Defect Tolerance in VLSI Circuits: Techniques and Yield Analysis

ISRAEL KOREN, FELLOW, IEEE, and ZAHAVA KOREN

Current very-large-scale-integration (VLSI) technology allows the manufacture of large-area integrated circuits with submicrometer feature sizes, enabling designs with several millions of devices. However, imperfections in the fabrication process result in yield-reducing manufacturing defects, whose severity grows proportionally with the size and density of the chip. Consequently, the development and use of yield-enhancement techniques at the design stage, to complement existing efforts at the manufacturing stage, is economically justifiable. Design-stage yield-enhancement techniques are aimed at making the integrated circuit "defect tolerant," i.e., less sensitive to manufacturing defects. They include incorporating redundancy into the design, modifying the circuit floorplan, and modifying its layout.

Successful designs of defect-tolerant chips must rely on accurate yield projections. This paper reviews the currently used statistical yield-prediction models and their application to defect-tolerant designs. We then provide a detailed survey of various yield-enhancement techniques and illustrate their use by describing the design of several representative defect-tolerant VLSI circuits.

Keywords*—*Critical area, defects, defect tolerance, faults, floorplan, layout, redundancy, yield, yield model.

I. Introduction and Preliminaries

The profitability of integrated circuits (IC's) manufacturing depends heavily on the fabrication yield, defined as the proportion of operational circuits to the total number of fabricated circuits. A yield of 100% is unlikely, due to various manufacturing defects that exist even under mature manufacturing conditions. Continuous advances in manufacturing technologies have reduced the defect densities (e.g., by using cleaner rooms). However, reduction of the design feature size (down to submicrometers) and further increases in the chip area (up to almost 1 in^2) have increased the number and density of devices on a single die, resulting in, once again, a decreased fabrication yield. Thus, chip designers and manufacturers will continue to be concerned with manufacturing defects in the foreseeable future.

In this paper, we describe the nature of manufacturing defects and the way they affect the operation of a chip, and then show how to project the yield of a designed chip using statistical defect-distribution models. More important, we describe some defect-tolerance techniques for yield enhancement that can be employed during the design process, such as added redundancy and floorplan and layout modifications, and demonstrate their use in existing very-large-scale-integration (VLSI) circuits. Previous reviews related to the topic of this paper include survey papers [47], [64], [71], [73], [94], books [26], [31], [32], and an edited collection of articles [15].

Manuscript received January 8, 1998; revised May 29, 1998. This work was supported in part by the National Science Foundation under Contract MIP-9710130.

The authors are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003 USA.

Publisher Item Identifier S 0018-9219(98)06004-6.

A. Manufacturing Defects and Circuit Faults

We start by introducing some of the basic terminology used in yield analysis. Manufacturing defects can be roughly classified into gross area defects (or global defects) and spot defects. Global defects are relatively large-scale defects, such as scratches from wafer mishandling, large-area defects from mask misalignment, and over- and underetching. Spot defects are random local (i.e., small) defects from materials used in the process and from environmental causes, mostly the result of undesired chemical and airborne particles deposited on the chip during the various steps of the process.

Both types of defects contribute to the yield loss. In mature, well-controlled fabrication lines, gross area defects can be minimized and almost eliminated. Controlling random spot defects is considerably more difficult, and as a result, the yield loss due to spot defects is typically much higher than the yield loss due to global defects. This is especially true for large-area integrated circuits, since the frequency of global defects is almost independent of the die size, while the expected number of spot defects increases with the chip area. Consequently, spot defects are of greater significance when yield projection and enhancement are concerned, and they are the focus of this paper.

Spot defects can be divided into several types according to their location and to the potential harm they may cause. Some cause missing patterns, which may result in open circuits, while others cause extra patterns, which may result in short circuits. These defects can be further classified into intra- and interlayer defects. Intralayer defects occur as a result of particles deposited during the lithographic

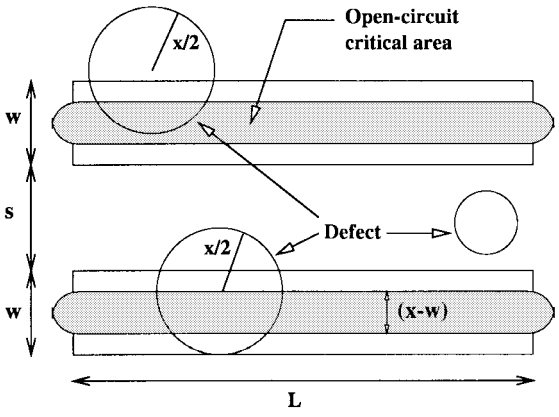


Fig. 1. The critical area for missing-metal defects of diameter x.

processes and are also known as photolithographic defects. Examples of these are missing metal, diffusion or polysilicon; and extra metal, diffusion or polysilicon. Also included are defects in the silicon substrate such as contamination in the deposition processes. Interlayer defects include missing material in the vias between two metal layers or between a metal layer and polysilicon; and extra material between the substrate and metal (or diffusion or polysilicon) or between two separate metal layers. These interlayer defects occur as a result of local contamination, e.g., dust particles.

Not all spot defects result in structural faults such as line breaks or short circuits. Whether or not a defect will cause a fault depends on its location and size and the layout and density of the circuit (see Fig. 1). For a defect to cause a fault, it has to be large enough to connect two disjoint conductors or disconnect a continuous pattern. Out of the three circular missing-material defects appearing in the layout of metal conductors in Fig. 1, the two top ones will not disconnect any conductor, while the bottom defect will result in an open circuit fault.

We make, therefore, the distinction between physical *defects* and circuit *faults*. A defect is any imperfection on the wafer, but only the fraction of defects that actually affect the circuit operation are called faults and are the ones causing yield losses. Thus, for the purpose of yield estimation, the distribution of faults, rather than that of defects, is of interest.

Some random defects that do not cause structural faults (also termed functional faults) may still result in parametric faults, i.e., the electrical parameters of some devices being outside the desired operational window, affecting the performance of the circuit. For example, a missing-material photolithographic defect may be too small to disconnect a transistor but may affect its performance. Parametric faults may also be the result of global defects, which cause variations in process parameters (see [19] and [87]). This paper concentrates on functional faults and does not deal with parametric faults.

B. Probability of Failure and Critical Area

We next describe how the fraction of manufacturing defects that result in functional faults can be calculated. This fraction, also called the *probability of failure* (POF), depends on the type of the defect, on its size (the larger the defect size, the higher the probability that it will cause a fault), and on the geometry of the circuit. A commonly adopted simplifying assumption is that a defect is circular with diameter x (as shown in Fig. 1). Accordingly, we denote by $\theta_i(x)$ the probability that a defect of type i and diameter x will cause a fault, and by θ_i the average POF for type i defects. Once $\theta_i(x)$ is calculated, θ_i can be obtained by averaging over all defect diameters x. Experimental data lead to the conclusion that the diameter x of a defect has a density function $f_d(x)$, which decreases as $1/x^p$ between x_0 and x_M [24], [95]. x_0 is usually the resolution limit of the lithography process [32] and x_M is the maximum size of a defect. The exact values of p and x_M can be determined empirically and may depend on the defect type. Typically, p ranges in value between 2 and 3.5 [58], [95]. Thus

$$f_d(x) = \begin{cases} \frac{k}{x^p} & \text{if } x_0 \le x \le x_M \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $k = (p-1)x_0^{p-1}x_M^{p-1}/(x_M^{p-1} - x_0^{p-1})$. θ_i can now be calculated as

$$\theta_i = \int_{x_0}^{x_M} \theta_i(x) f_d(x)\, dx. \tag{2}$$

Analogously, we define the critical area for defects of type i and diameter x, $A_i^{(c)}(x)$, as the size of the area in which the center of a defect of type i and diameter x must fall in order to cause a circuit failure, and by A_i^c the average over all defect diameters x of these areas. $A_i^{(c)}$ is called the critical area for defects of type i and can be calculated as

$$A_i^{(c)} = \int_{x_0}^{x_M} A_i^{(c)}(x) f_d(x)\, dx. \tag{3}$$

Assuming that given a defect, its center is uniformly distributed over the chip area, and denoting the chip area by A_{chip}, we obtain

$$\theta_i(x) = \frac{A_i^{(c)}(x)}{A_{\text{chip}}} \tag{4}$$

and consequently, based on (2) and (3)

$$\theta_i = \frac{A_i^{(c)}}{A_{\text{chip}}}. \tag{5}$$

Since the POF and the critical area are related through (5), any one of them can be calculated first. There are several methods of calculating these parameters. Some methods are geometry based, and they calculate $A_i^{(c)}(x)$ first, while in the Monte Carlo type methods, $\theta_i(x)$ is calculated first. We will briefly describe several methods for calculating the critical area/POF of an IC. For a more detailed description of how critical areas and POF's can be calculated, see [32, ch. 5].

We illustrate the geometrical method for calculating critical areas through the VLSI layout in Fig. 1, which shows two horizontal conductors. The critical area for a missing-material defect of size x in a conductor of length L and width w is the size of the shaded area in Fig. 1,

given by [45]

$$A_i^{(c)}(x) = \begin{cases} 0, & \text{if } x < w \\ (x-w)L + \frac{1}{2}(x-w)\sqrt{x^2-w^2}, & \text{if } x \geq w. \end{cases} \quad (6)$$

The critical area is a quadratic function of the defect diameter, but for $L \gg w$, the quadratic term becomes negligible. Thus, for long conductors, we can use just the linear term. An analogous expression for $A_i^{(c)}(x)$ for extra-material defects in a rectangular area of width s between two adjacent conductors can be obtained by replacing w by s in (6).

Other regular shapes can be similarly analyzed, and expressions for their critical area can be derived (e.g., [45]). Common VLSI layouts consist of many shapes in different sizes and orientations, and it is very difficult to derive the exact expression for the critical area of all but very simple and regular layouts. Therefore, other techniques have been proposed, including several more efficient geometrical methods and Monte Carlo simulation methods (e.g., [103]). One geometrical method is the polygon expansion technique, in which adjacent polygons are expanded by $x/2$ and the intersection of the expanded polygons is the critical area for short-circuit faults of diameter x (e.g., [31]). Other geometrical methods with a lower computation time have been developed [30], [32], [102]. A different geometrical method is the virtual artwork technique, in which an artificial layout is extracted from the given layout such that the estimation of the critical area is simplified [66].

In the Monte Carlo approach, simulated circles representing defects of different sizes are placed at random locations of the layout. For each such "defect," the circuit of the defective IC is extracted and compared with the defect-free circuit to determine whether the defect has resulted in a circuit fault. The POF $\theta_i(x)$ is calculated for defects of type i and diameter x as the fraction of defects that would have resulted in a fault. It is then averaged using (2) to produce θ_i and $A_i^{(c)} = \theta_i A_{\text{chip}}$. An added benefit of the Monte Carlo method is that the circuit fault resulting from a given defect is exactly identified. The Monte Carlo method has long been computation time consuming. Only recently have more efficient implementations been developed, allowing this method to be used for large IC's [93].

Once $A_i^{(c)}$ (or θ_i) is calculated for every defect type i, they can be used as follows. Let d_i denote the average number of defects of type i per unit area. Then the average number of manufacturing *defects* of type i on the chip is $A_{\text{chip}}d_i$. The average number on the chip of circuit *faults* of type i can now be expressed as $\theta_i A_{\text{chip}} d_i = A_i^{(c)} d_i$.

In the rest of this paper, we will assume that the defect densities are given and the critical areas are calculated. Thus, the average number of faults on the chip λ can be obtained using

$$\lambda = \sum_i A_i^{(c)} d_i = \sum_i \theta_i A_{\text{chip}} d_i \quad (7)$$

where the sum is taken over all possible defect types on the chip.

In Section II, we describe some basic yield models that can be used for predicting the yield of chips without any defect tolerance. Section III deals with defect tolerance through redundancy. We first extend the yield models described in Section II to chips with redundancy and then give some practical examples of memory chips and logic chips that have redundancy incorporated in their design. In Section IV, we describe two other techniques for yield enhancement, namely, layout modification and floorplan modification.

II. Basic Yield Models

To project the yield of a given chip design, some analytical probability model is necessary to describe the expected spatial distribution of manufacturing defects and, consequently, of the resulting circuit faults that eventually cause yield loss. The amount of detail needed regarding this distribution differs between chips that have some incorporated defect tolerance and those that do not. In the case of a chip with no defect tolerance, its projected yield is equal to the probability of no faults' occurring in the whole chip area. Denoting by X the number of faults on the chip, the chip yield, denoted by Y_{chip}, is given by

$$Y_{\text{chip}} = \text{Prob}(X = 0).$$

The yield is usually obtained by substituting $k = 0$ in the probability function $\text{Prob}(X = k)$. If the chip has some redundant components, projecting its yield requires a more intricate model, which will provide information regarding the distribution of faults over partial areas of the chip, as well as possible correlations among faults occurring in different subareas. In this section, we describe statistical yield models for chips without redundancy, while in Section III, we generalize these models for predicting the effects of redundancy on the yield.

