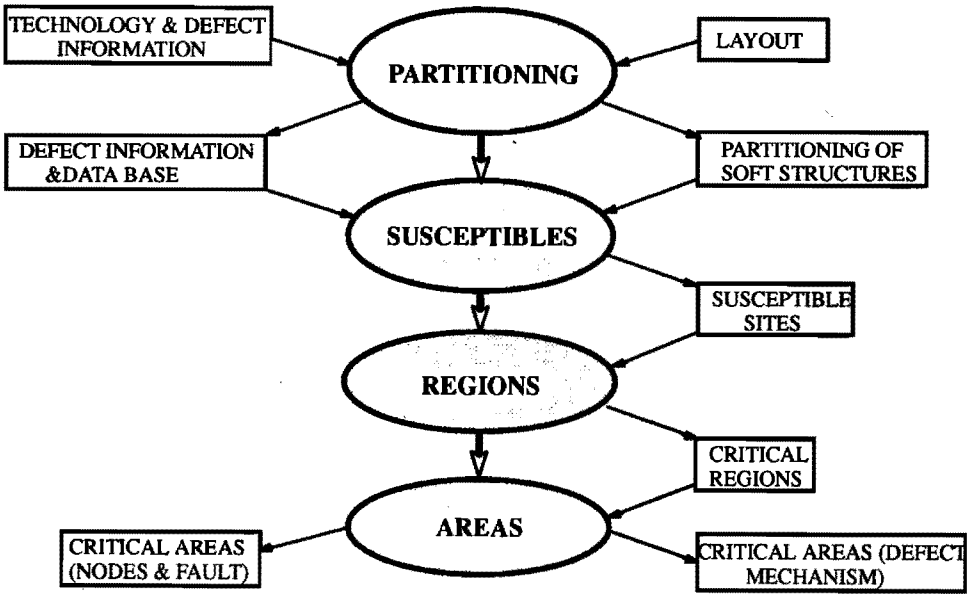


Figure 4.5. System framework for the computation of multi-layer critical areas

rectangles pointed by the ovals. The layout partition stage consists essentially in extracting soft-structures from the layout and in determining which defect mechanisms and in what manner do they affect them. The susceptible stage locates regions where defect mechanisms can potentially introduce defects in the soft-structures. The critical-region stage identifies the regions where spot defects of a known size affect the soft-structures. Finally, the area stage computes the total critical area per defect mechanism, and computes also the partial critical area per intersection of critical regions with different fault types and different electrical potentials. The former area can be used for yield prediction, and the latter for realistic fault analysis.

By convention, it is assumed in the next subsections that all line segments are classified in two types. When the interior of the corresponding connected point set lies above (below) the line segment we say that it is of type *BEGIN*(*END*).

4.5 A Spot-Defect Language

A simple language founded on the theory of section 2 was created as a user interface. This language provides information about the technology which is going to be used and the defect mechanisms, abstracted at the layout level, that may damage hard-structures. Fig. 4.6 shows its syntax in BNF notation; keywords and variables are identified as bolded and italic words, respectively.

```
1:  tech_file          ::= "(" TECHNOLOGY tech_name mask_descr ")" .
2:  mask_descr         ::= "(" MASKS "(" {masks} ")" ")"
                           {invalid_struct}.
3:  masks              ::= "<" conductor_mask ">"
                         |  "$" contact_mask
                         |  mask_name .
4:  conductor_mask     ::= mask_name .
5:  contact_mask       ::= mask_name .
6:  invalid_struct     ::= "(" SUPPRESS {invalids} ")" struct_descr .
7:  invalids           ::= "(" {mask_function}+ ")" .
8:  mask_function      ::= ["~"] mask_name .
9:  struct_descr       ::= "(" STRUCTURES {structures}+ ")"
                           {defect_descr} .
10: structures         ::= "(" struct_name {struct_function}+ ")"
                         |  "(" "(" struct_name {prime_struct}+ ")" ")" .
11: struct_function    ::= ["~"] mask_name
                         |  "<" {grouped_masks}+ ">" .
12: grouped_masks      ::= ["#"] conductor_mask .
13: prime_struct       ::= "(" prime_name {struct_function}+ ")" .
14: defect_descr       ::= "(" DEFECTS "(" min_size max_size step ")"
                           {struct_defects} ")" .
15: struct_defects     ::= "(" struct_name {mask_defects}+ ")"
                         |  "(" struct_name {prime_defect}+ ")" .
16: prime_defect       ::= "(" "(" prime_name {mask_defects}+ ")" ")" .
17: mask_defects       ::= "(" mask_name { "(" defect_type ")" }+ ")" .
18: defect_type        ::= "(" class mask_name sens_factor fault ")" .
19: class              ::= [PROTRUSION | "+"]
                         |  [INTRUSION | "-"]
                         |  [ISOLATED | "*"] .
20: fault              ::= fault_name .
```

Figure 4.6. Syntax of the Spot-Defect Language

Statements 2 and 3 describe the vector of masks $\boldsymbol{M} = (M_1, M_2, \ldots, M_{N_{mask}})$ classified by their conductor/contact properties. Masks which are neither conductor or contact are for instance implant, p-well, etc. Those state clauses which are meaningless in the technology are specified in statements 6 and 7. Statements 9 to 13 describe the set Ω of state clauses associated with mask stacks.

Each state clause, i.e. $\mu \in \Omega$, is specified by a boolean expression in which the names of the variables are replaced by the mask names. Also, since it is possible to specify a soft-structure by more than one state clause, alternate descriptions are possible through statement 10. The value of the mask variable in the definition of a state clause evaluates to "true" when the mask is specified, or evaluates to "false" when it is preceded by "~". When a mask does not appear it is taken as a "don't care".

To give information about the electrical nodes, angle brackets ,"<", are specified to indicate whether the mask remains with its same node or whether two or more masks are electrically merged. If a node is going to be split a symbol ,"#", preceding its mask is used.

Statements 14 through 20 describe the defect mechanisms that may affect every specified structure. Associated with each defect mechanism is the fault condition that arises together with its sensitivity factor. Each structure which is affected by defects has to specify which layers are sensitive and which defect mechanisms affect each one of the layers. This is explicitly shown in statement 18. If a structure is omitted from the defect specification it is assumed that the structure is insensitive to defects.

4.6 Layout Partitioning

The first task in this stage is to parse the technology file in order to create a database of defect-fault information of each state clause $\mu \in \Omega$. This database is created as an array of $2^{N_{mask}}$ elements. Each state clause is specified as a bit-vector of masks and assigned to its corresponding entries in the array. Each entry contains information of the defects that affect the soft-structure per

+ defect mechanism
+ individual masks of the state clause
+ mask where the defect manifests with associated sensitivity factors

Once the layout is read, each rectangle is decomposed in horizontal line segments. Let $\mathcal{R} = \{r_1, \ldots, r_n\}$ be the set of horizontal segments sorted lexicographically by y and x coordinates. T an auxiliary set of line segments which initially is empty, and P the set of partitions formed. Each $r_i \in \mathcal{R}$ has a bitvector such that each bit corresponds to a mask in the mask vector $\boldsymbol{M} = (M_1, \ldots, M_{N_{mask}})$. The algorithm sweeps the set $\mathcal{R}$ by

retrieving one line segment at a time. Assume that a line segment r is retrieved. Then, the partitions are formed by constructing rectangular point sets from the comparable sections between r and each line segment of T that is below r. New partitions are stored in P, non-comparable sections of r are stored in T, and each comparable section of the lines stored in T is split off from their corresponding line segments. If r is of type *BEGIN* a logical OR operation is performed between the mask-bitvectors of each of these comparable sections and r. Next, the comparable sections are reinstalled in T with their bitvectors updated. On the other hand, if r is of type *END*, an XOR logical operation is carried on, if the result of the XOR is not zero the comparable sections are also inserted in T, otherwise they are deleted from their corresponding line segment. After all the partitions are constructed, a common node assignment as it is done in layout to circuit extractors, is performed.

4.7 Extraction of Multi-Layer Susceptible Sites

Susceptible sites are obtained by performing two orthogonal sweeps in the layout. Thus, susceptible sites are found relative to their sweep. Furthermore, each susceptible site is "labeled" according to the sweep in which it was found. Corner susceptible sites are marked as "corners". This labeling is necessary because in order to determine the critical regions one must know which coordinates need to be considered, either the abscissae for horizontal sites, the ordinates for vertical sites, or both abscissae and ordinates for corner sites. The algorithm is essentially the same as Algorithm 3.1 but with some variations.

Assume now that the vertical sweep is carried on. Let T be an ordered set containing the horizontal line segments of each one of the masks of the partitions formed in the previous stage. Each line segment has information about the mask to which it belongs, its left, right and pos coordinates, the associated electrical node, and a counter that keeps track of overlapping sections between line segments. Let T be lexicographically ordered by y and x coordinates. Let $\mathcal{B}[N_{mask}]$ and $\mathcal{E}[N_{mask}]$ be two indexed sets that maintain the scanline status for *BEGIN* and *END* line segments, respectively. Each one of these sets is indexed by the mask to which the line segment belongs relative to its mask position in the mask vector $\boldsymbol{M} = (M_1, \ldots, M_{N_{mask}})$. Both $\mathcal{B}[N_{mask}]$ and $\mathcal{E}[N_{mask}]$ are initially empty.

The algorithm sweeps the set T by retrieving one line segment at a time. Every *BEGIN* line segment, say $k \in o$, $o \subset M_j$, is is installed in

$\mathcal{B}[N_{mask}]$ indexed by its mask position j and with its overlap counter initialized to one. If there are comparable sections between the line segments of $\mathcal{B}[j]$ and k, the set $\mathcal{B}[j]$ is updated in such a way that the comparable sections are split off from their corresponding line segments and reinstalled as independent line segments with their overlap counter incremented by one. $\mathcal{END}$ line segments are installed in $\mathcal{E}[N_{mask}]$ indexed by their mask position, with their overlap counter initialized to one, and with the magnitude of the associated internal susceptible site attached.

Assume now an $\mathcal{END}$ line segment k of some mask M_j. Multilayer internal lateral susceptible sites are formed from the comparable sections between k and those line segments installed in every $\mathcal{B}[i]$, $i = 1, \ldots, N_{mask}$, whose overlap counter is one. All comparable sections of $\mathcal{B}[j]$ are split off from their corresponding line segments, and are reinstalled independently only if after decrementing their overlap counters, the value of the overlap counter is bigger than zero. External susceptible sites are constructed after each internal susceptible site is established. The procedure is essentially the same, except that *i)* instead of using k as a reference, the $\mathcal{BEGIN}$ line segments of every internal susceptible site constructed from $\mathcal{B}[j]$ are used, *ii)* the set $\mathcal{E}[N_{mask}]$ is used to find comparable sections, and *iii)* external susceptible sites are established only when the associated electrical nodes are different. Corner susceptible sites are formed by finding the nearest line segments to the right (left) endpoint of k until one of those lines has its ordinate bigger than the ordinate of k, or bigger than one of the line segments already found.

Next, each susceptible site is processed to determine the possible defect mechanisms that may affect its related soft-structures.

4.7.1 Defect mechanisms

In the previous subsection it was described how to obtain susceptible sites. However, these sites are still per mask, and furthermore, since there is no knowledge about their related soft-structures it is not possible to establish which defect mechanisms are meaningful. In this section it will be described how to transform these sites into "multilayer" susceptible sites, and also how to relate them to their corresponding soft-structures. The general strategy will be described now, and later it will be addressed as to how to assess if a multilayer susceptible site is sensitive to defect mechanisms.

To construct multilayer internal susceptible sites all the established internal susceptible sites are geometrically intersected among each other to find maximal connected point sets, $S_{multiple}$, that contain subsets of susceptible sites of different masks. External multilayer susceptible sites $E_{multiple}(S, S_i)$ (where i is an index indicating each different internal susceptible) are constructed in the same geometrical way as internal multilayer susceptible sites.

Now establish a state clause μ from the result of an OR operation among the masks of a multilayer susceptible site. This state clause represents in fact the related soft-structure(s), and therefore it is possible to determine the defect mechanisms that affect each one of their masks.

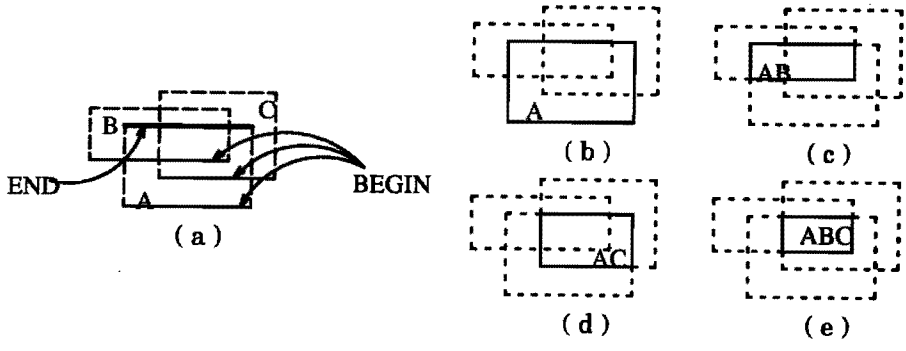


Figure 4.7. Forming multi-layer susceptible sites. (a) Three different masks. (b) Susceptible site for mask A. (c) Susceptible site for masks A and B. (d) Susceptible site for masks A and C. (e) Susceptible site for masks A, B, and C.

As an example of how to construct multilayer internal susceptible sites consider the case of Fig. 4.7a where *BEGIN* line segments from three susceptible sites related to masks, A, B and C, and one *END* line segment related to mask A are shown. Figs. 4.8b-4.8e display the constructed multilayer susceptible sites after the geometrical intersections take place.

