ISR TECHNICAL REPORT 2012-11

Refresh Matters: Energy and Performance Analysis of Large Last-Level Cache Built with Gain Cell Embedded DRAM

Mu-Tien Chang, Paul Rosenfeld, Shih-Lien Lu, and Bruce Jacob

Refresh Matters: Energy and Performance Analysis of Large Last-Level Cache Built with Gain Cell Embedded DRAM

Mu-Tien Chang*, Paul Rosenfeld*, Shih-Lien Lu†, and Bruce Jacob*
*University of Maryland †Intel Corporation

Abstract

Embedded dynamic random access memories (eDRAMs) have high density and low leakage futures, making them suitable for implementing large last-level caches (L^3Cs). However, refresh operations are required, which negatively impact the power and performance. This article investigates the impact of refresh on energy and performance of eDRAM-based L^3Cs. Experiments show that refresh has minor impact on system performance but continues to be the primary source of eDRAM-based L^3C energy consumption.

1. Introduction

Last-level cache (LLC) is efficient for bridging the performance and power gap between processor and memory. Future processors are expected to have more cores, emerging multi-core workloads are also shown to be memory intensive and have large working set size. As a result, the demand for large last-level caches has increased in order to improve the system performance and power/energy.

L^3Cs are often optimized for high density and low power. While SRAMs (static random access memories) have been the mainstream embedded memory technology due to their fast access time and logic compatibility, they are low density and have high leakage current. On the other hand, eDRAMs feature small cell size and low cell leakage, making them potential replacements for SRAMs in the context of L^3Cs. For instance, eDRAM has been used to implement the last-level L3 cache of the IBM Power7 processor.

Though they provide many benefits, refresh operations are required to preserve data integrity. Refresh introduces two problems: degraded cache bandwidth and increased power dissipation. First, normal cache accesses are stalled while the cache is refreshing. This problem can be mitigated by organizing a cache into multiple subarrays, allowing refresh operations and normal cache accesses to happen concurrently [7]. Second, refresh results in significant power overhead. For instance, our study shows that for an 8-core processor with a 32 MB last-level eDRAM cache, refresh power contributes to up to 50% of the total LLC power.

This article investigates to what extent refresh affects the energy and performance of eDRAM-based L^3C designs. Specifically, we use *gain cell*, a form of eDRAM that is standard CMOS compatible, as our case study. We demonstrate the impact of cache size, processor frequency, refresh policy, process variation, temperature, and technology scaling. At the conclusion of this study, we show that reducing refresh power is key to energy-efficient eDRAM L^3Cs in advanced process technologies.

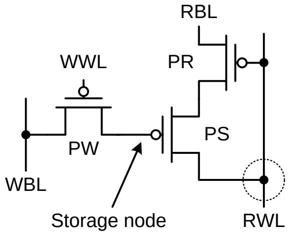


Figure 1. Schematic of the boosted 3T PMOS gain cell eDRAM [5].

2. Background

2.1. Gain Cell eDRAM

A gain cell can be built in standard CMOS technology, usually implemented using two or three transistors [10] [6] [13] [5], providing fast read path, non-destructive read operation, and better noise margins at low voltages due to its decoupled read and write paths. When compared to an SRAM cache, a gain cell eDRAM cache is approximately 2X denser and consumes 2.5X less leakage.

This study utilizes the boosted 3T gain cell [5] as the eDRAM cell structure due to its capability to operate at high frequency while maintaining an adequate data retention time. Figure 1 shows the schematic of the boosted 3T PMOS eDRAM gain cell. It is comprised of a write access transistor (PW), a read access transistor (PR), and a storage transistor (PS). PMOS transistors are utilized because a PMOS device has less leakage current compared to an NMOS device of the same size. Lower leakage current enables lower standby power and longer retention time.

During write access, the write bit-line (WBL) is driven to the desired voltage level by the write driver. Additionally, the write word-line (WWL) is driven to a negative voltage to avoid the threshold voltage drop such that a complete data '0' can be passed through the PMOS write access transistor from WBL to the storage node.

