

RVC-Based Time-Predictable Faulty Caches for Safety-Critical Systems

Jaume Abella¹, Eduardo Quiñones¹, Francisco J. Cazorla^{1,2},
Mateo Valero^{1,3}, Yanos Sazeides⁴

¹Barcelona Supercomputing Center (BSC)

²Spanish National Research Council (IIIA-CSIC)

³Universitat Politècnica de Catalunya (UPC)

⁴University of Cyprus

jaume.abella@bsc.es, eduardo.quinones@bsc.es, francisco.cazorla@bsc.es,
mateo.valero@bsc.es, yanos@cs.ucy.ac.cy

Abstract—Technology and V_{cc} scaling lead to significant faulty bit rates in caches. Mechanisms based on disabling faulty parts show to be effective for average performance but are unacceptable in safety critical systems where worst-case execution time (WCET) estimations must be safe and tight. The Reliable Victim Cache (RVC) deals with this issue for a large fraction of the cache bits. However, replacement bits are not protected, thus keeping the probability of failure still high.

This paper proposes two mechanisms to tolerate faulty bits in replacement bits and keep time-predictability by extending the RVC. Our solutions offer different tradeoffs between cost and complexity. In particular, the Extended RVC (ERVC) has low energy and area overheads while keeping complexity at a minimum. The Reliable Replacement Bits (RRB) solution has even lower overheads at the expense of some more wiring complexity.

I. INTRODUCTION

Modern technology nodes experience increasing timing variations due to an increasing relative impact of process variations and some degree of voltage scaling (V_{cc}). Testing and burn-in effectiveness decreases generation after generation due to (i) an increasing number of transistors to test (roughly 2X transistors in every new generation), (ii) a limited amount of time and power budget to perform tests and burn-in, and (iii) the faster degradation of smaller devices, whose fault-free lifetime decreases during burn-in. Overall, a larger number of actual and latent defects reaches products.

Timing variations can be tolerated to some extent in combinational circuits by allocating larger guardbands in the cycle time. However, SRAM cells in memory arrays such as cache memories and scratch-pads are particularly error-prone at low V_{cc} operation since process variations and noise jeopardize their data retention capabilities [1]. Faults in those memory arrays are particularly important given the large fraction of the chip area devoted to them. Cost to manage data allocation in scratch-pads and to fabricate specific chips for the safety-critical market lead

to the use of cache memories that are the main focus of this work.

General-purpose systems can tolerate some faulty bits in cache memories as long as they are detected and faulty parts deactivated [2]–[5], because the average impact in performance is low. However, safety-critical systems (e.g., flight management systems in avionics, brake-by-wire in automotive, etc.) rely on safe worst-case execution time (WCET) estimations and fault-free hardware¹. Thus, simply deactivating faulty cache storage is not an option given its either unknown or unacceptable WCET impact. For details on WCET estimation methods we refer the reader to [6].

Our recent work, the reliable victim cache (RVC), has shown that cache-assistant structures such as the eviction buffer, the write-combining buffer, the victim cache and the like provide cheap sparing for faulty cache lines (with lower cost than conventional sparing) and can be extended to keep time-predictability in the presence of faults [7]. RVC has been shown to be highly efficient for faults affecting bits in the cache lines; however, those bits associated to the cache sets such as replacement information have been neglected.

This paper extends the RVC with solutions to keep time-predictability in the presence of faults in replacement bits. Different alternatives are considered trading off between efficiency and complexity. Overall, by extending the RVC with our solutions full protection of cache memories is achieved to provide time-predictable (and therefore, analyzable) faulty caches for processors in safety-critical systems.

The rest of the paper is organized as follows. Section II describes the RVC. Section III presents our solutions to deal with permanent faults in set-dependent bits. Section IV presents results on probabilities of failure,

¹Although fault-free hardware does not exist, we can consider that hardware is fault-free if its probability of failure is below an acceptable threshold (e.g., 10^{-6}) so that failures can be simply neglected

and energy and area overheads. Section V discusses related work. Finally, Section VI concludes this paper.

