Circuit Designs for Increasing Reliability and Reducing Energy

Azamolsadat Seyedi

Department of Computer Architecture

Universitat Politècnica de Catalunya

A dissertation submitted in fulfillment of

the requirements for the degree of

Doctor of Philosophy / Doctor per la UPC

October 2015

Advisors: Osman Unsal, Adrian Cristal

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de calificación de tesis doctoral	Curso académico:

Nombre y apellidos

Programa de doctorado

Unidad estructural responsable del programa

Resolución del Tribunal

Reunido el Tribunal designado a tal efecto, el doctorando / la doctoranda expone el tema de la su tesis doctoral titulada _____

_____.

Acabada la lectura y después de dar respuesta a las cuestiones formuladas por los miembros titulares del tribunal, éste otorga la calificación:

☐ NO APTO ☐ APROBADO ☐ NOTABLE ☐ SOBRESALIENTE

(Nombre, apellidos y firma) Presidente/a	(Nombre, apellidos y firma) Secretario/a	
(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal

_____, _____ de _____ de _____

El resultado del escrutinio de los votos emitidos por los miembros titulares del tribunal, efectuado por la Escuela de Doctorado, a instancia de la Comisión de Doctorado de la UPC, otorga la MENCIÓN CUM LAUDE:

☐ SÍ ☐ NO

(Nombre, apellidos y firma) Presidente de la Comisión Permanente de la Escuela de Doctorado	(Nombre, apellidos y firma) Secretario de la Comisión Permanente de la Escuela de Doctorado

Barcelona a _____ de _____ de _____

ABSTRACT

Computing technology has witnessed an inimitable progress in the last decades which is the result of CMOS technology scaling commensurate with Moore's law. Transistor feature sizes have shrunk to half at each generation, and consequently the number of transistors per chip has doubled each two years. However, power-density problems and the difficulty of eking out more performance from complex out of order single core architectures forced the processor manufacturers to introduce chip multiprocessors (CMP) as a solution. Each processor core in these CMPs was relatively simpler, and the increased number of cores provided increased total performance with decreased power-density. However, the same problems of energy-efficiency wall and performance wall have resurfaced with further scaling; exacerbate by the problem of reliability. This motivates researchers to find effective solutions on a wide variety of aspects such as architecture and circuit levels to mitigate these problems. In this thesis, we cope with these issues and concern about unwelcome problems and struggle with them in circuit level.

To satisfy the power consumption problem, computer architects have focused on designs that integrate several processing cores on a single chip but at the cost of more complexity in programming applications in a parallel fashion. This motivates us in this thesis to concern about hardware transactional memory, one of the state of the art mechanisms which provide acceptable parallel performance and simple parallel code. We propose a circuit solution of such hardware mechanism which attempts to simplify data versioning management, one of the key aspects in hardware transactional memory, and improves the performance considerably.

In this thesis, we also deal with the power consumption in cache memories. Cache memories are known as critical components in nowadays processors especially from the energy consumption point of view. We propose two circuit designs which aim to reduce the power consumption of cache lines during cache access.

Furthermore, we investigate another power reduction method which is very attractive in reducing the energy consumption: supply voltage scaling. However, in spite of its

popularity, it increases the number of memory cell failures. Therefore, in this thesis we propose a cache memory design which is equipped with an effective circuit mechanism in order to be resilient to a large number of cell failures. Our proposed cache configures itself for different supply voltages from the nominal to the near threshold voltage levels and duplicates or triplicates each data line whenever higher reliability is required.

In this thesis, we also attend to one of the emerging technologies called NEMs (Nano-Electro-Mechanical) switches and design a CAM (Content Addressable Memory) cell based on both NEM and CMOS technologies. As a use case, we leverage our proposed cell to design one of the most frequently accessed components of a microprocessor, first-level TLBs (Translation Lookaside Buffers) in order to extremely reduce the energy consumption per search/write operation, standby mode and also usage area.

ACKNOWLEDGMENT

First of all, I would like to express my sincere gratitude to my advisors Adrian Cristal and Osman Unsal for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. I learned a lot from my advisors. Thanks Osman and Adrian. Your guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor for my PhD study. I would also like to acknowledge Ibrahim Hur for his impressive support during the initial phase of my PhD. I also thank Mateo Valero for his dedication and continuous effort in making the Barcelona Supercomputing Center such a great platform for research.

I am thankful to my PhD committee members: Wolfgang Karl, Oguz Ergin, Oscar Palomar, Ramon Canal and Santhosh Rethinagiri, for reviewing my thesis and providing me with valuable comments and constructive feedback.

My sincere thanks also go to Stefan Cosemans, who accepted me very warmly and provided me an opportunity to join IMEC as an intern student. I had a great and productive time in Belgium thanks to Stefan´s positive attitude and enthusiasm. Without his precious support it would not be possible to conduct the last step of this research.

I would like to express my thanks to my former supervisor during my studies at the University of Tehran, Ali Afzali-Kusha, for his endless support and kindness and also his help for preparing necessary documents of my PhD thesis deposit.

I would also like to acknowledge all my friends and colleagues from the office that helped me throughout my PhD; for their insights and expertise in technical matters, and for their unconditional support that has been crucial to keep me sane. Many thanks go to Adria Armejach, Ruben Titos, Vasilis Karakostas, Vesna Smiljković, Vladimir Gajinov, Daniel Nemirovsky, Oriol Arcas, Mario Nemirovsky, Gina Alioto, Nehir Sonmez, Behzad Salami, Srđan Stipić, Gulay Yalcin, Oscar Palomar, Omer Subaci, Nikola Marković, Chinmay Kulkarni, Cristian Perfumo, Timothy Hayes, Paul Carpenter, Francesco Ciaccia, Damian Roca, Cagri Eryilmaz, Saša Tomic, Ivan Ratkovic, Javier

Arias, Milan Stanic, Milovan Djuric, Tassadaq Hussain, Ferad Zyulkyarov, and many others. I sincerely thank you all for your help and all the great moments we have had together.

Finally I would like to express my warmest thanks to my family for supporting me during this endeavor. I would like to extend my deepest gratitude to my beloved uncle and aunt for their never ending concern and enthusiastic support especially during our stay in Belgium. I wish you health, happiness and all the bests you deserve. I especially thank my mom and dad. My hard-working parents have sacrificed their lives for us and provided unconditional love and care. I certainly would not reach this point without their constant supports and encouragements. I love them so much, and I take the chance here to thank them from the bottom of my heart. My special thanks go to my brother and sister, my best friends in all my life. I love them dearly and thank them for all their invaluable support and inspiration. I am always grateful to my beloved husband. He has been a true and great supporter and has unconditionally loved me during all my good and bad moments. This dissertation would have not been possible without him. By all mean this acknowledgement would not be complete without thanking my little son for all happiness he has created in our life and also for his patience during completing this thesis. Thank you my sweet love. I would like to dedicate my dissertation work to my family.

Azam Seyedi
Barcelona, October 2015

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Increasing the number of transistors over the last decades has led to a considerable performance improvement especially for monolithic single-core processors [1]. This was primarily achieved by aggressive technology scaling resulting in doubling of device density and performance doubling every 18 months commensurate with Moore's law [2]. However in the late 1990´s, power-density problems [3] and the difficulty of eking out more performance from complex out of order single core architectures forced the processor manufacturers to do a right-hand turn and introduce chip multiprocessors (CMP) as a solution. Each processor core in these CMPs was relatively simpler, and the increased number of cores provided increased total performance with decreased power-density. However, the same problems of energy-efficiency wall and performance wall have resurfaced with further scaling; exacerbate by the problem of reliability. Hence, industry has been forced to look for novel solutions on a wide variety of aspects from architecture to circuit levels targeting higher performance and efficiency. Motivated by these efforts we try to hit the walls in this thesis and propose some novel circuit designs based on the state of the art techniques to improve the performance, reliability and reduce the energy consumption.

Among computer components we concentrate on the design of two critical structures, cache memories and TLBs. Cache memories improve the processor performance by reducing the average time to access to the main memory. Caches are usually organized as a hierarchy of different levels (level-one, level-two, etc.), and whenever the processor requests the access to the main memory, they are first checked to see whether a copy of that data is there. If so, the processor directly writes to or reads from the cache, which is much faster than accessing to the main memory. If the memory location is not found in the cache, the cache allocates a new entry and copies data from the main memory. Then the processor request is executed. Based on these observations, caches especially first-level ones are activated at each main memory request so considering cache structure in power consumption, reliability and speed point of view are critical for designing future microprocessors. TLBs are also one of our research directions in this thesis. Translation

Lookaside Buffer (TLB) is a cache that is employed to accelerate virtual-to-physical address translation [4]. The processor searches the TLB on every memory operation using the virtual page number. Hence, The TLB is a crucial component for the performance and power consumption of the computers [4]. Based on mentioned discussion, we investigate on these two important components and propose improved level-one caches and TLBs in this thesis:

1- To satisfy the power consumption problem, computer architects have focused on designs that integrate several processing cores on a single chip. However, there are new challenges for designers to design parallel programing methodologies and tackle with multi-core hardware architecture which motivate us in this thesis for considering one of the state of the art mechanisms, transactional memory, in circuit level to greatly improve the performance. Transactional Memory (TM) [5], [6] has been introduced as a promising paradigm to provide acceptable parallel performance and simple parallel code. TM system tackles with two important key aspects: version management and conflict detection. Version management schemes try to apply the mechanisms in which the system can read and modify both old and new version of same logical data during transactional execution. In this thesis, we introduce the circuit design of such hardware mechanism which simplifies data versioning management and improves the performance considerably.

2- In this thesis, we also pay attention to power consumption increase in cache memories when they become bigger and more complicated. Cache optimization is vital from two points of view: first, nowadays caches occupy a major part of each core; therefore, cache memory optimization techniques are necessary to increase the performance and reduce the total power consumption [7]. Second, during each cache access only few lines are active and other lines only consume energy to keep the stored data which causes extra energy consumption. Therefore, in this thesis we propose some power management methods to manage the energy consumption during each cache access.

3- Beside power and performance issues which we investigate in this thesis, reliability is the other design constraint which we concern. Reducing the supply voltage in

microarchitectures and caches is an effective technique for reducing the overall power consumption but at the cost of performance. However, the reliability of this technique is limited by a considerable increase in the number of cell failures especially in on-chip memories [8], [9]. In this thesis, we propose an effective mechanism for level-one caches which provides the highest reliability during the supply voltage reduction.

4- Scaling of available technologies challenges with serious problems that occur due to leakage currents and reliability issues [10], and prevents manufacturers from fabricating smaller devices. In order to maintain the ability to scale circuits, various non-CMOS technologies (emerging technologies) have been employed and become a good replacement for volatile and non-volatile memories. In this thesis, we focus on one of those emerging technologies called NEMs (Nano-Electro-Mechanical switches). We design a CAM (Content Addressable Memory) cell based on NEM and CMOS technologies and employ it to design a power hungry component like TLBs (Translation Lookaside Buffers).

1.2 Problem Statement and Thesis Contribution

1.2.1 Circuit Design of a New L1 Data Cache to Improve Version Management

Transactional Memory (TM) [5] potentially simplifies parallel programming by providing atomicity and isolation for executed transactions. In this thesis, we focus on Hardware TM (HTM) implementations [11], [12], where version management implementation is a key aspect to obtain good performance. Traditional version management schemes, eager or lazy, fail to efficiently handle two versions, old and new, of the same logical data. An efficient version management scheme should be able to read and modify both versions during transactional execution using a fast hardware mechanism. Furthermore, this hardware mechanism should be flexible enough to work with two versions of the same logical data efficiently. We introduce the circuit design of such a hardware mechanism.

In the second chapter of this thesis, we propose a novel L1 data cache design with dual versioning SRAM cells (dvSRAM) for chip multi-processors (CMP) that implements optimistic concurrency proposals. In this new cache architecture, each dvSRAM cell has

two cells, a main cell and a secondary cell, which keep two versions of the same data, the new value and the old value. These values can be accessed, modified, moved back and forth between the main and secondary cells within the access time of the cache using special operations supported by the cell.

We design and simulate a dual-versioning L1 data cache, which we describe in detail as well as the supported operations that allow to efficiently solving existing version management problems. We also introduce three well-known use cases that make use of optimistic concurrency execution and that can benefit from our proposed design. Our experiments show that large speedups can be achieved with acceptable overall energy dissipation.

Also, we introduce a reconfigurable L1 data cache architecture that has two execution modes: a general purpose mode and a TM mode that is able to manage efficiently two versions of the same logical data. The latter allows handling old and new transactional values within the cache simultaneously when executing transactional workloads. We explain in detail the architectural design of this Reconfigurable Data Cache (RDC), as well as the supported operations that allow to efficiently solving existing version management problems. We also evaluate the area and energy effects of our proposal, and we find that RDC designs are more energy-delay efficient compared to baseline HTM systems, with negligible area impact on modern processors.

1.2.2 Two Circuit Power Management Solution to Improve Low Power Cache Designs

On-die caches already occupy more than 50% of the total chip area in recent designs and it will continue to grow as technology scales [7]. Since the leakage current depends on the total number of the on-chip transistors, the proportion of caches in the overall leakage current is large and will continue to do so in future designs. Therefore, leakage current management is an unavoidable approach in total power consumption control in nowadays microprocessors [13].

During each cache access only few cache lines become active and other unselected ones consume energy to retain their data properly. To alleviate this problem, some techniques

have been proposed to put infrequently accessed cache lines into low power standby mode. Besides that, a technique has been broadly called drowsy mode technique put the inactive cache lines into a low leakage state while retain their data properly [14], [15].

In the third chapter of this thesis, we propose two novel circuits to manage the power consumption of inactive cache cells in data retention mode. Both circuits have lower power consumption and area overheads when compared to previous proposals. The first proposed circuit (Dynamic Bias Control circuit or DB-Control circuit) dynamically tracks the reference current and sets the bias voltage of cells, while the second (Self-Adjust Bias Control circuit or SAB-Control circuit) has a self-adjust property to set the bias voltages and also alleviates the instability problems that appear due to noise injection.

