PROJECTING THE YIELD OF DEFECT-TOLERANT ICs

Israel Koren

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003

ABSTRACT

With the increased complexity of VLSI circuits, the incorporation of defect tolerance became a viable approach in the attempt to enhance the yield of integrated circuits to acceptable levels. To select a suitable defect tolerance technique and determine the optimal amount of redundancy, schemes for yield projection are necessary. In this paper we survey defect tolerant designs and present commonly used yield expressions.

1. INTRODUCTION

The area of monolithic VLSI chips has always been limited by fabrication defects, which appear impossible to eliminate in even the best manufacturing processes. The larger the circuit, the more likely it is to contain such a defect and fail to operate correctly. Thus, the high probability that spot defects will occur in any fabrication line limits the size of the largest defect-free chip that can be produced with commercially viable yields. Larger circuits must therefore, be designed with a fault tolerance to fabrication defects in order to reduce manufacturing costs.

Defect tolerant designs were first employed in memory chips and recently extended to random logic VLSI circuits and wafer scale circuits. These rely on a variety of techniques for incorporating redundancy in the design, with the simplest scheme being the addition of spare word lines and bit lines in memory chips. As the size of the designed chips increases, the need for more sophisticated techniques becomes more pressing as evidenced by the recent 16 Mbit dynamic RAM developed by IBM [5].

New designs of defect tolerant VLSI chips must rely on sufficiently accurate yield projections. These are necessary in order to decide on the defect tolerance strategy to be followed and determine the exact amount of redundancy to be incorporated. For accurate projection of the yield of future chips there is a need for modeling the distribution of defects on a wafer, understanding the conditions under which physical defects result in structural or parametric faults and finally knowing which faults can be tolerated by a given defect tolerant architecture.

Practical defect tolerant designs have been in the past almost exclusively limited to memory chips. Recently, defect tolerant designs for random logic circuits like microprocessors [13] have been proposed and successfully implemented. Even in the design of defect tolerant memory chips, new techniques have been developed to allow the fabrication of 16Mbit (and beyond) DRAM chips.

The objective of this paper is to provide a survey of the techniques used to design defect tolerant VLSI chips, and the methods employed to evaluate these designs and project their yield. We first describe in Section 2 the defects that can occur when manufacturing VLSI integrated circuits and the faults that may result. We then describe in Section 3 several defect-tolerant designs of memory ICs, logic ICs and wafer-scale circuits. Section 4 introduces yield models for chips with redundancy. Final conclusions are presented in Section 5.

2. MANUFACTURING DEFECTS

Manufacturing defects can be classified as 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, over- and under-etching, etc. Spot defects are random local defects from materials used in the process and environmental causes. These are mostly from undesired chemical and airborne particles deposited during the various steps of the process.

The above two classes of defects contribute to yield losses. In mature, well-controlled fabrication lines, gross area defects can be minimized. The yield loss due to random 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. Consequently, spot defects

are of greater concern.

Spot defects may cause missing patterns (open circuits) or extra patterns (short circuits). These defects can be further classified into intra-layer defects and inter-layer defects. Intra-layer defects occur as a result of particles deposited during the lithographic processes and are therefore, also known as photolithographic defects. Examples of these are: missing metal (or diffusion or poly-Si) and extra metal (or diffusion or poly-Si). Inter-layer defects include missing vias between two metal layers or between a metal layer and poly-Si, and shorts between the substrate and metal (or diffusion or poly-Si) or between two separate metal layers. These inter-layer defects occur as a result of local contamination, e.g., dust particles.

The above spot defects are structural defects that result in discrete faults such as line breaks and short-circuits. A defect causes a discrete fault only if it is large enough to connect two disjoint conductors or disconnect a continuous pattern. Thus, the probability that a defect will cause a structural fault depends on the exact geometrical position of the defect and on its size. We next briefly outline the method used to determine the percentage of manufacturing defects that result in discrete faults. This type of calculation is necessary to determine the expected number of circuit faults on the basis of which yield projection, as discussed in Section 4, is done.