When performing a read operation, once the read wordline (RWL) is switched from VDD to 0 V, the precharged read bit-line (RBL) is pulled down slightly if a data '0' is stored in the storage node. If a data '1' is stored in the storage node, RBL remains at the precharged voltage level. The gate-to-RWL coupling capacitance of PS enables preferential boosting: when the storage node voltage is low, PS is in inversion mode, which results in a larger coupling capacitance. On the other hand, when the storage node voltage is high, PS is in weak-inversion mode, which results in a smaller coupling capacitance. Therefore, when RWL switches from VDD to 0 V, a low storage node voltage is coupled down more than a high storage node voltage. The signal difference between data '0' and data '1' during a read operation is thus amplified through preferential boosting. This allows the storage node voltage to decay further before a refresh is needed, which effectively translates to a longer data retention time and better read performance.

2.2. Embedded DRAM Refresh Policies

Embedded DRAMs utilize some form of capacitor to store data. Since the stored charge gradually leaks away, refresh is necessary to prevent data loss. Reohr [11] presents several approaches to refresh eDRAM-based caches, including *periodic refresh*, *line-level refresh* based on time stamps, and *no-refresh*. For instance, Liang et al. [8] showed that by adopting the line-level refresh or the no-refresh approaches with intelligent cache replacement policies, 3T1D (three transistors one diode) eDRAM is a potential substitute for SRAM in the context of the L1 data cache.

The periodic refresh policy does a sweep of the cache such that all the cache lines are refreshed periodically. It uses the least logic and storage overhead but provides no opportunity to reduce the number of refresh operations.

The line-level refresh policy utilizes line-level counters to track the refresh status of each cache line. When a line is refreshed, its counter resets to zero. There are two types of refreshes: the *implicit refresh* and the *explicit refresh*. An implicit refresh happens when the line is read, written, or loaded; an explicit refresh happens when the line-level counter signals a refresh to the data array. Therefore, if two accesses to the same cache line occur within a refresh period, the cache line is implicit refreshed and no explicit refresh is needed. One drawback of this scheme is that it introduces more logic and storage overhead. For instance, a seven-bit counter is required for each line to provide 1% time stamp precision. Line-level refresh usually performs worse if the cache is not intensively accessed.

The no-refresh policy never refreshes the cache lines. Similar to the line-level refresh implementation, each cache line has a counter that tracks the time after an implicit refresh. When the counter reaches the retention time, the line is marked as invalid. As a result, the no-refresh policy eliminates refresh power completely but potentially introduces more cache misses.

3. Gain Cell eDRAM-Based Cache Modeling

3.1. Modeling

Our gain cell eDRAM-based cache model is built on top of the CACTI cache tool [15]. CACTI is a widely used analytical model that estimates the power, performance, and area of caches. We integrate the boosted 3T gain cell and its peripheral circuits into the tool. The peripheral circuits such as the sense amplifier, the precharge circuit, and the wordline driver are modified in order to reflect realistic gain cell eDRAM circuit behaviors. Additionally, although both circuit performance and power are temperature dependent, CACTI only models the dependence of leakage power dissipation on temperature. We enhance CACTI such that power (dynamic power, leakage power, refresh power) and performance (access time, cycle time, retention time) are all temperature dependent.

We use a look-up table approach to integrate the gain cell eDRAM model into CACTI. We first conduct circuit (HSPICE) simulations using the PTM LP CMOS models [2] to obtain functional gain cell memory arrays. The characteristics that are needed for the look-up table are then extracted from the HSPICE simulations. These include the capacitances of each of the cell's terminals, the driving and leakage currents of each of the cell's transistors, the data retention time, the power and performance of the modified sense amplifier, precharge circuit, drivers, and other circuit and device level details.

3.2. Validation

The gain cell eDRAM model is validated against [5] with respect to latency, retention time, and refresh power. Our model is based on CACTI utilizing the 65 nm PTM LP CMOS technology, while the hardware test chip presented in [5] is fabricated in a 65 nm LP CMOS process. Setting the same memory array size, operating voltage and temperature, our model shows 11% increase in latency and 20% decrease in retention time. In addition, with the same refresh rate, our model shows 13% more refresh power. These differences are possibly due to implementation differences between the processes and array organizations.