In the next subsections it will be described how to assess if a multilayer susceptible site is sensitive to defects. Let $S_{prot}[N_{mask}]$, $S_{iso}[N_{mask}]$, and $S_{intro}[N_{mask}]$ be three initially empty sets used to store susceptible sites sensitive to protrusion, isolated and intrusion defects, respectively. These sets are indexed by the mask where the defect originates relative to its mask position in the mask vector ***M***. Each susceptible site in any

of these sets will have attached its related electrical nodes, fault types, and label for further processing of critical regions and areas.

4.7.1.1 Intrusion defects

Assume a state clause $\mu \in \Omega$ of some internal susceptible site $S_{multiple}$. For each mask M_j associated to μ, the defect-fault database is inspected to check if an intrusion defect of that mask creates a fault condition for μ. If the fault exists, the sensitivity factor is retrieved, the Mag of the susceptible site is calculated, and finally the susceptible site is stored in the set $\mathcal{S}_{intr}[j]$, otherwise the susceptible site is discarded.

4.7.1.2 Isolated-spot defects

These defects are checked analogously to intrusion defects. Assume the state clause μ of some $S_{multiple}$. For each mask M_j, in the vector $\boldsymbol{M}$, different than any of the masks related to μ, the defect-fault database is inspected to verify if μ is affected by isolated-spot defects of M_j. If two or more masks are affected by the same defect, a valid susceptible site is established either when the faults are different, or when the faults are the same but the electrical nodes are different. If the susceptible site is valid, the associated sensitivity factor is retrieved, the Mag of the susceptible site is calculated, and as a last action the site is saved in the set $\mathcal{S}_{iso}[j]$.

4.7.1.3 Protrusion defects

For some $E_{multiple}(S, S_i)$, let ω and μ be the state clauses related to S and to S_i, respectively. For each mask associated to μ, the defect-fault database is inspected to verify if a protrusion in that mask affects ω. If ω is in fact sensitive, a valid susceptible site for protrusion defects is established. Since protrusion defects have a bilateral effect, the process is reversed and now it is checked whether protrusions of ω affect μ. Susceptible sites for protrusion defects are valid only when the electrical nodes are different. Each new susceptible site is stored in the set S_{prot}. indexed by the mask where the defect originates.

4.8 Construction of Multi-Layer Critical Regions

Critical regions are found by performing geometrical operations of shrinkage-expansion on each susceptible site found. Let $\mathcal{C}_{prot}[N_{mask}][N_{sizes}]$, $\mathcal{C}_{iso}[N_{mask}][N_{sizes}]$, and $\mathcal{C}_{intr}[N_{mask}][N_{sizes}]$ be three initially empty sets used to store critical regions for protrusion, isolated, and intrusion defects, respectively. Let these sets be indexed by the mask

where the defect originates relative to its mask position in the mask vector $\boldsymbol{M}$ and by the defect size in the given domain of defect sizes.

Remember that each susceptible site has information about the associated electrical node, the corner points of the site, the label either vertical, horizontal or corner, the related fault associated to the state clause to which the site belongs, and the respective failure criterion. The procedures that are just about to be described are based on eqs. (2.5-2.8), and are similar to Algorithm 3.2.

The algorithm sweeps each set of susceptible sites by retrieving one susceptible site at a time. Assume that $S_{intr}[j]$ is being swept. Then, for each defect size δ in the range of defect sizes and for each susceptible, three possible actions are taken depending on the site's label: VERTICAL) shrink the ordinates of the corner points by half of the defect size plus the failure criterion. If the new shrunk coordinates intersect each other, then the abscissae are expanded by half of the defect size plus the failure criterion (if applicable). HORIZONTAL) The same actions as with vertical sites, except that the shrinkage takes place on the abscissae and the expansion on the ordinates. CORNER) shrink both ordinates and abscissae by half of the defect size plus the failure criterion. If both ordinates and abscissae intersect each other make a valid critical region. Each established critical region is stored in its corresponding $C_{intr}[j][\delta]$.

4.9 Computation of Multi-Layer Critical Areas

Critical areas are computed for each defect size by sweeping the sets of critical regions. Each critical region has information concerning the fault type and the affected electrical nodes. Whenever two critical regions are intersected, the intersection will be the critical region for both fault types on all related nodes. This assertion can be illustrated by an example. Assume three different critical regions A, B, and C as depicted in Fig. 4.8. Assume that A has nodes 1 and 2 and fault type "short", that B has nodes 1 and 3 and fault type "new device", and that C has nodes 2 and 3 and fault type "new device". Clearly, if the center of the defect falls in the region A only a "short" arises, if it falls in region B only a "new device" is created, and if it falls in ABC both short circuit and "new device" faults are established. Hence, each intersection of critical regions has a different fault significance which can be weighted according to the partial critical area. Therefore, for each distinct defect mechanism, partial critical areas will be computed per faults and related electrical nodes. The total critical area will

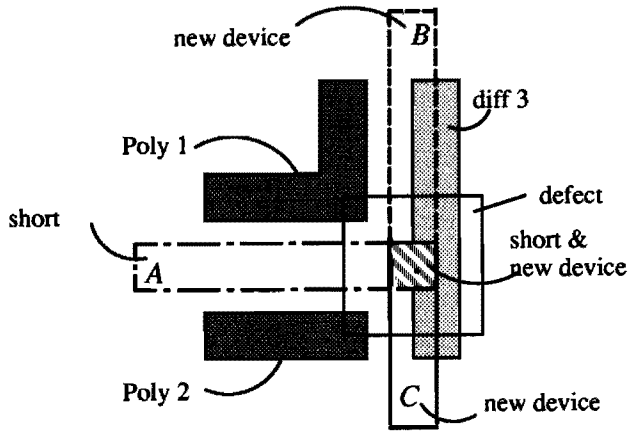


Figure 4.8. Critical regions after node and fault splitting.

simply be obtained by adding up the results of every partial critical area.

Without loss of generality let us denote the set $\mathcal{C}[N_{mask}][N_{sizes}]$ as the set containing the horizontal line segments of each of the critical regions found. Let it be indexed by each defect lexicographically sorted by y and x coordinates. Take one mask i and one defect size j. Each line segment $c \in \mathcal{C}[i][j]$ has two sets that keep track of the intersection of nodes and faults. Let us denote these sets as NODE and FAULT and their relationship to c as *c*.NODE and *c*.FAULT, respectively. Each element of these sets has information of the node number (fault type) and a node counter (fault counter) that keeps track of node (fault) multiplicity in the same line section. Let $A[2^{N_n}][2^{N_t}]$ be a matrix used to store the partial area for all possible combinations of nodes and faults, where N_n and N_t are the total number of nodes and type of faults extracted from the layout, respectively. In practice this matrix is too sparse to be used as such and also it can be very huge, therefore we use a linked list of meaningful elements instead. However, for ease of the explanations the matrix will still be used for the discussion.

The algorithm sweeps the set $\mathcal{C}[i][j]$ by retrieving one line segment at a time. Let T be an initially empty auxiliary set maintaining line segments. Assume now that a line segment $c \in \mathcal{C}[i][j]$ is retrieved. Then, for each comparable section, s, the area of the rectangle formed between c and s is computed. Next, the sets *s*.NODE and *s*.FAULT are scanned to construct two bit vectors, node and fault, such that each node and fault in s is assigned to a bit position. These bitvectors are used to index the matrix *A*[node][fault] in order to be able to accumulate the partial area for this specific combination of nodes and faults. This

partial area is also accumulated to the total critical area. If the type of c is *BEGIN* only its non-comparable sections are installed in T. Every comparable section, s, is split off from its corresponding line segments and reinstalled with the node and fault sets including the node and fault of c. If the nodes (faults) of c and s are the same, the respective counter is incremented, otherwise they are inserted into the corresponding sets of s with the respective counter initialized to one. If the type of c is *END*, the node and fault counters of each comparable section that has the same node and fault of c are decremented. If the node or fault counters become zero, the node or fault are removed from their corresponding sets. Whenever both fault and node sets are empty the associated line segment is removed from T.

4.10 Notes on Implementation

Scanline algorithms usually make use of data structures such as balanced trees, segment trees, etc [62]. These data structures are planned for minimizing time of operations such as delete, insert, and search on random accesses to the data structure's elements. However, for the applications explained in the previous sections it becomes quite difficult and laborious to perform these operations on these data structures. In most of the cases one has to deal only with comparable sections rather than whole line segments. Therefore, a simple data structure named *Static Line Array* was developed.

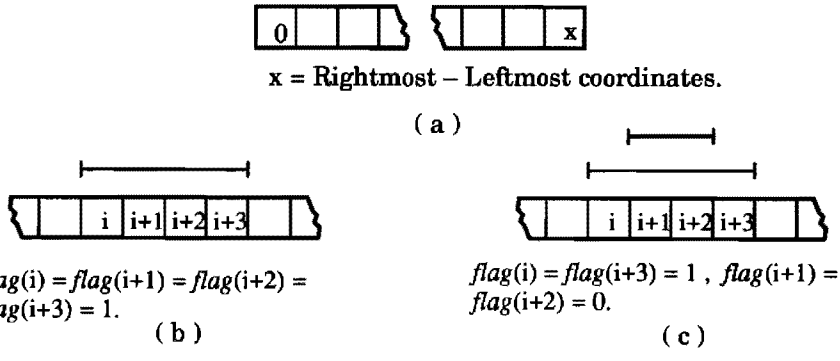


Figure 4.9. Static Line Array. (a) Data structure. (b) Insert operation. (c) Split-Delete operation.

Let SLA[N_{SLA}] be an array of N_{SLA} elements, where N_{SLA} is equal to the maximum (MAX) abscissa of the layout, or ordinate, minus the minimum (MIN) corresponding ones. If vertical layout sweeps are

performed, the abscissae are used, otherwise the ordinates. Each element of this array represents a section of a line segment of unit one. Each entry contains a *flag* that indicates whether the slot is occupied or not, the vertical (horizontal) position of horizontal (vertical) line segments, and some extra information which is related to the particular application, i.e. a bitvector of masks, nodes, faults, etc. Insert, delete and search operations become then quite simple. For instance, if a horizontal line segment (pos(α);l(α),r(α)) is going to be inserted, the flag is set on in all those i slots such that $l(\alpha) - MIN \le i < r(\alpha) - MIN$, and the position in each slot is set to be equal to pos(α). Delete operations are obviously done by resetting the flag in the corresponding slots, and search operations are done by finding those slots, in the range of the line segment of reference, whose flag is on. With the SLA simultaneous operations of insert-delete, insert-split, split-delete, search-split, etc. can be carried out. The performance using this data structure is quite fast. The Static Line Array is based on the fact that most of the line segments in a layout are of length far less than the total length of the layout [2]. So, in most of the cases the number of iterations is quite small in order to insert, delete or split a line segment. See Fig. 4.9 for an illustration of this data structure.

Let us study now the time complexity for insert, search and delete operations using the SLA. Let N be the number of line segments, b be the size of the slot, and $\lambda = \frac{MAX - MIN}{b}$ the number of slots in the SLA. The time to perform any operation for a given line segment k in the SLA is obviously $\frac{r(k) - l(k)}{b}$, hence the average time for N line segments will be $\frac{N\bar{l}}{b}$, where $\bar{l}$ is the average length of all lines obtained as $\bar{l} = \frac{1}{N}\sum_{i}^{N} r(k_i) - l(k_i)$. Thus, the average time complexity for performing operations on the SLA is O(cN). In spite of the linear relation the performance can be very bad if c is very large. One way to overcome this problem is by adjusting the size of the slot and by maintaining a linked list of line segments in every slot. This approach resembles the well known technique of hashing with collision resolution by separate chaining [27]. By using this approach the average time to insert N line segments is given by $e^{-\alpha} + \alpha$, where α is a load factor of the hash table expressed as $\frac{N}{\lambda}$ [27]. For the particular application of the SLA α is expressed as $\frac{N\bar{l}}{b\lambda}$. If b is chosen to be equal to $\bar{l}$, and N is a very large number, it can be seen that the average time complexity is O(N).

Similarly, the average time to delete or test for the existence of a line segment is given as $1 + 0.5\,\alpha$ [27] which also results in an average O(N) time complexity.

4.11 Experimental Results

This section is devoted to present the performance behavior of the system against the number of processed rectangles. All previously mentioned algorithms were implemented in C in an HP-9000/835 minicomputer. As a set of examples a series of benchmarks were implemented in a Standard Cells Place and Route approach for an NMOS technology of 6μ of minimum resolution features. The layout consists of six masks, namely: diffusion area (nd), polysilicon (np), buried contact (nb), contact (nc), implantation (ni), and metal (nm). For details on the composition of each hard-structure of this technology and the way in which each of their layers is affected refer to the technology file of Appendix 3. The computed faults together with the sensitivity factors for defects affecting the hard-structures are summarized in Tables 4.1 to 4.3.

Table 4.1. Fault conditions due to intrusion defects

Element	Sensitivity						Fault					
	nm	np	nd	nc	nb	ni	nm	np	nd	nc	nb	ni
poly_metal												
dif_metal												
poly_via	0.0	0.0		0.5			break	break		break		
dif_via	0.0		0.0	0.5			break		break	break		
buried_via		0.0	0.0		0.5			break	break		break	
poly_track		0.0						break				
dif_track			0.0						break			
metal_track	0.0						break					
enh_xtor.trans1		0.5						bad xtor				
depl_xtor.trans1		0.5				0.5		bad xtor				bad xtor

Table 4.2. Fault conditions due to isolated spot defects

Element	Sensitivity						Fault					
	nm	np	nd	nc	nb	ni	nm	np	nd	nc	nb	ni
poly_metal				1.0						short		
dif_metal				1.0						short		
poly_via												
dif_via		0.0						float_line				
buried_via												
poly_track												
dif_track		0.0					new_device					
metal_track												
enh_xtor.trans1					0.5	0.5					bad xtor	bad xtor
depl_xtor.trans1					0.5						bad xtor	

Table 4.3. Fault conditions due to protrusion defects

Element	Sensitivity						Fault					
	nm	np	nd	nc	nb	ni	nm	np	nd	nc	nb	ni
poly_metal												
dif_metal												
poly_via												
dif_via		-1.0						float_line				
buried_via												
poly_track		0.0						short				
dif_track		-1.0	0.0				new_device	short				
metal_track	0.0						short					
enh_xtor.trans1												
depl_xtor.trans1												

The examples were run for five different defect sizes ranging from 7μ to 19μ, in steps of 3μ, and for computing the total multilayer critical area per layer. This range of defect sizes provides a good characterization of the conditions prevailing in the manufacturing line, provided that the defect sizes obey the $1/x^3$ defect size distribution [74].