To express the average number of manufacturing defects of type i (for example, photolithographic defects of the open-circuit type) the term $d_i A$ is commonly used where d_i denotes the average number of defects (of type i) per unit of area and A is the chip area. To calculate the average number of circuit faults, denoted by λ_i, a probability θ_i that a defect of type i will result in a discrete type circuit fault is defined and thus, $\lambda_i = \theta_i d_i A$. The product $A\theta_i$ is also called the critical area for defects of type i and is denoted by $A_c^{(i)}$.

The probability θ_i may be constant for one type of defect or may depend on the size of the defects (relative to the physical dimensions of VLSI patterns). For example, the size of a inter-layer defect is in most cases relatively small and the probability that it will cause a circuit failure is the ratio between the area of the overlapping region and the total area. In contrast, photolithographic (intra-layer) defects have a randomly distributed size comparable to that of VLSI patterns. Therefore, the probability that such a defect will cause a failure depends on the pattern shape, its dimensions relative to the size of the defect and its exact geometrical position.

The critical area $A_c^{(i)}$ has been calculated for different geometrical patterns (e.g., [8]). In practice however, VLSI layouts consist of many patterns in different shapes, sizes and orientations. Consequently, the exact expression for the critical area of one layout will be different from that of another layout, making it very time consuming to calculate the critical area of all but very simple and regular layouts. Therefore, two other techniques have been proposed: Monte Carlo simulation [21] and virtual artwork [14]. In the Monte Carlo approach, circles representing defects are placed at random locations of the layout and the critical area is estimated. In the virtual artwork approach an artificial layout is extracted from the given layout such that the estimation of the critical area is simplified.

3. DEFECT-TOLERANT DESIGNS

The first integrated circuits to exploit fault-tolerant techniques were memory ICs which are particularly dense and therefore extremely vulnerable to manufacturing defects. Moreover, the demand for even higher densities is continuously increasing since denser memory chips reduce system integration costs, volume and power dissipation. In addition, the high regularity of memory arrays greatly simplifies the task of incorporating defect-tolerance into their design.

Many fault-tolerant techniques have been proposed and successfully implemented in memory ICs. We first review several such techniques and then present some recent proposals for defect-tolerant designs of logic ICs and their extensions to wafer scale integration. Most methods for incorporating fault-tolerance (i.e., redundancy) into VLSI integrated circuits have the following objectives in addition to their main goal of yield enhancement:

- Small additional area and power requirements.
- No or very limited impact of the added redundancy on performance.
- Transparency to the user.
- Equal or higher reliability.
- Fault-free ICs should require no (or limited) additional manufacturing steps.

Yield enhancement of integrated circuits is especially important for new designs and manufacturing processes. For these, the density of process-induced defects is high and the resulting yield is very low. Yield improvements of early prototypes of an IC can reduce the product introduction time and determine its commercial success. Defect tolerance has proved extremely successful in such cases and spectacular 30-fold increases in yield have been reported [17]. Yield improvements due to defect tolerance tend to decrease as the manufacturing process matures, but even for mature processes with lower defect densities a 1.5-to-3-fold yield increases have been experienced, proving the effectiveness of defect tolerance techniques.

3.1 Memory ICs

The incorporation of defect-tolerant techniques in memory ICs for yield enhancement started as early as 1979 with 64Kbit memories and continues now with 16Mbit RAMs and beyond. The first approach used in memory ICs, and still the most common, is the addition of spare rows and/or spare columns (also known as word line and bit lines, respectively). The high regularity of memory arrays allows the use of a limited number of spare rows and columns (i.e., a very low redundancy overhead) for a large number of repetitive circuits. A defective row or a row containing one or more defective memory cells can be disconnected and then replaced by a spare row. Each spare row has a dedicated programmable decoder allowing it to replace any defective row. Similarly, spare columns can replace defective ones. The number of spare rows and/or columns is determined so as to optimize the yield, after taking into account the additional area required for the redundant circuitry and the probability of defects occurring in these circuits.

