Reconfigurable CPU Cache Memory Design : Fault Tolerance and Performance Evaluation

H.T. Vergos, D. Nikolos, P. Mitsiadis & C. Kavousianos

Computer Engineering and Informatics Department, University of Patras, 26500 Rio, Patras, Greece & Computer Technology Institute Kolokotroni 3, Patras, Greece
Phone: +30-61 99 77 52, Fax: +30-61 99 19 09,
e-mail: nikolosd@cti.gr

Abstract

The yield of VLSI processors with on-chip cache can be enhanced considerably by tolerating cache defects. It has been shown that the performance degradation due to disabling the faulty blocks is small enough for set-associative caches while in the case of direct-mapped caches may be substantial. In this paper we present a reconfigurable cache capable of operating either as direct-mapped (DM) or as two-way set-associative (TW). In this way VLSI processor chips with defective cache blocks are not discarded, attaining a yield enhancement and are also used in the operation mode that minimises the performance degradation. Trace driven simulation has been used to determine the minimum number of faulty blocks after which the TW operation mode is more profitable. This minimum value depends on cache size, block size, the access time of the cache and the miss penalty time. For computing the access time of the caches, an analytical access time model for on-chip caches already proposed in the open literature has been used.

1 INTRODUCTION

High-performance single chip VLSI processors make extensive use of on-chip cache memories to sustain the memory bandwidth demands of the CPU (*Shoemaker-1990, Intel-1992, Mirapuri-1992, Motorola-1993, Edmodson-1995*). The area that these on-chip caches occupy is a significant portion of the total chip area (*Saxena-1995*) and is expected to grow further in the future. As the chip area devoted to the on-chip cache increases, a significant portion of the manufacturing

VLSI: Integrated Systems on Silicon R. Reis & L. Claesen (Eds.)

defects will occur in the cache portion. If these defects can be tolerated without a substantial performance loss, then the yield of VLSI processors with on-chip cache can be enhanced considerably.

A technique for tolerating defects is the use of redundancy (*Moore-1986, Sohi - 1989, Pour-1993*). A form of redundancy in cache memories is the use of spare blocks. After production test the defective cache blocks are substituted by spare blocks using electrical or laser fuses. A similar technique uses spare word and / or bit lines that are selected instead of faulty ones. This technique for example is used in the caches of the MIPS R4000 processor (*Mirapuri-1992*). The above redundancy techniques impose an area overhead for accommodating the spare circuitry and the logic needed to implement the reconfiguration. Redundancy can also have the form of extra bits per word for storing an error correcting code. The classical application of a Single Error Correcting and Double Error Detecting (SEC-DED) Hamming code in the on-chip cache was investigated by Sohi (*1989*) and it was found to be a non attractive option for yield enhancement of high-performance VLSI processors. In (*Vergos-1995 a*) it was shown that defects in the tag store of a cache have more serious consequences on the integrity and performance of a system than similar defects in the data store; to this end a new SEC-DED code exploitation method was introduced. Unfortunately, this technique is only capable of masking single errors per word as for example the errors caused by a bit line defect.

Cache memory is by itself a redundant module in the sense that it is not necessary for the correct operation of the processor; it affects only the performance. Thus, a possible technique to tolerate defects in cache memories is the disabling of the faulty cache blocks. This technique was investigated in (*Sohi-1989, Pour-1993*) and it was found that the mean relative miss ratio increase due to disabling the defective cache blocks decreases with increasing cache size and is negligible for a very small number of faulty blocks unless a set is completely disabled. Unfortunately, the number of faulty blocks can be large. A very small number of random spot defects in the tag part of a cache memory can affect a large number of tags (*Vergos-1995 a*), leading to disabling of a large number of blocks. Also, because of the clustering of defects (*Koren-1989*), and the fact that the on-chip cache is a large portion of the total chip area, it is possible a large number of defects to appear in the cache while all the critical resources of the chip to be defect free. Besides the above, in direct-mapped (DM) caches a set is comprised by just one block and thus disabling a faulty block results in the disabling of a set. Therefore, in DM caches disabling even a very small number of faulty cache blocks results in substantial performance degradation. Usually the access time of the first level on-chip cache imposes the cycle time of the high-performance VLSI processors. Then taking into account that DM caches offer smaller access times that their set-associative (SA) counterparts (*Hill-1988, Wada-1992, Wilton-1994*), it is implied that in these cases the use of DM caches is advantageous. Moreover in many cases DM caches offer smaller average access times than SA ones for

