Valero Bresó, A.; Sahuquillo Borrás, J.; Lorente Garcés, VJ.; Petit Martí, SV.; López Rodríguez, PJ.; Duato Marín, JF. (2012). Impact on performance and energy of the retention time and processor frequency in L1 macrocell-based data caches. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 20(6):1108-1117. doi:10.1109/TVLSI.2011.2142202.

Impact on Performance and Energy of the Retention Time and Processor Frequency in L1 Macrocell-Based Data Caches

Alejandro Valero, Julio Sahuquillo, *Member, IEEE*, Vicente Lorente, Salvador Petit, *Member, IEEE*, Pedro López, *Member, IEEE*, and José Duato

***Abstract*—Cache memories dissipate an important amount of the energy budget in current microprocessors. This is mainly due to cache cells are typically implemented with six transistors. To tackle this design concern, recent research has focused on the proposal of new cache cells. An n-bit cache cell, namely macrocell, has been proposed in a previous work. This cell combines SRAM and eDRAM technologies with the aim of reducing energy consumption while maintaining the performance. The capacitance of eDRAM cells impacts on energy consumption and performance since these cells lose their state once the retention time expires. On such a case, data must be fetched from a lower level of the memory hierarchy, so negatively impacting on performance and energy consumption. As opposite, if the capacitance is too high, energy would be wasted without bringing performance benefits. This paper identifies the optimal capacitance for a given processor frequency. To this end, the tradeoff between performance and energy consumption of a macrocell-based cache has been evaluated varying the capacitance and frequency. Experimental results show that, compared to a conventional cache, performance losses are lower than 2% and energy savings are up to 55% for a cache with 10 fF capacitors and frequencies higher than 1 GHz. In addition, using trench capacitors, a 4-bit macrocell reduces by 29% the area of four conventional SRAM cells.**

***Index Terms*—eDRAM memory cells, eDRAM capacitance, energy consumption, retention time, SRAM memory cells.**

I. Introduction

CACHE memory cells have been typically implemented in microprocessor systems using static random access memory (SRAM) technology because it provides fast access time and does not require refresh operations. SRAM cells are usually implemented with six transistors (6T cells). The major drawbacks of SRAM caches are that they occupy a significant percentage of the overall die area and consume an important amount of energy, specially leakage energy which is proportional to the number of transistors. Furthermore, this problem is expected to aggravate with future technology generations that will continue shrinking the transistor size.

Leakage energy can be reduced by using alternative technologies like Dynamic RAM (DRAM), which is typically used for main memory. Unlike SRAM cells, DRAM cells only require an active power supply during the memory access so their leakage currents are reduced by design. These cells require less area than SRAM cells since they are implemented by using only one capacitor and the corresponding pass transistor (1T-1C cells).

DRAM cells have been considered too slow for processor caches. Nevertheless, technology advances have recently allowed to embed DRAM cells using CMOS technology [1]. An embedded DRAM cell (eDRAM) integrates a trench DRAM storage cell into a logic-circuit technology and provides similar access delays as those presented by SRAM cells. As a consequence, some recent commercial processors use eDRAM technology to implement last-level caches [2], [3].

In an eDRAM cache design, the capacitance of the cells (i.e., the amount of electrical energy that can be stored in the eDRAM capacitor) impacts on performance and dynamic energy consumption. The reason is that capacitors lose their charge with time. That is, after a given period of time, which depends on the capacitance, the data information stored in the eDRAM cell cannot be retrieved any longer. This time is referred to as the cell retention time. If the capacitance is fixed too low (i.e., too short retention time), each time the retention time expires, an access from the processor to the data will result in a miss, that is, the processor will fetch the data block from a lower level of the memory hierarchy (e.g., L2 cache), which might negatively impact on performance and energy consumption. As opposite, if the capacitor charge is fixed too high, energy is wasted without bringing performance benefits.

In [4], it has been proposed an n-bit memory cell (from now on macrocell) with one SRAM cell and $n-1$ eDRAM cells designed to implement n-way set-associative caches. SRAM cells provide low latencies and availability while eDRAM cells allow to reduce leakage consumption and increase memory density. Results showed that a retention time around 50 K processor cycles was long enough to avoid performance losses in an L1 macrocell-based data cache (from now on M-Cache).

In this work, we perform a further step in the design of the macrocell, identifying the optimal eDRAM capacitance for the frequency at which the processor works. To this end, a detailed analysis of the energy consumed by caches for a 45-nm technology node has been performed. Experimental results show

Manuscript received August 09, 2010; revised January 05, 2011; accepted March 18, 2011. This work was supported in part by Spanish CICYT under Grant TIN2009-14475-C04-01, by Consolider-Ingenio 2010 under Grant CSD2006-00046, and by European community's Seventh Framework Programme (FP7/2007-2013) under Grant 289154.

The authors are with the Department of Computer Engineering (DISCA), Universitat Politècnica de València, Valencia 46022, Spain (e-mail: alvabre@gap.upv.es; jsahuqui@disca.upv.es; vlorente@disca.upv.es; spetit@disca.upv.es; plopez@disca.upv.es; jduato@disca.upv.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2011.2142202

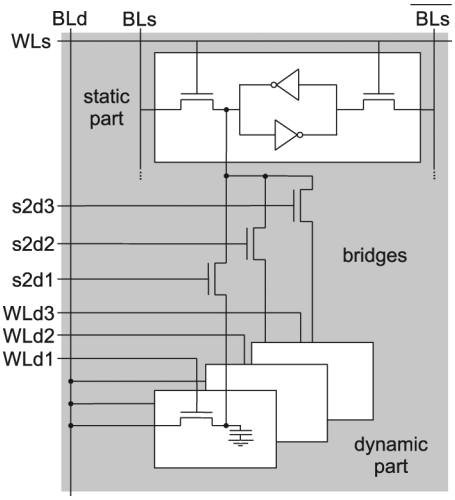

Fig. 1. 4-bit macrocell block diagram.

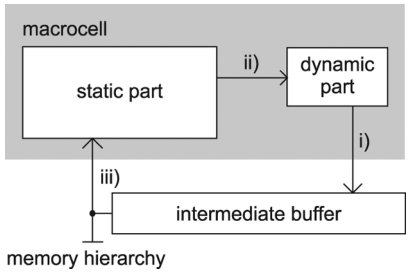

Fig. 2. Swap operation.

that a 10 fF capacitor is enough to avoid performance losses for processor frequencies higher than 1 GHz. Energy consumption is largely reduced with respect to a conventional SRAM cache with the same capacity. For instance, for a 4-way set-associative M-Cache, energy consumption can be reduced to the half of a conventional cache. Furthermore, a 4-bit macrocell implemented with trench capacitors provide an area reduction by about 29% compared to four conventional SRAM cells, regardless the capacitance of the cell.

The rest of this paper is organized as follows. Section II summarizes the macrocell design, its working behavior and presents a possible implementation of an M-Cache. Section III analyzes how the eDRAM capacitance impacts on retention and access times and estimates the area savings of the macrocell. Section IV shows performance and energy results for different cache organizations. Section V summarizes the related work. Finally, some conclusions are drawn.

II. Macrocell-Based Caches (M-Caches)

This section summarizes the macrocell behavior and describes how a set-associative cache can be implemented and accessed by using the proposed circuit.

