Yield - Performance Tradeoffs for VLSI Processors with Partially Good Two-Level On-Chip Caches

D. Nikolos, H. T. Vergos, A. Vazaios & S. Voulgaris
Dept. of Computer Engineering & Informatics
University of Patras, 26500 Patras, GREECE
e-mail : nikolosd@cti.gr

Abstract

In this paper a yield model for single chip VLSI processors with two level on-chip caches is derived. Using this model and trace driven simulations the distribution of the faulty cache blocks into the first and second level caches can be determined so as to achieve a significant yield enhancement with the minimum performance degradation.

1. Introduction

The area devoted to on-chip caches in the modern processors is already a large fraction of the chip area and is expected to be larger in the near future. The 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 [1, 2]. From the above we conclude that a substantial portion of the manufacturing defects will occur in the cache memory of a VLSI processor chip. If cache defects can be tolerated without a substantial performance loss, then the yield of VLSI processors with on-chip cache can be enhanced considerably. The performance degradation due to the disabling of the fault cache blocks of single chip VLSI processors with one level on-chip cache, as well as methods to reduce the performance degradation were considered in [3-7]. However, none of them has considered how the yield increases with respect to the number of the acceptable defective cache blocks.

Computer designers face the problem of building a cache that has both a low miss rate and a short access time. A solution is to provide more than one level of cache memory [9, 10]. In a two-level cache hierarchy, the level one cache is made small and fast to match the CPU speed and the level two cache is made larger, and thus slower, to keep the overall cache miss rate low. In this work single chip VLSI processors with two level on-chip caches are considered. The aim of this work is twofold. First aim is to investigate how the yield enhancement of VLSI processors with on-chip CPU cache relates with the number of acceptable faulty cache blocks, the distribution of the faulty cache blocks into the first and second level cache, the percentage of the cache area with respect to the whole chip area and various manufacturing process parameters such as defect densities and the fault clustering parameter. To this end, a yield expression was derived for the case that the first level cache consists from a data and an instruction cache (split cache) while the second level is a unified cache. This organization is the most commonly used organization, but the yield expression can easily be modified for any other case. The second aim is to consider the increase of the cache average access time (performance degradation), due to the disabling of the faulty cache blocks, as a function of the number of the faulty cache blocks and their distribution into the first and the second level cache. During the manufacture testing of VLSI processors with on-chip cache, dies with up to R_{in} and R_d faulty blocks in the first level instruction and data cache respectively and R_u faulty blocks in the second level cache will be accepted as good for yield enhancement. As we will see there are values of $R_{d,}$ R_{in} and R_u which increase slightly the yield, while degrade the processor performance significantly. The present work leads to the determination of the values of R_{in}, R_d and R_u, that offer good yield enhancement with small performance degradation.

1063-6722/96 $5.00 © 1996 IEEE

Proceedings of the 1996 Workshop on Defect and Fault-Tolerance in VLSI Systems (DFT)
1063-6722/96 $10.00 © 1996 IEEE

2. Yield Enhancement

We assume Poisson distribution for the defects, and we use 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 distribution function, we obtain the yield for the negative binomial model [8, 11].

Suppose that the first level data and instruction cache consist of N_d and N_{in} blocks respectively and the second level cache consists of N_u blocks. We accept chips that contain up to R_d, R_{in}, R_u not operational blocks in the three caches mentioned above respectively. Then the yield (Y) of the CPU chip is :
Y = Prob { chip operational } =
= Prob { at most R_d data cache blocks, R_{in} instruction cache blocks, R_u unified cache blocks are not operational and the rest chip is fault free }

We consider that the faults occurring in different modules are independent (as in the case that the faults follow the Poisson distribution }. Then : $Y = Y_d\, Y_{in}\, Y_u\, Y_s$, where
Y_d = Prob { at most R_d data cache blocks are not operational },
Y_{in} = Prob { at most R_{in} instruction cache blocks are not operational },
Y_u = Prob { at most R_u unified cache blocks are not operational }and
Y_s = Prob { the processor and the rest support circuit is fault free }.

