# Analyzing the Optimal Voltage/Frequency Pair in Fault-Tolerant Caches

Vicente Lorente[1], Alejandro Valero[1], Salvador Petit[1], Pierfrancesco Foglia[2], and Julio Sahuquillo[1]

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

[2]Dipartimento di Ingegneria dell'Informazione
Università di Pisa
Pisa, Italy
foglia@iet.unipi.it

***Abstract*—When the processor works at very-low voltages to save energy, failures in SRAM cells increase exponentially at voltages below $VCC_{min}$. In this context, current SRAM-error detection and correction proposals incur on a significant performance penalty since they increase access latency and disable cache lines that cannot be corrected, so decreasing the effective cache capacity. This reduction implies more cache misses, so enlarging the execution time which, contrary to expected, can turn in higher energy consumption.**

**This paper characterizes SRAM failures at very-low voltages and presents an evaluation methodology to analyze the impact on energy consumption of error correction approaches. To do so, several voltage/frequency pairs are studied and the optimal pair is identified from an energy point of view.**

**To focus the research, experimental results have been obtained for the recently proposed fault-tolerant HER cache. Results show that, for a 32nm technology node, the voltage/frequency pair of 0.45V/800MHz, which induces by 31% SRAM failure rate, provides the lowest overall energy consumption (by 62% energy savings compared to a non-faulty conventional cache).**

## I. Introduction

Current microprocessors support multiple power modes to exploit the trade-off between performance and power. In order to speedup the execution time, in *high-performance* modes the processor enables a high frequency which makes use of a high voltage level. In *low-power* modes, low voltage/frequency levels are used for energy savings.

Microprocessor caches are typically implemented with fast Static Random-Access Memory (SRAM) cells. Parameter variations due to imperfections in the fabrication process increase as transistor features continue shrinking in future technologies. This makes SRAM memory cells more unreliable at low voltages because process variation induces Static Noise Margin (SNM) variability in such cells, which causes failures [1] (known as hard errors) in some of them when working below a certain reliable voltage level, namely $VCC_{min}$.

To increase reliability in SRAM cache arrays, several techniques have been used by industry [2] as row/column redundancy or Error Detection/Correction Codes (EDC/ECC). However multi-bit error correction codes have high overhead [3] because they need additional storage for correction codes as well as complex and slow decoders to identify errors. Other SRAM fault-tolerant solutions basically allow the system to work below $VCC_{min}$ by disabling those segments of the cache where one or more bits fail, thus reducing the effective storage capacity [3]–[8]. Moreover, the highest fault coverage achieved by these techniques is below 10%, which makes them unsuitable for fault-dominated future technology nodes.

On the other hand, embedded Dynamic RAM (eDRAM) cells [9] have emerged in recent processors [10] [11] to build low-level caches since they allow high density and low power consumption. An interesting feature of these cells is that hard errors basically lump into the cell retention time instead of altering the stored value, thus variation problems can be addressed in eDRAM by increasing the refresh rate. To deal with performance and hard errors, both SRAM and eDRAM cells have been recently combined to implement a hybrid Hard Error Recovery (HER) L1 data cache architecture [12], which is able to support 100% of SRAM faulty cells in low-power modes.

Nevertheless, the HER cache also presents performance penalties at very-low voltages, since its effective SRAM storage capacity is severely reduced (up to 90% failure rate). In addition, the retention time of the eDRAM cells is also affected, which implies that eDRAM cell contents are lost faster causing noticeable rises in the miss ratio. Finally, average access time also increases due to the higher latency of eDRAM technology.

In summary, existing SRAM fault-tolerant proposals incur on a significant performance penalty since they increase access latency and reduce the effective cache capacity when working at low-power modes. At very-low voltages, the execution time can dramatically grow due to these effects, so extra energy is required to complete the program execution. Moreover, low voltages are necessarily paired with low processor frequencies, extending the cycle time in such a way that the execution time can be critically enlarged. Unfortunately, this can imply not only performance loss but also higher energy consumption with respect to higher voltage/frequency pairs. Therefore, despite the processor is working in a low-power mode and voltage is reduced for energy savings, the total energy consumption can exceed that consumed with a higher

voltage level. We found that this effect appears regardless of the effectiveness of the fault-tolerant technique, even if it is able to recover 100% SRAM errors in low-power modes.

This paper presents a methodology to evaluate the impact on energy consumption of error detection and correction proposals. To focus the research, experimental evaluation concentrates on the recently proposed hybrid eDRAM/SRAM HER L1 data cache [12]. This work analyzes a wide range of voltage/frequency pairs to find out the optimal pair for this fault-tolerant cache in terms of energy. The study is backed up by a deep analysis of the failure probability in SRAM cells implemented with 32nm technology for operating voltages ranging from 0.9V (0% SRAM failure rate) to 0.35V (90% SRAM failure rate). The devised methodology can be straightforwardly adapted to be used in any fault-tolerant technique, specially in those suffering significant performance losses.

