# Combining RAM technologies for hard-error recovery in L1 data caches working at very-low power modes

Vicente Lorente[1], Alejandro Valero[1], Julio Sahuquillo[1],
Salvador Petit[1], Ramon Canal[2], Pedro López[1], and José Duato[1]

[1]Department of Computer Engineering
Universitat Politècnica de València
Valencia, Spain
vlorente@disca.upv.es, alvabre@gap.upv.es
{jsahuqui, spetit, plopez, jduato}@disca.upv.es

[2]Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya
Barcelona, Spain
rcanal@ac.upc.edu

***Abstract*—Low-power modes in modern microprocessors rely on low frequencies and low voltages to reduce the energy budget. Nevertheless, manufacturing induced parameter variations can make SRAM cells unreliable producing hard errors at supply voltages below Vccmin.**

**Recent proposals provide a rather low fault-coverage due to the fault coverage/overhead trade-off. We propose a new fault-tolerant L1 cache, which combines SRAM and eDRAM cells in L1 data caches to provide 100% SRAM hard-error fault coverage.**

**Results show that, compared to a conventional cache and assuming 50% failure probability at low-power mode, leakage and dynamic energy savings are by 85% and 62%, respectively, with a minimal impact on performance.**

## I. Introduction

Most current processors support multiple power modes to exploit the trade-off between performance and power. In *high-performance* mode, the processor works at a high frequency together with a high voltage level to speedup the execution of the workloads. In *low-power* mode, low frequency/voltage levels are used to improve energy savings. However, as transistor features continue shrinking in future technologies, manufacturing variations make semiconductor cells more unreliable at low voltages. Moreover, if the voltage is lowered beyond a given reliable level, namely Vccmin, the probability of failure increases exponentially.

Microprocessor caches have been typically implemented using fast SRAM cells. Nevertheless, eDRAM cells are being used in some modern microprocessors [1]. Despite being slower than SRAM cells, they improve storage density by a 3x to 4x factor and also reduce energy. Nevertheless, eDRAM technology requires refresh operations to avoid capacitors to lose their state. Because of each technology presents its advantages and shortcomings, recent works [1] [2] [3] propose to combine different semiconductor technologies to build high-performance and energy-efficient cache hierarchies.

Process variation affects the behavior of the memory cell technologies in different ways. In SRAM cells, it induces static noise margin variability which causes failures [4] in some cells (also known as hard errors) when working below Vccmin. On the other hand, eDRAM technology is also susceptible to device variations that basically lump into the cell retention time. Thus, variation problems can be addressed by increasing the refresh rate. Most existing proposals provide a rather low (e.g., less than 10%, see Section II) fault-coverage (the percentage of faults that can be detected/corrected) which is insufficient for future technology nodes [5]. This paper proposes the Hard Error Recover (HER) cache, which combines SRAM and eDRAM technologies to provide 100% SRAM fault-coverage set-associative L1 data caches while reducing area and power with respect to a conventional cache. HER caches are designed with two different operation modes: high-performance and low-power.

At high-performance, the HER cache works using the entire storage capacity. Most accesses hit the fast SRAM banks, while the eDRAM banks allow both energy and area savings. At low-power, the proposed memory architecture uses the eDRAM banks to keep copy of SRAM banks (i.e., replicas). As process variation in eDRAM cells does not affect their contents, these replicas enable the processor to recover from *any* number of SRAM bit failures due to manufacturing imperfections.

When compared to an ideal (fault-free) 32KB-4way L1 SRAM cache, the HER cache in low-power mode reduces leakage and dynamic energy by 85% and 62%, respectively; while slightly affecting the performance (by 2.54% in the worst case). On the other hand, at high-performance mode and compared to the conventional cache, IPC losses are lower than 1.9%, whereas leakage is reduced by 62% and dynamic energy by 40%.