Obviously the advantage of the deterministic approach is the ability to process large layouts, in the order of tens of thousands of rectangles, in a relative short cpu time. Table 4.4 shows the cpu time for each benchmark. Notice that the running time is proportional to the number of rectangles in an almost linear relationship.

Table 4.4. CPU time (HP-9000/835)

Benchmark	# Rectangles	# Transistors	Time (hh:mm:ss)
f2	874	28	1:54.65
rd53	1768	65	3:55.55
radd	3091	91	7:03.28
alu2	5789	175	14:09.48
sao2	11999	281	30:30.94
9sym	15280	368	42:21.37
in6	22396	468	1:03:29.49

4.11.1 Failure analysis of a 6T-RAM cell

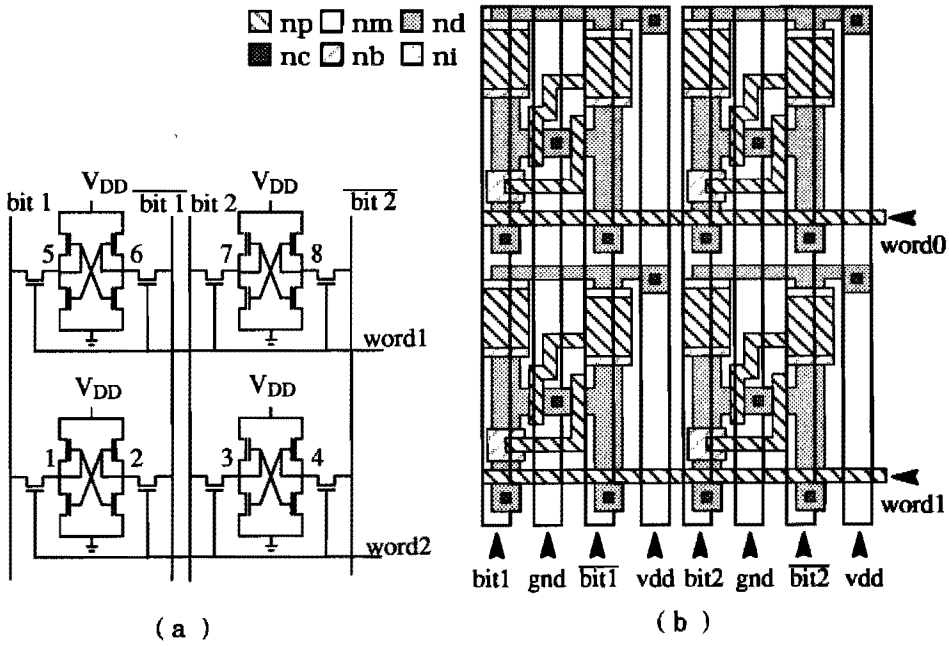
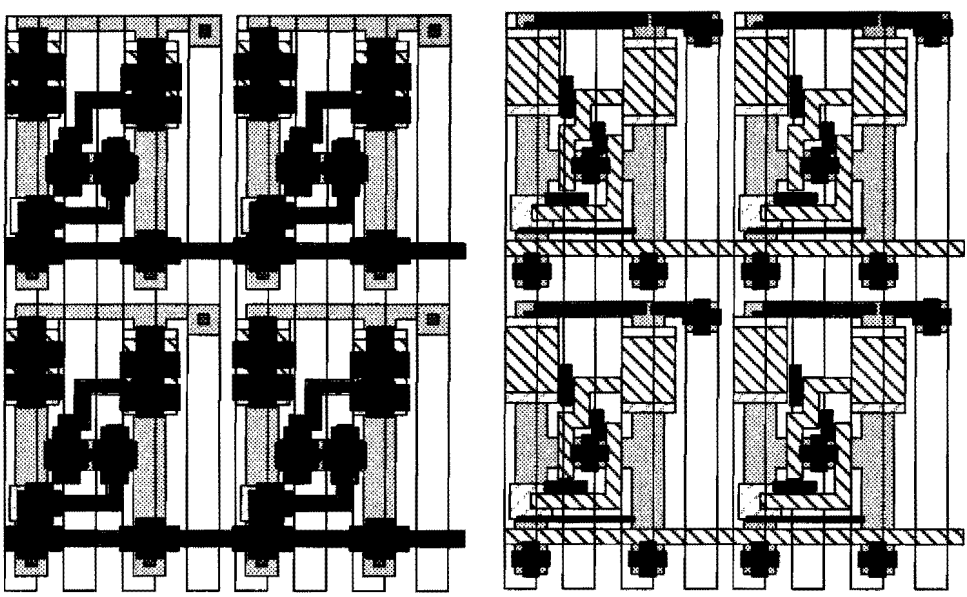


Figure 4.10. A static RAM. (a) Schematic. (b) Layout

The four memory cells shown in Fig. 4.10, were characterized for defects ranging from 0μ to 100μ using the fault conditions presented in Tables 4.1 to 4.3.

The critical regions for polysilicon spot defects of 10μ are displayed in Fig. 4.11. One can observe both vertical and horizontal effects of defects from this figure. Horizontal effects are visible when the defect affects patterns of the same polysilicon layer while vertical effects are visible

when the defect interacts with other layers. In particular, vertical effects for the intrusion defects of Fig. 4.11a appear in the form of crosses, in each via and in each enhancement transistor, and in the form of double crosses, in the upper sections of each cell in the depletion transistors and buried contacts. Fig. 4.11b shows the critical regions for protrusion and isolated defects. In this case the crosses are because isolated spots can impede the contact between metal and diffusion in the metal-diffusion vias. Also the critical regions in the upper section of each cell are the places where parasitic transistors can be formed.



(a) (b)

Figure 4.11. Critical regions for polysilicon defects of 10 μ. (a) Intrusions. (b) Protrusion and isolated spots.

The design's defect-sensitivity for protrusion, intrusion and isolated spot defects is shown in the histograms of Fig. 4.12. Worthwhile noticing is that the histograms show that defects smaller than the minimum resolution features can also be catastrophic. These defects appear in layer crossings, transistor's gate areas, vias, etc.

Two cases showing the results of defect-fault diagnosis of the memory cells were selected. The histogram of Fig. 4.13a shows the sensitivity for a short circuit among bit1, power supply, and ground.

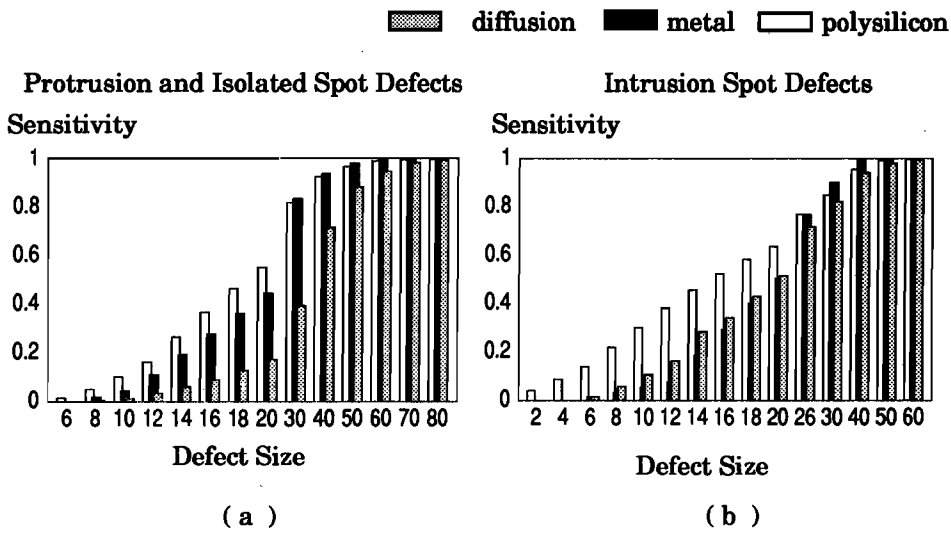


Figure 4.12. Defect-sensitivity of the static RAM. (a) Protrusion and isolated spot defects. (b) Intrusion defects.

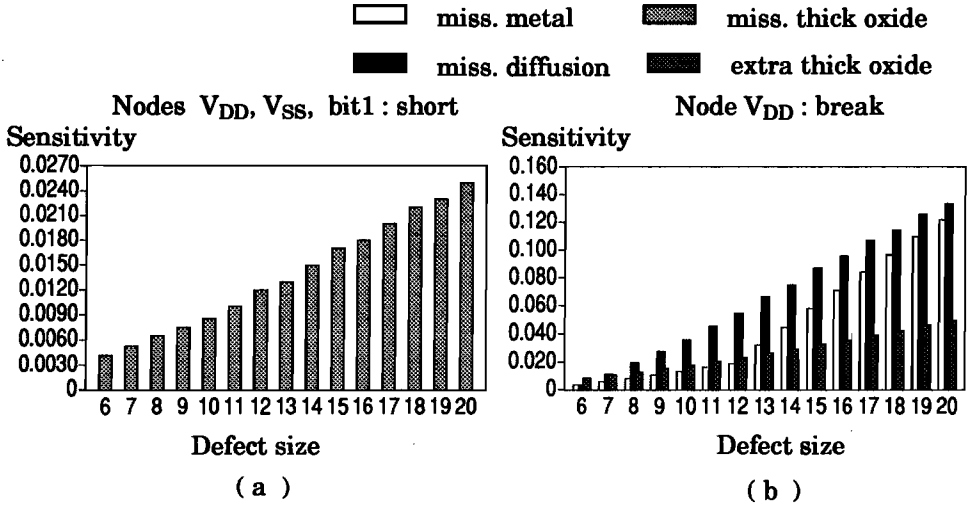


Figure 4.13. Defect sensitivity per fault and node. (a) One defect affecting more than one node. (b) More than one defect affecting the same node.

This is a case when only one defect affects several nodes inducing the same kind of fault. For this particular layout, this fault affects three nodes simultaneously when an intrusion defect of thick oxide appears in the upper left corner of each memory cell. The histogram of Fig.

4.13b, shows the sensitivity of a break between the memory cells and Vdd. This break occurs when intrusion defects of the metal or diffusion layers, or when isolated spot defects of thick oxide, in the form of extra material, appear in the via in the upper right corner of each memory cell. This case is the opposite of the previous one in the sense that three different defect mechanisms induce the same fault.

An analysis was conducted for three different defect statistics to observe the defect sensitivity of the electrical nodes due to possible variations of the manufacturing line. The analysis is based on the defect to fault collapsing technique presented in [10].

The defect size distributions were characterized according to the model presented in [16]. Every defect size distribution was forced to peak at a defect size of 6 μ. In the analysis it is assumed that every defect mechanism has the same defect size distribution. The three cases are:

Case 1: The defect size distribution obeys a $1/x^3$ law and the defect density for each defect mechanism has the same value of 2 defects/cm^2.

Case 2: The defect density distribution remains the same as in case 1, while the defect size distribution is constrained by $1/x^2$.

Case 3: The defect size distribution is the same as in case 1, but only the defect density of extra metal is increased to 6 defects/cm^2, the rest of the mechanisms remains with the same value of 2 defects/cm^2.

Case 1 represents a mature process, case 2 is chosen to show the effects of a defect size distribution allowing a higher probability of occurrence for large defect sizes, and case 3 is used to discover which nodes are sensitive to a change in the density of a particular defect mechanism.

Results for the weight variation of each node are shown in Fig. 4.14. Comparing cases 1 and 2, one can observe, as it is expected, that for case 2 all the nodes have a higher likelyhood of failure, see Fig. 4.14a. Also, Fig. 4.14b shows that not all the nodes are sensitive to variations in the metal defect density; for this particular design only the bit lines and power supply are likely to fail.

These kind of histograms are a useful guideline to endure the design against spot defects. For instance, one can try to balance or to minimize the nodal failure weights by restricting the use of the "sensitive" masks of the layout. Furthermore, with the output results of the system one can attempt to do reversed engineering. That is, given a fault one can localize in the layout which defect mechanisms and with what

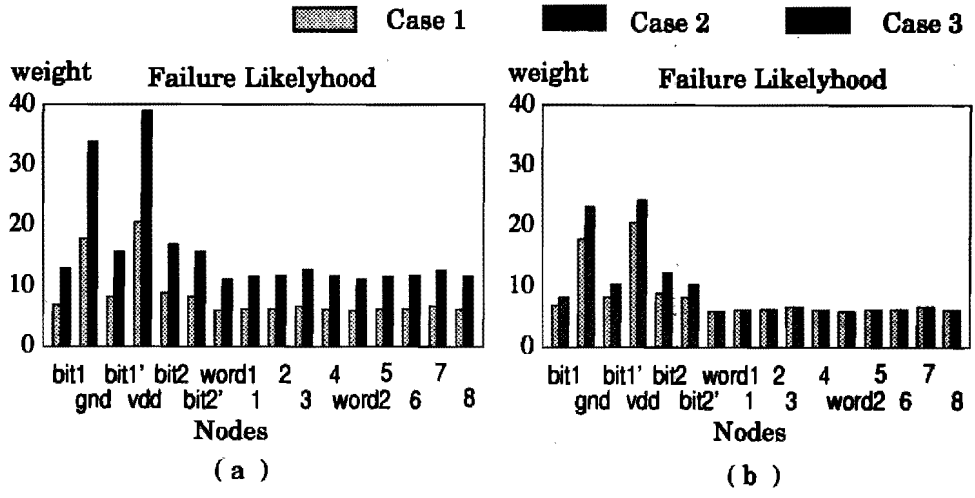


Figure 4.14. Failure likelyhood of the electrical nodes of the SRAM. (a) Variations of the defect size distribution. (b) Variations of the defect density for protrusion defects of metal.

likelyhood can they actually cause the fault.