Although any SRAM array can benefit from these circuits, to show their usefulness, we frame the study on our dvSRAM cache where leakage current has more effect on power dissipation and circuit instability. Therefore, we add the proposed bias control circuits to a dvSRAM cache and simulate and optimize the entire cache. The simulations demonstrate the effectiveness of our proposed circuits to considerably reduce the energy consumption of dvSRAM L1 data cache compared to the typical dvSRAM cache. This is achieved with a negligible area increase per sub-array and negligible delay overhead. We also show that instability problems are alleviated by using our second proposed circuit.

1.2.3 Circuit Design of a New L1 Data Cache to Improve Reliability

A very effective approach in reducing the energy consumption is to reduce the supply voltage (Vcc) close to the transistor's threshold. However, the energy reduction in the low-power mode comes with a drastic increase in the number of memory cell failures especially in large memory structures such as on-chip SRAM memories (such as L1 and L2 caches) [8], [9]. This motivates us to design a cache which is resilient to large number of cell failures and operates at lower supply voltages.

In the fourth chapter of this thesis, we propose a flexible and reliable L1 data cache design (Flexicache) with the unique capability of automatically adapting itself for different supply voltage levels and providing the highest reliability. Depending on the

supply voltage level, Flexicache defines three operating modes: (1) Single Version Mode (SVM), (2) Double Version Mode (DVM) or (3) Triple Version Mode (TVM). In high supply voltages, Flexicache operates in SVM and provides reliability through single-bit parity. In middle range of supply voltages DVM is used: Flexicache writes data to two separate cache-lines simultaneously in order to use one line for error recovery when the other line is faulty. In near threshold supply voltages, Flexicache operates in TVM and writes data to three separate cache-lines simultaneously in order to provide the correct data based on bitwise majority voter. According to our experimental results, the energy is reduced and latency as well as cache capacity usage is improved compared to typical previous proposals: Triple Modular Redundancy (TMR) (conventional triplication) and OLSC [16], [17].

1.2.4 NEM Based CAM Cell Designed for Highly Efficient TLBs

Among emerging technologies, Nano-electro-mechanical (NEM) switches have been considered as a promising memory technology providing some very appealing characteristics, such as near-zero leakage current and infinite subthreshold slope and non-volatility. However, in spite of their specific characteristics, they have an important constraint: finite write endurance. Therefore, we get benefits from NEM technologies whenever endurance is not a critical issue, e.g. for processor structures that are mostly read-only.

In the fifth chapter of this thesis, we propose a novel Content Addressable Memory (CAM) cell, NEMsCAM, based on both NEMs and CMOS technologies, to employ in processor structures where writes are relatively infrequent, for example the Translation Look-aside Buffers (TLB). The memory component of the proposed CAM cell is designed with two complementary non-volatile NEM switches (two 5-Terminal NEMs) and located on top of the CMOS-based comparison component. As a use case of the NEMsCAM cell, we design and simulate level-one data and instruction TLBs. The simulations show that the NEMsCAM TLB considerably reduces the energy consumption per search operation, write operation and standby mode and the area usage compared to a CMOS-only TLB with minimal performance overhead.

1.3 Thesis Outline

Chapter 2 introduces a level-one dvSRAM data cache as well as its design details, simulation results and evaluation. Also, the design details of the other version of our proposed L1 data cache, Reconfigurable Data Cache (RDC) will be presented in this chapter. Chapter 3 depicts the design details as well as simulation results of two novel circuits which have the ability to manage the power consumption of unselected cache lines in drowsy mode. Chapter 4 presents the circuit design and the evaluation results of a L1 cache design (Flexicache) which can automatically adapt itself for different supply voltage levels to provide the highest possible reliability. Chapter 5 introduces a novel 3-Dimensional Content Addressable Memory cell (NEMsCAM) designed based on both NEMs and CMOS technologies which is used in processor structures like TLBs where writes are relatively infrequent. Finally, chapter 6 concludes this dissertation.

CHAPTER 2

CIRCUIT DESIGN OF A DUAL VERSIONING L1 DATA CACHE FOR OPTIMISTIC CONCURRENCY PROPOSALS

2.1 Introduction

In recent decades, industry has benefited immensely from remarkable progress in chip multiprocessor designs to mitigate power consumption issues. Furthermore, various novel architecture implementations such as speculative multithreading [18], lock elision [19], and hardware transactional memory [6] have been proposed to get more performance offered by these multi-core chips. All of these proposals leverage optimistic concurrency, by assuming that conflicting data accesses will not occur; in case a conflict occurs, all tentative data updates have to be undone. Thus, multiple versions of the same data have to be maintained by the L1 data cache. However, the circuit design of such a cache which supports all these different proposals had not been proposed. In this thesis chapter, we propose a cache, called dual-versioning L1 data cache (dvSRAM) which keeps multiple versions of the same data and supports all mentioned proposals [20], [21].

The main characteristic of the dvSRAM cache is the inclusion of two cells in each dvSRAM cell, main cell and secondary cell, which keep two versions of the same data. These two values can be accessed separately or simultaneously, modified and exchanged within the cache access time using defined operations supported by the cache. Exchange circuits are introduced to connect main cell and secondary cell to each other. When the exchange circuits are inactive, the main cell and secondary cell are isolated from each other.

In Section 2.2 we explain a detailed circuit design of a 32KB dual-versioning SRAM (dvSRAM) L1 data cache, a four-way set associative cache, with 64B data lines, two clock cycles access time and 45nm Predictive Technology Model [22] at 2GHz processor frequency. We propose a complete description of the internal structure of each array and details of all its components, such as address decoders, control signal generator units, data and word-line buffers, sub-array cells, and necessary peripheral circuitries. We simulate both the dvSRAM and a typical SRAM array with HSPICE 2003.03, design the layouts [23], and calculate dynamic and static energy consumptions and access times

for all the operations and all process corners. Also we show that with a few modifications, the dvSRAM array can be presented as a Reconfigurable Data Cache (RDC) which provides two execution modes, a general purpose mode and optimistic concurrency mode.

In Section 2.3 we discuss how the aforementioned optimistic concurrency based systems can benefit from the efficient dual-versioning provided by dvSRAM. Moreover, we evaluate one of the systems using state-of-the-art baseline and benchmarking suite and show that significant speedups are achieved with acceptable overall energy consumption overhead. In Section 2.4 we discuss the related works, and finally we summarize the chapter in Section 2.5.

2.2 dvSRAM design details

In this section, we start with dvSRAM cell design. We describe the cell circuit, and we introduce the available operations of the dvSRAM array structure. Then we describe the different blocks that form the dvSRAM array structure in detail.

2.2.1 Dual-versioning cell design

Figure 2.1 depicts the structure of our proposed dvSRAM cell, which is composed of two typical 6T SRAM cells [24]: the main cell and the secondary cell. These two cells are connected via two tri-state inverters [24]; we call them exchange circuits which are two inverters coupled with two pairs of nMOS and pMOS transistors. The main and secondary cells of one dvSRAM cell keep different versions of the same data. The only time they are not isolated is when the exchange circuits are active, and the data of the main cell is stored to the secondary cell, or the data of the secondary cell is restored to the main cell. When the signals dvRstr and dvStr are low, and the exchange circuits are not active, there is more than one off transistor in the nMOS or pMOS stacks in the exchange circuits, which leads to a significant reduction in the subthershold leakage current that flows through them (stacking effect [25], [26], [27]).

In Table 2.1, we briefly explain dvSRAM cell operations. Read and Write act like read and write operations in a typical SRAM cell. Other operations are created based on our

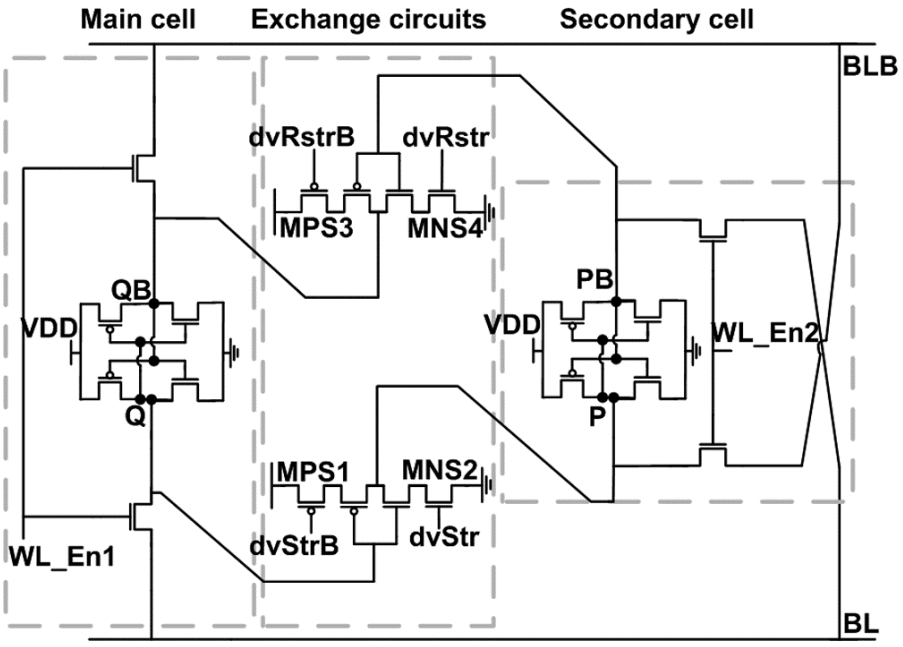


Figure 2.1: Circuit schematic of the proposed dvSRAM cell: a main cell with a secondary cell and exchange circuits.

goal in this chapter. When dvStr is high, transistors MPS1 and MNS2 are turned on, and the exchange circuit acts as an inverter by inverting Q to P. The secondary cell which is formed by two back to back inverters keeps the value of P when dvStr is low, and it inverts P to PB, so that PB has the same value as Q. Similarly, when dvRstr is high, PB in the secondary cell is inverted to QB and converted to Q, so that previously saved data in the secondary cell can be recovered. Note that dvStr operation acts just for a line; when we need to store all the cells to their secondary cells simultaneously, we use dvStrAll operation. We use dvRead to read the data from the secondary cell and dvWrite to write to the secondary cell. Finally, we use dvBWrite to write to both main and secondary cells simultaneously. In Section 2.2.2.2, we describe how these signals are generated.

Main and secondary cells have similar sizes for each dvSRAM cell; BL is connected to Q and PB and BLB is connected to QB and P via pass gate transistors for symmetric behavior. Transistor sizing is an important point in dvSRAM cell design because two SRAM cells retain two different data values. An update to one of the cells should not lead to any changes in the preserved data value of the other cell, so the leakage current

Operation	Description
Write	Writing to a main cell by activating WL_En1
Read	Reading from a main cell by activating WL_En1
dvWrite	Writing to the secondary cell by activating WL_En2
dvRead	Reading from the secondary cell by activating WL_En2
dvStr	~Q→P: Storing the main cell to the secondary cell by activating dvStr
dvRstr	~PB→QB: Restoring the secondary cell to the main cell by activating dvRstr
dvBWrite	Writing to both cells simultaneously by activating WL_En1 and WL_En2
dvStrAll	Storing all main cells to their secondary cells simultaneously

Table 2.1: Brief description of the dvSRAM cell operations.

of the exchange circuits should be as low as possible in order to avoid data values to change due to leakage current (We describe a new design in the next chapter to reduce the leakage current and the instability problem).

2.2.2 dvSRAM array structure design

In this subsection, we describe the high-level organization of the dvSRAM array, considering each part of its structure.

2.2.2.1 Brief description of the whole array structure

An appropriate cache modeling tool that chooses the optimal number of sub-banks, size and orientation is Cacti [28]. We use Cacti to determine the optimal number and size of dvSRAM array components and find the cache architecture with optimal access time and power consumption. For the L1 data cache configuration we assume 32KB, four way, 64 byte lines, two clock cycles access time; and we use the 45nm Predictive Technology Model [22]. For one bank array, Cacti suggests two identical sub-banks, one mat for each sub-bank and four identical sub-arrays in each mat as can be seen in Figure 2.2

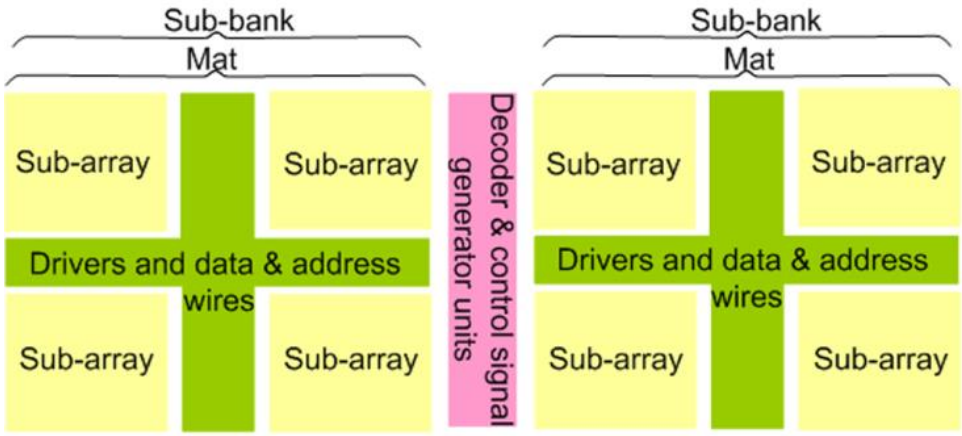


Figure 2.2: dvSRAM array with two sub-banks, one mat in each of them with four identical sub-arrays. Decoder and control signal generator units are in the middle part.

[29]. The number of sub-banks and mats in each sub-bank are the most important results we exploit from the Cacti suggestions ($N_{sub-banks} = \frac{N_{dbl}}{2}$, $N_{mats-in-sub-banks} = \frac{N_{dwl}}{2}$ [28]). The address decoder and control signal generator units are placed in the middle part of the array and necessary drivers and data and address wires in middle part of each sub-bank.

Figure 2.3 shows the complete description of one sub-bank of the dvSRAM array and the middle part, decoder and control signal generator units. Our sub-bank design is based on Cacti suggestions and SRAM configurations that Wang et al. [30] proposed. dvSRAM array parts names come from Cacti suggested names for more clarity and simplicity.