3.3. Refresh Controller

We integrate a refresh controller into a full-system simulator, which can be configured to perform a range of refresh policies, including periodic refresh, line-level refresh, and no-refresh. We also augment the simulator with parameterized refresh rate. The configurable refresh controller allows users to explore the effect of eDRAM caches using different refresh algorithms under a full-system simulation environment.

4. Experimental Methodology

This study utilizes MARSS [9], a full-system simulator for x86-64 CPUs. The default configuration is an 8-core,

Table 1. Baseline system configuration.

Processor	8-core, 2 GHz, out-of-order, 4-wide issue width
L1I (private)	32 KB, 8-way set associative, 64 B line size, 1 bank, MESI cache
L1D (private)	32 KB, 8-way set associative, 64 B line size, 1 bank, MESI cache
L2 (private)	256 KB, 8-way set associative, 64 B line size, 1 bank, MESI cache
L3 (shared)	32 MB, 16-way set associative, 64 B line size, 16 banks, write-back cache
Main memory	8 GB, 1 channel, 4 ranks/channel, 8 banks/rank

out-of-order 2 GHz system that operates at 75°C, with L1 and L2 private caches, and a 32 MB shared last-level L3 cache. A pseudo-LRU replacement policy [3] is used for the caches. The L1 caches are implemented using multi-port (2-read/2-write) high performance SRAMs and the L2 caches are built with single-port high performance SRAMs. In order to reduce leakage power while maintaining performance, we use gain cell eDRAMs to build the L3 cache data array, while using high performance transistors and low power SRAM cells to implement the peripheral circuitry and the L3 cache tag array, respectively. We configure the L3 cache such that it is sequentially accessed (i.e. the tag and data are accessed sequentially). Sequentially accessed cache saves the dynamic power of accessing the data array when the cache misses. We also use the periodic refresh policy for the eDRAM cache by default. The power and performance characteristics of the caches are based on our modified CACTI model. Moreover, DRAMSim2 [12], a cycle-accurate DRAM simulator is utilized for the main memory model, which is integrated with MARSS. The 8 GB main memory is configured as 1 channel, 4 ranks per channel, and 8 banks per rank, using Micron's DDR3 2 Gb device parameters [14]. Table 1 summarizes our system configuration.

Our system evaluation is based on multi-thread workloads from the PARSEC 2.1 benchmark suite [4] and the NAS parallel benchmark suite (NPB 3.3.1) [1]. They are configured as single-process, 8-threaded workloads. We use the input sets *simmedium* and *CLASS A* for the PARSEC and NAS benchmarks, respectively. When executing each workload, we skip the initialization phase and run 2.4 billion instructions in detailed simulation mode. All workloads run on top of Ubuntu 9.04 (Linux 2.6.31).

5. Results and Analysis

5.1. Cache Size

Increasing the LLC size potentially results in shorter system execution time, as shown in Figure 2(a). Unlike caches that are closer to the cores, LLCs are usually target to improve the on-chip cache hit ratio. Better on-chip cache hit ratio reduces the number of long accesses to the off-chip memory. Therefore, although a larger LLC has longer cache access latency and requires more refresh operations, it improves the system performance.

In addition, a larger LLC in some cases reduces the system energy consumption due to shorter execution time and reduced main memory active power, as shown in Figure 2(b). However, LLC energy increases with increasing cache size (Figure 2(c)). In particular, since the retention time is independent of the cache organization, more refresh operations are required for larger caches within the same refresh period. As a result, as the number of cache lines increases, refresh power becomes the primary source of LLC power dissipation.

5.2. Processor Frequency

Figure 3 shows the impact of processor frequency. As expected, higher frequency achieves better execution time (Figure 3(a)). On the other hand, when operating at a higher frequency, leakage and refresh become relatively less significant, but they still dominate the total LLC energy consumption (Figure 3(b)).