sufficiently large sizes (*Hill-1988*). Thus, as the size of the on-chip caches increases, the first level on-chip caches of the future systems is expected to be DM (*Jouppi-1990*). The performance recovery in DM faulty caches via the use of a very small fully associative spare cache was investigated in (*Vergos-1995 b*). The method which will be proposed in this paper in many cases is superior than the above technique with respect to the average access time, which is a good metric of the memory hierarchy performance (*Patterson-1990*).

As we have already mentioned, the miss ratio increase of SA caches due to disabling a number of faulty blocks is smaller than the corresponding DM caches. Therefore, in the cases that DM caches have smaller average access times in their fault free operation, we expect that when the number of disabled faulty blocks exceeds a specific number, depending on the cache and block size and the miss penalty time, a SA cache will have smaller average access time. If the above reasoning is valid, then an on-chip reconfigurable cache, capable of operating either in DM or in two-way set-associative (TW) mode will be very attractive. The number of faulty blocks in the on-chip cache of a VLSI processor can be determined during testing. Thus, the chip will be used in an application with the cache operating in the mode (DM or TW) that will offer the best average access time for the given configuration of the system. In this way we succeed in two targets. VLSI processor chips with defective cache blocks are not discarded, attaining a yield enhancement and are also used in a way that the consequences of the defects in the system performance are minimised.

In this paper we give the design of a reconfigurable cache with DM and TW mode of operation. However, to show that this design is not meaningless we have to answer the following question. How much longer (if any) is the access time of a reconfigurable cache operating in DM mode compared to a non reconfigurable optimal, with respect to access time, DM cache? The answer to this question is crucial, because if the access time of a reconfigurable cache operating in DM mode is longer than that of the optimal non reconfigurable TW cache with the same cache and block sizes, then the non reconfigurable TW cache can be used more efficiently instead of the corresponding reconfigurable cache. Using a well-established analytical access time model for on-chip caches (*Wilton-1994*) we show that for all practical cases (cashes with size greater than 4 KB) the access time of the reconfigurable cache operating in DM mode is equal to that of the optimal non reconfigurable DM cache. In the rest cases the imposed delay is very small and the access time of the reconfigurable cache operating in DM mode always remains smaller than that of a non reconfigurable TW cache. Using trace driven simulation we show that when the number of disabled faulty blocks exceeds a minimum number F, the reconfigurable cache operating in TW mode provides smaller average access time compared to that of the DM mode of operation. Also, using trace driven simulation we reveal the dependence of the value of F on the cache and block sizes as well as the miss penalty time.