## II. Motivation

The main reason of the low fault-coverage supported by existing proposals is that the devised solutions must trade off coverage for overhead (area, energy, performance, etc.). Table I

978-3-9815370-0-0/DATE13/©2013 EDAA

| Ref. | Coverage | Vmin | Cache | MHz in *lp* | IPC in *lp* | Power |
|---|---|---|---|---|---|---|
| [6] | 6/64 | 0.800 | 2MB, L2 | NA | NA | -6% |
| [7] | 4/64 | 0.490 | 32KB, L1<br>2MB, L2 | 500 | +10% | -71% |
| [8] | 419/32K | NA | 32KB, L1 | NA | NA | +1.8% |
| [9] | 4/256<br>20/512 | 0.490<br>0.475 | 32KB, L1<br>2MB, L2 | 500<br>500 | -10.7%<br>-10.7% | -85%<br>-85% |
| [10] | not required | 0.500 | 512B, L2<br>7KB, L1 | 10 | N A | -86% |
| [11] | 5/1024 | Refresh | 128MB, L3 | 2000 | -0.1% | -93% |
| HER | 100% SRAM | 0.500 | 32KB, L1 | 500 | -2.6% | -62% D<br>-85% S |

Table I
ERROR-FAILURE SCHEMES COMPARISON. LEGEND- NA: NOT AVAILABLE, LP: LOW-POWER.

summarizes, for a representative subset of recent proposals, performance and power in low-power mode. As observed, the highest fault coverage is achieved by [6], which is still less than 10%. Providing higher coverages in these proposals could become prohibitive in terms of area, delay, or power.

Unfortunately, technology projections [12] foresee that the ultimate nanoscale device will have a high percentage of non-functional devices from the beginning due to the high degree of variation produced in the manufacturing process. In this context, future fault tolerant caches must support a high percentage of failures. For example, we measured the probability of failure for a 22nm node with a Vcc ranging from 0.4 to 1V, and varying Vth due to process variation from 10% to 70%. Figure 1 shows the results. As observed, for a realistic Vth variation of 25% and 0.4V power supply (near threshold voltage), the probability of cell failure is by 20%, which is twice as large as the supported by the best existing proposal [6]. In short, the presented proposals will not match the coverage requirements of future technologies.

## III. MEMORY ARCHITECTURE PROPOSAL

The proposed technique presents four main contributions: i) 100% fault-coverage for process-variation induced faults; ii) just $1/n$ of storage capacity sacrificed for failure recovery in an $n$-way set-associative cache; iii) at high-performance mode, the whole cache storage capacity is enabled; and iv) no refresh operation is implemented.

Since the control bits and the tag array occupy an area much smaller than the data array, this paper focuses on the potential benefits in the data array, and assumes that special low-power non-defective cells [13] are used for the whole tag array and control bits.

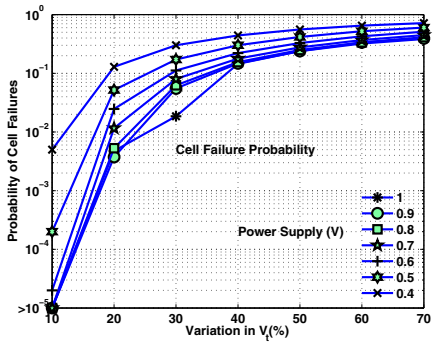



Figure 1. Probability of cell failures.

### A. High-Performance Mode

For the data array, the HER cache uses $k$ cache banks to implement an $n$-way set-associative cache, where $k/n$ banks are implemented with SRAM technology and the remaining $k - k/n$ with eDRAM technology. Figure 2 depicts a block diagram of a 4-way set-associative HER cache for $k = 8$.

In L1 caches, more than 90% of cache accesses hit the *Most Recently Used* (MRU) way [14]. Thus, to achieve good performance, in the HER cache the MRU block of each cache set is always stored in an SRAM bank. Note that L1 caches cannot be implemented only with eDRAM technology due to reduced availability and unacceptable performance drops [15].

The whole structure is accessed as a way-prediction cache (similarly as done in IBM POWER7 [1]) as follows. In the first cycle, the tags of all ways are checked and -only- the SRAM data way is read. On a hit in the SRAM way no eDRAM bank is accessed, so it avoids unnecessary accesses to the eDRAM banks. After checking the tags, if a hit occurs in any eDRAM block, the associated data is accessed (i.e., a destructive read), incurring additional penalty cycles and data is delivered to the CPU. Then, a swap operation between this block and the one stored in the SRAM way is triggered. Hence, the new MRU block is stored in an SRAM bank while the previous MRU block is moved to an eDRAM bank.