Considering the cache structure of our system, i.e., 32KB size, 64B lines and four-way set associativity, we need seven address bits to address a line in the array, so these seven bits and the necessary control signal bits which we describe later, enter to the left side of Figure 2.3 where the control signal generator units and address decoders can be seen. A complete description of middle part of the array is presented later in Section 2.2.2.2.

During an access, only one of the two sub-banks is activated; hence the numbers of bits that deal with decoder and produce word-line addresses are six; so 64 word-line

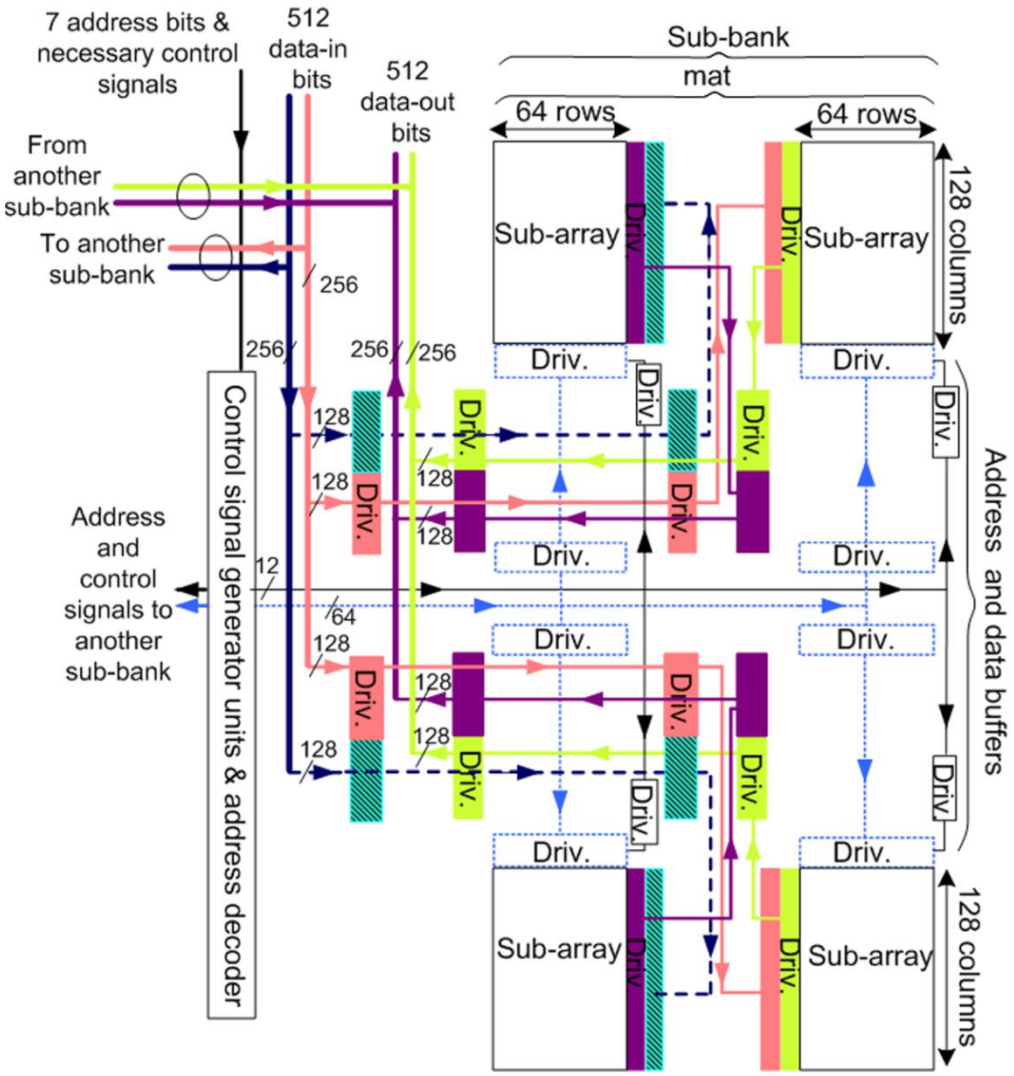


Figure 2.3: A view of one dvSRAM sub-bank and the middle part of array: four identical sub-arrays, each with 64 rows and 128 columns.

addresses and 12 control signals for both sub-banks are generated there in middle part of the array and enter to the sub-bank from the left side as can be seen in Figure 2.3. 512 data-in and 512 data-out bits are routed from the up side of the sub-bank.

As mentioned before, each mat has 4 identical sub-arrays; all of them are activated during an access; therefore each sub-array holds a part of the cache line, so 128 of the 512 data bits (64B) are distributed to each sub-array. The bottom sub-arrays are mirror

images of the top sub-arrays and the left hand side sub-arrays are mirror images of the right hand side ones. The word-line address signals enter the upper sub-arrays through their lower sides and the lower sub-arrays through their upper sides. The data-in and data-out signals enter to the left side of the right sub-arrays and right side of the left sub-arrays. Each sub-array is composed of data-in, data-out and word-line address buffers, a 2D matrix of memory cells and associated peripheral circuitry as explained in Section 2.2.2.3.

We put necessary optimized drivers (chain of two series inverters) in paths to reach their related loads in the sub-arrays. 512 data-in wires enter to up side of each sub-bank. They are divided into four groups of 128 data-in wires, each for one sub-array. For the 128 data-in wires of each sub-array, there are three levels of stage drivers in the path to reach their related sub-array; two stage drivers are in the path and the last one is connected to the sub-array. Each wire and its related drivers have the same color in Figure 2.3. The same method is used for data-out wires as can be seen in Figure 2.3. The upper part and lower part of the sub-array are symmetric.

64 address wires re-drive twice with two levels of stage drivers in their paths to reach each sub-array as can be seen in Figure 2.3. They re-drive with the first stage drivers and reach the second stage which is connected to each sub-array. The driver sizes are optimized to ensure that the access time for all sub-arrays is as equal as possible. As can be seen in Figure 2.3, we put necessary drivers for the 12 control signals as well, using the same method employed for the address wires.

There are two important points to consider when designing the dvSRAM array. First, the wire lengths for both data and addresses from the array edge to each sub-array should be as short as possible; and second, these wire lengths for all sub-arrays of each sub-bank should be as close as possible to each other.

2.2.2.2 Middle part of the array

Figure 2.4 depicts the address decoder, control signal units, and tri-state buffers that reside in the center of the dvSRAM array. To generate 64 word-line addresses from six address bits, A0...A5, we design a two-level decoder similar to the design of Amrutur

[31]. Two 3-to-8 pre-decoders produce partially decoded products, and the main decoder generates 64 word-line addresses from their outputs. 64 word-line addresses are connected to both sub-banks via two tri-state buffer groups with enable signals. We use the highest value bit, A6 bit, as an enable signal to select the sub-bank which is activated during the access time; so when A6 is equal to 1, the left sub-bank is activated and when A6 is equal to 0 the right one is activated.

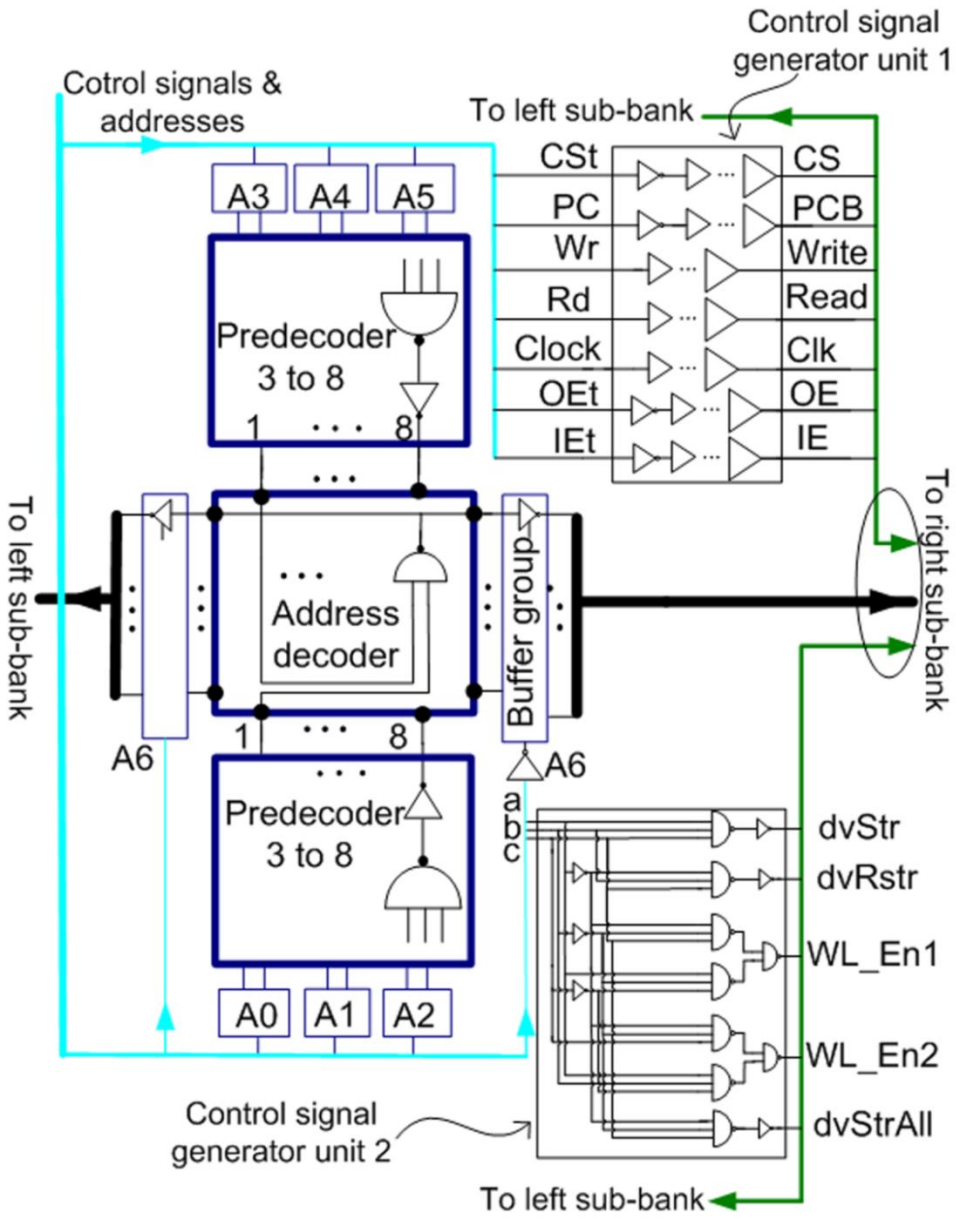


Figure 2.4: Middle part of dvSRAM array: address decoders, control signal units, and tri-state buffers. Boolean functions: dvStr = abc, dvRstr = $\bar{a}bc$, WL_En1 = $a\bar{b}\bar{c} + \bar{a}\bar{b}c$, WL_En2 = $\bar{a}\bar{b}c + ab\bar{c}$, dvStrAll =$\bar{a}\bar{b}\bar{c}$.

Control signal units are responsible to generate the necessary signals for executing the proposed operations and have a synchronous flow-thru SRAM. They use one external clock to time the operations of the dvSRAM array [32]. Control signal generator unit 1 consists of some optimized inverter chains that regulate the signals to control the

dvSRAM sub-array circuits such that main cell operations, typical read and write can be operated correctly. Control signal generator unit 2 is a 3-to-8 decoder, generating the necessary signals for optimistic concurrency proposals: dvStr, dvRstr, WL_En1, WL_En2 and dvStrAll according to its input signal levels, a, b and c.

2.2.2.3 Sub-array structure description

Figure 2.5 shows the structure of one sub-array that consists of 2D matrix of memory cells, data-in, data-out, word-line address buffers, and associated peripheral circuitry. On the left side of Figure 2.5, we can see the word-line address buffers and a set of NOR and inverter gates. As explained before, control signal generator unit 2 generates signals dvStr, WL_En1, WL_En2, and dvRstr to enable word-line address buffers. Each word-line address enters to its corresponding line via one of four buffers. For example, if we want to store the main cells of cache line 1 to their secondary cells, signals dvStr and WL1 are prepared in the control signal unit 2 and decoder and then the related buffer is activated. Control signal generator unit 2 also generates dvStrAll. When dvStrAll is activated, all the main cells of whole sub-array are stored to their secondary cells simultaneously. Data-in, data-out buffers are in the bottom part of the figure. They are typical data buffers that isolate a sub-array with interconnected wires and reduce injected noise effects.