A. Macrocell Internals

The main components of an n-bit macrocell are a typical SRAM cell, $n-1$ eDRAM cells and *bridge* transistors that communicate SRAM with eDRAM cells. Fig. 1 depicts an implementation of a 4-bit macrocell. The SRAM cell comprises the *static* part of the macrocell. Read and write operations in this part are managed in the same way as in a typical SRAM cell through the bitline signals (*BLs* and */BLs*).

The *dynamic* part is formed by $n-1$ eDRAM cells (three in the example). Each eDRAM cell consists of a capacitor and an NMOS pass transistor, controlled by the corresponding wordline signal (WLd_i). Wordlines allow each capacitor to be accessed through the corresponding bitline (*BLd*). Read and write operations perform as in a conventional eDRAM cell through the corresponding pass transistor.

The bridge transistors connect the SRAM cell to each eDRAM cell and are controlled by the corresponding *static to dynamic* signal $(s2d_i)$. Each bridge transistor acts as an unidirectional path to transfer data from the SRAM cell to a given eDRAM cell without using intermediate buffers. These transfers are referred to as internal since no bitline is involved.

Internal transfers provide a fast and low energy consumption mechanism to copy data stored in the SRAM cell to the dynamic part. The idea is to keep the data most recently used (MRU) by the processor always in the SRAM cell. Previous works [5] have shown that the MRU line in each cache set use to be accessed with a much higher probability than the remaining ones (for instance, 92.15% of the accesses in a 16 kB-4 way L1). Therefore, keeping the MRU data in the SRAM cell might provide energy benefits because SRAM reads are non-destructive. The remaining data is stored in the dynamic part which loses their state when the retention time expires. This will happen if this data is not referenced for long.

Internal transfers are triggered when the data required by the processor is not stored in the SRAM cell (SRAM miss). In this case, the content of the SRAM cell is copied to an eDRAM cell and replaced by the referenced data. To do so, the cache controller manages a swap operation in three sequential steps as shown in Fig. 2. First, the data stored in the dynamic part is written to an intermediate buffer, then an internal transfer is performed to copy the data from the static to the dynamic part by switching on the corresponding bridge transistor and after that, the data referenced by the processor is written to the static part. This data can come either from the intermediate buffer (i.e., on an eDRAM hit) or from a lower level of the memory hierarchy (i.e., on a cache miss).

In order to ensure that internal transfers work properly, the macrocell has been modeled with NGSPICE (a Berkeley's Spice3f5-based circuit simulator). NGSPICE allows to accurately simulate MOSFET behavior since it uses a BSIM4 MOSFET model. All the simulations are based on Predictive Technology Models (PTM) [6]. Values of key technology metrics of transistors have been taken from the 2007 ITRS [7] for 45-nm technology node.

In particular, we have evaluated: 1) whether the eDRAM capacitor is properly charged and 2) the absence of flips in the SRAM cell. Regarding the former issue, the pass transistor of a typical 1T-1C cell leads to a *Vth* voltage drop when transferring a logic "1" to the capacitor, which suffers a charge degradation. Therefore, in the macrocell, in order to guarantee that the capacitor is fully charged to the maximum *Vdd* voltage value, the

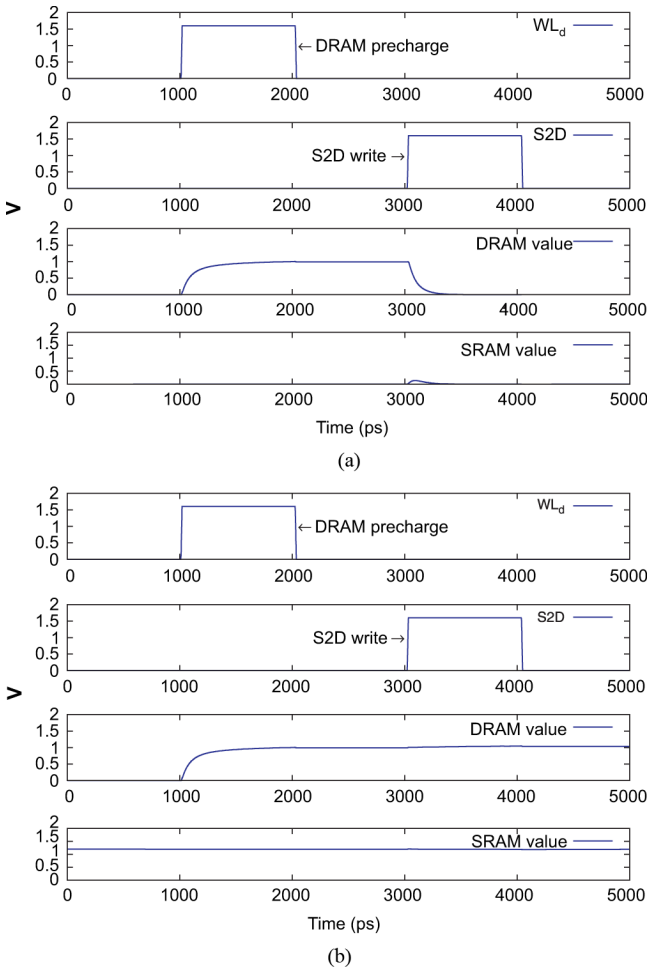

Fig. 3. Operation detail of the internal transfer. (a) Transfer a logic "0". (b) Transfer a logic "1".

wordline controlled by WLd_i must be boosted to a *Vpp* voltage (i.e., $\mathrm{Vpp} = \mathrm{Vdd} + \mathrm{Vth}$). Furthermore, in the case of the bridge transistor, the wordline controlled by $s2d_i$ must also be boosted to the same voltage. For eDRAM cells and 45-nm technology node, Vth and Vdd are usually set to 0.4 and 1.1 V, respectively.

Regarding the latter issue, the bitlines of the conventional SRAM cell must be precharged before a read operation in order to optimize the cell speed, area and stability relationship (e.g., the static-noise margin) [8]. This is mainly due to the differences between nMOS and pMOS transistor features. In this way, flips are avoided inside the SRAM cell, since nMOS transistors can drive more current than pMOS transistors. In accordance with the macrocell design, the capacitor of the eDRAM cell must also be precharged to Vdd to prevent flips. Fig. 3 illustrates how internal transfer works and highlights both the precharge process and how flips are avoided (both transfers writing a "0" and a "1", respectively). The performance and energy consumption penalties due to both design issues have been taken into account in the experimental results.

B. Accessing the M-Cache

The number of bits in a macrocell device defines the number of ways of the implemented M-Cache. In other words, n-bit macrocells are required to build an n-way set-associative cache.

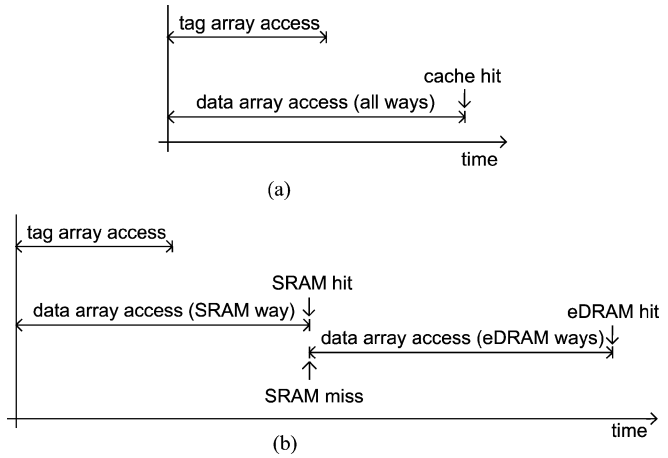

Fig. 4. Tag and data arrays access timing diagram. (a) Conventional cache. (b) M-Cache.