$Y_d = \sum_{i=0}^{R_d} \alpha_{i,N_d}$, where α_{i,N_d}= Prob { exactly *i* data cache blocks are not operational },

$Y_{in} = \sum_{j=0}^{R_{in}} \alpha_{j,N_{in}}$, where $\alpha_{j,N_{in}}$= Prob { exactly *j* instruction cache blocks are not operational } and

$Y_u = \sum_{m=0}^{R_u} \alpha_{m,N_u}$, where α_{m,N_u}= Prob { exactly *m* unified cache blocks are not operational }.

We note that in a cache there are exactly R not operational blocks, when s tags and q cache blocks, with s + q = R, are not operational. Making the assumption that the s tags and the q data blocks belong to different cache blocks (which inserts a very small error for small values of R), we get :

$\alpha_{i,N_d} = \sum_{s_I=0}^{i}$ Prob { exactly s_I tags and $q_I = i - s_I$ data cache blocks are not operational } $= \sum_{s_1=0}^{i} \beta_{s_1,q_1}$.

We have already assumed that the faults occurring in different modules are independent, thus we have: $\beta_{s_1,q_1} = h_{s_1} g_{q_1}$, where h_{S_1}= Prob { exactly s_I tags of the tag part of the data cache are faulty } and
g_{q_1}= Prob { exactly q_I blocks of the data part of the data cache are faulty }.

In the same way we get: $\alpha_{j,N_{in}} = \sum_{s_2=0}^{j} \beta_{s_2,q_2}$ with $\beta_{s_2,q_2} = h_{s_2} g_{q_2}$ and $\alpha_{m,N_u} = \sum_{s_3=0}^{m} \beta_{s_3,q_3}$ with $\beta_{s_3,q_3} = h_{s_3} g_{q_3}$.

In the case of the data part of a 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. Considering that the data part of a cache has N_t blocks and using binomial distribution we get : $g_{q_t} = \binom{N_t}{q_t} y^{N_t - q_t} (1-y)^{q_t}$ (1)

where y is the yield of a single cache data block, that is, $y = e^{-\lambda^t_{block}}$ and λ^t_{block} is the number of defects per block.By expanding $(1-y)^{q_t}$ into the binomial series $\sum_{k_t=0}^{q_t} (-1)^{k_t} \binom{q_t}{k_t} y^{k_t}$ and substituting in (1) we get

$$g_{q_t} = \binom{N_t}{q_t} \sum_{k_t=0}^{q_t} (-1)^{k_t} \binom{q_t}{k_t} e^{-(N_t - q_t + k_t)\lambda^t_{block}}$$

The above relation for t =1, 2 and 3 gives the expressions of g_{q_1}, g_{q_2} and g_{q_3} ($N_1 = N_d$, $N_2 = N_{in}$, $N_3 = N_u$).

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 a single fault to affect more than one tags is greater than the probability a tag to contain any number, greater than one, of faults. In our analysis we consider that one fault affects one tag. Assuming Poisson distribution for the defects of the tag memory, we get :

$$h_{S_t} = e^{-\lambda_{tag}^t}\left(\lambda_{tag}^t\right)^{s_t} \Big/ s_t!$$

The above relation for t = 1, 2 and 3 gives the expressions of h_{S_1}, h_{S_2} and h_{S_3}.

Following Poisson distribution for the defects in the processor and the rest support circuits of the chip we have : $Y_s = e^{-\lambda_{ck}}$ where λ_{ck} is the number of defects in the processor and the rest support circuits.

