Analysis of a Hybrid Defect-Tolerance Scheme for High-Density Memory ICs *

Israel Koren and Zahava Koren

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003
E-mail: koren@euler.ecs.umass.edu

Abstract

Recent increases in the density and size of memory ICs made it necessary to search for new defect tolerance techniques since the traditional methods are no longer effective enough. Several new such schemes have been recently proposed and implemented. Due to the high complexity of these new techniques compared to the simple row and column redundancy, Monte-Carlo simulations were used to evaluate their yield enhancement. In this paper we present a yield analysis of one such new design and compare its yield to that of the traditional design.

1. Introduction

The traditional method for incorporating defect tolerance in memory ICs through redundant rows and columns has been extremely successful for more than 15 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 marketplace. The effectiveness of these traditional methods is going down due to mainly two reasons.

The first is that the large size of the memory array makes it necessary to partition it into several sub-arrays in order to decrease the current and reduce the access time by shortening the length of the bit and word lines [7]. Using the conventional redundancy methods implies that each sub-array has its own redundant rows and columns, leading to situations where one sub-array has an insufficient number of spare lines to handle local defects while others still have several unused redundant lines.

The second reason is that the higher density of the new sub-micron memory ICs drastically increases the yield loss due to chip-kill defects, e.g., defects in core circuits like sense amplifiers and line drivers. The conventional technique using spare rows and columns is

*This work was supported in part by NSF, under contract MIP-9710130.

Proceedings of the 1997 Workshop on Defect and Fault-Tolerance in VLSI Systems (DFT '97)
0-8186-8168-3/97 $10.00 © 1997 IEEE

incapable of dealing with such defects, and the entire sub-array must be replaced. The yield loss due to chip-kill failures is expected to be about 50% [8].

It became apparent, therefore, that new and more efficient redundancy techniques must be developed [1, 2, 6, 7, 8]. 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 redundant lines at the cost of higher silicon area overhead due to the larger number of required programmable fuses. This approach has been followed in [7], where the design of an experimental 4 Mb SRAM was presented. A 3% increase in the area overhead and up to 61% increase in effective yield have been reported there.

A different approach was presented in [6]. There, fewer local redundant lines were used compared to the traditional technique. For added defect tolerance, the individual sub-array (called Macro in [6]) was fabricated in such a way that it could become part of up to four different memory ICs. The proposed technique was named the Flexible Multi-Macro (FMM) technique and was applied to a 1 Gb DRAM in 0.25μm CMOS technology.

In a previous paper [3] we have analyzed the FMM design and compared its yield to that of the most common defect tolerance technique of adding spare rows and columns to the memory array. Our most important conclusion was that the advantage of the FMM technique over the traditional one cannot be guaranteed. A very careful yield analysis must be performed since the new design can have a higher or a lower yield than the conventional design, depending on the system parameters.

Recently, a different approach for incorporating defect tolerance into memory ICs has been proposed and implemented [8]. This is a hybrid design which combines row and column redundancy with several redundant sub-arrays whose purpose is to replace those sub-arrays hit by chip-kill defects. In what follows we present a yield analysis of this design and show some numerical examples which demonstrate the effect of the different system parameters.

2. Yield Analysis

The designed chip is a 1Gbit multi-bank memory which is planned to be used in shared-memory multiprocessor systems. The block diagram of the chip is shown in Figure 1. The chip includes eight mats of size 128Mbit each and eight redundant blocks of size 1Mbit each. The redundant block consists of four basic 256Kbit arrays and has additional eight spare rows and four spare columns (see Figure 2).

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 defects.

Each mat consists of 512 basic arrays of size 256Kbit (see Figure 3) 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

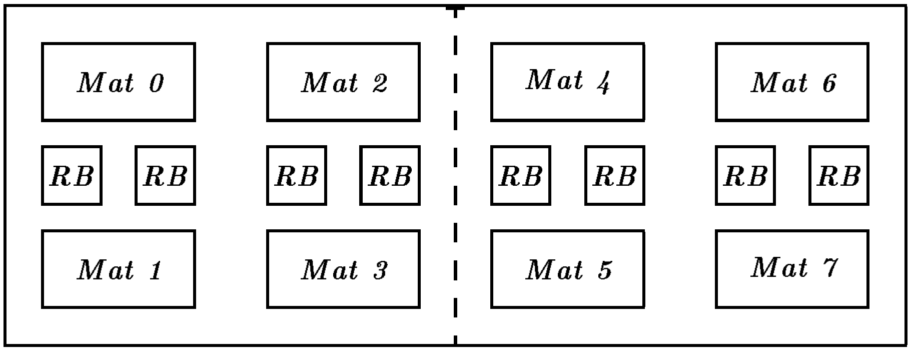

Figure 1: A block diagram of the chip with eight mats of size 128Mbit each and eight redundant blocks (***RB***) of size 1Mbit each.