II. OVERVIEW OF THE RVC

Cache memories consist of several SRAM arrays. Those arrays store line-dependent bits (data, tags, valid bits, dirty bits, write-permission bits, coherence state), etc. and set-dependent bits (replacement information). Most of the bits are associated to particular cache lines. The RVC has been shown to be effective to deal with permanent faults in all those line-dependent bits. The only line-dependent bit not covered by the RVC is the one indicating whether the cache line is faulty. As stated in [7], such bit needs specific protection mechanisms based on either spatial redundancy or cell hardening, which has significant relative area cost for such bit, but low absolute area cost for the whole cache. Keep in mind that each cache line may have several hundreds of bits, so even using hardened cells [8] would increase cache area largely below 1%.

The RVC is based on extending cache-assistant structures to provide cheap sparing. Those structures are extended with spare entries and two extra fields indicating whether a particular entry is being used for replacement of a faulty cache line (*lock bit*) and, if so, which cache line it replaces (*physical tag*). Such information is enough to allow programs to have a completely time-predictable and analyzable behavior. This is achieved by guaranteeing that all accesses that would hit in cache in a fault-free system will also hit in a faulty cache system with the RVC in place. Further, hit latency either remains unaffected or it is increased by a given (and fixed) latency for all accesses. Such information is available during the estimation of the WCET regardless of the number and location of faults.

Note that failing to provide safe and tight WCET estimates would have unaffordable costs for safety-critical systems. As shown in [7], just a single fault is enough to make all cache hits become misses in the worst-case scenario. Thus, the only way to keep the required safety levels is assuming miss latency for all cache accesses. However, such an assumption produces unacceptably high WCET estimates that make hardware designs extremely expensive. For instance, if WCET grows by 2X, we should set up a system 2X faster or 2 replicas of the original system to provide the required performance.

Figure 1 shows the schematic of a 2-way L1 data cache and a 6-entry fully-associative RVC. Victim caches (the RVC in our case) can be accessed either (i) in parallel with L1 caches, thus serving data with the same latency as L1 caches, or (ii) sequentially, but only in case of an L1 miss, with a total latency higher than that of an

L1 hit. However, latency of hits in a fault-free cache can be bounded to the total latency of the L1 cache and the RVC. The following scenarios illustrate how the RVC works:

- **Miss in a cache with RVC.** Both the cache and the RVC are looked up. Both of them report a miss and data are requested to the next cache level. Whenever data arrive they are allocated in cache if the corresponding cache line is fault-free. Otherwise, they are allocated in the RVC entry remapping the faulty cache line. Replacement information in cache is updated as if the cache line was fault-free.
- **Hit in a cache with RVC.** If the hit cache line is fault-free, everything works as in a fault-free system. Otherwise, the hit will happen in a locked entry of the RVC. Such entry will provide the data and will use the physical tag of the faulty cache line stored in the RVC to update the proper entry of the replacement information array.

There are some other corner cases that must be considered, such as how to deal with those accesses that would hit in the victim cache of a fault-free system, or how unlocked RVC lines are managed when a new line is fetched into the RVC. We refer the reader to the original paper for further details [7].

As explained before, the RVC provides cheap sparing and time-predictability for any fault affecting line-dependent bits. Set-dependent bits need further solutions. Almost fault-free replacement bits can be achieved by means of error correcting codes (ECC) [9] or cell hardening. However, ECC would impact latency because correction and encoding of the replacement bits should be performed on every access, therefore impacting the timing of the cache, potentially introducing stalls in consecutive accesses to the same set and increasing power dissipation to decode and encode replacement bits. Instead, hardening does not introduce significant timing degradation, but still introduces some power and area overheads. However, our experiments show that those overheads are lower than those of ECC. In this paper we propose different effective and scalable solutions to provide fault tolerance for those set-dependent bits such as replacement bits of a least recently used (LRU) cache.