To the best of our knowledge, this is the first work that proposes an evaluation methodology to determine the optimal voltage/frequency pair in terms of energy consumption taking into account performance losses caused by fault-tolerant techniques.

Experimental results show that a voltage/frequency pair of 0.45V/800MHz is the best in terms of energy consumption. Despite such a low voltage induces a probability of failure as high as 31% in SRAM cells, the overall energy savings are up to 62% on average with respect to an SRAM cache working at high-performance mode.

The rest of this paper is organized as follows. Section II discusses related work. Section III introduces the HER cache architecture. Section IV studies the failure probability in SRAM cells. Section V analyzes the experimental results, and finally, Section VI summarizes the paper.

## II. Background

Prior reliability-aware research focusing on SRAM caches can be classified into three main categories according to the type of technique they apply: i) Error Correcting Codes (ECC), ii) disabling failing portions of the cache, and iii) making use of error-resilient memory cells such as eDRAM-based cells or larger (e.g., 8T and 10T) SRAM-based cells.

Some approaches falling in the first category (e.g., [4] [5]) are able to recover the data stored in some defective cells, but they do not allow high voltage reductions because the additional storage needed for ECC becomes prohibitive. In [4], authors classify the cache memory blocks in three main types depending on the threshold voltage variation of their transistors (NMOS and PMOS). Then, different ECCs are applied to each block according to this classification. Alameldeen *et al.* [5] proposed an adaptive cache design that uses up to half the data array to store ECC information at low voltage to reduce energy. In high-performance mode, the whole data array is enabled. Additional hardware structures, monitored by the operating system, are required to select the desired reliability level. In low-power mode, some physical ways are used to store ECC information. For instance, to support *only* 4-bit error correction for each 64 bits segment at a 520mV supply voltage, the number of ways devoted to ECC is as high as half the number of cache ways.

Schemes in the second category (e.g., [3] [6]) go a step further and are applied when the number of errors cannot be successfully recovered with ECC techniques. For this purpose, they dynamically disable faulty cells when ECC codes are not enough, so reducing the effective cache capacity. In [3], authors proposed two architectural techniques, namely Word-disable and Bit-fix, that reduce the effective cache storage capacity by 50% and 25%, respectively. The former combines two consecutive cache lines in low voltage mode to form a single cache line without failing words. The latter uses a quarter of the ways to keep track of the faulty data (words and bits) in other ways of the set. A test is performed at boot time to identify those segments of the cache that fail at low voltage. In [6], Agarwal *et al.* presented a variation-aware cache architecture, which adaptively resizes the cache to avoid accessing faulty blocks. When a faulty block is accessed, the bitmap information is used to select a non-faulty block in the same row. The cache implements a self-test circuitry, which tests the entire cache and detects faulty cells. Tests are conducted whenever the operating conditions change. In [13], authors introduce an orthogonal scheme, which combines different leakage saving techniques to limit the effects of $VCC_{min}$ on power consumption.

Approaches belonging to the third category avoid failures by implementing alternative cells that increase reliability. Approaches based on large SRAM cells (e.g., [14] [7]) achieve this goal but increase the area occupation, while eDRAM-based techniques like the HER cache [12] do not present this drawback. In [14], Chang *et al.* propose an 8T SRAM cell design that avoids variation-induced read failures (i.e., bit flips), however, cell area increases by 30% with respect to typical 6T cells. The Reconfigurable Energy-efficient Near Threshold (RENT) cache architecture [7] implements a single 8T-based cache way and all the remaining ways with typical 6T SRAM cells. Energy consumption is saved by reducing the voltage in the 8T way, while the other ways work with a higher voltage level to avoid faulty cells. Finally, the HER cache architecture, which is used to evaluate the proposed methodology, is explained below.

## III. HER Cache Architecture

This section briefly describes how the HER cache architecture [12] works in both high-performance and low-power mode, hereafter referred to as *hp* and *lp* modes, respectively. This design implements $n$-way set-associative L1 data caches with one SRAM way and $n-1$ eDRAM ways. Each way is implemented with a couple of banks to reduce bank contention. For instance, Figure 1 depicts a diagram of the tag array and the data array of a 4-way HER cache with eight cache banks. Notice that the tag array is implemented with larger fault-free 8T SRAM cells.

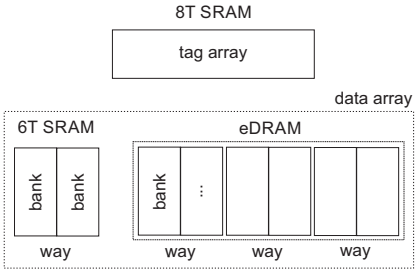



Figure 1. Block diagram of a 4-way HER cache with eight cache banks.

