

On the Yield of VLSI Processors with On-Chip CPU Cache

D. Nikolos, *Member, IEEE*, and H.T. Vergos

Abstract—Yield enhancement through the acceptance of partially good chips is a well-known technique [1], [2], [3]. In this paper, we derive a yield model for single-chip VLSI processors with partially good on-chip cache. Also, we investigate how the yield enhancement of VLSI processors with on-chip CPU cache relates with the number of acceptable faulty cache blocks, the percentage of the cache area with respect to the whole chip area, and various manufacturing process parameters as defect densities and the fault clustering parameter. One of the main conclusions is that the maximum effective yield is achieved by accepting as good, caches with a very small number of faulty cache blocks. One of the main conclusions is that the maximum effective yield is achieved by accepting as good, **processor chips containing** caches with a very small number of faulty cache blocks.

Index Terms—Fault tolerance, on-chip CPU caches, partially good chips, yield enhancement.

1 INTRODUCTION

ALL the recently developed high-performance single-chip VLSI processors incorporate one or more on-chip CPU caches [4], [5], [6]. The area occupied by these on-chip caches is already a great percentage of the total chip area and is expected to become greater in the near future. Cache memory can be thought as a “redundant” resource in the sense that the correctness of the processor operation does not depend on the presence of the cache. A processor can still operate correctly, although with degraded performance, in the absence of an architecturally invisible cache memory. Thus, to enhance the yield of single-chip VLSI processors with on-chip CPU caches, the acceptance of partially good chips (chips with the faulty cache blocks disabled) has been proposed and the way that the number of faulty cache blocks affects the miss ratio of the cache for various cache sizes and organizations has been investigated [7], [8]. It was shown in [7], [8] that, when the number of the cache faulty blocks is very small, the performance degradation due to the application of the faulty blocks disabling technique is very small. Also, methods to further reduce the performance degradation of the processors with partially good caches were proposed in [9], [10], [11]. The applicability of the faulty block disabling technique depends also on the yield enhancement that can be achieved by accepting as good the chips with a very small number of cache blocks.

To the best of our knowledge, no yield expression has been given for predicting the yield of VLSI processors with a partially good on-chip cache. A cache memory consists of two parts, the tag part and the data part. A defect in a word of the tag part is equivalent (has the same consequences) to one or more defects in the corresponding block of the data part. Thus, the yield expressions derived for partially good memories [2], [3] cannot be used in the case of cache memories. In this paper, we derive a yield model for single-chip VLSI processors with a partially good single level on-chip CPU cache. Using this model, we investigate the dependence of the yield (denoted hereafter as Y) of the partially good chips on the number of acceptable faulty cache

blocks and the percentage of the chip area occupied by the cache. During the manufacture of VLSI processors with on-chip cache, chips with up to R faulty cache blocks can be accepted as good for yield enhancement. The value of R will depend on the required yield and the maximum cache performance degradation that can be accepted. Given the required yield, the yield expression derived in this paper can be used to determine the value of R .

2 YIELD MODEL

It has been generally accepted [12], [13] that the Poisson distribution cannot be used to adequately model the manufacturing defects due to the fact that, in practice, defects are clustered, rather than evenly distributed throughout the wafer. Defect clustering can be modeled by assuming that the number of defects per area unit is Poisson distributed, with the parameter λ being a random variable:

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

The fact that λ is a random variable and not a constant leads to increased clustering, no matter what distribution it follows.

One choice often made [12], [13] of a distribution function for λ is the Gamma distribution with two parameters, α and γ :

$$f(\lambda) = \frac{1}{\gamma^\alpha \Gamma(\alpha)} \lambda^{\alpha-1} e^{-\frac{\lambda}{\gamma}}. \quad (2)$$

Averaging λ in (1) with respect to (2) results in the defects per unit area being distributed according to the negative binomial distribution:

$$\text{Prob}\{X = x\} = \frac{\Gamma(x + \alpha)}{x! \Gamma(\alpha)} \frac{\gamma^x}{(1 + \gamma)^{\alpha+x}}.$$

One of the most useful properties of the Poisson distribution, which the negative binomial one lacks, is the statistical independence between defects in disjoint areas. For overcoming this difficulty and to calculate the yield when the negative binomial distribution is assumed, we follow a method based on the well-known total probability theorem [14, p. 23]. That is, we assume Poisson distribution for the defects, utilizing the independence property of this distribution to calculate the yield for a fixed λ value. By averaging the result over all values of λ , using the Gamma density function, we obtain the yield for the negative binomial model [12], [13].