III. EXTENDING THE RVC CONCEPT FOR SET-DEPENDENT BITS

This section presents new solutions to provide cheap sparing and time-predictability for set-dependent bits (mainly replacement bits), thus extending the RVC protection for all cache arrays. Two main choices are presented: (i) a separate structure for replacement storage sparing and (ii) a solution where those bits are integrated into the RVC. For the sake of commodity we assume

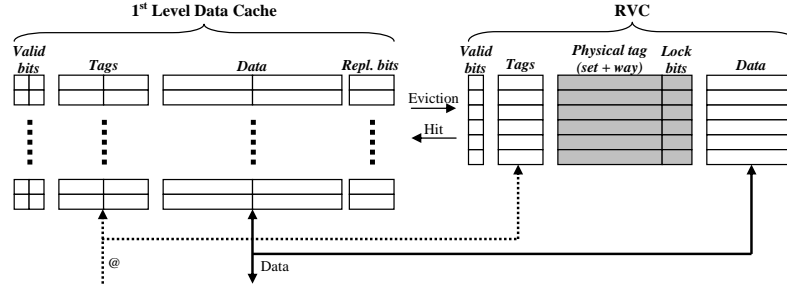


Fig. 1. Example of a 2-way 1st level data cache (DL1) with a 6-entry RVC. Grey fields correspond to the extra information needed by the RVC to provide time-predictability

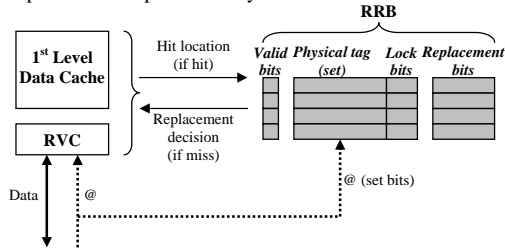


Fig. 2. Example of a 4-entry RRB connected to a cache extended with an RVC

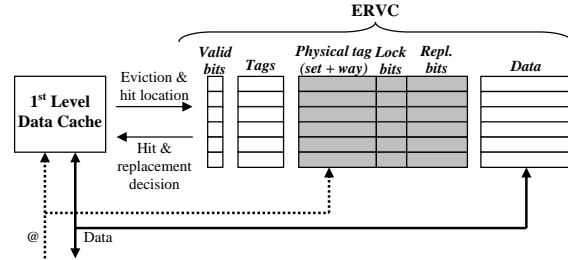


Fig. 3. Example of a 6-entry ERVC connected to a cache

LRU replacement policy, which is the most commonly used one for safety-critical systems although our solutions can be applied to any replacement policy.

A. RRB: Separate Set-Dependent Sparing

The first alternative requires setting up a new structure for cheap sparing of replacement bits in cache. We refer to this structure as RRB (Reliable Replacement Bits). Following the same philosophy as the the RVC, RRB entries have a physical tag indicating which cache set they are replacing (if any), a lock bit to indicate whether they are being used and the set of replacement bits. The RRB is depicted in Figure 2.

Whenever the cache is accessed, the set identifier is looked up in the RRB for a match in a locked entry.

- If no hit occurs, then the replacement bits of the cache set are fault-free.
- Otherwise, those bits are faulty in the cache and the entry hit in the RRB holds the required replacement bits.

(a) In case of a miss in cache the corresponding RRB entry overrides the replacement decision of the cache since replacement bits in the corresponding cache set are faulty. Note that such override does not produce any issue in terms of timing because data will take some cycles to reach the cache.

(b) If the access hits in cache the way within the set of the hit cache line must be propagated to the RRB to update the replacement information accordingly. Note that such propagation may delay the update of the RRB. However, since such delay is paid for all hits, updates will happen in the

same order, thus keeping predictability. Updates and replacement decisions in case of a miss can be also delayed in the same way to guarantee correct operation. Such delay can be afforded because it will be in the order of 1 or 2 cycles, which is largely below the turnaround latency for the following cache level (or memory) in the memory hierarchy.

The RRB is very effective in terms of storage space required since it adds only sparing for faulty replacement bits. Given that the fraction of bits devoted to store replacement information in cache is relatively low, the fraction of faults they may suffer is smaller than for data and tags. Thus, the number of entries required in the RRB is really small (smaller than for the RVC). As shown later, for faulty bit rates requiring 4 extra entries in the RVC an RRB with just 2 entries suffices to keep reliability level high. However, some extra signal routing is required from the cache and the RVC to the RRB.