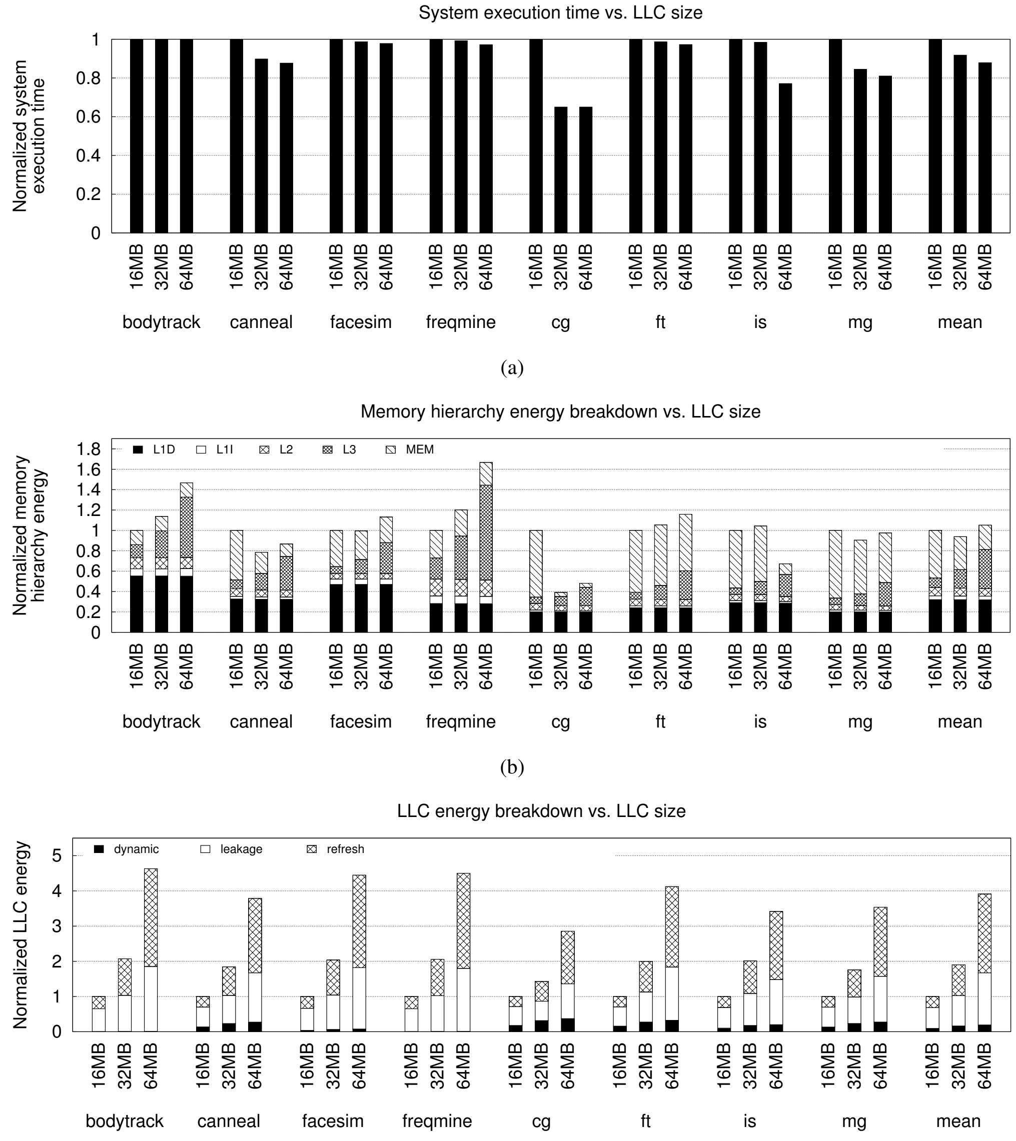
5.3. Refresh Policy

Figure 7 compares the system performance and LLC energy when using various refresh policies. In contrast to utilizing the line-level refresh policy for the L1 cache, applying it to the LLC results in slightly more energy usage. This is because the LLC is not as intensively accessed as the L1 caches, making the line-level refresh unlikely to take advantage of implicit refreshes. Furthermore, the line-level refresh policy shortens the refresh period because in the worst case scenario, all cache lines in a subarray reach the refresh threshold simultaneously. This means that in order to avoid data loss, cache lines must begin refreshing sooner than the refresh threshold so that no line in the subarray exceeds the retention time.