Hence, an n-way set-associative cache will have one way implemented with SRAM cells (the SRAM way) and the remaining $n-1$ ways with eDRAM cells (eDRAM ways).

Conventional caches use to access in parallel the tag and the data arrays, as shown in Fig. 4(a). Proceeding in this way in an M-Cache would result in energy wasting, since eDRAM cell reads are destructive. Thus, all the eDRAM cells in a set should be recharged each time the set is accessed regardless if the access is a hit or not. In other words, in an n-way set-associative cache, only one way can contain the requested data but, if all ways were accessed at the same time, the capacitors of the eDRAM ways should be recharged even if the access results in a hit in the SRAM way or in a cache miss.

To reduce energy consumption due to capacitor recharges, tag and data arrays are treated differently. The tag array is assumed to be implemented with typical SRAM cells, since it is much smaller than the data array (i.e., much lower energy and area reduction can be achieved) and all tags in the set need to be looked up to check whether the requested data is in cache or not. Thus, reading tags is not destructive and we focus on avoiding energy wasting due to unproductive capacitor reads in the data array. To this end, the cache works like a way prediction cache [9], [10]. That is, the tags of all ways in the set and the data of the SRAM way (i.e., the predicted way) are accessed first [see Fig. 4(b)]. On a hit, the cache presents two different lantencies depending on whether the requested data is in the SRAM or in an eDRAM way. A hit in the SRAM way results in an access time as fast as a hit in a direct-mapped cache and no eDRAM cell will subsequently be accessed. As opposite, a hit in an eDRAM way provides a slower access time since the corresponding eDRAM way must be subsequently accessed. In addition, in this case the aforementioned swap operation must be triggered after the processor gets the requested data. Notice that by combining the way prediction technique and the swap operation (explained in Section II), refresh is not required in the M-Cache. The former ensures that the *eDRAM* data is only read on an *eDRAM* hit and the latter accomplishes an implicit refresh of this eDRAM data by transferring it to SRAM cells.

Finally, in order to ensure that data stored on the eDRAM cells is valid (i.e., the capacitor has not been discharged), the

TABLE I
RETENTION TIME AND PERFORMANCE DEGRADATION. RETENTION TIME VALUES EXCEEDING 50 K CYCLES ARE SHOWN IN BOLDFACE (a) RETENTION TIME (CYCLES) (b) PERFORMANCE DEGRADATION (%) FOR A 32 kB 4-WAY M-CACHE WHEN RUNNING INTEGER BENCHMARKS

	eDRAM capacitance (fF)										
Frequency	0.04	0.08	0.16	0.31	0.63	1.25	2.5	5	10	20	∞
1GHz	248	496	992	1983	3966	7933	15865	31731	**63462**	**126923**	∞
2GHz	496	992	1983	3966	7933	15865	31731	**63462**	**126923**	**253846**	∞
3GHz	744	1487	2975	5950	11899	23798	47596	**95192**	**190385**	**380769**	∞

(a)

	eDRAM capacitance (fF)										
Frequency	0.04	0.08	0.16	0.31	0.63	1.25	2.5	5	10	20	∞
1GHz	1.48	1.26	0.83	0.55	0.33	0.22	0.11	0.06	0	0	0

(b)

M-Cache uses a sentry bit per eDRAM line. This bit is independent from the valid bit associated to the tag array and is implemented as a 1T-1C cell per cache line. Both sentry bit and macrocell capacitors must be charged at the same time when the data is stored. By choosing a lower capacitance in the sentry bit, the design ensures that if the sentry bit is interpreted as "1" (valid) the value of the associated data will be correct (i.e., the eDRAM capacitor has not been discharged yet). On the contrary, if the sentry bit is invalid, the macrocell may still contain valid data (as it uses a higher capacitance, with a higher retention time) but the design will conservatively assume that the data has been expired. As there is a sentry bit per cache line, there is a negligible hardware complexity overhead associated to the sentry bit. Nevertheless, the performance and energy penalties related to the sentry bit have been taken into account in the results.

III. TIMING AND AREA DETAILS

This section analyzes the retention time and estimates the access time of the macrocell for different capacitances and processor frequencies as well as the area of the n-bit macrocell compared to n conventional SRAM cells. The results presented in this section were calculated with the CACTI 5.3 tool [11], [12], which includes an analytical model for the leakage power, area, access time and dynamic power of caches and other memories. Then, the obtained values of the retention time and access time were used to feed the Hotleakage simulator [13], which is a cycle-by-cycle trace-driven simulator that implements the microarchitecture of a superscalar processor. This simulator was used to evaluate performance and energy consumption.

A. Retention Time

Retention time values for each eDRAM capacitance were quantified in processor cycles for three different machine speeds (1, 2, and 3 GHz) resembling those of existing commercial processors. To estimate the capacitor charge for each processor speed, a discrete amount of values ranging from 0.04 to 20 fF were analyzed. Table I(a) shows the retention time. In [4] it was shown that a 50 K-cycle retention time was enough to avoid performance losses in an M-Cache compared to an M-Cache with large capacitors (i.e., with an *infinite*-ideal retention time) when running the SPEC2000 benchmarks [14] in a superscalar processor with the same machine parameters as the assumed in this work. Thus, eDRAM data which is not referenced beyond this 50 K processor cycles has a very low probability to be referenced again. Results show that the minimum capacitances required to sustain a 50 K-cycle retention time are 10 fF for 1 GHz and 5 fF for 2 and 3 GHz. Table I(b) shows the performance degradation obtained for a 32 kB-4 way M-Cache with 1 GHz processor speed for integer benchmarks. It can be seen that 10 fF is enough to completely avoid performance losses (see Section IV-A).

B. Access Time

The cache access time depends on the cache geometry (cache size, line size and number of ways) and on the way the cache is accessed as well (e.g., conventional or way prediction). Therefore, for comparison purposes, in addition to the M-Cache, we also modeled a conventional cache (referred to as Conv-Cache) and a conventional way prediction cache (from now on WP-Cache) that always access the MRU line of a set as the predicted line. These three cache schemes have been compared across four different cache organizations varying the cache size (16 and 32 kB) and the number of ways (2 and 4).

A cache access internally involves the access to two main components: the tag and the data array. The access to the data array is composed of several latencies. First, the request has to be routed to the bank containing the requested data; then, the corresponding signals must traverse the row decoder, the bitlines, and the sense amplifiers. In contrast, the tag array is a much simpler structure with a shorter access time. The access to both array structures is performed in parallel in the Conv-Cache while it differs when the cache is accessed in a way prediction technique (M-Cache and WP-Cache schemes). In the latter case, two hit times can be distinguished depending on whether there is a hit in the predicted way or a hit in the remaining ones, where one or several additional cycles can be required.