### A. High-Performance Mode

When the processor works at *hp* mode, the whole structure is accessed as a way-prediction cache as follows. In the first cycle, the tags of all the ways are checked and -only- the SRAM data way is read. On a hit in the SRAM way, no eDRAM bank is accessed; so avoiding unnecessary accesses to eDRAM banks. After checking the tags, in case of hit in any eDRAM block, the target bank is accessed (i.e., a destructive read occurs), and delivered to the CPU in a longer hit time. Then, a swap operation between this block and that stored in the SRAM way is triggered. This ensures that the Most Recently Used (MRU) block is always kept in the *fast* SRAM banks for performance since the hit ratio in the MRU way of L1 caches is usually above 90% [15]. On a cache miss, the previous MRU block is transferred to the eDRAM bank storing the Least RU (LRU) data according to the replacement algorithm, whereas the incoming data (i.e., the new MRU block) is allocated in the SRAM way.

Refresh circuitry consumes important energy [8], which is avoided in the HER cache by design. Capacitors are allowed to lose their contents, which could lead to incorrect program execution in case of dirty blocks. To deal with this fact, the scheme distinguishes between two types of writeback operations: i) writebacks due to replacements and ii) write-backs 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 *scrubbing* the state of valid blocks located in eDRAM banks. If a valid block is found dirty, a preventive writeback to L2 is triggered before its retention time expires. Then, regardless of whether it is dirty or clean, the block is invalidated in L1. This way prevents accessing a block that has lost its data. The scrub operation is implemented with a single binary counter for the entire cache, initialized as the lowest retention time of the capacitors divided by the number of eDRAM blocks (scrub period). Every time that the counter counts down to zero, a given eDRAM block is checked. Please refer to [12] for further details.

### B. Low-Power Mode

In *lp* mode a copy of the contents of the SRAM way is placed in an eDRAM way referred to as replica, that is, the effective cache capacity is reduced by $1/n$. Faulty SRAM blocks are detected at runtime by comparing their contents with those of the eDRAM replica each time an SRAM block is accessed. If the comparison results false, a control bit per block, namely SRAM-faulty bit, is set to *one* to avoid subsequent comparisons.

Below we explain the actions that the controller must perform according to four main type of events: i) read hit in the SRAM way, ii) write hit in the SRAM way, iii) read/write hit in an eDRAM way, and iv) cache miss.

**Read Hit in the SRAM Way.** As in high-performance mode, the data are delivered to the processor as soon as they are read. At this point, it is unknown whether the read data are correct or not (assuming that the SRAM-faulty bit is cleared), since some SRAM bits may fail. Thus, the load instruction is allowed to proceed as speculative. Speculation is solved later as soon as the eDRAM replica is read and compared to the SRAM value. If the eDRAM replica is not valid, then the cache line is fetched from L2. If both SRAM and eDRAM values match, the load becomes non-speculative, and the processor continues its execution. On mispeculation, the load and subsequent instructions must be canceled by triggering the conventional recover mechanisms and the SRAM-faulty bit is set.

**Write Hit in the SRAM Way.** On a write hit event in the MRU block, a write is performed in the SRAM way (except if it has been detected as faulty) as well as in its eDRAM replica.

**Read/Write Hit in an eDRAM Way.** On a hit in an eDRAM way, the data must be copied from that way (that becomes the new eDRAM replica) to the SRAM way (in case of it is not faulty); thus, overwriting its contents. Notice that directly overwriting the SRAM way does not mean any loss of information, since the previous SRAM data had their replica. In case of write hit, both the SRAM way and the new replica are updated with the same data.

**Cache Miss.** On a cache miss, the block is fetched from L2 or main memory. The incoming block is written both in the SRAM way (MRU line) and in the eDRAM way (i.e., new replica) containing the victimized data identified by the LRU replacement algorithm.

## IV. Failure Probability in SRAM Cells

Manufacturing process produces variations in the transistor parameters mainly due to physical factors caused by processing and masking imperfections [16]. Variations affect the channel length, channel width, oxide thickness, threshold voltage, line-edge roughness, and random dopant fluctuations, and are typically classified in *inter-die* and *intra-die* variations.

Because of the small geometry of the SRAM cell, the main source of the device mismatch is the intrinsic fluctuation of the Vth of different transistors due to random dopant fluctuations [17], that is, random intra-die variations. Such device parameters mismatches severely affect SRAM cells in sub-50nm technologies [6].

These mismatches between the variations of close transistors caused by intra-die variations can result in the failure of the cell in four different ways when a voltage below $VCC_{min}$

is used: hold failure, read failure, write failure, and access failure. Below we discuss these types of failures.

**Hold Failure.** Each of the transistor pairs that forms an SRAM cell is referred to as a node. One of them contains a "1" and the other a "0". The voltage of the node storing "1" is the same as the power supply of the cell. When working at *lp* mode (i.e., reduced power supply), if the voltage of the node storing "1" is reduced below the trip-point[1] of the node storing "0" then a flip occurs, so loosing the stored value and producing a hold failure.