Similar to the line-level refresh policy, there is little opportunity for no-refresh to carry out implicit refreshes. Consequently, most of the cache lines become invalid before they are reused. The large LLC miss penalty makes the no-refresh overhead even more significant. Therefore, although no-refresh consumes less LLC energy, it degrades the system performance substantially. It also increases the main memory energy consumption by 2X on average. We thereby show that no-refresh is the least efficient policy, while the periodic refresh best suits eDRAM-based LLCs.

5.4. Process Variation

Process variation (PV) affects the retention time of a DRAM cell, whereas the refresh rate is determined by the weakest cells (i.e. cells that have the shortest data retention



(a)

(b)

(c)

Figure 2. The impact of LLC size. (a) System execution time. (b) Memory hierarchy energy breakdown. (c) LLC energy breakdown.

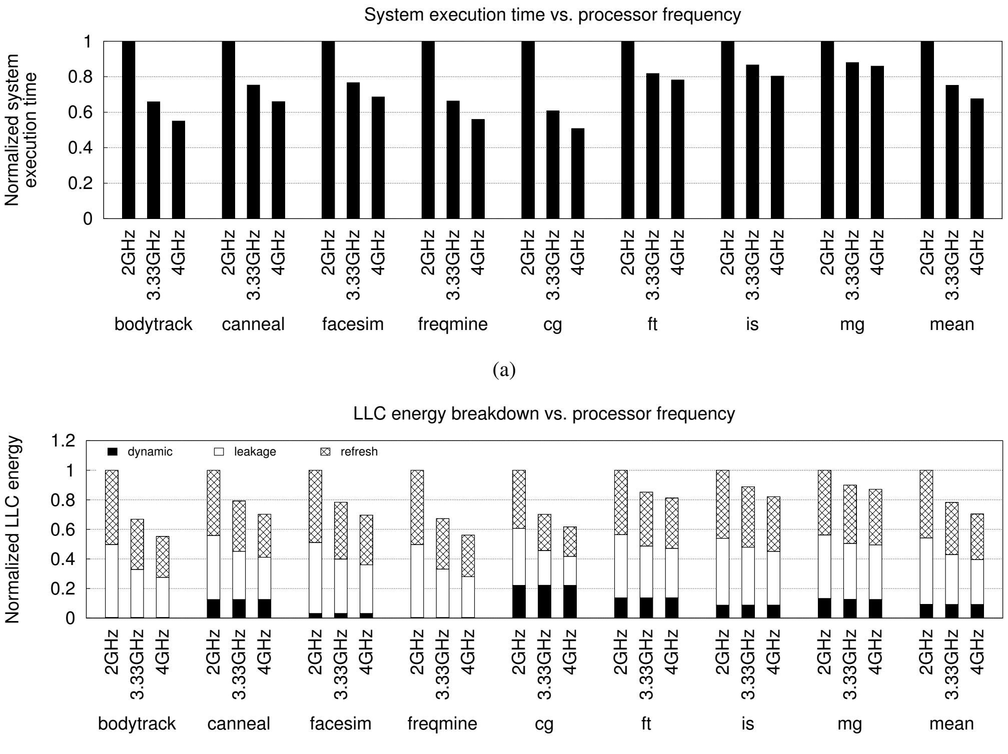
time). Therefore, as PV becomes more severe, the refresh rate increases, which translates into higher refresh power, as illustrated in Figure 5. Note that we did not show the impact on system execution time because for gain cell eDRAMs, PV has minor effect on the access time, thus the performance difference due to different degrees of PV is insignificant.

5.5. Temperature

High temperature results in increased cache access time and decreased eDRAM retention time. However, our study shows that the degraded LLC performance due to increased temperature has negligible impact on system performance (less than 2% execution time overhead). In contrast, high temperature results in more than 20% LLC energy overhead, as shown in Figure 6. In particular, temperature variation greatly affects the leakage and refresh power. On average, when increasing the temperature from 75^{o}C to 95^{o}C, the leakage and refresh power increase by 36% and 11%, respectively.