The chips, after manufacturing, are tested to determine the location of defects and then they are reconfigured by disabling the defective rows and/or columns and programming the decoders of some spare rows and columns to replace the defective ones. All these steps are performed in a fully automatic manner and require no manual intervention [17].

Associative Scheme

The commonly used scheme employing spare rows and columns is restricted to the replacement of individual faulty rows or columns. If replacement of larger blocks of cells is needed, as might be the case when defects are clustered rather than uniformly distributed, an associative approach as developed by Harassti at Hughes Aircraft [4] may prove attractive. The address of the defective block is stored in an associative memory and any incoming request to an address within the defective block will be redirected to a spare block. The spare block has a smaller size compared to the main memory array and consequently, its access time is substantially smaller. Thus, the overall access time increase is less than 2% [4]. The increase in power consumption is insignificant (less than 0.6%) but the area increase is substantially higher than for the spare row/column scheme ranging from 10% for 64 Kbit to 27% for 1Mbit.

Error Correcting Codes

Large memory systems frequently use error correcting codes (ECCs) to mask intermittent faults. Employing such codes to overcome manufacturing defects for yield enhancement can therefore, contribute to reliability improvement as well. However, the associated area overhead is much higher than the simple spare row/column scheme. One of the earliest examples of a memory IC employing an ECC for yield enhancement is Mostek's 1Mbit ROM in which seven parity bits were added to the 64 data bits resulting in more than 11% increase in area. The 71 memory cells selected simultaneously are positioned within the array so that any two selected cells are separated by 15 unselected cells. This allows to tolerate not only single cell failures but also clusters of multiple cell failures. The use of ECCs may slow down the memory since the error detection circuitry lies in the critical path. This circuit was therefore designed to minimize the increase in access time.

More recently, a 16 Mbit dynamic RAM was developed by IBM adding 9 check bits to every 128 data bits. This chip combines the use of ECC with the more traditional bit and word line redundancy and achieves higher yield enhancement [5].

Partially Good Chips

A somewhat different approach to yield enhancement relies on the use of partially good chips. If sections in a 1Mbit memory for example, are defective beyond repair, the chip can be reconfigured to a usable 0.5Mbit chip or even a 0.25Mbit chip. The circuitry of the chip has to be partitioned in such a way that fault-free sections can function indepen-

dently. This technique has been successfully used by several manufacturers of memory ICs like Motorola, IBM and Westinghouse. Note that this technique is orthogonal to other defect tolerance schemes and the individual sections within the chip may have, for example, spare rows and columns. Only when the available redundancy within a section is insufficient to overcome all the defects present in this section, the section will be declared unusable. The idea of using partially good chips has been refined by IBM. The independent sections in the memory chip are further divided into smaller blocks. These can be used separately after proper alignment by steering the data bits to their right positions [20].

3.2 Logic ICs

Efficient defect-tolerant designs for random logic ICs like microprocessors, are considerably more difficult to develop than for memory ICs. However, if some regularity in the structure of a given logic circuit exists, incorporation of redundancy may be possible. A natural target for defect-tolerant designs are programmable logic arrays (PLAs), which have a regular structure and are employed for implementing random logic circuits in VLSI chips. Large PLAs with up to 50 inputs and 190 product terms are commonly used in the control section of microprocessors. Since these PLAs require large silicon area, the incorporation of redundancy in their design can improve the overall yield considerably. Defect-tolerant designs of PLAs have been investigated [22] and the addition of spare programmable product lines, input lines and output lines was proposed to protect against all types of possible defects. This technique is similar to the redundant row/column scheme for memory ICs. However, unlike memory ICs where all defects can be identified by applying test patterns externally, the identification of defects in a PLA requires some built-in testing aids like adding inputs to the AND plane.