As a last example, a yield projection as a function of the number of bits was estimated using these four memory cells. The yield model employed is eq. 1.11. It is assumed that the manufacturing conditions are according to case 1, that no gross yield loss occurs and that the value of the cluster parameter is 3.

The area is estimated as the area of the four memory cells times 1000 and 10000 for a projection for 4k bits and 40k bits, respectively. Table 4.5 shows the results of these projections. Each row shows the yield per defect mechanism. A "+" sign stands for protrusion and isolated spot defects, and a "-" for intrusions. Pinholes were simulated as extra material spots of the buried contact mask; missing spots of thick oxide as extra material spots of the contact mask. The last row presents the total yield. From the results it is obvious that the main yield detractors in this technology are defects of "nc+", "np+" and "nb+". Thus, if yield were to be improved corrective actions should take place for these kind of defects, e.g. to reduce their defect density, or to change the design in order to minimize their effect.

Table 4.5. Yield Projections for a 4k and 40k bit SRAM

Defect Mech.	4 bits	4k bits	40k bits
np-	0.999962	0.962913	0.699024
np+	0.999993	0.993019	0.933024
nd-	0.999994	0.993756	0.939830
nd+	1.000000	0.999774	0.997744
nm-	0.999994	0.993686	0.939183
nm+	0.999998	0.998384	0.983998
nc-	0.999994	0.994336	0.945233
nc+	0.999837	0.852826	0.271148
nb-	1.000000	0.999521	0.995224
nb+	0.999963	0.963566	0.703300
ni-	0.999984	0.983853	0.853041
ni+	0.999979	0.979544	0.818284
total	0.999696	0.741812	0.070775

Chapter 5

SDSL vs. SDML - A Comparative Study

This chapter presents a comparative study for yield prediction based on critical area extractions of single and multiple layers. The objective is to show how much error is incurred by using an SDSL model and a given layout design style.

Disadvantages of SDSL models were mentioned in previous chapters and are now briefly recalled,

+ The calculated critical areas account only for short/break type of faults
+ All patterns in every layer are considered only as interconnectors, even when in real artworks some portions of those "connectors" are also part of devices like transistors.
+ The effect of layers like the implant-layers of an NMOS process, or the p-well-layers of a CMOS process are not considered.
+ It is a "single-layer" theory, no interdependence between layers is considered, as could be the case of a poly-metal via where three different layers are involved.

Nevertheless, SDSL models are good to evaluate the safeness of the artwork as a function of the probability of failure of its layers, that is, whether the patterns can undesirably be broken or joined.

In the long trajectory of yield modeling it was found that by obtaining the critical areas a more realistic yield prediction can be achieved [74]. However, it has been conjectured that the extraction of single-layer critical areas can be inaccurate in predicting yield [36, 58].

The case study presented in this chapter is intended to disclose these conjectures by quantifying the existing differences in yield prediction [53]. The experiment consisted in creating layouts in a Standard Cell Place and Route approach which were then used to compute defect-sensitivities and estimates of yield.

By way of introduction, section 5.1 presents some case situations where the inaccuracy of the SDSL model is highlighted. Sections 5.2.1 and 5.2.2 cover the set up and results of the experiment. Section 5.3 presents a discussion over the obtained results.

5.1 Uncovered Situations of the SDSL Model

Some cases where SDSL models fail to detect faults other than shorts and breaks will now be pointed out. For the rest of this section consider an NMOS process with essentially five layers, L_1, L_2, L_3, L_4, L_5.

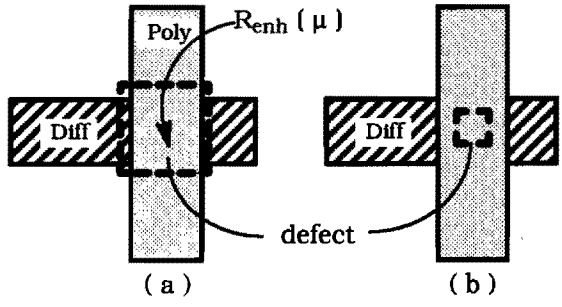


Figure 5.1. Enhancement transistor. (a) The gate is fully broken. (b) The gate is partially broken

+ ***Case 1.*** Consider the case of Fig. 5.1 where a hard-structure, R_{enh}, depicts the gate of an enhancement transistor. Let its state clause be μ = (SUBSTRATE, THIN OXIDE, POLY, **x**, **x**). Assume that the active pattern poly ⊂ L_3 originated from $R_{enh}(\mu)$ | L_3 has a width w. If an intrusion defect appears in the poly pattern, two fatal situations may arise: 1) The defect size is $\delta > w$ and breaks totally the pattern as shown in Fig. 5.1a, and 2) the defect is of size $\delta < w$ but it also breaks the pattern as depicted in Fig. 5.1b. In both cases the transistor's drain and source are shorted however no sensitive area is computed under the SDSL model for the former case.

+ ***Case 2.*** Assume now the state clauses μ and ω for the presence of poly and diffusion. Let these clauses be μ = (SUBSTRATE, OXIDE, POLY, **x**,**x**) and ω = (DIFFUSION, OXIDE, OXIDE, **x**, **x**), respectively. Consider two hard-structures R(μ) and R(ω) identifying

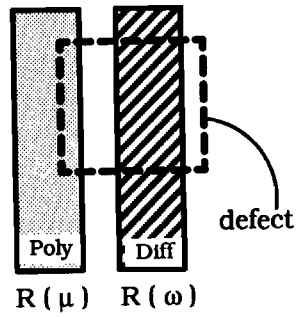


Figure 5.2. Creation of a parasitic transistor

two wires, one of poly and one of diffusion, as depicted in Fig. 5.2. If a protrusion defect of poly crosses the diffusion wire, the undesired crossing creates a parasitic transistor in series with the diffusion wire. However, since the SDSL model does not consider relationships among layers this situation is never covered.

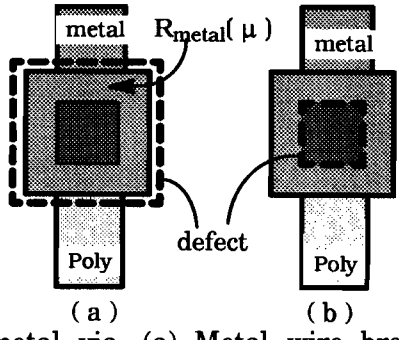


Figure 5.3. Poly-metal via. (a) Metal wire broken. (b) Metal wire partially broken

+ ***Case 3.*** In Fig, 5.3 a hard-structure "metal" $R_{metal}(\mu)$ is shown. Let its state clause be $\mu = (x, x, x, x, \text{METAL})$. Assume also that the pattern $\text{metal} \subset L_5$ originated from $R_{metal}(\mu) \mid L_5$ has a width w. If an intrusion defect is present in the metal pattern, two fatal situations may also arise: 1) The defect size is $\delta > w$ and breaks totally the pattern, see Fig. 5.3a, and 2) the defect size is $\delta < w$ but it occurs precisely on top of the area of the contact hole, see Fig. 5.3b. In both situations a circuit-break, or a floating line, occurs however no critical area for the latter situation is established under the SDSL model.

+ ***Case 4.*** Consider the situation depicted in Fig. 5.4 where a hard-structure "poly-metal crossing", $R_{cross}(\mu)$, is depicted. Let the state clause of this hard-structure be characterized by

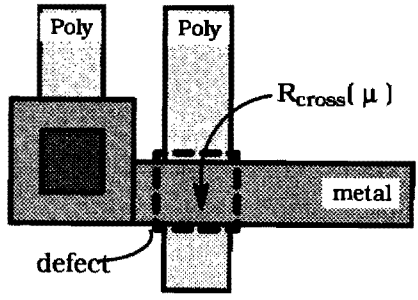


Figure 5.4. Involuntary via

μ = (SUBSTRATE, **x**, POLY, OXIDE, METAL). If an isolated spot defect of missing thick oxide occurs in this hard-structure, it is possible that the two layers L_3 and L_5 become involuntary connected. Since the SDSL model works only for single layer shorts and breaks, these kind of defects is not considered even when a potential fault is present.

+ ***Case 5.*** This example does not point out breaks or shorts but it rather projects a possible performance failure. Take into consideration the hard-structure "depletion transistor", $R_{dep}(\mu)$, depicted in Fig. 5.5. Let μ be described as μ = (IMPLANT, THIN OXIDE, POLY, **x**, **x**). Because the SDSL approach accounts only for single layer conductors, the effects that layer L_1 may have are discarded from consideration. However, the depletion transistor can be turned into a simple enhancement transistor if an intrusion defect of a significant magnitude is present in the implant.

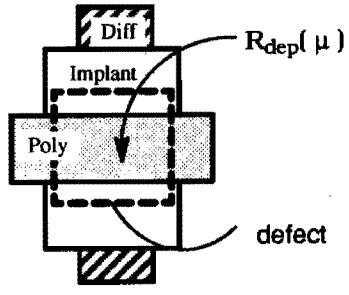


Figure 5.5. Depletion transistor affected by a spot of missing material in its implant layer.

5.2 Case-Study

The experiment consists of two main phases, 1) Defect-Sensitivity extraction from the layouts, and 2) Yield computation, on an analytical basis, founded on the results of the previous phase.

A "Standard Cells Place and Route" approach is used as the layout's design style for the experiment. The technology employed is NMOS of 6μ of minimum resolution features consisting of the following layers: Diffusion (nd), Polysilicon (np), Metal (nm), Thick oxide (nc), Thin oxide (nb), and Implantation (ni). (See Appendix 3 for a description of this technology, defect mechanisms, and fault types).

The circuits used for the analysis were obtained from a set of Logic benchmarks[1] and transformed into layout implementations by in-house software [85]. The benchmarks and their corresponding area and number of transistors are listed in Table 5.1.

Table 5.1. Benchmarks

Benchmark	Area (μ^2)	#Transistors
f2	429570	28
con1	530980	27
rd53	1027420	65
misex1	1137000	71
dc1	1193280	65
dk27	1787968	100
radd	2081856	91
co14	2249280	108
vg2	2565936	140
alu3	2779854	130
mish	3062016	106
rd73	3163264	163
5xp1	3458042	170
misg	4611636	139

It was assumed that the manufacturing conditions present a $1/x^3$ defect size distribution [74] with a same defect density for all possible defect mechanisms. In our specific technology, it is interesting to observe the effect of defects in the range from 6μ to 20μ. Defects bigger than 20μ

1. Logic benchmarks distributed by Microelectronic Center, NC (MCNC)

have a very low probability of occurrence. The employed yield model was eq. 1.11. A clustering defect parameter of 1 and no gross yield losses were assumed.

5.2.1 Set-up of the experiment

All hard-structures are characterized by a unique combination of layers, yet, not all of the hard-structures have the same sensitivity to each defect size. So obviously, it is necessary to explicitly state the sensitivity factors for each hard-structure and for each defect mechanism.

Trying to make the experiment's comparisons as fair as possible, most of the conditions for both single-layer and multi-layer extractions were set to equal values. Only two special cases were distinguished, namely, when half of the transistor's gate area was missing, and when isolated-spot defects of the implant layer cover half of the transistor's gate area. The reason for these two exceptions is because it is assumed that the performance degradation can be big enough to make the circuit fail. Table 5.2 shows the failure conditions of each element from the NMOS technology. The values present the sensitive factor of each hard-structure.

Table 5.2. Sensitivity Factors

Element	Multiple						Single					
	nm	np	nd	nc	nb	ni	nm	np	nd	nc	nb	ni
dif_via	0-	0*,-1+	0-	0-			x	x	x	0-		
poly_via	0-	0-		0-			x	x		0-		
buried_via		0-	0-		0-			x	x		0-	
metal_track	0+,0-						0+,0-					
poly_track		0*,0-						0+,0-				
dif_track		0-,1*,-1+	0+,0-					x	0+,0-			
poly_metal	0-	0-		1*			x	x		x		
dif_metal	0-		0-	1*			x		x	x		
enh_xtor.trans1		0.5-	0.5-		1*	0.5*		x	x		x	x
depl_xtor.trans1		0.5-	0.5-		1*	0.5-		x	x		x	0-

The meaningful hard-structures of the technology are identified under the column "Element". Under each of the two main columns, "Multiple" and "Single", are the six layers where defects can originate and be catastrophic to the element. Each value indicates the sensitivity factor for its corresponding defect type; columns marked as "x" are for conditions that the single-layer approach cannot cope with as compared to the multi-layer one, and columns left as blank are for defects in that

layer that have no effect on the element. A "+" sign means that the element is susceptible to protrusion defects, similarly the "-" and "*" signs are used to indicate that the element is susceptible to intrusion and isolated defects, respectively.