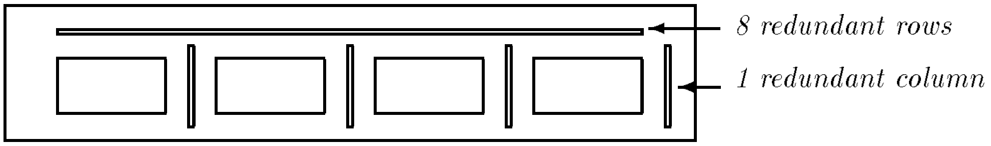

Figure 2: A redundant block including four 256Kbit arrays, eight redundant rows and four redundant columns.

a 16Mbit portion of the mat and eight spare columns are allocated to a 32 Mbit portion of the mat.

Our analysis is based on the two widely-used analytical fault models: The Poisson distribution and the large-area clustering negative binomial distributions [4]. We first calculate the yield based on the relatively simple Poisson distribution, and then use the compounding technique [5] to obtain the results according to the negative binomial model.

The chip consists of two identical, independent halves, and denoting by Y_c and Y_h the yield of the whole chip and that of half the chip, respectively, we have

$$Y_c = (Y_h)^2 \tag{1}$$

Each half chip consists of 4 mats and 4 spare blocks, whose purpose is to correct chip-kill defects in the mats. We denote byp_{ck} the probability that an array has a chip-kill defect, and by G_m and G_{rb} the probability that a mat or a redundant block, respectively, have all their row or column defects fixed using spare rows and columns but may possibly have some chip-kill defects. For a half-chip to be operational, all 4 mats must have no row or column defects, and at most 4 chip-kill defects to be corrected by the spare blocks. The probability of an operational spare block, Y_{rb}, is