Table II displays the access time quantified in processor cycles for the analyzed cache schemes. A pair of times has been used to indicate the access times of the way prediction cache schemes. The first value refers to the hit time in the predicted way and the second one to the additional cycles that would be required on a hit in any of the remaining ways. For instance, in the case of a 16 kB-2 way M-Cache with 1 GHz frequency, the access time is (1,+1) which means 1 and 2 cycles for a hit in the SRAM and eDRAM ways, respectively. Notice that the access time can be higher in the M-Cache than in the WP-Cache. This

TABLE II
ACCESS TIME (CYCLES) FOR THE STUDIED CACHE SCHEMES

	Conv-Cache			WP-Cache			M-Cache		
Cache Organization	1GHz	2GHz	3GHz	1GHz	2GHz	3GHz	1GHz	2GHz	3GHz
16KB-2w	1	2	2	(1,+0)	(2,+0)	(2,+1)	(1,+1)	(2,+1)	(2,+2)
16KB-4w	1	2	2	(1,+0)	(2,+0)	(2,+1)	(1,+1)	(2,+1)	(2,+2)
32KB-2w	1	2	3	(1,+1)	(2,+1)	(3,+1)	(1,+1)	(2,+2)	(3,+2)
32KB-4w	1	2	3	(1,+1)	(2,+1)	(3,+1)	(1,+1)	(2,+2)	(3,+2)

TABLE III
CELL AREAS (μm^2) AND REDUCTION FOR DIFFERENT MACROCELL SIZES

# data bits	SRAM	eDRAM	Macrocell	Reduction (%)
1-bit	0.30	0.06	-	-
2-bit	0.59	0.12	0.57	4
4-bit	1.18	0.25	0.84	29

TABLE IV
MACHINE PARAMETERS

Microprocessor core	
Issue policy	Out of order
Branch predictor type	Hybrid gShare/bimodal, 10-cycle penalty gShare: 14-bit history+16K 2-bit counters Bimodal: 4K 2-bit counters Choice predictor: 4K 2-bit counters
Fetch, issue, commit	4 instructions/cycle
ROB size (entries)	256
# ALUs	4 Int, 4 FP
Memory hierarchy	
Memory ports	4
L1 data cache	Variable geometry. 64 byte-line Latency: see Table II
L2 data cache	512KB, 8 ways, 64 byte-line, 10 cycles
Main memory	100 cycles

happens because the access time of eDRAM ways (in ns) is a bit higher than non-predicted ways of the WP-Cache.

Finally, it should be stated that the higher the capacitance, the higher the access time of the eDRAM ways. However, for the capacitances analyzed in this work (i.e., from 0.04 to 20 fF), this fact has negligible impact since it becomes masked when the access time is quantified in processor cycles.

C. Area

The n-bit macrocell saves area compared to n conventional cells (i.e., same number of data bits) since the former is partly implemented with eDRAM cells. The macrocell area savings for a 45-nm technology node were obtained with CACTI. The width and height of the n-bit macrocell have been estimated by accumulating those values of the SRAM and eDRAM cells in a pessimistic design where the area of each bridge transistor has been assumed to be the same as a 1T-1C cell.

The eDRAM cells of the macrocell are assumed to use trench storage capacitors. These capacitors etch deep holes into the wafer and are formed in the silicon substrate instead of above it. The deeper the hole the higher the capacitance. Thus, the cell area is not affected by the capacitance value [15]. Indeed, as shown in [16], the capacitance values analyzed in this work can be obtained with trench capacitors.

Table III shows the area (in μm^2) of the conventional SRAM and eDRAM cells and the macrocell varying the number of data bits from 1- to 4-bit. As expected, the area reduction increases with the number of bits. The 2- and 4-bit macrocells obtain an area reduction of 4% and 29% compared to 2 and 4 SRAM cells, respectively. Notice that the macrocell design does not allow a 1-bit macrocell.

IV. PERFORMANCE AND ENERGY EVALUATION

Both dynamic and leakage energy have been analyzed in this work for a 45-nm technology node. The corresponding values for each cache scheme were obtained per access and per processor cycle for dynamic and leakage energy, respectively, with CACTI. In addition, the Hotleakage framework was extended to accurately model the different cache schemes in order to obtain the overall execution time of a given workload and some statistics for specific cache events that are used to estimate the total leakage and dynamic energy, respectively. Notice that side effects due to the increase of L2 dynamic energy have been taken into account as well. For example, the access to an eDRAM cell that has lost its data triggers a L1 cache miss and a subsequent access to L2 that would not happen in a conventional cache.

Experiments have been performed configuring the Hotleakage for the Alpha ISA using the SPEC2000 benchmark suite. Both integer (Int) and floating-point (FP) benchmarks were run using the *ref* input sets and statistics were collected simulating 500 M instructions after skipping the initial 1B instructions. Table IV summarizes the architectural parameters used in the experiments.

A. Impact of the eDRAM Capacitance and Processor Frequency on Performance

Performance of the macrocell has been evaluated varying the eDRAM capacitance and the processor frequency and compared against the Conv-Cache and WP-Cache schemes.

Fig. 5 shows the results. For the sake of clarity, only a subset of the capacitance values analyzed in Section III are represented (i.e., 0.04, 0.31, 2.5, 10, and 20 fF). As expected, the performance grows with the capacitance. Although for some configurations 2.5 fF is enough capacitance to achieve the maximum performance regardless the processor frequency, a capacitance of about 10 fF ensures the maximum performance for all configurations. Of course, the 20 fF capacitance also obtains the maximum performance.

Notice that for a given cache size, the performance losses due to low capacitances increase with the number of ways since it implies a larger dynamic part and thus a higher number of accesses to eDRAM data or additional accesses to L2 that enlarge the execution time.

