Universitat Politècnica de València

Ph.D. Thesis

Cache Architectures Based on Heterogeneous Technologies to deal with Manufacturing Errors

Author:
Vicente Jesús Lorente Garcés

Advisors:
Prof. Salvador V. Petit Martí
Prof. Julio Sahuquillo Borrás

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
(Computer Engineering)

in the
Parallel Architectures Group
Department of Computer Engineering

November 2015

Doctoral Committee

- Prof. María Engracia Gómez Requena

  Universitat Politècnica de València, Valencia, Spain.

- Prof. Manuel Eugenio Acacio Sánchez

  Universidad de Murcia, Murcia, Spain.

- Prof. Manuel Pérez Malumbres

  Universidad Miguel Hernández, Elche, Spain.

Agraïments

Durant la realització d'aquesta tesi he passat per períodes difícils en els quals he trobat el recolzament incondicional dels meus directors, Julio i Salva. A ells els vull agrair la seua paciència i comprensió durant aquests anys, però sobretot els ànims que m'han donat per a que continuara endavant amb aquest projecte.

També estic molt agraït a Alex, amb el que he col·laborat en diferents articles. A més a més m'ha oferit sempre la seua ajuda donant-me una resposta ràpida a qualsevol dubte que li he plantejat.

Moltes gràcies als tres!

Contents

List of Figures

List of Tables

Abstract

Static Random-Access Memory (SRAM) technology has traditionally been used to implement processor caches since it is the fastest existing RAM technology. However, one of the major drawbacks of this technology is its high energy consumption. To reduce this energy consumption modern processors mainly use two complementary techniques: i) low-power operating modes and ii) low-power memory technologies. The first technique allows the processor working at low clock frequencies and supply voltages. The main limitation of this technique is that manufacturing defects can significantly affect the reliability of SRAM cells when working these modes. The second technique brings alternative technologies such as embedded Dynamic RAM (eDRAM), which provides minimum area and power consumption. The main drawback of this memory technology is that reads are destructive and eDRAM cells work slower than SRAM ones. To address these design concerns, heterogeneous (SRAM/eDRAM) cache organizations have recently been proposed with the aim of reducing consumption without sacrificing the performance.

This thesis presents three main contributions regarding low-power caches and heterogeneous technologies: i) an study that identifies the optimal capacitance of eDRAM cells, ii) a novel cache design that tolerates the faults produced by SRAM cells in low-power modes, iii) a methodology that allows obtain the optimal operating frequency/voltage level when working with low-power modes.

Regarding the first contribution, in this work SRAM and eDRAM technologies are combined to achieve a low-power fast cache that requires smaller area than conventional designs and that tolerates SRAM failures. First, this dissertation focuses on one of the main critical aspects of the design of heterogeneous caches: eDRAM cell capacitance. This capacitance affects both the performance and the energy consumption because when the capacitors' retention time of the eDRAM cells expires, they lose the stored logic value. In this dissertation the optimal capacitance for an heterogeneous L1 data cache is identified by analyzing the compromise between performance and energy consumption. Experimental results show that an heterogeneous cache implemented with 10fF capacitors offers similar performance as a conventional SRAM cache while providing 55% energy savings and reducing by 29% the cache area.

Regarding the second contribution, this thesis proposes a novel organization for a fault-tolerant heterogeneous cache. Currently, reducing the supply voltage is a mechanism widely used to reduce consumption and applies when the system workload activity decreases. However, SRAM cells cause different types of failures when the supply voltage is reduced and thus they limit the minimum operating voltage of the microprocessor. This limitation makes difficult further energy savings, since other parts of the system could correctly work with lower supply voltages. The proposal allows higher reductions on supply voltage than other existing solutions addressing SRAM error detection and correction. In the proposal, memory cells implemented with eDRAM technology serve as backup in case of failure of SRAM cells, because the correct operation of eDRAM cells is not affected by reduced voltages. The proposed architecture has two working modes: high-performance mode for supply voltages that do not induce SRAM cell failures, and low-power mode for those voltages that cause SRAM cell failures. In high-performance mode, the cache provides full capacity, which enables the processor to achieve its maximum performance. In low-power mode, the effective capacity of the cache is reduced because some of the eDRAM cells are dedicated to recover from SRAM failures. Experimental results show that the performance is scarcely reduced (e.g. less than 2.7% across all the studied benchmarks) with respect to an ideal SRAM cache without failures.

Finally, this thesis proposes a methodology to find the optimal frequency/voltage level regarding energy consumption for the designed heterogeneous cache. For this purpose, first SRAM failure types and their probabilities are characterized. Then, the energy consumption of different frequency/voltage levels is evaluated when the system works in low-power mode. The study shows that, mainly due to the impact of SRAM failures on performance, the optimal combination of voltage and frequency from the energy point of view does not always correspond to the minimum voltage.

Resumen

La tecnología *Static Random-Access Memory* (SRAM) se ha utilizado tradicionalmente para implementar las memorias cache debido a que es la tecnología de memoria RAM más rápida existente. Por contra, uno de los principales inconvenientes de esta tecnología es su elevado consumo energético. Para reducirlo los procesadores modernos suelen emplear dos técnicas complementarias: i) modos de funcionamiento de bajo consumo y ii) tecnologías de bajo consumo. La primeras técnica consiste en utilizar bajas frecuencias y voltajes de funcionamiento. La principal limitación de esta técnica es que los defectos de fabricación pueden afectar notablemente a la fiabilidad de las celdas SRAM en estos modos. La segunda técnica agrupa tecnologías alternativas como la *embedded Dynamic* RAM (eDRAM), que ofrece área y consumo mínimos. El inconveniente de esta tecnología es que las lecturas son destructivas y es más lenta que la SRAM. Para atacar este problema de diseño, recientemente se han propuesto organizaciones de cache implementadas con tecnologías heterogéneas con el objetivo de reducir el consumo sin sacrificar las prestaciones.

Esta tesis presenta tres contribuciones principales centradas en caches de bajo consumo y tecnologías heterogéneas: i) estudio de la capacitancia óptima de las celdas eDRAM, ii) diseño de una cache tolerante a fallos producidos en las celdas SRAM en modos de bajo consumo, iii) metodología para obtener la relación óptima entre voltaje y frecuencia en procesadores con modos de bajo consumo.

Respecto a la primera contribución, en este trabajo se combinan las tecnologías SRAM y eDRAM para conseguir una memoria cache rápida, de bajo consumo, área reducida, y tolerante a los fallos inherentes a la tecnología SRAM. En primer lugar, esta disertación se centra en uno de los aspectos críticos de diseño de caches heterogéneas: la capacitancia de los condensadores implementados con tecnología eDRAM. Esta capacitancia afecta tanto a las prestaciones como al consumo del procesador debido a que cuando expira el tiempo de retención de los condensadores, estos pierden el valor lógico almacenado. En esta disertación se identifica la capacitancia óptima de una cache de datos L1 heterogénea mediante el estudio del compromiso entre prestaciones y consumo energético. Los resultados experimentales muestran que condensadores de 10fF ofrecen prestaciones similares a las de una cache SRAM convencional ahorrando un 55% de consumo y reduciendo un 29% el área ocupada por la cache.

Respecto a la segunda contribución, esta tesis propone una organización de cache heterogénea tolerante a fallos. Actualmente, reducir el voltaje de alimentación es un mecanismo muy utilizado para reducir el consumo en condiciones de baja carga. Sin embargo, las celdas SRAM producen distintos tipos de fallos cuando se reduce el voltaje de alimentación y por tanto limitan el voltaje mínimo de funcionamiento del microprocesador. Esta limitación impide que se pueda reducir aún más el consumo, ya que otras partes del sistema podrían bajar más el voltaje sin verse afectado su funcionamiento. La cache heterogénea propuesta permite mayores reducciones del voltaje de alimentación que otras soluciones existentes de detección y corrección de errores basadas en tecnología SRAM. En la cache heterogénea propuesta, las celdas de memoria implementadas con tecnología eDRAM sirven de copia de seguridad en caso de fallo de las celdas SRAM, ya que el correcto funcionamiento de las celdas eDRAM no se ve afectado por tensiones reducidas. La arquitectura propuesta consta de dos modos de funcionamiento: *high-performance mode* para voltajes de alimentación que no inducen fallos en celdas implementadas en tecnología SRAM, y *low-power mode* para aquellos que sí lo hacen. En el modo *high-performance mode*, el procesador dispone de toda la capacidad de la cache lo que le permite alcanzar las máximas prestaciones. En el modo *low-power mode* se reduce la capacidad efectiva de la cache puesto que algunas de las celdas eDRAM se dedican a la recuperación de fallos de celdas SRAM. El estudio de prestaciones realizado muestra que éstas bajan hasta un máximo de 2.7% con respecto a una cache perfecta sin fallos.

Finalmente, en esta tesis se propone una metodología para encontrar la relación óptima de voltaje/frecuencia con respecto al consumo energético sobre la cache heterogénea previamente diseñada. Para ello, primero se caracterizan los tipos de fallos SRAM y las probabilidades de fallo de los mismos. Después, se evalúa el consumo energético de diferentes combinaciones de voltaje/frecuencia cuando el sistema se encuentra en un modo de bajo consumo. El estudio muestra que la combinación óptima de voltaje y frecuencia desde el punto de vista energético no siempre corresponde al mínimo voltaje debido al impacto de los fallos de SRAM en las prestaciones.

Resum

La tecnologia *Static Random-Access Memory* (SRAM) s'ha utilitzat tradicionalment per a implementar les memòries cau degut a que és la tecnologia de memòria RAM més ràpida existent. Per contra, un dels principals inconvenients d'aquesta tecnologia és el seu elevat consum energètic. Per a reduir el consum els processadors moderns solen emprar dues tècniques complementàries: i) modes de funcionament de baix consum i ii) tecnologies de baix consum. La primera tècnica consisteix en utilitzar baixes freqüències i voltatges de funcionament. La principal limitació d'aquesta tècnica és que els defectes de fabricació poden afectar notablement a la fiabilitat de les cel·les SRAM en aquests modes. La segona tècnica agrupa tecnologies alternatives com la *embedded Dynamic* RAM (eDRAM), que ofereix àrea i consum mínims. L'inconvenient d'aquesta tecnologia és que les lectures són destructives i és més lenta que la SRAM. Per a atacar aquest problema de disseny, recentment s'han proposat organitzacions de cau implementades amb tecnologies heterogènies amb l'objectiu de reduir el consum sense sacrificar les prestacions.

Aquesta tesi presenta tres contribucions principals centrades en caus de baix consum i tecnologies heterogènies: i) estudi de la capacitancia òptima de les cel·les eDRAM, ii) disseny d'una cau tolerant a fallades produïdes en les cel·les SRAM en modes de baix consum, iii) metodologia per a obtenir la relació òptima entre voltatge i freqüència en processadors amb modes de baix consum.

Respecte a la primera contribució, en aquest treball es combinen les tecnologies SRAM i eDRAM per a aconseguir una memòria cau ràpida, de baix consum, àrea reduïda, i tolerant a les fallades inherents a la tecnologia SRAM. En primer lloc, aquesta dissertació se centra en un dels aspectes crítics de disseny de caus heterogènies: la capacitancia dels condensadors implementats amb tecnologia eDRAM. Aquesta capacitancia afecta tant a les prestacions com al consum del processador a causa que quan expira el temps de retenció dels condensadors, aquests perden el valor lògic emmagatzemat. En aquesta dissertació s'identifica la capacitancia òptima d'una cache de dades L1 heterogènia mitjançant l'estudi del compromís entre prestacions i consum energètic. Els resultats experimentals mostren que condensadors de 10fF ofereixen prestacions similars a les d'una cau SRAM convencional estalviant un 55% de consum i reduint un 29% l'àrea ocupada per la cau.

Respecte a la segona contribució, aquesta tesi proposa una organització de cau heterogènia tolerant a fallades. Actualment, reduir el voltatge d'alimentació és un mecanisme molt utilitzat per a reduir el consum en condicions de baixa càrrega. Per contra, les cel·les SRAM produeixen diferents tipus de fallades quan es redueix el voltatge d'alimentació i per tant limiten el voltatge mínim de funcionament del microprocessador. Aquesta limitació impedeix que es puga reduir encara més el consum, ja que altres parts del sistema podrien baixar més el voltatge sense veure's afectat el seu funcionament. La cau heterogènia proposta permet majors reduccions del voltatge d'alimentació que altres solucions existents de detecció i correcció d'errors basades en tecnologia SRAM. En la cau heterogènia proposta, les cel·les de memòria implementades amb tecnologia eDRAM serveixen de còpia de seguretat en cas de fallada de les cel·les SRAM, ja que el correcte funcionament de les cel·les eDRAM no es veu afectat per tensions reduïdes. L'arquitectura proposada consta de dues maneres de funcionament: *high-performance mode* per a voltatges d'alimentació que no indueixen fallades en cel·les implementades en tecnologia SRAM, i *low-power mode* per a aquells que sí ho fan. En el mode *high-performance*, el processador disposa de tota la capacitat de la cau el que li permet aconseguir les màximes prestacions. En el mode *low-power* es redueix la capacitat efectiva de la cau posat que algunes de les cel·les eDRAM es dediquen a la recuperació de fallades de cel·les SRAM. L'estudi de prestacions realitzat mostra que aquestes baixen fins a un màxim de 2.7% pel que fa a una cache perfecta sense fallades.

Finalment, en aquesta tesi es proposa una metodologia per a trobar la relació òptima de voltatge/freqüència pel que fa al consum energètic sobre la cau heterogènia prèviament dissenyada. Per a açò, primer es caracteritzen els tipus de fallades SRAM i les probabilitats de fallada de les mateixes. Després, s'avalua el consum energètic de diferents combinacions de voltatge/freqüència quan el sistema es troba en un mode de baix consum. L'estudi mostra que la combinació òptima de voltatge i freqüència des del punt de vista energètic no sempre correspon al mínim voltatge degut a l'impacte de les fallades de SRAM en les prestacions.

Chapter 1

Introduction

This chapter introduces the basic concepts needed to understand and frame this PhD. Dissertation. First, the main reasons that motivated us to do this work are introduced. Then, several technology issues about the use of the different RAM technologies in cache memories are discussed. After that, the problems arising in low-power modes are detailed. Next, the main objectives and contributions of the thesis are described. Finally, the organization of the rest of this dissertation is presented.

1.1 Current cache design and motivation

Cache memories are a critical component in current microprocessors and have been used by computer architects for many decades to reduce the average memory access time. Their design and implementation have evolved and continue still evolving with advances in both processor architecture and technology.

To reduce memory access time, caches are organized in current processors in a multilevel hierarchy, where the first level cache (the closest to the processor core) is designed to be fast and the low-level cache to be large in order to capture memory requests and avoid, as many as possible, the access to the very slow off-chip main memory.

Manufacturing costs and technology issues dictate the design space and rules to be followed for a successful design. To reduce manufacturing costs, the maximum amount of cache storage capacity must be placed within the minimum die size. This is an important design concern, since current caches occupy more than half of overall microprocessor die in current processors. Some of them are deployed with a huge cache capacity; for instance, the IBM Power 8 implements a 96MB on-chip L3 cache. On the other hand, technology constraints need to be analyzed to select the best technology to keep energy cache consumption within a given power envelope. Technology issues, however, introduce novel design concerns that rise as a consequence of shrinking the transistor node and that need to be addressed in order to enable caches to efficiently work in current microprocessors. Examples of major issues are providing support for manufacturing errors due to process variation, and allowing operation with under-threshold voltages to reduce energy consumption.

In summary, we are living nowadays in a exciting scenario regarding cache design that presents many research challenges and a wide design space that offers a high number of possibilities. This thesis is aimed at providing an efficient cache design by combining diverse RAM technologies to address performance, area and energy. Technological issues like manufacturing errors and support for low-voltage thresholds will be also investigated. Next, we discuss the basics of current cache technologies.

1.2 RAM technologies

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, see Figure 1.1). The major drawbacks of SRAM caches are that they

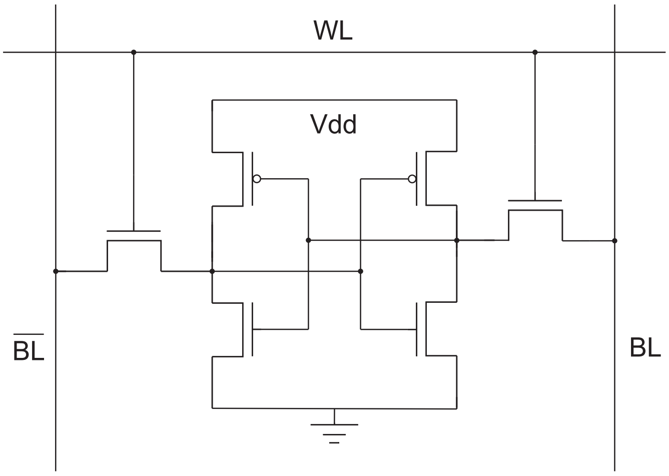


Figure 1.1: 6T SRAM cell

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, see Figure 1.2).

DRAM cells have been considered too slow for processor caches. Nevertheless, technology advances have recently allowed to embed DRAM cells using CMOS technology [30]. 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 large or huge and low-power last-level caches [41, 42, 44, 45].

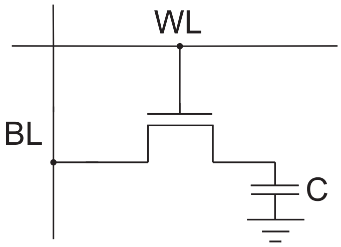


Figure 1.2: 1T-1C DRAM cell

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.

1.3 Manufacturing imperfections

As technology node continues to shrink, dealing with manufacturing imperfections is a major design concern since they affect the manufacturing yield, the energy consumption, and the performance of current and incoming processors.

Because of process variations, the manufacturing process provides transistors with different features (e.g. threshold voltage, channel length, or channel width) so that not all the transistors in a chip are able to properly work with the same voltage and frequency conditions. Due to this fact, manufacturers opt for relaxing the conditions and increase yield by introducing in the market chips that present a noticeable amount of transistors that are not able to properly work under the initial target design goal. For instance, some processors are sold cheaper when their speed is lowered below the originally targeted one to avoid process variation errors.

As a consequence, dealing with process variation in cache memories is a critical design issue. Failures due to the manufacturing process in caches mainly rise in destructive reads, unsuccessful writes, increase of the access times, and content destruction in stand-by mode; known as read, write, access time and hold failures, respectively.

The number of failures due to process variation is determined by the processor working conditions (power supply and frequency). In other words, most errors usually appear in a subset of the working conditions range. For instance, it may happen that an error reading a memory cell appears when the processor works at a given frequency, but that error might not appear when working at a slower frequency (i.e., access time failure). Other failures, such as destructive reads, can be avoided by increasing the voltage supply.

Understanding why errors appear and which conditions allow avoiding them (or most of them) is important for microprocessor architects in order to take the proper architectural design choices to achieve the best tradeoff between performance and power.

1.4 Working under the threshold voltage

Among the different transistor features, the most significant source of random manufacturing variations is the threshold voltage.

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 levels. In *low-power* modes, low voltage/frequency levels are used for energy savings.

SRAM memory cells are more unreliable at low voltages because process variation induces Static Noise Margin (SNM) variability in such cells, which causes failures [32] (known as hard errors) in some of them when working below a certain reliable voltage level, namely Vcc_{min}.

Providing support to work with under-threshold voltages requires different solutions depending on the target RAM technology. On the one hand, regarding SRAM cache arrays, several techniques have been used by the industry [40] as row/column redundancy or Error Detection/Correction Codes (EDC/ECC). However multi-bit error correction codes have high overhead [51] 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 [1, 2, 13, 36, 51, 52]. 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, regarding eDRAM cells, 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.