We use typical precharge [33], [24], write circuits [34], and sense amplifiers [33], [24], [34]; the control signal generator unit 1 generates signals to control these circuits. Each column is connected to an individual sense amplifier. The precharge circuit keeps the bit-lines (BL and BLB) high when PCB is low. Both Read and dvRead are like read operation in typical SRAM arrays. After preparing the appropriate word-line address in decoder and the signal Read or dvRead in control signal generator units 1 and 2, the desired word-line is raised, turning on the pass transistors of related dvSRAM cells. Then, the bit-lines relative to the cell nodes that contain the value ‘0’ begin discharging. At this time CS should be high and activates the related pass transistors to connect the sense amplifiers to the bit-lines. The sense amplifiers detect which of the bit-lines are discharging and hence read the stored values [34]. The outputs of sense amplifiers are

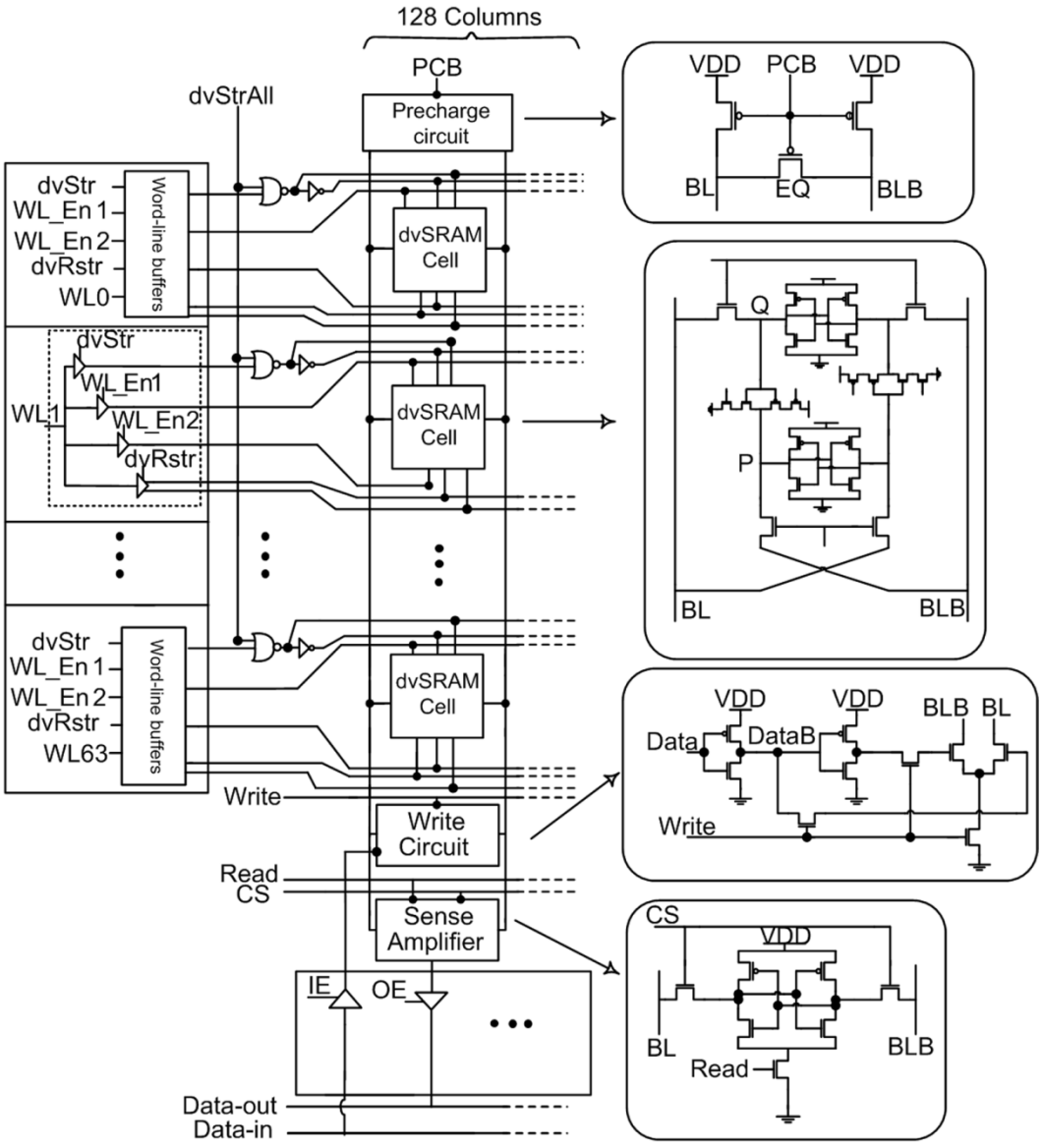


Figure 2.5: The structure of one sub-array.

connected to their related output buffers. When OE is low, the data outputs are always tri-stated. When OE is high, they are active and the data can appear at the outputs.

Similar to a read operation, a write cycle is initiated by asserting WL. At this moment the IE is high and the data inputs can be transferred by the data input buffers to the write circuits. The write circuit is a differential stage which is driven by two Data and Write signals. The Data and DataB signals are passed through write buffers and feed the

differential inputs. Two pass transistors and the current source for the differential amplifier is controlled by the Write signal. Then, the final values and their complements are loaded onto BL and BLB [34]. The circuit behavior for Write and dvWrite is similar. Two signals WL_En1 and WL_En2 are activated simultaneously for the dvBWrite operation and Data is stored to both main and secondary cells at the same time.

Similar to Thoziyoor et al.'s [28] implementation, a layer of multiplexing can be added, before output buffers, at the outputs of sense amplifiers to select the correct way in this 4-way set associative cache; but, for simplicity, we waive this option, and the signal CS (from control signal generator unit 1, from Figure 2.4) in the sense amplifiers selects the correct cache set. We show the simulation waveform of dvSRAM for all operations in Section 2.2.3.

2.2.2.4 Data and address signals distribution

Figure 2.6 shows the distribution of address, data and control signals, the equivalent wire resistance and capacitance and necessary optimized drivers for one sub-array. As seen before, 64 lines of word-line addresses, 128 data-in, 128 data-out, and 12 control signal wires are routed to the sub-array from the left side of the figure. These are long wires with considerable RC delay constant and big propagation delay. For compensating the voltage drop, reflection effect and for driving big word-line capacitances, we divide the wires to shorter length and put drivers (two series inverters) at the end of each piece of wire.

As explained in Figure 2.3, for each 512 data-in (data-out) wires, we put two stages of drivers in the path to reach each sub-array and one driver just in the sub-array and for the 64 word-line addresses and 12 control signal wires, we put one stage of drivers in the path to reach each sub-array and one another driver just in the sub-array. We optimize the sizes of these drivers by considering resistance and capacitance of wire sections and sub-array circuit sizes.

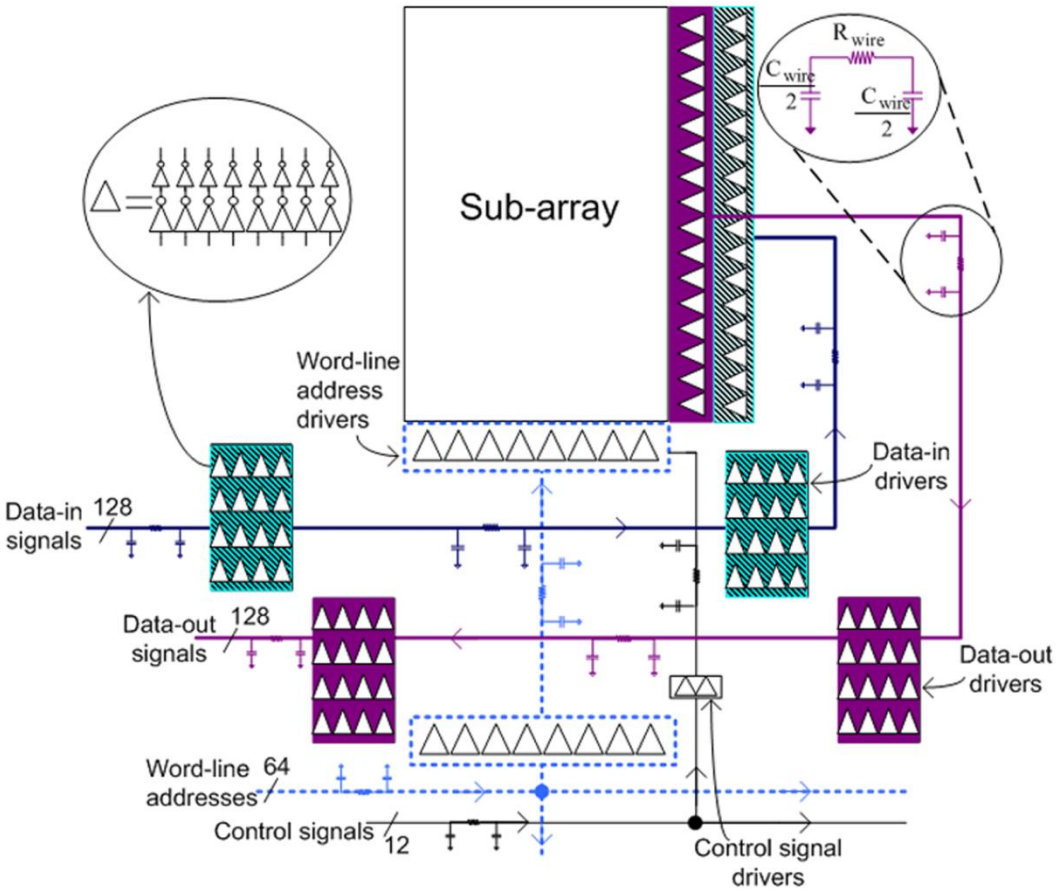


Figure 2.6: The distribution of address and data drivers in a sub-array. The equivalent resistance and capacitance of each wire is shown in lumped model. Each triangle in the driver groups is representing eight drivers (two series inverters).

We can place the drivers of each stage in one line as Cacti suggests [28] but it leads to high area and un-optimized area floor-plan so we use an optimal allocation of each stage drivers that we explain as follows for both data-in and data-out wires: Instead of locating all the drivers (512 drivers) of one stage in one line, we divide them into four groups of 128 drivers (each group for one sub-array) and then we put these 128 drivers in four lines and rotate them 90 degree anti-clockwise as shown in Figure 2.6. Setting word-line address and control signal drivers is easier and we locate them in one line at each stage. In the figure, for simplicity, all related drivers and wires have same color.

2.2.3 Design methodology and analysis

We construct, for one array of dvSRAM and one array of a typical SRAM, HSPICE transistor level net-lists that include the complete decoder, control signal units, and two sub-banks with necessary drivers and equivalent capacitances and resistances of wires. The high level structure of a typical SRAM array is the same as of a dvSRAM array, but with 6T typical SRAM cells, related word-line address buffers and control signal generator units.

We simulate and optimize both the dvSRAM and typical SRAM array with HSPICE using 45nm Predictive Technology Model (High Performance) for VDD=1V, 2GHz processor clock frequency. We calculate the access time, dynamic energy and static energy per access for all operations in dvSRAM and SRAM and present them in Table 2.2. Our analysis indicates that our dvSRAM design meets, as the typical SRAM, the target access time requirement of two clock cycles and acceptable power and area increase. The energy consumption of dvStrAll is much higher than others but as we explain in Section 2.3.4, its effect in total energy consumption is negligible.

Moreover, we calculate dvSRAM energy consumption and access time per operation for three supply voltages, 1V, 0.9V and 0.8V for 45nm Predictive Technology Model and T=110°C and show the results in Figure 2.7 and Figure 2.8. In addition, we calculate dvSRAM energy consumption and access time per operation for all process corners using the 45nm Predictive Technology Model and T=110°C as can be seen in Figure 2.9 and Figure 2.10.

Operation	Energy (pJ)		Time (ps)	
	SRAM	dvSRAM	SRAM	dvSRAM
Read	90.4	112.4	496	604
Write	80.2	99.3	736	838
dvRead	-	114.4	-	576
dvWrite	-	101.1	-	845
dvBWrite	-	122.6	-	870
dvStr	-	114.9	-	599
dvRstr	-	112.9	-	545
dvStrAll	-	1123.2	-	891
Static	35.1	51.4	-	-

Table 2.2: Typical SRAM and dvSRAM energy consumption and access time per operation for VDD=1V and T=25°.

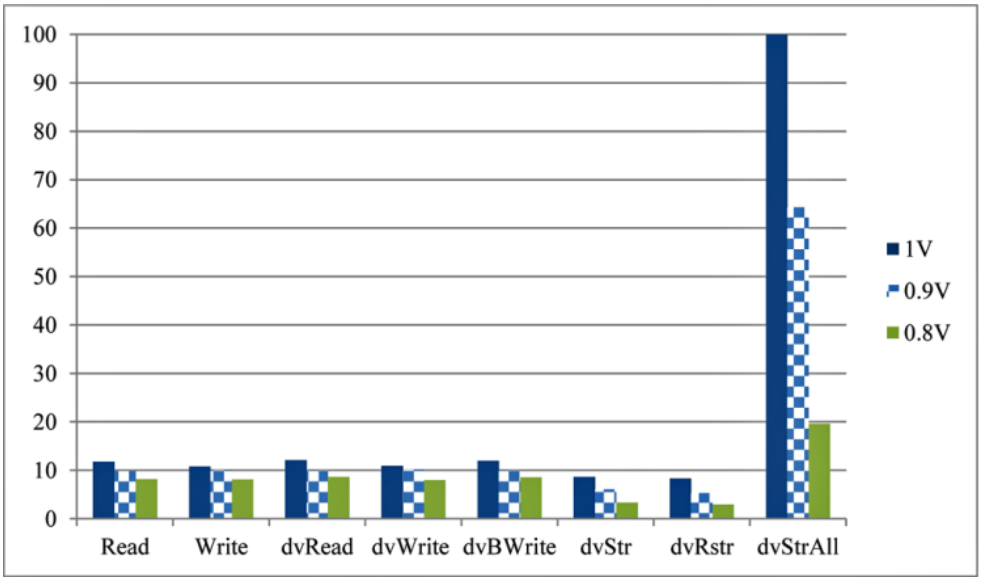


Figure 2.7: Normalized dvSRAM energy consumption per operation for three supply voltages, 1V, 0.9V and 0.8V and T=110°C (Normalized based on maximum value, i.e., energy consumption for strAll when VDD=1V).

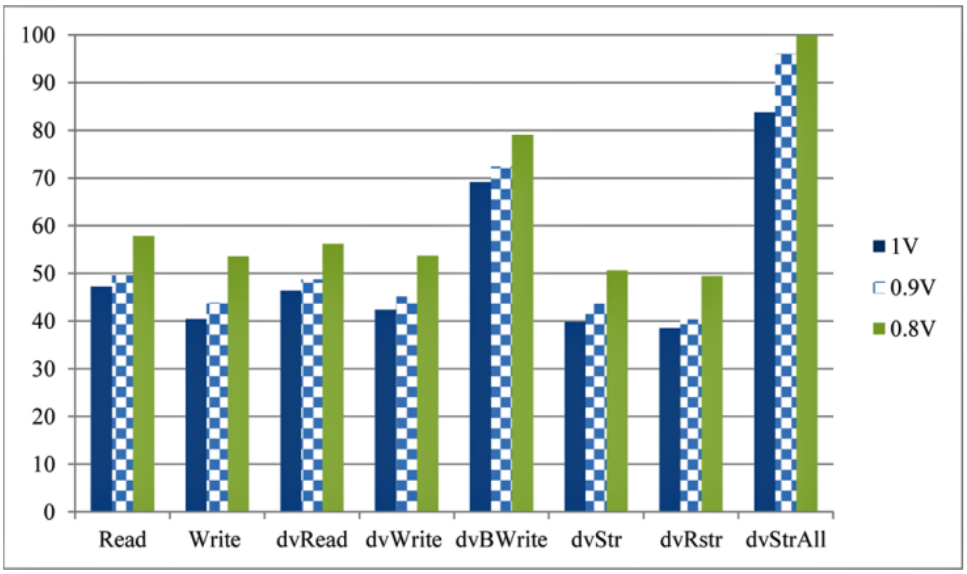


Figure 2.8: Normalized dvSRAM access time per operation for three supply voltages, 1V, 0.9V and 0.8V and T=110°C (Normalized based on maximum value, i.e., access time for strAll when Vdd=0.8V).