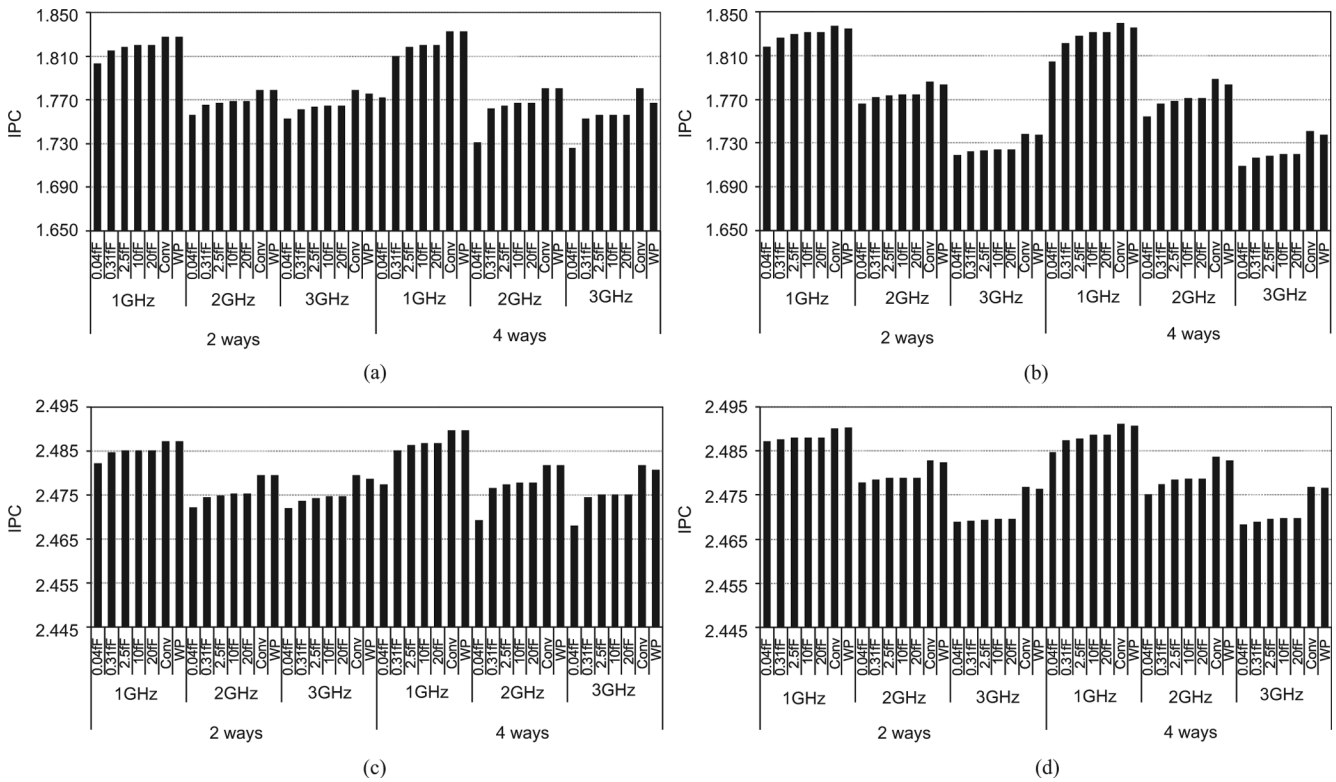

Fig. 5. Performance varying the eDRAM capacitance and processor frequency for the analyzed cache schemes. (a) 16 kB (Int benchmarks) (b) 32 kB (Int benchmarks). (c) 16 kB (FP benchmarks). (d) 32 kB (FP benchmarks).

Independently of the cache size and number of ways, the maximum performance of the M-Cache is slightly lower than the performance of the other schemes mainly due to two reasons: 1) accessing a non-MRU block takes, in general, more time with the way prediction schemes (WP-Cache and M-Cache, see Table II) and 2) the cache cannot be accessed while a block is being swapped (i.e., internal transfer). Nevertheless, using 10 fF, performance losses are always lower than 2% with respect to the Conv-Cache scheme. The worst results are those achieved by the 16 kB-4 way organization with 0.04 fF at 1 GHz when running integer benchmarks, where the performance loss is almost 4%. The former reason also explains why the WP-Cache achieves lower performance than the Conv-Cache.

Finally, regardless the cache organization and type of benchmark, the higher the frequency the lower the performance as higher frequencies require a higher number of cycles to access the cache.

B. Energy Evaluation

Dynamic energy dissipates when transistors change their state, which rises in specific operations or events. In this work, these events have been classified into four categories: loads, stores, misses and writebacks. Fig. 6 shows the M-Cache data array energy results (in mJ) for each benchmark.

As obtained in Section IV-A, the capacitance has been fixed to 10 fF since it is the minimum capacitance that ensures the maximum M-Cache performance for all tested configurations and the processor frequency has been set to 1 GHz. Results for the other frequencies are not shown since they exhibit the same energy trend.

Results show that, for a given cache size, on average, the higher the associativity degree the lower the energy consumption. This happens because the size of the SRAM way of the cache is smaller (e.g., 4 kB in the 16 kB-4 way cache against 8 kB in the 16 kB-2 way cache), thus reducing both leakage and dynamic energy consumption per access (i.e., reducing dynamic energy due to loads, stores, misses, and writebacks).

On the other hand, for a given associativity degree, the larger the cache size the higher the energy consumption. This is due to the fact that leakage and dynamic energy per access increase with the cache size. Leakage energy increases almost at the same rate as the cache size. The effect of leakage is more remarkable since it dominates the overall energy consumption. Nevertheless, the dynamic energy due to misses and writebacks decreases since the amount of these events is rather low.

Regarding the type of benchmark for a given cache organization, integer benchmarks (e.g., *181.mcf* and *300.twolf*) exhibit higher leakage energy than floating-point benchmarks since integer programs take more cycles to execute. In contrast, floating-point benchmarks dissipate more dynamic energy due to misses since these benchmarks (e.g., *178.galgel* and *179.art*) have more L1 misses.

Finally, notice that the percentages of leakage and dynamic energy with respect to the overall energy consumption depend on each benchmark execution. Running benchmarks which do not stress the caches will provide an energy consumption mainly due to leakage energy.

Fig. 7 shows the results for all the cache schemes varying the capacitance and processor frequency. The M-Cache scheme shows the best energy results mainly due to the much lower

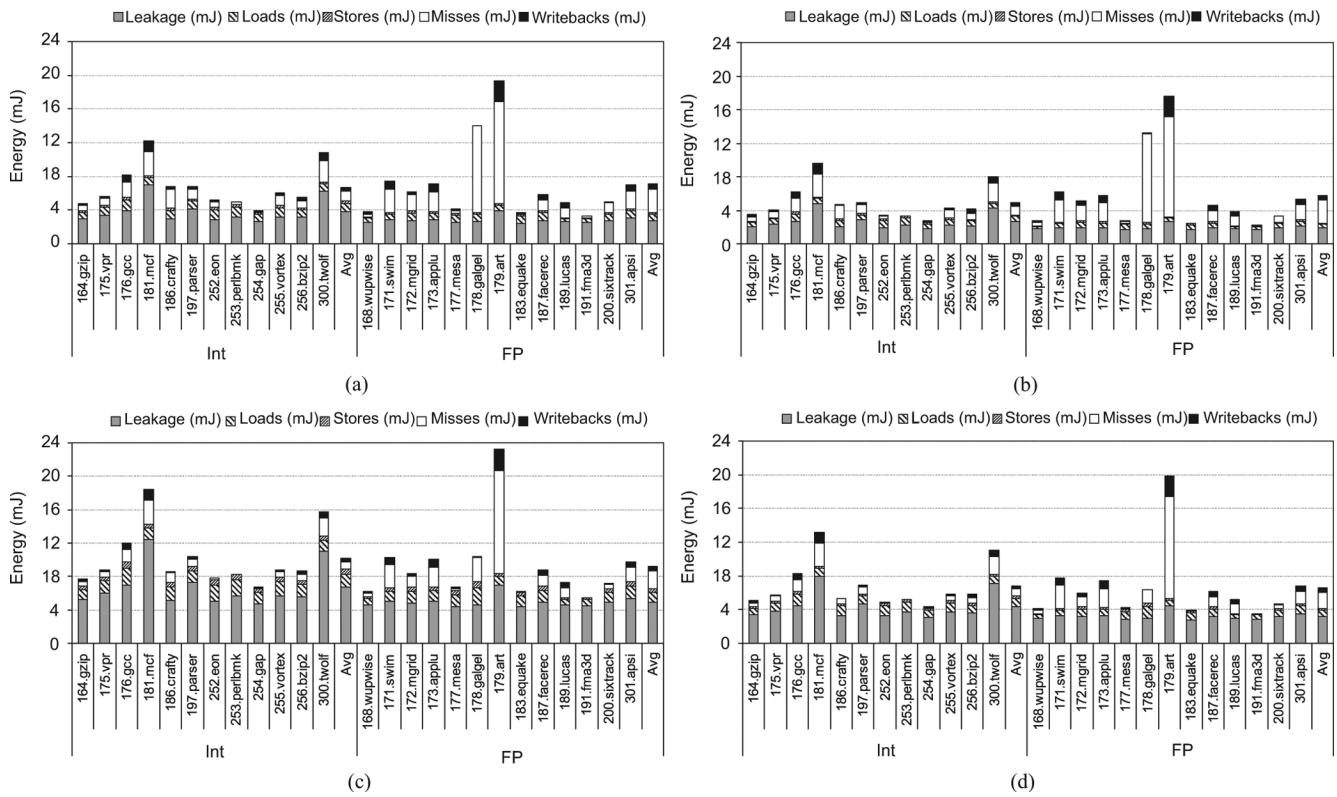

Fig. 6. M-Cache energy consumption (mJ) per benchmark. (a) 16 kB-2 way. (b) 16 kB-4 way. (c) 32 kB-2 way. (d) 32 kB-4 way.