B. ERVC: RVC-Integrated Set-Dependent Sparing

A different choice to reduce the wiring overhead to route signals from the cache and the RVC to another structure (e.g., the RRB) consists of integrating replacement information sparing in the RVC. In order to do so, replacement bits must be virtually attached to a particular cache line in their cache set (i.e., the first cache line). Whenever the corresponding cache line in the set or the replacement bits in the set are faulty, both are considered faulty and remapped to an RVC entry. An schematic of the extended RVC (ERVC) is depicted in Figure 3. As shown, each ERVC entry is extended with replacement bits. No bit is required to indicate whether

those replacement bits must be used. Such information is implicit in the way identifier of the physical tag (e.g., if the line is the first cache line in the set).

As expected, the operation of the ERVC differs slightly from that of the RVC for the replacement bits management. For the sake of clarity we assume that the replacement bits are virtually attached to the first line in their cache set. The ERVC will provide two match signals: (i) tag match (as in the RVC) and (ii) partial match if a hit occurs only in the bits indicating the set of an entry corresponding to the first way in the set.

- If an access occurs in a set whose first cache line is fault-free, the operation is exactly the same as for the RVC. In case of accessing the ERVC, either because a faulty cache line is hit or because a faulty cache line is being replaced, the replacement bits in the corresponding ERVC entry are simply ignored (no partial match occurs).
- If the first cache line of the cache set accessed is faulty the replacement bits are in the ERVC (there will be a partial match).
 - (a) In case of a hit (in the cache or the ERVC), data will be accessed as usual either in the cache or in the ERVC. However, since the first line in the cache set is faulty there will be a partial match in one of the ERVC entries. Note that up to two different ERVC lines can be accessed. For instance, the access may hit in the second line of a cache set whose first and second lines are faulty. The second line will observe a tag match only and its data will be accessed. The first line will observe a partial match only so its replacement bits will be updated accordingly.
 - (b) In case of a miss, the cache and the ERVC will be accessed and will detect the miss. The ERVC will observe a partial match in an entry whose replacement bits indicate which cache line (or ERVC entry if the line is faulty) must be replaced.

In essence, the ERVC combines the RVC and the RRB into a single structure. Thus, wiring overhead is reduced. However, it introduces some inefficiency because all ERVC entries will have space to store replacement bits. Although the ERVC is small, the number of extra bits is larger than that of the RRB. Similarly to state-of-the-art approaches for fault tolerance [4], [5], [7] we assume that there are mechanisms in place to detect faulty storage and configure the extra fields of the RRB or ERVC properly.

IV. EVALUATION

This section presents the evaluation framework and results in terms of probability of failure, power and area for the RRB and ERVC designs when the L1 data cache

Configuration	Faulty bits allowed	
	Non repl bits	Repl bits
VC=4	0	0
RVC=6 (repl non-prot)	2	0
RVC=6 (repl perf)	2	—
ERVC=6	2	2
RVC=6 & RRB=1	2	1
RVC=8 (repl non-prot)	4	0
RVC=8 (repl perf)	4	—
ERVC=8	4	4
RVC=8 & RRB=2	4	2

TABLE I
CONFIGURATIONS CONSIDERED

is faulty. Results are compared against different baseline designs with and without the RVC. WCET remains the same as for the RVC scheme, so it is omitted.

A. Probability of Failure

We have studied the probability of failure (*pfail*) for different random faulty bit rates between 0.01% and 0.000001%. Results have been collected for those configurations shown in Table I. All of them include a 16KB 4-way 32B/line LRU-replacement cache. VC stands for a conventional victim cache. Two versions of the RVC have been considered: (i) with non-protected replacement bits (*repl non-prot*) and (ii) with fault-free (hardened) replacement bits (*repl perf*)². 8 replacement bits per set have been considered: 2 bits to identify each of the 4 ways.