In our case, that is, processors with on-chip cache, we first consider statistical independence between defects in three disjoint areas, the data part of the cache, the tag part of the cache, and the rest part of the chip (processor and the cache support circuit). We then average the result over all values of λ (number of defects), using the Gamma distribution function. A chip is usable when the processor and the cache support circuits are fault-free, even if some of the tags and /or the data blocks are faulty. Thus, chips with faults in the processor and/or the cache support circuits are discarded, while those with fault-free processor and cache support circuits and some faulty tags and/or data blocks are accepted as good.

Let N denote the number of cache blocks. Then, the yield can be expressed as a probability as follows:

• The authors are with the Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Patras, Greece.
E-mail: {nikolosd, vergos}@cti.gr.

Manuscript received 24 Feb. 1997; revised 25 Aug. 1999.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 104013.

$Y = \{\text{Prob at least } M \text{ out of the } N \text{ cache blocks are operational}$
 $\text{and the rest chip is fault - free}\}$
 $= \text{Prob}\{\text{at most } R, R = N - M, \text{ cache blocks are not operational}$
 $\text{and the rest chip is fault - free}\}$

$$= \sum_{i=0}^R \alpha_{i,N},$$

where $\alpha_{i,N} = \text{Prob}\{\text{exactly } i \text{ cache blocks are not operational}$
 $\text{and the rest chip is fault - free}\}$.

(3)

i cache blocks are not operational means that s tags and q data blocks are not operational. We have to note here that a tag corresponds to just one data block. Let t be the number of faulty tags that correspond to faulty data blocks. Then, $0 \leq t \leq \min\{s, q\}$ and $s + q - t = i$. The assumption that the s tags and the q data blocks belong to different cache blocks, that is, $t = 0$, inserts a very small error for small values of R . The yield expression that will be derived by making the above assumption will result in slightly smaller values for the yield of the chips with partially good cache than in the case that this assumption is not made. It is evident that this assumption does not affect the perfect chip yield. We call the chips that are fault-free perfect chips.

Making the above assumption we get:

$$\alpha_{i,N} = \sum_{s=0}^i \text{Prob}\{\text{exactly } s \text{ tags and } q = i - s \text{ data blocks are not}$$

$$\text{operational while the rest chip is fault - free}\}$$

$$= \sum_{s=0}^i \beta_{s,q}.$$

(4)

We consider that the faults occurring in different modules are independent (as in the case where the faults follow the Poisson distribution). We then have:

$$\beta_{s,q} = c d_s g_q, \quad (5)$$

where

$c = \text{Prob}\{\text{the processor and the rest support circuit is fault - free}\}$
 $d_s = \text{Prob}\{\text{exactly } s \text{ tags of the cache tag part are faulty}\}$ and
 $g_q = \text{Prob}\{\text{exactly } q \text{ blocks of the cache data part are faulty}\}$.

Following Poisson distribution for the defects, we have

$$c = e^{-\lambda_{ck}}, \quad (6)$$

where λ_{ck} is the average number of defects per chip in the processor and the rest support circuits.

In the case of the tag part of the cache memory, the identical modules are the tags. Considering the area requirements of a tag, which are very small (in the order of the area occupied by a few static RAM cells), it is evident that the probability of a single fault affecting more than one tag is greater than the probability of a tag containing any number, greater than one, of faults. In our analysis, we consider that one fault affects one tag. In the case that one fault affects two or more tags, we consider that two or more faults have occurred. Assuming Poisson distribution for the defects of the tag memory, we get:

$$d_s = \frac{e^{-\lambda_{tag}} \lambda_{tag}^s}{s!}, \quad (7)$$

where λ_{tag} is the average number of defects per chip in the tag part of the cache.

In the case of the data part of the cache memory, the identical modules are the blocks which usually consist of 8, 16, or 32 bytes. Because of the large area of the block with respect to the area of spot defects, we consider that a module may have any number of faults. If the faults occurring in different modules are independent, using binomial distribution we can get

$$g_q = \binom{N}{q} y^{N-q} (1-y)^q, \quad (8)$$

where y is the yield of a single data block, given by $y = e^{-\lambda_{block}}$ and λ_{block} is the average number of defects per block. By expanding $(1-y)^q$ into the following binomial series, we get

$$(1-y)^q = \sum_{k=0}^q (-1)^k \binom{q}{k} y^k,$$

and, by substituting this in (8), we get

$$g_q = \binom{N}{q} \sum_{k=0}^q (-1)^k \binom{q}{k} e^{-(N-q+k)\lambda_{block}}. \quad (9)$$