leakage consumption thanks to the use of eDRAM cells. In particular, regardless the type of benchmark and processor frequency, the M-Cache reduces the leakage consumption with respect to the Conv-Cache by 41% and 62% for a 32 kB-2 way and a 32 kB-4 way cache, respectively.

Ideally, assuming that eDRAM cells dissipate negligible leakage, the M-Cache scheme would provide a leakage energy reduction over a conventional SRAM cache by about 50% and 75% for 2- and 4-way caches, respectively. However, experimental results do not reach these values because they consider the whole cache circuitry and not only the data array. Notice that leakage consumption for the Conv-Cache and WP-Cache schemes is almost the same since both are implemented with SRAM cells.

As expected, the leakage energy consumption increases with the cache size and decreases with the frequency. The former happens because the cache has more transistors and the latter because the programs take less time to execute.

Regarding the dynamic energy of the M-Cache, 10 fF is the capacitance with the lowest consumption since it avoids performance losses. Smaller capacitances incur in a larger number of internal transfers, misses and writebacks, which increase the dissipated dynamic energy. On the other hand, larger capacitances, although also avoid performance losses, increase the dynamic energy per access since the charge of the capacitor is more costly in terms of energy consumption.

The WP-Cache presents the lowest dynamic energy consumption, closely followed by the M-Cache with 10 fF capacitors. The reason is that a hit in the predicted way saves a significant amount of dynamic energy in both schemes. The differences between both schemes mainly appear due to the internal transfers in the M-Cache scheme. In particular, for integer benchmarks and regardless the frequency, the WP-Cache reduces the dynamic energy consumption with respect to the Conv-Cache by about 19% and 43% for a 32 kB 2-way and a 32 kB 4-way cache, respectively. The corresponding percentages for the M-Cache are 11% and 34%, respectively.

Concerning the energy consumption related to the tag array, since all schemes use the same tag array, the energy consumption slightly differs among the analyzed schemes. Taken into account both dynamic and leakage energy, the energy differences between the M-Cache and the Conv-Cache tag arrays were always lower than 0.38% (not shown).

Finally, compared to a Conv-Cache with the same capacity and associativity degree, an M-Cache with 10 fF exhibits, depending on the processor frequency and type of benchmark, a total energy reduction up to 33% and 55% for 2- and 4-way set-associative caches, respectively.

C. Tradeoff Between Energy Consumption and Performance

Neither performance nor energy can be evaluated in an isolated way in current systems, but rather a tradeoff must be reached between them. Fig. 8 plots the execution time (in ms) with the corresponding energy for the analyzed cache schemes, hereby obtaining the energy delay^2 product. The capacitance of the M-Cache has been fixed to 10 fF.

As expected, the M-Cache scheme shows the best energy delay^2 product results. For a given associativity degree, it can be seen that the energy delay^2 product grows with the cache size for all cache schemes. In contrast, for a given cache size,

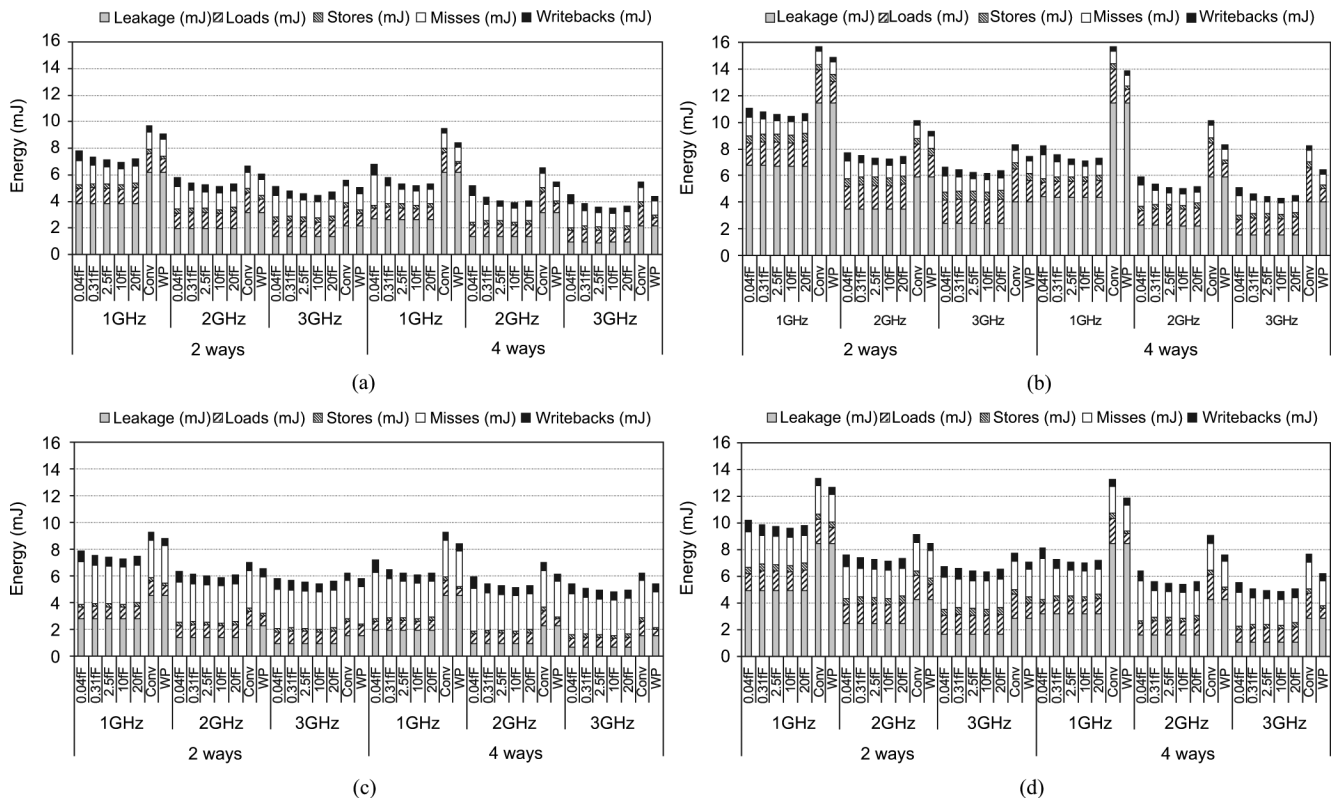

Fig. 7. Energy consumption (mJ) varying the capacitance and the frequency for the analyzed cache schemes. (a) 16 kB (Int benchmarks). (b) 32 kB (Int benchmarks). (c) 16 kB (FP benchmarks). (d) 32 kB (FP benchmarks).

this value decrases with the number of ways for both M-Cache and WP-Cache schemes. This effect is more noticeable for the M-Cache, since it reduces both leakage and dynamic energy. On the other hand, processor frequency highly impacts the energy delay^2 product, since both overall energy consumption and execution time decrease with the frequency.