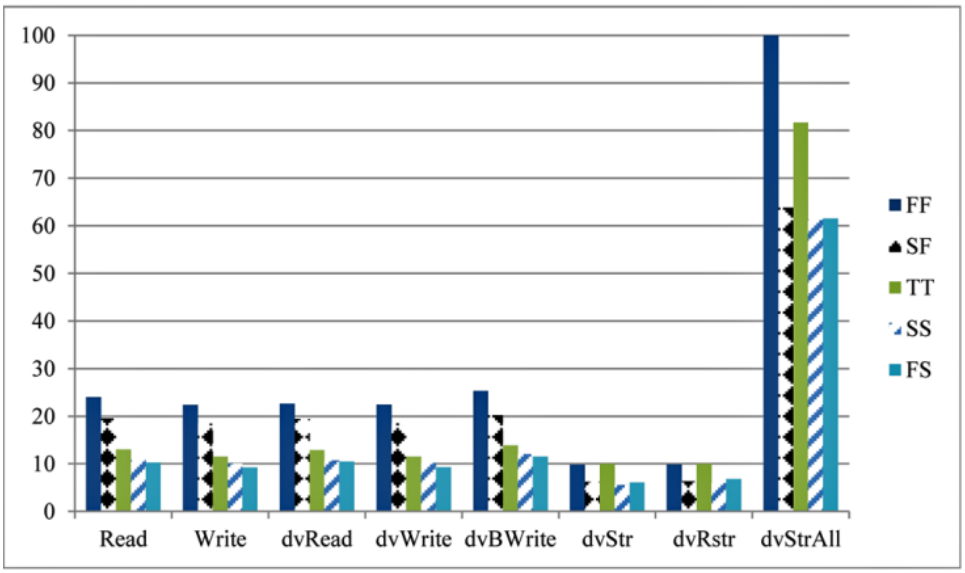


Figure 2.9: Normalized dvSRAM energy consumption per operation for all process corners and T=110°C (Normalized based on maximum value, i.e., energy consumption for strAll in FF process corner).

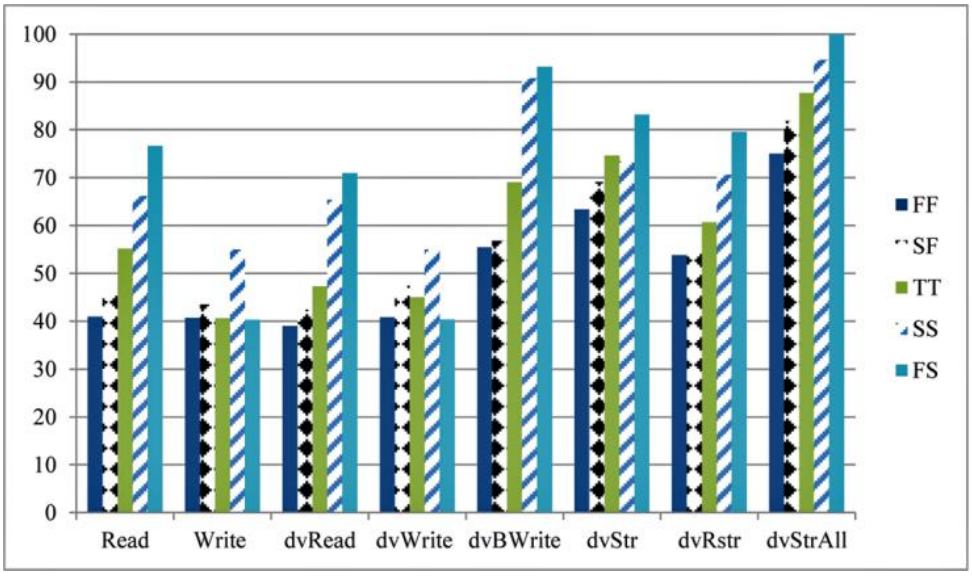


Figure 2.10: Normalized dvSRAM access time per operation for all process corners and T=110°C (Normalized based on maximum value, i.e., access time for strAll in FS process corner).

Figure 2.11 shows the layouts [23] of one sub-bank for both 32KB dvSRAM and the typical 32KB SRAM arrays. They include one sub-bank, address decoders, address and data wires, and the control signals. After adding PADs and I/O interfaces [35], the area increase of the dvSRAM compared to the typical SRAM is 46%. The area devoted to L1Ds in state-of-the-art CMPs is small compared to the entire die, for example, in Power7 [36] it is less than 1%, therefore the dvSRAM area overhead in die size is modest. With a few modifications, we present the dvSRAM array as a reconfigurable array which provides two execution modes, a general purpose mode and optimistic concurrency mode [37]. The general purpose mode allows for a 64KB cache and the optimistic concurrency mode for a 32KB cache with dual-versioning support. By using this technique, the area increase when comparing to a typical SRAM array of 64KB is 15.2%. We explain the design details in Section 2.2.4.

Figure 2.12 shows the simulation waveform for a sequence of eleven operations that take 11ns for the first cell at the last row of one sub-array. Q63, P63 and Out63 are internal nodes and dvStr63, dvRstr63, WL63_En1 and WL63_En2 are the outputs of the last

word-line address buffers. Signals PC, CS, CLK, OE, IE, a, b, c are omitted because their behaviors are similar to the related signals in a typical SRAM array [24].

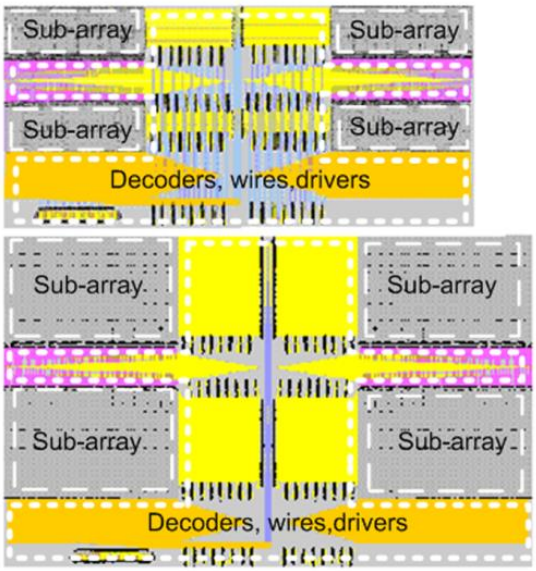


Figure 2.11: Typical SRAM (top) and dvSRAM (bottom) layouts. Showing one sub-bank and address decoders.

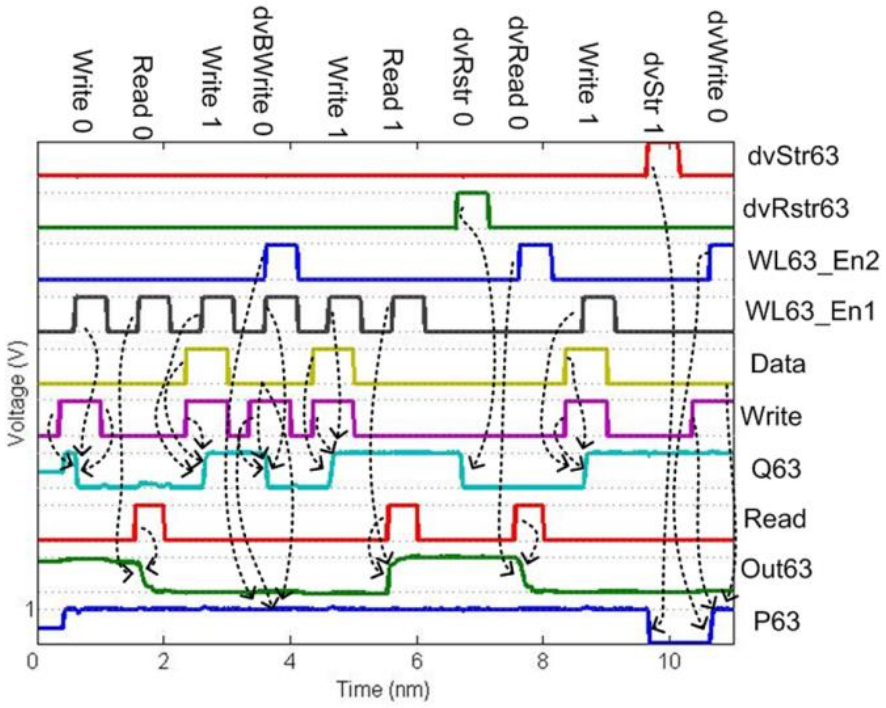


Figure 2.12: The simulation waveform of dvSRAM array for a sequence of eleven operations for the first cell of the last row of one sub-array. The operations are shown on top.

2.2.4 Reconfigurability: RDC execution modes

As mentioned before, with a few modifications, we present the dvSRAM array as a Reconfigurable Data Cache (RDC) which provides two execution modes, a general purpose mode and an optimistic concurrency mode. We explain this structure in this section.

The reconfigurable L1 data cache provides two different execution modes. The execution mode is indicated by a signal named Transactional Memory Mode (TMM). If the TMM signal is not set, the cache behaves as a 64KB general purpose L1D cache; if the signal is set, it behaves as a 32KB cache with the capabilities to manage two versions of the same logical data. Figure 2.13 and Figure 2.14 show an architectural diagram of RDC, the decoder details and its associated signals, which change depending on the execution mode; the design assumes 48-bit addresses and a 4-way organization with 64-byte cache lines.

2.2.4.1 General purpose mode (64KB)

In this mode, the upper and lower bit-cells inside of a cell contain data from different cache lines. Therefore, a cache line stored in the upper cells belongs to cache set i in way j, while a cache line stored in the corresponding lower cells belongs to set i+1 in way j (i.e., consecutive sets in the same way). This mode uses the first four operations described in Table 2.1, to perform typical read and write operations as in any general purpose cache. Figure 2.13 shows an architectural diagram of the RDC. As can be seen in the figure, the most significant bit of the index is also used in the tags to support the 32KB TM mode with minimal architectural changes, so tags have fixed size for both modes (35 bits). The eight index bits (A13...A6) are used to access the tags (since TMM is not set) and also sent to the decoder. In Figure 2.14 it can be seen how the seven most significant bits of the index are used to address the cache entry while the least significant bit (A6) determines if the cache line is located in the upper or the lower cells, by activating WL1 or WL2 respectively.

2.2.4.2 TM mode (32KB)

In this mode, each data bit has two versions: old and new. Old values are kept in the lower cells and new values are kept in the upper cells. These values can be accessed, modified, and moved back and forth between the upper and lower cells within the access time of the cache using the operations in Table 2.1. To address 32KB of data, only half of the tag entries that are present in each way are necessary. For this reason, as can be seen in Figure 2.13, the most significant bit of the index is set to ’0’ when the TMM signal is active. So, only the top-half tag entries are used in this mode. Regarding the decoder (Figure 2.14), in this mode, the most significant bit of the index is discarded, and the rest of the bits are used to find the cache entry, while the signals a, b, and c select the appropriate signal(s) depending on the operation needed.