5.6. Technology Scaling

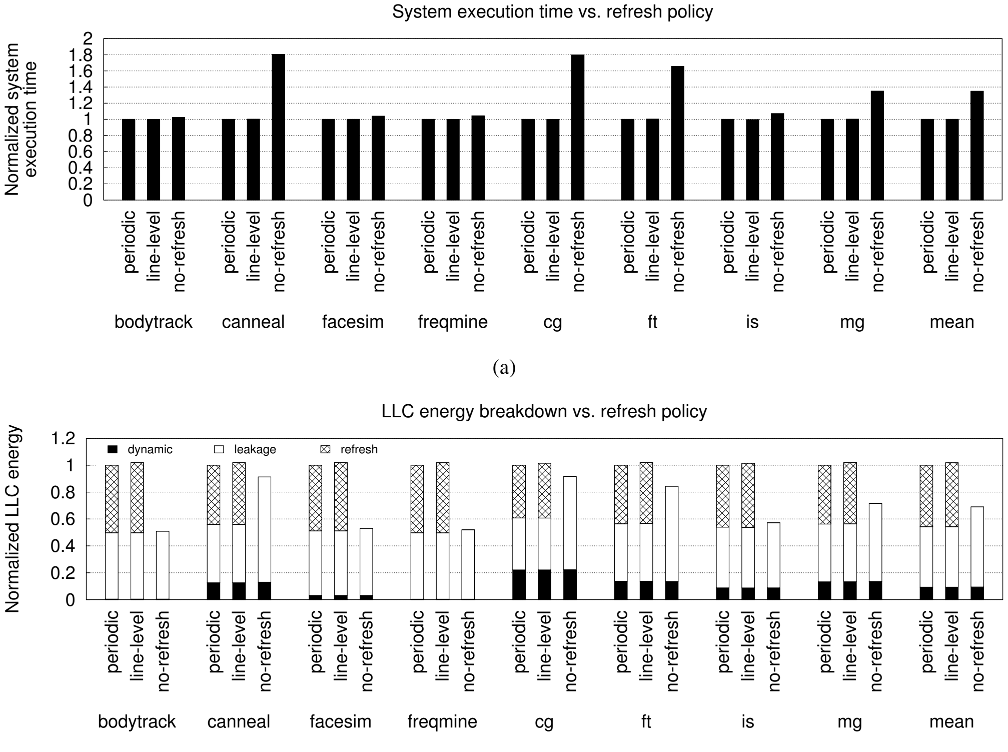
As technology scales down, caches become smaller, faster, and consume less active energy. For instance, a 32 nm cache is 49% smaller than a 45 nm cache, while a 22 nm cache is 55% smaller than a 32 nm cache. Although caches implemented using smaller technology nodes have shorter access latency, the effect is not fully reflected on the system performance, as shown in Figure 7(a). We expect the impact of technology scaling on system performance to be more visible when operating at a higher processor frequency (e.g. 4 GHz).



(a)

(b)

Figure 3. The impact of processor frequency. (a) System execution time. (b) LLC energy breakdown.



(a)

(b)

Figure 4. The impact of refresh policy. (a) System execution time. (b) LLC energy breakdown.

On the contrary, our study shows that technology scaling has great impact on energy usage, as illustrated in Figure 7(b). Both subthreshold and gate leakages increase significantly with decreasing feature size. The increasing leakage coupled

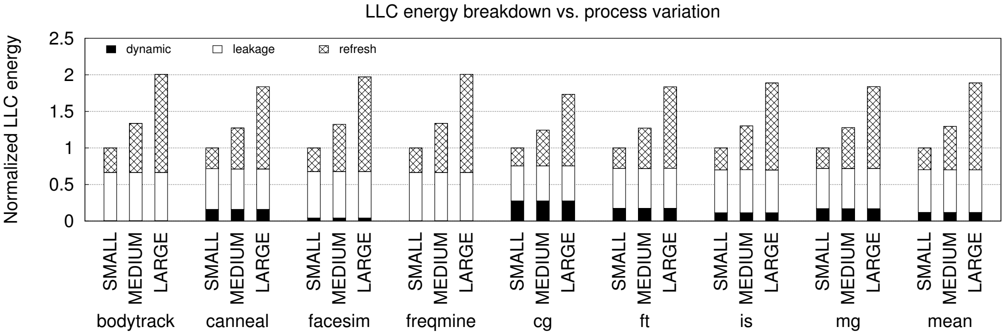


Figure 5. The impact of process variations on LLC energy. SMALL: small process variations; MEDIUM: typical process variations; LARGE: severe process variations.