A. The Poisson and Compound Poisson Yield Models

The most common statistical yield models appearing in the literature are the Poisson model and its derivatives—the compound Poisson models. Although other models have been suggested (e.g., [69]), we will concentrate in this paper on this family of distributions due to the ease of calculation when using the Poisson distribution, the relative ease of the integration (analytical or numerical) needed for the compounding, and the documented good fit of these distributions to empirical data [17].

Let λ denote the average number of faults occurring on the chip, i.e., the expected value of the random variable X. Assuming that the chip area is divided into a very large number n of small, statistically independent subareas, each with a probability λ/n of having a fault in it, we obtain the following binomial probability for the number of faults on the chip:

$$\begin{aligned} \text{Prob}(X = k) &= \text{Prob}\{k \text{ faults occur on chip}\} \\ &= \binom{n}{k} (\lambda/n)^k (1 - \lambda/n)^{n-k}. \end{aligned} \quad (8)$$

(a) (b)

Fig. 2. Effect of clustering on chip yield. (a) Nonclustered faults, $Y_{\text{chip}} = 0.5$. (b) Clustered faults, $Y_{\text{chip}} = 0.7$.

Letting $n \to \infty$ in (8) results in the Poisson distribution

$$\begin{aligned}\text{Prob}(X = k) &= \text{Prob}\{k \text{ faults occur on chip}\} \\ &= \frac{e^{-\lambda}\lambda^k}{k!} \end{aligned} \tag{9}$$

and the chip yield is equal to

$$Y_{\text{chip}} = \text{Prob}(X = 0) = e^{-\lambda}. \tag{10}$$

It has been known since the beginning of integrated circuit manufacturing that (10) is too pessimistic and leads to predicted chip yields that are too low when extrapolated from the yield of smaller chips or single circuits. It later became clear that the lower predicted yield was caused by the fact that defects, and consequently faults, do not occur independently in the different regions of the chip but rather tend to cluster more than is predicted by the Poisson distribution. Fig. 2 demonstrates how increased clustering of faults can increase the yield. The same six faults occur in both wafers, but the wafer in (b) has a higher yield due to the higher clustering.

Clustering of faults implies that the assumption that subareas on the chip are statistically independent, which led to (8) and consequently to (9) and (10), is erroneous. Several modifications to (9) have been proposed to account for fault clustering. The most commonly used modification is obtained by considering the parameter λ in (9) as a random variable rather than a constant. The resulting *compound Poisson distribution* produces a distribution of faults in which the different subareas on the chip are correlated, and which has a more pronounced clustering than that generated by the pure Poisson distribution.

The compounding procedure is demonstrated below. Let λ be the expected value of a random variable L with values l and a density function $f_L(l)$, where $f_L(l)\,dl$ denotes the probability that the chip fault average lies between l and $l + dl$. Averaging (or compounding) (9) with respect to this density function results in

$$\text{Prob}(X = k) = \int_0^\infty \frac{e^{-l}l^k}{k!} f_L(l)\,dl \tag{11}$$

and a chip yield given by

$$Y_{\text{chip}} = \text{Prob}(X = 0) = \int_0^\infty e^{-l} f_L(l)\,dl. \tag{12}$$

The function $f_L(l)$ in this expression is known as the *compounder* or *mixing function*. Any compounder must satisfy

$$\int_0^\infty f_L(l)\,dl = 1; \quad E(L) = \int_0^\infty l f_L(l)\,dl = \lambda.$$

Murphy [70] used as a compounder the triangular density function

$$f_L(l) = \begin{cases} \frac{l}{\lambda^2}, & 0 \le l \le \lambda \\ \frac{2\lambda - l}{\lambda^2}, & \lambda \le l \le 2\lambda \end{cases} \tag{13}$$

which results in the following expression for the chip yield:

$$Y_{\text{chip}} = \text{Prob}(X = 0) = \int_0^{2\lambda} e^{-l} f_L(l)\,dl = \left(\frac{1 - e^{-\lambda}}{\lambda}\right)^2. \tag{14}$$

Seeds [84] suggested the exponential density function

$$f_L(l) = \frac{e^{-l/\lambda}}{\lambda} \tag{15}$$

which gives a yield of

$$Y_{\text{chip}} = \text{Prob}(X = 0) = \int_0^\infty e^{-l} f_L(l)\,dl = \frac{1}{1 + \lambda}. \tag{16}$$

Okabe [77] and Stapper [88] suggested using as a mixing function the Gamma distribution with the two parameters λ and α

$$f_L(l) = \frac{\alpha^\alpha}{\lambda^\alpha \Gamma(\alpha)} l^{\alpha - 1} e^{-\frac{\alpha}{\lambda} l}. \tag{17}$$

Evaluating the integral in (11) with respect to (17) results in the well-known *negative binomial* yield formula

$$\text{Prob}(X = k) = \frac{\Gamma(\alpha + k)}{k!\Gamma(\alpha)} \frac{(\lambda/\alpha)^k}{(1 + \lambda/\alpha)^{\alpha + k}} \tag{18}$$

and

$$Y_{\text{chip}} = \text{Prob}(X = 0) = (1 + \lambda/\alpha)^{-\alpha}. \tag{19}$$

This last model is also called the *large-area clustering* negative binomial model. It implies that the whole chip constitutes one unit and that subareas within the same chip are correlated with regard to faults. The negative binomial yield model has two parameters and is therefore more flexible and easier to fit to actual data than the previously mentioned distributions. The parameter λ is the average number of faults per chip, while the parameter α is a measure of the amount of fault clustering, and smaller values of α indicate increased clustering. Actual values for α typically range between 0.3 and 5. The Seeds model (16) is a special case of (19) for $\alpha = 1$. When $\alpha \to \infty$, (19) becomes equal to (10), which represents the yield under the Poisson distribution, characterized by total absence of theoretical clustering. (In practice, there will be some clustering even under the Poisson distribution, due to the deviation of actual measurements from their theoretical expected values.)

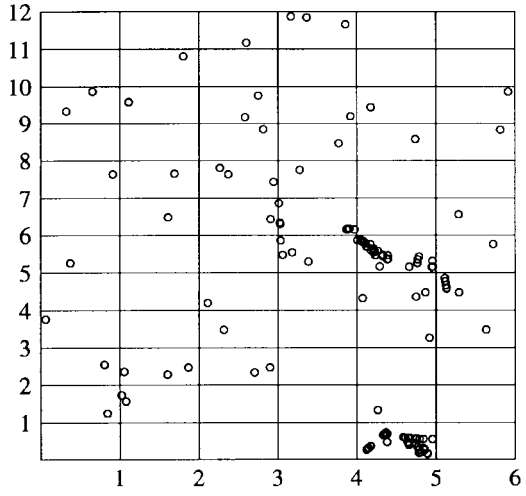


Fig. 3. A wafer defect map.

B. Variations on the Simple Yield Models

The large-area clustering compound Poisson models described above use two crucial assumptions—that the fault clusters are large compared to the size of the chip and that they are of uniform size. In some cases, it is clear from observing the defect maps of the manufactured wafers that the faults can be divided into two classes: heavily clustered and less heavily clustered (see Fig. 3) and clearly originate from two sources: systematic and random. In these cases, a simple yield model as described above will not be able successfully to describe the fault distribution. This inadequacy will be more noticeable when attempting to evaluate the yield of chips with redundancy. One solution that has been suggested in the past is including in the model a gross yield factor Y_0, denoting the probability that the chip is *not* hit by a gross defect. Gross defects are usually the result of systematic processing problems that affect whole wafers or parts of wafers. They may be caused by misalignment, over- or underetching, or out-of-spec semiconductor parameters such as beta transconductance or threshold voltage. It is shown in [78] that even fault clusters with very high fault densities can be modeled by Y_0. If the negative binomial yield model is used, then introducing a gross yield factor Y_0 results in

$$Y_{\text{chip}} = Y_0(1+\lambda/\alpha)^{-\alpha}. \tag{20}$$

As chips become larger, this approach becomes less practical, as very few faults will hit the whole chip. Instead, combining two fault distributions, each with a different set of parameters, has been suggested in [50]. X, the total number of faults on the chip, can be viewed as $X = X_1 + X_2$, where X_1 and X_2 are statistically independent random variables, denoting the number of faults of type 1 and of type 2, respectively, on the chip. The probability function of X can be derived from

$$\text{Prob}(X=k) = \sum_{j=0}^{k} \text{Prob}(X_1=j) \times \text{Prob}(X_2=k-j) \tag{21}$$

and

$$Y_{\text{chip}} = \text{Prob}(X=0) = \text{Prob}(X_1=0) \times \text{Prob}(X_2=0). \tag{22}$$

If X_1 and X_2 are modeled by a negative binomial distribution with parameters λ_1, α_1 and λ_2, α_2, respectively, then

$$Y_{\text{chip}} = (1+\lambda_1/\alpha_1)^{-\alpha_1}(1+\lambda_2/\alpha_2)^{-\alpha_2}. \tag{23}$$

Another variation on the simple fault distributions may occur in very large chips, where the fault clusters appear to be of uniform size but are much smaller than the chip area. In this case, instead of viewing the chip as one entity for statistical purposes, it can be viewed as consisting of statistically independent regions (called *blocks* in [49]). The number of faults in each block has a negative binomial distribution, and the faults within the area of the block are uniformly distributed. The large-area negative binomial distribution is a special case where the whole chip constitutes one block. Another special case is the small-area negative binomial distribution [98], which describes very small independent fault clusters and is sometimes confused with the Poisson distribution. Mathematically, the medium-area negative binomial distribution can be obtained, similarly to the large-area case, as a compound Poisson distribution, where the integration in (11) is performed independently over the different regions of the chip. Let the chip consist of B blocks and have an average of l faults. Each block will have an average of l/B faults, and according to the Poisson distribution, the chip yield will be

$$Y_{\text{chip}} = e^{-l} = \left(e^{-l/B}\right)^B \tag{24}$$

where $e^{-l/B}$ is the yield of one block.

When each factor in (24) is compounded separately with respect to (17), the result is

$$Y_{\text{chip}} = \left[\left(1+\frac{\lambda/B}{\alpha}\right)^{-\alpha}\right]^B = \left(1+\frac{\lambda}{B\alpha}\right)^{-B\alpha}. \tag{25}$$

It is also possible that each region on the chip has a different sensitivity to defects, and thus, block i has the parameters λ_i, α_i, resulting in

$$Y_{\text{chip}} = \prod_{i=1}^{B}\left(1+\frac{\lambda_i}{\alpha_i}\right)^{-\alpha_i}. \tag{26}$$

It is important to note that the differences among the various models described in this section become more noticeable when they are used to project the yield of chips with built-in redundancy.

To estimate the parameters of the yield model, some variation of the "window method" [47], [77], [78], [84], [97] is regularly used in the industry. Wafer maps that show the location of functioning and failing chips are analyzed using overlays with grids, or windows. These windows contain some chip multiples (e.g., one, two, and four), and the yield for each such multiple is calculated. Values

for the parameters Y_0, λ, and α are then determined by means of curve fitting. The "window method" has been extended in [49] to include estimation of the block size for the medium-area clustering yield model.

III. Yield Enhancement Through Redundancy

A. Yield Projection for Chips with Redundancy

In many integrated circuit chips, identical blocks of circuits are often replicated. In memory chips, these are blocks of memory cells that are also known as *subarrays*. In processor arrays, these basic circuit blocks are referred to as *processing elements*. In other digital chips, they are referred to as *macros*. We will use the term *modules* to include all these designations.

In very large chips, if the whole chip is expected to be fault free, the yield will be very low. The yield can be increased by adding a few spare modules to the design and accepting those chips that have the required number of fault-free modules. Clearly, the more spares added, the higher the resulting yield will be. However, adding redundant modules increases the chip area and reduces the number of chips that will fit into the wafer area. Consequently, a better measure for evaluating the benefit of redundancy is the *effective yield*, defined as

$$Y_{\text{chip}}^{\text{eff}} = Y_{\text{chip}} \frac{A_{\text{chip\_without\_redundancy}}}{A_{\text{chip\_with\_redundancy}}}. \tag{27}$$

The maximum value of $Y_{\text{chip}}^{\text{eff}}$ determines the optimal amount of redundancy to be incorporated into the chip.