Combining the results of the above analysis we get :

$$\mathrm{Y}=\mathrm{Y_d Y_{in} Y_u Y_s}=\left(\sum_{i=0}^{R_d}\alpha_{i,N_d}\right)\left(\sum_{j=0}^{R_{in}}\alpha_{j,N_{in}}\right)\left(\sum_{m=0}^{R_u}\alpha_{m,N_u}\right)e^{-\lambda_{ck}}=\sum_{i=0}^{R_d}\sum_{j=0}^{R_{in}}\sum_{m=0}^{R_u}\alpha_{i,N_d}\alpha_{j,N_{in}}\alpha_{m,N_u}e^{-\lambda_{ck}}=$$

$$=\sum_{i=0}^{R_d}\sum_{j=0}^{R_{in}}\sum_{m=0}^{R_u}\sum_{s_1=0}^{i}\sum_{s_2=0}^{j}\sum_{s_3=0}^{m}\beta_{s_1q_1}\beta_{s_2q_2}\beta_{s_3q_3}e^{-\lambda_{ck}}$$

We next have to apply the compounding procedure [8, 11] in order to calculate the yield when clustering of faults is allowed. We should not however, perform seven separate compounding steps (for the six 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.

$$\lambda = \lambda_{ck} + \lambda_{tag,d} + \lambda_{tag,in} + \lambda_{tag,u} + N_1\,\lambda_{block,d} + N_2\,\lambda_{block,in} + N_3\,\lambda_{block,u}$$

We consider as compounder the Gamma distribution function with two parameters α and β :

$$f(\lambda) = \left[1/\beta^{\alpha}\,\Gamma(\alpha)\right]\lambda^{\alpha-1}\,e^{-\frac{\lambda}{\beta}}$$

After the evaluation of the integral and combining the above equations we get :

$$\mathrm{Y}=\sum_{i=0}^{R_d}\sum_{j=0}^{R_{in}}\sum_{m=0}^{R_u}\sum_{s_1=0}^{i}\sum_{s_2=0}^{j}\sum_{s_3=0}^{m}\sum_{k_1=0}^{i-s_1}\sum_{k_2=0}^{j-s_2}\sum_{k_3=0}^{m-s_3}(-1)^{k_1+k_2+k_3}\binom{N_d}{i-s_1}\binom{i-s_1}{k_1}\binom{N_{in}}{j-s_2}\binom{j-s_2}{k_2}\binom{N_u}{m-s_3}\binom{m-s_3}{k_3}\cdot$$

$$\cdot\frac{\Gamma(\alpha+s_1+s_2+s_3)}{s_1!s_2!s_3!\Gamma(\alpha)}\left(\frac{\bar{\lambda}_{tag,d}}{\alpha}\right)^{S_1}\left(\frac{\bar{\lambda}_{tag,in}}{\alpha}\right)^{S_2}\left(\frac{\bar{\lambda}_{tag,u}}{\alpha}\right)^{S_3}\cdot$$

$$\cdot\left(1+\frac{\bar{\lambda}_{tag,d}+\bar{\lambda}_{tag,in}+\bar{\lambda}_{tag,u}+\left(N_d-(i-s_1)+k_1\right)\bar{\lambda}_{block,d}+\left(N_{in}-(j-s_2)+k_2\right)\bar{\lambda}_{block,in}+\left(N_u-(m-s_3)+k_3\right)\bar{\lambda}_{block,u}+\lambda_{ck}}{\alpha}\right)^{-\alpha-S_1-S_2-S_3}$$

Note that in the above expression α is the defect clustering parameter and $\bar{\lambda}_{ck} = \mathrm{A}_{ck}D_{ck}$, $\bar{\lambda}_{tag} = \mathrm{A}_{tag}D_{tag}$ and $\bar{\lambda}_{block} = \mathrm{A}_{block}D_{data}$, where A and D stand for the area and the defect density in the corresponding parts of the chip.

For applying the faulty block disabling technique an additional bit (availability bit) should be added for each block of the cache, whose value denotes whether the corresponding block is faulty or not.

For the computation of the area occupied by the data and tag part of various cache organizations, we used the area model presented in [12]. Since this model beyond the organizational parameters of the cache (size, block size, associativity), also takes into account several physical layout information, we used the access and cycle time model for on-chip caches presented in [13], for determining the layout parameters that lead to the optimal cache cycle time in any examined case.