Therefore, from (5), (6), (7), and (9), we have:

$$\beta_{s,q} = e^{-\lambda_{ck}} \frac{e^{-\lambda_{tag}} \lambda_{tag}^s}{s!} \binom{N}{q} \sum_{k=0}^q (-1)^k \binom{q}{k} e^{-(N-q+k)\lambda_{block}}.$$

We next have to apply the compounding procedure [12], [13] in order to calculate the yield when clustering of faults is allowed. We must not, however, perform three separate compounding steps (for the two types of modules and the support circuits) since the clustering of faults in one type of circuits is not independent of the clustering in the other two. Therefore, we must perform a single compounding step using the average number of faults in the complete chip, i.e., $\lambda = \lambda_{ck} + \lambda_{tag} + N\lambda_{block}$.

To simplify the integration which contains different multiples of λ , we define:

$$\delta_1 = \frac{\lambda_{ck}}{\lambda}, \quad \delta_2 = \frac{\lambda_{tag}}{\lambda}, \quad \delta_3 = \frac{N\lambda_{block}}{\lambda}.$$

Note that δ_1 , δ_2 , and δ_3 are constants which mainly depend on the ratio of the corresponding chip areas to the area of the whole chip. The exponential term now becomes:

$$e^{-\lambda_{ck} - \lambda_{tag} - (N-q+k)\lambda_{block}} = e^{-[\delta_1 + \delta_2 + (N-q+k)\delta_3/N]\lambda}.$$

Then, considering as compounder the Gamma distribution with two parameters α and γ (2), we get:

$$\beta_{s,q} = \binom{N}{q} \sum_{k=0}^q (-1)^k \binom{q}{k} \int_0^\infty e^{-[\delta_1 + \delta_2 + (N-q+k)\delta_3/N]\lambda} \frac{(\delta_2 \lambda)^s}{s!} f(\lambda) d\lambda.$$

After the evaluation of the integral (hints are provided in the Appendix), we get:

$$\beta_{s,q} = \binom{N}{q} \sum_{k=0}^q (-1)^k \binom{q}{k} \frac{\Gamma(\alpha + s)}{s! \Gamma(\alpha)} \left(\frac{\delta_2 \bar{\lambda}}{\alpha}\right)^s$$

$$\left(1 + \frac{[\delta_1 + \delta_2 + (N-q+k)\delta_3/N]\bar{\lambda}}{\alpha}\right)^{-\alpha-s}. \quad (10)$$

Finally, we define

$$\delta_1 \bar{\lambda} = \bar{\lambda}_{ck}, \quad \delta_2 \bar{\lambda} = \bar{\lambda}_{tag}, \quad \delta_3 \bar{\lambda} = N \bar{\lambda}_{block}. \quad (11)$$

Combining (3), (4), (10), and (11), we get the following yield expression for processors with a single level of partially good cache memory, when at most R cache faulty blocks are acceptable:

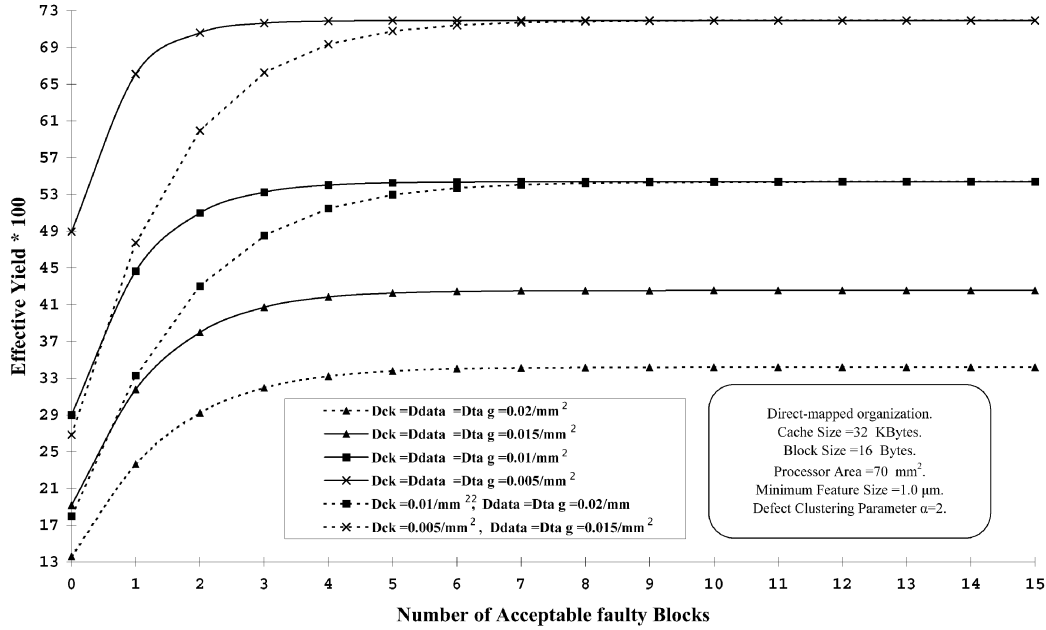


Fig. 1. Effective yield vs. acceptable faulty blocks for various defect densities.

$$Y = \sum_{i=0}^R \sum_{s=0}^i \left\{ \binom{N}{i-s} \sum_{k=0}^{i-s} (-1)^k \binom{i-s}{k} \frac{\Gamma(\alpha + s)}{s! \Gamma(\alpha)} \left(\frac{\bar{\lambda}_{tag}}{\alpha} \right)^s \left[1 + \frac{\bar{\lambda}_{ck} + \bar{\lambda}_{tag} + (N + k + s - i) \bar{\lambda}_{block}}{\alpha} \right]^{-\alpha - s} \right\} \quad (12)$$

$\bar{\lambda}_{block} = A_{block} D_{data}$, where A and D stand for the area and the defect density in the corresponding parts of the chip.

It is evident that the derived expression for the yield can be applied independently of the cache organization, direct mapped, set associative, or fully associative.

Note that in the above expression, a is the defect clustering parameter and $\bar{\lambda}_{ck} = A_{ck} D_{ck}$, $\bar{\lambda}_{tag} = A_{tag} D_{tag}$, and

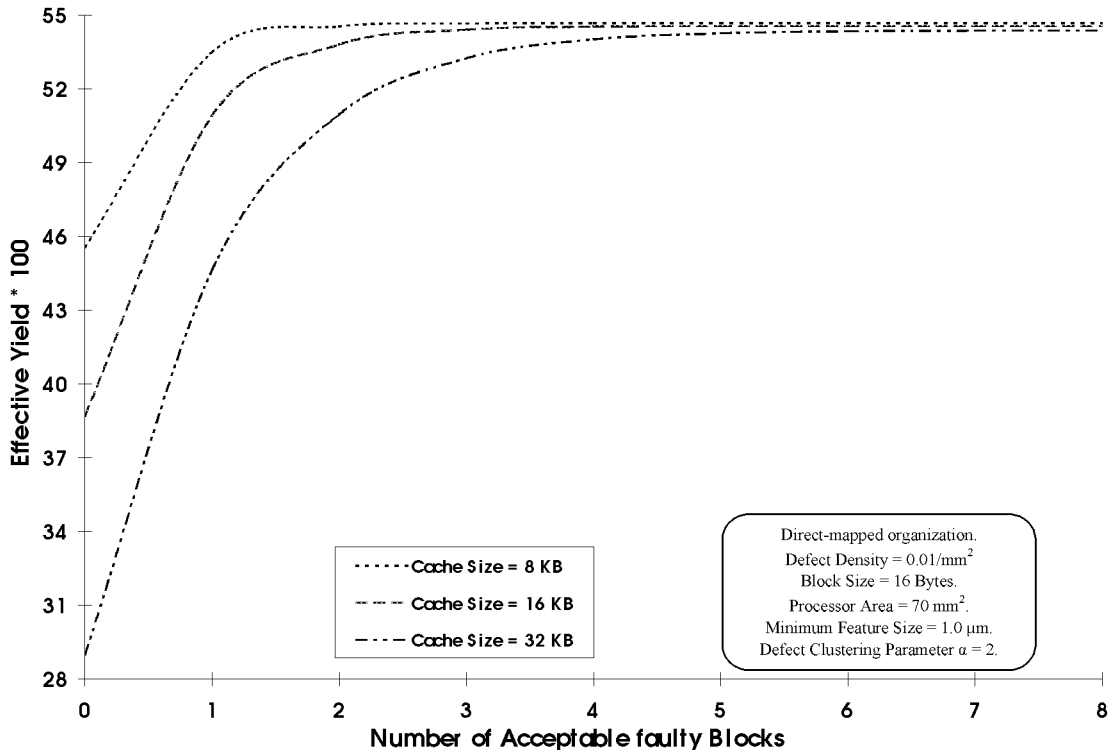


Fig. 2. Effective yield vs. acceptable faulty blocks for various cache sizes.