The yield of a chip with redundancy is the probability that it has enough fault-free modules for proper operation. To calculate this probability, a much more detailed statistical model than described earlier is needed, a model that specifies the fault distribution for any subarea of the chip as well as the correlations among the different subareas of the chip.

1) Chips with One Type of Module: For simplicity, let us first deal with projecting the yield of chips whose only circuitry is N identical modules, out of which R are spares and at least $N-R$ must be fault free for proper operation. Define the following probability:

$$F_{MN} = \text{Prob}\{\text{Exactly } M \text{ out of the } N \text{ modules are fault-free.}\}$$

Then the yield of the chip is given by

$$Y_{\text{chip}} = \sum_{M=N-R}^{N} F_{MN}. \tag{28}$$

Using the spatial Poisson distribution implies that for any partial area of size a of the chip, the number of faults occurring in this area has a Poisson distribution, with a parameter (which is also the average number of faults in this area) equal to $\lambda a/A_{\text{chip}}$, where A_{chip} is the chip area and λ is the average number of faults in the whole chip. The average number of faults per module $\lambda^{(m)}$ is therefore $\lambda^{(m)} = \lambda/N$. In addition, when using the Poisson model, the faults in any distinct subareas are statistically independent, and thus

$$\begin{aligned} F_{MN} &= \binom{N}{M}\left(e^{-\lambda^{(m)}}\right)^M\left(1-e^{-\lambda^{(m)}}\right)^{N-M} \\ &= \binom{N}{M}(e^{-\lambda/N})^M\left(1-e^{-\lambda/N}\right)^{N-M} \end{aligned} \tag{29}$$

and the yield of the chip is

$$Y_{\text{chip}} = \sum_{M=N-R}^{N}\binom{N}{M}(e^{-\lambda/N})^M\left(1-e^{-\lambda/N}\right)^{N-M}. \tag{30}$$

Although the Poisson distribution lends itself very easily to yield calculations, unfortunately it does not match actual defect and fault data. If any of the compound Poisson distributions is to be used, then the different modules on the chip are not statistically independent but rather correlated with respect to the number of faults. A simple formula like (30), which uses the binomial distribution, is therefore not appropriate. There are several approaches to calculating the yield in this case, all leading to the same final expression [47].

The first approach applies only to the compound Poisson models and is based on compounding the yield expression in (30) over $\lambda^{(m)}$ (as shown in Section II). Replacing λ/N by l, expanding $(1-e^{-l})^{N-M}$ into the binomial series $\sum_{k=0}^{N-M}(-1)^k\binom{N-M}{k}(e^{-l})^k$, and substituting into (30) results in

$$F_{MN} = \binom{N}{M}\sum_{k=0}^{N-M}(-1)^k\binom{N-M}{k}(e^{-l})^{M+k}. \tag{31}$$

By compounding (31) with a density function $f_L(l)$, we obtain

$$F_{MN} = \binom{N}{M}\sum_{k=0}^{N-M}(-1)^k\binom{N-M}{k}\int_0^\infty e^{-(M+k)l} f_L(l)\,dl.$$

Denoting $y_n = \int_0^\infty e^{-nl} f_L(l)\,dl$ (y_n is the probability that a *given* subset of n modules is fault free, according to the compound Poisson model) results in

$$F_{MN} = \binom{N}{M}\sum_{k=0}^{N-M}(-1)^k\binom{N-M}{k} y_{M+k} \tag{32}$$

and the yield of the chip is equal to

$$Y_{\text{chip}} = \sum_{M=N-R}^{N}\sum_{k=0}^{N-M}(-1)^k\binom{N}{M}\binom{N-M}{k} y_{M+k}. \tag{33}$$

y_{M+k} can be replaced by any of the expressions (10), (14), (16), or (19) with λ replaced by $(M+k)\lambda^{(m)} = (M+k)\lambda/N$. The Poisson model can be obtained as a special case by substituting

$$y_{M+k} = e^{-(M+k)\lambda/N}$$

while for the negative binomial model

$$y_{M+k} = \left(\frac{1+(M+k)\lambda}{N\alpha}\right)^{-\alpha} \tag{34}$$

and the yield of the chip is

$$Y_{\text{chip}} = \sum_{M=N-R}^{N} \sum_{k=0}^{N-M} (-1)^k \binom{N}{M} \times \binom{N-M}{k} \left(\frac{1+(M+k)\lambda}{N\alpha}\right)^{-\alpha}. \tag{35}$$

The approach described above to calculating the chip yield applies only to the compound Poisson models. A more general approach involves using the well-known inclusion and exclusion principle in order to calculate the probability F_{MN}. Defining as the desired event the event in which the ith module is fault free, F_{MN} is the probability of exactly M such events' occurring simultaneously, and according to the inclusion and exclusion principle

$$F_{MN} = \binom{N}{M} \sum_{k=0}^{N-M} (-1)^k \binom{N-M}{k} y_{M+k} \tag{36}$$

which is the same expression as (32), which leads to (33).

Since (33) can be obtained from the basic inclusion and exclusion principle, it is quite general and applies to a larger family of distributions than the compound Poisson models. The only requirement for it to be applicable is that for a given n, any subset of n modules has the same probability of being fault free, and no statistical independence among the modules is required.

As shown above, the yield for any compound Poisson distribution (including the pure Poisson) can be obtained from (33) by substituting the appropriate expression for y_n. If a gross yield factor Y_0 exists, it can be included in y_n. For the model in which the defects arise from two sources and the number of faults per chip X can be viewed as $X = X_1 + X_2$

$$y_n = y_n^{(1)} y_n^{(2)}$$

where $y_n^{(i)}$ denotes the probability that a given subset of n modules has no type i faults $(i = 1, 2)$. The calculation of y_n for the medium-size clustering negative binomial probability is slightly more complicated and will not be included here. It can be found in [49].

2) More Complex Designs: The simple architecture analyzed in the preceding section is an idealization, since actual chips rarely consist entirely of identical circuit modules. The more general case is that of a chip with multiple types of modules, each with its own redundancy. In addition, all chips include support circuits that are shared by the replicated modules. The support circuitry almost never has any redundancy and, if damaged, renders the chip unusable. In what follows, we derive yield expressions for chips with two different types of modules, and some support circuits. The extension to a larger number of module types is straightforward but cumbersome and is therefore not presented here.

Denote by N_i the number of type i modules, out of which R_i are spares. Each type i module occupies an area of size a_i on the chip $(i = 1, 2)$. The area of the support circuitry is a_{ck} ("ck" stands for chip kill, since any fault in the support circuitry is fatal for the chip). Clearly, $N_1 a_1 + N_2 a_2 + a_{\text{ck}} = A_{\text{chip}}$.

Since each circuit type has a different sensitivity to defects, it has a different fault density. Let $\lambda_1^{(m)}$, $\lambda_2^{(m)}$, and λ_{ck} denote the average number of faults per type 1 module, type 2 module, and the support circuitry, respectively. Denoting by F_{M_1,N_1,M_2,N_2} the probability that exactly M_1 type 1 modules, exactly M_2 type 2 modules, and all the support circuits are fault free, the chip yield is given by

$$Y_{\text{chip}} = \sum_{M_1=N_1-R_1}^{N_1} \sum_{M_2=N_2-R_2}^{N_2} F_{M_1,N_1,M_2,N_2}. \tag{37}$$

According to the Poisson distribution

$$F_{M_1,N_1,M_2,N_2} = \binom{N_1}{M_1} \left(e^{-\lambda_1^{(m)}}\right)^{M_1} \left(1 - e^{-\lambda_1^{(m)}}\right)^{N_1-M_1} \binom{N_2}{M_2} \times \left(e^{-\lambda_2^{(m)}}\right)^{M_2} \left(1 - e^{-\lambda_2^{(m)}}\right)^{N_2-M_2} e^{-\lambda_{\text{ck}}}. \tag{38}$$

To get the expression for F_{M_1,N_1,M_2,N_2} under a general fault distribution, we need to use the two-dimensional inclusion and exclusion principle

$$F_{M_1,N_1,M_2,N_2} = \sum_{\substack{M_1= \\ N_1-R_1}}^{N_1} \sum_{\substack{M_2= \\ N_2-R_2}}^{N_2} \sum_{k_1=0}^{N_1-M_1} \sum_{k_2=0}^{N_2-M_2} (-1)^{k_1} \times (-1)^{k_2} \binom{N_1}{M_1} \binom{N_1-M_1}{k_1} \binom{N_2}{M_2} \times \binom{N_2-M_2}{k_2} y_{M_1+k_1,M_2+k_2} \tag{39}$$

where y_{n_1,n_2} is the probability that a given set of n_1 type 1 modules, a given set of n_2 type 2 modules, and the support circuitry are all fault free. This probability can be calculated using any of the models described in Section II with λ replaced by $n_1\lambda_1^{(m)} + n_2\lambda_2^{(m)} + \lambda_{\text{ck}}$.

Two noted special cases are the Poisson distribution, for which

$$y_{n_1,n_2} = \left(e^{-\lambda_1^{(m)}}\right)^{n_1} \left(e^{-\lambda_2^{(m)}}\right)^{n_2} e^{-\lambda_{\text{ck}}} \tag{40}$$

and the large-area negative binomial distribution, for which

$$y_{n_1,n_2} = \left(1 + \frac{n_1\lambda_1^{(m)} + n_2\lambda_2^{(m)} + \lambda_{\text{ck}}}{\alpha}\right)^{-\alpha}. \tag{41}$$

Some chips (e.g., [107]) have a very complex redundancy scheme that does not conform to the simple M out of N redundancy. In these cases, it would be extremely difficult to develop closed yield expressions for any model with clustered faults (i.e., any model other than the Poisson model). One possible solution is using Monte Carlo simulation, in which faults are thrown at the wafer randomly, according to the statistical underlying model, and the percentage of

operational chips is calculated. Another solution that is much less time consuming is calculating the yield using the Poisson distribution, which is relatively easy (although for complicated redundancy schemes it may require some nontrivial combinatorial calculations). This yield is then compounded with respect to λ using an appropriate compounder. If the Poisson yield expression can be expanded into a power series in λ, analytical integration is possible. Otherwise, which is more likely, numerical integration must be performed. This very powerful compounding procedure was employed to derive yield expressions for interconnection buses in VLSI chips [46], for partially good memory chips [99], and for hybrid redundancy designs of memory chips [51], [53].

B. Memory Arrays with Redundancy

Defect-tolerance techniques have been successfully applied to many designs of memory arrays since the late 1970's due to their high regularity, which greatly simplifies the task of incorporating redundancy into their design. A variety of defect-tolerance techniques have been exploited in memory designs, from the simple technique using spare rows and columns (also known as word lines and bit lines, respectively) through the use of error-correcting codes [48]. These techniques have been successfully employed by many semiconductor manufacturers, resulting in significant yield improvements ranging from 30-fold increases in the yield of early prototypes to 1.5–3-fold yield increases in mature processes.

One of the earliest implementations of defect-tolerant memory array was a 16 Kb chip designed at IBM [82]. It included six redundant bit lines, four redundant word lines, and the associated decoders, resulting in an added area of 7%. A defective row, for example, or a row containing one or more defective memory cells can be disconnected by blowing a fusible link [48]. The disconnected row is then replaced by a spare row, which has a programmable decoder with fusible links, allowing it to replace any defective row. It has been estimated [82] that the yield of the chip with no redundancy would have been less than 2%, increasing to 31% with the added redundancy. One of the main reasons for the still-low overall yield was that only faults in the memory array (and not all of them) could be taken care of by the redundant bit and word lines. Any faults in the remaining 17% of the chip were chip-kill faults, which could not be fixed by redundancy.

There were also a few attempts at incorporating other redundancy techniques into memory designs. For example, a memory chip designed at Hughes Aircraft [33] included spare blocks to be used upon a failure of several cells in the main array of cells. A small associative memory was included in the chip, and the addresses of faulty locations were stored there, directing the incoming addresses to the spare blocks.

A more recent nontraditional design of a defect-tolerant memory was reported in [38]. A 16-Mb dynamic random-access memory chip employing the conventional redundancy technique (using spare rows and columns) as well as an error-correcting code (ECC) was designed at IBM. The chip includes four independent quadrants with 16 redundant bit lines and 24 redundant word lines per quadrant. In addition, for every 137 data bits, nine check bits were added to allow the correction of any single bit error within these 137 bits. To reduce the probability of two or more faulty bits in the same word (due to clustered faults, for example), every eight adjacent bits in the quadrant were assigned to eight separate words. It was demonstrated in [38] that the benefit of the combined strategy for yield enhancement was larger than the sum of the expected benefits of the two individual techniques. The reason for this is that the ECC technique is very effective against individual cell failures, while redundant rows and columns are very effective against several defective cells within the same row or column, as well as completely defective rows and columns. The ECC technique is commonly used in large memory systems to protect against intermittent faults' occurring while the memory is in operation in order to increase its reliability. The reliability improvement due to the use of the ECC was shown to be only slightly affected by the use of the check bits to correct defective memory cells.