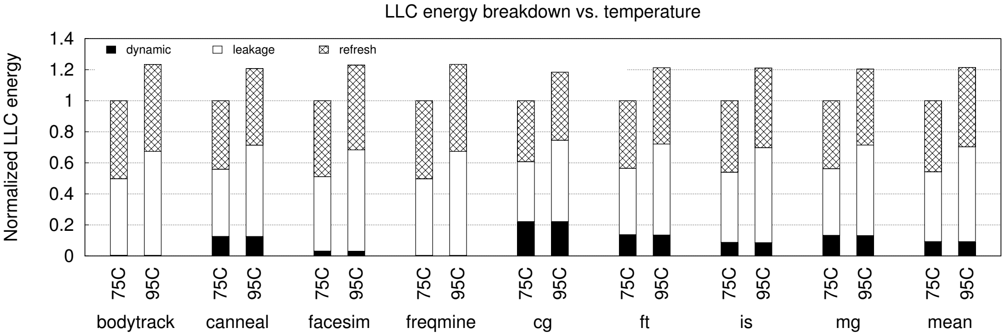


Figure 6. The impact of temperature on LLC energy.

with smaller cell storage capacitance results in shorter retention time. As a result, the LLC leakage and refresh power become worse in advanced technologies.

6. Conclusion

Embedded DRAM, featuring high density and low leakage, is a viable alternative to SRAM in the context of L^3C. As future processors are expected to have larger LLCs implemented using more advanced process technologies, refresh power becomes the major source of power dissipation. Reducing refresh power is thus key to energy-efficient eDRAM-based L^3Cs.

References

[1] NAS Parallel Benchmarks. http://www.nas.nasa.gov/Resources/Software/npb.html.

[2] Predictive Technology Model. http://ptm.asu.edu/.

[3] H. Al-Zoubi, A. Milenkovic, and M. Milenkovic. Performance Evaluation of Cache Replacement Policies for the SPEC CPU2000 Benchmark Suite. In *Proc. of the 42nd Ann. Southeast Regional Conference*, pages 267–272. ACM Press, 2004.

[4] C. Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, Jan. 2011.

[5] K. C. Chun, P. Jain, J. H. Lee, and C. H. Kim. A 3T Gain Cell Embedded DRAM Utilizing Preferential Boosting for High Density and Low Power On-Die Caches. *IEEE J. Solid-State Circuits*, 46(6):1495–1505, Jun. 2011.

[6] N. Ikeda, T. Terano, H. Moriya, T. Emori, and T. Kobayashi. A Novel Logic Compatible Gain Cell with Two Transistors and One Capacitor. In *Proc. Symp. VLSI Technology*, pages 168–169. IEEE Press, 2000.