2 DESIGN OF A RECONFIGURABLE CACHE

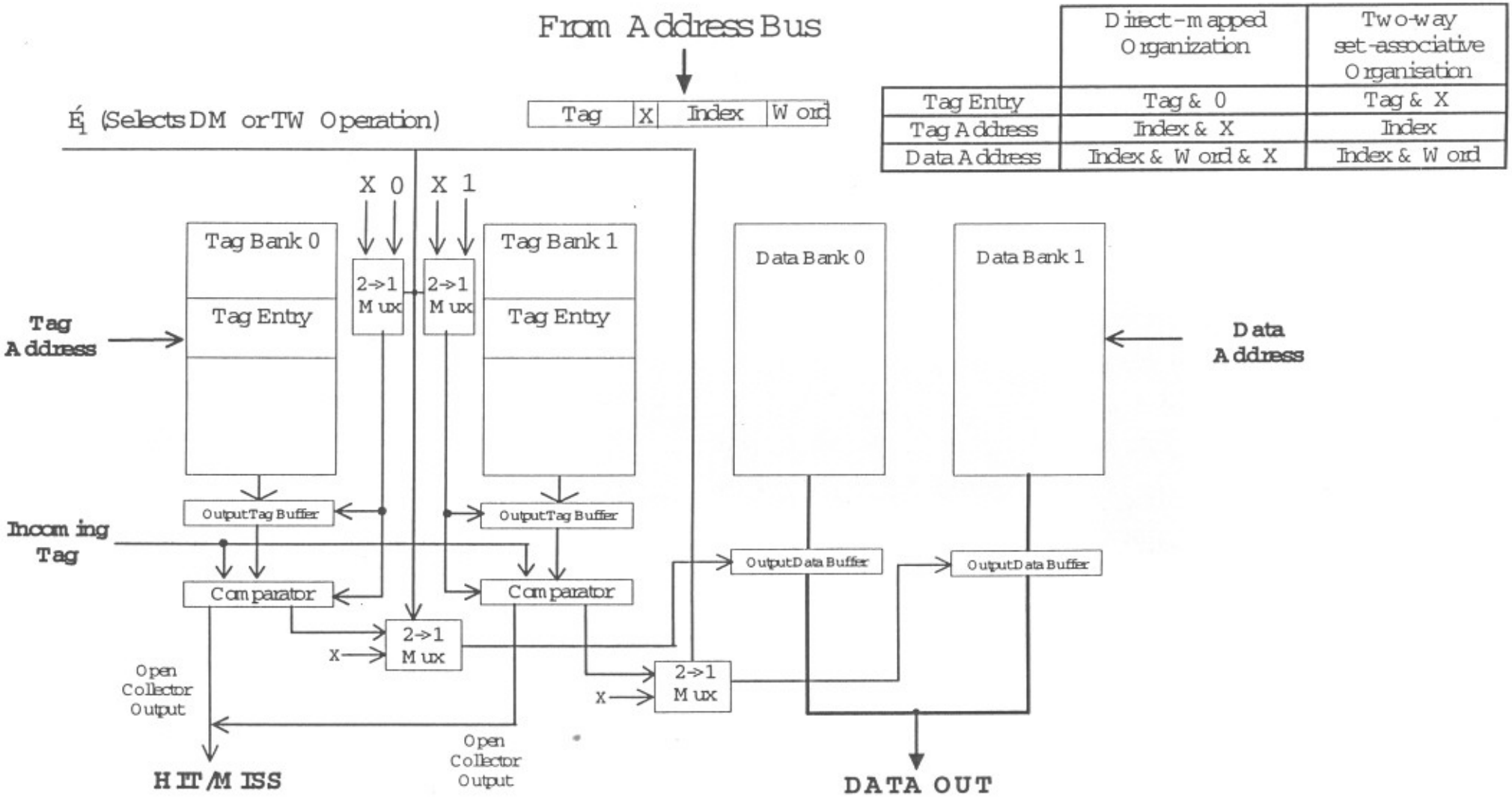
In DM caches each block of main memory can be placed in one specific frame of the cache memory while in TW caches it can be placed in any of two specific frames of the cache. The address used to access a DM or a TW cache is considered to consist of three parts <tag, index, word>. When both caches have the same size and block size the length of the word part is the same while the index part in the case of the TW cache is shorter by one bit than that of the DM cache. Of course, the opposite occurs with respect to the tag part. Figure 1 presents a block diagram of a reconfigurable cache that can operate in either DM or TW mode. The signal I_1 is used for selecting the required mode of operation and corresponds to a pin of the chip. According to the above, in a reconfigurable cache capable to operate in either DM or TW mode an address bit denoted by X in Figure 1 will be used as a part of the index when the cache operates in DM mode and as a part of the tag in the other case.

2.1 DM mode of operation.

In this mode of operation all multiplexers (MUXs) in Figure 1 permit signal X to pass at their outputs.

We will examine what happens during an access for reading. The Data Address is used to address both data banks simultaneously. Also the Tag Address is used to address both tag banks simultaneously. Although both data banks are accessed in parallel, only one of the output buffers (selected by the value of X bit) is permitted to place its contents when they become available on the Data Out bus. Also only one of the two tags (selected by the value of X bit) is routed to the selected comparator. Only the output of the selected comparator can then affect the Hit/Miss signal. The multiplexers are not on the critical path of the cache, thus no delay is imposed on the access time of the cache over the corresponding non reconfigurable DM cache's access time with the same layout.