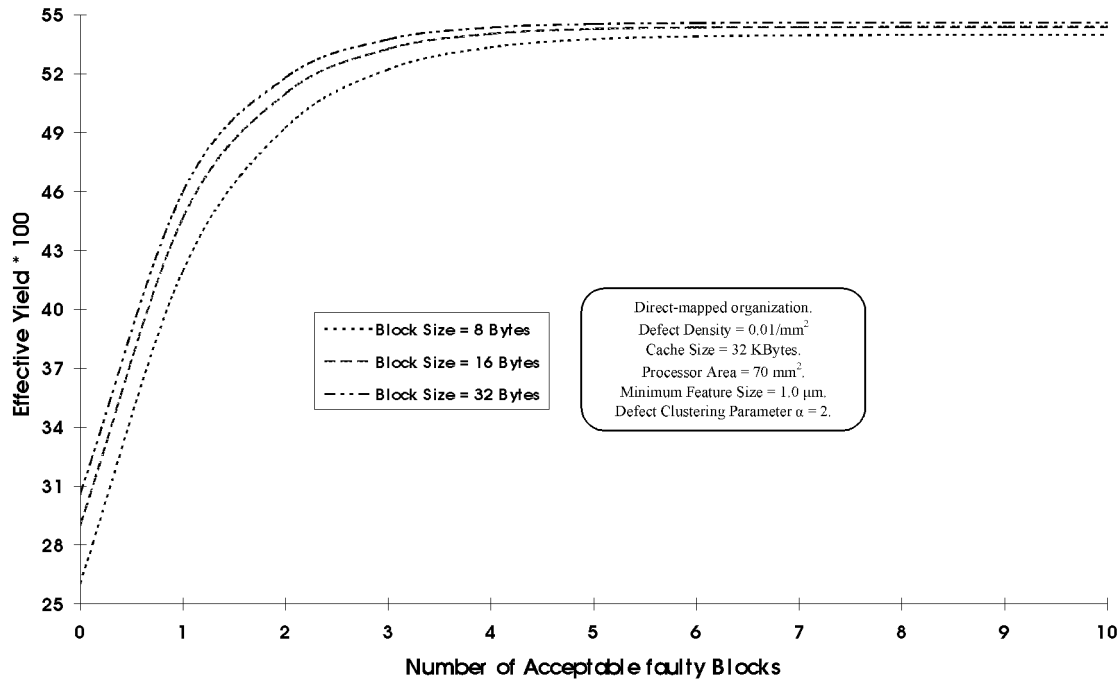


Fig. 3. Effective yield vs. acceptable faulty blocks for various block sizes.