Still, the traditional method for incorporating defect tolerance in memory IC's through redundant rows and columns has been used more often than any other technique and proved to be extremely successful for more than 15 years. This technique has even been incorporated in the design of large cache units in microprocessors in the last five years. The advantage of employing redundant rows and columns has been especially significant in the early stages of production when the yield is still low, allowing for earlier introduction of new products into the market.

Increases in the size of memory chips in the last several years made it necessary to partition the memory array into several subarrays in order to decrease the current and reduce the access time by shortening the length of the bit and word lines [106]. Using the conventional redundancy method implied that each subarray should have its own spare rows and columns, leading to situations where one subarray had an insufficient number of spare lines to handle local faults while other subarrays still had some unused redundant lines.

As memory IC's become denser, the submicrometer process technology becomes more complex and the manufacturing yield is expected to decrease [106]. Consequently, defect-tolerance techniques are important not only in the early stages of the production but also in the mass-production stages. It became apparent, therefore, that new and more efficient redundancy techniques must be developed. One obvious approach is to turn some (or even all) of the local redundant lines into global redundant lines, allowing for a more efficient use of the spare lines at the cost of higher silicon area overhead due to the larger number of required programmable fuses. This approach has been followed in [106], where the design of an experimental 4-Mb static RAM at Mitsubishi was presented. A 3% increase in the area overhead and up to 61% increase in effective yield [see (27)] have been reported there.

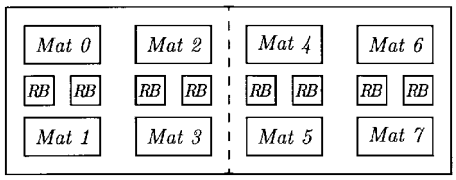


Fig. 4. A 1-Gb chip with eight mats of size 128 Mb each and eight RB's of size 1 Mb each.

Several other approaches were proposed and implemented in recent years [40], [41], [100], [106], [107]. One such approach has appeared in [100], describing the design, at NEC, of a flexible multimacro (FMM) 1-Gb DRAM in 0.25 μm complementary metal–oxide–semiconductor (CMOS) technology. This design used fewer redundant lines than the traditional technique, and the redundant lines were kept local. For added defect tolerance, each subarray of size 256 Mb (which was called macro and constituted a quarter of the chip) was fabricated in such a way that it could become part of up to four different memory IC's.

To allow this flexibility, the area of the macro had to be increased by 2%. To keep the overall area of the macro identical to that in the conventional design, row redundancy was eliminated, thus saving about 2% of the total area, but column redundancy was still implemented. Furthermore, since the chip boundaries were not predetermined, 16 additional macros were fabricated on each 8-in wafer beyond the original 96 macros (constituting 24 IC's), allowing further flexibility in combining macros to form IC's.

The yield of the FMM chip was analyzed in [51] and compared to the yield of the same size chip with the conventional row and column redundancy technique. It has been shown there that if the faults are almost evenly distributed (i.e., the Poisson distribution can be used), there is almost no advantage in using the new design. There is, however, a considerable increase in yield if the medium-area negative binomial distribution (described in Section II) is used. The improvement in yield is highly dependent on the exact values of the fabrication parameters.

Recently, another approach for incorporating defect tolerance into memory IC's has been proposed and implemented at Samsung [107]. This is a hybrid design that combines row and column redundancy with several redundant subarrays whose purpose is to replace those subarrays hit by chip-kill faults. The designed chip is a 1-Gb memory that includes eight mats of size 128 Mb each and eight redundant blocks (RB's) of size 1 Mb each (see Fig. 4). The redundant block consists of four basic 256 Kb arrays and has an additional eight spare rows and four spare columns (see Fig. 5). The purpose of the spare rows and columns is to increase the probability that the redundant block is operational and can be used for replacing a block with chip-kill faults.

Each mat consists of 512 basic arrays of size 256 Kb and has 32 spare rows and 32 spare columns. However, these spare rows and columns cannot be used to replace every defective row or column in the entire mat. Four spare rows are allocated to a 16-Mb portion of the mat, and eight spare columns are allocated to a 32-Mb portion of the mat.

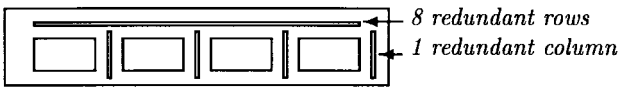


Fig. 5. A redundant block including four 256 Kb arrays, eight redundant rows, and four redundant columns.

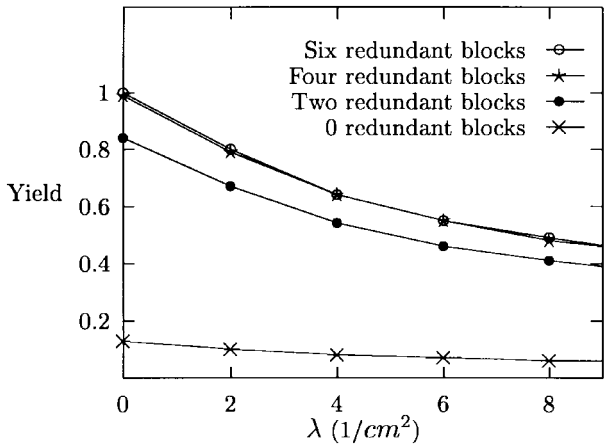


Fig. 6. Yield as a function of λ for different numbers of redundant blocks per half chip (chip-kill probability $= 5 \times 10^{-4}$).

The yield of this new design of a memory chip was analyzed in [53] and compared to that of the traditional design with only row and column redundancy. Fig. 6 shows the yield of the chip with different numbers of redundant blocks, clearly demonstrating the benefits of some amount of block redundancy. The increase in the yield is much higher than the 2% area increase required for the redundant blocks. Further analysis in [53] has shown that column redundancy is still beneficial even when redundant blocks are incorporated, and that the optimal number of such redundant columns is independent of the number of spare blocks.

C. Logic Integrated Circuits with Redundancy

In contrast to memory arrays, very few logic IC's have been designed with any built-in redundancy. Some regularity in the design is necessary if a low overhead for redundancy inclusion is desired. For completely irregular designs, duplication and even triplication are currently the only available redundancy techniques, and these are impractical due to their large overhead. Regular circuits like programmable logic arrays (PLA's) [104] and processor arrays [5] require less redundancy, and consequently, various defect-tolerance techniques have been proposed (and some implemented) for their designs in order to enhance their yield [55], [60], [105]. These techniques, however, require extra circuits such as spare product terms, reconfiguration switches, and additional input lines to allow the identification of faulty product terms [60]. Unlike memory IC's, where all defective cells can be identified by applying external test patterns, the identification of defective elements in logic IC's (even for those with regular structure) is more complex and usually requires the addition of some built-in testing aids. Thus, testability must also be a factor in choosing defect-tolerant designs for logic IC's.

The situation becomes even more complex in random logic circuits like microprocessors. When designing such circuits, it is necessary to partition the design into separate components, preferably with each having a regular structure. Then, different redundancy schemes can be applied to the different components, including the possibility of no defect tolerance in components for which the cost of incorporating redundancy becomes prohibitive.

We describe next two experimental designs: a defect-tolerant microprocessor and a wafer-scale design. These experiments demonstrate the feasibility of incorporating defect tolerance for yield enhancement in the design of processors and prove that the use of defect tolerance is not limited to the highly regular memory arrays.

The `Hyeti` microprocessor is a 16-b defect-tolerant microprocessor that was designed and fabricated as part of the European ESPRIT project [59], [60] to demonstrate the feasibility of a high-yield, defect-tolerant microprocessor. This microprocessor may be used as the core of an application-specific microprocessor-based system that is integrated on a single chip. The large silicon area consumed by such a system would most certainly result in a low yield unless some defect tolerance in the form of redundancy were incorporated into the design.

The data path of the microprocessor contains several functional units like registers, an arithmetic and logic unit, bus circuitry etc. Almost all the units in the data path have circuits that are replicated 16 times, leading to the classic bit-slice organization. This regular organization was exploited for yield enhancement by providing a spare slice, which can replace a defective slice. Not all the circuits in the data path, though, consist of completely identical subcircuits. The status register, for example, has each bit associated with a unique random logic and therefore has no added redundancy.

The control part has been designed as a hardwired control circuit that can be implemented using PLA's only. The regular structure of a PLA allows a straightforward incorporation of redundancy for yield enhancement through the addition of spare product terms [55], [104], [105]. The design of the PLA has been modified to allow the identification of defective product terms. The numbers of redundant terms that have been added to the seven PLA's and to the data path in the `Hyeti` microprocessor are, respectively, 2, 2, 2, 2, 4, 4, 1, 1 [59].

A detailed yield analysis of this microprocessor (presented in [60]) confirmed that the optimal redundancy for the data path is a single 1-b slice. The optimal redundancy for all the PLA's, however, was determined to be one. A higher than optimal redundancy was implemented in most PLA's, since the floorplan of the control unit allowed the addition of a few extra product terms to the PLA's with no area penalty. A practical yield analysis should take into consideration the exact floorplan of the chip and allow the addition of a limited amount of redundancy beyond the optimal amount. However, not all the available area should be used up for extra spares, since this will increase the switching area, which will in turn increase the chip-kill area. This higher chip-kill area can at some point offset the yield increase resulting from the added redundancy.

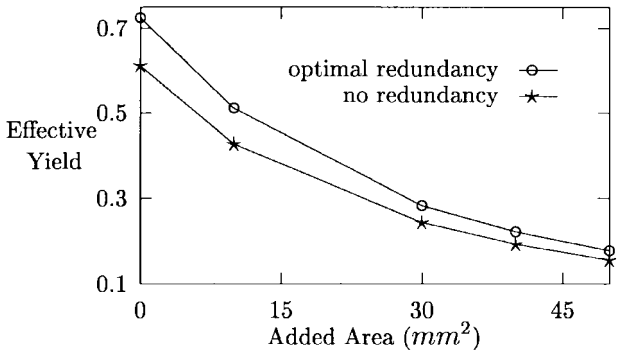


Fig. 7. The effective yield as a function of the added area, without redundancy and with optimal redundancy, for $\lambda = 0.05/\text{mm}^2$ and $\alpha = 2$.

Fig. 7 depicts the effective yield [see (27)] without redundancy in the microprocessor and with the optimal redundancy as a function of the area of the circuitry added to the microprocessor, which serves as a controller of an application-specific microprocessor-based integrated circuit. The figure shows that an increase in yield of about 18% can be expected when the optimal amount of redundancy is incorporated in the design.

The second experiment with defect tolerance in non-memory designs, described next, is the three-dimensional (3-D) computer, an example of a wafer-scale design. The 3-D computer, designed by Hughes Research Laboratories [108], is a cellular array processor implemented in wafer scale integration technology. The most unique feature of its implementation is its use of stacked wafers. The basic processing element is divided into five functional units, each of which is implemented on a different wafer. Thus, each wafer contains only one type of functional unit and includes spares for yield enhancement as explained below. Units in different wafers are connected vertically through microbridges between adjacent wafers to form a complete processing element. The first working prototype of the 3-D computer was of size 32×32. The second prototype included 128×128 processing elements.

Defect-tolerance in each wafer is achieved through an interstitial redundancy scheme [86], where the spare units are uniformly distributed in the array and are connected to the primary units with local and short interconnects. In the 32×32 prototype, a (1,1) redundancy scheme was used, i.e., each primary unit has a separate spare unit. A (2,4) scheme was used in the 128×128 prototype. In this scheme, each primary unit is connected to two spare units, and each spare unit is connected to four primary units, resulting in a redundancy of 50% rather than the 100% for the (1,1) scheme. The (2,4) interstitial redundancy scheme can be implemented in a variety of ways. The exact implementation in the 3-D computer and its effect on the yield are further discussed in Section IV-B.

Since it is highly unlikely that a whole fabricated wafer will be fault free, the yield of the processor would be zero

if no redundancy were included. With the implemented redundancy, the observed yield of the 32 × 32 array after repair was 45%. For the 128 × 128 array, the (1,1) redundancy scheme would have resulted in a very low yield (about 3%) due to the high probability of having faults in a primary unit and in its associated spare. The yield of the 128 × 128 array with the (2,4) scheme was projected to be much higher.

