IC spot-defect and fault semantics - a unified framework

IC SPOT-DEFECT AND FAULT SEMANTICS
- A UNIFIED FRAMEWORK -

Jose Pineda de Gyvez

J.A.G. Jess

Eindhoven University of Technology
Department of Electrical Engineering
P.O. Box 513, 5600 MB, Eindhoven
The Netherlands

Abstract

We present a thorough theoretical framework to model spot defects with their related faults in any IC technology. The defect models considered are unintended geometrical variations introduced in the shape of the patterns of the IC. The transcendence of a defect is determined by the impact that it has at several levels of abstractions. We call this impact a fault. Our framework is a mathematical construction which encompasses a hierarchical fault modeling that avoids irrelevant information at every level of abstraction. The framework encloses consistency requirements on fault modeling which can be used to analyze the origins and reasons of malfunctions in production chips.

1. INTRODUCTION.

The variety of IC technologies increases the difficulties of choosing realistic fault models [1]. Traditional approaches to fault modeling assume a convenient high-level abstraction to model the IC malfunctions without considering the technology, yet these faults have their origins in changes of the chemical and matter compositions occurring in the IC.

Defects have very complex physical characteristics [2] and may be significant different from technology to technology. The adequacy in fault modeling can be expressed in terms of the specific defects occurring in the specific technology.

This paper presents a formal framework to model defects and faults in any IC technology. The framework is an elegant theoretical description of the physical properties of microelectronic processing enclosing the relationship between process induced defects and faults [3-6]. It provides the link between faults at a low level, namely spot defects degenerating the geometry of the IC patterns, and some higher levels such that electrical, logical, and behavioral fault models can efficiently be employed.

In section 2 of the paper a mathematical framework for IC microelectronic processing is developed. In section 3 formal definitions for defects and faults introduced in the microelectronic processing are given, consistency requirements for fault modeling are presented, and a hierarchical fault modeling framework is shown.

2. IC TECHNOLOGY

Microelectronic processing consists of a series of steps carried out in a specific order. The goal of these steps is to transform an electrical circuit design into an operable device, the integrated circuit (IC). Thus, a microelectronics technology $\mathcal{T}$ is an ordered set of process steps $\{t_1,t_2,\ldots,t_n\}$ which are concerned with changes in matter no more than a few microns above or below the surface of a working material.

The flat working material on which the processing steps take place is called *the substrate*. The substrate is a single-crystal slice cut from a larger crystal called *wafer*. By far the most common substrate is the silicon wafer. Each distinct matter on the surface of the wafer has a set of properties, $H_{matter} = \{h_1,h_2,\ldots,h_n\}$, such as its corresponding level of impurities, its thickness, its shape, etc., that characterize the integrity of the matter.

Geometrically, an IC can be seen as a 3-D Euclidean space with "lateral" coordinates (x,y) and vertical coordinate z. In the z-direction we introduce a partition into intervals by fixing points in the z-axis z_i, $i \in I$, where $I = \{1,2,..,N\}$ is the set of the number of distinct matters. Those z-points define "matters" as open connected point sets as follows: assume some $i \in I$ then we define

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

L_0 is actually the substrate with its background doping, whereas L_∞ actually is established by the (electrically passive) "air" on the top of the IC. In between we have a set of matters such that each different matter has unique electrical properties. To such an arrangment of matter in *layers* we refer to as *silicon layer structure*. We denote the set of layers as $L = \{L_i \mid i \in I\}$.

Among the processing steps we distinguish several subsets of steps whose technological objective is the same, i.e. to deposit a matter, to wipe out a matter, to etch holes in the matter, to model the shape of the matter, etc. Therefore, during the fabrication process several layers exist only temporarily, such as the photoresist layer in a technology with a photolithographic process step, and some layers exist permanently such as a polysilicon layer, common to all modern MOS technologies. The former layers are denoted as *transitory layers* and the latter ones are members of a set *structure layers*, $L_s \subset L$, defined as $L_s = \{l_1,\ldots,l_n\}$.

Each layer is shaped by a series of process steps such as oxidation, etching, etc. The connected point sets P_i, $i \geq 1$, henceforth *patterns*, divide the layer in two disjoint regions: *active* and *inactive*. The set *active region* $A=\{a_1,a_2,\ldots,a_n\}$ is the set of patterns that are left after the shaping process. We denote these patterns as *active patterns*. The set *inactive region*, $\bar{A}$ is the set of patterns interleaved by the *active patterns* and corresponds to the layer portions that are etched away during the shaping process. Generalizing, we say that a shaped layer $L_i \in L$ has an image such that $Image(L_i) \equiv A$.