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.

1.5 Objectives of the thesis

The overall goal of this PhD Dissertation is the design of L1 caches aimed at addressing two critical technological issues, area and energy consumption, while sustaining the performance. At the same time, the design should work considering realistic conditions and therefore transistor parameter variations due to the manufacturing process has been taken in account.

To achieve this overall objective, this work has been developed in three main stages as follows: i) design of a cache combining different technologies (e.g. a low-power technology with a faster one), ii) study of transistor parameter variations, and iii) extend the design developed in i) to consider the effects studied in ii). All these points are addressed from a timing, area, and power points of view.

1.6 Contributions of the thesis

The three major contributions of this thesis are described below:

- The first contribution of this thesis is the work developed to meet the first objective. We focused in a heterogeneous cache design that combines both SRAM and eDRAM technologies. The former allows a fast access time while the latter occupies much less area and reduce energy consumption per bit. The challenge lies on combining them to save area and energy without hurting the performance. For this purpose, architectural mechanisms to handle this cache were devised.

- This thesis presents as second contribution an study about the different types of SRAM failures due to variations on the manufacturing process. For this purpose, a detailed transistor-level simulation was carried out to obtain the probability of failure of the distinct cells.

- The third contribution is the design of a heterogeneous cache considering failures in some SRAM cells due to the effects of manufacturing process. To this end, we extended the original design with new architectural aspects. To provide a more realistic scenario, i) the processor was enhanced to support different voltage/frequency levels, ii) a low-power processor working mode was introduced for further energy savings, and iii) the L1 cache was modelled assuming the probability of failure mentioned above.

1.7 Thesis outline

This dissertation is composed of seven chapters. Chapter 2 presents the state-of-the-art related with the work developed in this thesis. Chapter 3 describes both the tools and simulators used to obtain the simulation results to evaluate the proposals. Chapter 4 introduces the heterogeneous cache, which combines SRAM and eDRAM technologies to reduce area and energy consumption while maintaining performance. Chapter 5 extends the heterogeneous cache design to deal with SRAM cell failures in low-power modes. Chapter 6 consists of main two parts, i) a detailed analysis of the probability of failure of SRAM cells in a wide range of voltages, and ii) the proposal of a new methodology that provides the best voltage/frequency pair in low-power modes taking into account both performance and energy consumption. Finally, Chapter 7 presents some concluding remarks, discusses future work and enumerate the related publications.

Chapter 2

Related work

This chapter provides a summary of the most relevant publications related with this PhD. dissertation. The chapter has been organized in two main sections. First, in Section 2.1, publications related with leakage reduction are described. After that, Section 2.2 focuses on fault-tolerant caches that address low-voltages issues. For this purpose, this section classifies solutions in three main categories according to the type of technique they apply.

2.1 Leakage reduction in SRAM caches

The problem of leakage energy consumption in conventional SRAM caches has been addressed in diverse research works that can be classified in two main 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 [23, 38]. Hence, subsequent accesses to those lines result in a 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 [14, 37], 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 [28] 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. [18] 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 re-writing 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. [53] 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.

2.2 Fault-tolerant caches

Due to inter and intra-die process parameter variations, memory cells that are marginally functional during manufacturing tests can undergo runtime failures due to voltage/thermal noise or aging effects. Depending on the impact of these effects, different segments of a memory array may move to different reliable design corners that can be determined using post-fabrication characterization. Once unreliable blocks have been identified, 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., [25][36][2]) are able to recover the data stored in some defective SRAM cells, but they do not allow high voltage reductions because the additional storage needed for ECC becomes prohibitive. Kim et al. [25] use error correcting codes and redundancy to improve memory cell failures in SRAM caches, considering both systematic and random intra-die variations. This study takes into account three device parameters; the channel width, the channel length, and the threshold voltage. In [36], 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.* [2] 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. Respect to DRAM technology, refresh power potentially represents a large fraction of the overall system power, particularly during low-power states when the processor is idle. In [52], Wilkerson et al. reduce cache refresh power by increasing the refresh time from 30μs (worst-case) to 440μs. This increase reduces power substantially but causes errors due to capacitor discharges. This problem can be solved by using costly ECC codes.

Schemes in the second category (e.g., [51][1]) 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 [51], 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 [1], 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 [4], 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., [11][13]) achieve this goal but increase the area occupation, while eDRAM-based techniques like the HER cache [29] do not present this drawback. In [11], 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 [13] 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.

Chapter 3

Experimental Framework

This chapter focuses on the simulation tools that have been used to develop the research presented in this work. First, a general picture of the simulation environment is presented in Section 3.1 which illustrates the multiple simulation tools used to obtain performance, energy, and area results. After that, a brief description of each simulator tool is provided in sections 3.2, 3.3, and 3.4. At the end of the chapter, Section 3.5 describes the benchmarks used as workload in the simulations.

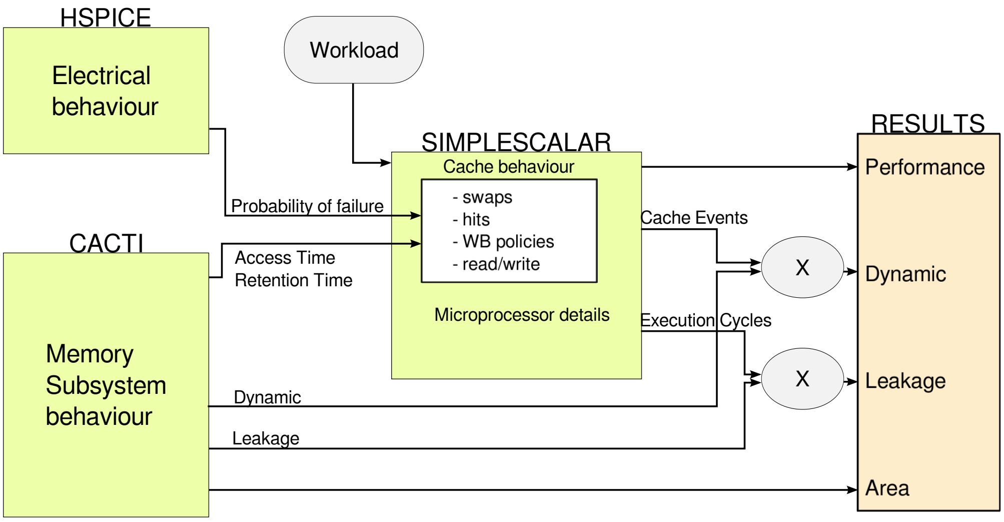


FIGURE 3.1: Block diagram of the simulation environment

3.1 General view of the simulation environment

This thesis presents and analyzes result values of performance, energy, and area taking into account failures due to manufacturing variations. Therefore, different abstractions levels are involved: i) transistor level, ii) cache organization level, and iii) processor microarchitecture level. Nowadays, there is not a single simulator framework covering all these levels. Instead, it is required to combine different tools to perform the experiments. Figure 3.1 depicts the composed framework, showing the combined tools, as well as their interactions.

The figure shows that the central tool is SimpleScalar [7], a detailed processor simulator, which has been modified to accept failure probabilites and latency information provided, respectively, by other two simulation programs, HSPICE [17] and CACTI [46, 47]. The results provided by this simulation environment are performance, dynamic energy consumption, static energy consumption (also known as leakage), and area. Performance is expressed in instructions per cycle (IPC) and it is directly provided by SimpleScalar. Area is directly provided by CACTI. Dynamic energy consumption and leakage results, however, are calculated using results provided by both SimpleScalar and CACTI.

In the next sections each tool used in the simulation environment is described. In Section 3.2, the HSPICE circuit simulator is presented. In Section 3.3, the CACTI memory subsystem simulator is introduced. Next, Section 3.4 presents the SimpleScalar processor architecture simulator, which has been extended to model the proposals presented in this PhD. dissertation. Finally, Section 3.5 presents the simulated applications used as workload in SimpleScalar.

3.2 HSPICE

HSPICE from Synopsys is one of the most trusted and comprehensive circuit simulators. HSPICE is based on the SPICE family of circuit simulators, which allow simulating circuitry at transistor level. In our simulation environment, HSPICE is used to obtain the probability of failure of SRAM cells used in Chapter 6. Among the different SPICE-based simulators, we choose HSPICE because HSPICE significantly reduces the amount of time needed to calculate the probability of failure. Note that to obtain reliable results, a minimum of 40000 samples must be simulated with the MonteCarlo methodology [39].

In addition, SPICE-based simulators are used in this PhD. dissertation to test the internal transfers and electronic behavior of the macrocell presented in Chapter 4. Among the available SPICE-based simulators, we found that Ngspice [34] was the most suitable to simulate in detail these internal transfers. Ngspice is a mixed-level/mixed-signal circuit simulator whose code is based on three open source software packages: Spice3f5, Cider1b1 and XSPICE. Ngspice accurately simulates MOSFET behavior since it uses a BSIM4 MOSFET model. All the simulations performed with Ngspice are based on Predictive Technology Models (PTM) [55].

3.3 CACTI

To analyze the hybrid cache as well as its associated architectural mechanisms, the CACTI cache simulation tool, developed by Hewlett-Packard, has been used. CACTI integrates a cache and memory access time, cycle time, area, leakage, and dynamic power model that allows obtaining different metrics for cache and main memory organizations built with SRAM and eDRAM technologies. CACTI is intended to help computer architects to better understand the performance tradeoffs inherent to memory system organizations. CACTI is widely used in the computer architects community and its results have been published at top international conferences related with computer architecture (e.g. MICRO and ISCA).

In particular, in this PhD. dissertation, CACTI has been used to obtain: i) the area occupied by the studied cache architectures, ii) the dynamic energy consumption due to the different simulated cache events (e.g., conventional cache accesses, replacements, etc.), iii) the static energy consumed each processor cycle, and iv) the retention time of eDRAM cells in pure eDRAM and hybrid SRAM/eDRAM memory organizations.

3.4 SimpleScalar

SimpleScalar is a cycle-accurate simulator that models in detail the microarchitecture and the memory hierarchy of a superscalar processor. It enables the simulation of the execution of real programs in a wide range of processor designs. SimpleScalar is open source and it is widely used by universities and enterprises for research and instruction goals. For instance, in 2000, more than one third of all papers published in top computer architecture conferences used SimpleScalar to evaluate their proposals.

In particular, in this PhD. dissertation, SimpleScalar has been extensively modified to: i) model the specific characteristics of the different memory architectures, ii) account several cache events to compute the energy consumption, and iii) study the impact on processor performance of the proposed designs.

3.5 Workload

The SPEC CPU2000 V1.3 benchmark suite has been used as workload in this PhD. dissertation. SPEC CPU is an standard application suite for evaluating computer performance released by The Standard Performance Evaluation Corporation (SPEC).

In contrast to synthetic benchmarks, these benchmark applications are developed from actual end-user applications. SPEC CPU provides the source code of different compute-intensive applications that emphasize the performance offered by processors, memory architectures, and compilers. SPEC CPU benchmarks are classified in two sets according to the type of compute-intensive application: i) CINT2000, which contains 11 integer

Name	**Description**
164.gzip	Data compression utility
175.vpr	FPGA circuit placement and routing
176.gcc	C compiler
181.mcf	Minimum cost network flow solver
186.crafty	Chess program
197.parser	Natural language processing
252.eon	Ray tracing
253.perlbm	Perl
254.gap	Computational group theory
255.vortex	Object Oriented Database
256.bzip2	Data compression utility
300.twolf	Place and route simulator

Table 3.1: CINT2000 applications

Name	**Description**
168.wupwise	Quantum chromodynamics
171.swim	Shallow water modeling
172.mgrid	Multi-grid solver in 3D potential field
173.applu	Parabolic/elliptic partial differential equations
177.mesa	3D Graphics library
178.galgel	Fluid dynamics: analysis of oscillatory instability
179.art	Neural network simulation; adaptive resonance theory
183.equake	Finite element simulation; earthquake modeling
187.facerec	Computer vision: recognizes faces
188.ammp	Computational chemistry
189.lucas	Number theory: primality testing
191.fma3d	Finite element crash simulation
200.sixtrack	Particle accelerator model
301.apsi	Solves problems regarding temperature, wind, velocity and distribution of pollutants

Table 3.2: CFP2000 applications

applications, and ii) CFP2000, which includes 14 floating-point applications. Most applications in CINT2000 are written in C, except 252.eon, which is written in C++. In contrast, in CFP2000, 6 applications are written in Fortran-77, 4 in Fortran-90 and only 4 in C. Tables 3.1 and 3.2 enumerate the applications included in CINT2000 and CFP2000, respectively.

Chapter 4

Heterogeneous Caches

This chapter presents the design and implementation of heterogeneous set associative caches, composed of both, SRAM and eDRAM cells. The basic element is the macrocell, which implements an n-bit cell.

The rest of this chapter is organized as follows. Section 4.1 summarizes the macrocell design, its working behavior, and presents a possible implementation of an M-Cache. Section 4.2 analyzes how the eDRAM capacitance impacts on retention and access times and estimates the area savings provided by the macrocell. Section 4.3 shows performance and energy results for different cache organizations. Section 4.4 introduces an alternative implementation of the design presented in Section 4.1. Finally, a summary of this chapter is drawn in Section 4.5.

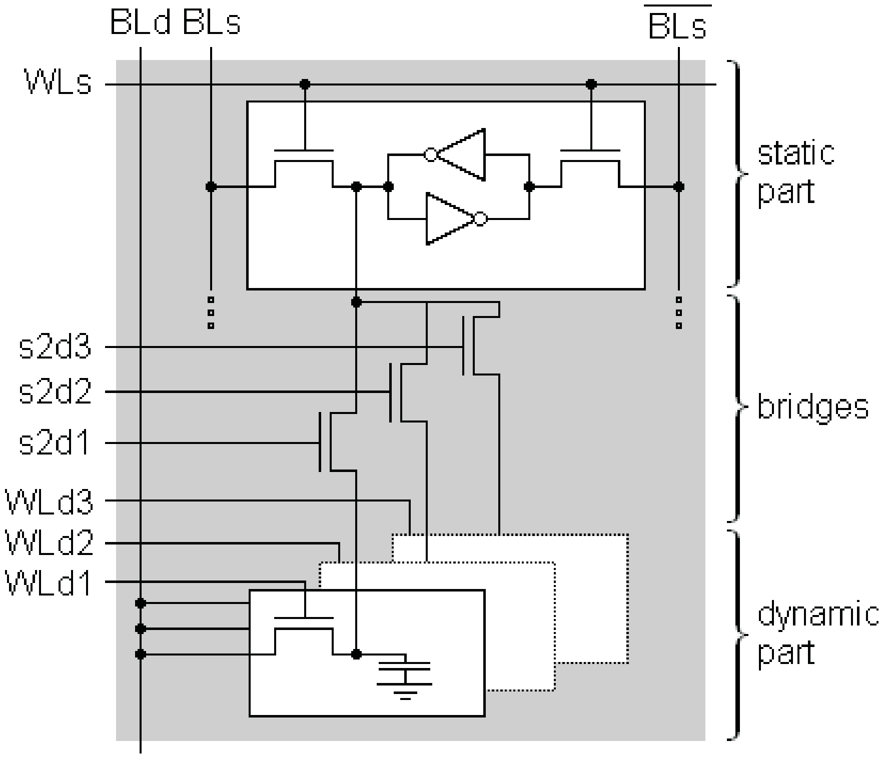


FIGURE 4.1: 4-bit macrocell block diagram

4.1 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.

4.1.1 Macrocell internals

The main components of an n-bit macrocell are one SRAM cell, n-1 eDRAM cells, and *bridge* transistors that communicate SRAM with eDRAM cells. Figure 4.1 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.

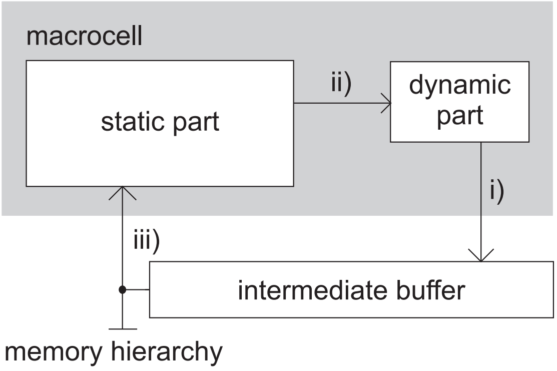


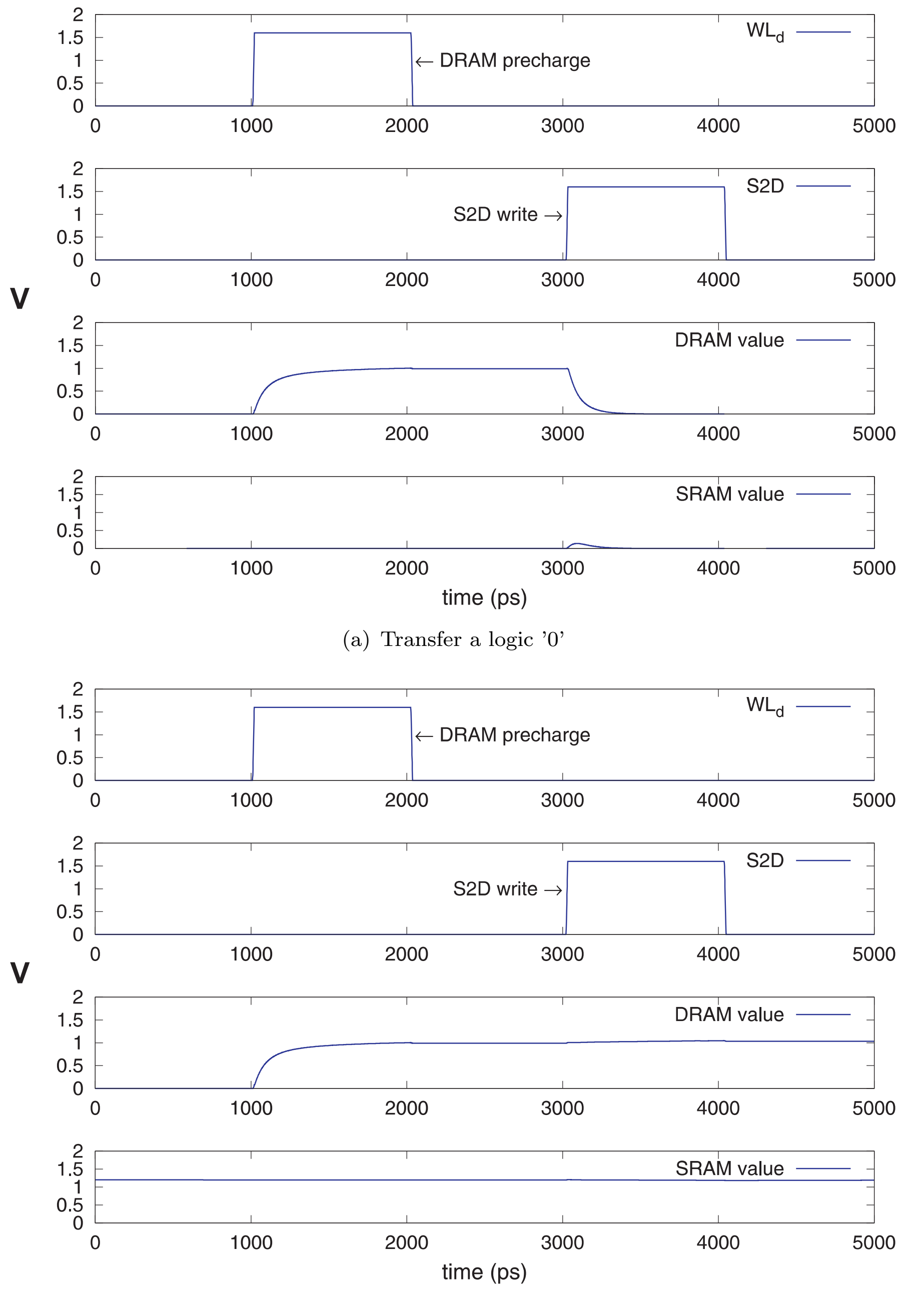
FIGURE 4.2: Swap operation block diagram

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 [37] 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 16KB-4way 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 Figure 4.2. 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. Values of key technology metrics of transistors have been taken from the 2007 ITRS [20] for 45nm technology node.