Despite the lower performance obtained by the M-Cache with 10 fF compared to the Conv-Cache (i.e., the execution time using the M-Cache is higher than the execution time of the Conv-Cache), for a given cache size and number of ways, its energy delay^2 product is always lower than the value of the Conv-Cache because of the energy savings.

In particular, compared to a Conv-Cache with the same capacity and associativity, the M-Cache reduces the energy delay^2 product up to 33% and 54% for 2- and 4-way set-associative caches, respectively. These results are in context with the percentages of the energy savings in Section IV-B.

V. Related Work

The problem of leakage energy consumption in conventional SRAM caches has been addressed in diverse research works that can be classified in two groups depending on whether the proposed technique removes or reduces the power supply. Both approaches assume that during a fixed period of time the cache activity is only focused on a low number of cache lines. Thus, these approaches act on selected lines, requiring from a strategy to select the target lines to reduce energy.

In the first group, leakage is reduced by turning off those cache lines which hold data that is not likely to be used again [17], [18]. Hence, subsequent accesses to those lines result in a

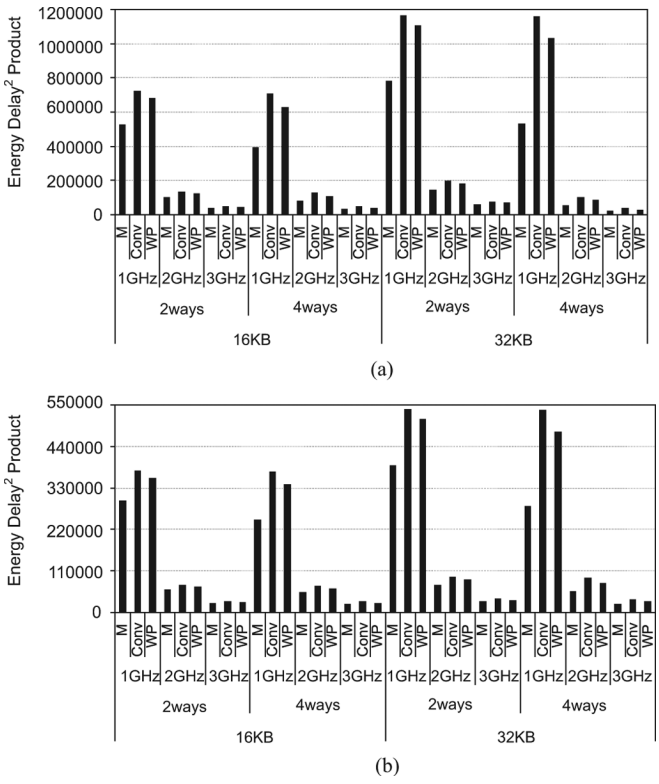

Fig. 8. Energy delay^2 product for different cache organizations and processor frequencies. (a) Int benchmarks. (b) FP benchmarks.

cache miss and an extra access to the next level of the memory hierarchy must be performed, thus hurting the performance.

The techniques falling in the second group put the selected lines into a state-preserving low-power mode [5], [19], reaching the same hit rate as a conventional cache. However, they reduce less leakage since the lines are not completely turned off. In addition, reducing the power supply to a given line increases its access time.

As DRAM cells have, by design, very low leakage currents, some research works focused on the design of new DRAM-like cells for caches. Liang *et al.* [20] proposed the 3T1D (three transistors and a diode) DRAM cell. The speed of these cells is comparable to the speed of 6T SRAM cells. Thus, the 3T1D cells can be used for critical latency structures such as L1 data caches. However, although reads are non-destructive, the diode charge get lost over time, requiring from refresh schemes that might have a severe impact on performance. The 3T1D cell can be smaller than the 6T SRAM cell but the smaller the cell the lower the retention time of the diode capacitance.

Juang *et al.* [21] proposed a dynamic cell from a 6T SRAM cell which does not include the two transistors connected to Vdd that restore the charge loss due to the leakage currents. Thus, the circuit results in a non-static cell with only 4 transistors (the quasi-static 4T cell). This cell offers an easy method for DRAM implementation in a logic process production, especially in embedded systems. Compared to 6T SRAM cells, the 4T cells require less area while achieving almost the same performance. In contrast, the data access is a bit slower and destructive. Likewise in the 1T-1C cell, this problem can be solved by rewriting the read data immediately after the read operation or before the retention time expires.

Other research works focused on mixing technologies to take advantage of the best characteristics that each technology offers. In this context, Wu *et al.* [22] proposed a multilevel cache hierarchy that can be built with different technologies such as SRAM, eDRAM, MRAM, and PRAM. These technologies can be applied at different levels of the memory hierarchy (each technology at a single level) or at the same level (two technologies on a single level). Concerning the latter approach, the L2 and L3 caches are flatten into two regions forming a single level: a small and fast region (SRAM) and a large and slow region (eDRAM or MRAM or PRAM), while the L1 level is implemented with SRAM technology. The most accessed lines reside in the fast region and swap operations are taken into account in order to transfer data among regions. Compared to a conventional SRAM design, the SRAM/eDRAM L2 cache improves performance and reduces power under the same area constraint.

VI. Conclusion

Memory cell design for cache memories is a major concern in current microprocessors mainly due to its impact on energy consumption, area and access time. In this context, the macrocell-based cache (M-Cache), which combines SRAM and eDRAM technologies, has been shown as an efficient device to implement cache memories, since its design deals with the mentioned issues.

The fact of using capacitors in the eDRAM cells of the macrocell has a significant impact on performance and dynamic energy consumption. The capacitors maintain the stored data for a period of time (namely retention time), whose corresponding number of processor cycles varies depending on the capacitance and processor frequency. Capacitances must be precisely established in order to avoid either performance drops or energy wasting.

In this work, the optimal eDRAM capacitance has been identified depending on the frequency at which the processor works. To this end, a detailed analysis of performance and energy has been performed. Experimental results have shown that 10 fF is the optimal capacitance since it is enough to avoid performance losses and exhibits the lowest energy consumption for processor frequencies higher than 1 GHz in L1 M-Caches.

Compared to a conventional cache with the same storage capacity and associativity degree, an M-Cache with 10 fF capacitance obtains an energy reduction about 33% and 55% for 2- and 4-way set-associative caches, respectively; while having scarce impact on performance (lower than 2%). Regarding area, a 4-bit macrocell using trench capacitors provide an area reduction by 29% with respect to 4 conventional SRAM cells, regardless the capacitance. Finally, the energy delay^2 product provided by an M-Cache is always lower than the value of a conventional cache. Therefore, the M-Cache design stands as a cost-effective cache design for technologies smaller than 45-nm.

References

[1] R. E. Matick and S. E. Schuster, "Logic-based eDRAM: Origins and rationale for use," *IBM J. Res. Develop.*, vol. 49, no. 1, pp. 145–165, 2005.