5.2.2 Comparative results

Tables 5.3 and 5.4 show the summarized results -for three different defect sizes- of the computed defect-sensitivity per defect mechanism. The data are presented as the mean and standard deviation of all benchmarks. Each defect mechanism shows three rows, S, M, and Δ, which stand for SDSL model, SDML model, and the error between them, respectively. The error was evaluated as follows

$$\Delta = \frac{M - S}{M} \tag{5.1}$$

Table 5.3. Sensitivity Analysis - Intrusion Defects

Mechanism	Model	Size					
		6		12		18	
		mean	std.	mean	std.	mean	std.
nm	S	0.000	0.000	0.140	0.008	0.312	0.015
	M	0.008	0.001	0.177	0.008	0.346	0.015
	Δ	1.000	0.000	0.209	0.018	0.099	0.014
np	S	0.000	0.000	0.112	0.011	0.246	0.022
	M	0.028	0.004	0.165	0.014	0.303	0.025
	Δ	1.000	0.001	0.322	0.024	0.192	0.018
nd	S	0.000	0.000	0.003	0.001	0.010	0.002
	M	0.003	0.001	0.020	0.004	0.046	0.008
	Δ	1.000	0.000	0.857	0.021	0.776	0.035
nc	S	0.000	0.000	0.000	0.000	0.003	0.001
	M	0.072	0.007	0.128	0.013	0.187	0.019
	Δ	1.000	0.000	1.000	0.000	0.984	0.006
nb	S	0.000	0.000	0.000	0.000	0.000	0.000
	M	0.023	0.004	0.050	0.009	0.083	0.014
	Δ	1.000	0.000	1.000	0.000	1.000	0.000
ni	S	0.000	0.000	0.000	0.000	0.000	0.000
	M	0.005	0.001	0.011	0.002	0.019	0.003
	Δ	1.000	0.000	1.000	0.000	1.000	0.000

It can be observed that for intrusion defects, see Table 5.3, the metal and poly layers are the most prone to be damaged and also that they are the ones that have the lowest error between both models. The error for poly is bigger because poly is used not only as an interconnect, but also as a part of transistors and vias where defects smaller than the

Table 5.4. Sensitivity Analysis - Protrusion & Isolated Defects

Mechanism	Model	Size					
		6		12		18	
		mean	std.	mean	std.	mean	std.
nm	S	0.000	0.000	0.065	0.007	0.157	0.015
	M	0.000	0.000	0.063	0.007	0.148	0.015
	Δ	0.000	0.000	-0.035	0.009	-0.056	0.011
np	S	0.000	0.000	0.027	0.007	0.072	0.019
	M	0.003	0.001	0.041	0.007	0.103	0.017
	Δ	0.927	0.073	0.352	0.112	0.316	0.104
nd	S	0.000	0.000	0.000	0.000	0.001	0.001
	M	0.000	0.000	0.002	0.000	0.009	0.002
	Δ	0.000	0.000	0.842	0.098	0.911	0.056
nc	S	0.008	0.001	0.041	0.003	0.089	0.006
	M	0.008	0.001	0.041	0.003	0.089	0.006
	Δ	0.000	0.000	0.000	0.000	0.000	0.000
nb	S	0.000	0.000	0.001	0.000	0.007	0.001
	M	0.000	0.000	0.003	0.000	0.008	0.001
	Δ	0.000	0.000	0.583	0.000	0.193	0.000
ni	S	0.000	0.000	0.000	0.000	0.000	0.000
	M	0.018	0.004	0.039	0.009	0.064	0.014
	Δ	1.000	0.000	1.000	0.000	1.000	0.000

minimum resolution features are catastrophic. Due to the layout's design style, the diffusion layer is quite insensitive yet the error between both models is big because the SDML approach can detect critical areas in the vias while the SDSL one cannot. Worthwhile noticing is that defects in the form of missing thick oxide are relevant. These defects appear in the crossings of metal and poly, and metal and diffusion, yet the limited scope of the SDSL models cannot capture them. The incurred error for defects in the thin oxide and implant layers is 100% because these defects appear in transistors which are hard-structures that cannot be recognized by the SDSL model.

For the case of protrusion and isolated defects, see Table 5.4, the metal layer's sensitivity is less prone to errors, less than 5%. Also, notice that the single-layer approach extracts more critical areas. This excess in critical areas is to be expected, consider for instance two metal wires running parallel to each other, and also consider that a poly wire interconnects them. For the single-layer approach the two metal wires are independent patterns and thus the critical area is created, yet for the multi-layer approach both metal wires have the same electrical potential and thus no critical area is established.

In the case of metal-poly and metal-diffusion vias the error between SDSL and SDML approaches is zero because the sensitivity factors for extra thick oxide defects have the same value in both approaches. Yet, there is an error present for poly-diffusion contacts despite the fact that the sensitivity factors for defects of extra thin oxide have the same value in both approaches. The reason is because the intersection between poly, buried contact, and diffusion is smaller than the geometrical pattern of the contact itself. The SDSL model tends to extract critical areas for the entire geometry of the contact while the SDML approach extracts critical areas only from the intersection of the three masks. Finally, the error for isolated spot defects of the implant layer is 100% due to the fact that the SDSL approach cannot cope with multi-layer relations -these defects appear mainly in the gate areas of enhancement transistors.

Table 5.5. Yield Prediction

Bench.	$2\frac{def}{cm^2}$			$10\frac{def}{cm^2}$			Area x 10 $2\frac{def}{cm^2}$			Area x 10 $10\frac{def}{cm^2}$		
	S	M	Δ	S	M	Δ	S	M	Δ	S	M	Δ
f2	.999	.998	-.001	.997	.991	-.005	.993	.983	-.011	.967	.916	-.055
con1	.999	.998	-.001	.996	.989	-.007	.992	.979	-.013	.961	.900	-.068
rd53	.998	.995	-.003	.991	.976	-.015	.983	.954	-.031	.917	.791	-.159
misex1	.998	.995	-.003	.990	.974	-.017	.980	.949	-.033	.907	.773	-.174
dc1	.998	.995	-.003	.991	.976	-.015	.982	.954	-.030	.914	.791	-.155
dk27	.997	.992	-.005	.985	.962	-.024	.971	.927	-.048	.864	.689	-.254
radd	.997	.992	-.005	.984	.961	-.024	.968	.923	-.048	.851	.677	-.257
co14	.997	.991	-.005	.983	.958	-.026	.966	.918	-.052	.843	.661	-.276
vg2	.995	.989	-.007	.977	.944	-.035	.954	.892	-.070	.796	.576	-.382
alu3	.995	.988	-.007	.975	.942	-.036	.951	.887	-.072	.782	.562	-.393
mish	.995	.988	-.006	.973	.944	-.031	.947	.891	-.063	.769	.574	-.339
rd73	.995	.987	-.008	.974	.935	-.041	.948	.875	-.084	.772	.527	-.464
5xp1	.994	.986	-.009	.971	.931	-.043	.943	.867	-.088	.752	.504	-.492
misg	.992	.983	-.009	.962	.919	-.046	.925	.846	-.093	.687	.452	-.519

Yield results are presented in Table 5.5. Two cases for different defect densities were addressed, namely 2 def/cm^2 and 10 def/cm^2, additionally two special cases are presented in which it was assumed that the defect-sensitivity of each layout was the same but the area was ten times bigger. This was with the purpose of verifying how much error is introduced in large area layouts.

The results show that by using single-layer critical area extractions, yield is estimated as an optimistic upper bound. For small area layouts

the error is also small, less than 10%. However, for large layouts the error can grow very fast if the defect density is significant, in this analysis we can see errors of up to 51%. This is to be expected because of the exponential form of the yield model. In practice one expects that the larger the layout the more interactions among layers, and consequently the more overlooked situations for the single-layer approach. Clearly, the larger the layout, the more poly-metal, diffusion-metal crossings, etc, not to mention the increase in the number of transistors whose gate areas are sensitive to defects smaller than the minimum resolution features.

5.3 Summary and Discussion

An interesting observation is that for layouts of similar area the estimated yields lie in the same range. This leads to a practical consideration such as to characterize a sample of layouts, with similar areas and same design style, and then to obtain the defect-sensitivity, per defect mechanism, within some confidence intervals. The result of this characterization is that it is possible to have a *statistically equivalent layout* which has *statistically equivalent defect-sensitivities*. In this form, by knowing the area and design style of a layout, no matter what function the circuit performs, it is possible to predict yield by using analytical formulae without having to recur to expensive critical area extractions or complex full simulations.

Two benchmarks are of special interest, namely `dc1` and `mish`. Although they have a bigger area than their corresponding preceding benchmarks, the reported yields are bigger than the ones of `misex1` and `alu3` for defect densities greater than $10 def/cm^2$. After examining the layouts of `misex1` and `alu3` it was discovered that they are "denser" than `dc1` and `mish`. In other words, `misex1` and `alu3` were generated with less empty spaces. Also their sensitivities were larger. Hence, critical areas not only can highlight the defect sensitivity of the design but can also be used as a figure of merit to evaluate the design's "density". In yield prediction one can see that the difference of `dc1` and `mish` with their corresponding preceding benchmarks is less than 2%. Thus, "the bigger the area" does not necessary imply the "smaller the yield".

The sensitivity analysis reveals that it is quite obvious that the single layer approach cannot manage the effects of defects in the implant, and, the thin and thick oxide layers. In most of the cases the relative error was very big, if not 100%. Also, it can be observed that the

smaller the defect size the higher the error in sensitivity evaluation. This is because the single-layer approach treats all the patterns as interconnectors and cannot detect situations such as vias, transistor gate areas, crossings of layers, etc. in which defects smaller than the minimum resolution features are catastrophic. Except for sensitivities in the metal layer, all the sensitivities extracted from the single-layer approach are smaller than the multi-layer approach.

The results presented here apply for layout design styles using a Standard Cells Place and Route approach and for the multi-layer conditions presented in section 5.2.1. It might be that for other design styles the results are completely different. Think for instance of a PLA where long conductors of diffusion are laid out. This conductors are places where spots of polysilicon can form parasitic transistors. Hence, the changes in the sensitivity due to extra material of polysilicon can be drastic; not only they are due to bridges among poly patterns but also in large part to parasitic transistors formed with the diffusion wires. Nevertheless, it is possible to stipulate that by using the single-layer approach the predicted yield is an upper optimistic bound, and that the incurred error depends on the layout style and on the multi-layer conditions imposed.

Chapter 6

Discussion

This chapter is devoted to present a summary of the research presented in this thesis as well as to identify potential areas of subsequent research. It is in particular of interest to point out those areas where little or no work has been carried out but that has a direct connection with the concepts of defect and fault modeling at the lowest level of abstraction, the layout.

6.1 Further Research

The research shown in this thesis was aimed at providing an understanding of defects and their impact on yield losses. By understanding the effect of defects in IC designs it is possible to devise yield tolerant methodologies, i.e. module allocation with balanced defect-sensitivities, "defect-tolerant" driven techniques for placement and routing, etc. In this concern little research has been carried on, only one paper in the literature has been presented [61].

It is obvious that as the complexity of ICs increases towards designs of more than 1 million transistors, even small percentages of defects result in an enormous amount of faults; a 1% faults in such a design results in 10000 transistors that can be operating incorrectly. Thus, it is imperative to approach these problems with systematic methodologies for defect-tolerant designs. However these methodologies should be based on realistic situations that consider the presence and the geometrical effects of defects. As it was shown in this thesis and by other researchers [10, 88], the presence of defects is very significant to the design. These papers demonstrate how the design's robustness is affected by variations of the defect size and density distributions. The results imply that even in the same production line

these defect fluctuations can result in drastic changes in yield and in testability.

The problem cannot easily be solved without the help of CAD. In the last 5 years we have seen an increase in attention towards the automation of systems for considering defect mechanisms in ICs [13,50,63,71]. The work presented in this thesis contributes two novel approaches that correctly find critical areas from complex layouts. Previous approaches emphasized critical area models for simple layouts and for single layers. These restrictions were overcome by performing a theoretical analysis of the properties of critical areas, and by deriving strict defect and fault semantics that allow a good and flexible modeling for sequential defect-sensitivities. The results of this research were the implementation of two systems aimed at critical area extractions, one for SDSL models and the other for SDML ones. Both systems perform the extraction of critical areas deterministically, a fact which is reflected in short computation times especially for large layouts.

These systems, and all the previous research in this area, have concentrated on sequential defect sensitivity models. It is to be expected that in industry, situations that imply concurrent models also exist. However little effort has been applied towards this end. Moreover, all statistical models and defect size distributions that have appeared in the literature are for sequential models. Thus, this is a field for open research which should be explored. In the same way, defects involving parametric faults have also scarcely been explored [83].

In the last two years there has been a large amount of research conducted towards delay faults. As usual, all fault models are at a convenient high level, let us say gate level, which neglects the origins of these kind of faults. A way of approaching these problems could be by finding the defect-sensitivity of transistors for defect sizes that can disturb their operation but not to the point that the transistor is fully inoperative, i.e. those defects that change the width-length ratio of the transistor up to the point that the capacitor loading capability is trimmed.

ICs are 3-D objects and defects manifest themselves also as 3-D disturbances. No research towards this end has been considered. Assume the defect shown in Fig. 6.1(a). In a 2-D geometry such a defect is considered catastrophic, yet in a 3-D geometry the body of the defect may look like Fig. 6.1 (b). Obviously this defect is not catastrophic. It

would thus be necessary to generalize the concept of critical areas to "critical volumes" in order to capture these situations. Also there is a complete lack of data reported in the literature concerning three dimensional defects.

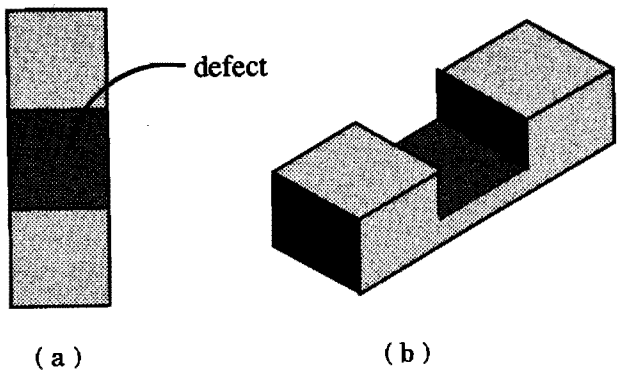


Figure 6.1. 3-D defects. (a) 2-D view. (b) 3-D view

One cannot talk about defects and put testing aside. Testing is a field which should profit from a knowledge of defect behavior. Traditionally, testing is practiced at a convenient high level of abstraction without considering the real causes of the fault. It was only until a couple of years ago that this way of thinking has been changing [44, 67]. By considering a realistic list of faults better quality test vectors can be obtained. Also, rather than simulating all theoretically possible combinations of faulty nodes a considerable reduction is achieved by simulating those faults, extracted from the layout, that may in fact occur in reality. Obviously, our system developed for SDML models is a vehicle for detecting realistic faults. However the system has now a shortcoming in its implementation. This shortcoming is that the critical areas are found per node and type of fault. Yet for breaks it is necessary to know the critical areas for every electrical branch. Fanout stems have to be analyzed ! But again, this is rather a problem of implementation and not a conceptual one.