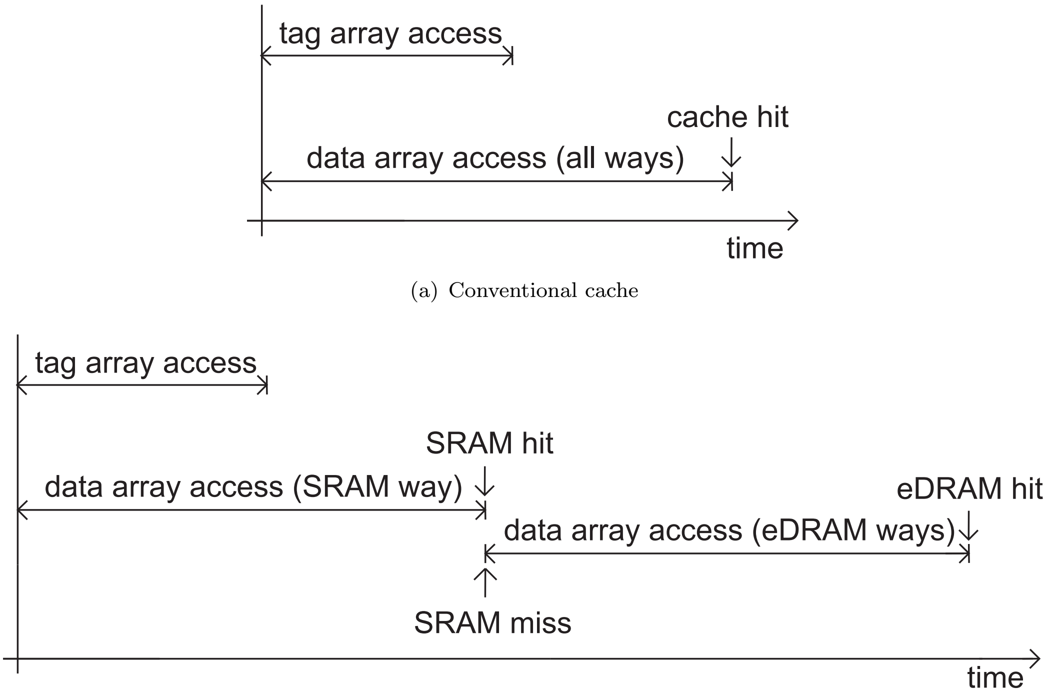


(a) Transfer a logic '0'

(b) Transfer a logic '1'

Figure 4.3: Operation detail of the internal transfer

In particular, we have evaluated: i) whether the eDRAM capacitor is properly charged, and ii) 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 wordline controlled by WLd_i must be boosted to a *Vpp* voltage (i.e., Vpp = Vdd + Vth). Furthermore, in the case of the bridge transistor, the wordline controlled by


tag array access
cache hit
data array access (all ways)
time
tag array access
SRAM hit
data array access (SRAM way)
eDRAM hit
data array access (eDRAM ways)
SRAM miss
time

(a) Conventional cache

(b) M-Cache

FIGURE 4.4: Tag and data arrays access timing diagram

$s2d_i$ must also be boosted to the same voltage. For eDRAM cells and 45nm technology node, Vth and Vdd are usually set to 0.4V and 1.1V, 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) [50]. 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. Figure 4.3 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.

4.1.2 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. 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 Figure 4.4(a). Proceeding in this way, 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, 19]. 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 Figure 4.4(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 4.1), 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 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.

	eDRAM capacitance (fF)										
Freq.	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**	∞

TABLE 4.1: Retention time (retention times exceeding 50K cycles are shown in boldface)

	eDRAM capacitance (fF)										
Freq.	0.04	0.08	0.16	0.31	0.63	1.25	2.5	5	10	20	∞
Perf. degradation (%)	1.48	1.26	0.83	0.55	0.33	0.22	0.11	0.06	0	0	0

TABLE 4.2: Performance degradation (32KB-4way M-Cache with 1GHz for integer benchmarks)

Nevertheless, the performance and energy penalties related to the sentry bit have been taken into account in the results.

4.2 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. Assuming no wasting area, the results presented in this section were calculated with the CACTI 5.3 tool [46, 47], 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 [54], 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.

4.2.1 Retention time

Retention time values for each eDRAM capacitance were quantified in processor cycles for three different machine speeds (1GHz, 2GHz, and 3GHz) resembling those of existing commercial processors. To estimate the capacitor charge for each processor speed, a discrete amount of values ranging from 0.04fF to 20fF were analyzed. Table 4.1 shows the retention time. In [49] it was shown that a 50K-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 [43] in a superscalar processor with the same machine parameters as the assumed in this work. Thus, eDRAM data which is not referenced beyond this 50K processor cycles

	Conv-Cache			WP-Cache			M-Cache		
Frequency	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 4.3: Access time (in processor cycles) for the studied cache schemes

has a very low probability to be referenced again. Results show that the minimum capacitances required to sustain a 50K-cycle retention time are 10fF for 1GHz and 5fF for 2- and 3-GHz. Table 4.2 shows the performance degradation obtained for a 32KB-4way M-Cache with 1GHz processor speed for integer benchmarks. It can be seen that 10fF is enough to completely avoid performance losses (see Section 4.3.1).

4.2.2 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 (16KB and 32KB) 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 4.3 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 16KB-2way M-Cache with 1GHz frequency, the

	Area			
# data bits	SRAM	eDRAM	Macrocell	Area 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 4.4: Cell areas (μm^2) and reduction for different macrocell sizes

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 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 0.04fF to 20fF), this fact has negligible impact since it becomes masked when the access time is quantified in processor cycles.

4.2.3 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 45nm 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 [24]. Indeed, as shown in [26], the capacitance values analyzed in this work can be obtained with trench capacitors.

Table 4.4 shows the area (in μm^2) of the conventional SRAM and eDRAM cells, and the macrocell varying the number of data bits from 1-bit to 4-bit. As expected, the area reduction increases with the number of bits. The 2-bit and 4-bit macrocells obtain an area reduction of 4% and 29% compared to 2 SRAM cells and 4 SRAM cells, respectively. Notice that the macrocell design does not allow a 1-bit macrocell.

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
# Int/FP ALUs	4 Int/4 FP
Memory hierarchy	
Memory ports	4
L1 data cache	Variable geometry. Latency: see Table 4.3
L2 data cache	512KB, 8 ways, 64 byte-line, 10 cycles
Main memory	100 cycles

Table 4.5: Machine parameters

4.3 Experimental evaluation

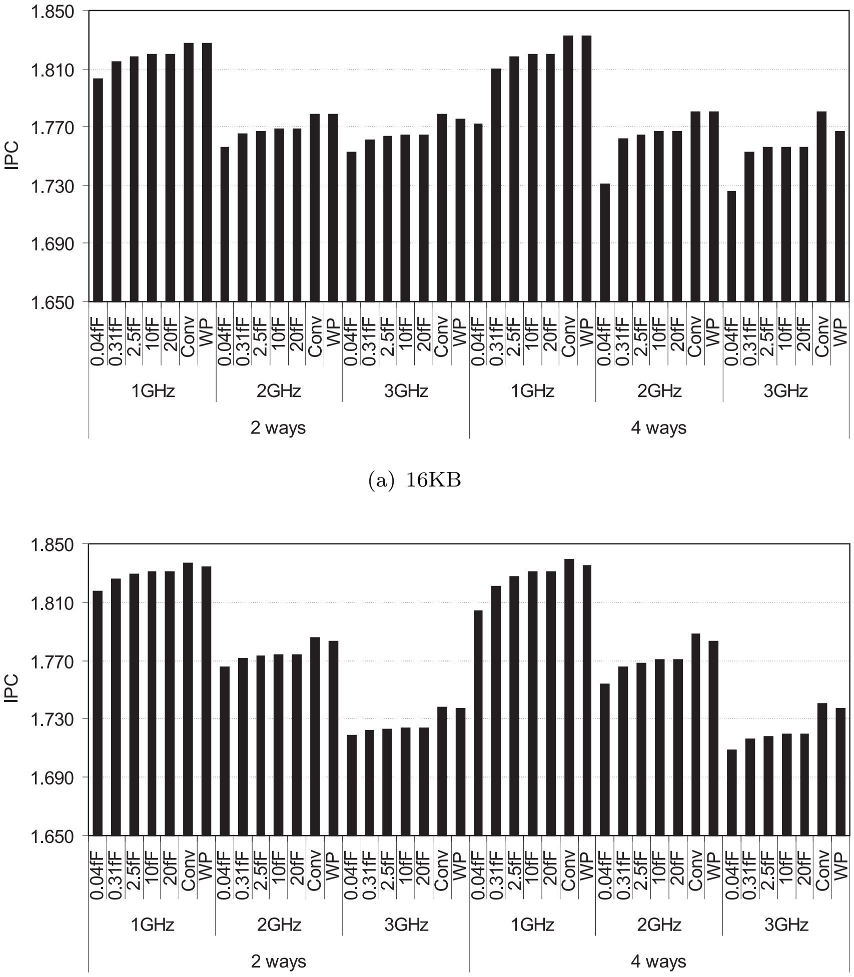
Both dynamic and leakage energy have been analyzed in this work for a 45nm technology node. The corresponding values for each cache scheme were obtained per access and per processor cycle for dynamic and leakage energy, respectively, with CACTI. In addition, the SimpleScalar framework was extended to 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 SimpleScalar 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 500M instructions after skipping the initial 1B instructions. Table 4.5 summarizes the architectural parameters used in the experiments.

4.3.1 Performance analysis varying capacitance & processor frequency

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.

Figures 4.5 and 4.6 show the results. For the sake of clarity, only a subset of the capacitance values analyzed in Section 4.2 are represented (i.e., 0.04fF, 0.31fF, 2.5fF,



(a) 16KB

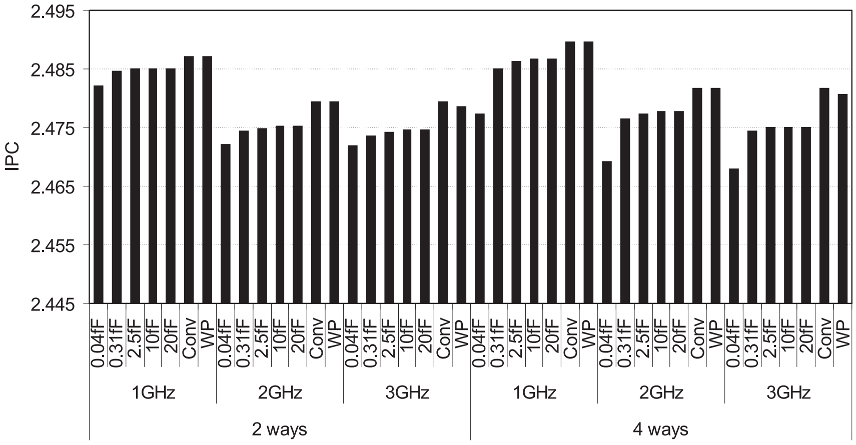
(b) 32KB

FIGURE 4.5: Performance varying the eDRAM capacitance and processor frequency for the analyzed cache schemes (Integer benchmarks)

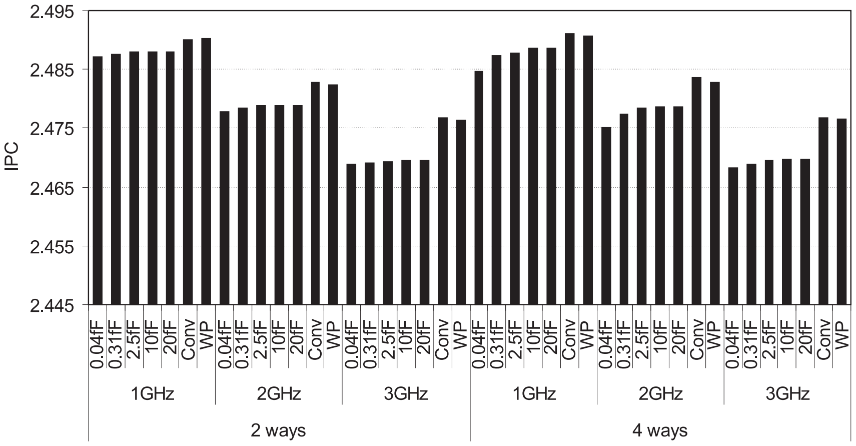
10fF, and 20fF). As expected, the performance grows with the capacitance. Although for some configurations 2.5fF is enough capacitance to achieve the maximum performance regardless the processor frequency, a capacitance of about 10fF ensures the maximum performance for all configurations. Of course, the 20fF 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.

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, i) accessing a non-MRU block takes, in general, more time with the way prediction schemes (WP-Cache and M-Cache, see Table 4.3), and ii) the cache cannot be



(a) 16KB

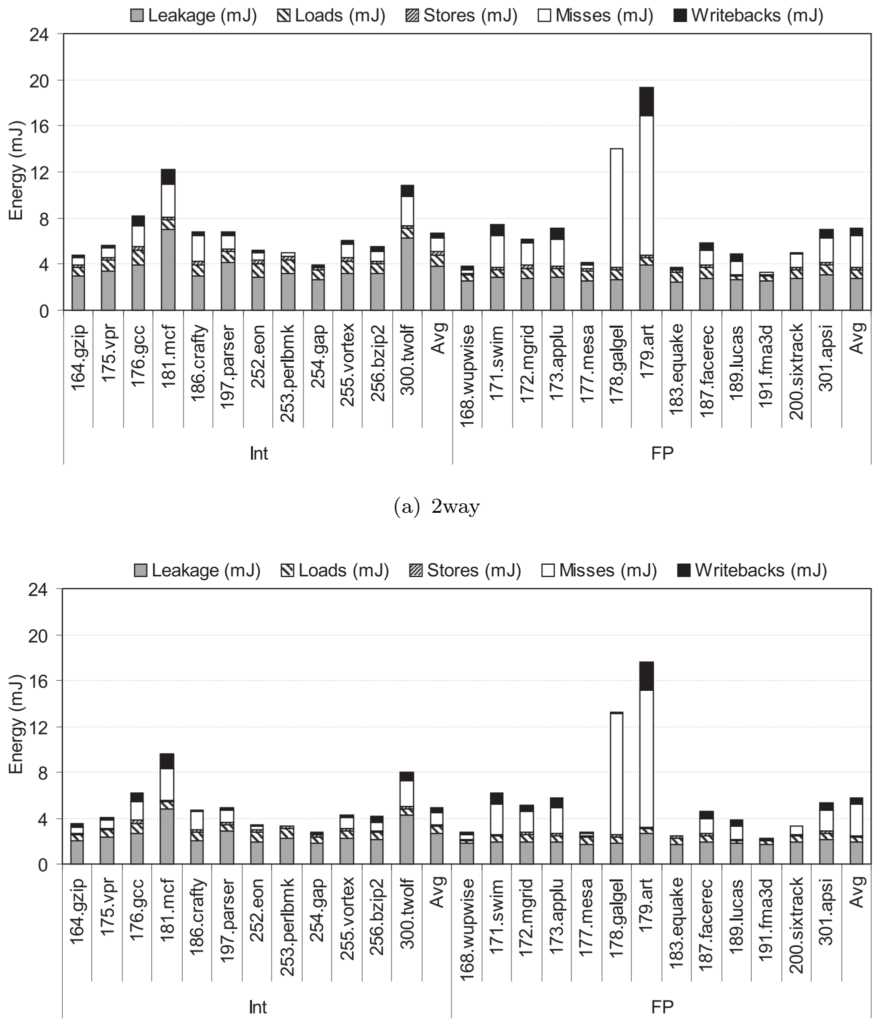


(b) 32KB

Figure 4.6: Performance varying the eDRAM capacitance and processor frequency for the analyzed cache schemes (Floating Point benchmarks)

accessed while a block is being swapped (i.e., internal transfer). Nevertheless, using 10fF, performance losses are always lower than 2% with respect to the Conv-Cache scheme. The worst results are those achieved by the 16KB-4way organization with 0.04fF at 1GHz 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.



(a) 2way

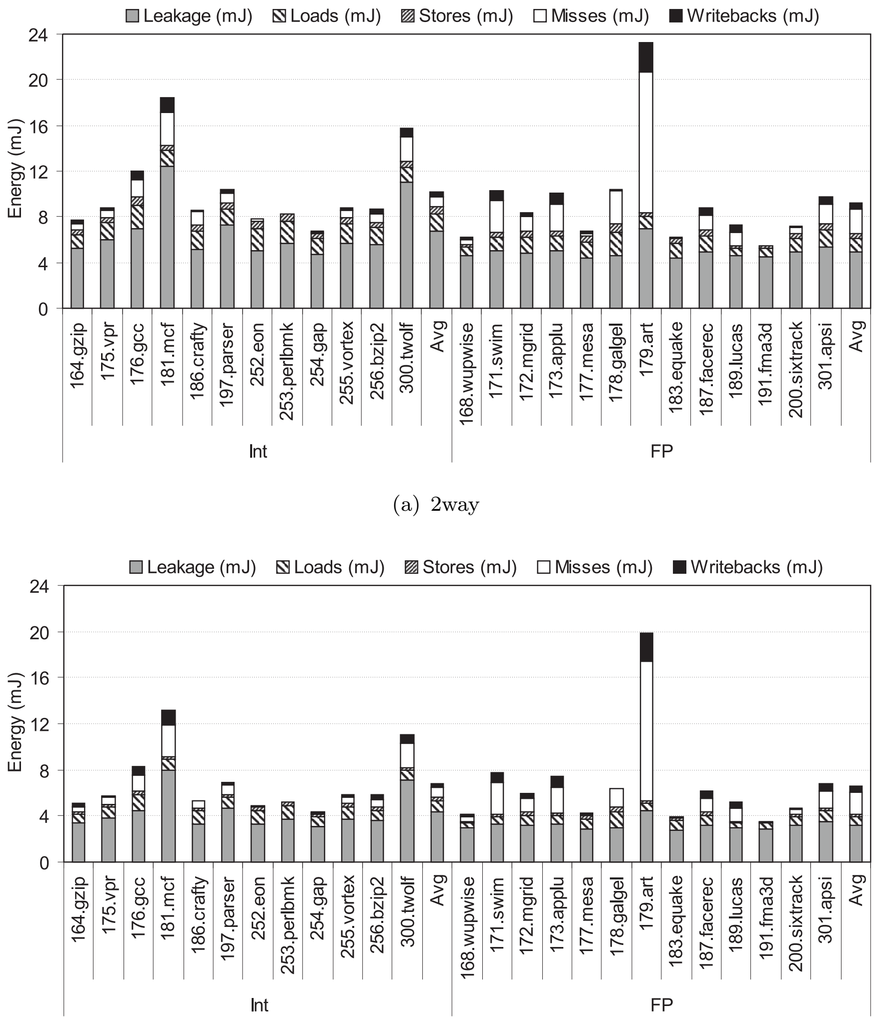
(b) 4way

Figure 4.7: M-Cache energy consumption (mJ) per benchmark. 16KB

4.3.2 Energy distribution analysis

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. Figures 4.7 and 4.8 show the M-Cache data array energy results (in mJ) for each benchmark.