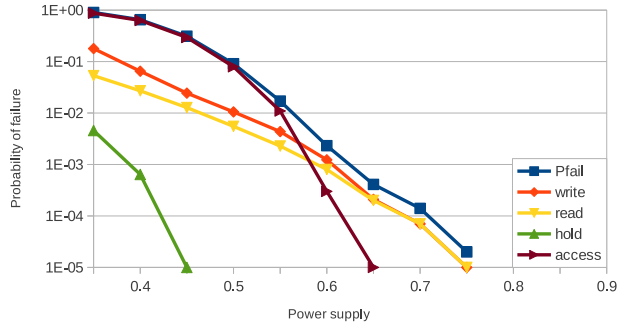
**Read Failure.** Before the read is performed, the bitline and its complementary are precharged to Vdd. When the wordline is activated, the pass-transistors communicate both bitlines with the nodes of the cell. Then, the node storing "0" discharges the associated bitline while the node storing "1" remains to Vdd. The voltage increases for a while in the node storing "0" to a positive value due to the voltage divider action. When this increase is greater than the trip-point of the node storing "1", a flip is produced, which is known as a read failure.

**Write Failure.** In a write operation, the bitline is precharged to "0" or "1" according to the value to be written. A write failure is produced when a "0" cannot be written in the cell. When the wordline is activated, the pass-transistor communicates the node storing "1" (Vdd) with the bitline (0V). To be a successful write operation, the node storing "1" must reduce the voltage below the trip-point of the node storing "0" while the wordline is active. Due to process variation, this decrease may be too slow. In other words, the time the wordline is active can be not long enough to decrease the voltage below the trip-point.

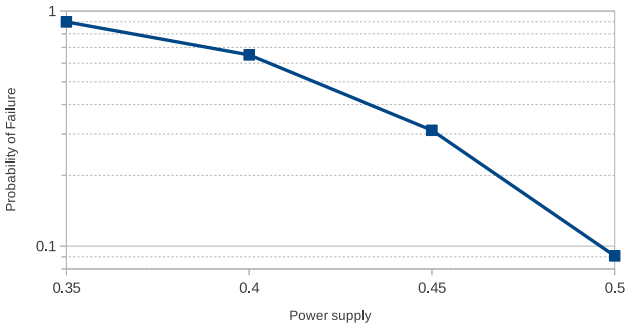
**Access Failure.** The cell access time is defined as the time required to produce the necessary voltage difference to excite the sense amplifier in a read operation. This voltage difference is typically by 10% of Vdd and must be reached while the wordline is active. To perform a read, both bitlines are precharged to Vdd and the bitline of the node storing "0" is discharged to 0V. The time needed to discharge that bitline depends on the pass transistor and the NMOS features. Due to process variation, the mismatch in these transistors can affect the discharging speed. If this process is too slow, the difference required to excite the sense amplifier can be not achieved.

SRAM cell failure probabilities have been estimated simulating the cell with the HSPICE circuit level simulator. Simulations assumed transistors based on 32nm nodes with high-performance profile from the Predictive Technology Model (PTM) [18]. We used the BSIM4 MOSFET model that addresses the MOSFET physical effects into the sub-100nm regime. Transistor sizes have been chosen according to [1] to ensure read and write-ability as well as to provide a good layout. Regarding area, device parameters (channel width W and channel length L) relationships (W/L) for the different types of transistors in the cell, access NMOS, pull-up PMOS, and pull-down NMOS were modeled as $6/2\lambda^2$, $4/2\lambda$, and $8/2\lambda$, respectively.

[1]The trip-point is the required voltage at the input of the node to change the output.




(a) Breakdown of SRAM failures




(b) Low-power range

Figure 2. SRAM cell failure probability for a 32nm technology node.

Intra-die random variations can be summarized as Vth fluctuations, which have been modeled for each transistor (NMOS and PMOS) of the cell as an independent Gaussian random variable with $\mu$ and $\sigma_{VT0}$ equal to 0 and 14%, respectively, and with 42% maximum Vth deviation [19]. The Monte Carlo simulation method was used to generate 100K samples of cells.

Figure 2(a) illustrates the failure probability for each type of SRAM failure. As discussed above, access and write failures appear because the time the wordline is active is not enough to perform the operation, while hold and read failures are time independent operations. Results show that for low-level voltages, the probability of failure (*Pfail*) is dominated by access failures, while in higher voltages Pfail is dominated by write and read failures. Notice that fault-free voltages are those higher than 0.75V. This work assumes that the *hp* mode has the associated typical supply voltage of 0.9V.