The generation time of a block is defined as the elapsed time since the block is brought into the cache until it is evicted. During the generation time of a given block *A*, whenever this block is accessed, it is stored in an SRAM bank (since it is the MRU block). In this situation, if another block *B* in the same cache set is accessed, a swap operation is triggered. Thus, block *B* becomes the MRU block and block *A* is transferred to the eDRAM way that previously stored block *B*. From this point, the eDRAM capacitors must retain the data from block *A* until it is referenced again. We define this time as the *required retention time*. In order to avoid refresh circuitry, which is costly in terms of area and energy consumption [11], eDRAM capacitors in the HER cache must retain data for longer than the maximum required retention time observed among the blocks.

The contents of a block may be lost while it is stored in an eDRAM way until it is evicted (i.e., the end of its generation time), since this period may be longer than the required retention time. This could lead to incorrect program execution in case of block contents were dirty. To avoid this situation, the scheme distinguishes between two types of writeback operations: i) writebacks due to replacements and ii) writebacks due to capacitor discharges. The first type, like in conventional caches, is triggered when a dirty block is selected for replacement. The second type is triggered when checking regularly (*scrubbing*) the state of all the valid blocks located in

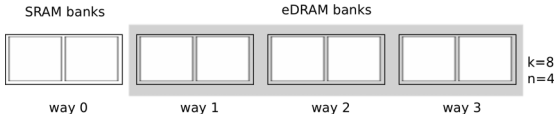



Figure 2. A 4-way HER cache with 8 banks.

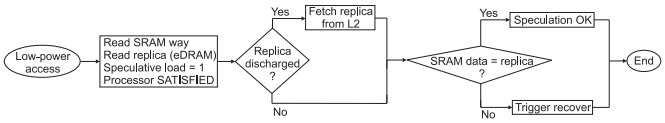



Figure 3. State diagram for a read hit in the SRAM way.

eDRAM banks. If the valid block is found dirty, a preventive writeback to L2 is triggered. In addition, whether it is dirty or not, the block is invalidated in L1. This prevents accessing to an eDRAM block that has lost its data.

The scrub operation can be implemented with a single binary counter [16] for the entire cache initialized to the required retention time divided by the total number of eDRAM blocks in cache, and guarantees that all eDRAM blocks are checked (i.e., written back if dirty and invalidated) before this retention time expires. The impact of bank contention on performance is minimal because: i) most accesses hit in the SRAM banks and ii) most banks are eDRAM, so when one of them is being scrubbed, the remaining ones can be accessed.

### *B. Manufacturability Issues*

The proposal uses error-free SRAM cells designed to work at low voltages (160mV) [13] to build the tag array and control bits. The main drawback of these cells is the large area (they are twice as large as conventional cells) they occupy, which makes them inappropriate to implement the entire cache. In the HER cache, we compensate the additional tag array area with the reduction of data array area when using eDRAM banks. In fact, the total cache area is reduced by 31% compared to the conventional SRAM cache [1].

SRAM and eDRAM technologies require different process steps when manufacturing. To ease the manufacturability, each bank is implemented with a single technology. The design assumes that both technologies are compatible with current logic processes [17] [18]. In fact, some companies [19] manufacture eDRAM using logic technologies with no or minimal changes in manufacturing processes. Engineers also consider the adoption of capacitor-less DRAM structures (i.e., using the gate capacitance of another transistor). This work proposes an architecture level approach for fault-tolerance that can accommodate any technological alternative that meets the timing and retention values derived.

### *C. Low-Power Working Behavior*

By design, SRAM cells in the data array can be faulty at low voltages due to manufacturing imperfections. In contrast, eDRAM cells can work correctly at very-low voltages [17]. In this case, a low voltage is stored in the cell, so the access latency increases and the retention time decreases. To enlarge the retention time, the capacitance must be higher. Therefore, any number of faulty SRAM cells can be managed through the use of an eDRAM way as a backup of the SRAM way (also referred to as replica), provided that the data in the replica is accessed within the required retention time.

This work assumes that a relatively wide range of voltage values can be supported at low-power mode. We propose to detect the faulty lines at runtime with a single control bit for each SRAM line, and by comparing the SRAM contents with those of the eDRAM replica each time an SRAM line is accessed at low-power mode. If the comparison is false, the control bit, namely *SRAM-faulty* bit, is set to one in order to avoid wasting energy in subsequent comparisons. As opposite, if the comparison is true, the SRAM-faulty bit remains cleared. Notice that subsequent comparisons are still required since the value of the defective SRAM may match the right value.