A failure is assumed whenever there are more blocks with faults or more sets with faulty replacement bits than can be repaired. Roughly speaking VC tolerates no faults; RVC (*repl perf*) and ERVC tolerate as many faults as extra entries they have; and RVC & RRB tolerates as many faults in line-dependent (set-dependent) bits as extra RVC (RRB) entries. RVC (*repl non-prot*) tolerates faults anywhere but in the replacement bits, where no faulty bits are allowed.

Figure 4 depicts the *pfail* obtained using binomial probability analysis for all configurations. As shown, the *pfail* for both the ERVC and RRB configurations are very similar to their RVC counterpart configuration with fault-free replacement bits. This is so because the extra number of bits required by the ERVC and RRB with respect to the RVC design is low. For instance, ERVC=8 and RVC=8 & RRB=2 increase bit count by 0.1% and 0.05% respectively with respect to RVC=8. The relative growth of the *pfail* for the ERVC and the RRB with respect to the RVC is below 0.5% and 0.3% respectively in all cases. If replacement bits are not protected their *pfail* becomes much more significant than that of data and tags despite having RVC in place.

²ECC introduces significant overheads to protect few bits (larger than those of hardening), so we use hardening as a baseline for fault-free bits

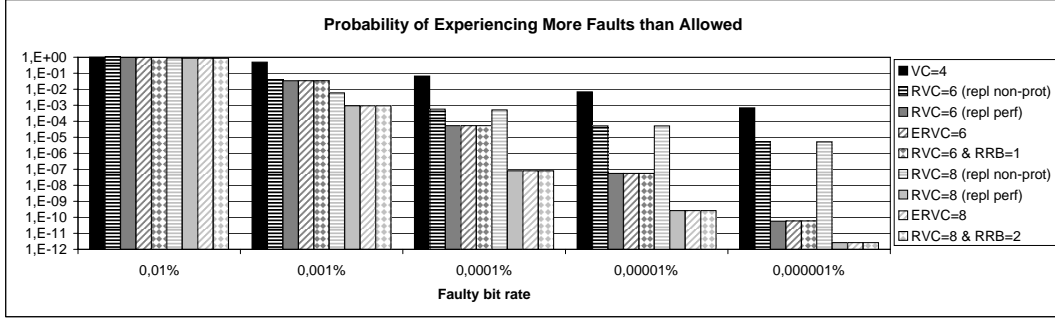


Fig. 4. Probability of failure for baseline, RVC, ERVC and RRB designs under different faulty bit rates. (logarithmic scale)

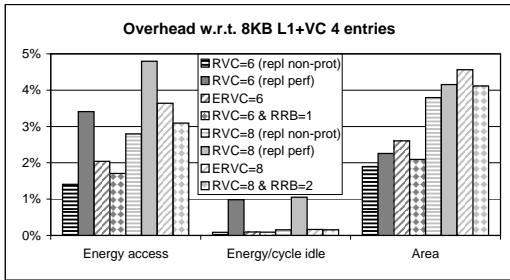


Fig. 5. Relative energy and area overhead for the different schemes with respect to a 8KB 4-way 32B/line cache with a 4-entry VC

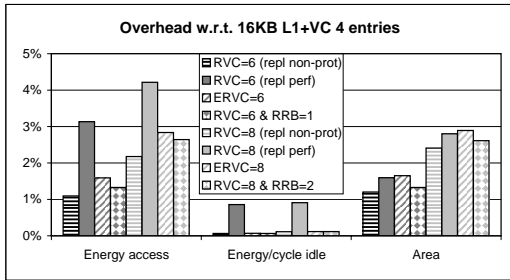


Fig. 6. Relative energy and area overhead for the different schemes with respect to a 16KB 4-way 32B/line cache with a 4-entry VC

We can observe that RVC, ERVC and RRB designs provide p_{fail} several orders of magnitude lower than those of the baseline cache. For instance, $RVC=6$ (*repl perf*), $ERVC=6$ and $RVC=6 \& RRB=1$ decrease the p_{fail} by 3 orders of magnitude for a faulty bit rate of 0.0001% (1 faulty bit every 1,000,000 bits). Similarly, $RVC=8$ (*repl perf*), $ERVC=8$ and $RVC=8 \& RRB=2$ decrease the p_{fail} by 6 orders of magnitude for the same faulty bit rate. Conversely, $RVC=6$ (*repl non-prot*) and $RVC=8$ (*repl non-prot*) only provide 2 orders of magnitude lower p_{fail} because around 1% of the bits of the cache are replacement bits and they are non-protected.