Figure 2(b) focuses on the range from 0.5V to 0.35V, which covers a significant Pfail range (from 9% to 90%). We will consider four different *lp* modes varying the voltage in steps of 0.05V. Table I summarizes the *hp* and *lp* modes with their associated voltage, processor frequency, and Pfail. The frequency values are similar to those considered in [20].

[2]$\lambda$ is defined as half the feature size (for 32nm nodes, $\lambda$=16nm).

Table I
OPERATION MODES WITH THEIR VOLTAGE, FREQUENCY, AND SRAM PROBABILITY OF FAILURE.

| Operation mode | *hp* | *lp1* | *lp2* | *lp3* | *lp4* |
|---|---|---|---|---|---|
| Voltage (V) | 0.90 | 0.50 | 0.45 | 0.40 | 0.35 |
| Frequency (MHz) | 3000 | 1000 | 800 | 600 | 400 |
| Pfail (%) | 0 | 9 | 31 | 65 | 90 |

## V. EXPERIMENTAL EVALUATION

The HER L1 data cache architecture has been modeled on top of an extensively modified version of the SimpleScalar (with Alpha ISA) simulation framework [21] to obtain the execution time and memory events (i.e., cache hits, misses, replacement, writebacks, and swaps) required to estimate the energy consumption. The CACTI [22] [23] simulator was used to obtain cache latencies, leakage, and dynamic energy per access type (e.g., read or write) for the different studied supply voltages and a 32nm technology node. Bank contention for all the memory events has been also modeled.

Table II summarizes the architectural parameters. A representative set of four memory intensive, four half-intensive, and four non-intensive applications from the SPEC CPU benchmark suite [24] was run using the *ref* input sets. Statistics were collected during 500M instructions after skipping the initial 1B instructions.

Notice that the access time (in processor cycles) depends on the voltage/frequency pairs of each operation mode and the latency of the type of bank where the data are located (SRAM or eDRAM banks). For all the *lp* operation modes these values are the minimum possible (i.e., 1 and 2-cycle when hitting the predicted SRAM way and the remaining eDRAM ways, respectively), because in these modes the processor cycle is much longer than the access times provided by CACTI.

For comparison purposes, a conventional SRAM L1 cache with the same cache organization and working at high-performance mode has been considered. Its access time matches that given for the SRAM banks of HER caches in *hp* mode.

Table II
ARCHITECTURAL MACHINE PARAMETERS.

| Microprocessor core | |
|---|---|
| Issue policy | Out of order |
| Branch predictor type | Hybrid gShare/Bimodal: gShare has 14-bit global history plus 16K 2-bit counters<br>Bimodal has 4K 2-bit counters, and choice predictor has 4K 2-bit counters |
| Branch predictor penalty | 10 cycles |
| Fetch, issue, commit width | 4 instructions/cycle |
| ROB size (entries) | 256 |
| # Int / FP ALUs | 4 / 4 |
| **Memory hierarchy** | |
| L1 data cache | 32KB-4way, 64 B-line, 8 banks (2 SRAM and 6 eDRAM) |
| L1 data cache access time | *hp* mode: 2-cycle SRAM and 4-cycle eDRAM<br>*lp* modes: 1-cycle SRAM and 2-cycle eDRAM |
| L2 unified cache | 512KB-8way, 64 B-line |
| L2 cache access time | 10 cycles |
| Main Memory access time | 100 cycles |

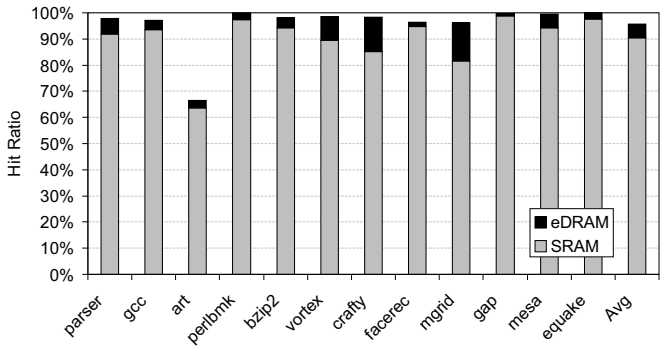



Figure 3. SRAM and eDRAM hit ratio per benchmark for the HER cache in *hp* mode.

### A. Performance

This section evaluates the hit ratio of the HER cache and its impact on performance. Then, the IPC losses of the HER cache with respect to the conventional design are analyzed.

First, the hit ratio for each application in *hp* mode is analyzed for comparison purposes with the *lp* modes. Figure 3 plots the results. The hit ratio is broken down into hits in SRAM and hits in eDRAM banks. In general, the overall hit ratio is above 90%, and thanks to the swap operation, most of the hits concentrate on the *fast* cache way storing the MRU line (i.e., SRAM banks). In contrast, the eDRAM hit ratio is only by 5.4% on average. Moreover, we found that the overall hit ratio matches that of the conventional cache since the retention time (see Table IV) is large enough to avoid accessing data that have been previously invalidated by the scrub operation (see Section III-A).