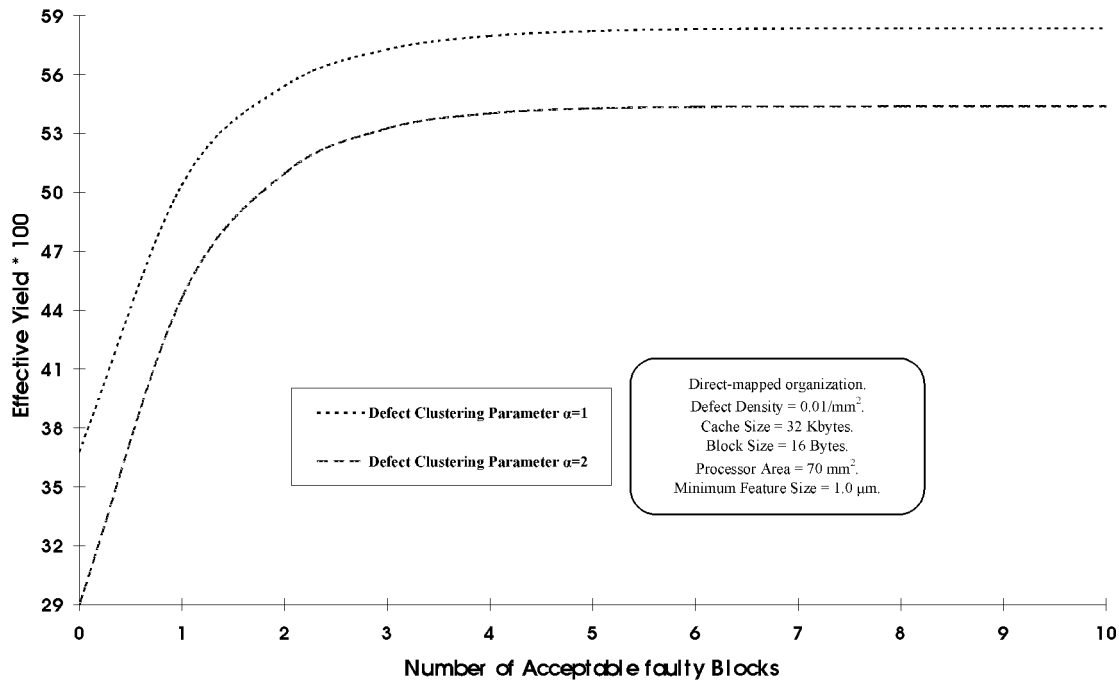


Fig. 4. Effective yield vs. acceptable faulty blocks for various clustering parameter values.

3 DISCUSSION

Having obtained an expression for the yield, we can study how the yield depends on various parameters as the acceptable number R of faulty cache blocks, the percentage of the cache area with respect to the whole chip area, the values of D_{ck} , D_{tag} , D_{data} , as well as the fault clustering parameter a .

We have to note that any yield enhancement technique that is used for the on-chip cache of VLSI processor chips requires some extra implementation area. This extra implementation area should

be kept as small as possible because any additional area may be the cause for increased number of defects per chip and may result in perfect chip yield loss. Moreover, when the area of a chip increases, the number of chips per wafer tends to decrease. Therefore, we have to consider the effective yield, which is the chip yield multiplied by the area increase factor, as a most suitable metric rather than the yield itself. For applying the faulty block disabling technique, one additional bit (availability bit) should be added in each tag of the cache, whose value denotes whether the corresponding block is faulty or not.

TABLE 1

Defect densities	Effective yield			Effective yield enhancement	
	perfect chip (Y_p)	one faulty cache block (Y_1)	maximum value (Y_{max})	$Y_1 - Y_p$	$Y_{max} - Y_p$
$D_{ck}=D_{data}=D_{tag} = 0.02/\text{mm}^2$	13.599	23.647	34.194	10.048	20.595
$D_{ck}=D_{data}=D_{tag} = 0.015/\text{mm}^2$	19.165	31.772	42.549	12.607	23.384
$D_{ck}=D_{data}=D_{tag} = 0.01/\text{mm}^2$	29.005	44.654	54.387	15.649	25.382
$D_{ck}=D_{data}=D_{tag} = 0.005/\text{mm}^2$	48.949	66.102	71.952	17.153	23.003
$D_{ck} = 0.01/\text{mm}^2, D_{data}=D_{tag}=0.02/\text{mm}^2$	17.991	33.281	54.387	15.290	36.396
$D_{ck} = 0.005/\text{mm}^2, D_{data}=D_{tag}=0.015/\text{mm}^2$	26.842	47.741	71.952	20.899	45.110

Using a large set of values for the parameters of the yield expression, we derived a large set of curves for the yield of VLSI processor chips with partially good on-chip cache as a function of the number of the accepted cache faulty blocks. Figs. 1, 2, 3, and 4 present representative samples. The yield model presented in this paper can be applied equally well no matter which layout organization for the on-chip cache is followed. We have considered the layout organization that leads to the best cache cycle time (as computed by the model presented in [15]). For estimating the area that the on-chip cache occupies, we used the area model presented in [16]. The choice $D_{ck} \leq D_{tag} = D_{data}$ is based on the fact that cache arrays are fabricated with the tightest feature and scaling rules available in a given technology which means that caches are more susceptible to faults [17], [18]. Only experimental data obtained by monitoring wafers can show which values of D_{ck} , D_{tag} , and D_{data} must be used in the yield expression.

- Fig. 1 presents the effective yield as a function of the number of the acceptable faulty blocks for various values of the defect densities. We can see that, in all cases, the effective yield increases significantly with the number of acceptable cache faulty blocks until we reach a value beyond which the effective yield is practically constant. Even if we accept as good chips those with just one faulty cache block, we achieve a significant yield enhancement. Specific values corresponding to Fig. 1 are given in Table 1. We can also see that the maximum effective yield is achieved by accepting a small number of faulty cache blocks. Therefore, there is no need to accept as good, chips with a large number of faulty blocks and, hence, a significantly degraded cache performance.

From Table 1 and Fig. 1, we can also see that the effective yield enhancement is greater for small values of the defect densities and even greater for $D_{ck} < D_{tag} = D_{data}$. This implies that in mature fabrication technologies, where the values of the defect densities are smaller, the technique of accepting chips with partially good caches will be more effective.