One interesting observation is that the relationship between primitive faults at the layout level, i.e. bridges, cuts, extra and missing devices, and faults at higher levels such as switch or gate or even behavioral levels is not well established. Isolated work has been performed to make these connections. To this end, a unified framework that links defects and their impacts at different levels of abstractions was recently proposed [59], yet the fault formulation at each level is left

open for further investigations.

A simple method to handle fault-modeling complexity could be to support several levels of abstraction in the description of a fault. For example, a system designer will be interested in fault models describing the faults in the architectural modules of the design rather than in models describing faults in the IC layers.

For each level of abstraction the fault models can be described in certain primitives appropriate to that level. Each level describes the fault models to some extent avoiding irrelevant information to the specific level. Consider for instance a layout level that describes faults using only the geometry of connections and devices while omitting process related information such as the concentration of dopants, thickness of patterns, etc. or, a circuit level that describes the faults as a function of transistors, resistors, etc., on which the geometrical information is lost. Some possible levels are shown in Table 6.1.

Table 6.1. Levels of abstraction in a hierarchical fault modeling

Level	Abstraction	Fault Types	Primitives
1	System	Behavioral	modules
2	Logic	Functional	gates
3	Circuit	Electrical	devices
4	Symbolic	Geometrical	soft-structs
5	Physical	Process	hard-structs

At the highest level of abstraction, the *system* faults describe the functional faults of module units such as PLA's, ALU's, registers, etc, and provide a *behavioral fault description* of the unit. The *logic* abstraction describes internal faults of the modules in terms of logical expressions. *Functional faults* in terms of gates are provided in this level. At the next lower level, the *circuit* abstraction describes the *electrical* faults of the design. This description includes lists of faulty nodes and elements such as transistors, resistors, etc. all of which are relations between the set of nodes. The *symbolic* abstraction contains the *geometrical* faults of the design, such as the unintended geometrical deviations of the soft-structures caused by spot defects in the hard-structures. The last level, the *physical* level, describes the *process related* faults containing information such as defective patterns and process incongruities.

Worthwhile noticing is that not every fault at a given level implies also a fault at an immediate higher level. As an example consider the case

when two patterns are unintendly joined in a layout. At the symbolic level of abstraction this is a "geometrical fault" that belongs to the "bridge fault class". At the circuit level, the bridge among the patterns can be a "circuit fault" only when both patterns carry different potentials such that a fault of the "short-circuit class" arises. Thus, what for the symbolic level appears to be a fault for the circuit level may not be.

Finally, without loss of generality the fault classes at every level of the hierarchy fall in two categories: 1) wrong interconnection ordering of primitives, and 2) incorrect behavior of primitives. For instance, at the circuit level stuck-at, stuck-on transistors, etc, belong to the second category, while breaks, short-circuits etc. belong to the first one.

Another area of interest is manufacturing debugging [38]. New methods could be developed to compute defect statistics from measured fault statistics. By correctly combining fault measurements at the various products, it may be possible to establish a one to one and thus invertible mapping between the statistical fault parameters and the statistical defect parameters. Assuming that the method works it is possible to assemble defect statistics from the measurements of fault statistics at products rather than at yield monitors. Although this requires some overhead at the test sites (because electrical measurements must be continued for circuits already found "no go"), the efficiency and the reliability of computing defect statistics would be greatly enhanced. Moreover, the products establish many more samples than the yield monitors. Thus, statistical convergence is much faster.

6.2 Conclusions

The research presented in this dissertation was aimed at providing a comprehensive theory for critical area modeling, as well as to providing a solid semantic for defect and fault modeling. Relevant contributions of this research can be summarized as follows:

- A general semantic model for microelectronic technologies that encloses process induced defects with their related defect mechanisms.
- A new taxonomic model for defect-sensitivity analysis.
- A thorough mathematical study of the properties of critical areas.
- An efficient automation for the computation of single-layer critical areas.

- A prototype system for the computation of multi-layer critical areas.
- Quantification in differences of results obtained from real layouts for SDSL and SDML models.
- By using a deterministic approach for the computation of critical areas, defect sensitivity results are not influenced by defect statistics. Furthermore, the general strategy for yield analysis can be done in phases; rather than doing a complete yield simulation in which for every change in, say, defect densities or defect size distributions, the full simulation has to be run again and again, by extracting first the defect-sensitivity of the design and later combine these results with some analytical formulae a more efficient method which avoids the penalty paid in cpu time execution is obtained.

Appendix 1

Sources of Defect Mechanisms

- **Human and material contaminations**
  1. Dried spittle
  2. Sneeze residue
  3. Perspiration residue
  4. Face powder residue
  5. Dandruff residue
  6. Pencil powder
  7. Wood
  8. Paper

- **Mask disturbances**
  1. Pinspot
  2. Protrusion
  3. Bridge
  4. Opaque spot
  5. Pinhole
  6. Break
  7. Missing geometry
  8. Glass damage
  9. Solvent spot

- **Etching disturbances**
  1. Uniform overetch over substrate
  2. Uniform underetch over substrate
  3. Localized underetch
  4. Localized overetch

- **Resist process disturbances**
  1. Poor resist adhesion
  2. Incomplete development (scumming)
  3. Distortion of patterns
  4. Attack on resist pattern by developer
  5. Resist thickness

- **Mask exposure disturbances**
  1. Incorrect dimensions
  2. Distinctly printed edges
  3. Fuzzy edges
  4. Notch corners
  5. Misalignment
  6. Linewidth variations

- **Layer deposition disturbances**
  1. Uneven film thickness

2. Incorrect film thickness
3. Step coverage
4. Change in properties
   + grain structure
   + adhesion

- **Diffusion process disturbances**

1. Incorrect dopant concentration
2. Lateral diffusion
3. Junction depth
4. Crystal defects
   + Interstitial
   + Vacancy
   + Substitutional
   + Frenkel defect
   + Line dislocation
   + Skew dislocation

- **Oxidation process disturbances**

1. Incorrect thickness
2. Uneven oxide thickness
3. Oxidation induced stacking faults (OSF's)
4. Selective oxide growth
   + Bird's beak

Appendix 2

End Effects of Critical Regions

For the discussion in this appendix assume two parallel patterns $a \subset L_i$, characterized by $R(\omega) \mid L_i$ and $b \subset L_j$, characterized by $R(\omega) \mid L_j$ which are affected by a protrusion defect d^p of size δ and layer of origin L_i. Assume that both *a* and *b* share the same left and right coordinates. For terminology simplifications the failure criterion of patterns *a* and *b* will be denoted as Φ_{prot} and Φ_{prot}', respectively. Because of the symmetry at both extremes of the patterns, the explanations to follow are restricted only to the right end.

When creating critical regions, besides the pattern extension proportional to $\frac{\delta}{2}$, an additional extension, proportional to the failure criterion of the patterns has to be considered. The magnitude of this extension is evaluated as follows:

$$f(\delta,\Phi_{prot},\Phi_{prot}',s) = \begin{cases} 0 & \Phi_{prot} \le 0 \vee \Phi_{prot}' \le 0 \\ \min(\Phi_{prot},\Phi_{prot}') & \Phi_{prot} > 0 \wedge \Phi_{prot}' > 0 \wedge \\ & \max(\Phi_{prot},\Phi_{prot}') > s - \delta \\ h(\delta,\Phi_{prot},\Phi_{prot}',s) & \Phi_{prot} > 0 \wedge \Phi_{prot}' > 0 \wedge \\ & \max(\Phi_{prot},\Phi_{prot}') \le s - \delta \end{cases} \quad \text{(A1)}$$

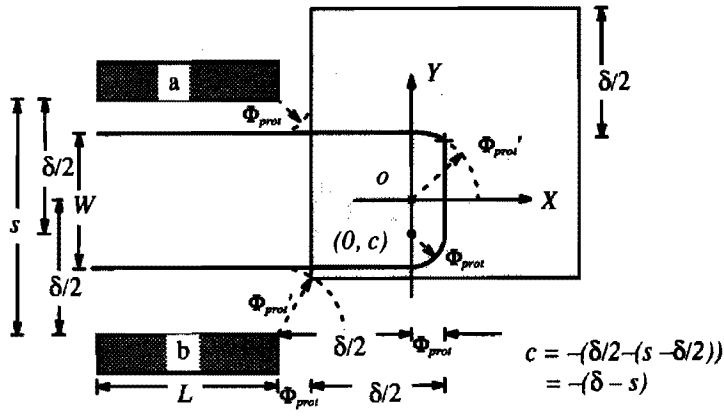
where δ is the defect size, s is the magnitude of the external susceptible site between both patterns, and h is a function taking values between 0 and $\min(\Phi_{prot},\Phi_{prot}')$. This function will later be described in the context of this appendix. For the rest of the discussion, the abbreviations f and h for $f(\delta,\Phi_{prot},\Phi_{prot}',s)$ and $h(\delta,\Phi_{prot},\Phi_{prot}',s)$, respectively, will be used. Eq. (A1) has the following physical interpretation

i) $\Phi_{prot} \le 0 \vee \Phi_{prot}' \le 0$

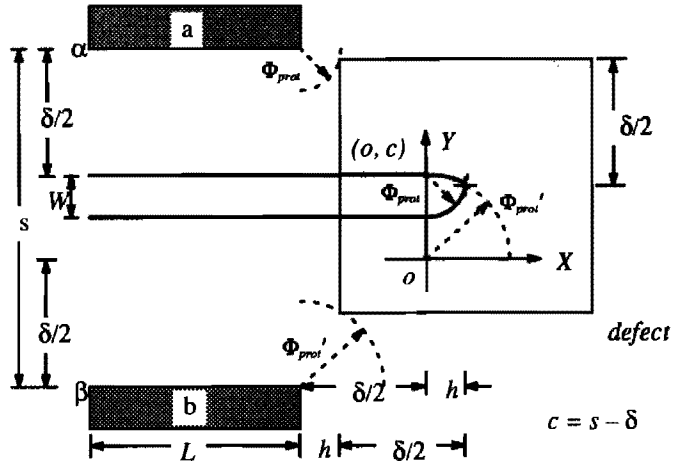
When either Φ_{prot} or Φ_{prot}' is zero means that the condition to make a bridge begins when the defect intersects the edge of the pattern. Since a defect positioned ahead of $\frac{\delta}{2}$ never satisfies this condition the extension is zero. When one of Φ_{prot} or Φ_{prot}' is negative, it means that the defect has to overlap the pattern by certain amount of area. Since a defect positioned at a further extension of $\frac{\delta}{2}$ never overlaps the pattern, the extension is also zero.

ii) $\Phi_{prot} > 0 \wedge \Phi_{prot}' > 0$

When Φ_{prot} and Φ_{prot}' are positive means that the condition to make a bridge begins when the edges of the defect are at some distance Φ_{prot} from the edge of pattern *a*, and simultaneously at a distance Φ_{prot}'



(a)



(b)

Figure A.1. End effects of critical regions for protrusion defects. (a) defect size bigger than the space between the patterns, (b) defect size smaller than the space between the patterns.

from pattern *b*. Consider the case where $\Phi_{prot}' > \Phi_{prot}$. The extension f is not zero and in this case takes the maximum value of Φ_{prot} because if the center of the defect is positioned at $\frac{\delta}{2} + \Phi_{prot}'$, its left edge will be at a distance bigger than Φ_{prot} and will never satisfy the condition for a bridge with *a*. Furthermore, the shape of its corresponding subset of the total critical region is not rectangular. It

is bounded by two arcs, or two arcs and one line segment as shown in Figs. A.1a and A.1b. Choosing the center of the bounding arc due to Φ_{prot}', of pattern *b*, as origin (see Fig. A.1a), this section of the critical region is best described as a set of connected points C_{ext}, where

$$C_{ext} = \left\{(x,y) \mid \begin{array}{l} 0 < x \le f \wedge (s - \delta - \Phi_{prot}) < y \le \Phi_{prot}' \wedge \\ (x^2 + y^2 \le (\Phi_{prot}')^2 \vee x^2 + (y - (s - \delta))^2 \le \Phi_{prot}^2) \end{array}\right\} \quad \text{(A2)}$$

This follows from the rationale that the distance between the lower left (upper left) corner of the defect and the right corner of *b* (*a*) has to be at most the failure criterion Φ_{prot}' (Φ_{prot}). Since this distance has to be constant, in the extreme case, we can draw a circle of radius Φ_{prot}' (Φ_{prot}) centered at the right corner point of *b* (*a*), such that any point in the perimeter of this circle corresponds exactly to the coordinates of the lower left (upper left) corner of the defect. Because the corner of the defect moves along the circle, the center of the defect also moves following the same trajectory. With such established critical region, it can be observed that for a defect of size $\delta > s - \Phi_{prot}'$, the two bounding arcs will intersect at the line $x = \Phi_{prot}$, e.g. in order to make a bridge, the extension f takes the value of Φ_{prot}. However, for a defect of size $\delta \le s - \Phi_{prot}'$, such as the case of Fig. A.1b, the two bounding arcs will intersect at line $x = h$ with $h < \Phi_{prot}$. As a result, the extension f can not be greater than h in order to make a bridge. The exact value of h can be found by solving the boundary equations in (A2) for x as

$$\begin{cases} x^2 + y^2 = (\Phi_{prot}')^2 \\ x^2 + (y - (s - \delta))^2 = \Phi_{prot}^2 \end{cases} \quad \text{(A3)}$$

For $\Phi_{prot}' < \Phi_{prot}$, the symmetric conclusions can be drawn, which results in eq. (A1).