Depending on the cache and block sizes either the Hit/Miss signal generation path or the data access path may be the critical path of the cache. The processor can use the data even before the generation of the Hit/Miss signal if this results in better access time (*Chang-1987*). In this case the processor must be equipped with rollback capabilities that are used when a miss is discovered. Similar actions must take place for writing accesses. Modification of data though can not begin before the generation of Hit/Miss signal is complete.



	Direct-mapped Organization	Two-way set-associative Organisation
Tag Entry	Tag & 0	Tag & X
Tag Address	Index & X	Index
Data Address	Index & Word & X	Index & Word

Figure 1. Block diagram of a reconfigurable cache that can operate in either DM or TW mode.

2.2 TW mode of operation.

In this mode of operation all MUXs inhibit signal X to pass at their outputs.

Again we will examine what happens during an access for reading. The Tag Address is used for addressing both tag banks in parallel. Also the Data Address is used to address both data banks. The two accessed tags are driven to the corresponding comparators for parallel comparison with the incoming tag. Although the data and tag banks are accessed in parallel, the Data Out bus can not be driven before the completion of the tag comparison procedure. Only one comparator can detect an equality (hit) and this will affect the Hit/Miss signal and will enable the corresponding Output Data Buffer to place its contents on the Data Out bus. We can see that in this case the MUX can be on the critical path, thus the access time of the reconfigurable cache operating in the TW mode may be longer than the access time of a non reconfigurable TW cache with the same layout by at most the delay of a MUX. The MUX will be on the critical path of the cache, if:

$$t_{tag} + t_{comp} + t_{mux} \geq t_{data},$$

where t_{tag} and t_{data} are the access time of the Tag and Data Banks respectively, and t_{comp}, t_{mux} are the delays of the comparator and the multiplexer respectively. In this case the imposed delay on the access time of the cache will be equal to:

$$\min \{ t_{mux}, t_{tag} + t_{comp} + t_{mux} - t_{data} \}.$$

SA caches that allow the required word of the most recently used block in the selected set to appear on the data out lines before the tag comparison is complete (*Chang-1987*) is out of the scope of this paper. Updating can be done in parallel with sending the required data to the microprocessor and no additional delay is imposed on the access time of the cache over that of the corresponding non reconfigurable TW cache with the same layout.

2.3 Time and hardware overhead of the reconfigurable cache.

In this section we will investigate the time and hardware overhead of the reconfigurable caches with respect to the corresponding non reconfigurable DM caches. We will firstly consider the cache access time overheads.

Analytical models are the best way to evaluate trade-offs among various alternatives without the cost of implementing each different alternative. To the best of our knowledge, two analytical models (*Wada-1992, Wilton-1994*) have been proposed for computing the access time of an on-chip cache that take into account the layout parameters along with the organisational parameters (cache size, associativity and block size). Both models have been validated by comparison with a Spice model. The model presented in (*Wilton-1994*) is more accurate, since it extends the work of (*Wada-1992*) by the inclusion of an additional array organisational parameter, improved decoder and word line models, precharged and column multiplexed bit lines and a tag array model with

comparator and multiplexer drivers. Thus, the model presented in (*Wilton-1994*) is used in this work.

Table 1 The number of segments per data and tag bit lines in the optimal DM caches

	Block=8 Bytes		*Block=16 Bytes*		*Block=32 Bytes*	
Cache Size	N_{dbl}	N_{tbl}	N_{dbl}	N_{tbl}	N_{dbl}	N_{tbl}
8 KB	2	2	2	2	2	2
16 KB	2	2	2	2	2	2
32 KB	4	4	2	2	2	2
64 KB	4	4	4	4	4	2
128 KB	8	4	4	4	4	4
256 KB	8	8	8	4	4	4

As can be seen from Figure 1 each of the tag and data stores of the proposed reconfigurable cache must consist of at least two banks (two segments per tag and data bit line). Table 1 presents the number of segments per tag bit line (N_{tbl}) and per data bit line (N_{dbl}) of an optimal, with respect to access time, DM cache, according to the analytical time model presented in (*Wilton-1994*). Cache sizes up to 256 KB and block sizes of 8, 16 and 32 bytes are considered.