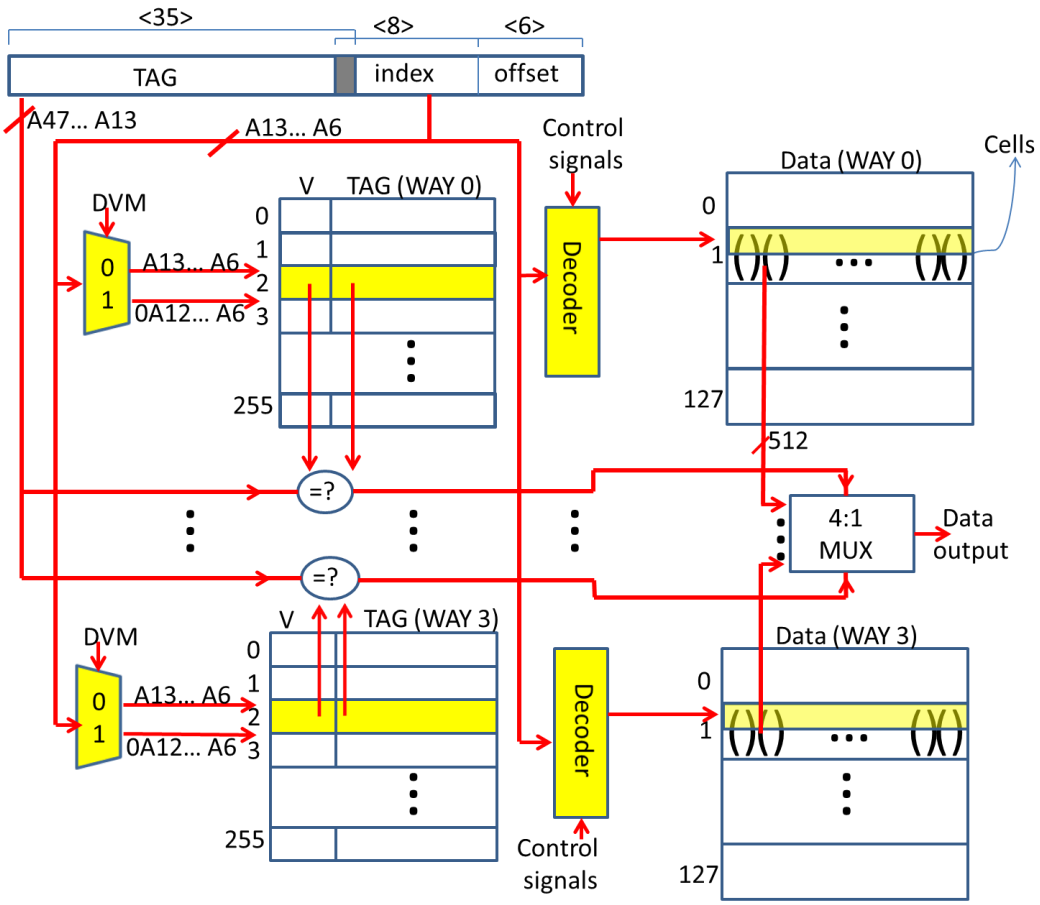
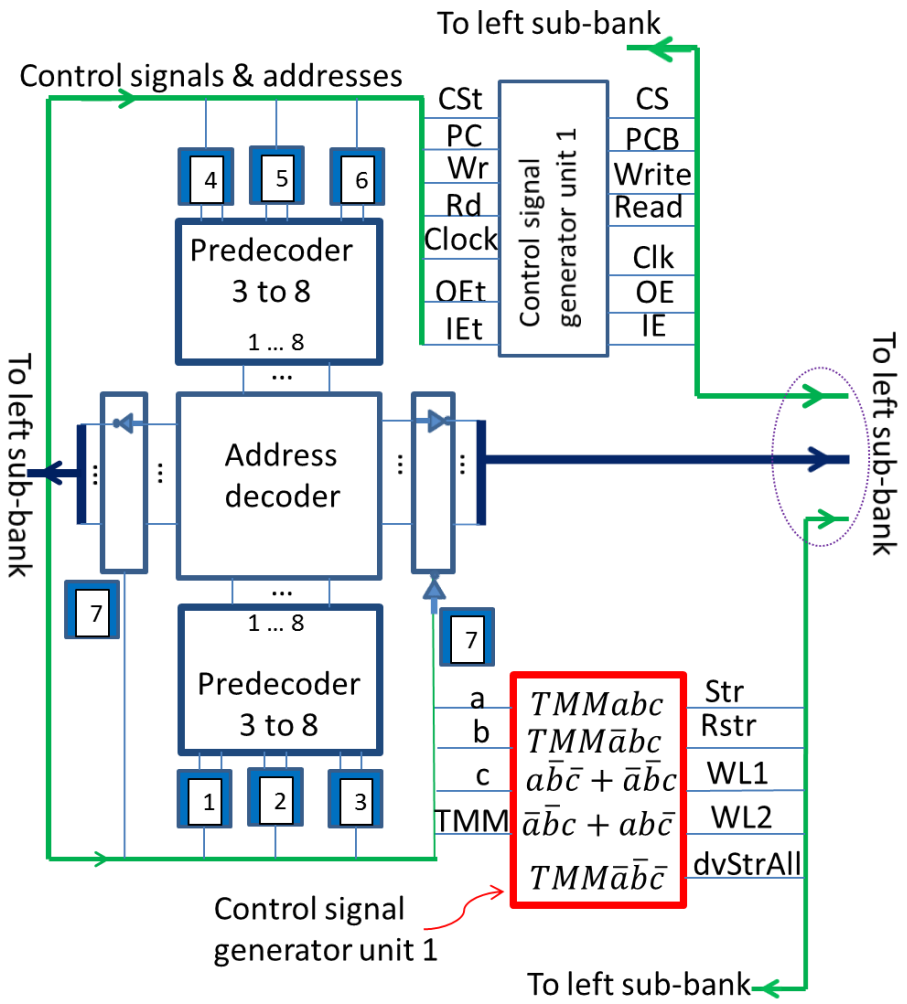


Figure 2.13: RDC architectural diagram: considering a 4-way RDC, with 64B cache-lines and 48b addresses.



(a)

	Address bits and control signals										
Execution Mode	1	2	3	4	5	6	7	a	b	c	TMM
64KB general purpose	A7	A8	A9	A10	A11	A12	A13	1	A6	0	0
32KB dual-versioning	A6	A7	A8	A9	A10	A11	A12	X	X	X	1

(b)

Figure 2.14: (a) Decoder details and associated signals used for each execution mode. (b) Depending on the TMM signal, address bits and control signals for an execution mode are generated and passed to the decoder.

2.3 Use cases

In this section we discuss several use cases for the proposed dvSRAM. Moreover, a short evaluation for one of the use cases follows to show the impact of the dvSRAM.

2.3.1 Speculative multithreading (SpMT)

SpMT [38] is a concurrency mechanism that attempts to speedup sequential executions by partitioning the workload in two threads. The second thread executes the bottom half of the sequential program optimistically. By using a mechanism like the dvSRAM, each thread can use one of the independent cells, providing an easy and efficient versioning management and a fast in-place conflict detection mechanism with state bits.

2.3.2 Speculative lock elision (SLE)

In SLE [19] the system tries to avoid waiting in a lock when it might not be necessary by predicting potential non conflicting executions, allowing several threads to execute the same critical section optimistically. Similarly to TM, SLE has to deal with speculative updates and can benefit from the efficient dual-versioning management of the dvSRAM.

2.3.3 Transactional memory (TM)

Transactional memory [6] is a promising technique that helps with parallel program development by abstracting away the complexity of managing shared data. TM uses optimistic concurrency, assuming that conflicting data accesses will not occur; in case a conflict occurs then one or more transactions must abort, undoing all tentative data updates. This requires a multi-versioning mechanism to restore previous state. By using the dvSRAM, a partial snapshot of the past state of the system can be maintained at L1D level, leveraging a fast mechanism to restore previous state, and rely in slower mechanisms only if it is strictly necessary.

In Figure 2.15 we illustrate how the dvSRAM is used in an optimistically concurrent system like the ones described above. The example shows one possible way to use the dvSRAM, even though others might be more convenient depending on the applicability. As we can see, when an optimistic execution (speculation) starts, the system creates copies in the secondary cells by using the dvStrAll operation (Figure 2.15a). During the

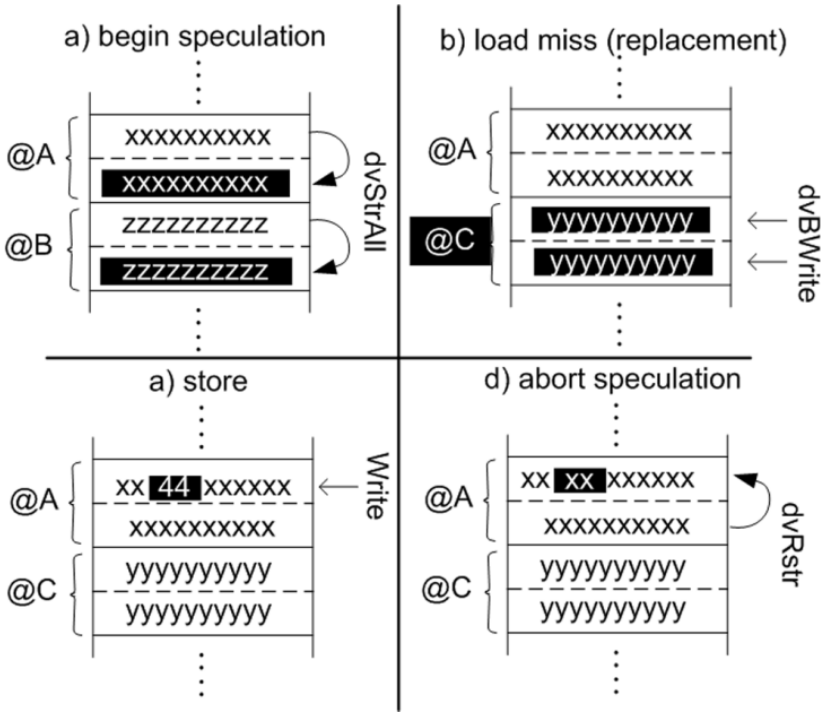


Figure 2.15: Simple usage example of the dvSRAM in an optimistically concurrent system. Black back-ground indicates state changes.

speculative section, new lines can be added to the dvSRAM upon a miss (Figure 2.15b), and modifications are done in the main cell (Figure 2.15c). If the system has to abort due to a conflict, it rolls-back its state by using fast dvRstr operations (Figure 2.15d). A comprehensive proposal of the TM use case is discussed in [37].

2.3.4 Results: transactional memory

State-of-the-art HTM proposals [12] can be adapted to work with the dual-versioning cache. In particular, we evaluate in a Full-System simulation environment a state-of the-art log-based Hardware Transactional Memory (HTM) as baseline (labeled Base), and a modified version of the baseline that uses our proposed dvSRAM and logs as a fall-back mechanism only for cache lines that overflow the L1D cache (labeled DV) [37]. We selected four applications of the STAMP [39] benchmark suite, which represent different amounts of contention, to evaluate the dvSRAM TM applicability in both performance and power consumption. In general, highly contended benchmarks scale poorly due to a larger number of aborts. This is the case of Intruder which performs 6.31× better when

using the dvSRAM. Yada with moderate contention also shows a significant speedup of 2.24×, while low contention benchmarks like Genome and KMeans show 1.15× and 1.09× speedups respectively. This is achieved by completely avoiding the use of software logs in a large percentage of transactions (over 90% on average).

Regarding power consumption, as can be seen in Figure 2.16, two applications are less energy efficient compared to a typical SRAM cache, while the other two are more energy efficient due to larger execution time speedups. In addition, for all the applications, the amount of energy spent in specific dual-versioning operations is not significant compared to the usual (Read, Write) operations and the static energy. Note that the energy results are considering only L1 energy consumption, and the L1 data cache accounts for a very small fraction of an entire processor, as discussed in Section 2.2.3. Thus, the energy impact considering an entire processor would be palliated.

Table 2.3 shows the average number of each operation for all tested applications of the STAMP benchmark suite. It is observed that the total number of accesses to the secondary cells is much lower than the total number of accesses to the main cells (4% of total accesses are accesses to the secondary cells). It can be seen that in spite of high energy consumption of dvStrAll, the number of times this operation is performed is very low compared to the number of Read and Write operations.

2.4 Related work

Ergin et al. [40] proposed similar work using a shadow cell SRAM design for checkpointed register files. In that technique, each bit-cell has a shadow-copy cell to store the temporal value which can be recovered later. They use two cells, consisting of two back to back inverters, connected to each other using two pass transistors and two inverters. Even when both check-point and recover signals (the enable signals copy the data to the shadow-copy cell and recover it later) are inactive, pMOS transistors or nMOS transistors of these inverters are always on leading to power consumption increase. Moreover, if we want to modify this structure for our purpose, we should make upper inverters of cells bigger to mitigate the imbalance of this structure. While the situation is a bit better when we replace the inverters and pass transistors with each

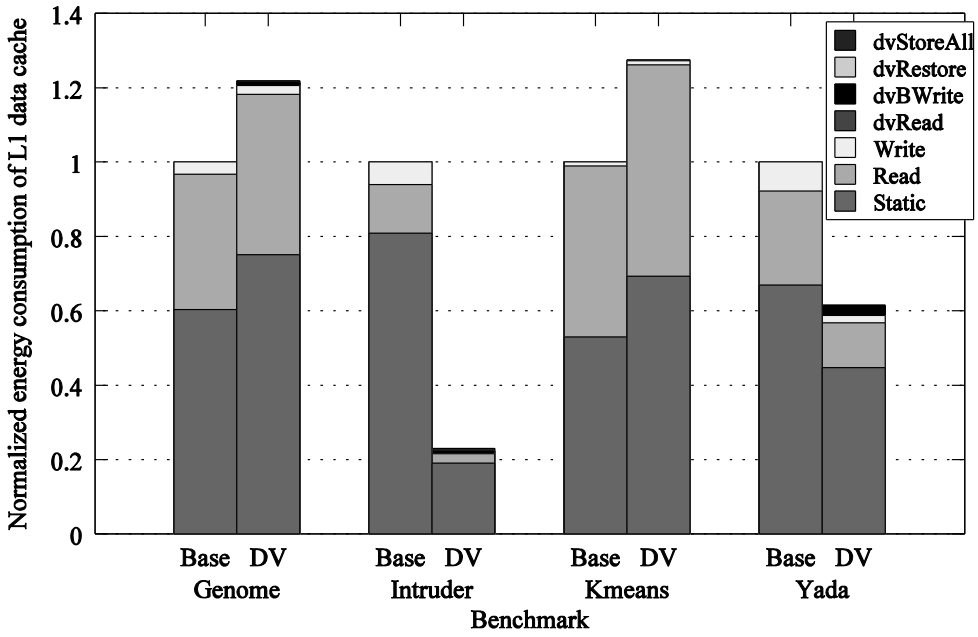


Figure 2.16: Normalized energy consumption break down of L1 data cache (Base: Baseline-HTM, DV: Dual-Versioning-HTM).

Read	Write	dvRead	dvBWrite	dvStr	dvRstr	dvStrAll
12912744	879772	184	431475	0	178729	20852

Table 2.3: The average number of each operation for all tested applications of the STAMP benchmark suite (gnome, intruder, kmeans and yada).

other, we still have imbalance problem which cause instability issues. All these problems lead us to design a new cell with more capabilities that we can use in L1 data cache for optimistic concurrency.

In our dvSRAM structure, both cells, the main cell and the secondary cell are isolated from each other with two exchange circuits and this leads to lower static power consumption. Also, there are no unnecessary on transistors in the active mode. We calculate and compare the static power for our dvSRAM cell and Ergin's proposed cell to prove our discussion with HSPICE using 45nm Predictive Technology Model (HP) for VDD=1V. The static power of the dvSRAM cell is 28.35µW compared to the 51.08µW of Ergin's cell [40].

Valero et al. proposed a SRAM-DRAM hybrid macrocell to implement first level data caches [41]. The proposed macrocell stores n-bits using one 6T SRAM cell and (n-1) 1T1C DRAM cells. Both SRAM and DRAM cells are accessible externally but data can only transfer internally from SRAM cell to each DRAM cell by a pass gate transistor which acts as a unidirectional bridge; so it is not suitable for optimistic concurrency proposals where a main cell and a secondary cell data should be accessed, modified, and moved back and forth independently.