$$Y_{rb} = G_{rb}(1 - p_{ck})^4$$

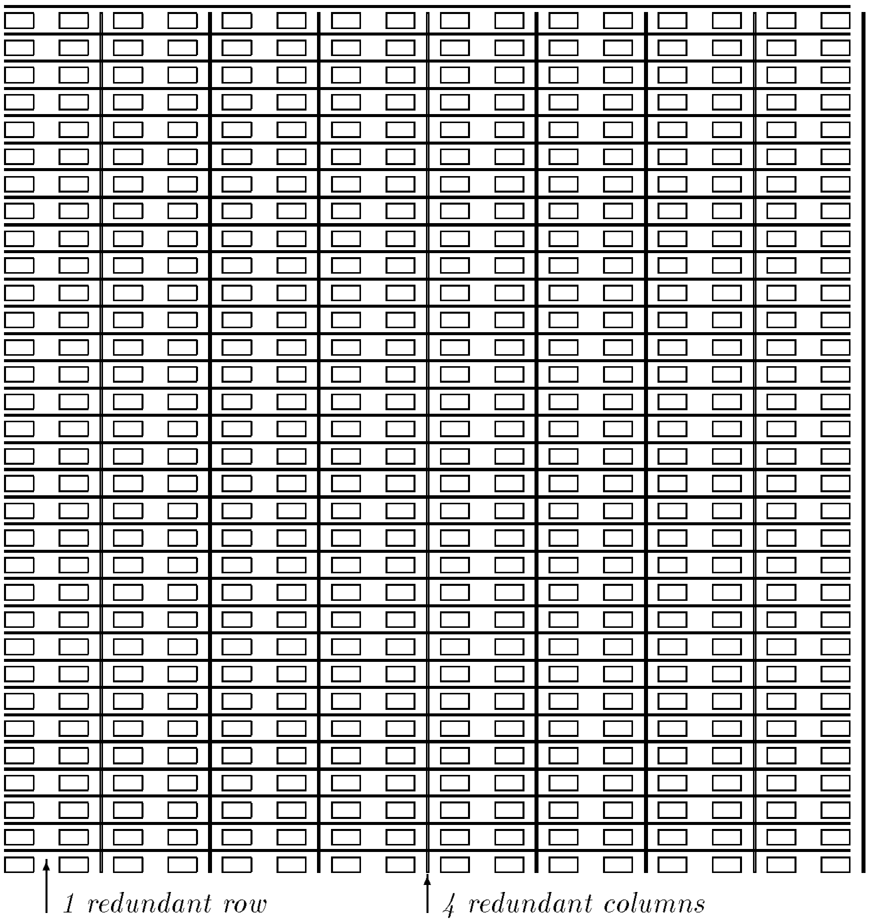

Figure 3: A 128Mb mat containing a 32×16 matrix of 256Kbit arrays.

and the probability of an operational half chip, Y_h, is

$$Y_h = G_m^4 \sum_{i=0}^{4} bin(2048, p_{ck}, i) \sum_{j=i}^{4} bin(4, Y_{rb}, j) \qquad (2)$$

where

$$bin(n,p,k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad (3)$$

The first sum in (2) is the probability that no more than 4 among the 2048 arrays in a half-chip have chip-kill defects, and the second sum is the probability that enough of the spare blocks will be operational to replace those defective arrays.

The probability of an array having a chip-kill defect, p_{ck}, cannot be calculated from the design and is considered a system parameter, but G_m and G_{rb} need to be calculated.

To calculate G_{rb} we make the following approximation: Instead of having 8 spare rows and 4 spare columns, we assume that the block has 12 spare rows. This is a reasonable approximation, since the probability of two defects occurring in the same row or column is very small and can be ignored. Therefore, G_{rb} is the probability of no more than 12 defective rows per block. Denoting by l the average number of defects per bit and using the Poisson distribution we obtain

$$G_{rb} = \sum_{i=0}^{12} bin(n_r + 12, 1 - e^{-n_c l}, i) \qquad (4)$$

where $n_r = 512$ is the number of rows in a block, $n_c = 2048$ is the number of columns per block (or the length of a row), and bin is the binomial probability function defined in (3).

The calculation of G_m is more complicated due to the special redundancy scheme used in each mat, in which four spare rows are allocated to a 16Mbit portion of the mat and eight spare columns are allocated to a 32 Mbit portion of the mat (Figure 3). Dividing the mat into 32 sub-arrays, each consisting of 4×4 arrays, and denoting by d_{ij} the number of defects in sub-array (i, j), the repairable region S consists of the following defect distributions:

$$S = \left\{ (d_{ij}) \mid 1 \le i \le 8,\ 1 \le j \le 4,\ \sum_{i=1}^{8} c(d_{ij}) \le 8\ (j = 1, ..., 4),\ \sum_{j=1}^{4} r(d_{ij}) \le 4\ (i = 1, ..., 8) \right\}$$

where $c(x) = min(x, 8)$ and $r(x) = x - c(x)$
and therefore,

$$G_m = \sum_{S} \prod_{i=1}^{8} \prod_{j=1}^{4} bin(n_b, 1 - e^{-l}, d_{ij}) \qquad (5)$$

where $n_b = (2048 + 4) \times (2048 + 8)$ is the number of bits in a sub-array, l is the average number of defects per bit, d_{ij} is the number of defects in the (i, j)-th sub-array, bin is defined in (3), and S is the set of all repairable defect distributions.

(4) and (5) are now substituted into (2) and then into (1), which results in the chip yield according to the Poisson distribution. The negative binomial distribution can be obtained by compounding the Poisson distribution [5], i.e., assuming that the average number of defects per bit, l, is a random variable with the density function

$$f(l) = \frac{\alpha^\alpha}{\lambda^\alpha , (\alpha)} l^{\alpha - 1} e^{-\frac{\alpha}{\lambda} l} \qquad (6)$$

Multiplying (1) by (6) and integrating with respect to$l$ results in the yield according to the negative binomial distribution with a defect density of λ and a clustering parameter of α. Since the analytic integration is very complicated, we used numerical integration to obtain the numerical results presented in the next section.

3. Numerical Results

In order to evaluate the yield enhancement of the new design as well as to assess the effect of the different system parameters we performed several numerical calculations, all based on the negative binomial model for the defect distribution. The yield is calculated as a function of the defect density λ, for a clustering parameter $\alpha = 0.25$.