- Fig. 2 presents the effective yield as a function of the number of the acceptable faulty blocks for caches with capacity 8 KB, 16 KB, and 32 KB and constant processor

area. We can see that the yield enhancement achieved by accepting as good, chips with one, two, or more cache faulty blocks increases as the percentage of the total chip area devoted to the on-chip cache increases.

As expected, keeping all other parameters constant, the perfect yield of the chips depends heavily on the cache size (that is, the area occupied by the cache). However, as the number of the accepted faulty cache blocks increases, the yield of the chips with partially good caches approximates almost the same value Y_a independently of the cache size. The number of the faulty cache blocks that should be accepted in order to approximate the value Y_a increases when the cache size becomes larger. From Fig. 2, we can also see that, when we accept as good chips with up to one faulty cache block, the effective yield of the chips with a cache of size equal to 16 KB is greater than the perfect chip yield of the chips with a cache of size equal to 8 KB. We think that this observation is very significant and its exploitation is under investigation.

- Fig. 3 presents the effective yield as a function of the number of the acceptable faulty cache blocks for caches with block size equal to 8, 16, and 32 bytes. When the block size increases, for constant cache size, the number of cache blocks, as well as the number of tags, decreases. Thus, when we move to larger block sizes, the tag memory occupies less area and the yield is increased. This can be verified from Fig. 3. However, the effective yield enhancement that can be achieved by the acceptance of processors with partially good on-chip cache does not depend strongly on the block size.
- Fig. 4 presents the effective yield as a function of the number of the acceptable faulty cache blocks with parameter the value of the defect clustering parameter α . Table 2 gives characteristic values of Fig. 4. From Fig. 4 and Table 2, we can see that as the value of α gets significantly smaller as the achieved effective yield enhancement becomes slightly smaller. Therefore, the yield enhancement due to accepting chips with partially good cache is insensitive to the value of the defect clustering parameter α .

TABLE 2

Defect clustering parameter α	Effective yield			Effective yield enhancement	
	perfect chip (Y_p)	one faulty cache block (Y_1)	maximum value (Y_{max})	$Y_1 - Y_p$	$Y_{max} - Y_p$
$\alpha = 1$	36.809	50.407	58.368	13.598	21.559
$\alpha = 2$	29.005	44.654	54.387	15.649	25.382

As we have mentioned earlier, every yield enhancement technique for processors with on-chip cache requires some extra implementation area. The extra area requirements for implementing the availability bit for the faulty block disabling technique are very small and, hence, the perfect chip yield degradation is negligible. For example, consider a chip in which the area occupied by the processor and the support circuits is 70 mm², the defect clustering parameter is 2, the defect density is 0.01 defects/mm², and the chip is fabricated in 1.0 μ m feature size. If the cache is 32KB and the block size is 32 Bytes, the yield of the perfect chips when no redundancy is used is 30.723 percent, whereas, when the cache is equipped with the availability bits, the effective yield is 30.609 percent, that is, only 0.37 percent lower. The corresponding values for the 16KB-16Bytes case, are 38.9 percent and 38.723 percent and the perfect chip yield loss is only 0.46 percent.

4 CONCLUSION

In this paper a yield model for single-chip VLSI processors with partially good on-chip cache was derived. Using this model, we have shown that by accepting as good chips with a very small number of faulty cache blocks, we achieve a significant increase of the yield. We have also shown how the yield depends on various parameters as the percentage of the cache area with respect to the whole chip area, the cache block size, and various manufacturing process parameters such as defect densities and the fault clustering parameter. The cache faulty block disabling technique can be used alone or along with the technique of spare rows and/or spare columns that was used earlier in RAMs [19] and recently in caches memories [20].

APPENDIX

$$\left(\frac{d}{dz}\right)^s \int_0^\infty e^{-z\lambda} f(\lambda) d\lambda = (-1)^s \int_0^\infty e^{-z\lambda} \lambda^s f(\lambda) d\lambda.$$