Depending upon the electrical property of the matter, a structure layer can have one of the following properties: *insulator* or *conductor/semiconductor*. An insulator layer prevents two conductor/semiconductor layers from making contact. A conductor/semiconductor layer is usually used for interconnections. Among the patterns of a conductor layer the inactive patterns act also as insulators.

Within one layer the inactive patterns may be filled by some other layers. For instance in layer L_i we may encounter thick oxide and metal. Assume some point $(x_i,y_i,\eta) \mid z_{i-1}<\eta<z_i$. After processing, such a point may be element of an inactive pattern filled with metal. In fact we can define a set in terms of an ennumerated type $U_i = \{thick\ oxide, metal\}$. Fixing $z=\eta$ defines a 2-dimensional Euclidean space which in the case of $z_{i-1}<\eta<z_i$ will be denoted by $L_i(\eta)$. Then, for any point $(x,y) \in E^2$ we define a function $S_i(\eta):L_i(\eta)\rightarrow U_i(\eta)$ which in fact assigns a value from S_i to any pair of coordinates. We call this value the *state* of the layer at (x,y) and U_i the *stateset*.

From the above it follows that given the set of layers L there is an associated set of statesets $U=\{U_i \mid i \in I\}$. Now we establish the product set of all layers by letting $L=(L_1,L_2,\ldots,L_{|I|})$ be an $|I|$ dimensional vector. Analogously we establish the product set $U=(U_1,U_2,\ldots,U_{|I|})$.

CH2868-8/90/0000-2712$1.00 © 1990 IEEE

Logically $S=(S_1,S_2,\ldots,S_{|I|})$ becomes a vector function such that we can write $S(x,y,\eta):L(x,y,\eta)\rightarrow U$.

Obviously the state characterization of silicon layer structures can be applied to identify electrical components by multivalued logical clauses. Adopting "X" as a don't care notion, consider for instance the clause $S(x,y)$ = (*bulk, thin oxide, X, X*) true for some point in $(x,y) \in E^2$ (in a 4-layer structure |I| = 4) describing a point of an active gate area of an NMOS-transistor. The clause $S(x,y)$=(*drain–source dope, metal* $\vee$ *field oxide, X, X*) indicates that (x,y) belongs to a source or drain region. We denote these clauses as *state clauses*. A technology can be characterized by a set of, say, k state clauses $\Sigma = (S(i) \mid 1 \leq i \leq k)$. Any of the state clauses identifies a silicon layer structure chararacterizing a constituent of a set of electrical components. To be able to associate the silicon layer structure within a characterized technology Σ to a circuit schematic, there must be a one to one correspondence between the elements of Σ and the set of constituents of the circuit schematic, say C.

Assume $\sigma \in \Sigma$. Consider a point (x,y) such that $S(x,y)=\sigma$. Assume now that point is an inner point of a closed connected maximal set R with the property that for any $(\alpha,\beta) \in R$ we have $S(\alpha,\beta) = \sigma$. Then we call $R(\sigma)$ a *hard-structure*. After elimination of the z-dimension, the IC is considered as an open connected rectangular subset of the 2-D Euclidean space. There may be many hard-structures $R_i(\sigma)$, $i \in N$, on such a chip.

The state clause σ characterizes the circuit constituent $c \in C$ (or rather the type of circuit component in question); in addition a hard-structure R_i supplies all the geometrical information that completes the description of an instance of the respective circuit constituent. In other words, any $R_i(\sigma)$ corresponds to a circuit element.

The analysis of all the hard-structures induced by all the state clauses in Σ maps one to one into a circuit schematic. The process to compute a circuit schematic from a set of hard-structures is called *extraction*.

How the patterns in the layers are determined is specified by an IC *artwork*. An IC artwork represents geometrically the electrical information of the circuit design to be transformed by the processing steps. Formally, an *IC artwork* $\mathcal{L}$ is an ordered set of vector *masks* $(m_1,m_2,\ldots,m_J)$ which are concerned with the geometrical representation of the electrical information of a circuit design. Each mask, m_i, is geometrically defined as a 2-D euclidean space, i.e. $m_i = \{(x,y) \in E^2\}$. Let us denote $\mathcal{M} = \{m_j \mid j \in J\}$ as the set of masks, where $J = \{1,2,\ldots,K\}$, and K is the number of masks.