We first compared the yield of a chip with only row and column redundancy to that of the same chip with different numbers of redundant blocks and two values of the chip-kill probability. The results are depicted in Figures 4 and 5. We clearly see that some amount of block redundancy is beneficial in both cases, and the improvement is more significant for the higher value of the chip-kill probability. The increase in the yield is much higher than the 2% area increase required for the redundant blocks [8]. We can also conclude from Figure 5 that the number of redundant blocks selected in [8] (i.e., four) is optimal and any increase in this number will not further increase the yield. However, for the lower value of the chip-kill probability in Figure 4, the optimal number of spare blocks is 2 (per half chip) rather than 4.

In Figures 6 and 7 we analyze the effect on the yield of the number of redundant columns in a mat. In Figure 6 the number of spare blocks (per half chip) is four, as it is in the original design, while in Figure 7 there are no spare blocks. In both figures the conclusion is that the optimal number of redundant columns is 32, independent of the number of spare blocks.

Finally, since the exact value of the clustering parameter α cannot always be estimated exactly, we show in Figure 8 the yield as a function of α for different numbers of redundant columns. Although the actual yield depends upon α, we see that the optimal number of redundant blocks is independent of the exact value of α.

4. Conclusion

It has been recently realized that the traditional method for incorporating defect-tolerance (through redundant rows and columns) is no longer effective in sub-micron process technologies. Consequently, designers of memory ICs have proposed and implemented in the last few years new techniques for defect-tolerance. In this paper we analyzed one such design ([8]) which combines the traditional row and column redundancy with spare blocks to handle the increasing number of chip-kill defects. Such an analysis allows the designer to decide on the right number of spare blocks, spare rows and spare columns for maximum yield.

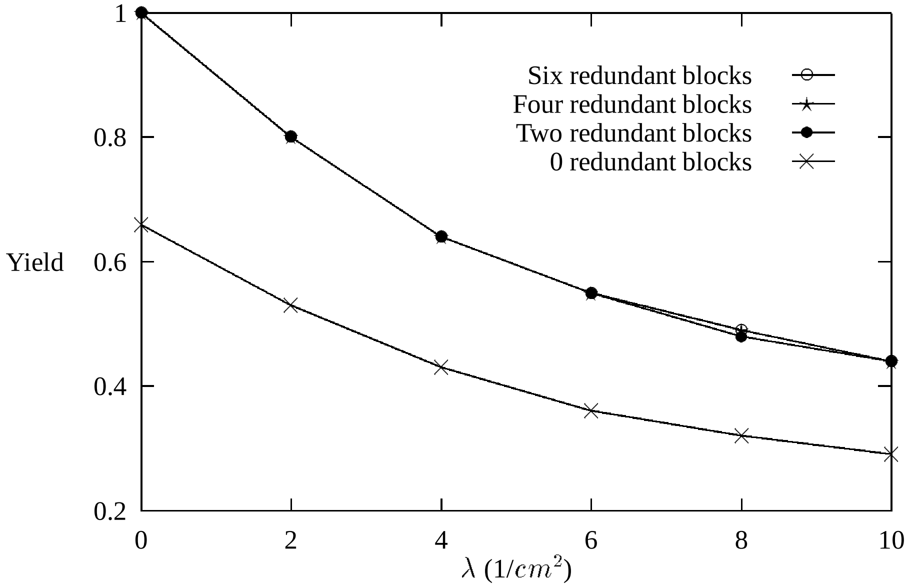

Figure 4: Yield as a function of λ for different numbers of redundant blocks per half chip (Chip kill probability = $1{\cdot}10^{-4}$).