IV. Additional Yield-Enhancement Techniques

A. Layout Modification

The traditional approach to yield enhancement, defect tolerance through redundancy (discussed in Section III), has its disadvantages. It is applicable mainly to highly regular designs, usually requires an increase in the chip area, and involves the development of specialized redundancy techniques for each design. In contrast, the newer layout modification approach discussed next is applicable to all design styles, does not require any additional resources in terms of silicon area, and can be automated and made part of the physical design tools (e.g., compaction, routing) so that it is transparent to the designer.

The layout modification method for yield enhancement consists of making local variations in the layout of some layers in such a way that the critical area, and consequently the sensitivity of the layer to point defects, is reduced. For example, the spacing of some lines can be increased so that the total critical area of that layer decreases. When these changes are made in the interconnect logic, they do not introduce any functional/parametric changes to the circuit, and the RC characteristics remain almost the same. However, when similar changes are made in the active logic, special attention should be paid to maintaining the functional and performance requirements.

The effect of reduction in the critical area on the yield of a chip depends on its size, as shown in Fig. 8. Yields are calculated using the negative binomial model [see (20)] with $Y_0 = 0.95$, $\alpha = 2.0$, and $\lambda = 0.5/\text{cm}^2$. For example, the yield of a 3.0 cm^2 chip can be improved by 14.2% (from 0.310 to 0.354) with a 15% reduction in the critical area.

Layout modifications can be performed at the last stage of the physical design process, i.e., the compaction stage, or at earlier stages like routing. We describe below the different approaches to layout modification for yield improvement, some or all of which can be applied in order to obtain the maximum possible yield.

1) Compaction Strategies for Yield Enhancement: The main purpose of the compaction stage is to perform area minimization whose goal is to increase the number of chips in a wafer. While the primary goal of all compactors is to minimize the area [6], [20], most include some secondary objectives like minimizing the total wire length and minimizing the number of jogs with the goal of performance improvement. Though the importance of yield enhancement has been recognized [6], [61], so far only limited attention has been paid to it in physical design tools.

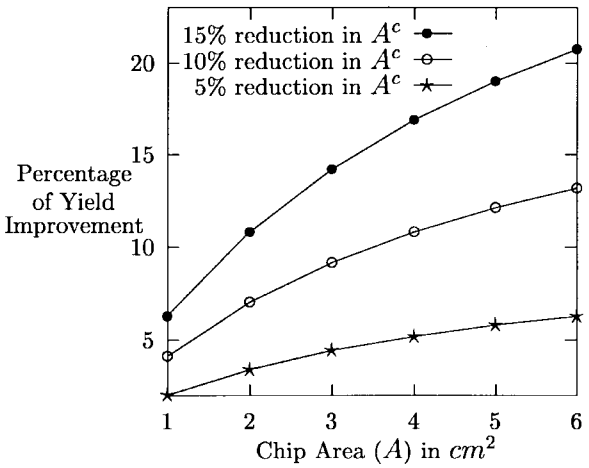


Fig. 8. The effect of critical area reduction on yield improvement.

Compactors generate actual layouts that occupy minimum area either from symbolic layouts or from actual layouts generated by other layout synthesis tools. In constraint graph-based compaction algorithms [57], physical connectivity and separation constraints between the elements are represented by a directed graph. The minimum achievable size of the layout is determined by the longest (critical) path of the constraint graph. The elements on the critical path are placed at the minimum distance allowed by the design rules in order to minimize the area, and thus have no freedom to move. In contrast, elements that do not lie on the critical path can be placed in a variety of ways.

This freedom in placing the noncritical elements has so far been utilized by several compactors only to optimize the performance through wire length minimization, e.g., [20]. Some other compactors place all circuit elements as close as the design rules permit, packing unnecessarily many noncritical elements very close together, resulting in a large critical area for short-circuit defects. Moreover, some compactors stretch various wire segments in order to maintain the original topology, resulting in longer nets with a large critical area for open-circuit defects.

The opportunity for yield improvement provided by the freedom in placing the noncritical elements has been recognized by Allan *et al.* [1], who proposed local modifications such as increasing the contact size, wire segment displacement, and increased wire segment width. A somewhat different approach to layout modifications was presented by Chiluvuri and Koren in [10] and [13]. They proposed a postcompaction algorithm to improve the yield without increasing the layout area by reducing the sensitivity of the layout to both short- and open-circuit type defects. Decreasing the sensitivity to short-circuit type defects is achieved by redistributing the spacing between noncritical elements. The sensitivity to open-circuit defects is minimized by increasing the width of several noncritical elements in the layout. The exact modifications performed during these two steps depend on the given manufacturing conditions, i.e., the densities and the size distributions of the different types of defects in the various layers of the layout.

Since the defect size distribution is inversely proportional to the defect size raised to the pth power [see (1)], changes in the critical area will be nonuniform. Increasing the spacing between two wire segments from 2 to 3 μm will be considerably more beneficial than increasing their spacing from 10 to 11 μm. Note, however, that when changes are made in the layout to minimize the sensitivity of the design to one type of defects, the sensitivity to other defect types may increase. For example, when the width of the metal lines is increased to minimize the sensitivity of the layout to open-circuit defects, its sensitivity to short-circuit defects and pinhole defects might increase. Therefore, critical area of all types of defects should be considered while looking for an optimal location for the noncritical elements.

The location for a noncritical element can be optimized by minimizing the function

$$\lambda(y) = d_{\text{op}} \int_{x_0}^{x_M} A_{\text{op}}^{(c)}(x,y) f_d(x)\, dx + d_{\text{sh}} \int_{x_0}^{x_M} A_{\text{sh}}^{(c)}(x,y) f_d(x)\, dx \qquad (42)$$

where $\lambda(y)$ is the number of faults which can affect the functionality of the element, y represents the width and the location of the element, $A_{\text{sh}}^{(c)}(x,y)$ $(A_{\text{op}}^{(c)}(x,y))$ is the critical area of short-circuit (open-circuit) defects of diameter x, d_{sh} (d_{op}) is the defect density of short-circuit (open-circuit) defects, x_0 and x_M are the minimum and maximum sizes of a defect, respectively, and $f_d(x)$ is the density function of the defect size (see Section I). An important feature of this method is that a layout can be optimized for any given manufacturing conditions, e.g., the ratio $d_{\text{op}}/d_{\text{sh}}$.

The yield-enhancement algorithm presented in [13] has been implemented as an additional feature in an IBM compactor [20]. The results of applying this algorithm to two large circuits have also been reported in [13]. These circuits consist of several thousands of active devices, and two metal layers are used as interconnect layers. Their layouts were first compacted without enabling the yield-optimization feature, and the POF of each interconnect layer for open- and short-circuit defects was measured using *Xlaser* [31]. The layouts were then compacted by enabling the yield-enhancement option, and the area of the layout remained unchanged during the yield-optimization phase. In these circuits, the POF of the metal-1 layer for short-circuit defects was reduced by 8.2%, while the POF for open-circuit defects was slightly increased. Since the short-circuit type of defects had a higher density than the open-circuit ones, reducing the sensitivity of the layout to the first type of defects was more beneficial. Further details of the above algorithm and the resulting yield improvements are included in [13].

In a more recent work, Bamji and Malavasi [3] present a new compaction algorithm that determines the optimal spacing between objects so that the critical area for short-circuit type defects is minimized (open-circuit type defects are not considered). Their method transforms the problem into a network flow problem that is capable of handling general convex objective functions. This allows the minimization of other circuit performance measures (e.g., cross talk) in addition to short-circuit critical area.

2) Routing Strategies for Yield Enhancement: Since compaction is the last stage of the layout synthesis, the effectiveness of the yield enhancement at this stage is highly dependent on the quality of the layout generated by the previous stages. Additional yield improvements can therefore be achieved through strategies for routing, layer assignment, and the like.

Most existing routers try to minimize the number of vias in the layout. The minimum width and spacing requirements for vias are larger than those for wire segments, and thus, more compact designs are usually possible with fewer vias [23]. Sometimes, to avoid a via, routers may introduce very long wire segments, which clearly result in a higher critical area. However, in certain situations, it may be worthwhile to add new vias (or leave some vias intact) to avoid unnecessary additional wiring. For example, for the defect densities reported in [16], the fault probability of one metal-1/polysilicon contact is equivalent to that of a polysilicon wire segment of length 15 μm and width 1.5 μm. Therefore, adding a via that can eliminate more than 15 μm of polysilicon will reduce the critical area.

In an early work [79] on routing for yield improvement, only the adjacency information of horizontal tracks (in channel routing) was considered as a measure for defect sensitivity. The vertical layer (used for the vertical wires connecting the nets' terminals to the horizontal wires) was not considered at all, and as a result, an increase in the overall critical area was found in some of the generated examples by this router. In [56], Kuo presented a new channel routing algorithm for yield improvement using layer reassignment and via shifting. Layer reassignment can lead to shorter vertical wire segments, resulting in reduced total wire length, which, in turn, reduces the critical area. Since the horizontal wire segments always have the same length, moving them has no impact on the total wire length.

In [37], the cost function of a maze router for a sea of gates was modified to take into account the probability of failure for spot defects. This reduced the layout sensitivity to defects by 6.4% on average. In a more recent work [14], the routing in a two-layer channel was modified in order to reduce the wire length as well as the number of vias to achieve higher yield improvement. The modifications include moving nets from one track to another, interchanging nets, and interchanging entire tracks. The results of this algorithm were compared to those obtained by formulating the problem as an integer linear programming problem, illustrating the near optimality of the algorithm. When applied to a set of benchmark examples, the algorithm reduced the total wire length of the vertical layer by 14.6% on average. This reduction in wire length results in a similar reduction in the sensitivity to open- and short-circuit type defects. The number of vias was also reduced significantly (by about 30%), further decreasing the defect

sensitivity of the layout. An algorithm for layer assignment in a two-layer routing, which reduces the critical area due to via defects and open- and short-circuit defects was described in [8]. Yield-enhanced routing was recently presented in [101] for a gridless channel routing, which allows a more flexible positioning of the horizontal wire segments.

3) Topological Layout Design Techniques: For PLA's, yield improvement through layout modification can be achieved even before compaction is performed by modifying the topology of the design. A topological optimization technique for yield enhancement of PLA-based designs was presented in [11] and [13]. There, the topological representation of the PLA is altered so that the critical area is minimized, primarily by minimizing the wire length in one or more layers.

In one example (the *misex1* PLA in [13]), the length of the input polysilicon lines was reduced by permuting the product terms in row positions, achieving a 19% wire length reduction in the polysilicon layer and a resulting reduction of 17% in the critical area of this layer. There was also an incidental reduction in the wire length of the other layers, e.g., metal-1 and diffusion layers, allowing further reduction of the critical area. The overall reduction in the critical area was about 24% in the polysilicon layer and about 11% in metal-1 and diffusion layers. There was no change in the maximum delay of 1.93 ns in this PLA, which was verified using a timing analysis tool.

4) Layout Modifications Versus Redundancy: The most significant advantage of yield enhancement through layout modifications is that no additional area is required. The only additional cost might be some increase in the computational time of the computer-aided design tools [37]. Another important advantage is that a layout can be optimized for any given manufacturing conditions. A comparison between the layout modification technique and the more traditional redundancy technique was performed in [9]. Several designs of adders were modified either by incorporating redundancy or by introducing layout modifications. The regular structure of adders enables a simple implementation of defect tolerance through a redundant bit slice. The conclusion was that for high defect densities, the redundancy technique is better, while for low defect densities, the additional redundancy could not be justified and the layout modification technique proved superior.

Still, the layout modification techniques for yield enhancement should supplement rather than replace the more traditional defect-tolerance techniques. The complexity of future integrated circuits will be too high to achieve the yield targets with either of these two methods alone. The effectiveness of these two approaches is highly dependent on the design structure, complexity, and process defect density. In very regular architectures, most notably memory units, redundancy techniques are expected to have a higher contribution toward yield improvement. As the design becomes less regular, the contribution of the layout techniques is expected to increase.

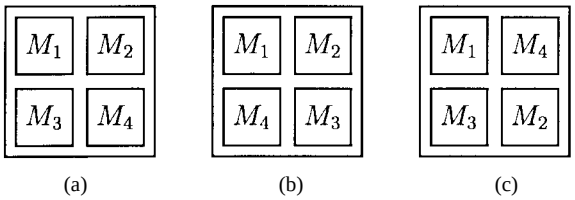


Fig. 9. Three floorplans of a 2 × 2 array.

B. Modifying the Floorplan

Until recently, VLSI designers rarely considered yield issues when selecting a floorplan for a newly designed chip. This is still justified for chips that are small and whose fault distribution can be accurately described by either the Poisson or the compound Poisson yield models with large-area clustering (i.e., the size of the fault clusters is larger than the size of the chip). For those chips, selecting a different floorplan will not affect the projected yield of the designed chip.