Within a mask, a connected point set O is defined by a finite set of line segments such that every segment extreme is shared by exactly two edges and no subset of edges has the same property. The set O partitions the mask in two disjoint regions: The interior which is bounded by O and the exterior which is unbounded (in fact it is bounded by the 2-D space of the mask). We denote the bounded region as the *opaque zone* and the unbounded as the *transparent zone*. Note that the transparent zone is in fact $\overline{O}$. When a lithographic process step, t_i, employs positive image projection, the image *Image*(m) of the shaped mask is identical to the opaque zone O, otherwise the mask image is identical to $\overline{O}$.

Thus, the shape obtained in the silicon layer structures are controlled by masks that are physically present as reticles, or for instance in the form of mask data controlling E-beam equipment. We can create a silicon layer structure by placing the masks on top of each other, correctly aligned to establish a *mask stacking* and in the sequence they are processed.

Very much as in the case of actual silicon layer structures we can define state clauses for any point (x,y) of a mask stacking. Any mask of the mask stacking may at any point be either opaque or transparent establishing essentially two sets per mask. In addition we use "X" as a don't care notion. If we denote $M=(M_1,M_2,\ldots,M_{|J|})$ as the set of state clauses associated with some mask stacking, then with any point $(x,y) \in E^2$ we have a boolean cube $M(x,y)$ as a vector of "1", "0" and "X" entries, where "1" stands for opaque and "0" indicates the attribute transparent.

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

By means of this notation we can also perform extraction on the mask stacking. In order to obtain consistency between the circuit schematic, the mask stacking, and the silicon structure layers we must establish a one to one correspondence between the set of *hard-structures* Σ, the set of *soft structures* $\mathcal{U}$ and the set of *circuit constituents* C. On one hand, this consistency is established by proper definitions of the set C and the sets Σ and $\mathcal{U}$. The relation between Σ and $\mathcal{U}$ is induced of course by the proper interpretation of the effects of the processing steps $t_i \in \mathcal{T}$.

3. FAULT MODELING BY PROCESS INDUCED DEFECTS

3.1 The modeling of defects and faults

Undesired *disturbances* can cause variations in the outcome of the manufacturing process of the IC. As an example consider a dust particle affecting hard-structures in such a form that active patterns are transformed into inactive patterns, and viceversa, or two or more conducting layers that become joint unintendly, etc.

Definition 1: If D is a function such that its domain $dom(D)=Z$ is the set of *defect mechanisms* capable of disturbing a silicon layer structure, and its range $ran(D)=L$, is the set of deformations. We say that D is a *defect transformation* from Z into L and write $D:Z\rightarrow L$. We also say that the IC has a defect if the state of any of the layers belonging to any "deformed" hard-structure has changed. We denote the set of "deformed" hard-structures as R_{def}.

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 (x,y,z) in the boundary of d^p belongs also to the boundary of 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.

Protrusion and intrusion defects generated during process step t_i affect active patterns at the same layer where they occurred, and may also have impact on some active patterns at different structure layers processed at some $t_j, j>i$. Isolated spot defects do not affect active patterns in the layer where they originated but may affect some active patterns in other different layers [7].

Often enough defects reproduce the silicon layer structure of a hard-structure yet cause a deviation of the shape of such structure. The transcendence of a defect is determined by the impact that it has on the image of the layer, this impact is called a *fault*.

Definition 2: A *fault model f* maps the set of altered hard-structures, R_{def} onto the fault class F, $f:R_{def}\rightarrow F$. The range of f, $ran(f)$, is the set of fault types, that is, classes of equivalence of faults including an empty class referring to a fault free state. The empty fault class has as its domain the set of hard-structures with the characteristic that their electrical properties are unaltered. This class of hard-structures are insensitive towards a given defect transformation, therefore we denote the defects as *benevolent defects*. The complement of this set of benevolent defects is the set of hard-structures which have assigned nonempty fault classes. We denote these kind of defects as *lethal defects*.

Our notation allows us 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 $\mathcal{U}$. 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 set of values may be $\mathcal{U}$ = {*X, thin oxide, field oxide, polysilicon*} and the presence of the isolated spot

would be indicated by some clause, say, σ = *(X, polysilicon, X, X)*. The shape of the defect would be captured by using the concept of a hard-structure.

Now, let us present some aspects concerning the observability of defects in hard-structures and the consistency requirements on fault modeling. Let us first introduce the notion of *structure graph* defined as $G_S = (S_H, E_S)$, with S_H the set of vertices and E_S the set of edges. Any $v_s \in S_H$ is associated with exactly one hard-structure. Any $e_s = (v_s, w_s) \in E_S$ defines the topological connection between any two hard-structures v_s and w_s.