B. Energy and Area

Power and area results have been obtained by means of the CACTI 6.5 tool [10]. CACTI is a delay, power and area model for cache memories developed at the HP Labs. Results have been gathered for 32nm process

technology, assuming low power design style, 330K temperature and 0.6V operating voltage. We have validated that observed trends hold for other technology nodes, design styles, and temperature and voltage values.

We have studied the impact in energy (both dynamic and leakage) and area of the different schemes. In particular, we have measured those overheads with respect to a baseline scheme as the one described in the previous subsection (a 4-way cache and a 4-entry VC). The schemes studied are as follows:

- RVC (*repl non-prot*). Overheads come from the extra fields and entries in the VC required to implement the RVC.
- RVC (*repl perf*). It has the same overheads as RVC (*repl non-prot*) plus the overheads coming from using hardened SRAM cells for the replacement bits in cache (10-T SRAM cells [8]). 10-T cells have been included in CACTI adapting cell capacitances, resistances and geometry.
- $ERVC$. It has the same overheads as RVC (*repl non-prot*) plus the extra fields in the RVC.
- $RVC \& RRB$. It has the same overheads as RVC (*repl non-prot*) plus the RRB overheads.

Results are shown in Figure 5 and Figure 6 for baseline 8KB and 16KB caches respectively. We can observe that the lowest overheads are those for RVC (*repl non-prot*). However, as shown before, its probability of failure is too high. The straightforward solution based on hardening replacement bits (RVC (*repl perf*)) is the most expensive in terms of dynamic energy. While the total fraction of bits hardened is small, which translates into low leakage and area increase, dynamic energy consumption grows noticeably because replacement bits are a significant fraction of the bits read and written on every access. In fact, they are accessed twice (read and updated). $ERVC$ scheme has low overheads, particularly for large caches (e.g., 16KB) since its cost does not grow linearly with cache size as it is the case for RVC (*repl perf*). For instance, $ERVC=8$ has a relative dynamic energy overhead of 2.8% with respect to the baseline and 1.5% with respect to RVC (*repl non-prot*).

ERVC extra area is even lower and leakage is completely negligible. Similarly, although not shown in the paper, *ERVC* overheads grow much less than those of the *RVC* (*repl perf*) when associativity is increased. Finally, *RVC* & *RRB* is the most efficient scheme although it requires some more routing than *ERVC*. In particular, *RVC=8* & *RRB=2* increases dynamic energy by 2.6% for a 16KB cache with respect to the baseline and by 1.3% with respect to *RVC* (*repl non-prot*).

In summary, *ERVC* and *RRB* are very efficient solutions to decrease the probability of failure at low cost. *RRB* overheads are lower than those of *ERVC* at the expense of some extra signal routing overhead. Both *ERVC* and *RRB* show better scaling properties with cache size and associativity than simply hardening replacement bits. Thus, depending on the degree of IP block changes allowed we will choose *ERVC* or *RRB*.

V. RELATED WORK

Several approaches have been proposed to deal with hard errors due to defects and degradation. The most relevant approaches consider alternative memory cell designs which offer higher robustness [8], [11]. However, those memory cells require larger area, and hence, their costs are prohibitive.

ECC has been shown to be very effective to mitigate unfrequent soft errors [9]. However, if ECC is employed for permanent faults, errors must be corrected before deploying the data, thus increasing cache latency. Moreover, those cache lines with permanent faults become less resilient to soft errors since part of the correcting capabilities are devoted to deal with permanent faults.