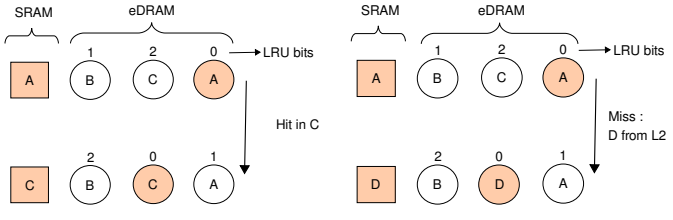
Figure 3 depicts the state diagram of the cache controller to deal with a read hit event in the SRAM way. As in high-performance mode, the data is delivered to the processor as soon as it is read. However, from this point, it is unknown whether the read data is correct or not, since some SRAM bits may fail. Thus, the load instruction is allowed to proceed using a speculative value. Then, the eDRAM replica is read and compared to the SRAM value to solve the speculation. Notice that, unlike high-performance mode, each time an eDRAM block is read it must be rewritten. If the eDRAM replica is not valid, the data block must be fetched from L2. If the SRAM and eDRAM values match, the load becomes non-speculative; in other words, the processor is already working with the correct data. On mispeculation, the load and subsequent instructions will be aborted by triggering the conventional recovery mechanisms.

When a line is faulty due to process variation at a given low-voltage level, then all subsequent load instructions to that address would incur on mispeculation. Energy consumption due to mispectulation can be largely saved thanks to the SRAM-faulty bit, since the comparison between the read SRAM data and its replica is not performed when the SRAM-faulty bit is set.

Regarding a write hit event on the SRAM way, a write must be performed both in that way and its eDRAM replica, except if the SRAM-faulty bit is set. In such a case, the write should be performed only in the replica.

Figure 4(a) shows an example of a read hit in an eDRAM way other than the replica. In this event, and in case that the SRAM way is not faulty, the data must be copied to the SRAM way, thus overwriting its contents and the LRU control bits are accordingly updated. Data only moves from eDRAM to SRAM, but no bidirectional swap is performed (as done in high-performance mode). Notice that overwriting the SRAM way does not mean any loss of information, since the previous SRAM data remains in the previous eDRAM replica. In case of write hit, both the SRAM way and the replica are updated with the same data.

Finally, on a cache miss, in both read and write operations,




(a) Read hit in eDRAM (b) Cache miss

Figure 4. Example of accesses in low-power mode.

[1] Area details are not shown due to space restrictions.

the block is fetched from L2 (or a lower level of the memory hierarchy). The incoming block will be written both in the SRAM way (MRU line) and in an eDRAM way (i.e., replica), which is provided by the LRU algorithm. Figure 4(b) shows an example where the accessed block *D* replaces the block *C* of the SRAM way. Notice that no data movements are required between eDRAM and SRAM ways.

### D. Mode Changes

The processor must also provide support to change from high-performance mode to low-power mode and vice versa. Changing from high-performance to low-power mode causes the generation of a replica for each SRAM cache block. For this purpose, all SRAM blocks are written (copied) in the LRU way of its set. Of course, if the block in that way is dirty, it must be written back to L2. After that, the voltage can be lowered to the desired target level.

On the contrary, changing from low-power to high-performance mode requires, i) rising the voltage to the target high-performance mode and ii) moving the contents of each replica to the SRAM way and invalidating (i.e., freeing the space) the eDRAM lines storing the replicas. This invalidation is needed so that the whole cache capacity is available again. Remark that all the replicas must be copied to the SRAM lines regardless of the value of the SRAM-faulty bit since this bit is updated only if the line is accessed.

Finally, notice that voltage can be reduced or increased when changing among low-power modes. So, if the voltage is reduced (hence, new defective bits can appear) there is no need to reset the SRAM-faulty bits. On the contrary, false-positives can appear if the voltage is increased. In such a case, all the SRAM-faulty bits must be cleared to enable the data comparisons.

## IV. Experimental Evaluation

The proposal has been modeled on top of the SimpleScalar (with Alpha ISA) simulation framework [20] to obtain the execution time and memory events required for estimating dynamic energy (e.g., cache hits, misses, both types of write-back operations, swaps, etc.). All the bank contention induced by these events has also been modeled. However, accesses to different cache banks may be concurrently performed. The CACTI 5.3 tool [21] with the eDRAM cell proposed in [17] has been used to estimate leakage power, dynamic energy per cache access type (e.g., read or write), access time, and capacitances for 45nm. The overall dynamic energy was calculated combining the results of both simulators.