3. Performance Degradation

For determining the performance degradation imposed by accepting chips with partially good on-chip cache, we firstly needed to determine the miss ratios of the on-chip caches for the non-faulty and the

1063-6722/96 $10.00 © 1996 IEEE

faulty cases. To this end, we developed a cache simulator capable of handling two level of cache hierarchy as well as both inclusive and exclusive caching strategies for the second level [9]. In order to insert as little error as possible in our simulations, we use the BACH traces [15]. The trace length is adequate (well over $1,5*10^8$ references) to exercise large second level caches with low miss ratios without the risk of inserting much error due to the cold start effect during simulation [16]. The miss ratios for on-chip caches with a number of disabled faulty blocks, are computed from the non-faulty cache miss ratios and the occurrence probability of each faulty combination [4].

Once the miss ratios and the cache cycle times (by the use of the model given in [13]) are determined for a particular cache organization and assuming that the machine cycle time equals the cycle time of the first level caches the results can be easily combined into execution time by the following formula :

Execution Time = Number of instructions * cycle time of the first level cache
+ Number of hits in the second level cache * (Cycles required for the transfer of a block between second and first level caches + cycle time of the first level cache)
+ Number of Misses in the second level cache * (Cycles required for the transfer of a block between off-chip memory and second level cache + cycles required for the transfer of a block between second and first level caches + cycle time of the first level cache).

4. Discussion and Conclusions

Figure 1 presents some results of the application of the derived yield model. The fault clustering parameter was set equal to 2, in accordance with [14]. Each point in this figure has a label of the form a:b:c, where a, b and c stand for the number of faulty blocks in the first level data cache, the first level instruction cache and the second level unified cache respectively. Since we consider equal sized first level caches, pairs with values x:y:z and y:x:z lead to exactly the same result and are plotted only once. Only the combinations that offer the maximum yield enhancement and those that are significant with respect to performance degradation (as it will be discussed later) are presented.

From fig. 1 we can see that the yield can be increased significantly by accepting as good, chips with small number of faulty blocks only in the second level cache. Then from figure 2 we can see that the performance degradation is practically equal to zero. This was expected due to the large capacity and associativity of the second level cache. A further increase of the yield can be achieved by accepting as good chips with faulty blocks also in the data cache of the first level. Then we have some performance degradation (fig. 2) which becomes even larger when we accept chips with faulty blocks also in the instruction cache. However in this case the maximum yield is achieved.

References

[1] Saxena N. R., et. al., "Fault-Tolerant Features in the HaL Memory Management Unit", IEEE Trans. on Comp., Vol. 44, No. 2, pp. 170-179, Feb. 1995.

[2] Gallup M. G., et. al., "Testability Features of the 68040", in Proc. of I.T.C, Sept., 1990, pp. 749-757.

[3] Sohi G. S., "Cache Memory Organization to Enhance the Yield of High-Performance VLSI Processors", IEEE Trans. on Comp., Vol. 38, No. 4, April 1989, pp. 484 - 492.

[4] Pour A. F. and Hill M. D., "Performance Implications of Tolerating Cache Faults", IEEE Trans on Comp., Vol. 42, No. 3, Mar. 1993, pp. 257 - 267.

[5] Vergos H. T., Nikolos D., "Efficient Fault Tolerant CPU Cache Memory Design", Microprocessing and Microprogramming - The Euromicro Journal, vol. 41, pp. 153-169, May 1995.

[6] Vergos H. T., Nikolos D., "Performance Recovery in Direct-Mapped Faulty Caches via the Use of a Very Small Fully Associative Spare Cache", In Proc. Of IPDS '96, Erlangen, Germany, pp. 326-332, April 24-26, 1995.

[7] Nikolos D., et. al ., "Reconfigurable CPU Cache Memory Design : Fault Tolerance and Performance Evaluation", In Proc. of the 1st IEEE Inl On-Line Testing Workshop, Nice, France, pp. 8-10, July 4-5, 1995.

[8] Koren I. and Stapper C. H., "Yield Models for Defect-Tolerant VLSI Circuits: A review", Defect and Fault Tolerance in VLSI Systems, Vol. 1, Koren, ed., Plenum, New York, pp. 1-21, 1989.