As obtained in Section 4.3.1, the capacitance has been fixed to 10fF 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 1GHz. Results for the other frequencies are not shown since they exhibit the same energy trend.



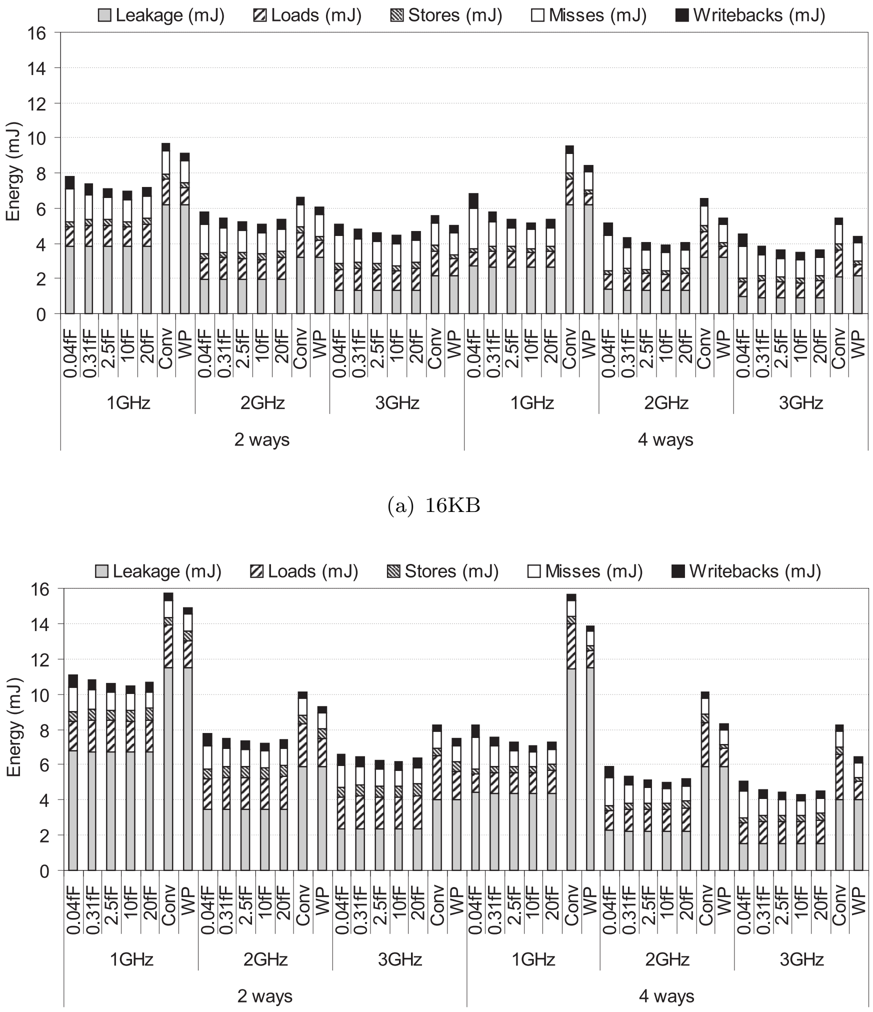
(a) 2way

(b) 4way

FIGURE 4.8: M-Cache energy consumption (mJ) per benchmark. 32KB

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., 4KB in the 16KB-4way cache against 8KB in the 16KB-2way 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.



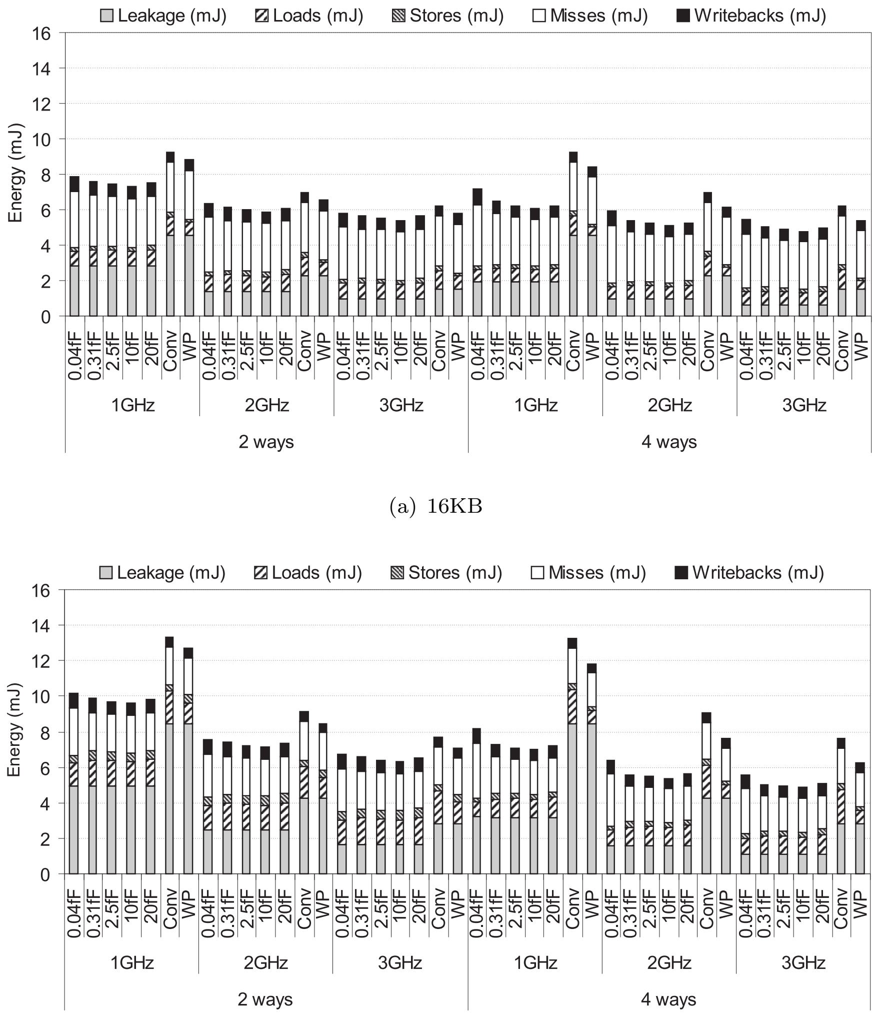
(a) 16KB

(b) 32KB

Figure 4.9: Energy consumption (mJ) varying the capacitance and the frequency for the analyzed cache schemes. Integer benchmarks.

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.



(a) 16KB

(b) 32KB

FIGURE 4.10: Energy consumption (mJ) varying the capacitance and the frequency for the analyzed cache schemes. Floating Point benchmarks.

4.3.3 Impact of capacitance & processor frequency on energy

Figures 4.9 and 4.10 show 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 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 32KB-2way and a 32KB-4way 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-way 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, 10fF 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.

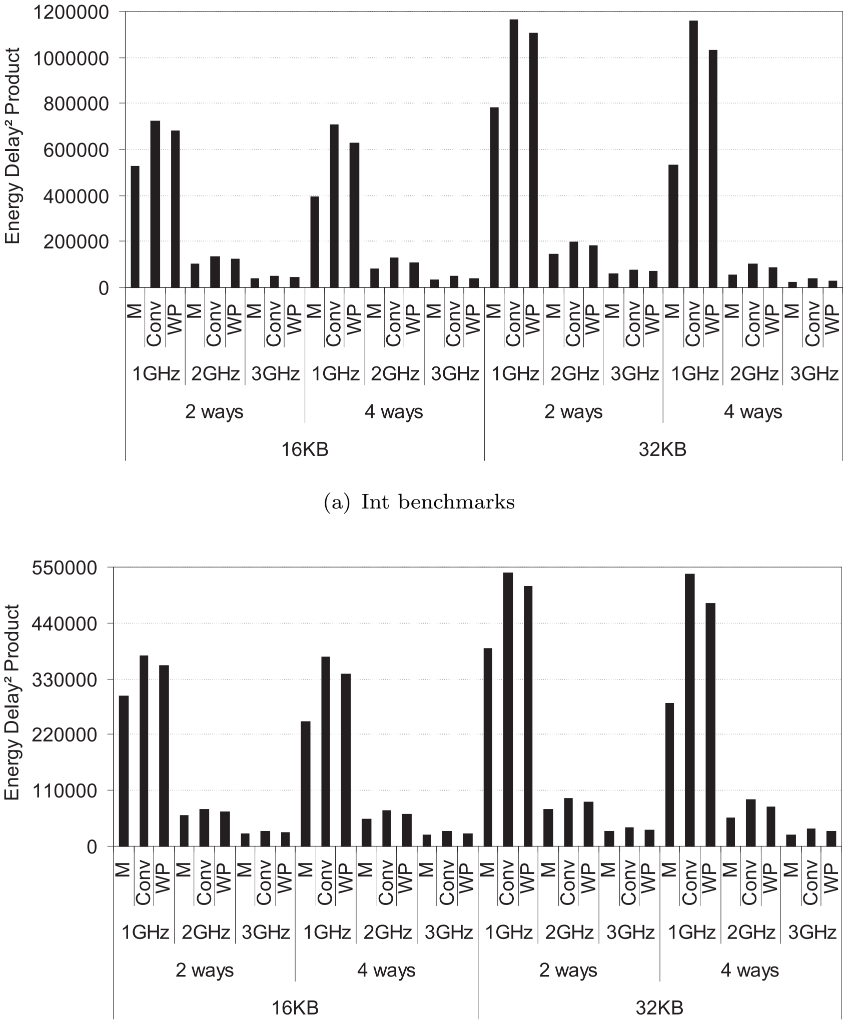
The WP-Cache presents the lowest dynamic energy consumption, closely followed by the M-Cache with 10fF capacitors. The reason is that a hit in the predicted way saves a significant amount of dynamic energy in both schemes. The differences between both schemes mainly appear due to the internal transfers 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 32KB-2way and a 32KB-4way 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 10fF exhibits, depending on the processor frequency and type of benchmark, a total energy reduction up to 33% and 55% for 2-way and 4-way set-associative caches, respectively.

4.3.4 Trade-off between energy consumption and performance

Neither performance nor energy can be evaluated in an isolated way in current systems, but rather a trade-off must be reached between them. Figure 4.11 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 10fF.



(a) Int benchmarks

(b) FP benchmarks

Figure 4.11: Energy delay^2 product for different cache organizations and processor frequencies

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, 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 10fF compared to 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.

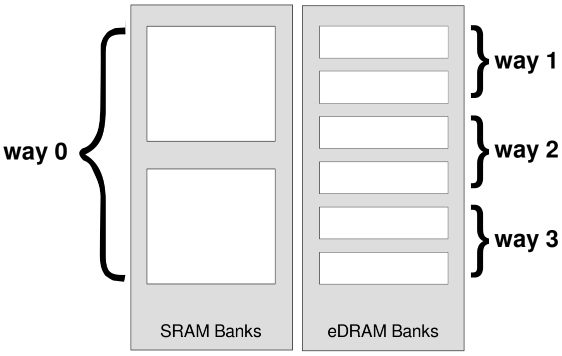


FIGURE 4.12: Alternative heterogeneous cache implementation using independent banks instead of macrocells

In particular, compared to Conv-Cache with the same capacity and associativity, the M-Cache reduces the energy delay^2 product up to 33% and 54% for 2-way and 4-way set-associative caches, respectively. These results are in context with the percentages of the energy savings shown in Section 4.3.2.

4.4 Implementation constraints and alternative implementation

From the implementation point of view, it is required to waste some spare area due to differences in current manufacturing processes of SRAM and eDRAM technologies. To avoid this area waste, we modified the heterogeneous cache implementation as shown in Figure 4.12. This alternative implementation segregates the cells that compose one macrocell in two types of cell banks: SRAM banks and eDRAM banks. By avoiding cells implemented in different technologies to be integrated in the same bank, this alternative implementation does not incur in any significant area waste due to manufacturing constrains.

However, in this implementation, internal transfers are not possible through the bridge transistors implemented in the original macrocell. Therefore, in this implementation, internal transfers must be done through intermediate buffers (e.g., the sense amplifiers). Chapter 5 proposes a new fault-tolerant heterogeneous cache architecture that is based on this alternative implementation.

4.5 Summary

This chapter has presented the macrocell-based cache (M-Cache), which combines SRAM and eDRAM technologies. The macrocell has been shown as an efficient device to implement cache memories, since its design deals with energy consumption, area, and access time.

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 10fF is the optimal capacitance since it is enough to avoid performance losses and exhibits the lowest energy consumption for processor frequencies higher than 1GHz in L1 M-Caches.

Compared to a conventional cache with the same storage capacity and associativity degree, an M-Cache with 10fF capacitance obtains an energy reduction about 33% and 55% for 2-way 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 45nm technology nodes.

The implementation described and studied in this chapter has some drawbacks due to manufacturing constraints. At the end of the chapter an alternative implementation has been outlined as a possible solution to solve these constraints.

Following chapters will focus on more realistic scenario where SRAM cells suffer variations due to the manufacturing process, which affects to the stability of the cells, causing bit failures.

Chapter 5

Fault-Tolerant Heterogeneous Caches

In this chapter we propose the Hard Error Recover (HER) cache, which combines SRAM and eDRAM technologies to deal with hard-error recovery at low-power mode while sustaining the performance at high-performance mode. Low-power modes rely on low frequencies and voltages to reduce energy budget. However, manufacturing-induced parameter variations can make SRAM cells unreliable at supply voltages below Vcc_{min}. Moreover, the probability of failure increases exponentially in such voltages.

The HER cache has two operation modes, high-performance and low-power. At high-performance, the HER cache works using the entire storage capacity and architectural decisions are devised to address performance losses. At low-power, the proposal provides 100% SRAM fault-coverage in set-associative L1 data caches while reducing area and power with respect to a conventional cache.

The remainder of the chapter is organized as follows. Section 5.1 presents the main reasons that motivated this work. Section 5.2 describes the working behavior of the proposal in both high-performance and low-power operation modes. Section 5.3 analyzes the access time for each operation mode. Section 5.4 shows performance, leakage power, dynamic energy, and area results. Finally, a summary of the chapter is drawn in Section 5.5.

Ref.	Scheme name	Coverage	Vmin	Cache	Freq. in *lp*	IPC in *hp*	IPC in *lp*	Area	Power
[36]	Rel. driven	6/64 bits	0.800V	2MB, L2	na	na	na	+31%	-6%
[2]	MS-ECC	4/64bits	0.490V	32KB, L1 2MB, L2	500MHz	na	+10%	na	-71%
[1]	Var. aware	419/32KB	na	32KB, L1	na	+5.7%	na	+0.5%	+1.8%
[51]	Word dis. Bit fix	4/256 bits 20/512 bits	0.490V 0.475V	32KB, L1 2MB, L2	500MHz 500MHz	-5% -5%	-10.7% -10.7%	+15% +7%	-85% -85%
[13]	RENT cache	not required	0.500V	512B, filter 7KB, L1	10MHz	-2.1%	na	na	-86%
[52]	Hi-ECC	5/1KB line	Refresh	128MB, L3	2 GHz	na	-0.1%	+2%	-93%
-	HER cache	100% SRAM	0.500V	32KB, L1	500MHz	-1.9%	-2.7%	-37%	-68% dy. -89% le.

Legend: na: not available, lp: low-power, hp: high-performance

Table 5.1: Error-failure schemes comparison

5.1 Motivation

The main reason of the low fault-coverage supported by existing proposals is that the devised solutions must trade off coverage for overhead (area, energy, performance, etc.). Table 5.1 summarizes, for a representative subset of the recent proposals, performance (both in high-performance (*hp*) and low-power (*lp*) modes), power and area. Basically, these solutions allow the system to work below Vcc_{min} by disabling those segments of the cache where one or more bits can fail, thus reducing the effective storage capacity. As observed, the highest fault coverage is achieved by [36], which is still less than 10%. Providing higher coverages in these proposals could become prohibitive in terms of area, delay, or power.

Unfortunately, technology projections [21] foresee that the ultimate nanoscale device will have a high percentage of non-functional devices from the beginning due to the high degree of variation produced in the manufacturing process. In this context, future fault tolerant caches must support a high percentage of failures. As example, we measured the probability of failure for a 22nm node ranging Vcc from 0.4 to 1V, and varying Vth due to process variation from 10% to 70%. Figure 5.1 shows the results. As observed, for a realistic 25% Vth variation and 0.4V power supply (near threshold voltage), the probability of cell failure is by 20%, which is twice as large as the supported by the best existing proposal [36].

In short, the presented state-of-the-art proposals will not meet the coverage requirements of future technologies. Moreover, in [35], it is further argumented why less than 100% fault-coverage is unsuitable for fault-dominated future technology nodes.

With this aim, the proposal combines eDRAM and SRAM technologies. eDRAM cells back up the SRAM cells (that may be faulty due to process variation), thus achieving 100% fault-coverage due to this kind of errors. In the following sections, the HER cache

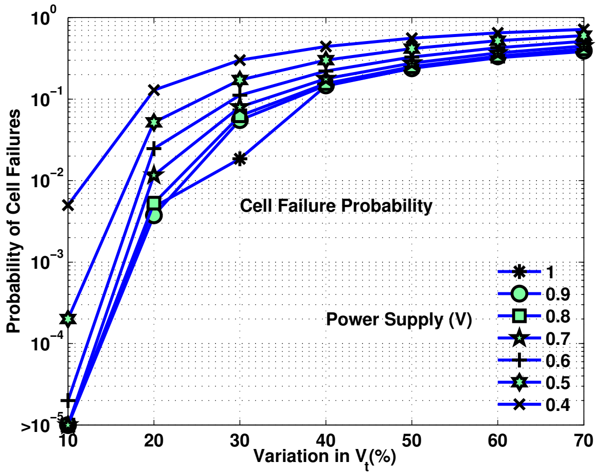


FIGURE 5.1: Probability of cell failures varying the power supply and the variation in Vt

is described along its operating modes, timing, overheads, and manufacturability issues due to the use of both technologies.

5.2 Bank-based HER cache

This section presents the proposed Hard Error Recover (HER) cache architecture for both high-performance and low-power modes. The HER cache is based in the bank-based heterogeneous cache organization presented in Chapter 4. The proposed technique presents four main contributions over prior proposals:

1. 100% fault-coverage for process-variation induced faults.
2. Just $1/n$ of storage capacity sacrificed for failure recovery in an n-way set-associative cache. For instance, sacrificing 12.5% of storage capacity in an 8-way set-associative cache allows 100% coverage. Notice that this sacrifice does not impact negatively on area, since the use of eDRAM cells allows reducing the silicon area too.
3. At high-performance mode, the whole cache storage capacity (i.e. all ways) is enabled.
4. No refresh operation is implemented, since eDRAM periodic refresh can represent an important fraction of the system power [52].

Since the control bits and the tag array is much smaller than the data array, this work focuses on the potential benefits in the data array, and assumes that special low-power non-defective cells [27] are used for the whole tag array and control bits.