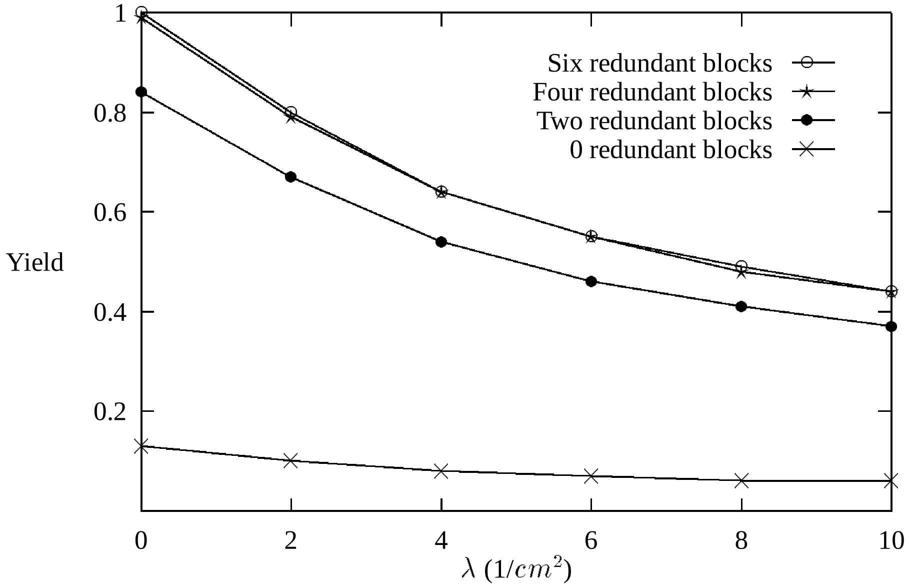

Figure 5: Yield as a function of λ for different numbers of redundant blocks per half chip (Chip kill probability = $5{\cdot}10^{-4}$).

0-8186-8168-3/97 $10.00 © 1997 IEEE

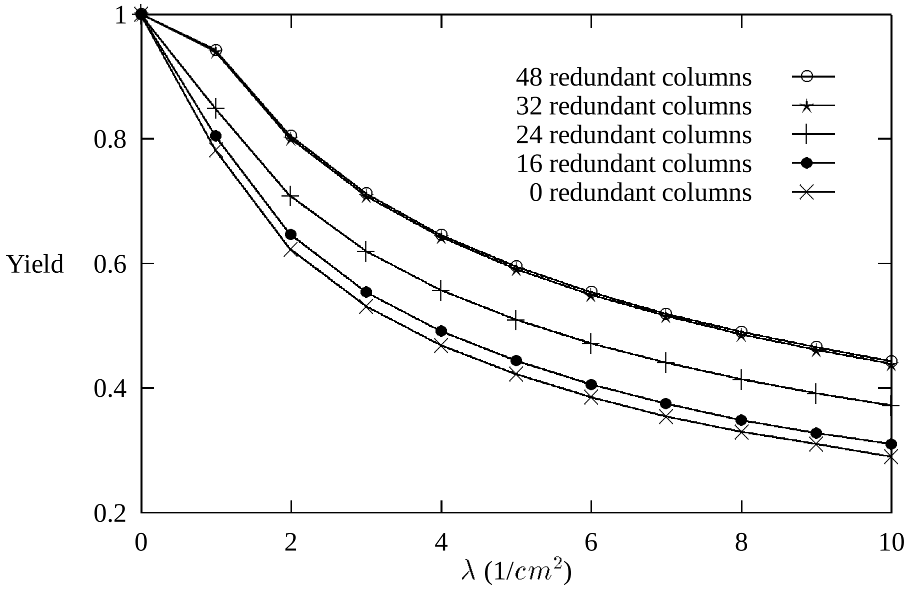

Figure 6: Yield as a function of λ for different numbers of redundant columns per mat (four spare blocks per mat and Chip kill probability = $1{\cdot}10^{-4}$).