All these results have been obtained for each operation mode, from now on referred to as *hp* (high-performance) and *lp* (low-power) modes, respectively. For the *hp* mode, it has been assumed a voltage/frequency pair of 1.3V/3GHz and no bit failures. Two different voltage/frequency pairs have been studied in *lp* mode, referred to as *lp1* and *lp2*, respectively. *lp1* assumes 0.7V/1.4GHz and *lp2* 0.5V/500MHz similarly to [7]. The probability of failure for a 45nm node was calculated as 20% and 50% of the SRAM bits, respectively. Although some defective bits can be in the same line, the results presented assume the worst case, that is, all defective bits are located in different cache lines.

| Microprocessor core | |
|---|---|
| Issue policy | Out of order |
| Branch predictor type | Hybrid gShare/Bimodal:<br>gShare has 14-bit global history plus 16K 2-bit counters<br>Bimodal has 4K 2-bit counters |
| Branch predictor penalty | 10 cycles |
| Fetch, issue, commit width | 4 instructions/cycle |
| ROB size (entries) | 256 |
| **Operation Modes** | |
| High-performance (*hp*) | 1.3V/3GHz, 0% errors |
| Low-power 1 (*lp1*) | 0.7V/1.4GHz, 20% errors |
| Low-power 2 (*lp2*) | 0.5V/500MHz, 50% errors |
| **Memory hierarchy** | |
| L1 data cache | 32KB-4way, 64 byte-line<br>2 SRAM and 6 eDRAM banks |
| L1 data cache hit latency | SRAM: 2cc in *hp*; 1cc in *lp*<br>eDRAM: 4cc in *hp*; 2cc in *lp* |
| L2 data cache | 512KB-8way, 64 byte-line |
| L2 data cache hit latency | 10 cycles |
| Memory access latency | 100 cycles |

Table II
ARCHITECTURAL MACHINE PARAMETERS.

For comparison purposes, a conventional SRAM cache has also been modeled. In this scheme, no error failures are assumed regardless of the operation mode. At high-performance mode, the scheme is referenced as Conv cache and at low-power as ZfConv (zero-failure conventional) cache.

Table II summarizes the architectural parameters. Access time in the 32KB-4way L1 caches was estimated according to the bank technology and processor speed. Although the access time increases with the eDRAM capacitance, this increase is almost negligible for the capacitances analyzed in this work and, since in *lp* modes the cycle time is longer, the access time (quantified in processor cycles) becomes lower. Remark that the length of the swap operation is determined by the sum of the latencies of the different accesses to the involved cache banks, and the processor stalls until this operation finishes if it demands data located in the same bank (i.e., contention). Integer (Int) and floating-point (FP) SPEC 2000 benchmarks [22] were run using the *ref* input sets. Statistics were collected during 500M instructions after skipping 1B instructions.

### A. Performance Evaluation

Figure 5 shows, for each application, the normalized performance in *hp* mode of a 32KB-4way HER cache implemented with an *infinite retention time* (i.e., theoretical capacitors without charge losses) compared to a conventional cache

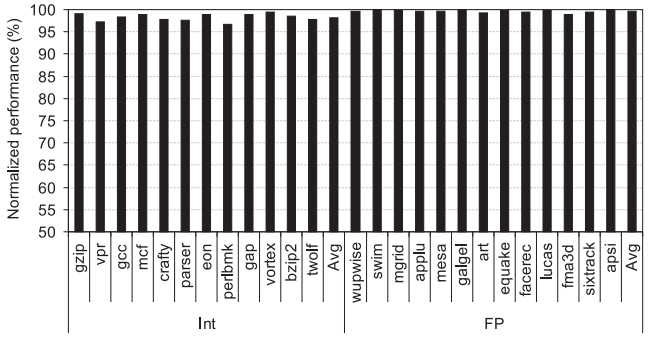



Figure 5. Normalized performance in *hp* mode.

| Operation mode | Retention time (cycles) | Capacitance (fF) | Perf. deg. (%) | |
|---|---|---|---|---|
| | | | Int | FP |
| *hp* | 38K | 2 | 1.88 | 0.38 |
| *lp1* | 44K | 8 | 1.75 | 0.21 |
| *lp2* | 62K | 43 | 2.54 | 0.29 |

Table III
HER CACHE CHARACTERISTICS.