For the ease of modeling and simplification of the algorithms for computing critical areas, the shape of this section can always be approximated as a rectangle with length equal to $\text{Ramp}(\min(\Phi_{prot}, \Phi_{prot}'))$ and width equal to $\delta - s + \Phi_{prot} + \Phi_{prot}'$, where Ramp is the standard "ramp function" defined as

$$\text{Ramp}(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases} \quad \text{(A4)}$$

Let us analize now the error incurred in the computation of the critical area for this bridge model.

Solving eq.(A3) for y, the exact critical area is computed as

$$Area = (L + \delta)w + 2(A_{real}) \tag{A5a}$$

where

$$A_{real} = \int_0^f \left| \sqrt{(\Phi_{prot}')^2 - x^2} - \left(- \sqrt{\Phi_{prot}^2 - x^2} + s - \delta \right) \right| dx \tag{A5b}$$

and w is derived as follows:

$$w = pos(\beta) + \frac{\delta}{2} + \Phi_{prot}' - \left(pos(\alpha) - \frac{\delta}{2} - \Phi_{prot}\right) \tag{A5c}$$

given that the magnitude of the external site between both patterns is obtained as

$$s = Mag(E) = pos(\alpha) - pos(\beta)$$

eq. (A5c) results in

$$w = \delta - s + \Phi_{prot} + \Phi_{prot}' \tag{A5d}$$

For the computational model of critical areas the approximated area is estimated as follows

$$Area' = (L + \delta)w + 2(A_{estimated}) \tag{A6a}$$

where

$$A_{estimated} = \int_0^{\min(\Phi_{prot}, \Phi_{prot}')} w dx = \int_0^{\min(\Phi_{prot}, \Phi_{prot}')} (\delta - s + \Phi_{prot} + \Phi_{prot}') dx \tag{A6b}$$

The magnitude of the error incurred in this approximation is calculated by

$$error = \left| \frac{Area - Area'}{Area} \right| = \left| \frac{2(A_{estimated} - A_{real})}{(L + \delta)w + 2A_{real}} \right| \tag{A7}$$

For $\Phi_{prot} + \Phi_{prot}' = c$, where c is a constant, the worst case error appears when $\delta \leq s - \max(\Phi_{prot}, \Phi_{prot}')$ and $\Phi_{prot} = \Phi_{prot}'$ because for certain δ and s satisfying the previous conditions, $A_{estimated}$ and A_{real} take a maximum and minimum value, respectively. In order to simplify the manipulation of eq. (A7) let the upper boundary of the integrals take the value of Φ_{prot}. In the case of eq. (A5b) A_{real} results in an even smaller area. Thus, the error will be even more pessimistic. By substituting $\Phi_{prot} = \Phi_{prot}'$ in eqs. (A5a) and (A6b) and after algebraic simplifications

eq. (A7) results in

$$\text{error} = \left| \frac{4 \int_0^{\Phi_{prot}} (\sqrt{\Phi_{prot}^2 - x^2} - \Phi_{prot})dx}{(L + \delta)w + 2 \int_0^{\Phi_{prot}} (2\sqrt{\Phi_{prot}^2 - x^2} - \Phi_{prot})dx} \right| \qquad \text{(A8)}$$

$$= \left| \frac{(4 - \pi)\Phi_{prot}^2}{(L + \delta)w + (\pi - 2)\Phi_{prot}^2} \right|$$

By substituting the assumptions $\delta \leq s - \max(\Phi_{prot}, \Phi_{prot}')$ and $\Phi_{prot} = \Phi_{prot}'$ in eq. (A5d) one can see that w takes values between 0 and Φ_{prot}. In order to estimate the error let us observe the effect of w in eq. (A8). For very small w the error is relatively big, however the contribution of the computed area to the total crtitical area is insignificant. For large w the error is simply very small. Also with the assumption that $L \gg \Phi_{prot}$ (mostly in real layouts it is acceptable), it is easy to conclude that the error introduced is small. Since the error is almost neglectable it is stipulated that this approximation is a good compromise.

Appendix 3

NMOS Technology File

```
(technology NMOS
 ( masks
   ( <nm> $nc $nb ni <np> <nd> ) )

 {* nm = metal nc = contact ni = implant
  nd = diffusion np = poly nb = buried contact *}

 ( suppress
   (nc ~nm ~np ~nd ~nb)
   (nc ~nm np nd )
   (nm nc np nd ~nb)
   (nb ~np ~nd ~nm)
   (ni ~np ~nd  ~nm ~nc ~nb) )

 ( structures
   (poly_metal    <np> <nm> ~nd ~nc)
   (dif_metal     ~np <nd> <nm> ~nc)
   (poly_via      nc <nm np> ~nd)
   (dif_via       nc <nm nd> ~np)
   (buried_via    <np nd> nb ~nc)
   (pdm_via       <nm nd np> nb nc)
   (poly_track    <np> ~nd ~nm)
   (dif_track     ~np <nd> ~nm)
   (metal_track   ~np ~nd <nm>)
   (enh_xtor
    (trans_1     ~ni <np #nd> ~nm ~nb ~nc)
    (trans_2     ~ni <np #nd> <nm> ~nb ~nc))
   (dep_xtor
    (trans_1     ni <np #nd> ~nm ~nb ~nc)
    (trans_2     ni <np #nd> <nm> ~nb ~nc))
 )

 ( defects (7 19 1)
   (poly_metal
    (np (*nc 1.0 short) )
    (nm (*nc 1.0 short) ))
   (dif_metal
    (nd (*nc 1.0 short) )
    (nm (*nc 1.0 short) ))
```

```
  (poly_via
   (np (-np 0.0 break))
   (nc (-nc 0.5 break))
   (nm (-nm 0.0 break)) )
  (dif_via
   (nd (-nd 0.0 break) (+np -1.0 float_line)(*np 0.0 float_line))
   (nc (-nc 0.5 break))
   (nm (-nm 0.0 break)) )
  (buried_via
   (nd (-nd 0.0 break))
   (nb (-nb 0.5 break))
   (np (-np 0.0 break)) )
  (poly_track
   (np (-np 0.0 break) (+np 0.0 short) (+nd -1.0 new_device)))
  (dif_track
   (nd (-nd 0.0 break) (+np -1.0 new_device)(*np 0.0 new_device)
   (+nd 0.0 short)))
  (metal_track
   (nm (+nm 0.0 short) (-nm 0.0 break)))
  ((enh_xtor
   (trans_1
   (nd (*nb 0.5 miss_device)(*ni 0.5 new_device))
   (np (*nb 0.5 miss_device)(-np 0.5 miss_device)) )))
  ((dep_xtor
   (trans_1
   (ni (-ni 0.5 miss_device))
   (nd (*nb 0.5 miss_device))
   (np (*nb 0.5 miss_device) (-np 0.5 miss_device) ) )))
 )
)
```

References

[1] Dimitri A. Antoniadis and Robeert W. Dutton, "Models for Computer Simulation of Complete IC Fabrication Process," *IEEE Trans on Electron Devices*, vol. ED-26, no. 4, pp. 490-500, April 1979.

[2] Jon Louis Bentley, Dorothea Haken, and Robert W. Hon, "Fast Geometric Algorithms for VLSI Tasks," *1980 IEEE Compcon Spring*, pp. 88-92, 1980.

[3] Jon Louis Bentley and Derick Wood, "An Optimal Worst Case Algorithm for Reporting Intersections of Rectangles," *IEEE Transcations on Computers*, vol. C-29, no. 7, pp. 571-577, July 1980.

[4] J. L. Bentley and T. Ottmann, "Algorithms for Reporting and Counting Geometric Intersections," *IEEE Trans. on Computing*, vol. C-28, September 1979.

[5] Susan Powell Billat, "Automated Defect Detection on Patterned Wafers," *Semiconductor International*, pp. 116-119, May 1987.

[6] P. Burggraaf, "CAM Software-Part 1: Choices and Capabilities," *Semiconductor International*, vol. 10, no. 6, pp. 56-61, June 1987.

[7] F. Camerik, P.A.J. Dirks, and J.A.G. Jess, "Quantification and Qualification of Process-Induced Product-Related Defects," *Proc. Int. Test Conference*, pp. 643-652, Aug. 1989.

[8] I. Chen and A. J. Strojwas, "Realistic Yield Simulation for VLSIC Structural Failures," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 6, pp. 965-980, Nov. 1987.

[9] James A. Cunningham, "The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing," *IEEE Trans. on*

Semiconductor Manufacturing, vol. 3, no. 2, pp. 60-71, May 1990.

[10] Chennian Di and Jose Pineda de Gyvez, "A Spot Defect to Fault Collapsing Technique," *33rd Midwest Symp. on Circ. and Syst.*, Aug. 1990.

[11] D. G. Edwards, "Testing for MOS IC Failure Modes," *IEEE Trans. Reliability*, vol. R-31, no. 1, pp. 9-18, April 1982.

[12] F. Fantini and C. Morandi, "Failure modes and mechanisms for VLSI ICs - a review," *IEE Proceedings*, vol. 132, Pt. G., no. 3, pp. 74-81, June 1985.

[13] F.J. Ferguson and J.P. Shen, "A CMOS Fault Extractor for Inductive Fault Analysis," *IEEE Trans. on Computer-Aided Design*, vol. CAD-7, no. 11, pp. 1181-1194, Nov. 1988.

[14] A. V. Ferris-Prabhu, "Modeling the Critical Area in Yield Forecast," *IEEE J. Solid-State Circuits*, vol. SC-20, no. 4, pp. 874-877, Aug. 1985.

[15] Albert V. Ferris-Prabhu, "Defect Size Variations and their Effect on the Critical Area of VLSI Devices," *IEEE J. Solid-State Circuits*, vol. SC-20, no. 4, pp. 878-880, Aug. 1985.

[16] Albert V. Ferris-Prabhu, "Role of Defect Size Distributions in Yield Modeling," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 1727-1736, Sept. 1985.

[17] A. V. Ferris-Prabhu and M. A. Retersdorf, "The Effect on Yield of Clustering and Radial Variations in Defect Density," *Proc. Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 40-50, Oct. 1989.

[18] A. V. Ferris-Prabhu, L. D. Smith, H. A. Bonges, and J.K. Paulsen, "Radial Yield Variations in Semiconductor Wafers," *IEEE Circuits & Devices Magazine.*, vol. 3, no. 2, pp. 42-47, March 1987.

[19] M. J. Flynn, "Very High Speed Computing Systems," *Proc. IEEE*, vol. 54, pp. 1901-1909, Dec. 1966.

[20] J. Galiay, Y. Crouzet, and M. Vergniault, "Physical Versus Logical Fault Models MOS LSI Circuits: Impact on Their Testability," *IEEE Trans. on Computers*, vol. C-29, no. 6, pp. 527-531, Jun. 1980.

[21] Sophie Gandemer, Bernard C. Tremintin, and Jean-Jacques Charlot, "Critical Area and Critical Level Calculation in I.C. Yield

Modeling," *IEEE Trans. Electron Devices*, vol. 35, no. 2, pp. 158-166, Feb. 1988.

[22] Sophie Gandemer, Bernard C. Tremintin, and Jean-Jacques Charlot, "A Method for Determining Critical Areas and Critical Levels for IC Yield Estimation," in *Yield Modelling and Fault Tolerance in VLSI*, ed. Will Moore, Wojciech Maly and Andrzej Strojwas, pp. 101-110, Adam Hilger, 1988.

[23] L.P.P.P van Ginneken, J.T.J. van Eijndhoven, and J. Brouwers, "Doubly Folded Transistor Matrix Layout," *IEEE Int. Conference on Computer Aided Design*, pp. 134-137, Nov. 1988.

[24] Anil Gupta, W. A. Porther, and Jay W. Lathrop, "Defect Analysis and Yield Degradation of Integrated Circuits," *IEEE J. Solid-State Circuits*, vol. SC-9, no. 3, pp. 96-103, Jun. 1974.

[25] W.E. Ham, "Yield-Area Analysis:Part I - A Diagnostic Tool for Fundamental Integrated Circuit Process Problems.," *RCA Rev.*, vol. 39, pp. 231-249, June 1978.

[26] S. R. Hofstein and F. P. Heiman, "The Silicon Insulated-Gate Field Effect Transistor," *Proc. IEEE*, vol. 51, no. 9, pp. 1190-1202, Sept. 1963.

[27] Donald E. Knuth, *The Art of Computer Programming*, Volume 3 Sorting and Searching, Addison-Wesley Publishing Company, Inc., Philippines, 1973.

[28] Charles Kooperberg, "Circuit Layout and Yield," *IEEE J. Solid-State Circuits*, vol. vol.23, no. no.4, pp. 887-892, Aug. 1988.

[29] George M. Koppelman and Michael A. Wesley, "OYSTER: A Study of Integrated Circuits as Three-Dimensional Structures," *IBM J. Res. Develop*, vol. 27, no. 2, pp. 149-163, March 1983.

[30] Israel Koren, "The Effects of Scaling on the Yield of VLSI Circuits," in *Yield Modelling and Fault Tolerance in VLSI*, ed. Will Moore, Wojciech Maly and Andrzej Strojwas, pp. 91-99, Adam Hilger, 1988.

[31] John A. Lange, "Sources of Semiconductor Wafer Contamination," *Semiconductor International*, pp. 124-128, April 1983.

[32] Michael Lorenzetti, Paul Magill, Alexander Dalal, and Paul Franzon, "McYield: A CAD Tool for Functional Yield Projections for VLSI," *Proc. Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 100-110, Nov. 1990.