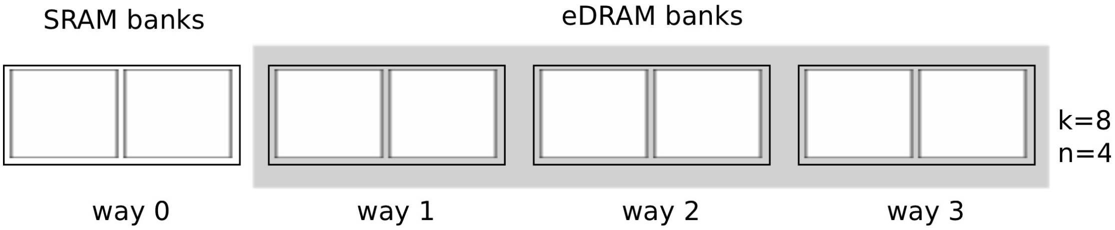


FIGURE 5.2: Block diagram of a 4-way HER cache architecture implemented with 8 banks

5.2.1 High-performance mode

The HER cache uses k cache banks to implement an n-way set-associative cache, where k/n banks are implemented in SRAM technology and the remaining $k - k/n$ in eDRAM technology. Figure 5.2 depicts a block diagram of a 4-way set-associative HER cache for $k = 8$. The tags and the control bits are stored in a different array (not shown in the figure).

The whole structure is accessed as a way-prediction cache, similarly as done in recent commercial processors like the IBM POWER7 [41], as follows. In the first cycle, the tags of all ways are checked and -only- the SRAM data way is read. On a hit in the SRAM way, no eDRAM bank is accessed; so avoiding unnecessary accesses to eDRAM banks. After checking tags, if a hit occurs in any eDRAM block, the associated data is accessed (i.e., a destructive read occurs), incurring in additional penalty cycles and data is delivered to the CPU. Then, a swap operation between this block and the one stored in the SRAM way is triggered. Hence, the new MRU block is always stored in an SRAM bank while the previous MRU block is written back to an eDRAM bank. During the swap operation, the two banks involved are unavailable for servicing loads/stores.

This behavior provides good performance since most accesses hit the MRU block in L1 data caches (see Section 5.4.1.2). Thus, our proposal ensures that the MRU block of each cache set is always allocated in an SRAM bank (i.e., way 0 in the figure) by swapping cache lines. Note that in a conventional cache the MRU block can be allocated in any way.

To estimate the required eDRAM retention time, let's see a typical generation time of a generic block. The generation time starts when it is brought into the cache and finishes when the block is evicted, and is split into live and dead times. Figure 5.3 depicts these times for block A. The live time refers to the elapsed time since the block is fetched (*t1*) until its last access (*t4*) before eviction, and dead time refers to the remaining time until the block is evicted (*t4* to *t5*).

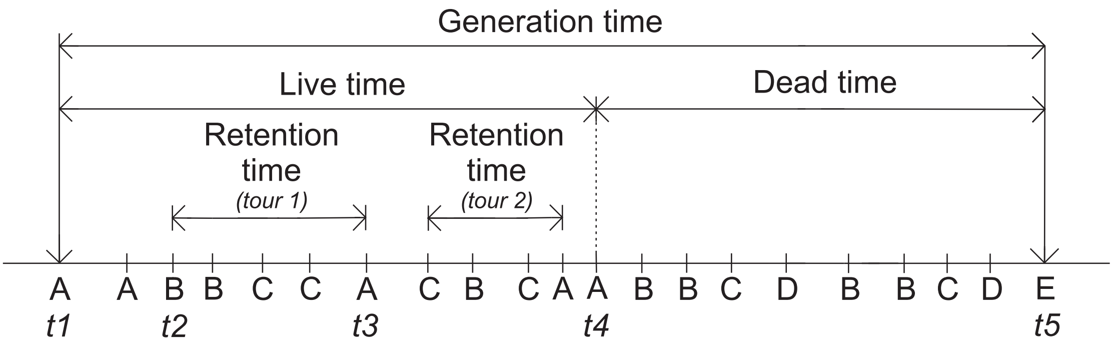


FIGURE 5.3: Generation time of block A (blocks A, B, C, D, and E map to the same set of 4 ways)

Regarding the proposal, the first time the block A is referenced (*t1*), it is stored in an SRAM bank since it is the MRU block. When block B is accessed at time *t2*, a swap operation is triggered and block A is transferred to an eDRAM bank. From this point, the eDRAM capacitors must retain the data until block A is referenced again (thus, transferred to an SRAM bank) at *t3*. This *tour* (since a block is stored in a eDRAM bank until it comes back to an SRAM bank) can happen several times along the live time.

Note that the fact that a block is not accessed for a while does not mean that it is in its dead time, thus eDRAM retention time must be chosen carefully to avoid hurting performance. We measured these times [1] and found that, on average, live and dead times are by 53K and 40K processor cycles, respectively. We also found that a capacitor retaining the charge for more than 38K processor cycles achieves the same performance as a theoretical capacitor with infinite retention time (see Section 5.4.1.1).

Refresh circuitry incurs important energy consumption [52]. As mentioned above, HER caches avoid this circuitry by design so attacking both area and energy. Consequently, capacitors contents may be lost during the dead time of a block. This could lead to incorrect program execution in case that the block contents were dirty. To avoid this situation, the scheme distinguishes between two types of writeback operations: i) writebacks due to replacements, and ii) writebacks due to capacitor discharges. The first type, like in conventional caches, is triggered when a dirty block is selected for replacement. The second type is triggered when checking regularly (*scrubbing*) the state of all the valid blocks located in eDRAM banks. If the valid block is found dirty, a preventive writeback to L2 is triggered. In addition, whether it is dirty or not, the block is invalidated in L1. This prevents accessing to an eDRAM block that has lost its data.

[1] With the machine parameters described in Section 5.4.

The scrub operation can be implemented with a single binary counter [23, 28] for the entire cache, initialized to the retention time divided by the total number of eDRAM blocks in the cache, and guarantees that all eDRAM blocks are checked (i.e., written back if dirty and invalidated) before the retention time expires. The impact of bank contention on performance is minimal because, i) most accesses hit in SRAM banks, and ii) most banks are eDRAM, so when one of them is being scrubbed, the remaining ones can be accessed.

Notice that accessing the dirty bit (which is stored in the tag array) to check if a block must be written back does not block the access to the data array. In fact, the data array is only blocked when a writeback operation is triggered (which happens scarcely).

5.2.2 Design issues: manufacturability and low voltages

As mentioned above, the proposal uses error-free SRAM cells specifically designed to work at low voltages (160 mV) [27] to build the tag array and control bits. The main drawback of these cells is the large area (they are twice as large as conventional cells) they occupy, which makes them non appropriate to implement the entire cache. In the HER cache, we compensate the additional tag array area with the reduction of data array area when using eDRAM banks. Actually, the total cache area is reduced up to 37% compared to the conventional SRAM cache (see Section 5.4.3).

SRAM and eDRAM technologies require different steps at the manufacturing process. To ease manufacturability, each bank is implemented with a single technology. In addition, the design assumes that all banks share a common power supply since both technologies are compatible with current logic processes [5, 30]. In fact some companies [48] manufacture eDRAM using logic technologies, with minimal or no changes in manufacturing processes. Engineers also consider the adoption of capacitor-less DRAM structures (i.e. using the gate capacitance of another transistor to create the cell capacitance). Any of these possible eDRAM implementations can reach the performance we assume for eDRAM. Note that this work focuses on an architecture-level proposal for fault-tolerance that can accommodate any technological alternative meeting the imposed timing and retention constraints.

By design, SRAM cells in the data array can be faulty at low voltages due to manufacturing imperfections; in contrast, eDRAM cells can work correctly at very-low voltages [5]. In this case, a lower voltage can be stored in the cell, so the access latency increases and the retention time decreases. The study of the side effects of these issues on performance and energy is left for future work.

The mentioned assumption enables the management of any number of faulty SRAM cells through the use of an eDRAM way as a backup of the SRAM way (also referred to as replica), provided that data in the replica can be accessed within the worst-case eDRAM cell retention time. There is not a fixed eDRAM way devoted to keep the replica, instead such a way is dynamically chosen at runtime (see Section 5.2.3).

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

5.2.3 Low-power working behavior

The cache controller must carry out different actions at low-power mode depending on the result of the access (i.e., hit or miss), the type of access (read or write) and the bank type (eDRAM and/or SRAM). 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. Table 5.2 summarizes these actions. Unlike high-performance mode, in this mode, each time an eDRAM block is read it must be rewritten.

In low-power mode, a variant of the widely used LRU replacement algorithm is devised. As the SRAM way always contains the MRU block, there is no need to store LRU information for this way. On the other hand, the eDRAM way acting as replica will always have its LRU control bits cleared (similarly to the bits of the MRU block in typical LRU implementations).

5.2.3.1 Read hit in the SRAM way

Figure 5.4 depicts the actions and state diagram of the cache controller to deal with a read hit event in the SRAM way. As in high-performance mode, the data is delivered to the processor as soon as it is read. However, from this point, it is unknown whether the read data is correct or not, since some SRAM bits may fail. Thus, the load instruction is allowed to proceed using a speculative value. Then, the eDRAM replica is read and compared to the SRAM value to solve the speculation.

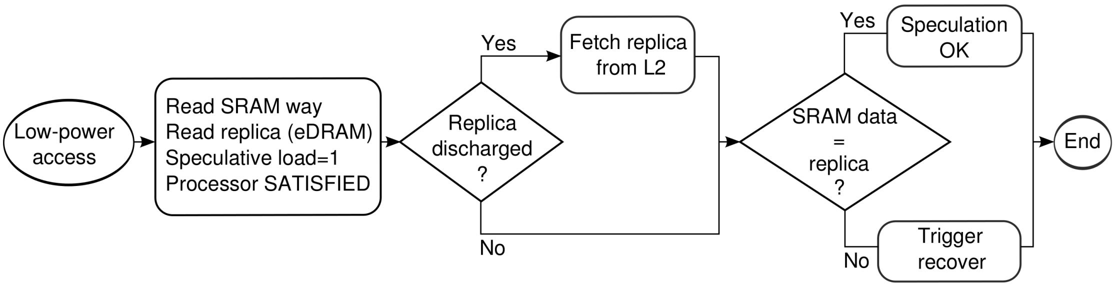


Figure 5.4: State diagram and actions carried out by the cache controller on a read hit in the SRAM way

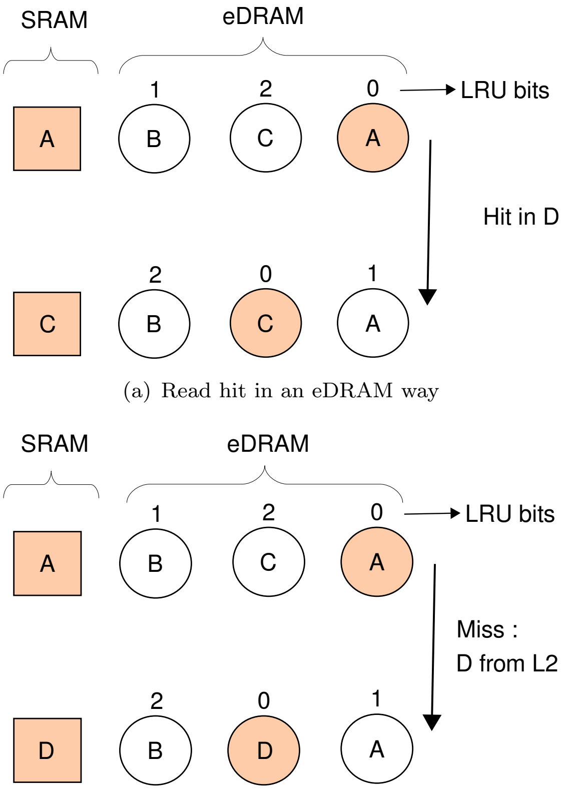
Event	SRAM-faulty bit	Data comparison	Write SRAM	LRU changes?
Read hit (SRAM)	1			
	0	X		
Write hit (SRAM)	1			
	0		X	
Read/Write hit (eDRAM)	1			X
	0		X	X
Read/Write miss	1			X
	0		X	X

Table 5.2: Summary of the actions performed by the cache controller

If the eDRAM replica is not valid, the data block must be fetched from L2. If both SRAM and eDRAM values match, the load becomes non-speculative, that is, the processor is already working with the correct data. On mispeculation, the load and subsequent instructions will be aborted by triggering the conventional microprocessor recovery mechanisms.

Notice that if the read data is faulty due to process variation at a given low-voltage level, then all subsequent load instructions to that address would also incur on mispeculation. Energy consumption due to mispeculation is largely saved thanks to the SRAM-faulty bit. As indicated in Table 5.2, the comparison between the read SRAM data and its replica is only performed when a read hit occurs in an SRAM block having its faulty bit cleared.

This work focuses on line size granularity as implemented in commercial processors [10]. However, this solution could be refined by associating the SRAM-faulty bit to a segment smaller than the line size.



(a) Read hit in an eDRAM way

(b) Cache miss

FIGURE 5.5: Example of accesses in low-power mode

5.2.3.2 Write hit in the SRAM way

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

5.2.3.3 Read/Write hit in an eDRAM way

Upon a read hit in an eDRAM way, if the SRAM block is not faulty, the data must be copied from its line (that will become the new eDRAM replica) to the SRAM way; thus, overwriting its contents. Notice that directly overwriting the SRAM way does not mean any loss of information, since the previous SRAM data remains its eDRAM replica. In case of write hit, both the SRAM way and the replica are updated with the same data.

Figure 5.5(a) shows an example of a read hit in an eDRAM way. As observed, the LRU control bits must be updated accordingly. Data only moves from eDRAM to SRAM, but no bidirectional swap is performed (as done in high-performance mode).

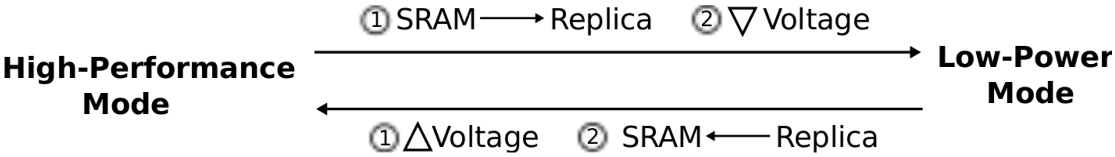


Figure 5.6: Involved actions when changing from high-performance mode to low-power and vice versa (the number in the action establish the order in which they will be performed)

5.2.3.4 Cache miss

On a cache miss, in both read and write operations, the block is fetched from L2 (or a lower level of the memory hierarchy). The incoming block is written both in the SRAM way (MRU line) and in an eDRAM way (i.e. replica), which is provided by the LRU replacement algorithm. As explained above, the previous SRAM contents are not lost. Figure 5.5(b) shows an example where the accessed block D replaces the block C of the SRAM way but the replica of the previously stored block in the SRAM way (block A) remains after handling the miss. Notice that no data movements are required between eDRAM and SRAM ways.

5.2.4 Mode changes

The processor must also provide support to change from high-performance mode to low-power mode and vice versa. Figure 5.6 summarizes the required actions.

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

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

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

	hp		*lp1*		*lp2*	
	1.3V/3GHz - 0% errors		0.7V/1.4GHz - 20% errors		0.5V/500MHz - 50% errors	
Data type	Conv-Cache	HER-Cache	ZfConv-Cache	HER-Cache	ZfConv-Cache	HER-Cache
SRAM	2	2	1	1	1	1
eDRAM	–	4	–	2	–	2

Table 5.3: Access time in cycles for the cache schemes studied (cycle time varies across the operation modes)

5.3 Timing details

This section analyzes the access time (in processor cycles) of the proposed cache for each operation mode, from now on referred to as *hp* (high-performance) and *lp* (low-power) modes, respectively. For *hp* mode, it has been assumed a voltage/frequency pair of 1.3V/3GHz and no bit failures. In addition, to study the impact of the amount of failures on performance and power consumption, two different voltage/frequency pairs have been studied in *lp* mode, referred to as *lp1* and *lp2*, respectively. The former assumes 0.7V/1.4GHz and the latter 0.5V/500MHz similarly to [2]. The probability of failure of the SRAM bits for a 45nm node was calculated as 20% and 50% , respectively. Although some defective bits can be in the same line, the results presented assume the worst case, that is, all defective bits are located in different cache lines.

For comparison purposes, a conventional cache has been modeled with no error failures regardless the operation mode. At high-performance mode, the scheme is referenced as Conv cache and at low-power as ZfConv (zero-failure conventional) cache. All the studied schemes have been modeled with the CACTI 5.3 tool [46, 47] to obtain the access time. The eDRAM cell proposed in [5] was modeled in CACTI to obtain the access time and capacitances, as well as energy and area results. In order to reduce leakage, we used the high-Vt pass transistors simulated by CACTI. This implies that on low-power mode, the stored voltage is lower, which has been taken into account to compute retention times.

Table 5.3 shows the access time in cycles for the cache schemes and cache organizations (32KB-4way and 32KB-8way) evaluated in this work. Since CACTI gives the same access in cycles for both organizations, the timing results only depend on the operation mode (*hp*, *lp1*, and *lp2*) and cache scheme (HER and Conv). For HER caches, the access time also depends on whether the accessed data is located in an SRAM or an eDRAM bank. Note that latency of writing is typically slower than that of reading. However, reading a line involves two actions: the read and a subsequent restore operation. Considering both actions, reading and writing latencies are similar [5]. Finally, although the access time increases with the eDRAM capacitance, this increase is almost negligible for the capacitances analyzed in this work and, since in *lp* modes the cycle time is longer, the access time (quantified in processor cycles) becomes lower.

Memory hierarchy	
L1 data cache	32KB, 4-way, 64 byte-line, 8 banks (2 SRAM and 6 eDRAM) 32KB, 8-way, 64 byte-line, 8 banks (1 SRAM and 7 eDRAM)
L1 data cache hit latency	See Table 5.3
L2 data cache	512KB-8way, 64 byte-line, unlimited banks, hit latency: 10 cycles
Memory access latency	100 cycles

TABLE 5.4: Architectural machine parameters

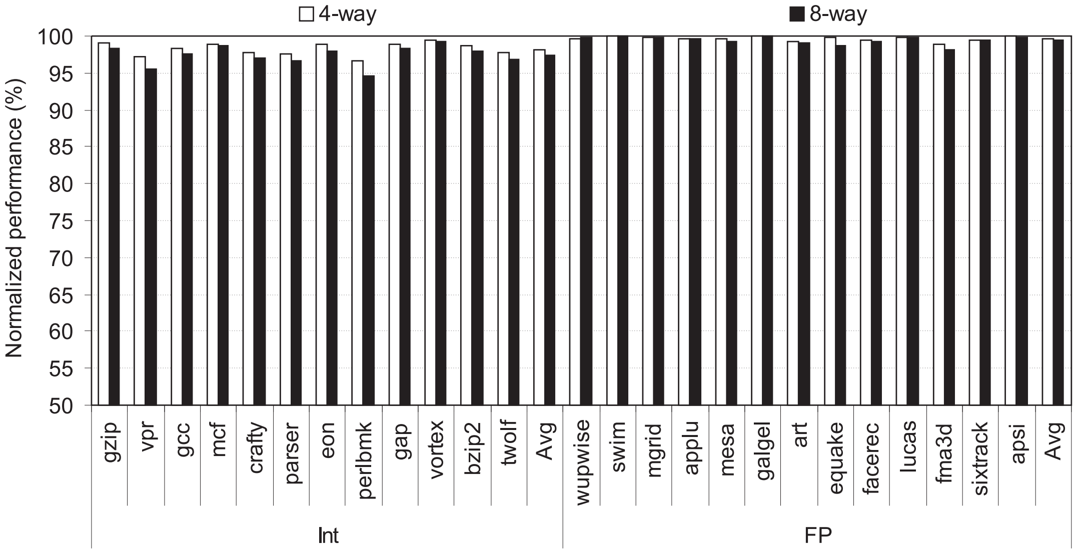


FIGURE 5.7: Normalized performance (%) of the HER cache with respect to the Conv cache in *hp* mode

5.4 Experimental results

Table 5.4 summarizes the architectural parameters. Bank contention (including contention due to swaps and writebacks) has been also modeled. However, accesses to different cache banks may be concurrently performed. Remark that the length of the swap operation is determined by the sum of the latencies of the different accesses to the involved cache banks, and the processor stalls until this operation finishes if it demands data located in these banks.