| Bench. type | Hit rate (%) | Conv | HER | | |
|---|---|---|---|---|---|
| | | *hp* | *hp* | *lp1* | *lp2* |
| Int | SRAM | 97.8 | 92.6 | 73.4 | 42.1 |
| | eDRAM | – | 5.2 | 4.7 | 4.7 |
| | eDRAM replica | – | – | 19.3 | 50.7 |
| | total | 97.8 | 97.8 | 97.5 | 97.5 |
| FP | SRAM | 93.7 | 86.7 | 68.8 | 41.9 |
| | eDRAM | – | 7.0 | 6.0 | 6.0 |
| | eDRAM replica | – | – | 17.9 | 44.8 |
| | total | 93.7 | 93.7 | 92.7 | 92.7 |

Table IV
HIT RATE DISTRIBUTION.

(Conv) with the same storage capacity. Notice that the Conv cache scheme imposes an upper-bound since this cache does neither use way-prediction nor is implemented with the *slower* eDRAM cells. In other words, these values are the maximum performance that a HER cache can achieve. The IPC losses for integer benchmarks are, on average, by 1.88%; and much lower (0.38%) for floating-point benchmarks.

Table III shows, for each operation mode, the required retention time that real capacitors should have to allow the cache to match the performance (i.e., harmonic mean of IPCs) of a HER cache implemented with infinite retention time. To estimate the required capacitances, the power supply of each operating mode has also been considered. Finally, the average performance degradation with respect to the conventional cache is also presented.

Trench capacitors can be used to obtain values up to 30fF [23]. Thus, both *hp* and *lp1* operation modes can be supported with them. However, in *lp2* mode, the 30fF capacitor allows a retention time of *only* 44K cycles, which is smaller than the optimal required (62K). Results show a rather low performance degradation even in this case (less than 2.6%).

As expected, the required retention time increases with the probability of line failure in *lp* modes since more accesses perform in the eDRAM banks; thus, these banks must retain their data for longer. Performance degradation also increases with the probability of failure. This is mainly due to the *slower* access to eDRAM replicas and bank contention. Notice that performance in high-performance and low-power modes cannot be directly compared since the processor speed differs.

Table IV shows the average hit rate of both cache schemes across the studied operation modes. The hits in the eDRAM replica are also presented for the low-power modes. Remark that in *hp* mode the total hit rate is the same as the hit rate of the conventional cache, which confirms that the obtained required retention time does not yield to performance losses due to capacitor discharges. As expected, on average, most accesses hit the MRU line (SRAM and eDRAM replica). Notice that the SRAM hit rate in HER caches is much lower in the more defective *lp2* mode than in the *lp1* mode. Finally, the reduction of the effective eDRAM capacity due to replicas has a negligible impact on the eDRAM hit rate.

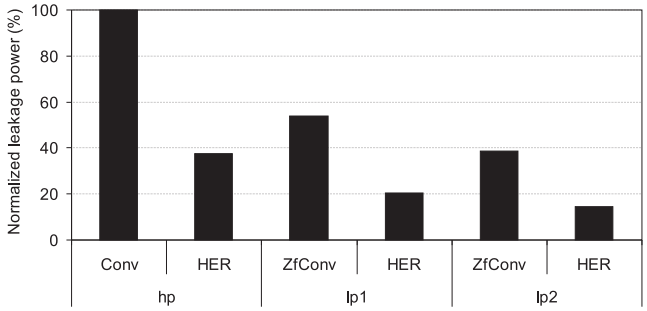



Figure 6. Normalized leakage power.

### B. Power and Energy Consumption

Figure 6 illustrates the normalized leakage power with respect to the conventional SRAM approach. Thanks to the use of eDRAM cells, the HER cache reduces leakage by design, regardless of the operation mode. However, in *lp* mode, benefits are also achieved because of the lower voltage supply.

In contrast, power savings of the non-defective ZfConv approach come only from the reduction in the supply voltage. Leakage savings provided by the HER cache can be as high as 62% in *hp* mode and 85% in *lp2* mode. Note that leakage, that is proportional to the number of transistors, is the dominant source of energy consumption in current technologies.