Since defects can be represented by hard-structures, the presence of hard-structures induced by defects may imply changes in the structure graph. The general axiom however, is that the model is set up in such a way that any defect can be observed as a topological change of the structure graph relative to the structure graph of the defect free model. The topological change induced by defects in terms of additional hard-structures will either break paths (i.e. intrusion defects) or establish new ones by joining hard-structures together (i.e. protrusion defects affecting two or more hard-structures). For the former case the changes can be observed as new vertices appearing in the graph and for the latter case as a "melting" of vertices, thus as some missing vertices.

3.2 Process Disturbances and IC defects

Process disturbances can be characterized in terms of the physical nature of the disturbance or in terms of the effects that they have on the IC. An overview of the sources of random phenomena that occur in the manufacturing process is given in [8,9].

Defects can be classified as *geometrical* and *electrical-related* degradations. Geometrical degradations concern changes in the shape of a layer (*image defects*). *Process-related defects* concern changes in the electrical properties of the layer.

Image defects are present in those layers whose image *Image*(L) differs from their corresponding mask(s) image *Image*(M). These defects are mainly displacements of the initial boundary of some active pattern P_i caused by processing steps with anisotropic characteristics, such as lateral diffusion, lateral oxidation, over/under etching, over/under exposures of photoresit, etc. Process-related deformations concern changes in the thickness of the layers, p-n junction depth variations, changes in the 3-D impurity distributions of the conducting and semiconducting properties of a layer, and in changes in the charge distributions of the insulator properties of a layer.

If the same defects are present in all the dice of the wafer, we call them *global defects*. Examples are defects caused by mask misalignments, overetching, etc. If the defects are present only in several dice, we call them *local defects*. Examples are spot defects caused by isolated dust particles.

3.3 Hierarchical fault modeling framework

A simple method to handle fault-modeling complexity is 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 are 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 1.

At every level i of the hierarchy the fault-free representation is defined by a graph $G_i = (V_i, E_i)$. The set of vertices is given by $V_i = \{v_i \mid v_i \text{ is a primitive } \wedge\, pr(v_i) = V'_{i+1},\ V'_{i+1} \subset V_{i+1}\}$ where $pr(v_i)$ is a function returning the primitives at the next lower level of abstraction that constitute the current level primitive. The set of edges describing the topological interconnection flow between primitives is given by $E_i = \{(v_i, w_i) \mid v_i, w_i \in V_i \wedge \Delta(v_i, w_i) = conn(pr(v_i), pr(w_i))\}$, where $\Delta(v_i, w_i)$ is a function returning the topological path of the "logical" signal flow between the corresponding primitives of v_i and w_i at a lower level of the hierarchy. The physical level is represented by the structure graph $G_S = (S_H, E_S)$, for which $pr(v_s \in S_H)$ returns the set of structure layers that constitute v_s, and $\Delta(v_s \in S_H, w_s \in S_H)$ returns the layers which interconnect geometrically v_s and w_s.

TABLE 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 incongruencies.

Faults incurred at every level i of abstraction are similarly represented by a *fault graph* defined as $G_i^{(f)} = (V_i^{(f)}, E_i^{(f)})$, where $V_i^{(f)} = \{v_i^{(f)} \mid v_i^{(f)} \in V_i \wedge pr^*(v_i^{(f)}) = V'^{\{f\}}_{i+1},\ V'^{\{f\}}_{i+1} \subset V^{\{f\}}_{i+1} \wedge f_i : pr^*(v_i^{(f)}) \rightarrow F_i\}$ is the set of vertices describing faulty primitives with $pr^*(v_i^{(f)})$ a function returning the faulty primitives at a lower level in the hierarchy and F_i is the fault class at the current hierarchy level in which lower level faulty primitives are mapped. The set of edges relating primitives affected by the same spot defect is given as $E_i^{(f)} = \{(v_i^{(f)}, w_i^{(f)}) \mid v_i^{(f)}, w_i^{(f)} \in V_i^{(f)} \wedge \Delta^*(v_i^{(f)}, w_i^{(f)}) = conn(pr^*(v_i^{(f)}), pr(w_i^{(f)}))\}$, where $\Delta^*(v_i^{(f)}, w_i^{(f)})$ is a function returning the (faulty) topological path of the signal flow between the corresponding primitives of $v_i^{(f)}$ and $w_i^{(f)}$ at a lower level of the hierarchy. At the physical level $pr^*(v_i^{(f)})$ returns the faulty structure layers of $v_i^{(f)}$, and $\Delta^*(v_i^{(f)}, w_i^{(f)})$ returns the faulty structure layers common to $v_i^{(f)}$ and $w_i^{(f)}$.

Worth noticing is that F_i determines whether faulty primitives at a lower level of abstraction cause also a fault at the current 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.