5.4.1 Performance evaluation

This section first analyzes the minimum required retention time to achieve the same performance as capacitors with a theoretical *infinite* retention time (i.e., capacitors without charge losses). Then, the hit rate of both HER and Conv cache schemes is evaluated.

Cache organization	Operation mode	Retention time (cycles)	Capacitance (fF)	Perf. deg. (%)	
				Int	FP
32KB-4way	*hp*	38K	2	1.88	0.38
	lp1	44K	8	1.75	0.21
	lp2	62K	43	2.54	0.29
32KB-8way	*hp*	59K	4	2.65	0.58
	lp1	74K	14	2.02	0.27
	lp2	88K	62	2.69	0.40

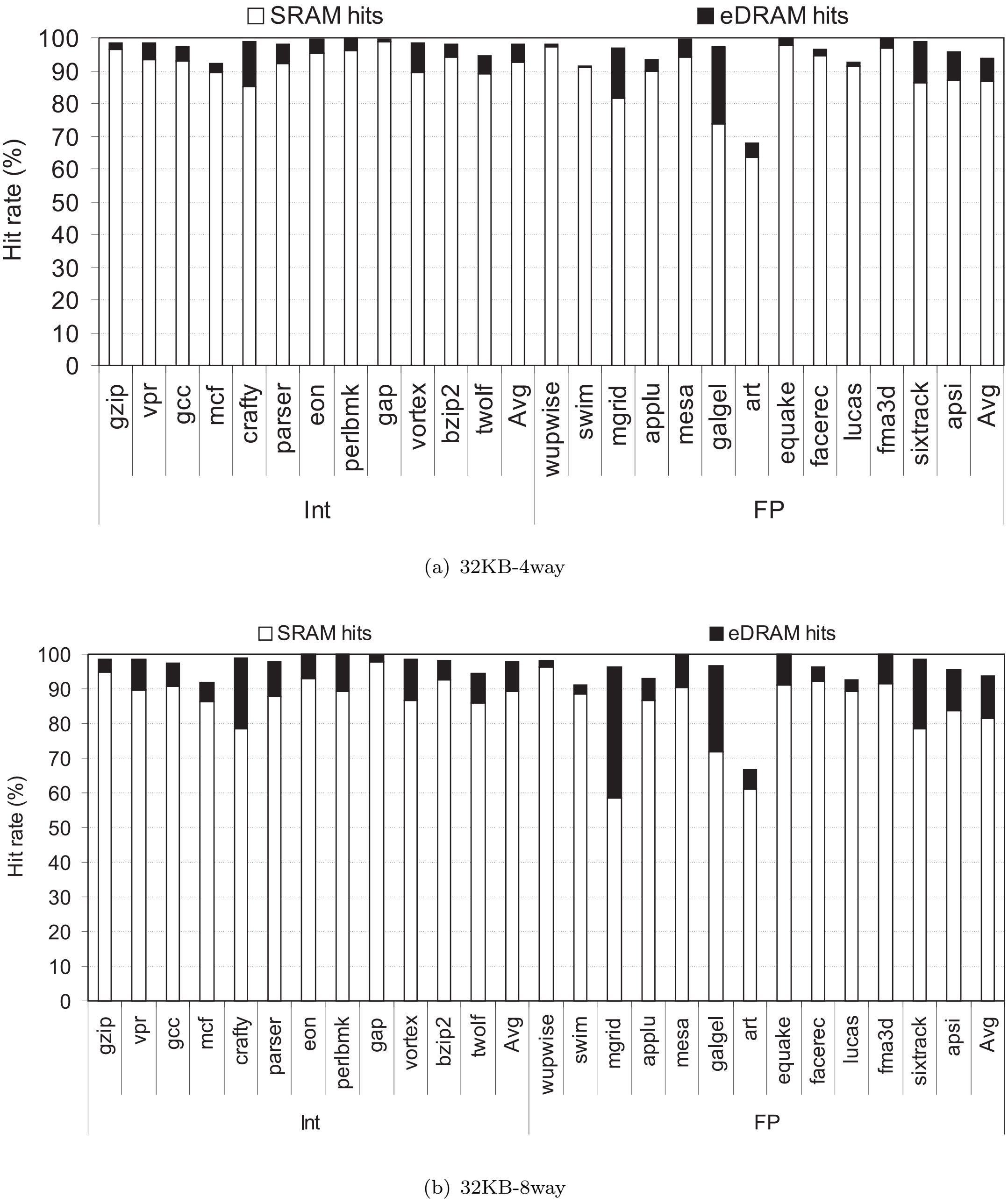
Table 5.5: Retention time (cycles), capacitance (fF), and performance degradation (%) of the HER cache compared to the Conv cache for the different operation modes

5.4.1.1 Performance and retention time analysis

Figure 5.7 shows, for each application, the normalized performance in *hp* mode of a HER cache implemented with an infinite retention time compared to a conventional cache. Notice that the Conv cache scheme imposes an upper-bound since this cache does neither use way-prediction nor is implemented with the *slower* eDRAM cells (see Table 5.3). In other words, these values are the maximum performance that a HER cache with a limited retention time can achieve. As expected, IPC losses increase with the number of ways since the storage capacity associated to the SRAM cells decreases with higher associativities for a given cache size. The IPC losses for integer benchmarks are 1.88% and 2.65% for a 4- and 8-way set-associative HER cache, respectively; and much lower (0.38% and 0.58%) for floating-point benchmarks.

The required retention time, quantified in processor cycles, varies according to the processor speed. Table 5.5 shows, for each operation mode, the minimum retention time that real capacitors should have to allow the cache to match the performance (i.e. harmonic mean of IPCs) of a cache implemented with theoretical capacitors with *infinite* retention time. To estimate the required capacitances with CACTI, the power supply of each operating mode has been also considered. Finally, the average performance degradation with respect to the conventional cache is also presented. Remember that in low-power mode, results are compared to a conventional cache with no failures (ZfConv cache).

Trench capacitors can be used to obtain values up to 30fF [24, 31]. Thus, both *hp* and *lp1* operation modes can be supported with them. However, in *lp2* mode, the 30fF capacitor allows a retention time of *only* 44K cycles, which is smaller than the optimal estimated (62K and 88K). Provided that the counter used to scrub dirty blocks (see Section 5.2.1) is properly initialized, the limitation in the capacitor values will have some performance impact. Experimental results show a rather low performance degradation even in this case (less than 2.7%).



(a) 32KB-4way

(b) 32KB-8way

Figure 5.8: Hit rate (%) split into SRAM and eDRAM hits per benchmark for the HER cache in *hp* mode

As expected, for a given operation mode, the required retention time increases with the number of ways since more blocks are stored in eDRAM banks; thus, these banks must retain their data for longer. On the other hand, the probability of failure increases as the voltage level falls (20% and 50% in *lp1* and *lp2*, respectively), so more accesses need to be done to eDRAM replicas (so increasing the bank contention). This means that, for a given cache, the lower the supply voltage, the longer the required retention time. Consequently, performance degradation is higher in *lp2* mode than in *lp1*. Finally, notice

Bench. type	Hit rate (%)	32KB-4way				32KB-8way			
		Conv	HER			Conv	HER		
		hp	*hp*	*lp1*	*lp2*	*hp*	*hp*	*lp1*	*lp2*
Int	SRAM	97.8	92.6	73.4	42.1	97.9	89.4	71.6	44.6
	eDRAM	–	5.2	4.7	4.7	–	8.5	8.2	8.1
	eDRAM replica	–	–	19.3	50.7	–	–	17.9	45.1
	total	97.8	97.8	97.5	97.5	97.9	97.9	97.8	97.8
FP	SRAM	93.7	86.7	68.8	41.9	93.7	81.6	66.0	41.2
	eDRAM	–	7.0	6.0	6.0	–	12.1	11.9	12.0
	eDRAM replica	–	–	17.9	44.8	–	–	15.7	40.4
	total	93.7	93.7	92.7	92.7	93.7	93.7	93.6	93.6

Table 5.6: Hit rate (%) split into SRAM, eDRAM, and eDRAM replica hits for the cache schemes across the studied operation modes.

that performance in high and low operation modes cannot be directly compared since the processor speed differs.

5.4.1.2 Hit rate evaluation

Figure 5.8 shows the cache hit rate for each application in high-performance mode for a HER cache implemented with the retention times shown in Table 5.5. The hit rate has been split into hits in SRAM banks and hits in eDRAM banks. As expected, on average, the SRAM hit rate decreases while the eDRAM hit rate increases with the number of ways. These results also confirm that, on average, most accesses hit the MRU line (SRAM bank).

Table 5.6 shows the average hit rate of both cache schemes across the studied operation modes. For the low-power modes, hits in the eDRAM replica are also presented. Remark that in *hp* mode the total hit rate is the same as the hit rate of the conventional cache, which confirms that no eDRAM cache line is encountered invalid when it is accessed, that is, the retention times do not yield to performance losses due to capacitor discharges. Notice that the SRAM hit rate in HER caches is much lower in the more defective *lp2* mode than in the *lp1* mode. Finally, the reduction of the effective eDRAM capacity due to the replicas has a minimal impact on the eDRAM hit rate.

5.4.2 Power and energy consumption

Both leakage and dynamic energies have been analyzed. Regarding leakage, Figure 5.9 illustrates the normalized power with respect to the conventional SRAM approach.

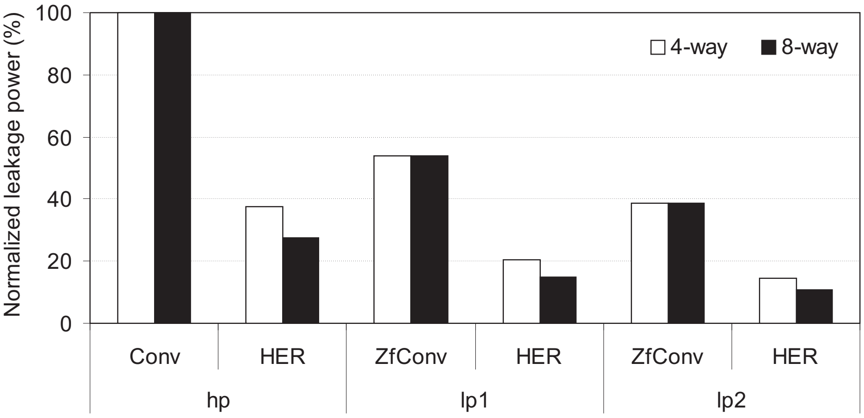


Figure 5.9: Normalized leakage power (%) of the HER cache scheme with respect to the Conv cache

Thanks to the use of eDRAM cells, the HER cache reduces leakage currents by design, regardless the operation mode; however in *lp* mode, benefits are also achieved because of the lower voltage supply. In contrast, power savings of the non-defective ZfConv approach mainly comes from the reduction in the supply voltage. Leakage savings provided by the HER cache can be as high as 73% in *hp* mode and 89% in *lp2* mode in the 8-way cache.

The dynamic energy evaluated is the energy consumed by the data array after running 500M instructions. Figure 5.10 shows the normalized dynamic energy in the HER cache with respect to the conventional cache in *hp* mode. Results also include the dynamic consumption of accessing to the L2 cache because of L1 cache misses and writebacks. It can be seen that energy savings are larger in 8-way set-associative caches. This is mainly due to HER caches access first to the SRAM banks (way-prediction), and SRAM storage capacity is lower in 8-way sets than in 4-way sets. Energy benefits can be, on average, as high as 55% for integer benchmarks.

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

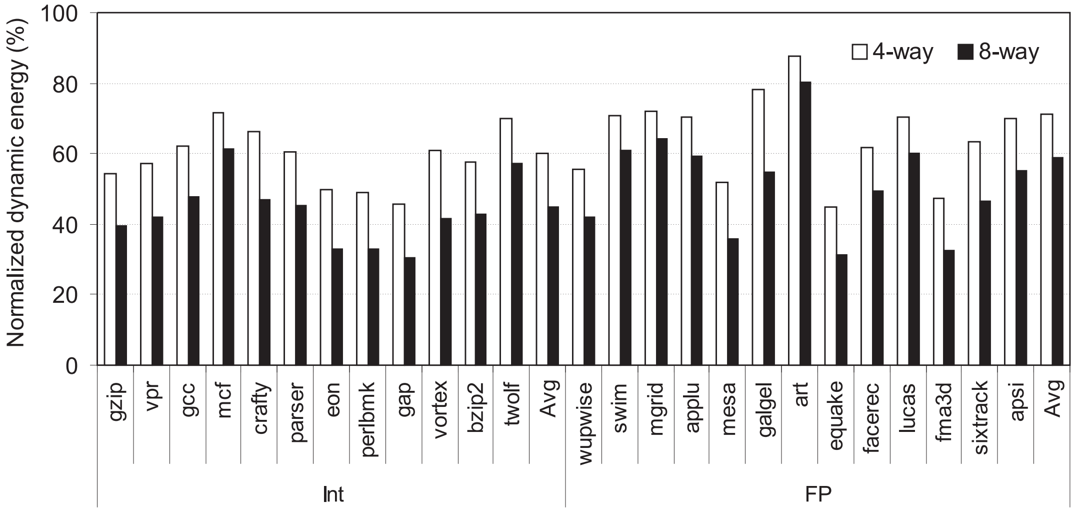
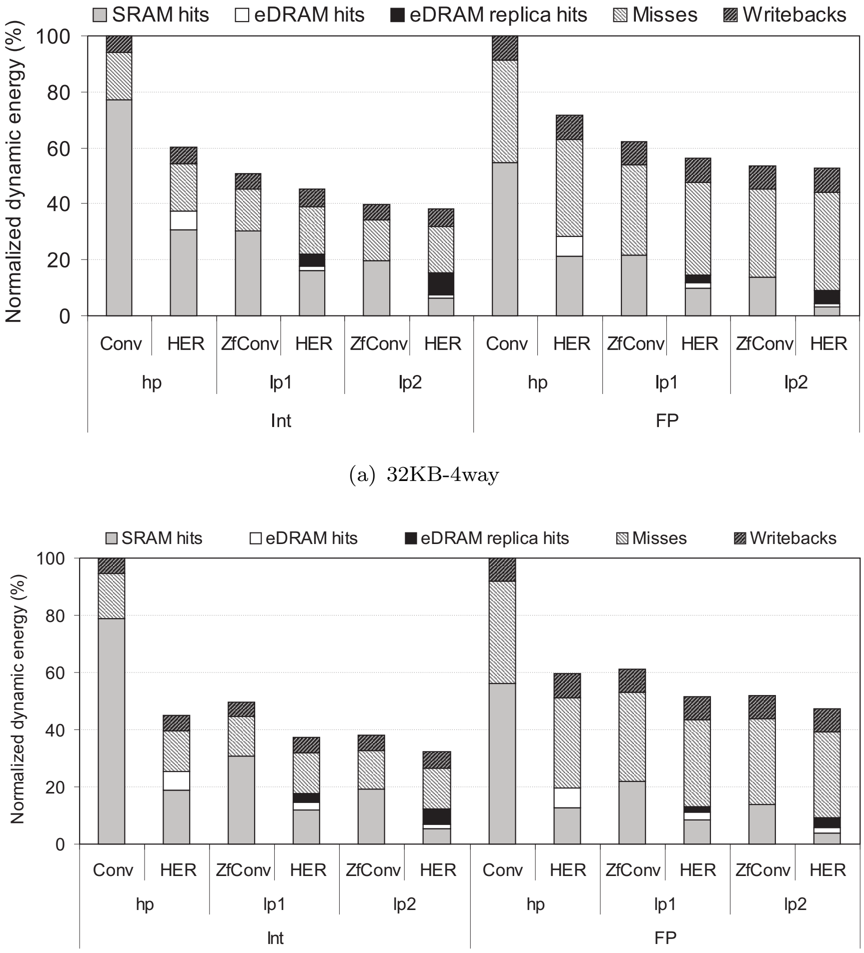


Figure 5.10: Normalized dynamic energy (%) of the HER cache with respect to the Conv cache in *hp* mode

Figure 5.11 shows these values normalized with respect to the conventional cache. Important differences appear in the SRAM hits category mainly because of the proposal implements only one SRAM way. Notice also that the energy required by swaps does not noticeably affects the total energy. In addition, the eDRAM replica category has a minor impact on the total energy, being a bit larger in the more defective *lp2* mode. The misses category slightly varies across the different schemes; the major differences appear in the *lp* modes mainly due to the effective storage capacity is smaller. Finally, remark that the writeback policy of the HER cache has a little impact on the overall energy since we measured the amount of writebacks performed by this policy and found that, on average, the overall number of writebacks increases by 5.3%. An interesting observation is that the HER cache saves dynamic energy compared to the non-defective ZfConv cache even in *lp2* mode where the probability of defective lines is 50%. Energy benefits are higher in integer benchmarks since they present a lower number of cache misses. Compared to the Conv cache, these benefits can be as high as 68% for the 8-way cache.

5.4.3 Area

Basically, dealing with the problem of SRAM faulty cells when working at very low voltages can be broken down in two main categories: *redundancy* [3] and *sacrifying cache capacity* [51]. Both alternatives incur an increase in area.



(a) 32KB-4way

(b) 32KB-8way

Figure 5.11: Normalized dynamic energy categorized with respect to the Conv cache

Unlike previous works, the proposal supports error failures by using less area than conventional caches. This section analyzes the area savings of the HER scheme compared to the conventional SRAM cache scheme. Table 5.7 shows the results for a 45nm technology node, where SRAM and eDRAM cells occupy an area of $0.296\mu m^2$ and $0.062\mu m^2$ [46, 47], respectively. As the results show, the area of the data array can be reduced as much as 46% and 53% for the 32KB-4way and the 32KB-8way cache, respectively.

However, remember that special SRAM cells have been used to implement the tag arrays of the proposal, whose size is twice as large as the size of conventional SRAM cells [27]. Even considering both tag and data arrays, area savings are as large as 31% and 37% for the 32KB-4way cache and 32KB-8way cache, respectively.

	Tag array (mm^2)		Data array (mm^2)		Data array area	Total area
# ways	Conv cache	HER cache	Conv cache	HER cache	savings (%)	savings (%)
4	0.013	0.026	0.113	0.061	46	31
8	0.013	0.026	0.114	0.054	53	37

Table 5.7: Tag array and data array areas (in mm^2) of both Conv cache and HER cache schemes

5.5 Summary

In this chapter, based on the bank-based heterogeneous cache organization presented in Chapter 4, SRAM and eDRAM technologies have been combined to design hybrid caches, namely HER caches, with two different operation modes: high-performance mode for performance, and low-power mode to tackle bit failures due to the manufacturing process variations. An n-way set-associative HER cache implements one cache way with SRAM bank and the remaining ones with eDRAM banks, which allows reducing silicon area of the data array by half.

In high-performance mode, the HER cache works with its whole cache capacity and despite the higher access time of the eDRAM ways, thanks to storing the MRU blocks in the faster SRAM banks, IPC losses for a 32KB-4way are maintained as low as 1.9% compared to a conventional cache.