To provide insights in the dynamic energy savings, the total dynamic energy has been divided into five categories: SRAM hits, eDRAM hits, eDRAM replica hits, misses, and writebacks. The SRAM hits category includes the access to the eDRAM replica and the access to all SRAM ways, which are accessed in parallel in the Conv cache; the eDRAM hits category includes accessing both the SRAM way and the target eDRAM way. In addition, it also considers the energy due to swaps (unidirectional transfers in *lp*); the eDRAM replica hits category also considers the access to the SRAM faulty lines; the misses category also includes unidirectional transfers in *hp* mode; and finally, both misses and writebacks include the energy consumed by both L1 and L2 cache accesses.

Figure 7 shows these values normalized with respect to the conventional cache. Important differences appear in the SRAM hits category mainly due to the proposal implements only one SRAM way, which is accessed first. Notice also that the energy required by swaps does not noticeably affect the total energy. In addition, the eDRAM replica category has a minor impact on the total energy, being a bit larger in the more defective *lp2* mode. The misses category slightly varies across the different schemes. The major differences appear

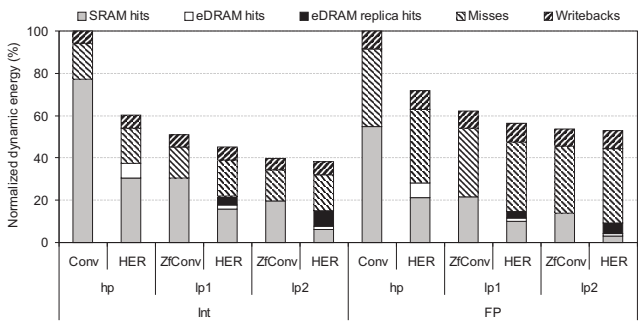



Figure 7. Normalized dynamic energy *categorized*.

in *lp* modes mainly due to the effective storage capacity is smaller. Finally, remark that the writeback policy (including writebacks due to replacements and capacitor discharges) of the HER cache has little impact on the overall energy since we measured the amount of writebacks performed by this policy and found that, on average, the overall number of writebacks increases by 5.3%. In *lp2* mode, energy benefits can be, on average, as high as 62% for integer benchmarks. An interesting observation is that the HER cache saves dynamic energy compared to the non-defective ZfConv cache even in *lp2* mode where the probability of defective lines is 50%.

## V. Related Work

Due to inter and intra-die process parameter variations, memory cells that are marginally functional during manufacturing tests can undergo runtime failures due to voltage/thermal noise or aging effects. Depending on the impact of these effects, different segments of a memory array may move to different reliable design corners that can be determined using post-fabrication characterization. Once unreliable blocks have been identified, solutions can be classified in two main approaches: correcting unreliable blocks and avoiding the use of those blocks. Regarding the former approach, in [6], Somnath et al. classify memory blocks in three main groups and apply different Error Correcting Codes (ECC) to restore blocks according to the group they belong to. In [7], Alameldeen et al. propose an adaptive cache design that uses up to half the data array to store ECC information in low-power mode. Solutions belonging to the latter approach perform a test that is required to identify those segments of the cache that fail at low voltage. Agarwal et al. [8] propose a variation-aware cache architecture, which adaptively resizes the cache. Wilkerson et al. [9] propose two architectural techniques that reduce the effective cache storage capacity up to 50%.

Another solution [10] proposes to reduce bit failure by enlarging SRAM cells and choosing an appropriate supply voltage. This scheme consists of a near threshold tolerant cache way and several conventional SRAM ways. The former way is implemented with large error resilient 8T cells, whereas the remaining ways are implemented with conventional cells.

Refresh power potentially represents a large fraction of the overall system power, particularly during low-power states when the processor is idle. In [11], Wilkerson et al. reduce cache refresh power by increasing the refresh time from 30$\mu$s (worst-case) to 440$\mu$s. This increase reduces power substantially but causes errors due to capacitor discharges. This problem can be solved by using costly ECC codes.

## VI. Conclusions

This paper has combined SRAM and eDRAM technologies to support 100% fault-coverage of errors caused by process variation imperfections, which is a major design concern in future technology nodes.

HER caches have been designed with two different operation modes: high-performance and low-power. In high-performance mode, the HER cache works with its whole cache capacity and despite the higher access time of the eDRAM ways, thanks to storing the MRU blocks in the faster SRAM banks, IPC losses for a 32KB-4way are by 1.9% compared to a conventional cache with the same storage capacity.