From Table 1 we can see that for cache sizes greater than or equal to 8 KB the optimal non reconfigurable DM cache consists of two or more segments per tag and data bit line. In these cases the access time of the reconfigurable cache operating in DM mode is the same with the access time of the corresponding optimal non reconfigurable DM cache. Then taking into account that modern single chip VLSI processors usually offer on-chip caches greater than or equal to 8 KB we conclude that for cache sizes of practical interest the access time of the reconfigurable cache operating in DM mode is the same with the access time of the corresponding non reconfigurable DM cache.

Using the analytical time model (*Wilton-1994*) we verified that the access time of the large as well as the small reconfigurable caches operating in DM mode is shorter than that of the optimal TW caches with the same capacity and block size.

The hardware overhead of the reconfigurable cache with respect to a non reconfigurable DM cache with the same cache and block size consists of:

a) Four 2->1 MUXs as shown in Figure 1.

b) An extra tag comparator.
c) Three extra bits per tag word. These bits are used as follows:

- The disabling of the faulty blocks can be achieved by the addition of a second valid (availability) bit. The availability bits can be implemented by either non volatile memory cells or by normal static RAM. In the latter case the silicon area required is smaller than that needed by the non volatile memory implementation. However, each time the system is powered up, the availability information must be loaded into the availability bits from a safe copy.
- As we have previously discussed, the tag of the reconfigurable cache when operating in TW mode is by one bit longer than the tag in the DM mode of operation. Hence, an extra tag bit is required for storing the increased tag. This bit is not used in the DM mode of operation.
- A bit per tag word is used for the implementation of Least Recently Used (LRU) policy in the TW mode of operation.

To get an estimation of the above overhead in area we used the area model presented in (*Mulder-1991*). Applying the model, we got that the reconfigurable 4 KB and 32 KB caches with 16 bytes blocks, require only 1.87% and 1.88% respectively more area than the optimal non reconfigurable DM corresponding caches. It is obvious that the area overhead is very small. (In these calculations we considered that the availability bits are implemented by static RAM).

3 EVALUATION OF THE RECONFIGURABLE CACHE

In order to compare the alternative cache configurations the average memory access time will be used, which is a good metric of memory hierarchy performance (*Patterson-1990*). For the average access time we have:

$$T = t_{cache} + m\ TM,$$

where t_{cache} is the access time of the cache and TM is the miss penalty time, while m is the miss ratio of the cache. To this end, we need to determine the miss ratios when either none or some of the cache's blocks are disabled and the access time of each cache configuration.

3.1 Miss Ratio Determination and Access Time Calculation

Trace driven simulation is the best way for determining the miss ratio of a cache when no faults have occurred (*Smith-1982*). For our simulations we used the ATUM traces because they include both operating system references and multiprogramming effects and the way that these traces were gathered introduces minor errors (*Agarwal-1986*). Due to the large number of traces, we present results only for one combined trace described as *all* in (*Pour-1993*). Table II in (*Pour-1993*) lists the number of instruction fetches, data reads and data writes for each individual trace used as well as a brief description of their origins. The *all* trace was formed by concatenating the individual traces with cache flushes inserted

between them. Since each individual trace is only about 400000 references long, we simulate cache sizes up to 32 KB. Larger cache simulation would be impossible without inserting much error (*Pour-1993, Stone-1987*).

Two alternatives can be used for determining a cache's miss ratio when some blocks have been disabled due to faults, namely trace driven simulation for each possible or for a number of the possible faulty combinations (*Sohi-1989*) and probabilistic theory based on least recently used distances (*Pour-1993*). The second alternative was chosen in this work since it provides accurate results and requires less time. According to this approach the mean miss ratios of caches with a number of disabled faulty blocks can be computed from the miss ratios of the non faulty caches and the occurrence probability of each faulty combination.

To calculate the cache access time, we used the time model presented in (*Wilton-1994*) as already mentioned. The delay of a 2->1 MUX in an implementation technology with minimum feature size of 0.8 micron was considered to be equal to 0.35 ns.

Table 2 Number of faulty blocks F after which the TW mode of operation offers shorter average access time. Block size = 16 Bytes.