This situation is now changing with the introduction of integrated circuits with a total area of 2 cm^2 and up. These chips usually consist of different types of components with different fault densities and have some incorporated redundancy. It has been shown in [52] that if chips with these attributes are hit by medium-sized fault clusters, then changes in the floorplan can affect their projected yield.

Consider the following example, depicted in Fig. 9, of a chip consisting of four equal-area modules (functional units): M_1, M_2, M_3, and M_4. The chip has no incorporated redundancy, and all four modules are necessary for the proper operation of the chip.

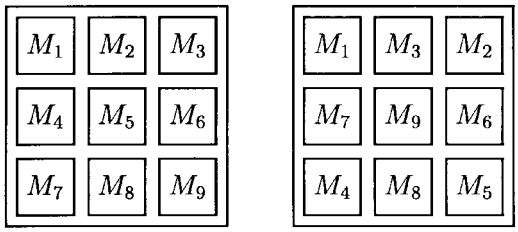
Assuming that the defect clusters are medium sized relatively to the chip size and that the four modules have different sensitivities to defects, we use the medium-area negative binomial distribution (described in Section II) for the spatial distribution of faults, with parameters λ_i (for module M_i) and α (per block), and $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$.

This chip has $4! = 24$ possible floorplans. Since rotation and reflection will not affect the yield, we are left with three distinct floorplans, shown in Fig. 9. If small-area clustering (clusters smaller than or comparable to the size of a module) or large-area clustering (clusters larger than or equal to the chip area) are assumed, the projected yields of all possible floorplans will be the same. This is not the case, however, when medium-area clustering (i.e., horizontal or vertical blocks of two modules) is assumed.

Assuming horizontal defect blocks of size two modules, the yields of floorplans $(a), (b)$, and (c) are

$$\begin{aligned} Y(a) &= Y(b) \\ &= (1+(\lambda_1+\lambda_2)/\alpha)^{-\alpha}(1+(\lambda_3+\lambda_4)/\alpha)^{-\alpha} \\ Y(c) &= (1+(\lambda_1+\lambda_4)/\alpha)^{-\alpha}(1+(\lambda_2+\lambda_3)/\alpha)^{-\alpha}. \end{aligned} \quad (43)$$

A simple algebraic calculation shows that under the condition $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$, floorplans (a) and (b) have the higher yield. Similarly, for vertical defect blocks of size



(a) (b)

Fig. 10. Two floorplans of a 3 × 3 array.

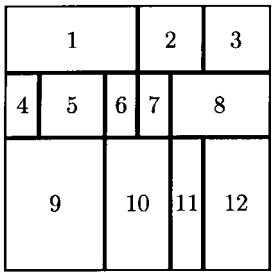
two modules

$$Y(a) = Y(c) = (1+(\lambda_1+\lambda_3)/\alpha)^{-\alpha}(1+(\lambda_2+\lambda_4)/\alpha)^{-\alpha}$$
$$Y(b) = (1+(\lambda_1+\lambda_4)/\alpha)^{-\alpha}(1+(\lambda_2+\lambda_3)/\alpha)^{-\alpha} \quad (44)$$

and floorplans (a) and (c) have the higher yield. Thus, floorplan (a) is the one that maximizes the chip yield for any cluster size. An intuitive explanation to the choice of (a) is that the less sensitive modules are placed together, increasing the chance that the chip will survive a cluster of defects.

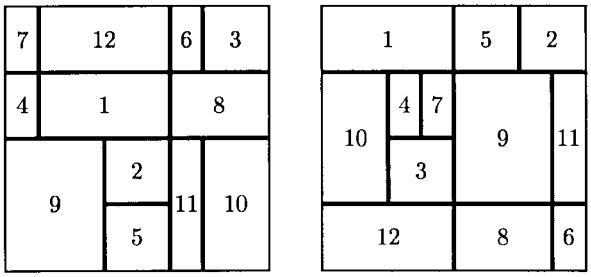
If the previous chip is generalized to a 3 × 3 array (as depicted in Fig. 10), and $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_9$ where λ_i denotes the fault density of module M_i, then, unfortunately, there is no one floorplan that is always the best, and the optimal floorplan depends on the cluster size. However, some generalizations can be made [52]. For all cluster sizes, the module with the highest fault density (i.e., M_9) should be placed in the center of the chip, and each row or column should be rearranged so that its most sensitive module is in its center (such as, for example, floorplan (b) in Fig. 10). Note that we reached this conclusion without assuming that the boundaries of the chip are more prone to defects than its center. The intuitive explanation to this recommendation is that placing highly sensitive modules at the chip corners increases the probability that a single fault cluster will hit two or even four adjacent chips on the wafer. This is less likely to happen if the less sensitive modules are placed in the corners. The above principles are next illustrated through the analysis of the floorplan of Matsushita's ADENART microprocessor [75].

This microprocessor has a 64-b reduced instruction set computer superscalar architecture containing a data cache and an instruction cache. It has been implemented in a 0.8-μm CMOS technology and contains 1300 K transistors in a total area of 14.7 × 15.3 mm^2. A simplified diagram of the chip's floorplan is depicted in Fig. 11(a). The microprocessor includes two register files (floating-point registers and pointer registers), an instruction decode unit (IDU), a data bus control unit, a read-only memory (ROM), and five execution units: a floating-point add and subtract unit, a floating-point multiply and divide unit, a load address add unit, a pointer arithmetic and logic unit, and a flow control unit (FCU). The 12 blocks have six different transistor densities, with the ROM having the highest density and the FCU and IDU the lowest density. Assuming that the fault



Unit	Name	λ
1	*FCU*	λ_{lg_1}
2	*IDU*	λ_{lg_1}
3	*ICU*	λ_{cache}
4	*PR*	λ_{lg_4}
5	*PNU*	λ_{lg_2}
6	*LDU*	λ_{lg_2}
7	*ROM*	λ_{rom}
8	*FMU*	λ_{lg_3}
9	*DCU*	λ_{cache}
10	*DBC*	λ_{lg_5}
11	*FR*	λ_{lg_4}
12	*FAU*	λ_{lg_3}

(a)



(b) (c)

Fig. 11. The original and two alternative floorplans for the ADENART chip.

densities are linearly proportional to the transistor densities, we define six fault densities that satisfy

$$\lambda_{lg\_1} < \lambda_{lg\_2} < \lambda_{lg\_3} < \lambda_{lg\_4} < \lambda_{lg\_5} < \lambda_{\text{cache}} < \lambda_{\text{rom}}.$$

These fault densities are assigned to the individual blocks as shown in Fig. 11(a). Based on the transistor densities reported in [75], the approximate fault densities satisfy

$$\lambda_{\text{rom}} : \lambda_{\text{cache}} : \lambda_{lg\_5} : \lambda_{lg\_4} : \lambda_{lg\_3} : \lambda_{lg\_2} : \lambda_{lg\_1} = 8.88 : 7.69 : 3.27 : 2.42 : 2.27 : 1.69 : 1.$$

The original floorplan of the chip does not follow the guidelines stated above and is therefore not optimal with regard to yield. To demonstrate the effect of a different floorplan on the yield of the microprocessor, we examine two other floorplans. Floorplan (b), shown in Fig. 11(b), in which the modules with the higher fault density are moved to the boundaries and which is expected to have a lower yield than the original, and floorplan (c), shown in Fig. 11(c), which follows the guidelines and is expected to have a higher yield than the original.

Calculating the yield using the medium-area negative binomial distribution results in, as expected

$$Y(b) < Y(a) < Y(c)$$

with $Y(c)$ larger by approximately 9% than $Y(a)$ and $Y(a)$ larger by 5% than $Y(b)$. The improvement in the yield of (c) compared to b is therefore more than 14%.

The next example is that of a chip with redundancy. The chip consists of four modules: M_1, S_1, M_2, and S_2, where S_1 is a spare for M_1 and S_2 is a spare for M_2. The three

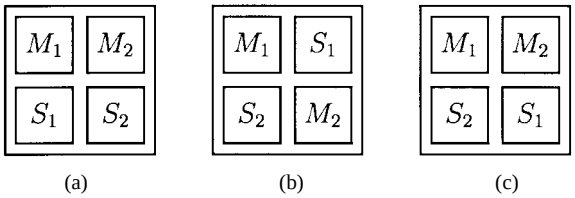


Fig. 12. Three alternative floorplans for a chip with redundancy.

topologically distinct floorplans for this chip are depicted in Fig. 12. Let the number of faults have a medium-area negative binomial distribution with an average of λ_1 for M_1 and S_1, and λ_2 for M_2 and S_2, and a clustering parameter of α per block. Assuming that the defect clusters are horizontal and of size two modules each, the yields of the three floorplans are

$$\begin{aligned} Y(a) = Y(c) &= 2[1+(\lambda_1+\lambda_2)/\alpha]^{-\alpha} \\ &\quad + 2[1+\lambda_1/\alpha]^{-\alpha}[1+\lambda_2/\alpha]^{-\alpha} \\ &\quad - 2[1+(\lambda_1+\lambda_2)/\alpha]^{-\alpha}[1+\lambda_1/\alpha]^{-\alpha} \\ &\quad - 2[1+(\lambda_1+\lambda_2)/\alpha]^{-\alpha}[1+\lambda_2/\alpha]^{-\alpha} \\ &\quad + [1+(\lambda_1+\lambda_2)/\alpha]^{-2\alpha} \qquad (45) \\ Y(b) &= [2(1+\lambda_1/\alpha)^{-\alpha} - (1+2\lambda_1/\alpha)^{-\alpha}] \\ &\quad \times [2(1+\lambda_2/\alpha)^{-\alpha} - (1+2\lambda_2/\alpha)^{-\alpha}]. \qquad (46) \end{aligned}$$

It can be easily proven that for any values of λ_1 and λ_2, $Y(a) = Y(c) \geq Y(b)$.

If, on the other hand, the defect clusters are vertical and of size two modules, then clearly $Y(a)$ is given by (46) and $Y(b) = Y(c)$ is given by (45). In this case, $Y(b) = Y(c) \geq Y(a)$ for all values of λ_1 and λ_2. Floorplan (c) should, therefore, be preferred over (a) and (b). An intuitive justification for the choice of floorplan (c) is that it guarantees the separation between the primary modules and their spares for any size and shape of the defect clusters. This results in a higher yield, since it is less likely that the same cluster will hit both the module and its spare, thus killing the chip.

This last recommendation is exemplified by the design of the 3-D computer, described in Section III-C. The (2,4) structure that has been selected for implementation in the 3-D computer is shown in Fig. 13(a) [108]. This floorplan has every spare unit adjacent to the four primary units that it can replace. This layout has short interconnection links between the spare and any primary unit that it may replace, and as a result, the performance degradation upon a failure of a primary unit is minimal. However, the close proximity of the spare and primary units results in a low yield in the presence of clustered faults, since a single fault cluster may cover several of these units. This phenomenon has been experienced in practice [109].

Several alternative floorplans can be designed that place the spare farther apart from the primary units connected to it (as recommended above). One such floorplan is shown in Fig. 13(b). The yields of the 128 × 128 array using the original floorplan [Fig. 13(a)] or the alternative floorplan [Fig. 13(b)] are shown in Fig. 14. The yield has been calculated using the medium-area negative binomial distribution with a defect block size of two rows of primary units [see Fig. 13(a)]. Fig. 14 clearly shows that the alternative floorplan, in which the spare unit is separated from the primary units that it can replace, has a higher projected yield.

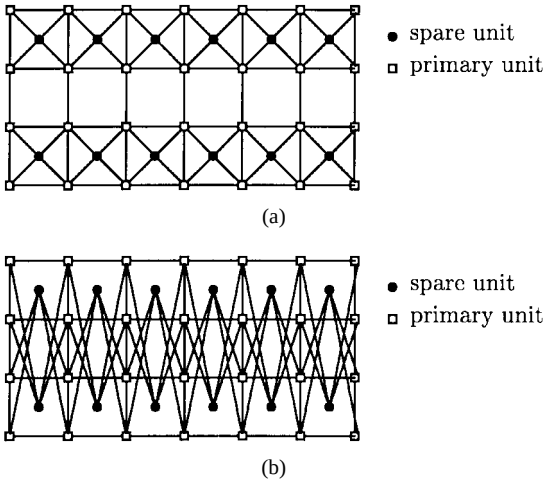


Fig. 13. (a) The original and (b) an alternative floorplan of a wafer in the 3-D computer.