Approaches based on replacing faulty storage impact delay due to signal re-routing or indirections, and introduce significant wiring overhead [12]–[14]. Conversely, simply disabling faulty storage does not provide any WCET guarantee [4], [5], [15], so such a solution cannot be used in safety critical systems.

Some recent works provide some degree of time predictability in the presence of cache faults [5], [16], but not WCET guarantees. To the best of our knowledge, only the *RVC* [7] provides WCET guarantees in the presence of faults at low cost. The *RVC* is shown to be an efficient solution for line-dependent bits, but ignores set-dependent bits such as replacement bits. Our work builds upon the *RVC* concept to cover those set-dependent bits in an efficient manner.

VI. CONCLUSIONS

Process variations in modern technology nodes lead to higher faulty bit rates in caches. This issue is particularly relevant for safety critical systems where unpredictable latency of memory accesses leads to non-deterministic systems or unaffordable WCET estimates. The *RVC* has

been proposed recently for efficient and time-predictable fault tolerance in caches. However, the *RVC* does not fully solve this issue.

In this paper we extend the *RVC* with two schemes, the *ERVC* and the *RRB*, to tolerate permanent faults in cache replacement bits. The *ERVC* and the *RRB* decrease the probability of failure of caches by several orders of magnitude at the expense of small energy and area overheads.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Education and Science under grant TIN2007-60625 and HiPEAC. Jaume Abella has been also supported by the Generalitat of Catalunya under grant Beatriu Pinós 2009 BP-B 00260. Eduardo Quiñones has been also supported by the Spanish Ministry of Science and Innovation under the grant Juan de la Cierva JCI-2009-05455. Yanos Sazeides has been also partially funded from the University of Cyprus and Intel.

REFERENCES

- [1] A. Bhavnagarwala, T. Xinghai, and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 4, 2001.
- [2] A. Pour and M. Hill, "Performance implications of tolerating cache faults," *IEEE Trans. Comput.*, vol. 42, no. 3, 1993.
- [3] H. Lee, S. Cho, and B. Childers, "Performance of graceful degradation for cache faults," in *ISVLSI*, 2007.
- [4] D. Roberts, N. Kim, and T. Mudge, "On-chip cache device scaling limits and effective fault repair techniques in future nanoscale technology," in *DSD*, 2007.
- [5] J. Abella et al., "Low v_{ccmin} fault-tolerant cache with highly predictable performance," in *MICRO*, 2009.
- [6] R. Wilhelm et al., "The worst-case execution-time problem overview of methods and survey of tools," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, 2008.
- [7] J. Abella, E. Quinones, F. Cazorla, Y. Sazeides, and M. Valero, "RVC: A mechanism for time-analyzable real-time processors with faulty caches," in *HiPEAC*, 2011.
- [8] J. Kulkarni, K. Kim, and K. Roy, "A 160 mV, fully differential, robust schmitt trigger based sub-threshold SRAM," in *ISLPEd*, 2007.
- [9] C. Chen and M. Hsiao, "Error-correcting codes for semiconductor memory applications: A state of the art review," *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 124–134, 1984.
- [10] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "CACTI 6.0: A tool to understand large caches," *HP Tech Report HPL-2009-85*, 2009.
- [11] S. Jain and P. Agarwal, "A low leakage and SNM free SRAM cell design in deep sub micron CMOS technology," in *VLSID*, 2006.
- [12] I. Koren and Z. Koren, "Defect tolerance in vlsi circuits: techniques and yield analysis," *Proceedings of the IEEE*, vol. 86, no. 9, pp. 1819–1838, 1998.
- [13] T. Haraszti, "A novel associative approach for fault-tolerant MOS RAMs," *IEEE Journal of Solid-State Circ.*, vol. 17, no. 3, 1982.
- [14] F. Bower et al., "Tolerating hard faults in microprocessor array structures," in *DSN*, 2004.
- [15] A. Das et al., "Evaluating the effects of cache redundancy on profit," in *MICRO*, 2008.
- [16] N. Ladas, Y. Sazeides, and V. Desmet, "Performance-effective operation below V_{cc-min} ," in *ISPASS*, 2010.