[33] R.K. Lowry, J.H. Linn, G.M. Grove, and C.A. Vicroy, "Analysis of Human Contaminants Pinpoint Sources of IC Defects," *Semiconductor International*, pp. 73-77, July 1987.

[34] C. L. Mallory, D. S. Perloff, T. F. Hasan, and R. M. Stanly, "Spatial Yield Analysis in Integrated Circuit Manufacturing," *Solid State Technology*, vol. 26, no. 11, pp. 121-127, November 1983.

[35] Wojciech Maly, "Modeling of Point Defect Related Yield losses for CAD of VLSI Circuits," *Proc. Int. Conf. Computer-Aided Design*, pp. 161-163, Nov. 1984.

[36] Wojciech Maly, "Modeling of Lithography related yield losses for CAD of VLSI circuits," *IEEE Trans. Computer-Aided Design*, pp. 166-177, 1985.

[37] Wojciech Maly, "Realistic Fault Modeling for VLSI Testing," *Proc. 24th Design Automation Conf.*, pp. 173-180, 1987.

[38] Wojciech Maly, in *Atlas of IC Technologies*, ed. The Benjamin/Cummings Publishing Company Inc., CA., USA, 1987.

[39] Wojciech Maly, "Yield Simulation - A Comparative Study -," *Proc. Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, Oct. 1989.

[40] Wojciech Maly, "Computer-Aided Design for VLSI Circuit Manufacturability," *Proc. of the IEEE*, vol. 78, no. 2, pp. 356-392, Feb. 1990.

[41] Wojciech Maly and Samir B. Naik, "Process Monitoring Oriented IC Testing," *Proc. Int. Test Conference*, pp. 527-532, Aug. 1989.

[42] Wojciech Maly, Andrzej J. Strojwas, and Stephen W. Director, "VLSI Yield Prediction and Estimation: A Unified Framework," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, no. 1, pp. 114-130, Jan. 1986.

[43] Wojciech Maly, "Design Methodology for Defects Tolerant Integrated Circuits," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 27.5.1-27.5.4, 1988.

[44] W. Maly and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation," *Electronic Letters*, vol. 19, no. 6, pp. 226-227, March 1983.

[45] W. Maly, F. J. Ferguson, and J. P. Shen, "Systematic Characterization of Physical Defects for Fault Analysis of MOS IC Cells," *Proc. 15th Int. Test Conf.*, pp. 390-399, 1984.

[46] W. Maly, M. Thomas, J. Chinn, and D. Campbell, "Characterization of Type, Size and Density of Spot Defects in the Metalization Layer," in *Yield Modelling and Fault Tolerance in VLSI,*, ed. Will Moore, Wojciech Maly and Andrzej Strojwas, pp. 71-91, Adam Hilger, 1988.

[47] W. Maly, B. Trifilo, R. A. Hughes, and A. Miller, "Yield Diagnosis through Interpretation of Tester Data," *Proc. of Int. Test Conf.*, pp. 10-20, 1987.

[48] B. T. Murphy, "Cost-size Optima of Monolithic Integrated Circuits," *Proc. IEEE*, vol. 52, no. 12, pp. 1537-1545, Dec. 1964.

[49] H. Murrmann and D. Kranzer, "Yield Modeling of Integrated Circuits," *Siemens Forsch. u. Entwickl.-Ber.*, vol. 9, no. 1, pp. 38-40, Feb. 1980.

[50] Phil Nigh and Wojciech Maly, "Layout-Driven Test Generation," *Proc. Int. Conf. Computer-Aided Design*, pp. 154-158, Nov. 1989.

[51] T. Okabe and S. Shimada, "Analysis on Yield of Integrated Circuits and a New Expression for the Yield," *Elec. Eng. Japan*, vol. 92, no. 6, pp. 135-141, Dec. 1972.

[52] Jose Pineda de Gyvez, "LASER: A Aayout Sensitivity ExploreR. Report and User's Manual," EUT Report 89-E-216 ISBN 90-6144-216-8, March 1989.

[53] Jose Pineda de Gyvez, "Single vs. Multi Layer Critical Area Extractions. - A Comparative Study for Yield Prediction -," *Int. Workshop on Defects and Fault Tolerance in VLSI Systems*, pp. 47-61, Nov. 1990.

[54] Jose Pineda de Gyvez and Chennian Di, "IC Defect-Sensitivity for Footprint Type Spot Defects," *IEEE Trans. on Computer-Aided Design*, accepted for publication, Jan. 1991.

[55] Jose Pineda de Gyvez and J. A. G. Jess, "On the Design and Implementation of a Wafer Yield Editor," *IEEE Trans. on Computer-Aided Design*, vol. 8, no. 8, pp. 920-925, August 1989.

[56] Jose Pineda de Gyvez and J. A. G. Jess, "Systematic Extraction of Critical Areas from IC Layouts," *Proc. Int. Workshop on Defects and Fault Tolerance in VLSI Systems*, pp. 27-39, Tampa, Oct.

1989.

[57] Jose Pineda de Gyvez and J. A. G. Jess, "A Layout Defect-Sensitivity Extractor," *Proc. Int. Conf. Computer-Aided Design*, pp. 538-541, Nov. 1989.

[58] Jose Pineda de Gyvez and J.A.G. Jess, "On the Definition of Critical Areas for IC Photolithographic Spot Defects," *1st European Test Conference*, pp. 152-158, April 1989.

[59] Jose Pineda de Gyvez and J. A. G. Jess, "IC Spot Defect and Fault Semantics - A Unified Framework -," *Int. Symp. on Circuits and Systems*, pp. 2712-2715, May 1990.

[60] Jose Pineda de Gyvez and J. A. G. Jess, "Layout Verification for Fault and Yield Analysis," *Proc. 2nd Symp. on Digital Systems (PRORISC)*, pp. 89-98, Apr. 1990.

[61] A. Pitaksanonkul, S. Thanawastien, C. Lursinap, and J.A. Gandhi, "DTR: A Defect Tolerant Routing Algorithm," *26th ACM/IEEE Des. Automation Conf.*, pp. 795-798, 1989.

[62] F.P. Preparata and M. I. Shamos, in *Computational Geometry: An Introduction*, ed. Springer-Verlag, 1985.

[63] Rahul Razdan and A.J. Strojwas, "A Statistical Design Rule Developer," *IEEE Trans. on Computer-Aided Design*, vol. CAD-5, no. 4, pp. 508-520, October 1986.

[64] Michel Rivier, "Random Yield Simulation Applied to Physical Circuit Design," in *Yield Modelling and Fault Tolerance in VLSI,* , ed. Will Moore, Wojciech Maly and Andrzej Strojwas, pp. 111-120, Adam Hilger, 1988.

[65] K. Saito and E. Arai, "Experimental Analysis and New Yield Modeling of MOS LSI Yield Associated with the Number of Elements," *IEEE J. Solid-State Circuits*, vol. SC-17, no. 1, pp. 28-33, Fen. 1982.

[66] R. B. Seeds, "Yield, Economic, and Logistic Models for Complex Digital Arrays," *IEEE Int. Conv. Rec.*, pp. 61-66, Apr. 1967.

[67] John P. Shen, Wojciech Maly, and F. Joel Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits," *IEEE Design and Test of Computers*, pp. 13-26, Dec. 1985.

[68] Barry Simon, Javier Prado, and Larry Day, "Software Tools for Analysis of Wafer Sort Yield Data," *Proc. Int. Test Conference*, pp.

670-679, 1987.

[69] Bruce B. Sindahl, "Interactive Graphical Analysis of Bit-Fail Map Data using Interactive Pattern Recognition," *Proc. Int. Test Conf.*, pp. 687-692, 1987.

[70] Peter H. Singer, "Photomask Reticle Defect Detection," *Semiconductor International*, pp. 66-73, April 1985.

[71] Costas J. B. Spanos and S. W. Director, "Parameter Extraction for Statistical IC Process Characterization," *IEEE Trans. on Computer-Aided Design*, vol. CAD-5, no. 1, pp. 66-78, January 1986.

[72] C.H. Stapper, "Defect Density Distribution for LSI Yield Calculations," *IEEE Trans. Electron Devices*, vol. ED-20, pp. 655-657, Jul. 1973.

[73] C.H. Stapper, "LSI Yield Modeling and Process Monitoring," *IBM J. Res. & Dev.*, vol. 20, pp. 228-234, May 1976.

[74] C. H. Stapper, "Modeling of Integrated Circuit Defect Sensitivities," *IBM J. Res. Develop.*, vol. 27, no. 6, pp. 549-557, Nov. 1983.

[75] C. H. Stapper, "Yield model for fault clusters within integrated circuits," *IBM J. Res. & Dev.*, vol. 28, no. 5, pp. 636-639,, September 1984.

[76] C. H. Stapper, "Modeling of defects in integrated circuit photolithographic patterns," *IBM Res. Develop.*, vol. 28, no. 4, pp. 461-475, Jul. 1984.

[77] C. H. Stapper, "The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions," *IBM J. Res. & Dev.*, vol. 29, no. 1, pp. 87-97, January 1985.

[78] C.H. Stapper, "On yield, fault distributions, and clustering of particles," *IBM J. Res. & Dev.*, vol. 30, no. 3, pp. 326-338, May 1986.

[79] C.H. Stapper, "Fact and Fiction in Yield Modeling," *Microelectronics Journal*, vol. 20, no. 1-2, pp. 129-151, 1989.

[80] C.H. Stapper, "Small-area fault clusters and fault tolerance in VLSI circuits," *IBM J. Res. Develop.*, vol. 33, no. 2, pp. 174-177, March 1989.

[81] C.H. Stapper, "Large-area fault clusters and fault tolerance in VLSI circuits: A review," *IBM J. Res. Develop.*, vol. 33, no. 2, pp. 162-173, March 1989.

[82] C. H. Stapper, F. Armstrong, and K. Saji, "Integrated Circuit Yield Statistics," *Proc. IEEE*, vol. 71, pp. 453-470, Apr. 1983.

[83] Marek Syrzycki, "Modeling of Spot Defects in MOS Transistors," *Proc. International Test Conference*, pp. 148-157, 1987.

[84] R. G. Taylor and E. Stephens, "Microcircuit Failure Analysis," *British Telecomm. Engr.*, vol. 4, pp. 39-46, April 1985.

[85] J.F.M. Theeuwen and M.R.C.M. Berkelaar, "Logic optimisation with technology and delay in mind," *Notes of the International workshop on logic synthesis*, Research Triangle Park, North Carolina,, May 12-15, 1987.

[86] C. Timoc, M. Buehler, T. Griswold, C. Pina, F. Scott, and L. Hess, "Logical Models of Physical Failures," *Proc. Int. Test Conference*, pp. 546-553, 1983.

[87] Arthur C. Titus, "Photomask Defects: Causes and Solutions," *Semiconductor International*, pp. 94-100, October 1984.

[88] D.M.H. Walker, "Yield Analysis for Fault Tolerant Arrays," *Proc. Int. Conf. on Wafer Scale Integration*, pp. 257-265, San Francisco, California, Jan. 1989.

[89] D.M.H. Walker and D.S. Nydick, "DVLASIC: Catastrophic Yield Simulation in a Distributed Processing Environment," *IEEE Trans. on Computer-Aided Design*, pp. 655-664, June 1990.

[90] Hank Walker and Stephen Director, "VLASIC: A Catastrophic Yield Simulator for Integrated Circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, no. 4, pp. 541-556, Oct. 1986.

[91] J. T. Wallmark, "Design Considerations for Integrated Electron Devices," *Proc. IRE*, vol. 48, no. 3, pp. 293-300, Mar. 1960.

[92] Takayuki Yanagawa, "Yield Degradation of Integrated Circuits Due to Spot Defects," *IEEE Trans. Electron Devices*, vol. ED-19, no. 2, pp. 190-197, Feb. 1972.

Stellingen by het proefschrift van José Pineda de Gyvez

1. Sometimes simple phrases such as "life is hard but beautiful" do not really make their way up to our brains. Is then life always easy ? Or, is it a subconscious rejection to acknowledge the problems in our surroundings ?

2. A "very" small percentage of undetected faults becomes "very" important when the fault occurs in one of the ICs of the landing control unit of an airplane.

3. The recent changes in world politics concerning the East block and Middle Asia reaffirm once more the words of the Mexican President Benito Juarez: "Among individuals as well among nations the respect to each other's right is the peace".

   [Justo Sierra, "Juarez su Obra y su Tiempo", Editora Latinoamericana S.A., Guatemala 10-220 Mexico 1 D.F., Mexico, 9 Sep. 1960]

4. With a deterministic approach to determine the defect-sensitivity of a design (and the reluctance of industry to provide defect data), the most one can do is to assess the failure likelyhood of a design.

   [this thesis]

5. "Realistic" reliability measures on the "failure of interconnects" must also take into account the concept of critical areas to predict the failure rate of interconnections due to electromigration problems.

   [Kevin G. Kemp, Kelvin F. Poole and David F. Frost, "The Effects of Defects on the Early Failure of Metal Interconnects," *IEEE Trans. on Reliability*, vol. 39, no. 1, pp. 26-29, April 1990]

6. A good manager points out what the engineer did correctly - while making constructive corrections on what to improve.

7. During talk presentations it is not unusual to find people in the audience posing questions designed to show how smart and informed they themselves are. Sometimes I wonder why they deign to hear the talk ...

8. The "new" generation of CAD systems for yield loss diagnosis should be capable to invert the defect-to-fault flow such that symbolic manufacturing debugging becomes possible.

9. The integration of design for testability into logic synthesis, better known as "logic synthesis for testability", is a fine idea. However, this approach still fails to capture some faults because all faults are "layout-level" and not "gate-level" dependent.

10. One can conclude that parallel computing will still remain a myth until the new hardware avoids the von Neumann principle.

    [R.W. Hartenstein, A. Hirschbiel, M. Riedmueller, K. Schmidt and M. Weber, "A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware", *Int. Conf. on Information Technology*, Oct. 1990]