As the voltage is reduced, the number of SRAM errors rises, which impacts on the hit ratio. Table III shows the average hit ratio across the studied *lp* operation modes. In this case, the hit ratio also includes hits in the eDRAM replica. The SRAM hit ratio in *lp* decreases with the probability of failure, while the eDRAM replica hit ratio increases since more accesses concentrate on the replicas. The scarce total hit ratio differences of *lp1* and *lp2* modes with respect to *hp* appear because the effective cache capacity becomes smaller due to replicas. These differences are larger in both *lp3* and *lp4* modes since the retention time is shorter, which in turn induces data losses because the scrub operation is more often applied.

Table III
HIT RATIO (%) OF THE HER CACHE IN THE ANALYZED OPERATION MODES.

| Hit ratio (%) | *hp* | *lp1* | *lp2* | *lp3* | *lp4* |
|---|---|---|---|---|---|
| SRAM | 90.4 | 80.6 | 51.8 | 25.2 | 9.3 |
| eDRAM | 5.4 | 5.1 | 5.1 | 5.1 | 5.0 |
| eDRAM replica | 0 | 9.8 | 38.6 | 65.0 | 80.7 |
| Overall | 95.8 | 95.5 | 95.5 | 95.3 | 95.0 |

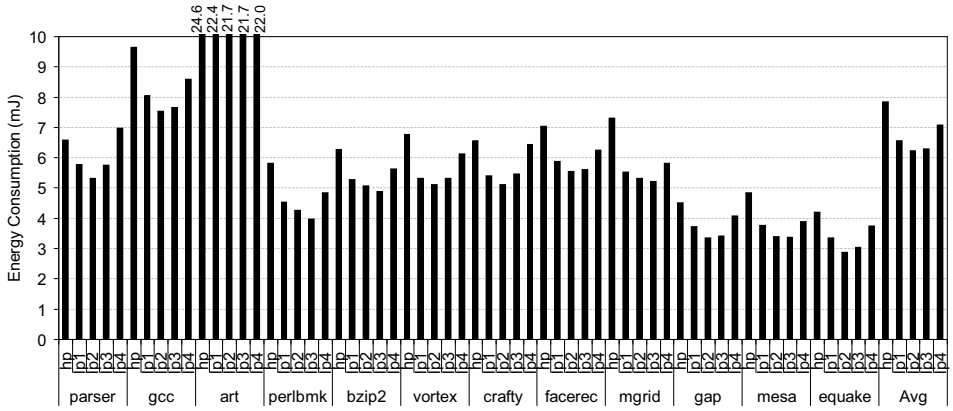



Figure 4. Energy consumption (in mJ) of the HER cache for each benchmark.

Table IV
RETENTION TIME (PROCESSOR CYCLES), IPC LOSSES (%) IN ABSOLUTE PROCESSOR CYCLES, AND NORMALIZED EXECUTION TIME OF THE HER CACHE WITH RESPECT TO THE CONVENTIONAL SRAM CACHE IN *hp* MODE.

| Operation mode | Retention time (cycles) | IPC degradation (%) | Normalized execution time |
|---|---|---|---|
| *hp* | 321429 | 1.16 | 1.01 |
| *lp1* | 59523 | 2.52 | 3.07 |
| *lp2* | 42856 | 2.72 | 3.84 |
| *lp3* | 28571 | 3.56 | 5.17 |
| *lp4* | 16666 | 4.08 | 7.80 |

Table IV summarizes the IPC degradation and the normalized execution time compared to the conventional cache working at *hp* mode. This cache represents an upper bound since it does not use way-prediction nor it is implemented with *slower* eDRAM banks. Besides, retention time values (in processor cycles) are also presented.

Note that the retention time becomes shorter with lower voltages and frequencies because capacitors are charged with less voltage and the cycle time increases. IPC losses increase in the most defective operation modes mainly due to the fact that more accesses concentrate on *slow* replicas. Similarly, the normalized execution time also increases with lower voltage/frequency pairs. In this case, differences are more noticeable due to the execution time is affected by the processor frequency (i.e., lower frequencies imply larger execution time).

Finally, results also show the effectiveness of the HER cache, since the performance degradation working at high-performance mode is minimal with respect to the conventional design.

### *B. Energy Consumption*

The aim of this section is to find out the best frequency/voltage pair, that is the best operation mode, regarding L1 energy consumption. For this purpose, this section analyzes the energy consumed when running the studied benchmarks in the different *lp* modes.