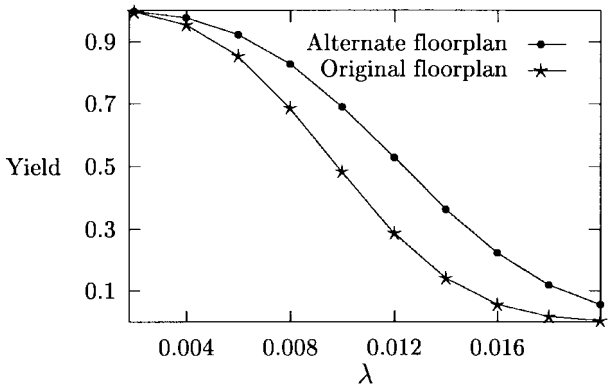


Fig. 14. The yield of the original and alternate floorplans, depicted in Fig. 13, as a function of λ $(\alpha = 2)$.

V. Conclusion

Current VLSI technology allows the manufacture of integrated circuits with several millions of devices. Imperfections in the fabrication process cause logical circuit failures, which reduce the yield of these IC's. The high cost of IC manufacturing justifies the development and use of yield-enhancement techniques at the design stage to complement existing efforts at the manufacturing stage.

We have described various design-stage defect-tolerance techniques aimed at enhancing the yield of VLSI integrated circuits, and illustrated their application in both memory and logic IC's. We have also outlined the analytic yield models that are used in evaluating the effectiveness of these techniques and in selecting some of their parameters.

One of these techniques, namely, the incorporation of redundant circuits, is well established and has been in use

for quite some time, mainly through adding redundant rows and columns to memory IC's. New redundancy schemes for very large memory IC's have been developed and implemented by major semiconductor companies like NEC, Hitachi, IBM, Toshiba, and Samsung, and two of these have been briefly described in this paper. We can expect further development of such techniques in the near future, with some of them migrating to cache memory units in the next generation of microprocessors.

The other yield-enhancing techniques, namely, layout and floorplan modification, have only recently been suggested and, as a result, have only preliminary implementations. They seem to be more suitable to random logic IC's, in which redundancy requires a large overhead.

The even higher device densities and larger chip areas that we are guaranteed to see in the next few years will further increase the need for defect-tolerance techniques. However, since the reliability of very dense circuits operating at extremely high frequencies in submicrometer technology is becoming a major concern, new design methods that combine both yield enhancement and reliability improvement will have to be developed.

ACKNOWLEDGMENT

The writing of this paper was first contemplated in 1993 with the authors' colleague Dr. C. H. Stapper as a coauthor. Unfortunately, before they had the opportunity to start this project, he was involved in a major car accident, from the effects of which he is still fighting to recover. The authors dedicate this paper to Dr. Stapper, their dear friend and colleague.

REFERENCES

[1] G. A. Allan, A. J. Walton, and R. J. Holwill, "A yield improvement technique for IC layout using local design rules," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1355–1362, Nov. 1992.

[2] G. A. Allan and J. A. Walton, "Hierarchical critical area extraction with the EYE tool," in *Proc. 1995 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1995, pp. 28–36.

[3] C. Bamji and E. Malavasi, "Enhanced network flow algorithm for yield optimization," in *Proc. 33rd Design Automation Conf., DAC-96*, June 1996, pp. 746–751.

[4] W. J. Bertram, "Yield and reliability," in *VLSI Technology*, 2nd ed., S. M. Sze, Ed. New York: McGraw-Hill, 1988.

[5] A. Boubekeur, J.-L. Patry, G. Saucier, and J. Trilhe, "Configuring a wafer scale two-dimensional array of single-bit processors," *Computer*, vol. 25, pp. 29–39, Apr. 1992.

[6] D. G. Boyer, "Symbolic layout compaction review," in *Proc. 25th ACM/IEEE Design Automation Conf.*, 1988, pp. 383–389.

[7] I. Chen and A. J. Strojwas, "RYE: A realistic yield simulator for VLSIC structural failures," in *Proc. IEEE Int. Test Conf.*, 1987, pp. 31–42.

[8] Z. Chen and I. Koren, "Layer assignment for yield enhancement," in *Proc. 1995 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1995, pp. 173–180.

[9] ——, "Techniques for yield enhancement of VLSI adders," in *Proc. ASAP'95—Int. Conf. Application-Specific Array Processors*, July 1995, pp. 222–229.

[10] V. K. R. Chiluvuri and I. Koren, "New routing and compaction strategies for yield enhancement," *IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1992, pp. 325–334.

[11] ——, "Topological optimization of PLA's for yield enhancement," in *Proc. 1993 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Oct. 1993, pp. 175–182.

[12] V. K. R. Chiluvuri, I. Koren, and J. L. Burns, "The effect of wire length minimization on yield," in *Proc. 1994 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Oct. 1994, pp. 97–105.

[13] V. K. R. Chiluvuri and I. Koren, "Layout synthesis techniques for yield enhancement," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 178–187, May 1995.

[14] ——, "Wire length and via reduction for yield enhancement," in *Proc. 1996 SPIE Microelectronics Manufacturing Conf.*, Oct. 1996, pp. 103–111.

[15] B. Ciciani, Ed., *Manufacturing Yield Evaluation of VLSI/WSI Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1998.

[16] R. S. Collica, J. Dietrich, R. Lambracht, and D. G. Lau, "A yield enhancement methodology for custom VLSI manufacturing," *Digital Tech. J.*, vol. 4, no. 2, pp. 83–99, Spring 1992.

[17] J. A. Cunningham, "The use and evaluation of yield models in integrated circuit manufacturing," *IEEE Trans. Semiconduct. Manufact.*, vol. 3, no. 2, pp. 60–71, May 1990.

[18] A. R. Dalal, P. D. Franzon, and M. J. Lorenzetti, "A layout-driven yield predictor and fault generator for VLSI," *IEEE Trans. Semiconduct. Manufact.*, vol. 6, no. 1, pp. 77–81, Feb. 1993.

[19] S. W. Director, W. Maly, and A. J. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*. Boston, MA: Kluwer, 1990.

[20] "EDA CircuitBench user's guide," IBM Corp., Yorktown Heights, NY, 1994.

[21] F. Duvivier and M. Rivier, "Approximation of critical areas of IC's with simple parameters extracted from the layout," in *Proc. 1995 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1995, pp. 1–9.

[22] R. B. Fair, "Challenges to manufacturing submicron, ultra-large scale integrated circuits," *Proc. IEEE*, vol. 78, pp. 1687–1705, Nov. 1990.

[23] S. C. Fang, K. E. Chang, and W. S. Feng, "Via minimization with associated constraints in three-layer routing problem," in *Proc. Int. Symp. Circuits and Systems*, 1990, pp. 1632–1635.

[24] A. V. Ferris-Prabhu, "Role of defect size distribution in yield modeling," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 1727–1736, Sept. 1985.

[25] A. V. Ferris-Prabhu, L. D. Smith, H. A. Bonges, and J. K. Paulsen, "Radial yield variations in semiconductor wafers," *IEEE Circuits Devices Mag.*, vol. 3, pp. 42–47, Mar. 1987.

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

[27] V. F. Flack, "Introducing dependency into IC yield models," *Solid State Electron.*, vol. 28, no. 6, pp. 555–559, June 1985.

[28] ——, "Estimating variations in IC yield estimates," *IEEE J. Solid-State Circuits*, vol. SSC-21, pp. 362–365, Apr. 1986.

[29] S. Gandemer, B. C. Tremintin, and J. J. Charlot, "Critical area and critical levels calculation in IC yield modeling," *IEEE Trans. Electron Devices*, vol. 35, pp. 158–166, Feb. 1988.

[30] J. P. Gyvez and C. Di, "IC defect sensitivity for footprint-type spot defects," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 638–658, May 1992.

[31] J. P. Gyvez, *Integrated Circuit Defect-Sensitivity: Theory and Computational Models*. Boston, MA: Kluwer, 1993.

[32] J. P. Gyvez, Ed., *IC Manufacturability: The Art of Process and Design Integration*. Los Alamitos, CA: IEEE Computer Society Press, to be published.

[33] T. P. Haraszti, "A novel associative approach for fault-tolerant MOS RAM," *IEEE J. Solid-State Circuits*, vol. SSC-17, pp. 539–546, June 1982.

[34] J. C. Harden, "Comments on sources of failures and yield improvement for VLSI and restructurable interconnects for RVLSI and WSI," *Proc. IEEE*, vol. 74, pp. 515–516, Mar. 1986.

[35] J. C. Harden and N. R. Strader, "Architectural yield optimization for WSI," *IEEE Trans. Comput.*, vol. 37, pp. 88–110, Jan. 1988.

[36] N. J. Howard, A. M. Tyrell, and N. M. Allinson, "The yield enhancement of field-programmable gate arrays," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 115–123, Mar. 1994.

[37] E. P. Huijbregts, H. Xue, and J. A. G. Jess, "Routing for reliable manufacturing," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 188–194, May 1995.

[38] H. L. Kalter, C. H. Stapper, J. E. Barth, J. Dilorenzo, C. E. Drake, J. A. Fifield, G. A. Kelley, S. C. Lewis, W. B. Van Der