Previously proposed ECC-based techniques in low performance-mode recover a limited number of errors in SRAM cells in the same line. In contrast, the proposal provides higher fault coverage, even for transient errors, since eDRAM cells are more robust and less sensitive to transient faults. Moreover, HER caches support 100% fault-coverage of errors caused by process variation imperfections, which is a major design concern to be addressed in future technology nodes.

Two different low voltage levels with different probability of failure (20% and 50%) have been analyzed. Experimental results have shown that, for a 8-way cache, leakage savings can be as high as 89% and dynamic energy consumption is reduced by 68%. Moreover, this is achieved by maintaining performance degradation always below 2.7%.

Finally, we can conclude that combining SRAM and eDRAM technologies shows to be a good alternative for implementing memory structures at low voltages in the presence of high failure rates. This approach can be extended to be applied in large L2 or L3 caches, where the benefits that might be achieved are potentially much higher. However, since data localities differ, the design must be revisited.

Chapter 6

Trade-offs between SRAM Failures and Operation Modes

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.

Therefore there is a tradeoff among cache access time, probability of failure (effective cache capacity) and energy. This chapter is aimed at providing some preliminary results addressing these issues in order to help computer architects to devise the best design choice according to storage requirements of the running workloads.

To do so, this chapter characterizes SRAM failures and estimates their probabilities at very-low voltages. Once obtained these failure probabilites, the chapter provides simple techniques to decrease them by reducing the processor frequency. Finally, an evaluation methodology to analyze the impact on energy consumption of error correction approaches is presented. The technique enables architects to indentify the optimal operation mode (i.e., voltage/frequency pair) from an energy point of view.

The rest of this chapter is organized as follows. Section 6.1 describes the different types of SRAM cell failures. Section 6.2 characterizes the different types of SRAM hard errors at very-low voltages and explores frequency-based techniques to reduce failure probabilities. Section 6.3 presents an evaluation methodology to analyze the impact on energy consumption of error correction approaches. Finally, Section 6.4 summarizes the chapter.

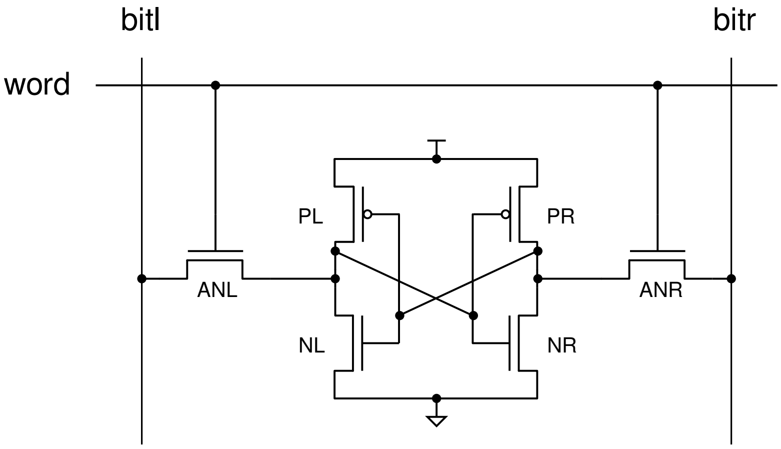


FIGURE 6.1: 6T SRAM cell details

6.1 Background on SRAM cell failures

Manufacturing process produces variations in the transistor parameters mainly due to physical factors caused by processing and masking imperfections [33]. 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.

Inter-die variations affect all the transistors of a given die in the same way (e.g. threshold voltage of all the transistors either increase or reduce). As opposite, *intra-die* variations may affect transistors in the same chip in a different way (e.g. the Vth of some transistors can increase with respect to the nominal one whereas some others can have a Vth lower than the nominal one). In turn, *intra-die* variations can be either systematic or random. *Systematic* are variations depending on the variations of neighboring transistors while *random* variations are independent of the neighboring transistors.

Variations in different device parameters result in large spread in transistor threshold voltage [16]. Among them, the random placement of dopants causes threshold voltage mismatches among transistors that are spatially close to each other. 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 [6, 8, 15], that is, random intra-die variations. These device parameters mismatches severely affect SRAM cells in sub-50nm technologies [1].

Each SRAM cell contains two pairs of transistors forming a logical not. Figure 6.1 shows these two pairs, one formed by PR-NR and the other formed by PL-NL. Any mismatch between devices of a pair degrades the stability of the cell and results in a cell failure when working at voltages lower than the design one. 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: hold failure, read failure, write failure, and access failure. Below we discuss them.

6.1.1 Hold failure

The transistors in each of the mentioned transistor pairs are interconnected by a node. One of the nodes in a cell stores a "1" and the other one stores a "0". The voltage of the node storing "1" is the same as the power supply of the cell. Most current microprocessors implement a low power mode which highly reduces the power supply to save energy. When working at this mode, 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 losing the stored value and producing a hold failure.

6.1.2 Read failure

Before the read is performed, both bitlines (i.e. *bitl* and *bitr*) must be 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 connected 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, since it occurs when the cell is being read.

6.1.3 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 may not be longer enough to decrease the voltage below the trip-point.

[1] The trip-point is the voltage necessary at the gates of the transistors connected to a given node to change the node stored value.

6.1.4 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% the 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 to the discharging speed. If this speed is too slow, the difference needed to excite the sense amplifier is not achieved.

6.2 SRAM cell failure characterization

Power consumption strongly depends on the power supply. More precisely, dynamic and static power grow quadratically and linearly with the power supply, respectively. Thus many research has focused on reducing the power supply to reduce energy. Nevertheless, reducing power supply negatively impacts on the number of errors so hurting the performance. Therefore, process variation introduces a tradeoff between performance and power. This section characterizes the behavior of the SRAM cells, quantified in probability of failure, varying the power supply and taking into account Vth variations.

The probabilities of hard error due to the failures described in Section 6.1 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) [55]. We used the BSIM4 MOSFET model that addresses the MOSFET physical effects into the sub-100nm regime.

Transistor sizes have been chosen according to [32] 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.

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 μ and σ_{VT0} equal to 0 and 14%, respectively, and with 42% maximum Vth deviation [22]. The MonteCarlo simulation method was used to generate 100K samples of cells.

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

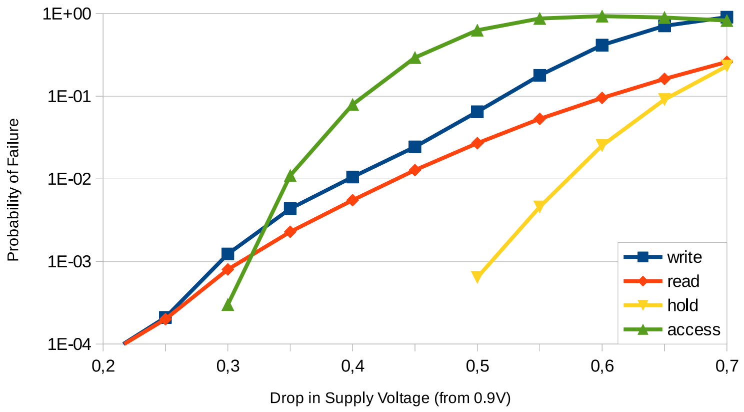


Figure 6.2: Breakdown of SRAM cell failure probabilities for a 32nm technology node reducing the power supply from 0.9V

6.2.1 Impact of the power supply on the failure probabilities

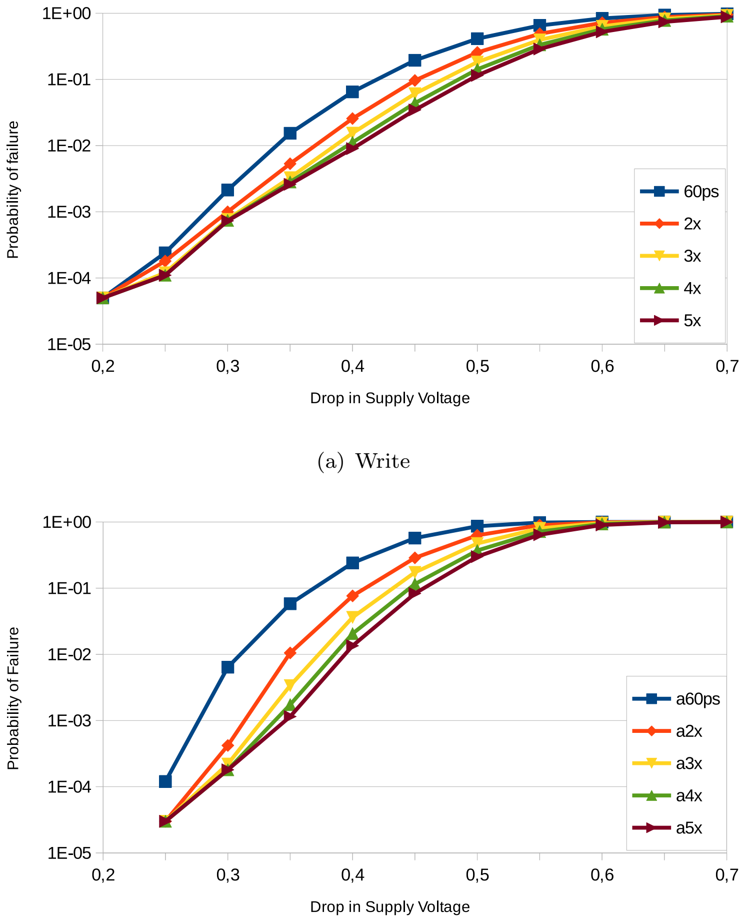
Figure 6.2 shows the probability of failure for the four types of SRAM errors described above as the voltage supply drops. The nominal power supply has been set to 0.9V and drops are drawn in steps of 50mV; that is, the X axis shows the power supply ranging from 0.7V to 0.2V.

Results show that for low-level voltages, the probability of failure is dominated by access and write failures, while in higher voltages write and read failures dominate. Anyway, hold failures are the least likely. This means that the major performance benefits can be achieved by attacking and reducing access and write failures.

As discussed above in Section 6.1, 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. In other words, access failures, as well as write failures, are affected by the WL pulse time. That is, these errors can be reduced by using a larger WL pulse.

6.2.2 Impact of the WL pulse length on write and access failures

This section provides an study of the relation between the WL pulse length and SRAM access and write failures. Access and write failures can be reduced with longer WL pulse lengths. A simple and effective way to increase the length of the WL pulse is to reduce the processor frequency. At first sight, it may seem that a slower frequency may increase execution time, but indeed, it may not if the reduction of the number of failures significantly avoids the performance penalties due to using error recovery techniques.



(a) Write

(b) Access

Figure 6.3: Probability of write and access failures varying the WL pulse

To estimate the pulse width the cell was tested using transistors with no variations and a power supply of 0.9V. The adequate pulse width is that that meets the specifications, that is, a WL pulse that gets a voltage difference between bitlines higher than 10% of Vdd. Attending to the results, we used a 60ps as the nominal pulse width.

Nevertheless, as discussed above, write and access failures are produced because, due to transistor variations, there is not enough time to carry out the operation. This section analyzes how longer pulses can help reducing write and access failures. To this end, we enlarged the WL pulse in a 2-, 3-, 4-, and 5x factor.

Figure 6.3 shows the probability of access and write failures varying the WL pulse. Results are shown for WL pulses as large as 2-, 3-, 4- and 5 times the original WL pulse length. As observed, differences among pulse lengths curves rise with low voltage drops in both types of failures (left side of Figure 6.3). Regarding access failures, in it can be observed (Figure 6.3(b)) that curves for the different pulse lengths begin to converge

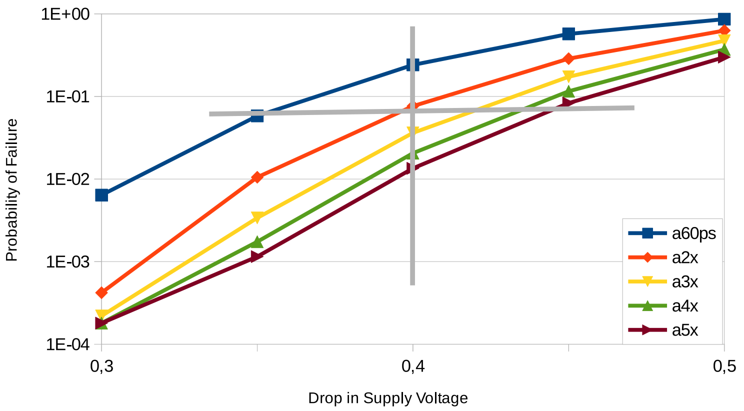


FIGURE 6.4: Probability of access failure

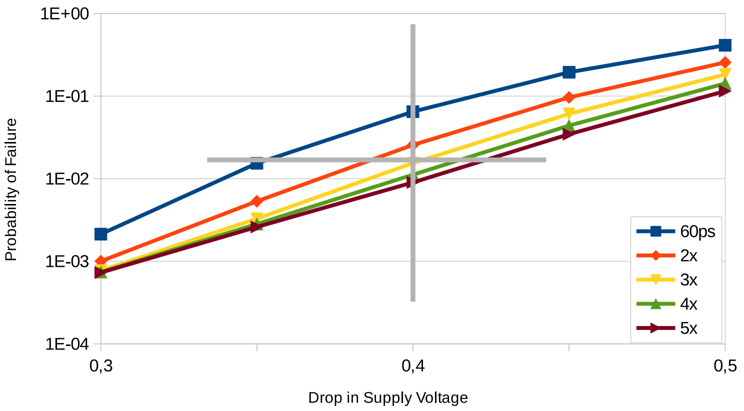


FIGURE 6.5: Probability of write failure

with a 0.6V power-supply drop. That is, from such a drop no improvement can be done. On the other hand, in Figure 6.3(a), convergence appears with a 0.65V power-supply drop.

Computer architects need insights on access time and probability of failures since depending on the workload behavior it could be a better choice a larger access time but with a lower number of failures than a shorter access time with a larger amount of failures or vice versa. To this end, Figures 6.4 and 6.5 present a zoom of the most interesting parts of Figure 6.3. Results on these figures can be analyzed in two main ways. By drawing a vertical line crossing the different curves, it can be estimated how much the pulse length can improve the probability of failure. On the other hand, by drawing an horizontal line, the curve traversing the crossing point identifies the pulse width required to work at a given voltage for a given probability of failure.

For instance, the vertical line drawn in the Figure 6.4 indicates that, for 0.4V, the probability of access failures can be reduced around 68% by doubling pulse length, and as much as 94% by enlarging 5 times the nominal pulse.

Although the probability of failure increases with the voltage drop, the plotted horizontal line shows the WL pulse length that would be required in order to keep the same probability of failure. For instance, for a 0.1V voltage drop, the required pulse to keep the same probability of failure as the original voltage should be 5 times the original one.

Regarding write failures, Figure 6.5 shows that enlarging the WL pulse provides an effect close to the observed in access failures. The vertical line shows that the probability of write failure can be reduced about 60% by doubling the WL pulse length, and up to 86% by enlarging it 5 times. On the other hand, the horizontal line shows that we must triple the WL pulse length to keep the same probability of write failure if we drop 0.05V the power supply.

6.3 Optimal voltage/frequency pairs in fault-tolerant caches

Existing SRAM fault-tolerant proposals incur on a significant performance penalty since they increase access latency and/or 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 as the HER cache.

This section presents a methodology to evaluate the impact on energy consumption of error detection and correction proposals. To focus the research, experimental evaluation concentrates HER cache organization explained in Chapter 5. We analyze a wide range of voltage/frequency pairs to find out the optimal pair for this fault-tolerant cache in terms of energy. The devised methodology can be straightforwardly adapted to be used in any fault-tolerant technique, specially in those suffering significant performance losses.

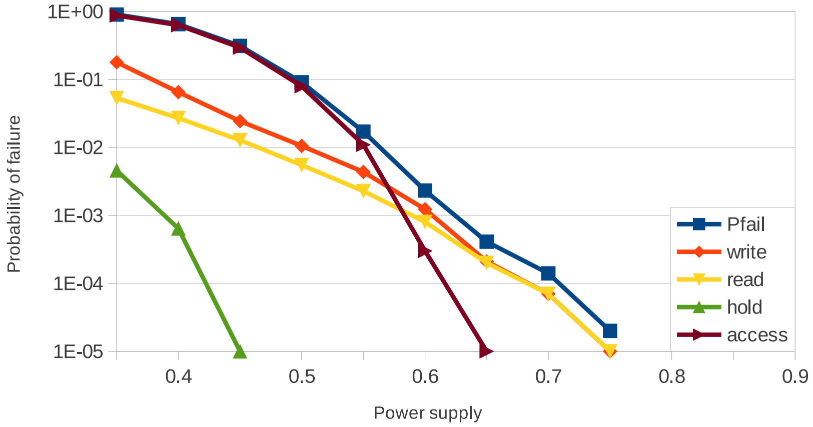


FIGURE 6.6: Overall SRAM cell failure probabilities for a 32nm technology node

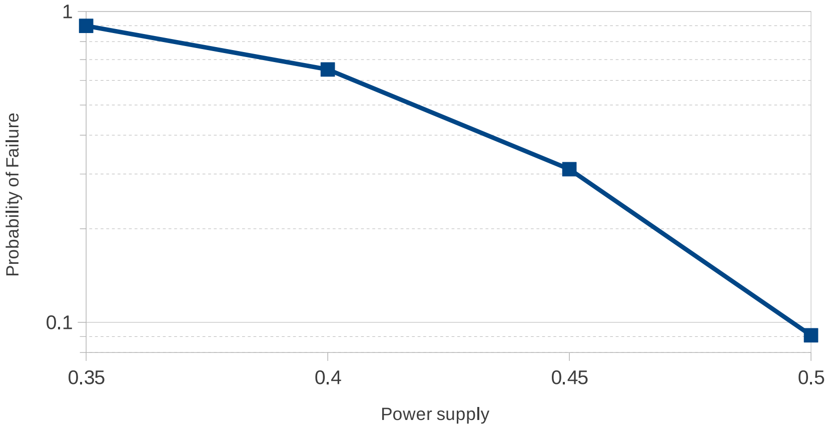


FIGURE 6.7: Low-power range

6.3.1 Operation modes

The methodology presented in this section is backed up by the deep analysis of the failure probability provided by Section 6.2. Figure 6.6 illustrates the overall failure probability (*Pfail*) when varying the power supply between 0.35V to 0.9V. Notice that fault-free voltages are those higher than 0.75V. This work assumes that the *high-performance (hp)* mode has the associated typical supply voltage of 0.9V.

Figure 6.7 focuses on the range from 0.5V to 0.35V, which covers a significant range of probability of failure (from 9% to 90%). Taking into acount these results, we consider four different *low-power (lp)* modes varying the voltage in steps of 0.05V. Table 6.1 summarizes the *hp* and *lp* modes with their associated voltage, processor frequency, and probability of failure. The frequency values are similar to those considered in [12].

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

Table 6.1: Operation modes with their voltage, frequency, and SRAM probability of failure

Microprocessor core	
Issue policy	Out of order
Branch predictor type	Hybrid gShare/Bimodal: gShare has 14-bit global history plus 16K 2-bit counters 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 *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

Table 6.2: Architectural machine parameters

6.3.2 Experimental evaluation

Table 6.2 summarizes the architectural parameters. 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

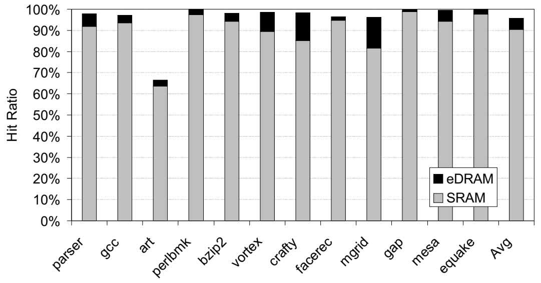


Figure 6.8: SRAM and eDRAM hit ratio per benchmark for the HER cache in *hp* mode

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 6.3: Retention time (in 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

matches that given for the SRAM banks of HER caches in *hp* mode.