Figure 4 shows the total energy results (in mJ) of the HER cache for each benchmark and operation modes. In most of the applications (8 of 12) like *parser* and *crafty*, *lp2* (0.45V/800MHz) is the operation mode with the lowest overall energy consumption. In fact, this mode obtains the lowest average value, closely followed by *lp3* (0.40V/600MHz). Thus, although lowering the supply voltage from 0.45V down to 0.40V and the operating frequency from 800MHz to 600MHz seems an intuitive way to reduce overall energy consumption, the performance degradation incurred in *lp3* (see Table IV) diminishes the energy benefits of this choice, and even produces the contrary effect in some benchmarks (e.g., *parser*, *vortex*, *crafty*, and *equake*). This negative effect is much more magnified when running in *lp4* (0.35V/400MHz) mode. On average, this mode increases energy consumption by 15% with respect to *lp2*.

To provide insights on the increase of energy consumption incurred by the lowest power modes, Figure 5 depicts the average consumption for each operation mode distinguishing between leakage and dynamic energy. Leakage expenses have been accounted for cycle by cycle considering the tag and data arrays, whereas dynamic energy has been divided into seven categories according to the different cache events: SRAM hits, eDRAM hits, eDRAM replica hits, swaps, writebacks, misses, and tag array. The SRAM hits category includes the consumption of accessing both the SRAM way and the replica; the eDRAM hits expenses consider the access to the predicted SRAM way and the target eDRAM way; the eDRAM replica hits category also includes the consumption of the previously accessed SRAM faulty way; the consumption of the swap operation has been calculated as the sum of a read access to the SRAM banks, a read and a write access to an eDRAM bank, and a write access to an SRAM bank. The writebacks and misses include the energy consumed by both L1 and L2 cache accesses; and finally, the tag array energy is accounted on each cache access.

The different components of the energy consumption widely differ among *lp* modes. Regarding leakage, despite this energy is proportional to the supply voltage, it increases as voltage falls because execution time is enlarged due to lower frequencies and higher performance degradation (see Table IV). On the other hand, dynamic energy decreases with lower voltages

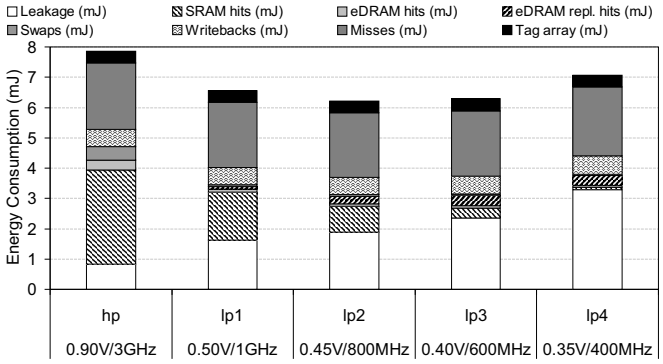



Figure 5. *Categorized* average energy consumption (in mJ).

because it is proportional to the squared supply voltage. The most noticeable effect is the decrease of SRAM hit energy with lower voltage/frequency pairs, since the probability of failure increases and more accesses are performed to the replicas. The consumption of accessing the replicas increases with the faulty lines, while the eDRAM hits, swaps, and tag array energy represent a small fraction of the overall consumption. Differences among the remaining components (i.e., misses and writebacks) are due to the cache capacity is reduced. This mainly occurs in the more defective modes (i.e., *lp3* and *lp4*) because of data losses. There is an extra amount of writebacks due to scrubbing, whose maximum impact can be observed in the *lp4* mode, where the writeback energy is by 0.63mJ.

In summary, dynamic consumption is reduced with lower voltage/frequency pairs, while leakage steadily increases. These effects make the *lp2* mode consume less overall energy than the other operation modes.

Finally, regarding the *hp* mode, the HER cache consumes half the overall energy of the conventional cache (not shown for simplification purposes). Leakage currents and dynamic energy are significantly reduced mainly because of the use of eDRAM cells and way-prediction, respectively. Compared to the conventional cache, the HER cache working at *lp2* mode reduces the overall energy consumption on average by 62%.

## VI. Conclusions

This paper has presented an evaluation methodology that analyzes the impact of error detection and correction proposals on energy consumption when the processor works at low-power modes to save energy. The devised method is aimed at identifying the optimal voltage/frequency pair in fault-tolerant cache approaches that brings more energy savings.

The evaluation has focused on the recently proposed fault-tolerant HER cache that provides 100% SRAM fault tolerance, although the devised methodology can be applied to any existing error correction scheme.

Contrary to expected, energy does not always decrease as the voltage is reduced because existing fault-tolerant proposals trade off coverage with performance. Such a performance penalty enlarges the execution time of the programs, which adversely impacts on energy. Moreover, low voltages are paired with low frequencies, which also extend the execution time.

Experimental results have shown that for a 32nm technology node, the 0.45V/800Mhz voltage/frequency pair, which has by 31% of SRAM faulty cells, is the most efficient in terms of energy consumption. The overall leakage and dynamic energy is largely reduced (by 62% on average) with respect to a conventional SRAM cache working at high-performance mode.