	Cache Size (Bytes)							
Miss penalty time	*256*	*512*	*1K*	*2K*	*4K*	*8K*	*16K*	*32K*
25	3	5	10	19	39	55	142	280
50	0	1	3	6	14	20	56	116
75	0	0	0	3	7	9	31	68
100	0	0	0	0	4	4	19	45
125	0	0	0	0	0	0	12	31
150	0	0	0	0	0	0	0	22
175	0	0	0	0	0	0	0	16
200	0	0	0	0	0	0	0	11

3.3 Results.

Tables 2 and 3 give the value of F for cache sizes from 256 bytes up to 32 KB and for block sizes 16 and 32 bytes respectively. TM is varied from 25 up to 200 ns. A value of F equal to zero means that a reconfigurable cache operating in TW mode always offers better average access time. The conclusions that can be drawn from these tables are:

1. For constant values of cache and block sizes, increasing TM decreases F.
2. For constant values of block size and TM, increasing cache size increases F. This is because a larger cache contains more blocks than any smaller with the

Table 3 Number of faulty blocks F after which the TW mode of operation offers shorter average access time. Block size = 32 Bytes.

	Cache Size (Bytes)							
Miss penalty time	*256*	*512*	*1K*	*2K*	*4K*	*8K*	*16K*	*32K*
25	2	3	6	12	22	45	73	190
50	0	1	2	4	8	17	30	79
75	0	0	0	2	5	9	17	48
100	0	0	0	1	3	6	11	33
125	0	0	0	0	2	4	7	24
150	0	0	0	0	0	2	5	19
175	0	0	0	0	0	0	3	14
200	0	0	0	0	0	0	2	11

same block size. The hit ratio deterioration caused by faulty block disabling depends on the percentage of the disabled faulty blocks out of the total. If the same number of faulty blocks is disabled, in a smaller cache a greater percentage of the total cache capacity gets disabled and the hit ratio deterioration is larger.

3. The dependence of F on the block size can be explained similarly. Since a larger block size for the same cache size, means a smaller number of blocks, it is expected that the value of F will drop upon moving to larger block sizes and verified by Tables 2, 3. Disabling of a larger block means that a greater percentage of the total cache capacity gets disabled.

In (*Vergos-1995 b*) the performance recovery in DM faulty caches via the use of a very small fully associative spare cache was investigated. In many cases the reconfigurable cache design proposed in this paper offers better average access time than the use of the method proposed in (*Vergos-1995 b*).

4 CONCLUSIONS

To achieve high performance in single chip VLSI processors, on-chip cache memories are used. With the increase of the chip area devoted to on-chip caches, it is expected a substantial portion of the manufacturing defects to occur in the cache portion of the VLSI processor chip. If the cache defects are tolerated without a noticeable performance degradation, the yield of VLSI processors can be enhanced considerably. It has been shown that the performance degradation due to disabling the faulty blocks is small enough for SA cache while in the case of DM

caches can be substantial. However, DM caches are the fastest and also offer smaller average access time than SA ones for sufficiently large size. Thus, as the size of the on-chip cache increases the use of direct-mapped caches is favoured.

In this paper we have designed a reconfigurable cache capable of operating in either DM or TW mode. The proposed design offers the ability VLSI processor chips with a partially good reconfigurable cache to be used in the operation mode (DM or TW) that minimises the average access time increase due to faulty block disabling. Since the use of the chips with partially good reconfigurable caches implies a very small performance degradation, we believe that these chips can be accepted during production testing leading to a significant yield enhancement. Apart from the production testing faulty cache block disabling can also take place during the on field testing. Then, if the total number of disabled faulty blocks (caused by manufacturing defects and permanent operational faults) exceeds F, the operation mode of the reconfigurable cache can be switched from DM to TW. A significant feature of the reconfigurable cache is that the access and average access time of it operating in DM mode for cache sizes of practical interest (specifically sizes greater than 4 KB) is the same with that of the corresponding optimal DM cache. Therefore, VLSI processor chips with a fault free reconfigurable cache will operate equally fast with the chips with optimal DM cache. Also, the area overhead of the reconfigurable cache with respect to the corresponding optimal DM was estimated to be very small, about 1.88%. In this paper we have also investigated the dependence on cache and block size and on miss penalty of the minimum number F of faulty cache blocks after which the TW operation mode of the cache offers a shorter average access time.

5 REFERENCES

Agarwal, A. et. al. (1986) ATUM: A New Technique for Capturing Address Traces Using Microcode, *Proc. of the 13th Annual Symposium on Computer Architecture*, 119-129.