On the other hand, Kei et al. proposed a SRAM-DRAM hybrid memory where each SRAM cell is augmented with n DRAM branches (2n 1T1C DRAM cells) in a way that a value can be locally copied between the SRAM cell and one of the DRAM branches within a cell [42]. Data can be internally moved between SRAM and DRAM while only allowing access through SRAM cells which is applicable to register files not for first level data caches.

We can use single-electron transistors and implement L1 data cache with multi-valued SRAM [43], [44] which has much smaller area compared to our circuit; but this technique is too complex for our purpose because of complicated input and output circuitries for each cell.

2.5 Summary

In this chapter, we propose a new L1 data cache with dual-versioning SRAM cells (dvSRAM) that aims to overcome the data versioning problem present in optimistic concurrency mechanisms. We present the design details and its available operations. We calculate the power consumption, access time and design the layout. We introduce three use cases that can benefit from our proposed design. We evaluate one of them, Hardware Transactional Memory, to show our design impact in terms of performance and energy consumption. Our experiments show that the dvSRAM allows for significant performance gains with acceptable costs in power, delay and area. Also, with a few modifications, we present the dvSRAM array as a reconfigurable array which provides two execution modes, a general purpose mode and optimistic concurrency mode.

CHAPTER 3

NOVEL SRAM BIAS CONTROL CIRCUITS FOR A LOW POWER L1 DATA CACHE

3.1 Introduction

On-die caches can consume considerable percentage of the chip area in recent designs as technology scales [7]. With a huge increase in leakage current on every technology generation, and with the dependence of the leakage power consumption on the number of transistors, caches will continue to account for a large component of leakage power dissipation in microprocessors [13]. Therefore, in order to keep the cache leakage current under control, deployment of some circuit techniques is unavoidable.

In the other side, when a cache line is accessed, only a few parts of the whole cache get activated, and all un-selected rows just consume leakage power for data retention, so a leakage reduction technique is desirable to allow activating only small portions of the cache so that unused SRAM cells can be put into a low leakage state while retaining their data properly. This technique has been broadly called drowsy mode in prior works [14], [15].

In this thesis chapter, we propose two novel bias control circuit techniques to reduce the power dissipation in drowsy mode [45]. Our first proposed circuit, termed Dynamic Bias Control (DB-Control) circuit can dynamically track a reference current and produce an accurate cell bias voltage to reduce the power consumption. In DB-Control circuit, an external current source tracks and compensates any changes in the array´s leakage current; it enters each array and is transferred to a simple current to voltage converter circuit to produce the virtual VSS voltage level in drowsy mode.

Our second proposed circuit termed Self-Adjust Bias Control (SAB-Control) circuit sets the bias voltage to reduce the power consumption, and also has a self-adjust property to alleviate instability problems that might happen in drowsy mode. SAB-Control circuit has two components: one generates virtual VDD, and the other generates virtual VSS; each consists of two diode-connected transistors in series with a resistor. By proper transistor sizing, the desired virtual VDD or virtual VSS level can be achieved;

moreover, SAB-Control circuit adapts the voltage levels in the presence of noise injection and thus helps to reduce instability problems.

Our proposed bias control circuits are easy to implement and also have less overhead in terms of energy consumption and area compared to recent designs [15]. In addition, SAB-Control circuit alleviates instability problems; this is important for drowsy caches, which usually tend to particularly suffer from instability problems due to noise [46].

We evaluate our proposed bias control circuits by adding them to our recently proposed 32KB dual-versioning SRAM (dvSRAM) cache (we describe its detail in Chapter 2), where active leakage current has a strong effect on energy consumption and instability problems. Then we optimize and simulate the entire cache. Our simulation results show that the energy consumption of the dual-versioning SRAM cache when using our proposed bias control circuits (dvSRAM-D) is, on average, 35.8% less than in the typical dvSRAM cache (dvSRAM-T). This is achieved by negligible delay overhead and reasonable area increase at, 1.6% per sub-array, about 50% lower compared to previous techniques which required around 3% area increase per sub-array [15].

The rest of this chapter is organized as follows: Section 3.2 reviews some previous related works in cache power reduction in drowsy mode. In Section 3.3 we introduce our bias control circuits. In Section 3.4 we analyze the activation probability of dvSRAM cell components. In Section 3.5 we include the block diagram of a low power dvSRAM array when we add our proposed bias control circuits (dvSRAM-D). Section 3.6 details the simulation results. In Section 3.7, we explain how our bias control circuit implementation alleviates aforementioned instability problem; and finally we summarize the chapter in Section 3.8.

3.2 Background: available techniques to reduce cache power consumption

During a typical cache access, only few cache circuit blocks are activated and un-active parts only dissipate leakage power. Many circuit techniques have been proposed to reduce the leakage power of un-selected cache parts. Some of them keep the un-active parts in sleep mode without attention to whether their data can be retained or not [14],

while some of them define the drowsy mode for un-selected parts and can preserve the state of transistors in cache cells [47]. In drowsy mode techniques, the cell bias voltage is reduced and it is clamped at the level where the memory cell data can safely remain. There are many proposed drowsy mode techniques; one simple method is to use a sleep transistor to set VSS to a voltage higher than zero. However, in spite of its simplicity, the sleep transistor needs to be sized properly to meet the wake-up timing requirement, to set the optimal VSS voltage level during the data-retention mode and to keep the IR droop low enough in the current path during active mode [48], [49].

Another drowsy mode technique is the use of a diode-connected transistor to control the VSS voltage level and clamp it to the desired voltage level [50]. Even in some improved techniques proposed later, such as active replica cell technique [51] and source-line self-bias technique [52], the major shortcoming is not solved completely: the variation of V_{th} directly impacts the accuracy of the VSS voltage level, and also the VSS voltage level is going to be around V_{th} of the clamped transistor which is too high.

Furthermore, programmable bias transistors technique is introduced to solve the above problems [53]. This technique has to be set for the worst case or maximum SRAM leakage current. In addition, aging can change the device characteristics and thus should be accounted as well. This makes programmable bias transistors technique less effective in reducing cache leakage power [15].

Also, a sleep transistor design with an active feedback based on Opamp is also proposed to overcome the variation problems. The proposed configuration guarantees that the voltage magnitude of the VSS node is no greater than a V_{REF} [15], [49]. The benefit of this technique compared to the previous ones is that the standby power minimization is done by tracking and compensating any changes in leakage current. The major drawback of this scheme is the DC current power consumed by the Opamp which has to be replicated along each data bank to provide the needed granularity. Against all of them, our proposed circuits have lower area overhead, lower energy consumption and settling time, and they also are easier to implement. In Section 3.3, we explain the design details.

3.3 The proposed bias control circuits

In this section, we introduce our proposed circuits. Even though these circuits can be used in any component of the cache when in standby mode, we suggest, given their characteristics, to apply the DB-Control circuit to memory cells, which have to retain saved data while in drowsy mode; and the second circuit, SAB-Control circuit, which is even easier to implement, for the rest of the cache components.

3.3.1 DB-Control Circuit

Figure 3.1 shows DB-Control circuit to generate and control the virtual VSS voltage level. To design this circuit, we use a reference current and current to voltage converter circuits. The desired reference current is generated outside of the array and enters to a central current mirror circuit [54], which is located in the middle part of the array. The central current mirror circuit transfers this reference current to each sub-array via one intermediate current mirror circuit and a current to voltage converter circuit which its output node is connected to the virtual VSS rail of the sub-array. We use this intermediate current mirror circuit to increase the accuracy of our design because there is long distance between the central current mirror and the sub-array, so we cut the wire in the middle and add this current mirror [55].

Because of large resistance and capacitance of long wires, the gate to source voltage of MN4 is lower than the gate to source voltage of MN1 which forces us to apply an accurate current mirror model similar to what has been proposed by Redouté and Steyaert [55]. If a typical current mirror structure [54] was used, equalizing the currents through MN1, MN4 and MN7 and also setting virtual VSS to the desired voltage level would be very time consuming.

Transistor MN3 isolates the mirror node from the drain of MN1, transistor MN2 completes the DC biasing and finally transistors MN2 and MN3 fix the gate to source voltage of MN1 [55]. By optimizing the transistor sizes and considering resistance and capacitance of interconnections, the current I1 is equal to the current through MN1, IREF. A diode-connected PMOS Transistor (MP1) acts as a load for the intermediate

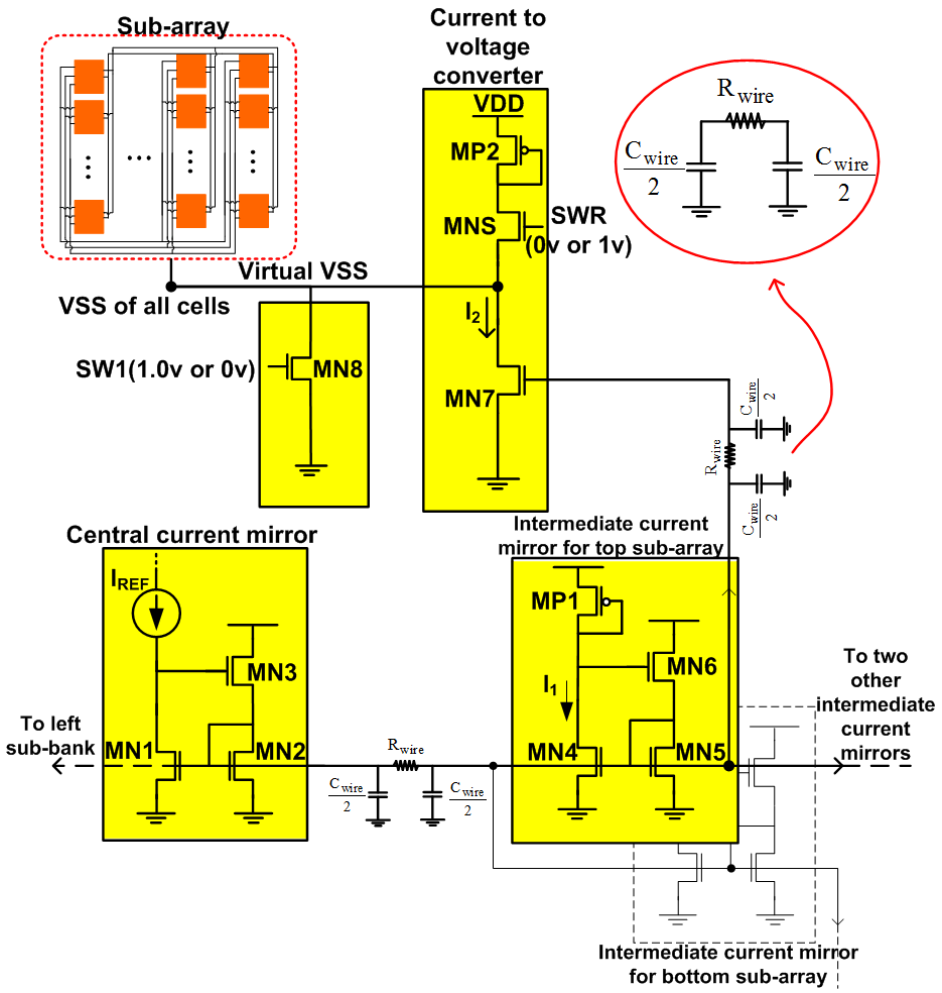


Figure 3.1: The proposed DB-Control circuit to generate and control virtual VSS of one sub-array. In the figure, a central current mirror circuit (located in the middle part of the array) and two intermediate current mirrors for the top and bottom sub-arrays (of the right sub-bank) are depicted; in addition, there is one current to voltage converter circuit for the top sub-array. Note that the central current mirror is also connected to two other intermediate current mirrors used by the rest of the sub-arrays of this sub-bank. Our cache structure is similar to dvSRAM cache structure [20].

current mirror circuit [54]. MN5 and MN6 act similarly than MN2 and MN3 to regulate the gate to source voltage of MN4.

By proper transistor sizing and considering resistance and capacitance of interconnections, I1 becomes equal to I2. The current to voltage converter consists of transistor MN7 and a diode-connected PMOS transistor, MP2, which converts the current I2 to the desired voltage level at the output node. Moreover, the diode connected

PMOS transistor (MP2) tries to keep the virtual VSS voltage at the desired level. When the SRAM cells are not active, SW1 (SWR) is equal to zero (one) and transistor MN8 (MNS) is turned off (on). In active mode SW1 (SWR) is high (low) and transistor MN8 (MNS) is turned on (off), and virtual VSS is discharged to zero.

3.3.2 SAB-Control Circuit

Here, we introduce the SAB-Control circuit to generate and control virtual VDD and virtual VSS. Figure 3.2 shows two bias control circuits, one for controlling virtual VDD and another for controlling virtual VSS. Virtual VSS is connected to VSS (Ground) via one resistor, R1, in parallel with a switch transistor, MN9, and also to VDD via a series of two diode-connected PMOS transistors, MP3 and MP4. A similar structure is used to generate virtual VDD; virtual VDD is connected to VSS via a series of two diode-connected NMOS transistors, MN10 and MN11 and also to VDD via one resistor, R2, in parallel with a switch transistor, MP5.

When the sub-array is accessed, SW3 is high, SW4 is low, and virtual VDD and virtual VSS are strongly connected to VDD and VSS. Whenever the sub-array is not accessed, SW3= 0V and SW4=1V, the virtual VSS rises to a voltage level equal to the voltage drop across R1 and also the virtual VDD is reduced to a voltage level lower than VDD as the magnitude of the voltage drop across R2. By proper sizing of transistors and resistors, we can clamp the virtual VDD and virtual VSS to the desired voltage levels in drowsy mode. So whenever the sub-array is not accessed, the bias voltage level is reduced which leads to a leakage power reduction. In Section 3.7, we explain how this bias control circuit can reduce the instability problems that might happen during drowsy mode.