## Acknowledgments

This work was supported by the Spanish *Ministerio de Economía y Competitividad* (MINECO) and FEDER funds under Grant TIN2012-38341-C04-01. Additionally, it was also supported by the Intel Early Career Honor Programme Award and the Intel Doctoral Student Honor Programme Award.

## References

[1] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859–1880, 2005.

[2] S. E. Schuster, "Multiple Word/Bit Line Redundancy for Semiconductor Memories," *IEEE Journal of Solid-State Circuits*, vol. 13, no. 5, pp. 698–703, 1978.

[3] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, 2008, pp. 203–214.

[4] S. Paul, F. Cai, X. Zhang, and S. Bhunia, "Reliability-Driven ECC Allocation for Multiple Bit Error Resilience in Processor Cache," *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 20–34, 2011.

[5] A. R. Alameldeen, Z. Chishti, C. Wilkerson, W. Wu, and S.-L. Lu, "Adaptive Cache Design to Enable Reliable Low-Voltage Operation," *IEEE Transactions on Computers*, vol. 60, pp. 50–63, 2011.

[6] A. Agarwal, B. C. Paul, S. Mukhopadhyay, and K. Roy, "Process Variation in Embedded Memories: Failure Analysis and Variation Aware Architecture," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1804–1814, 2005.

[7] R. G. Dreslinski, G. K. Chen, T. Mudge, D. Blaauw, D. Sylvester, and K. Flautner, "Reconfigurable Energy Efficient Near Threshold Cache Architectures," in *Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture*, 2008, pp. 459–470.

[8] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S.-L. Lu, "Reducing Cache Power with Low-Cost, Multi-bit Error-Correcting Codes," in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, 2010, pp. 83–93.

[9] R. E. Matick and S. E. Schuster, "Logic-based eDRAM: Origins and rationale for use," *IBM Journal of Research and Development*, vol. 49, no. 1, pp. 145–165, 2005.

[10] B. Sinharoy, R. Kalla, W. J. Strake, H. Le, R. Cargnoni, J. A. V. Nostrand, B. J. Stuecheli, J. Leenstra, G. L. Guthrie, D. Q. Nguyen, B. Blaner, C. F. Marino, E. Retter, and P. Williams, "IBM POWER7 multicore server processor," *IBM Journal of Research and Development*, vol. 55, no. 3, pp. 1–29, 2011.

[11] J. Stuecheli, "POWER8," *Hot Chips*, 2013.

[12] V. Lorente, A. Valero, J. Sahuquillo, S. Petit, R. Canal, P. López, and J. Duato, "Combining RAM technologies for hard-error recovery in L1 data caches working at very-low power modes," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2013, pp. 83–88.

[13] A. Bardine, M. Comparetti, P. Foglia, and C. A. Prete, "Evaluation of Leakage Reduction Alternatives for Deep Submicron Dynamic Nonuniform Cache Architecture Caches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 185–190, 2014.

[14] L. Chang, D. M. Fried, J. Hergenrother, J. W. Sleight, R. H. Dennard, R. K. Montoye, L. Sekaric, S. J. McNab, A. W. Topol, C. D. Adams, K. W. Guarini, and W. Haensch, "Stable SRAM cell design for the 32 nm node and beyond," in *Symposium on VLSI Technology. Digest of Technical Papers*, 2005, pp. 128–129.

[15] S. Petit, J. Sahuquillo, J. M. Such, and D. Kaeli, "Exploiting Temporal Locality in Drowsy Cache Policies," *Proceedings of the 2nd Conference on Computing Frontiers*, pp. 371–377, 2005.

[16] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations," in *IEEE Conference on Custom Integrated Circuits*, 2001, pp. 223–228.

[17] A. J. Bhavnagarwala, X. Tang, and J. D. Meindl, "The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 4, pp. 658–665, 2001.

[18] W. Zhao and Y. Cao, "Predictive Technology Model for Nano-CMOS Design Exploration," *Journal on Emerging Technologies in Computing Systems*, vol. 3, no. 1, pp. 1–17, 2007.

[19] ITRS, *Semiconductor Industries Association, International Technology Roadmap for Semiconductors, available online at http://www.itrs.net/.*, 2011.

[20] P. Chaparro, "Thermal Aware Microarchitectures," *Ph.D Thesis, Universitat Politècnica de Catalunya*, 2008.

[21] D. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0," *ACM SIGARCH Computer Architecture News*, vol. 25, no. 3, pp. 13–25, 1997.

[22] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "CACTI 5.1," *Technical Report, Hewlett-Packard Laboratories, Palo Alto*, 2008.

[23] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi, "A Comprehensive Memory Modeling Tool and its Application to the Design and Analysis of Future Memory Hierarchies," *Proceedings of the 35th Annual International Symposium on Computer Architecture*, pp. 51–62, 2008.

[24] SPEC, *Standard Performance Evaluation Corporation, available online at http://www.spec.org/cpu2000/.*