Hoeven, and J. A. Yankosky, "A 50-ns 16 Mb DRAM with 10-ns data rate and on-chip ECC," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1118–1128, Oct. 1990.
[39] M. B. Ketchen, "Point defect yield model for wafer scale integration," *IEEE Circuits Devices Mag.*, vol. 1, no. 4, pp. 24–34, July 1985.
[40] T. Kirihata, Y. Watanabe, H. Wong, and J. K. DeBrosse, "Fault-tolerant designs for 256 Mb DRAM," *IEEE J. Solid-State Circuits*, vol. 31, pp. 558–566, Apr. 1996.
[41] G. Kitsukawa, M. Horiguchi, Y. Kawajiri, and T. Kawahara, "256-Mb DRAM circuit technologies for file applications," *IEEE J. Solid-State Circuits*, vol. 28, pp. 1105–11101, Nov. 1993.
[42] I. Koren and M. A. Breuer, "On Area and yield considerations for fault-tolerant VLSI processor arrays," *IEEE Trans. Comput.*, vol. C-33, pp. 21–27, Jan. 1984.
[43] I. Koren and D. K. Pradhan, "Yield and performance enhancement through redundancy in VLSI and WSI multiprocessor systems," *Proc. IEEE*, vol. 74, pp. 699–711, May 1986.
[44] ____, "Modeling the effect of redundancy on yield and performance of VLSI systems," *IEEE Trans. Comput.*, vol. C-36, pp. 344–355, Mar. 1987.
[45] I. Koren, "The effect of scaling on the yield of VLSI circuits," in *Yield Modeling and Defect Tolerance in VLSI*, W. Moore, W. Maly, and A. Strojwas, Eds. Bristol, UK: Adam Hillger, 1988, pp. 91–99.
[46] I. Koren, Z. Koren, and D. K. Pradhan, "Designing interconnection buses in VLSI and WSI for maximum yield and minimum delay," *IEEE J. Solid-State Circuits*, vol. 23, pp. 859–866, June 1988.
[47] I. Koren and C. H. Stapper, "Yield models for defect tolerant VLSI circuits: A review," *Defect and Fault Tolerance in VLSI Systems*, I. Koren, Ed. New York: Plenum, vol. 1, 1989, pp. 1–21.
[48] I. Koren and A. D. Singh, "Fault tolerance in VLSI circuits," in *Computer*, vol. 23, pp. 73–83, July 1990.
[49] I. Koren, Z. Koren, and C. H. Stapper, "A unified negative binomial distribution for yield analysis of defect tolerant circuits," *IEEE Trans. Comput.*, vol. 42, pp. 724–437, June 1993.
[50] ____, "A statistical study of defect maps of large area VLSI IC's," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 249–256, June 1994.
[51] I. Koren and Z. Koren, "Yield analysis of a novel scheme for defect-tolerant memories," in *Proc. 1996 IEEE Int. Conf. Innovative Systems in Silicon*, Oct. 1996, pp. 269–278.
[52] Z. Koren and I. Koren, "On the effect of floorplanning on the yield of large area integrated circuits," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 3–14, Mar. 1997.
[53] I. Koren and Z. Koren, "Analysis of a hybrid defect-tolerance scheme for high-density memory IC's," in *Proc. 1997 IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, Oct. 1997, pp. 166–174.
[54] M. Kuboschek, H. J. Iden, U. Jagau, and J. Otterstedt, "Implementation of a defect tolerant large area monolithic multiprocessor system," *Int. Conf. Wafer Scale Integration*, 1992, pp. 28–34.
[55] S. Y. Kuo and W. K. Fuchs, "Fault diagnosis and spare allocation for yield enhancement in large reconfigurable PLA's," *IEEE Trans. Comput.*, vol. 41, pp. 221–226, Feb. 1992.
[56] S. Y. Kuo, "YOR: A yield-optimizing routing algorithm by minimizing critical areas and vias," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1303–1311, Sept. 1993.
[57] T. Lengauer, *Combinational Algorithms for Integrated Circuit Layout*. London, England: Wiley, 1990.
[58] S. Levasseur and F. Duvivier, "Application of a yield model merging critical areas and defectivity data to industrial products," in *Proc. 1997 IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, Oct. 1997, pp. 11–19.
[59] R. Leveugle, M. Soueidan, and N. Wehn, "Defect Tolerance in a 16 Bit Microprocessor," in *Defect and Fault Tolerance in VLSI Systems*, I. Koren, Ed. New York: Plenum, 1989, vol. 1, pp. 179–190.
[60] R. Leveugle, Z. Koren, I. Koren, G. Saucier, and N. Wehn, "The HYETI defect tolerant microprocessor: A practical experiment and a cost-effectiveness analysis," *IEEE Trans. Comput.*, vol. 43, pp. 1398–1406, Dec. 1994.
[61] M. Lorenzetti, "The effect of channel router algorithms on chip yield," presented at the MCNC International Workshop on Layout Synthesis, May 1990.
[62] M. Lorenzetti, P. Magill, A. Dalal, and P. Franzon, "McYield: A CAD tool for functional yield projections," in *Proc. IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1990, pp. 100–110.
[63] N. Maldonado, G. Andrus, A. Tyagi, M. Madani, and M. Bayoumi, "A post-processing algorithm for short-circuit defect sensitivity reduction in VLSI layouts," in *Proc. IEEE Int. Conf. Wafer Scale Integration*, San Francisco, CA, 1994, pp. 52–60.
[64] W. Maly, "Computer-aided design for VLSI circuit manufacturability," *Proc. IEEE*, vol. 78, pp. 356–392, Feb. 1990.
[65] ____, "Modeling of lithography related yield losses for CAD of VLSI circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 166–177, July 1985.
[66] W. Maly, W. R. Moore, and A. Strojwas, "Yield loss mechanisms and defect tolerance," in *Yield Modeling and Defect Tolerance in VLSI*, W. R. Moore, W. Maly, and A. Strojwas, Eds. Adam Hillger, 1988, pp. 3–30.
[67] W. Maly, A. J. Strojwas, and S. W. Director, "VLSI yield prediction and estimation: A unified framework," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 114–130, 1986.
[68] T. E. Mangir, "Sources of failures and yield improvement for VLSI and restructurable interconnects for RVLSI and WSI: Part I—Sources of failures and yield improvement for VLSI," *Proc. IEEE*, vol. 72, pp. 690–708, June 1984.
[69] F. Meyer and D. K. Pradhan, "Modeling defect spatial distribution," *IEEE Trans. Comput.*, vol. 38, pp. 538–546, Apr. 1989.
[70] B. T. Murphy, "Cost-size optima of monolithic integrated circuits," *Proc. IEEE*, vol. 52, pp. 1537–1545, Dec. 1964.
[71] T. L. Michalka, R. C. Varshney, and J. D. Meindl, "A discussion of yield modeling with defect clustering, circuit repair, and circuit redundancy," *IEEE Trans. Semiconduct. Manufact.*, vol. 3, pp. 116–127, Aug. 1990.
[72] A. Mirza, G. O'Donoghue, A. Drake, and S. Graves, "Spatial yield modeling for semiconductor wafers," in *Proc. Advanced Semiconductor Manufacturing Conf. and Workshop*, pp. 276–281.
[73] W. R. Moore, "A review of fault-tolerant techniques for the enhancement of integrated circuit yield," *Proc. IEEE*, vol. 74, pp. 684–698, May 1986.
[74] P. K. Nag and W. Maly, "Hierarchical extraction of critical area for shorts and very large IC's," in *Proc. 1995 IEEE Int. Workshop Defect and Fault Tolerance in VLSI Systems*, Nov. 1995, pp. 19–27.
[75] H. Nakano, M. Nakajima, Y. Nakakura, and T. Yoshida, "An 80-MFLOP's (Peak) 64-b microprocessor for parallel computer," *IEEE J. Solid-State Circuits*, vol. 27, pp. 365–371, Mar. 1992.
[76] R. Negrini, M. G. Sami, and R. Stefanelli, *Fault Tolerance Through Reconfiguration in VLSI and WSI Srrays* Cambridge, MA: MIT Press, 1989.
[77] T. Okabe, M. Nagata, and S. Shimada, "Analysis of yield of integrated circuits and a new expression for the yield," *Elec. Eng. Jpn.*, vol. 92, pp. 135–141, Dec. 1972.
[78] O. Paz and T. R. Lawson, Jr., "Modification of Poisson statistics: Modeling defects induced by diffusion," *IEEE J. Solid-State Circuits*, vol. SSC-12, pp. 540–546, Oct. 1977.
[79] A. Pitaksanonkul, S. Thanawastien, C. Lursinsap, and J. A. Gandhi, "DTR: A defect-tolerant routing algorithm," in *Proc. 26th IEEE Design Automation Conf.*, 1989, pp. 795–798.
[80] R. D. Rung, "Determining IC layout rules for cost minimization," *IEEE J. Solid-State Circuits*, vol. SSC-16, pp. 35–42, Feb. 1981.
[81] J. E. Price, "A new look at yield of integrated circuits," *Proc. IEEE*, vol. 58, pp. 1290–1291, Aug. 1970.
[82] S. E. Schuster, "Multiple word/bit redundancy for semiconductor memories," *IEEE J. Solid-State Circuits*, vol. SSC-13, pp. 698–703, Oct. 1978.
[83] P. Schvan, D. Y. Montuno, and R. Hadaway, "Yield projection based on electrical fault distriction and critical structure analysis," in *Defect and Fault Tolerance in VLSI Systems*, I. Koren, Ed. New York: Plenum, vol. 1, pp. 117–127, 1989.
[84] R. B. Seeds, "Yield, economic, and logistic models for complex digital arrays," in *1967 IEEE Int. Conv. Rec.*, pt. 6, pp. 61–66.
[85] ____, "Yield and cost analysis of bipolar LSI," in *Proc. 1967 IEEE Int. Electron Devices Meeting.*, Washington, DC, Oct. 1967, p. 12.

[86] A. D. Singh, "Interstitial redundancy: An area efficient fault tolerance scheme for larger area VLSI processor array," *IEEE Trans. Comput.*, vol. 37, pp. 1398–1410, Nov. 1988.
[87] R. Spence and R. S. Soin, *Tolerance Design of Electronic Circuits*. Reading, MA: Addison-Wesley, 1988.
[88] C. H. Stapper, "Defect density distribution for LSI yield calculations," *IEEE Trans. Electron Devices*, vol. ED-20, pp. 655–657, July 1973.
[89] ——, "On a composite model of the IC yield problem," *IEEE J. Solid-State Circuits*, vol. SSC-10, pp. 537–539, Dec. 1975.
[90] C. H. Stapper, A. N. McLaren, and M. Dreckmann, "Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product," *IBM J. Res. Develop.*, vol. 20, pp. 398–409, 1980.
[91] C. H. Stapper, "Comments on some considerations in the formulation of IC yield statistics," *Solid-State Electron.*, vol. 24, pp. 127–132, Feb. 1981.
[92] C. H. Stapper and R. J. Rosner, "A simple method for modeling VLSI yields," *Solid-State Electron.*, vol. 25, pp. 487–489, June 1982.
[93] ——, "Integrated circuit yield management and yield analysis: Development and implementation," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 95–102, May 1995.
[94] C. H. Stapper, F. M. Armstrong, and K. Saji, "Integrated circuit yield statistics," *Proc. IEEE*, vol. 71, pp. 453–470, Apr. 1983.
[95] C. H. Stapper, "Modeling of defects in integrated circuit photolithographic patterns," *IBM J. Res. Develop*, vol. 28, no. 4, pp. 461–474, July 1984.
[96] ——, "The effects of wafer to wafer density variations on integrated circuit defect and fault distributions," *IBM J. Res. Develop*, vol. 29, pp. 87–97, Jan. 1985.
[97] ——, "On yield, fault distributions and clustering of particles," *IBM J. Res. Develop.*, vol. 30, pp. 326–338, May 1986.
[98] ——, "Small-area fault clusters and fault-tolerance in VLSI circuits," *IBM J. Res. Develop.*, vol. 33, Mar. 1989.
[99] ——, "Block alignment: A method for increasing the yield of memory chips that are partially good," in *Defect and Fault Tolerance in VLSI Systems*, I. Koren, Ed. New York: Plenum, 1989, pp. 243–255.
[100] T. Sugibayashi, I. Naritake, S. Utsugi, K. Shibahara, and R. Oikawa, "A 1-Gb DRAM for file applications," *IEEE J. Solid-State Circuits*, vol. 30, pp. 1277–1280, Nov. 1995.
[101] A. Venkataraman, H. Chen, and I. Koren, "Yield enhanced routing for high-performance VLSI designs," in *Proc. Microelectronics Manufacturing Yield, Reliability and Failure Analysis, SPIE'97*, Oct. 1997, pp. 50–60.
[102] I. A. Wagner and I. Koren, "An interactive VLSI CAD tool for yield estimation," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 130–138, May 1995.
[103] D. M. H. Walker, *Yield Simulation for Integrated Circuits*. Boston, MA: Kluwer, 1987.
[104] N. Wehn, M. Glesner, K. Caesar, P. Mann, and A. Roth, "A defect tolerant and fully testable PLA," in *Proc. 25th Design Automation Conf.*, 1988, pp. 22–27.
[105] C. L. Wey, "On yield considerations for the design of redundant programmable logic arrays," *IEEE Trans. Computer-Aided Design*, vol. CAD-7, pp. 528–535, Apr. 1988.
[106] T. Yamagata, H. Sato, K. Fujita, Y. Nishmura, and K. Anami, "A distributed globally replaceable redundancy scheme for sub-half-micron ULSI memories and beyond," *IEEE J. Solid-State Circuits*, vol. 31, pp. 195–201, Feb. 1996.
[107] J.-H. Yoo, C.-H. Kim, K.-C. Lee, and K.-H. Kyung, "A 32-bank 1 Gb self-strobing synchronous DRAM with 1 GB/s bandwidth," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1635–1643, Nov. 1996.
[108] M. Yung, M. J. Little, R. D. Etchells, and J. G. Nash, "Redundancy for yield enhancement in the 3D computer," in *Proc. Int. Conf. Wafer Scale Integration*, Jan. 1989, pp. 73–82.
[109] M. Yung, private communication, 1995.

Israel Koren (Fellow, IEEE) received the B.Sc., M.Sc., and D.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, in 1967, 1970, and 1975, respectively, all in electrical engineering.

He currently is a Professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. Previously, he was with the Departments of Electrical Engineering and Computer Science at the Technion—Israel Institute of Technology. He also has held visiting positions with the University of California at Berkeley, University of Southern California, Los Angeles, and University of California, Santa Barbara. He has been a Consultant to a number of companies, including IBM, Intel, Analog Devices, AMD, Digital Equipment Corp., National Semiconductor, and Tolerant Systems. His current research interests are fault-tolerance techniques, models for yield and performance, and computer arithmetic. He has published more than 120 publications in refereed journals and conferences. He also has been Program Chair and General Chair for several conferences and Program Committee member for numerous conferences. He has edited and coauthored the book *Defect and Fault-Tolerance in VLSI Systems*, vol. 1 (Plenum, 1989). He is the author of the textbook *Computer Arithmetic Algorithms* (Prentice-Hall, 1993).

Dr. Koren has published extensively in several IEEE TRANSACTIONS. He was a Coguest Editor for the IEEE TRANSACTIONS ON COMPUTERS special issue on "High Yield VLSI Systems," April 1989, and was on the Editorial Board of that TRANSACTIONS from 1992 to 1997.

Zahava Koren received the B.A. and M.A. degrees in mathematics and statistics from The Hebrew University, Jerusalem, Israel, in 1967 and 1969, respectively, and the D.Sc. degree in operations research from the Technion—Israel Institute of Technology, Haifa, in 1976.

She currently is a Senior Research Fellow at the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Previously, she was with the Department of Industrial Engineering at the University of Massachusetts, the Department of Statistics, University of Haifa, the Departments of Industrial Engineering and Computer Science at the Technion—Israel Institute of Technology, and the Department of Business and Economics, California State University in Los Angeles. Her main interests are stochastic analysis of computer networks, yield of integrated circuits, and reliability of computer systems.