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.

As an example consider Fig. 1. Fig 1a shows the different levels of abstraction of a flip-flop. Fig. 1b shows the fault free graph at every level of abstraction. The circles show the primitives in their corresponding level, the arrows point to primitives at a lower level that constitute the current primitive. Fig. 1c shows the faulty graph when an intrusion defect (represented as a square vertex) is present in DIF3. Notice that only the primitives related to the defect appear at every level of abstraction.

4. CONCLUSIONS.

In this paper a unified framework for describing spot defects and their impact at several levels of abstraction of the IC was provided. The framework encompasses a technology description that encloses process induced defects with their related defect mechanisms, and a hierarchical fault modeling based on the impact that spot defects have on the functional behavior of the IC. At every level of abstraction of the IC, graph representations were employed to model both fault-free and faulty elements, and the repercussions of those faulty elements at the next higher level of abstraction. This hierarchical way of fault modeling allows to decide which spot defects do really introduce faults, and also provides only relevant fault information at the specific level in the hierarchy.

5. REFERENCES.

[1] W Maly, "Realistic Fault Modelling for VLSI Testing",*24th IEEE/ACM Design Automation Conference"*, pp. 173-180.

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

[3] J.P. Shen, W. Maly, F.J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits", *IEEE Design and Test of Computers*, December 1985, pp. 13-26.

[4] H. Walker, S. Director, "VLASIC: A Catastrophic Fault Yield Simulator for Integrated Circuits", *IEEE Trans. Computer-Aided Design*, Vol. CAD-5, No. 4, October 1986, pp. 541-556.

[5] I. Chen, A. Strojwas, "Realistic Yield Simulation for IC Structural Failures", *IEEE Int. Conference on Computer Aided Design*, Nov. 1986, pp. 220-223.

[6] W. Maly, F. J. Ferguson, J. P. Shen, "Systematic Characterization of Physical Defects for Fault Analysis of MOS IC Cells", *1984 IEEE International Test Conference*, pp. 390-399.

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

[8] W. Maly, A. J. Strojwas, S. 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.

[9] Wojciech Maly, Will Moore, Andrezej Strojwas, "Yield Loss Mechanisms and Defect Tolerance", in *Yield Modelling and Defect Tolerance*, editors Will Moore, Wojciech Maly, and Andrezej Strojwas, publisher Adam Hilger, 1987, pp. 3-30.

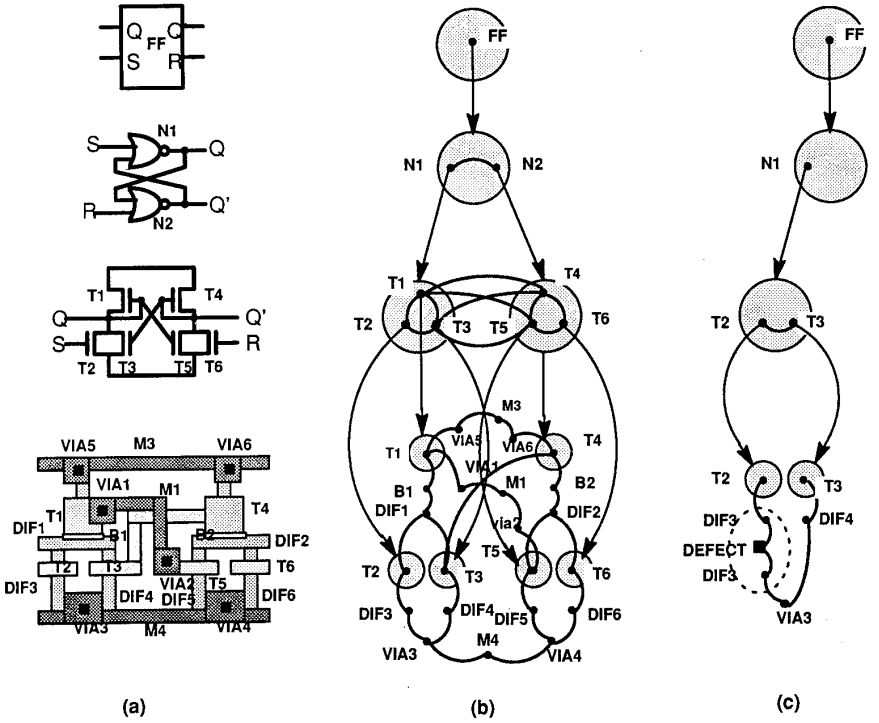

Fig. 1. Different levels of abstraction. (a) Fault–free graph. (b) Faulty graph with an intrusion defect in DIF3.