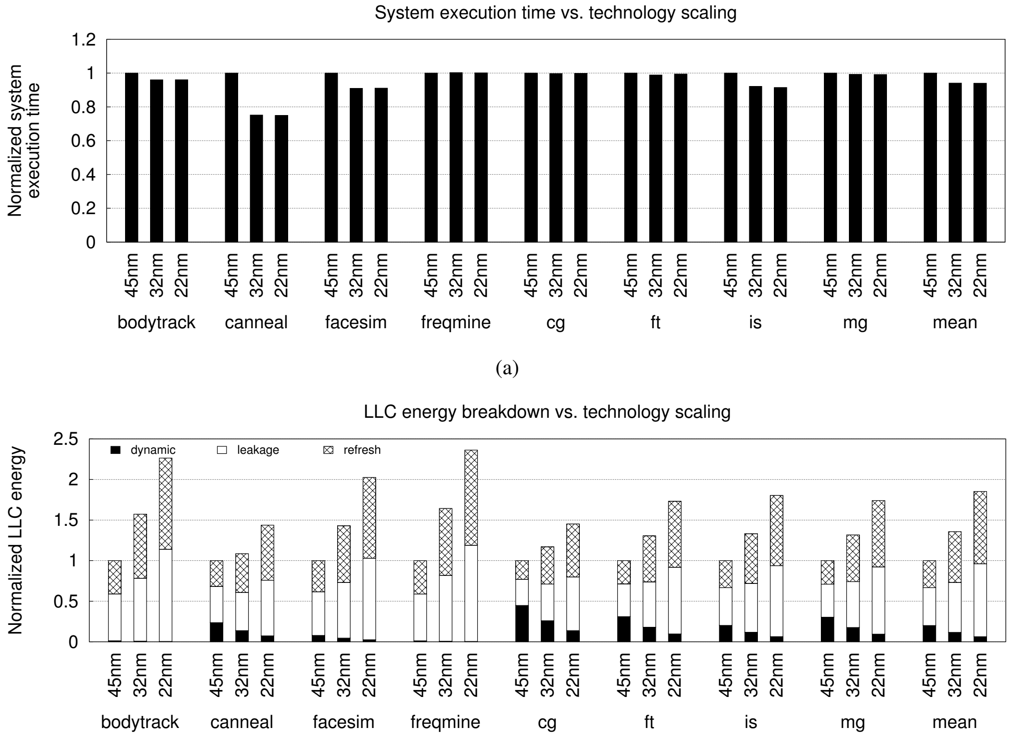
[7] T. Kirihata, P. Parries, D. R. Hanson, H. Kim, J. Golz, G. Fredeman, R. Rajeevakumar, J. Griesemer, N. Robson, A. Cestero, B. A. Khan, G Wang, M. Wordeman, and S. S. Iyer. An 800-MHz Embedded DRAM with a Concurrent Refresh Mode. *IEEE J. Solid-State Circuits*, 40(6):1377–1387, Jun. 2005.

[8] X. Liang, R. Canal, G. Y. Wei, and D. Brooks. Process Variation Tolerant 3T1D-Based Cache Architectures. In *Proc. 40th Ann. IEEE/ACM Int'l Symp. on Microarchitecture (MICRO 07)*, pages 15–26. IEEE CS Press, 2007.

[9] A. Patel, F. Afram, S. Chen, and K. K. Ghose. MARSS: A Full System Simulator for x86 CPUs. In *Proc. 48th Design Automation Conference (DAC 11)*, pages 1050–1055. ACM Press, 2011.

[10] W. Regitz and J. Karp. A Three Transistor-Cell, 1024-bit, 500 ns MOS RAM. In *Proc. Int'l Solid-State Circuits Conf. (ISSCC 1970)*, pages 42–43. IEEE Press, 1970.

[11] W. R. Reohr. Memories: Exploiting Them and Developing Them. In *Proc. Int'l Systems-on-Chip Conf. (SOCC 06)*, pages 303–310. IEEE Press, 2006.



(a)

(b)

Figure 7. The impact of technology scaling. (a) System execution time. (b) LLC energy breakdown.

[12] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. DRAMSim2: A Cycle Accurate Memory System Simulator. *IEEE Computer Architecture Letters*, 10(1):16–19, Jan. 2011.

[13] D. Somasekhar, Y. Ye, P. Aseron, S. L. Lu, M. M. Khellah, J. Howard, G. Ruhl, T. Karnik, S. Borkar, V. K. De, and A. Keshavarzi. 2 GHz 2 Mb 2T Gain Cell Memory Macro With 128 GBytes/sec Bandwidth in a 65 nm Logic Process Technology. *IEEE J. Solid-State Circuits*, 44(1):174–185, Jan. 2009.

[14] Micron Technology. DDR3 SDRAM. http://micron.com/document_download/?documentId=424, 2010.

[15] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. 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 Ann. Int'l Symp. Computer Architectures (ISCA 08)*, pages 51–62. IEEE CS Press, 2008.