Chang, J. H., et. al. (1987) Cache Design of a Sub-Micron CMOS System/370, *Proc. of the 14th Annual Symposium on Computer Architecture*, 208-213.

Edmodson, J. H. et. al. (1995) Superscalar Instruction Execution in the 21164 Alpha Microprocessor, *IEEE Micro*, 33-43, April 1995.

Hill, M. D. (1988) A Case for Direct-Mapped Caches, *IEEE Micro*, 25-40.

Intel (1992) i860 XP Microprocessor. Multimedia and Supercomputing Microprocessors Data Book.

Jouppi, N. P. (1990) Improving Direct-mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers, *Proc. of 17th Annual Symposium on Computer Architecture*, 364-373.

Koren, I. and Stapper C. H. (1989) Koren ed., Yield Models for Defect-Tolerant VLSI Circuits: A review, Defect and Fault Tolerance in VLSI Systems, vol. 1, Plenum, New York, 1-21.

Mirapuri, S. et al. (1992) The MIPS R4000 Processor, *IEEE Micro*, **4**, 10-22.

Moore, W. R. (1986) A Review of Fault-Tolerant Techniques for the Enhancement of Integrated Circuit Yield, *Proc. of the IEEE*, **4**, 684-698.

Motorola (1993) PowerPC 601-RISC Microprocessor User's Manual, Motorola Semiconductor Technical Data Book.

Mulder, J. M. Quach, N. T. and Flynn M. J. (1991) An Area Model for On-Chip Memories and its Application, *IEEE JSSC*, **2**, 98-106.

Patterson, D. A. and Henessy, J. L. (1990) Morgan Kauffman Publishers, Computer Architecture: A Quantitative Approach, San Mateo, California.

Pour, F. and Hill M. D. (1993) Performance Implications of Tolerating Cache Faults, *IEEE Trans. on Computers*, **4**, 257-267.

Saxena, N. R. et. al. (1995) Fault-Tolerant Features in the HaL Memory Management Unit, *IEEE Trans. on Computers*, **2**, 170 - 179.

Shoemaker, K. (1990) The i486 Microprocessor Integrated Cache and Bus Interface. *Proc. of COMPCON '90 IEEE International Conference*, 248-253.

Smith, A. J. (1982) Cache Memories, *ACM Computing Surveys*, **3**, 473-530.

Sohi, G. (1989) Cache Memory Organisation to Enhance the Yield of High-Performance VLSI Processors, *IEEE Trans. on Computers*, **4**, 484-492.

Stone, H. S. (1987) Addison - Wesley Publishing Company, High-Performance Computer Architecture.

Vergos, H. and Nikolos, D. (1995 a) Efficient Fault Tolerant CPU Cache Memory Design, *Micropr. & Microprogram - The Euromicro Journal*, **41**, 153-169.

Vergos H. T., Nikolos D. (1995 b), Performance Recovery in Direct-Mapped Faulty Caches via the Use of a Very Small Fully Associative Spare Cache, *Proc. of IEEE IPDS'95, Erlangen, Germany,* 326-332.

Wada, T. Rajan, S. Przybylski, S. A. (1992) An Analytical Access Time Model for On-Chip Cache Memories, *IEEE JSSC*, **8**, 1147-1156.

Wilton, S. J. E. and Jouppi, N. R (1994), An Enhanced Access and Cycle Time Model for On-Chip Caches, *DEC Western Research Lab, T.R. 93/5.*

6 BIOGRAPHIES

Haridimos T. Vergos received the diploma and the PhD degree from the Comp. Engineering Dept. of University of Patras in 1991 and 1996 respectively.

Dimitris Nikolos received the BSc degree in Physics in 1979, the MSc degree in electronics in 1981 and the PhD degree in Computer Science in 1985, all from the University of Athens, Greece. He is currently Associate Professor in the Dept. of Comp. Engineering in the Univ. of Patras and Director of the Hardware and Comp. Architecture Division.

Petros Mitsiadis received his diploma from Comp. Engineering Dept of Univ. of Patras, Greece, in 1996.

Chrisovalantis Kavousianos after receiving his diploma in Comp. Engineering in 1996, is currently pursuing his PhD in the same Department