6.3.2.1 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 6.8 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 6.3) is large enough to avoid accessing data that have been previously invalidated by the scrub operation. (see Section 5.2.1).

As the voltage is reduced, the number of SRAM errors rises, which impacts on the hit ratio. Table 6.4 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

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

Table 6.4: Hit ratio (%) of the HER cache in the analyzed operation modes

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 6.3 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.

6.3.2.2 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 6.9 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

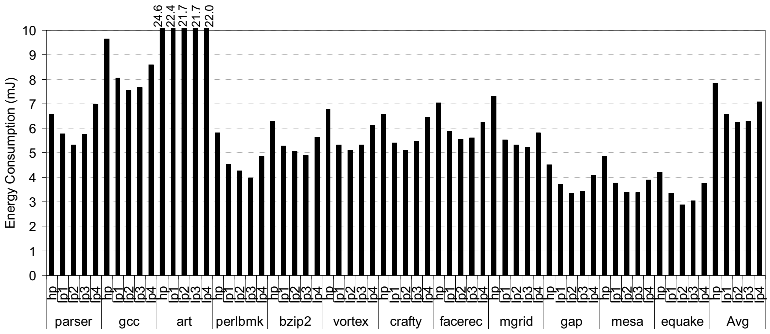


FIGURE 6.9: Energy consumption (in mJ) of the HER cache for each benchmark

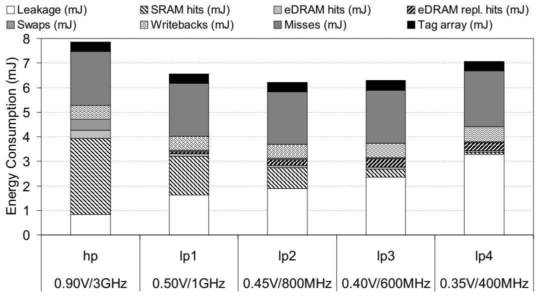


FIGURE 6.10: *Categorized* average energy consumption (in mJ)

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 6.3) 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 6.10 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 6.3). On the other hand, dynamic energy decreases with lower voltages 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%.

6.4 Summary

In this chapter we have characterized the four types of failures that can be produced in SRAM cells (read, write, access and hold) analyzing how threshold voltage variation affects the probability of failure ranging the power supply voltage from 0.9V to 0.2V, and for 32nm feature size. Reducing the voltage, the power consumption can be highly reduced but at expense of suffering a higher amount of failures and reduced effective cache capacity. Therefore, there is a tradeoff between performance and power.

Simulation results showed that access failure is the predominant type of failure. A deeper analysis has been conducted to explore the impact of those types of failures that are dependent of the WL pulse length, that is, access and write failures. The main goal behind this work is to help computer architects to take architectural decisions to deal with the aforementioned tradeoff. In this context it would be worth to know the relationship between probability of failure and pulse length. Results have shown that the access failure probability can be reduced up to 68% and write failure probability by 60% when doubling the WL pulse length. In addition enlarging the WL pulse length also allows to keep the same probability of failure when the power supply is reduced.

Finally, this chapter 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.

Chapter 7

Conclusions

This chapter draws some concluding remarks about the work developed in this thesis. First, we summarize the main contributions, and then, we provide the major directions planned as for future work. Finally, a list of publications related with the work developed in this thesis is presented.

In summary, this work has proposed to combine SRAM and eDRAM technologies to deal with area, power consumption, performance, and SRAM failures in L1 data caches. First, the macrocell was presented, which combines SRAM and eDRAM technologies to implement a L1 cache architecture that reduce area and power while keeping the performance. Then, the HER (*Hard Error Recover*) cache was proposed, which mingles the mentioned RAM technologies to deal with very low power issues.

7.1 Contributions

Memory cell design for cache memories is a major concern in current microprocessors mainly due to its impact on energy consumption, area, access time, and errors. The macrocell-based cache (M-Cache) has been presented in Chapter 4, which combines SRAM and eDRAM technologies. It has been shown as an efficient device to implement cache memories, since its design deals with the mentioned issues. The use of eDRAM cells implies that the storage in these cells can lose data if the information stored is not refreshed. In order to avoid the refresh operation, a detailed analysis has identified the optimal cell capacitance that exhibits the lowest energy consumption and avoids performance losses.

In Chapter 5, SRAM and eDRAM cells have been also combined in a new L1 data cache architecture, HER (*Hard Error Recover*) that is able to recover the 100% error failures produced in SRAM cells when the processor is working in low power modes, that is, at very low voltages. In this chapter the working behaviour of the HER cache is described. It has two operational modes, high-performance mode for performance, and low-power mode that manages bit failures produced in some cells due to the manufacturing process variations at low level voltages. Performance, power, and area evaluation of the HER cache has been analysed concluding that the combination of SRAM and eDRAM technologies is a good alternative for implementing memory structures in the presence of high failure rates due to low power modes.

Chapter 6 highlighted two main contributions. First, a characterization of the four types of SRAM failures have been done (read, write, access, and hold failures), analyzing how parameter variations produced in the manufacturing process affect the probability of failure in SRAM cells. The probability of failure for each voltage from 0.7V to 0.2V in steps of 0.5V is obtained as result of this analysis. In addition, a deeper analysis on time dependent failures (access and write) also provides some clues on how to reduce the probability of failure by extending the word line pulse width. The second contribution of this chapter is an evaluation methodology to identify the optimal voltage/frequency pair in fault-tolerant caches. Voltage scaling in nowadays processors is a technique used to reduce energy consumption. However, this study shows that the optimal voltage not always is the lowest voltage. The evaluation methodology takes into account performance and energy in the analysis, and results show that 0.45V/800Mhz is better than 0.40V/600Mhz and even better than 0.35V/400Mhz.

7.2 Future Directions

Three main directions are planned as for future work, i) analysis of the probability of failure in eDRAM cells, ii) study the use of HER caches in L2 and LLC caches, and iii) analysis of the viability of alternative memory technologies (e.g. PRAM or MRAM).

When the cells are subjected to a low voltage power supply, below the minimum voltage (Vcc_{min}) that ensures the proper functionality of the cell, some cells may not properly work. As explained in Chapter 5, low voltages can affect in a different way to SRAM and eDRAM cells. At low voltages, SRAM technology produces unreliable cells, while eDRAM technology produces variations in the retention time of the cells. The HER cache presented in this thesis is an example of fault tolerant cache that manages SRAM failing cells at very low voltage levels. An interesting extension of this work would be the analysis of eDRAM effects under low voltages in the devised hybrid architectures.

Other work we plan to do is to adapt the proposed cache designs to caches placed at lower levels of the cache hierarchy, which are much larger and present less data locality.

Recently, new RAM technologies have been developed by numerous memory manufacturers. Among these technologies we would like to mention the Phase Change RAM (PRAM) and the Magnetic RAM (MRAM) technologies. Common features of these emerging technologies are that they present high density (PRAM density is higher than DRAM), low leakage (both new technologies present lower current leakages than DRAM technology), and no refresh is needed. The main drawback of these technologies is that their access times are slower than DRAM technologies. Therefore, a deeper analysis of performance, power, and area is required to check the viability of these technologies.

7.3 Publications

The main results of the work developed in this manuscript have been published in different international journals and conferences, as well as some domestic conferences. Below, the list of papers is presented broken down in journals, international conferences, and domestic conferences.

Journals:

- A. Valero, J. Sahuquillo, V. Lorente, S. Petit, P. López, a nd J. Duato. 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* (TVLSI), volume 20, issue 6, pages 1108-1117, 2012.

International Conferences:

- 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 *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO), pages 213-221, New York, NY, USA, 2009. This publication received a HiPEAC Paper Award.
- V. Lorente and J. Sahuquillo. Effects of Process Variation on the Access Time in SRAM cells. In *Lecture Notes in Computer Science –Europar-2012: Parallel Processing Workshops–* (LNCS), Rhodes Islands, Greece, 2012.
- 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* (DATE), pages 83-88, Grenoble, France, 2013.
- V. Lorente, A. Valero, and R. Canal. Enhancing Performance and Energy Consumption of HER Caches by Adding Associativity. In *Lecture Notes in Computer Science –Europar-2013: Parallel Processing Workshops–* (LNCS), Aachen, Germany, 2013.
- V. Lorente, A. Valero, S. Petit, P. Foglia, and J. Sahuquillo. Analyzing the Optimal Voltage/Frequency Pair in Fault-Tolerant Caches. In *Proceedings of the IEEE International Conference on High Performance Computing and Communications* (HPCC), pages 19-26, Paris, France, 2014.

Domestic Conferences:

- A. Valero, V. Lorente, J. Sahuquillo, S. Petit, P. López, J. Duato. Memoria dinámica en caches de datos de primer nivel sin necesidad de refresco. In *Actas de las XX Jornadas de Paralelismo* (JP), pages 265-270, A Coruña, Spain, 2009.

V. Lorente, A. Valero, J. Sahuquillo, S. Petit, P. López, J. Duato. Cache Híbrida de Primer Nivel Tolerante a Fallos. In *Actas de las XXV Jornadas de Paralelismo* (JP), pages 353-359, Valladolid, Spain, 2014.

References

[1] 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*, 40(9):1804–1814, 2005.

[2] Alaa R. Alameldeen, Zeshan Chishti, Chris Wilkerson, Wei Wu, and Shih-Lien Lu. Adaptive Cache Design to Enable Reliable Low-Voltage Operation. *IEEE Transactions on Computers*, 60:50–63, 2011.

[3] A. Ansari, S. Gupta, Shuguang Feng, and S. Mahlke. Maximizing Spare Utilization by Virtually Reorganizing Faulty Cache Lines. *IEEE Transactions on Computers*, 60(1):35–49, 2011.

[4] 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*, 22(1):185–190, 2014.

[5] J. Barth, W. R. Reohr, P. Parries, G. Fredeman, J. Golz, S. E. Schuster, R. E. Matick, H. Hunter, C. C. Tanner, J. Harig, Kim Hoki, B. A. Khan, J. Griesemer, R. P. Havreluk, K. Yanagisawa, T. Kirihata, and S. S. Iyer. A 500 MHz Random Cycle, 1.5 ns Latency, SOI Embedded DRAM Macro Featuring a Three-Transistor Micro Sense Amplifier. *IEEE Journal of Solid-State Circuits*, 43(1):86–95, 2008.

[6] A. J. Bhavnagarwala, Xinghai Tang, and J. D. Meindl. The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability. *IEEE Journal of Solid-State Circuits*, 36(4):658–665, 2001.

[7] Doug Burger and Todd M Austin. The simplescalar tool set, version 2.0. *ACM SIGARCH Computer Architecture News*, 25(3):13–25, 1997.

[8] D. Burnett, K. Erington, C. Subramanian, and K. Baker. Implications of Fundamental Threshold Voltage Variations for High-Density SRAM and Logic Circuits. In *Symposium on VLSI Technology. Digest of Technical Papers*, pages 15–16, 1994.

[9] B. Calder, D. Grunwald, and J. Emer. Predictive Sequential Associative Cache. In *Proceedings of the 2nd International Symposium on High-Performance Computer Architecture*, pages 244–253, 1996.

[10] J. Chang, Ming Huang, J. Shoemaker, J. Benoit, Szu-Liang Chen, Wei Chen, Siufu Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava. The 65-nm 16-MB Shared On-Die L3 Cache for the Dual-Core Intel Xeon Processor 7100 Series. *IEEE Journal of Solid-State Circuits*, 42(4):846–852, 2007.

[11] 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*, pages 128–129, 2005.

[12] Pedro Chaparro. Thermal Aware Microarchitectures. *Ph.D Thesis, Universitat Politècnica de Catalunya*, 2008.

[13] Ronald G. Dreslinski, Gregory K. Chen, Trevor Mudge, David Blaauw, Dennis Sylvester, and Krisztian Flautner. Reconfigurable Energy Efficient Near Threshold Cache Architectures. In *Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture*, pages 459–470, 2008.

[14] K. Flautner, Nam Sung Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy Caches: Simple Techniques for Reducing Leakage Power. *Proceedings of the 29th Annual International Symposium on Computer Architecture*, pages 148–157, 2002.

[15] R. Heald and P. Wang. Variability in Sub-100nm SRAM Designs. In *IEEE/ACM International Conference on Computer Aided Design*, pages 347–352, 2004.

[16] D.E. Hocevar, P.F. Cox, and P. Yang. Parametric yield optimization for mos circuit blocks. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 7(6):645 –658, jun 1988.

[17] Synopsys HSPICE. Inc., dec. 2010. *Version E-2010.12.*

[18] Zhigang Hu, Philo Juang, Phil Diodato, Stefanos Kaxiras, Kevin Skadron, Margaret Martonosi, and Douglas W. Clark. Managing Leakage for Transient Data: Decay and Quasi-Static 4T Memory Cells. *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*, pages 52–55, 2002.

[19] Koji Inoue, Tohru Ishihara, and Kazuaki Murakami. Way-Predicting Set-Associative Cache for High Performance and Low Energy Consumption. In *Proceedings of the 2nd International Symposium on High-Performance Computer Architecture*, pages 273–275, 1999.

[20] ITRS. *Semiconductor Industries Association, " International Technology Roadmap for Semiconductors", 2007, available online at http://www.itrs.net/.*, 2007.

[21] ITRS. *Semiconductor Industries Association, "International Technology Roadmap for Semiconductors", Process Integration, Devices, and Structures, 2009*, 2009.

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

[23] S. Kaxiras, Z. Hu, and M. Martonosi. Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power. In *Proceedings of the 28th Annual International Symposium on Computer Architecture*, pages 240–251, 2001.

[24] Brent Keeth, R. Jacob Baker, Brian Johnson, and Feng Lin. *DRAM Circuit Design. Fundamental and High-Speed Topics.* John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.

[25] Jangwoo Kim, Mark Mccartney, Ken Mai, and Babak Falsafi. Modeling sram failure rates to enable fast, dense, low-power caches. IEEE Workshop on Silicon Errors in Logic, March 2009.

[26] Toshiaki Kirihata, Paul Parries, David R. Hanson, Hoki Kim, John Golz, Gregory Fredeman, Raj Rajeevakumar, John Griesemer, Norman Robson, Alberto Cestero, Babar A. Khan, Geng Wang, Matt Wordeman, and Subramanian S. Iyer. An 800-MHz Embedded DRAM With a Concurrent Refresh Mode. *IEEE Journal of Solid-State Circuits*, 40(6):1377–1387, 2005.

[27] J. P. Kulkarni, Keejong Kim, and K. Roy. A 160 mV, Fully Differential, Robust Schmitt Trigger Based Sub-threshold SRAM. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 171–176, 2007.

[28] Xiaoyao Liang, R. Canal, Gu-Yeon Wei, and D. Brooks. Process Variation Tolerant 3T1D-Based Cache Architectures. In *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 15–26, 2007.

[29] Vicente Lorente, Alejandro Valero, Julio Sahuquillo, Salvador Petit, Ramon Canal, Pedro López, and José 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*, pages 83–88, 2013.

[30] Richard E. Matick and Stanley E. Schuster. Logic-based eDRAM: Origins and rationale for use. *IBM Journal of Research and Development*, 49(1):145–165, 2005.

[31] W. Mueller, G. Aichmayr, W. Bergner, E. Erben, T. Hecht, C. Kapteyn, A. Kersch, S. Kudelka, F. Lau, J. Luetzen, A. Orth, J. Nuetzel, T. Schloesser, A. Scholz, U. Schroeder, A. Sieck, A. Spitzer, M. Strasser, P.-F. Wang, S. Wege, and R. Weis. Challenges for the DRAM Cell Scaling to 40nm. *IEEE International Electron Devices Meeting*, pages 4 pp.–339, 2005.

[32] 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*, 24(12):1859–1880, 2005.

[33] S. R. Nassif. Modeling and Analysis of Manufacturing Variations. In *IEEE Conference on Custom Integrated Circuits*, pages 223–228, 2001.

[34] Paolo Nenzi and Holger Vogt. Ngspice users manual version 23, 2011.

[35] Shuou Nomura, Matthew D. Sinclair, Chen-Han Ho, Venkatraman Govindaraju, Marc de Kruijf, and Karthikeyan Sankaralingam. Sampling + DMR: Practical and Low-overhead Permanent Fault Detection. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, pages 201–212, 2011.

[36] S. Paul, Fang Cai, Xinmiao Zhang, and S. Bhunia. Reliability-Driven ECC Allocation for Multiple Bit Error Resilience in Processor Cache. *IEEE Transactions on Computers*, 60(1):20–34, 2011.

[37] 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*, pages 371–377, 2005.

[38] 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. *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, pages 90–95, 2000.

[39] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

[40] Stanley E Schuster. Multiple Word/Bit Line Redundancy for Semiconductor Memories. *IEEE Journal of Solid-State Circuits*, 13(5):698–703, 1978.

[41] B. Sinharoy, R. Kalla, W. J. Strake, H.Q. Le, R. Cargnoni, J. A. Van 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*, 55(3):1–29, 2011.

[42] B. Sinharoy, R N. Kalla, J M. Tendler, R J. Eickemeyer, and J B. Joyner. POWER5 System Microarchitecture. *IBM Journal of Research and Development*, 49(4/5):505–521, 2005.

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

[44] J. Stuecheli. POWER8. *Hot Chips*, 2013.

[45] J M. Tendler, J S. Dodson, J S. Fields, H. Le, and B. Sinharoy. POWER4 System Microarchitecture. *IBM Journal of Research and Development*, 46(1):5–25, 2002.

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

[47] Shyamkumar Thoziyoor, Jung Ho Ahn, Matteo Monchiero, Jay B. Brockman, and Norman 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*, pages 51–62, 2008.

[48] uniramtech. *http://www.uniramtech.com/embedded_dram.php.*

[49] Alejandro Valero, Julio Sahuquillo, Salvador Petit, Vicente Lorente, Ramon Canal, Pedro López, and José Duato. An Hybrid eDRAM/SRAM Macrocell to Implement First-Level Data Caches. In *Proceedings of the 42th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 213–221, 2009.

[50] N H E. Weste, D. Harris, and A. Banerjee. *CMOS VLSI Design: A Circuits and Systems Perspective.* Pearson/Addison-Wesley, 2005.

[51] C. Wilkerson, Hongliang Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and Shih-Lien Lu. Trading off Cache Capacity for Reliability to Enable Low Voltage Operation. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, pages 203–214, 2008.

[52] Chris Wilkerson, Alaa R. Alameldeen, Zeshan Chishti, Wei Wu, Dinesh Somasekhar, and Shih-Lien Lu. Reducing Cache Power with Low-Cost, Multi-bit Error-Correcting Codes. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, pages 83–93, 2010.

[53] Xiaoxia Wu, Jian Li, Lixin Zhang, Evan Speight, Ram Rajamony, and Yuan Xie. Hybrid Cache Architecture with Disparate Memory Technologies. *Proceedings of the 36th Annual International Symposium on Computer Architecture*, pages 34–45, 2009.

[54] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. Hotleakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. *University of Virginia Department of Computer Science, Technical Report*, 2003.

[55] Wei Zhao and Yu Cao. Predictive Technology Model for Nano-CMOS Design Exploration. *Journal on Emerging Technologies in Computing Systems*, 3(1):1–17, 2007.