[2] B. Sinharoy, R. N. Kalla, J. M. Tendler, R. J. Eickemeyer, and J. B. Joyner, "POWER5 system microarchitecture," *IBM J. Res. Develop.*, vol. 49, no. 4/5, pp. 505–521, 2005.

[3] J. M. Tendler, J. S. Dodson, J. S. Fields, H. Le, and B. Sinharoy, "POWER4 system microarchitecture," *IBM J. Res. Develop.*, vol. 46, no. 1, pp. 5–25, 2002.

[4] A. Valero, J. Sahuquillo, S. Petit, V. Lorente, R. Canal, P. López, and J. Duato, "An hybrid eDRAM/SRAM macrocell to implement first-level data caches," in *Proc. 42th Annu. IEEE/ACM Int. Symp. Microarch.*, 2009, pp. 213–221.

[5] S. Petit, J. Sahuquillo, J. M. Such, and D. Kaeli, "Exploiting temporal locality in drowsy cache policies," in *Proc. 2nd Conf. Comput. Frontiers*, 2005, pp. 371–377.

[6] W. Zhao and Y. Cao, "Predictive technology model for Nano-CMOS design exploration," *J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 1, pp. 1–17, 2007.

[7] Semiconductor Industries Association, "International Technology Roadmap for Semiconductors," 2007. [Online]. Available: http://www.itrs.net/

[8] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, MA: Addison-Wesley, 2010.

[9] B. Calder, D. Grunwald, and J. Emer, "Predictive sequential associative cache," in *Proc. 2nd Int. Symp. High-Perform. Comput. Arch.*, 1996, pp. 244–253.

[10] K. Inoue, T. Ishihara, and K. Murakami, "Way-predicting set-associative cache for high performance and low energy consumption," in *Proc. 2nd Int. Symp. High-Perform. Comput. Arch.*, 1999, pp. 273–275.

[11] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "CACTI 5.1," Hewlett-Packard Development Company, Palo Alto, CA, 2008.

[12] 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," in *Proc. 35th Annu. Int. Symp. Comput. Arch.*, 2008, pp. 51–62.

[13] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects," Dept. Comput. Sci., Univ. Virginia, Charlottesville, 2003.

[14] Standard Performance Evaluation Corporation, 2000. [Online]. Available: http://www.spec.org/cpu2000
[15] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM Circuit Design. Fundamental and High-Speed Topics*. Hoboken, NJ: Wiley, 2008.
[16] T. Kirihata, P. Parries, D. R. Hanson, H. Kim, J. Golz, G. Fredeman, R. Rajeevakumar, J. Griesemer, N. Robson, A. Cestero, B. A. Khan, G. Wang, M. Wordeman, and S. S. Iyer, "An 800-MHz embedded DRAM with a concurrent refresh mode," *IEEE J. Solid-State Circuits*, vol. 40, no. 6, pp. 1377–1387, Jun. 2005.
[17] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: Exploiting generational behavior to reduce cache leakage power," in *Proc. 28th Annu. Int. Symp. Comput. Arch.*, 2001, pp. 240–251.
[18] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories," in *Proc. Int. Symp. Low Power Electron. Design*, 2000, pp. 90–95.
[19] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *Proc. 29th Annu. Int. Symp. Comput. Arch.*, 2002, pp. 148–157.
[20] X. Liang, R. Canal, G.-Y. Wei, and D. Brooks, "Process variation tolerant 3T1D-Based cache architectures," in *Proc. 40th Annu. IEEE/ACM Int. Symp. Microarch.*, 2007, pp. 15–26.
[21] Z. Hu, P. Juang, P. Diodato, S. Kaxiras, K. Skadron, M. Martonosi, and D. W. Clark, "Process variation tolerant 3T1D-Based cache architectures," in *Proc. Int. Symp. Low Power Electron. Design*, 2002, pp. 52–55.
[22] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *Proc. 36th Annu. Int. Symp. Comput. Arch.*, 2009, pp. 34–45.

Alejandro Valero was born in Valencia, Spain, on November 14, 1985. He received the B.S. degree in computer engineering from the Universitat Politècnica de València, València, Spain, in 2009. He has earned a 4-year Spanish Research Grant and is currently pursuing the Ph.D. degree from the Department of Computer Engineering at the same university. His Ph.D. research focuses on the design of hybrid caches.

His research topics include energy consumption and memory hierarchy design.

Julio Sahuquillo (M'04) received the B.S., M.S., and Ph.D. degrees in computer science from the Universitat Politècnica de València, València, Spain.

Since 2002, he has been an Associate Professor with the Department of Computer Engineering, Universitat Politècnica de València. He has taught several courses on computer organization and architecture. He has published over 90 refereed conference and journal papers. His research topics have included multiprocessor systems, cache design, instruction-level parallelism and power dissipation.

Prof. Sahuquillo is a member of the IEEE Computer Society.

Vicente Lorente was born in Valencia, Spain, in 1971. He received the M.S. degree in computer engineering from the Universitat Politècnica de València, València, Spain, in 1998, where he is currently pursuing the Ph.D. degree in computer engineering.

Since 2001, he has been an Assistant Professor with the Department of Computer Engineering, Universitat Politècnica de València. His research topics include memory hierarchy design, process variation effects on memory hierarchy, real-time operating systems and worst case execution time analysis.

Salvador Petit (M'07) received the Ph.D. degree in computer engineering from the Universitat Politècnica de València (UPV), València, Spain.

Currently, he is an Associate Professor with the Department of Computer Engineering, UPV, where he has taught several courses on computer organization. His research topics include multithreaded and multicore processors, memory hierarchy design, as well as real-time systems.

Prof. Petit is a member of the IEEE Computer Society.

Pedro López (M'02) received the B.Eng. degree in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from the Universitat Politècnica de València, València, Spain, in 1984, 1990, and 1995, respectively.

He is a Full Professor in computer architecture and technology with the Department of Computer Engineering, Universitat Politècnica de València. He has taught several courses on computer organization and architecture. His research interests include high performance interconnection networks for multiprocessor systems and clusters and networks on chip. He has published over 100 refereed conference and journal papers.

Prof. López is a member of the editorial board of Parallel Computing journal and a member of the IEEE Computer Society.

José Duato received the M.S. and Ph.D. degrees in electrical engineering from the Universitat Politècnica de València, València, Spain, in 1981 and 1985, respectively.

He was an Adjunct Professor with the Department of Computer and Information Science, The Ohio State University, Columbus. He is currently a Professor with the Department of Computer Engineering (DISCA), Universitat Politècnica de València. His research interests include interconnection networks and multiprocessor architectures. He has published over 380 refereed papers. He proposed a powerful theory of deadlock-free adaptive routing for wormhole networks. Versions of this theory have been used in the design of the routing algorithms for the MIT Reliable Router, the Cray T3E supercomputer, the internal router of the Alpha 21364 microprocessor and the IBM BlueGene/L supercomputer. He is the first author of *Interconnection Networks: An Engineering Approach* (Morgan Kaufmann, 2002).

Dr. Duato was a member of the editorial boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, and IEEE COMPUTER ARCHITECTURE LETTERS. He was cochair, member of the steering committee, vice-chair, or member of the program committee in more than 55 conferences, including the most prestigious conferences in his area: HPCA, ISCA, IPPS/SPDP, IPDPS, ICPP, ICDCS, EuroPar, and HiPC.