PLAs with spare programmable product lines and added inputs for defect identification have been recently implemented within a 16 bit microprocessor [13]. This microprocessor includes also a defect-tolerant data path. The data path of a microprocessor includes arithmetic and logic units, registers and busses and usually consumes a large percentage of the overall area. A bit-sliced data path, with the inclusion of one or more spare slices, can exploit the regularity in the circuit. However, not all parts of the data path are regular. For example, the logic circuits associated with the status bits are highly irregular. Such parts cannot be replaced by common spare circuits and have to be excluded from the bit slice organization.

If the structure of a logic IC is irregular, duplication or even triplication of certain circuits may prove beneficial. If duplication is used, fault identification and then restructuring must be employed after manufacturing. In the case of triplication, these additional steps can be avoided by using a majority voter at the output, if only one defective circuit out of the identical ones is allowed to fail. Replication was employed for defect tolerance in Trilogy's attempt to build a mainframe based on wafer scale technology. However, the extremely large overhead (2-fold and up) associated with these techniques has substantially limited their use in general.

3.3 Wafer Scale Integration

Wafer scale designs may never have a non-zero yield without incorporating some defect-tolerance techniques. Using defect tolerance techniques similar to those outlined above, a number of experimental wafer-scale systems have been implemented. Digital signal processing systems fabricated through the Restructurable VLSI technology developed at Lincoln Lab. have demonstrated the practicality of wafers with a few heterogeneous cell types. The European ESPRIT project is also aggressively pursuing WSI, including memory, microprocessor and array processor designs. The ELSA (European Large SIMD Array) 2D array processor [15] employs a two-level hierarchical defect tolerance approach. The 3D Computer being developed by Hughes Research Laboratory from wafer scale circuits [1] is designed for very high performance and employs the Interstitial Redundancy scheme [16] to ensure that restructured interconnections are short. A 32×32 processor prototype rated at 600 MOPS has already been demonstrated and a 128×128 design is under development.

4. YIELD PROJECTIONS

To select a defect tolerance technique for a VLSI circuit one must project its yield. This allows the designer to determine the optimal amount of redundancy and the suitability of a proposed defect tolerance strategy.

Models for estimating the yield of fault-tolerant integrated circuit chips are complex mainly due to

the clustering of manufacturing defects during chip fabrication. Yield modeling is relatively simple when Poisson statistics are used for describing the distribution of the number of faults per chip. According to this distribution the probability of having exactly x faults in a chip is given by

$$Prob\{X = x\} = \frac{e^{-\lambda}\ \lambda^x}{x!} \qquad (1)$$

where X is a random variable denoting the number of faults and λ denotes the average number of faults expected per chip. For chips with no redundancy the yield is

$$Y = Prob\{X = 0\} = e^{-\lambda}. \qquad (2)$$

As was shown in Section 2, the average number of faults per chip is given by

$$\lambda = \sum_i d_i A_c^{(i)} \qquad (3)$$

where d_i is the density of type i defects and $A_c^{(i)}$ is the critical chip area for type i defects.

Since the early days of integrated-circuit manufacturing it has been known that the above yield formula is too pessimistic and leads to predicted chip yields that are lower than actual yields. It later became clear that the very low predicted yield was the result of ignoring the clustering of faults, a phenomenon observed in practice.

Several modifications to the above yield formula to account for fault clustering have been proposed. The most commonly used modification assumes the number of faults to be Poisson distributed, but considers the parameter λ to be a random variable rather than a constant. Making λ a random variable results in clustering of faults, no matter what type of distribution is assumed for λ.