Then, taking into account that

$$\int_0^\infty e^{-z\lambda} f(\lambda) d\lambda = \left(1 + \frac{z\lambda}{\alpha}\right)^{-\alpha}$$

and

$$\begin{aligned} \left(\frac{d}{dz}\right)^s \left(1 + \frac{z\lambda}{\alpha}\right)^{-\alpha} \\ = (-1)^s a(a+1)(a+2) \dots (a+s-1) \left(\frac{\lambda}{a}\right)^s \left(1 + \frac{z\lambda}{\alpha}\right)^{-\alpha-s}, \end{aligned}$$

we get

$$\int_0^\infty e^{-z\lambda} \lambda^s f(\lambda) d\lambda = \frac{\Gamma(\alpha+s)}{\Gamma(\alpha)} \left(\frac{\lambda}{\alpha}\right)^s \left(1 + \frac{z\lambda}{\alpha}\right)^{-\alpha-s}.$$

ACKNOWLEDGMENTS

The authors would like to thank Prof. G. Moustakides for his help in the evaluation of the above integral. A preliminary version of this work was presented at the Second European Dependable Computing Conference, Taormina, Italy, 2-4 October 1996.

REFERENCES

- [1] I. Koren and A.D. Singh, "Fault Tolerance in VLSI Circuits," *Computer*, pp. 73-83, July 1990.
- [2] 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. Research and Development*, vol. 20, pp. 398-409, 1980.
- [3] 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.
- [4] "PowerPC 601—RISC Microprocessor User's Manual," Motorola Semiconductor Technical Data Book, 1993.
- [5] S. Miraburi et al., "The MIPS R4000 Processor," *IEEE Micro*, pp. 10-22, Apr. 1992.
- [6] J.H. Edmondson et al., "Superscalar Instruction Execution in the 21164 Alpha Microprocessor," *IEEE Micro*, pp. 33-43, Apr. 1995.
- [7] G. Sohi, "Cache Memory Organization to Enhance the Yield of High-Performance VLSI Processors," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 484-492, Apr. 1989.
- [8] A.F. Pour and M.D. Hill, "Performance Implications of Tolerating Cache Faults," *IEEE Trans. Computers*, vol. 42, no. 3, pp. 257-267, Mar. 1993.
- [9] H.T. Vergos and D. Nikolos, "Efficient Fault Tolerant CPU Cache Memory Design," *Microprocessing and Microprogramming—The Euromicro J.*, vol. 41, pp. 153-169, May 1995.
- [10] H.T. Vergos and D. Nikolos, "Performance Recovery in Direct-Mapped Faulty Caches via the Use of a Very Small Fully Associative Spare Cache," *Proc. IEEE Int'l Computer Performance and Dependability Symp. (IPDS '95)*, pp. 326-332, Erlangen, Germany, Apr. 1995.
- [11] H.T. Vergos et al., "Reconfigurable CPU Cache Memory Design: Fault Tolerance and Performance Evaluation," *VLSI: Integrated Systems on Silicon, Proc. Ninth IFIP Very Large Scale Integrated Systems Conf. (VLSI '97)*, R. Reis and L. Claesen, eds., pp. 103-114, Gramado, Brazil, 1997.
- [12] 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, no. 3, pp. 859-865, June 1988.
- [13] I. Koren and C.H. Stapper, "Yield Models for Defect-Tolerant VLSI Circuits: A Review," *Defect and Fault Tolerance in VLSI Systems*, vol. 1, pp. 1-21, I. Koren, ed., New York: Plenum, 1989.
- [14] L.J. Bain and M. Engelhardt, *Introduction to Probability and Math. Statistics*, second ed. Belmont, Calif.: Duxbury Press, 1991.
- [15] S.J.E. Wilton and N.P. Jouppi, "An Enhanced Access and Cycle Time Model for On-Chip Caches," Technical Report 93/5, DEC Western Research Lab, 1994.
- [16] J.M. Mulder, N.T. Quach, and M.J. Flynn, "An Area Model for On-Chip Memories and its Application," *IEEE J. Solid-State Circuits*, vol. 26, no. 2, pp. 98-106, Feb. 1991.
- [17] M.G. Gallup et al., "Testability Features of the 68040," *Proc. Int'l Test Conf.*, pp. 749-757, Washington, D.C., Sept. 1990.
- [18] N.R. Saxena et al., "Fault-Tolerant Features in the HaL Memory Management Unit," *IEEE Trans. Computers*, vol. 44, no. 2, pp. 170-179, Feb. 1995.

- [19] W.R. Moore, "A Review of Fault-Tolerant Techniques for the Enhancement of Integrated Circuit Yield," *Proc. IEEE*, vol. 74, no. 4, pp. 684-698, May 1986.
- [20] W.J. Bowhill et al., "Circuit Implementation of a 300-MHz 64-bit Second-Generation CMOS Alpha CPU," *Digital Technical J.*, vol. 7, no.1, pp.100-117, 1995.