In low-power mode, two different voltage levels with different probability of failure (20% and 50%) have been analyzed. Experimental results have shown that, for the more defective mode, leakage savings can be as high as 85% and dynamic energy is reduced by 62%. Moreover, this is achieved by maintaining performance degradation always below 2.6%.

## Acknowledgments

This work was supported by the Spanish MICINN (TIN2010-18368) with the Consolider-Ingenio 2010 Programme co-funded by the European Commission FEDER funds (CSD2006-00046) and co-funded with the Plan E funds (TIN2009-14475-C04-01). Additionaly, it was supported by Generalitat de Catalunya (2009SGR1250), by FP7 program of the European Commission (TRAMS-248789), and by Spanish MINECO (TIN2012-38341-C04-01).

## References

[1] B. Sinharoy *et al.*, "IBM POWER7 multicore server processor," *IBM J. Research and Development*, vol. 55, no. 3, pp. 1–29, 2011.
[2] X. Wu *et al.*, "Hybrid Cache Architecture with Disparate Memory Technologies," in *Proc. ISCA-36*, 2009, pp. 34–45.
[3] A. Valero *et al.*, "An Hybrid eDRAM/SRAM Macrocell to Implement First-Level Data Caches," in *Proc. MICRO-42*, 2009, pp. 213–221.
[4] S. Mukhopadhyay *et al.*, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859–1880, 2005.
[5] S. Nomura *et al.*, "Sampling + DMR: Practical and Low-overhead Permanent Fault Detection," in *Proc. ISCA-38*, 2011, pp. 201–212.
[6] S. Paul *et al.*, "Reliability-Driven ECC Allocation for Multiple Bit Error Resilience in Processor Cache," *IEEE Trans. Computers*, vol. 60, no. 1, pp. 20–34, 2011.
[7] A. R. Alameldeen *et al.*, "Adaptive Cache Design to Enable Reliable Low-Voltage Operation," *IEEE Trans. Comp.*, vol. 60, pp. 50–63, 2011.
[8] A. Agarwal *et al.*, "Process Variation in Embedded Memories: Failure Analysis and Variation Aware Architecture," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1804–1814, 2005.
[9] C. Wilkerson *et al.*, "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," in *Proc. ISCA-35*, 2008, pp. 203–214.
[10] R. G. Dreslinski *et al.*, "Reconfigurable Energy Efficient Near Threshold Cache Architectures," in *Proc. MICRO-41*, 2008, pp. 459–470.
[11] C. Wilkerson *et al.*, "Reducing Cache Power with Low-Cost, Multi-bit Error-Correcting Codes," in *Proc. ISCA-37*, 2010, pp. 83–93.
[12] *Semiconductor Industries Association, "International Technology Roadmap for Semiconductors", 2009*.
[13] J. P. Kulkarni *et al.*, "A 160 mV, Fully Differential, Robust Schmitt Trigger Based Sub-threshold SRAM," in *Proc. ISLPED*, 2007.
[14] S. Petit *et al.*, "Exploiting Temporal Locality in Drowsy Cache Policies," *Proc. 2nd Conference Computing Frontiers*, pp. 371–377, 2005.
[15] P. G. Emma *et al.*, "Rethinking Refresh: Increasing Availability and Reducing Power in DRAM for Cache Applications," *IEEE Micro*, vol. 28, no. 6, pp. 47–56, 2008.
[16] S. Kaxiras *et al.*, "Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power," in *Proc. ISCA-28*, 2001, pp. 240–251.
[17] J. Barth *et al.*, "A 500 MHz Random Cycle, 1.5 ns Latency, SOI Embedded DRAM Macro Featuring a Three-Transistor Micro Sense Amplifier," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 86–95, 2008.
[18] R. E. Matick and S. E. Schuster, "Logic-based eDRAM: Origins and rationale for use," *IBM J. Research and Development*, vol. 49, no. 1, pp. 145–165, 2005.
[19] *http://www.uniramtech.com/embedded_dram.php*.
[20] D. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0," *ACM SIGARCH Computer Arch. News*, vol. 25, no. 3, pp. 13–25, 1997.
[21] S. Thoziyoor *et al.*, "CACTI 5.1," *HP Laboratories, Tech. Rep.*, 2008.
[22] *http://www.spec.org/cpu2000*.
[23] W. Mueller *et al.*, "Challenges for the DRAM Cell Scaling to 40nm," *IEEE International Electron Devices Meeting*, p. 339, 2005.