Averaging yield formula (2) with respect to a probability density function of λ, denoted by $f(\lambda)$, results in the following modified yield formula

$$Y = \int_0^\infty e^{-\lambda} f(\lambda)\ d\lambda\ . \qquad (4)$$

The function $f(\lambda)$ is known as a *compounding function*. Several compounding functions have been proposed in the past leading to different yield formulae. A common one, the Gamma distribution [19], results in the well-known yield formula

$$Y = (1 + \overline{\lambda}/\alpha)^{-\alpha} \qquad (5)$$

where α is called the clustering parameter and $\overline{\lambda}$ is the average number of faults per chip. $\overline{\lambda}$ is, in effect, the expected value of λ. When the clustering parameter α is large, i.e., when $\alpha \to \infty$, the yield in expression (5) becomes equal to yield formula (2). This represents the case of random faults and absence of clustering. Smaller values of α indicate increased clustering. Experimentally derived values for α typically range between 0.3 and 5.

If the same compounding procedure is applied to the Poisson probability function for the number of faults in (1) the negative binomial distribution is obtained:

$$Prob\{X = x\} = \frac{\Gamma(\alpha + x)}{x!\ \Gamma(\alpha)}\ \frac{(\overline{\lambda}/\alpha)^x}{(1 + \overline{\lambda}/\alpha)^{\alpha+x}}\ . \qquad (6)$$

Formula (5) accounts only for faults which are the result of spot defects. To account for gross area defects affecting large wafer areas, a gross yield factor Y_0 must be included leading to

$$Y = Y_0(1 + \overline{\lambda}/\alpha)^{-\alpha}\ . \qquad (7)$$

4.1 Yield of Chips with Redundancy

Frequently, integrated circuit designs contain several circuit modules which are replicated. Chips containing a number of identical modules (of one type or more) can often be used even if some of the modules do not function correctly, obtaining this way partially good chips. Alternatively, one can add a few redundant modules to the design and accept only those chips which have the necessary number of fault-free modules.

For example, consider chips with a single type of identical modules and let N denote the number of these modules. Define the following probability $a_{M,N} = Prob\{$Exactly M out of the N modules are fault–free$\}$. This probability can be used to calculate the yield of chips with redundancy. For example, if R out of N modules are spares meaning that a chip with at least $(N - R)$ fault-free modules is acceptable, then the yield of the chip is given by

$$Y = Y_0 \sum_{M=N-R}^{N} a_{M,N}\ . \qquad (8)$$

In order to derive an expression for $a_{M,N}$ we need to know how to calculate the yield of a subset of M modules. To this end we have to make some assumptions regarding the change in the parameters $\overline{\lambda}$ and

α of the yield formula when partial areas are considered. The average number of faults depends linearly on the number of modules, M. The dependence of the clustering parameter α on M is however, less straightforward. Most studies on integrated-circuit yield assume that the parameter α is the same when the whole chip is considered or when only part of the chip is considered. This assumption is based on the so-called "large area clustering" meaning that the clusters of defects are larger than the chip size. This assumption often proved reasonable since most clustering is caused by wafer-to-wafer variations of fault densities; this is especially true for small area chips. Only recently a modified yield model allowing changes in the clustering parameter has been proposed and preliminary analysis has shown that in certain circumstances it can provide more accurate yield projection [12].

Under the "large area clustering" assumption $a_{M,N}$ can be calculated by first computing the probability that a given number of faults occur in the complete chip and then distributing these faults uniformly among the N modules. Thus, the probability that exactly $(N-M)$ modules will contain faults is

$$a_{M,N} = \sum_{x=N-M}^{\infty} Q_{x,(N-M)}^{(N)} \cdot Prob\{X_N = x\} \quad (9)$$

where $Prob\{X_N = x\}$ is the probability that the chip has x faults and $Q_{x,j}^{(N)}$ is the probability that the x faults are distributed into exactly j out of N modules given that there are x faults. Assuming that faults are distinguishable, the latter equals [7]

$$Q_{x,j}^{(N)} = \sum_{k=0}^{j} (-1)^k \binom{N}{k} \binom{N-k}{N-j} \left[\frac{j-k}{N}\right]^x \quad (10)$$

for $x \geq j$ and $0 < j \leq N$. Substituting the negative binomial distribution from (6) in the above equation yields

$$a_{M,N} = \sum_{k=0}^{N-M} (-1)^k \binom{N-M}{k} \binom{N}{M} \left[1+\frac{(M+k)\overline{\lambda}}{\alpha}\right]^{-\alpha} \quad (11)$$