[9] Jouppi N. P. and Wilton S. J. E., "Tradeoffs in Two-Level On-chip Caching", in Proc. of the 21st Annual Int. Symposium on Computer Architecture, Chicago IL, USA, 18-21 April 1994, pp. 34 - 45.

[10] Edmodson J. H. et. al., "Superscalar Instruction Execution in the 21164 Alpha Microprocessor", IEEE Micro, vol. 15, no. 2, pp. 33-43, April 1995.

[11] Koren I., Koren Z. and Pradhan D. K., "Designing Interconnection Buses in VLSI and WSI for Maximum Yield and Minimum Delay", IEEE J. of Solid-State Circuits, Vol. 23, No. 3, pp. 859 - 865, June 1988.

[12] Mulder J. M. et. al., "An Area Model for On-Chip Memories and its Application", IEEE JSSC, pp. 98 - 106, Feb 1991.

[13] Wilton S. J. E. and Jouppi N. P., "An Enhanced Access and Cycle Time Model for On-Chip Caches", DEC Western Research Lab, Tech Report 93/5.

[14] Leveugle R., et. al., "The Hyeti Defect Tolerant Microprocessor : A Practical Experiment and its Cost-Effectiveness Analysis", IEEE Trans. on Comp., Vol. 43, No. 12, pp. 1398 - 1406, Dec. 1994.

[15] Flanagan J. K., et. al., "BACH: BYU Address Collection Hardware, the Collection of Complete Traces", In Proc. of the 6th Int Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, pp. 128 - 137, 1992.

[16] Stone H. S., "High-Performance Computer Architecture", Addison-Wesley Publishing Company, October 1987.

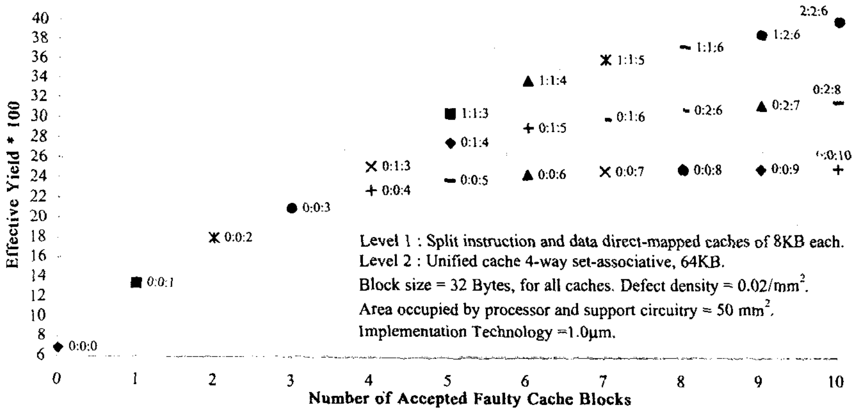

Figure 1

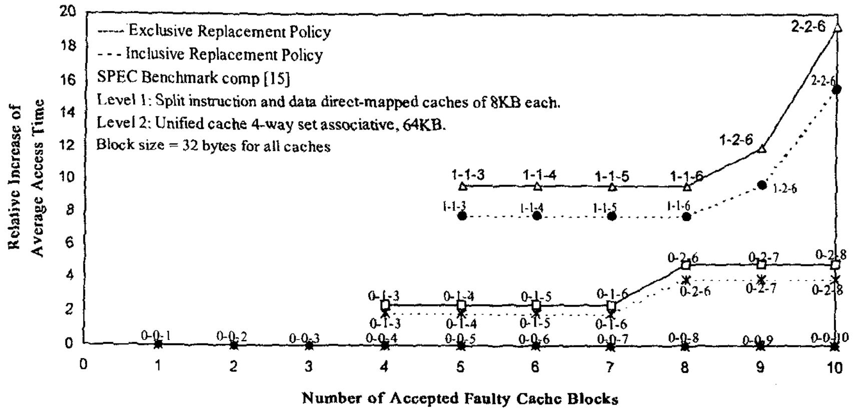

Figure 2

1063-6722/96 $10.00 © 1996 IEEE