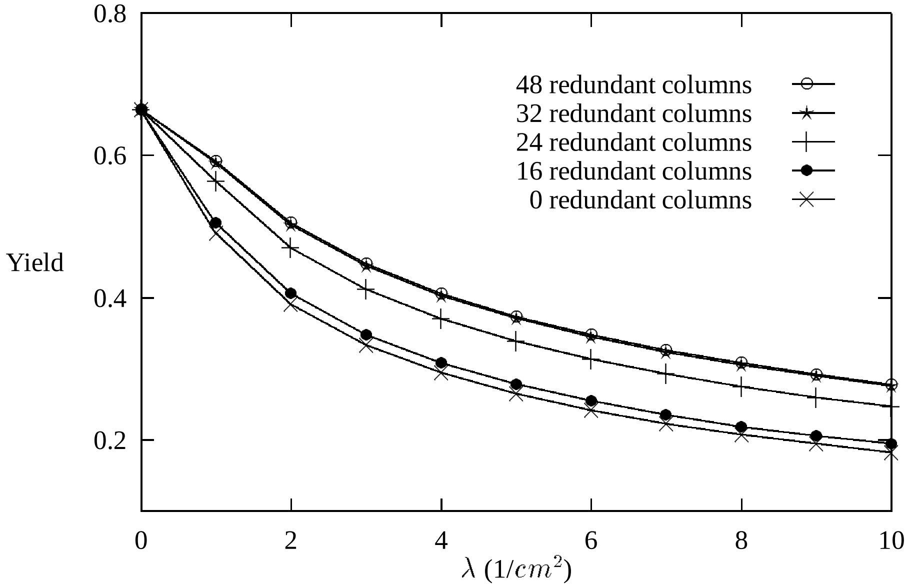

Figure 7: Yield as a function of λ for different numbers of redundant columns per mat (no spare blocks and Chip kill probability = $1{\cdot}10^{-4}$).

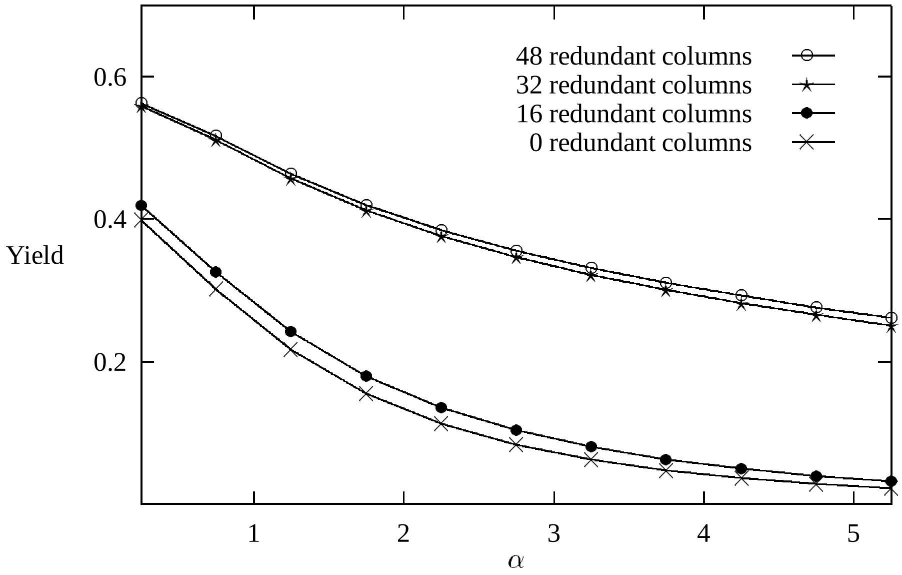

Figure 8: Yield as a function of α for different numbers of redundant columns per mat (Chip kill probability = $1 \cdot 10^{-4}$, $\lambda = 5/cm^2$).

References

[1] T. Kirihata *et al.*, "Fault-Tolerant Designs for 256 Mb DRAM," *IEEE J. of Solid-State Circuits*, vol. 31, pp. 558-566, April 1996.

[2] G. Kitsukawa*et al.*, "256-Mb DRAM Circuit T echnologies for File Applications,"*IEEE J. of Solid-State Circuits*, vol. 28, pp. 1105-11101 Nov. 1993.

[3] I. Koren and Z. Koren, "Yield Analysis of a Novel Scheme for Defect-Tolerant Memories," *Proc. of the 1996 IEEE International Conference on Innovative Systems in Silicon,* pp. 269-278, Austin, Texas, October 1996.

[4] I. Koren, Z. Koren and C.H. Stapper, "A Unified Negativ e Binomial Distribution for Yield Analysis of Defect Tolerant Circuits," *IEEE Trans. on Computers*, vol. 42, pp. 724-437, June 1993.

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

[6] T. Sugibayashi *et al.*, "A 1-Gb DRAM for File Applications," *IEEE J. of Solid-State Circuits*, vol. 30, pp. 1277-1280, Nov. 1995.

[7] T. Yamagata *et al.*, "A Distributed Globally Replaceable Redundancy Scheme for Sub-Half-micron ULSI Memories and Bey ond,"*IEEE J. of Solid-State Circuits*, vol. 31, pp. 195-201, Feb. 1996.

[8] J-H. Yoo *et al.*, "A 32-Bank 1Gb Self-Strobing Synchronous DRAM with 1GB/s Bandwidth," *IEEE J. of Solid-State Circuits*, vol. 31, pp. 1635-1643, Nov. 1996.