As we have seen above, the negative binomial distribution is obtained from the Poisson distribution by averaging over all values of λ, using the Gamma distribution function. This compounding procedure can be applied to any statistical measure. We can derive an expression for the desired measure assuming the convenient Poisson distribution (whose most useful property is the statistical independence between faults in different modules), and then apply the compounding procedure to obtain the required expression for the negative binomial model. This powerful compounding procedure was employed to derive yield expressions for interconnection buses in VLSI chips [9] and for partially good memory chips [18].

One should realize that the architecture analyzed above is an idealization; actual chips rarely consist entirely of identical circuit modules. In all chips there are support circuits (like power supply lines, clocks, input and output buffers etc) which are shared by the replicated modules. These chips become unusable when the support circuits are damaged. Since the clustering of the support circuit faults is not independent of the clustering of the module faults, we need to include in formula (11) the average number of faults that cause these support circuits to be defective. This results in

$$a_{M,N} = \sum_{k=0}^{N-M} (-1)^k \binom{N-M}{k} \binom{N}{M} \cdot \left[1+\frac{\overline{\lambda}_{CK}+(M+k)\overline{\lambda}}{\alpha}\right]^{-\alpha} \quad (12)$$

where $\overline{\lambda}_{CK}$ is the average number of fatal faults or *chip-kill* faults in the support circuits.

4.2 Chips with Multiple Types of Modules

The previous discussion was restricted to the case where redundancy is provided to tolerate faults in a single type of circuit modules. In this section we extend the previous results to fault tolerant chips with multiple types of modules. We derive yield expressions for chips with two different types of modules, say, Type 1 and Type 2. The extension to a larger number of module types is straightforward.

Assume that there are redundant modules of both types and that the modules of each type can be reconfigured separately when necessary. Then, we can calculate the yield of each module type separately (assuming that faults follow the Poisson distribution), and multiply the two results to obtain the overall yield

$$Y = Y_1 \cdot Y_2 \quad (13)$$

where Y_i $(i = 1, 2)$ is the yield of the set of N_i

modules of type i. This yield can be calculated using

$$Y_i = \sum_{M_i=N_i-R_i}^{N_i} a_{M_i,N_i} \qquad (14)$$

The final expression for the yield Y will consist of $R_1 \cdot R_2$ terms. However, in many practical chip architectures there is no such *architectural independence* and a fault in a module of type 1 may affect the usefulness of type 2 modules, i.e., a type 2 module may become useless when a type 1 module is defective. In such a case not all $R_1 \cdot R_2$ possible terms should be included in the expression for Y. In well-structured architectures we can easily identify those terms that should be included and we can therefore define a "coverage factor" as follows, $c_{M_1,M_2} = 1$ if the chip is acceptable with M_1 and M_2 fault-free modules of type 1 and 2, respectively. Otherwise, $c_{M_1,M_2} = 0$. Consequently,

$$Y = \sum_{M_1=N_1-R_1}^{N_1} \sum_{M_2=N_2-R_2}^{N_2} a_{M_1,N_1} \cdot a_{M_2,N_2} \cdot c_{M_1,M_2} \qquad (15)$$

c_{M_1,M_2} serves to select all the *fixable* combinations out of all combinations of fault-free modules of type 1 and type 2.

In less structured architectures, the number of fault-free modules of either type may be insufficient to determine whether the chip is fixable or not; we may also need to know the exact position of the fault-free modules. In such a case, c_{M_1,M_2} will not be a factor assuming only the values 0 and 1, but the fraction of fixable patterns out of all patterns consisting of M_1 and M_2 fault-free modules of type 1 and 2, respectively.

Finally, the expression for the yield of a chip with two types of modules and support circuitry when the Poisson distribution is assumed, is as follows

$$Y = Y_0 \sum_{\substack{M_1= \\ N_1-R_1}}^{N_1} \sum_{\substack{M_2= \\ N_2-R_2}}^{N_2} \sum_{k_1=0}^{N_1-M_1} \sum_{k_2=0}^{N_2-M_2} (-1)^{k_1+k_2}$$

$$\binom{N_1}{M_1}\binom{N_1-M_1}{k_1}\cdot\binom{N_2}{M_2}\binom{N_2-M_2}{k_2}$$

$$\cdot e^{-(M_1+k_1)\lambda_1-(M_2+k_2)\lambda_2-\lambda_{CK}} \cdot c_{M_1,M_2} \qquad (16)$$

We now apply the compounding procedure and obtain [10]

$$Y = Y_0 \sum_{\substack{M_1= \\ N_1-R_1}}^{N_1} \sum_{\substack{M_2= \\ N_2-R_2}}^{N_2} \sum_{k_1=0}^{N_1-M_1} \sum_{k_2=0}^{N_2-M_2} (-1)^{k_1+k_2}$$

$$\cdot \binom{N_1}{M_1}\binom{N_1-M_1}{k_1}\binom{N_2}{M_2}\binom{N_2-M_2}{k_2}\cdot c_{M_1,M_2}$$

$$\cdot \left[1+\frac{\left((M_1+k_1)\bar{\lambda}_1+(M_2+K_2)\bar{\lambda}_2+\bar{\lambda}_{CK}\right)}{\alpha}\right]^{-\alpha} \qquad (17)$$

The yield expressions in (8) and (17) can help find the optimal amount of redundancy for a given fault-tolerant scheme. The optimal redundancy is that which maximizes the number of acceptable chips per wafer. When the redundancy increases, the yield of the individual chip tends to increase but the number of chips per wafer tends to decrease. We need therefore, to maximize the so-called *effective yield* which is the chip yield multiplied by the ratio between the number of chips with and without redundancy on the same size wafer.

Clustering affects the projected yield and the resulting optimal amount of redundancy as has been demonstrated in [7] and [11]. Ignoring the clustering phenomenon for the sake of simplifying the task of yield projection may lead to incorrect decisions regarding the amount of redundancy to be incorporated.

5. CONCLUSION

The incorporation of defect-tolerance in VLSI circuits have made remarkable progress in recent years. The theoretical approaches for introducing and optimising redundancy, as well as the supporting technologies, appear to be in place for more wide spread use of redundancy for yield enhancement. Meanwhile, as VLSI feature sizes approach physical limits, the need for larger area chips to meet the ever growing demand for more complex monolithic systems, is likely to become much more pressing. The use of defect tolerance techniques to provide viable yields for these large chips can be expected to become routine in such an environment.

Defect tolerance techniques are a must for wafer scale designs. One such design is the 40 Mbyte Wafer Stack module consisting of two 6 inch-diameter wafers that was recently developed by Anamartic [2]. This and future circuits rely on the already well-established theory of fault tolerance in VLSI circuits and the well-developed supporting technologies.

Acknowledgment

This work was supported in part by the National Science Foundation under contract MIP-8805586.

REFERENCES

[1] M. Campbell, M. Little and M. Yung, "Hierarchical Fault Tolerance for 3D Microelectronics," *Proc. IEEE Conf. on Wafer Scale Integration*, Jan. 1990.

[2] L. Curran, "Wafer Scale Integration Arrives in "Disk" Form," *Electronic Design*, pp. 51-54, October 1989.

[3] A.V. Ferris-Prabhu, "Defect Size Variations and Their Effect on the Critical Area of VLSI Devices," *IEEE Journal of Solid-state Circuits*, Vol. sc-20, pp. 878-880, 1985.

[4] T.P. Haraszti, "A Novel Associative Approach for Fault-Tolerant MOS RAM," *IEEE J. of Solid-State Circuits*, pp. 539-546, June 1982.

[5] H.L. Kalter *et al.* "An Experimental 50-nsec 16Mbit DRAM with 10-nsec Data Rate," *1990 IEEE International Solid-State Circuit Conf.*, paper FPM 14.1, Feb. 1990.

[6] 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.

[7] I. Koren and D.K. Pradhan, "Modeling the Effect of Redundancy on Yield and Performance of VLSI Systems," *IEEE Trans. on Computers*, Vol. C-36, pp. 344-355, March 1987.

[8] I. Koren, "The Effect of Scaling on the Yield of VLSI Circuits," *Yield Modelling and Defect Tolerance in VLSI*, W.R. Moore, W. Maly and A. Strojwas (Eds.), pp. 91-99, Adam Hillger Ltd., 1988.

[9] I. Koren, Z. Koren and D.K. Pradhan, "Designing Interconnection Buses in VLSI and WSI for Maximum Yield and Minimum Delay," *IEEE J. of Solid-state Circuits*, pp. 859-866, June 1988.

[10] I. Koren and C.H. Stapper, "Yield Models for Defect Tolerant VLSI Circuits: A Review," *Defect and Fault Tolerance in VLSI Systems*, Vol. 1, I. Koren (ed.), pp. 1-21, Plenum, 1989.

[11] I. Koren and A.D. Singh, "Fault Tolerance in VLSI Circuits," *Computer*, Special Issue on Fault-Tolerant Systems, Vol. 23, pp. 73-83, July 1990.

[12] I. Koren, Z. Koren and C.H. Stapper, "Employing the Unified Negative Binomial Distribution for Yield Analysis of Empirical Data," *Proc. of the 1990 IEEE Internl. Workshop on Defect and Fault Tolerance in VLSI Systems*, Grenoble, France, November 1990.

[13] R. Leveugle, M. Soueidan, and N. Wehn, "Defect Tolerance in a 16 Bit Microprocessor," *Defect and Fault Tolerance in VLSI Systems*, Vol. 1, I. Koren (ed.), pp. 179-190, Plenum, 1989.

[14] W. Maly, W.R. Moore and A. Strojwas, "Yield Loss Mechanisms and Defect Tolerance," *Yield Modelling and Defect Tolerance in VLSI*, W.R. Moore, W. Maly and A. Strojwas (Eds.), pp. 3-30, Adam Hillger Ltd., 1988.

[15] J-L. Patry and G. Saucier, "Practical Experiences on the Design of a WSI 2-D Array," *Defect and Fault Tolerance in VLSI Systems*, Vol. 2, C.H. Stapper, V.K. Jain and G. Saucier (eds.), Plenum 1990.

[16] A.D. Singh, "Interstitial Redundancy: An Area Efficient Fault Tolerance Scheme for Larger Area VLSI Processor Array," *IEEE Trans. on Computers*, Vol. 37, No. 11, November 1988, pp. 1398-1410.

[17] R.T. Smith *et al.* "Laser Programmable Redundancy and Yield Improvement in a 64K DRAM," *IEEE J. of Solid-State Circuits*, pp. 506-513, Oct. 1981.

[18] C.H. Stapper, A. 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.

[19] C.H. Stapper, F.M. Armstrong and K. Saji, "Integrated Circuit Yield Statistics," *Proc. IEEE*, Vol. 71, pp. 453-470, April 1983.

[20] C.H. Stapper, "Block Alignment: A Method for Increasing the Yield of Memory Chips that are Partially Good," *Defect and Fault Tolerance in VLSI Systems*, I. Koren (ed.), pp. 243-255, New York: Plenum, 1989.

[21] D.M.H. Walker, *Yield Simulation for Integrated Circuits*, Boston: Kluwer Academic Publishers, 1987.

[22] C.L. Wey, "On Yield Considerations for the Design of Redundant Programmable Logic Arrays," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-7, pp. 528-535, April 1988.