3.4 Analysis of the DVSRAM L1 data cache

3.4.1 Profiling dvSRAM Components Activation

In Chapter 2, we propose the dvSRAM cell structure, introduce its predefined operations, detail a level one cache implementation based on this novel cell, and finally report the results for a Hardware Transactional Memory (HTM) use case [19]. In this chapter, we use a similar setup to obtain the results shown in Table 3.1. The data

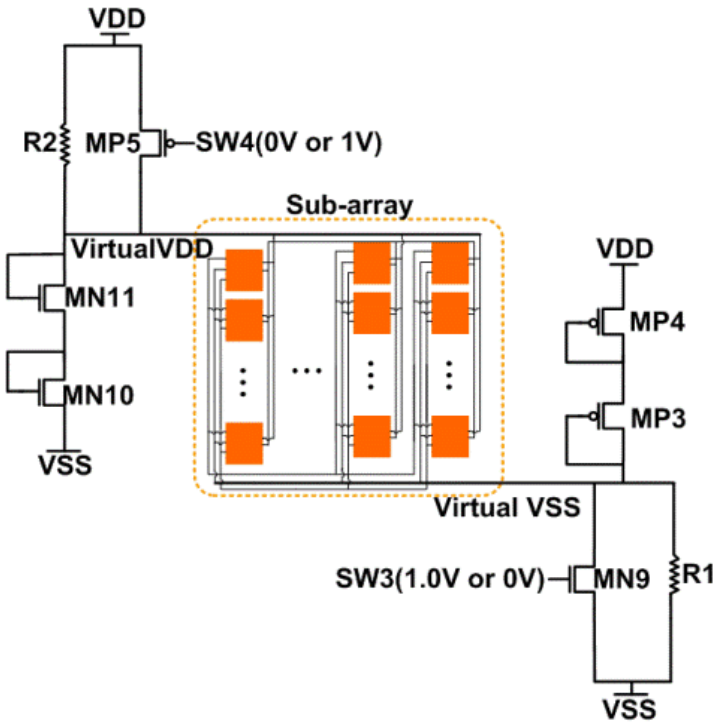


Figure 3.2: The proposed SAB-Control circuit to generate and control virtual VDD and virtual VSS for one sub-array.

indicates the number of operations of each kind done by the dvSRAM array for the four tested applications of the STAMP [39] benchmark suite, the state-of-the-art suite for HTM evaluation. It can be seen that the number of Read and Write operations is much higher compared to the others.

Note that, for each operation, different parts of the selected dvSRAM cells are activated. For example, for the Read operation the main cells are activated, but for the dvBWrite operation, both the main and secondary cells are activated. Table 3.2 shows the active parts of dvSRAM cells for each operation. If it turns out that the probability of access to secondary cells is low, potentially large energy savings can be obtained from techniques that target secondary cells. Next, we calculate these probabilities:

The number of accesses to the main cells is calculated by adding up the total number of the following operations: Read, Write, dvBwrite, dvStr, dvRstr and dvStrAll. The number of accesses to the secondary cells is calculated by adding up the total number of the following operations: dvRead, dvWrite, dvBwrite, dvStr, dvRstr and dvStrAll, and

	Read	Write	dvRead	dvBWrite	dvStr	dvRstr	dvStrAll
Genome	400449	25028	5	6873	0	103	406
Intruder	200258	31571	2	22546	0	13237	5089
Kmeans	1582689	34707	0	2092	0	27	530
Yada	711347	129899	46	142981	0	3461	634

Table 3.1: Number of each dvSRAM operation for tested applications of STAMP benchmark suite (Genome, Intruder, Kmeans, Yada) for one core.

Operation	Activated parts of a dvSRAM cells
Read	Main
Write	Main
dvRead	Secondary
dvWrite	Secondary
dvBWrite	Main, Secondary
dvStr	Main, Secondary, Exchange circuit
dvRstr	Main, Secondary, Exchange circuit
dvStrAll	Main, Secondary, Exchange circuit

Table 3.2: Activated parts of a dvSRAM cell for each operation.

	P(main)	P(secondary)	P(exchange)
Gnome	0.9999	0.0170	0.0011
Intruder	0.9999	0.1498	0.0672
Kmeans	1	0.0016	0.0003
Yada	0.9999	0.1488	0.0041

Table 3.3: The probability of access to each part of dvSRAM cells for tested applications of the STAMP benchmark suite at each execution time.

the number of accesses to the exchange circuits is calculated by adding up the number of the following operations: dvStr, dvRstr, and dvStrAll. The probability of each part activation is then calculated by dividing the above numbers by the total number of operations for each tested benchmark as can be seen in Table 3.3.

It is observed that the probability of main cells activation is much higher than the two others (secondary cells and exchange circuits). On average, probability of activation for main, secondary and exchange circuits is 0.9999, 0.0637, and 0.0146 respectively. The secondary cells have to retain the data when accessing the main cells, which leads to substantial leakage power consumption; so if we want to use the benefits of dvSRAM

cells in optimistic concurrency proposals, deploying some leakage control techniques that specifically target the secondary cells leakage current is vital.

3.5 DVSRAM-D array structure design

In this section, we add our proposed bias control circuit to a typical dvSRAM array (which we explain its detail in Chapter 2) and show the block diagram of dvSRAM-D. As we explained in Section 3.4, for each operation different parts of dvSRAM cell are active so we exert different drowsy mode techniques for each part. The DB-Control circuit is suitable for controlling bias voltage of cells (main and secondary cells), so we design two DB-Control circuits to generate virtual VSS (VSS1 and VSS2) and connect the VSS of all main cells and secondary cells of one sub-array to VSS1 and VSS2 respectively. Furthermore, SAB-Control circuit is acceptable for controlling bias voltage of exchange circuits, so we design a SAB-Control circuit to generate virtual VDD and VSS (VDD3 and VSS3), then we connect the VDD and VSS of all exchange circuits of one sub-array to nodes VDD3 and VSS3 respectively.

We consider that all the parts are in their drowsy mode unless related operations activate them. Main cells (secondary cells) are in their drowsy mode when none of them is active; at this time, VSS1 (VSS2) rises to the data retention voltage level that can be regulated with the DB-Control circuit. The exchange circuits are in their drowsy mode as well when all of them are inactive, but in this case, both of the VDD3 and VSS3 are adjusted with the SAB-Control circuit.

Figure 3.3 depicts the block diagram of a dvSRAM-D sub-array with data-in and data-out drivers, word-line address drivers and our proposed bias control circuits. The central current mirror circuits (designed for VSS1 and VSS2) are in the middle part of the array nearby decoders and control signal generators. The intermediate current mirror circuits are in the path to receive to the sub-array. Bias control circuits available for VDD3, VSS1, VSS2, and VSS3 generation can be seen in the upper and lower parts of the sub-array.

3.6 Evaluation

We simulate and optimize a 32KB typical dvSRAM array (dvSRAM-T) and a 32KB dvSRAM array including the proposed bias control circuits (dvSRAM-D) with Hspice 2003.03 using 45nm Predictive Technology Model [21], VDD=1V and T=25°C at the 2GHz processor clock frequency. We calculate the active energy per operation and static energy for both cases and show them in Table 3.4. For each operation, only the necessary parts of the SRAM cells are activated, and the other parts remain in drowsy mode as shown in Table 3.2. As can be seen in Table 3.4, the energy reduction for each operation depends on the number of active parts. It means that our proposed techniques have a considerable effect on energy reduction for Read, Write, dvRead and dvWrite compared to other operations. Obviously, the energy consumption reduction for dvStr and dvRstr is less than the other operations.

Also, we construct a dvSRAM-D transistor level net-list, but in this case, we replace the two DB-Control circuits that generate VSS1 and VSS2 with actively clamped sleep transistors [15] designed based on self-biased complementary folded cascade Opamps [56]. However, we do not move the SAB-Control circuits that generate VDD3 and VSS3 (because we want to keep the instability under control). Although, the energy consumption of added circuits is small, we calculate the energy consumption of this setup for each operation and compare it with our proposal. The results indicate that the average overhead energy in our proposed technique is 59% lower than the average overhead energy consumption in the structure using actively clamped sleep transistors [15].

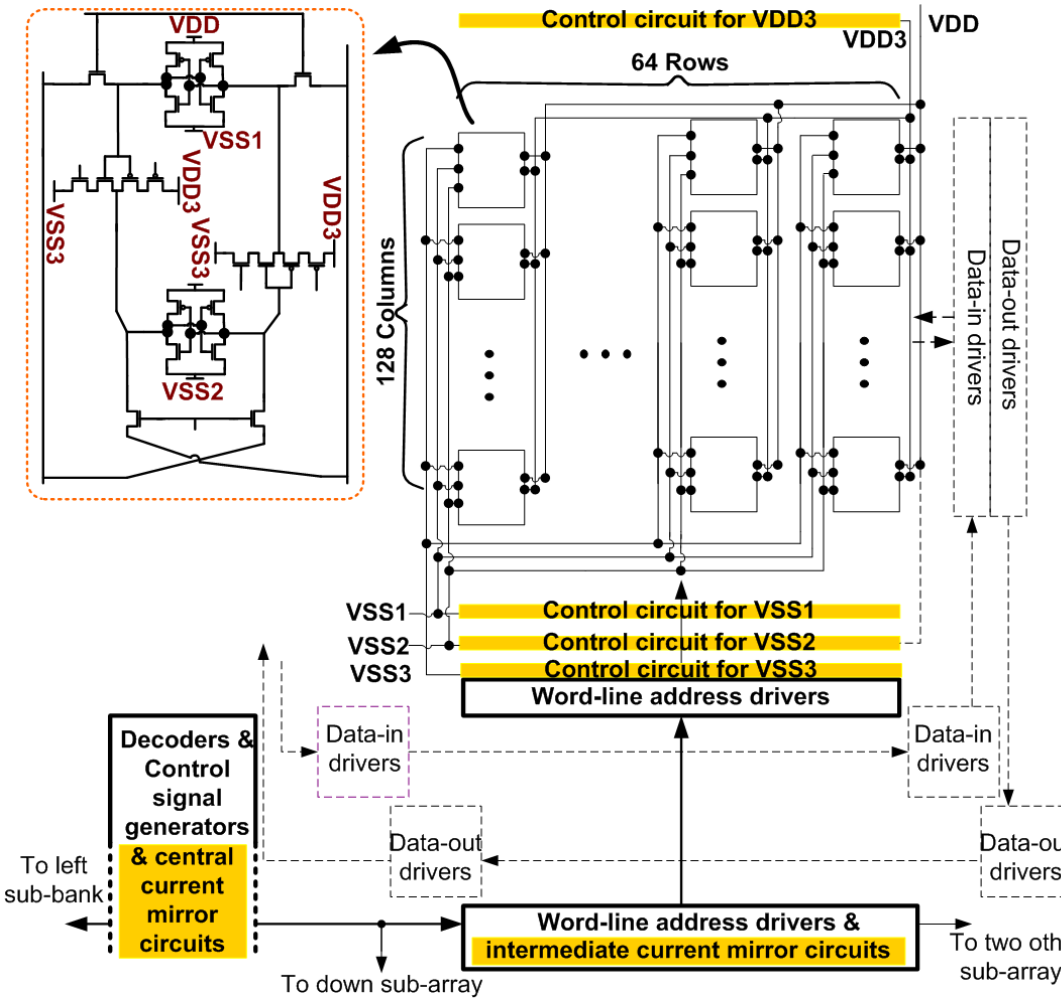


Figure 3.3: The block diagram of a dvSRAM-D sub-array. The proposed circuits are highlighted.

Operation	Energy per operation (pJ/operation)	
	dvSRAM-T	dvSRAM-D
Read	112.4	59.1
Write	99.3	57.6
dvRead	114.4	63.1
dvWrite	101.1	59.0
dvBwrite	122.6	81.9
dvStr	114.9	84.6
dvRstr	112.9	84.7
dvStrAll	1123.2	819.9
Static	51.4	34.7

Table 3.4: Energy consumption per operation for dvSRAM-T & dvSRAM-D.

Figure 3.4 shows normalized energy consumption for the tested applications of the STAMP benchmark suite using dvSRAM-T and dvSRAM-D variants. The energy consumption reduction is 37.94% for Genome, 35.68% for Intruder, 35.09% for

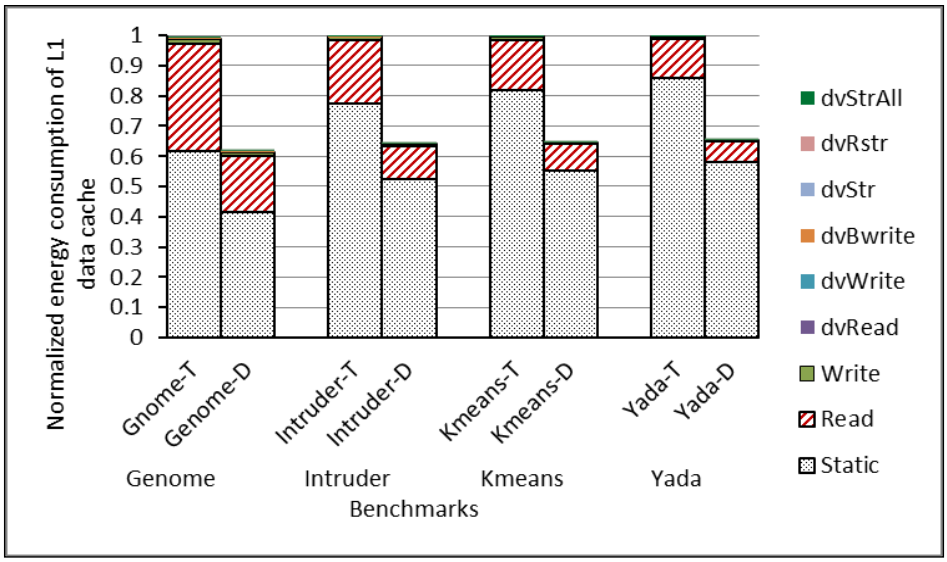


Figure 3.4: Normalized energy consumption of dvSRAM L1 data cache for tested applications of STAMP benchmark suite for dvSRAM-T & dvSRAM-D.

Kmeans, and 34.49% for Yada. Also, we estimate the area overhead occupied by necessary bias control circuits [57] and add it to the total area. The area increase is 1.6% per sub-array, and it is lower than in previous techniques, which required around 3% increase per sub-array [15].

Figure 3.5 shows the simulation waveforms of a random sequence of eleven operations. The simulation considers the first cell of the last row of a sub-array and takes a total of 15ns. The dvSRAM-D is designed to work at a two clock cycle access time (1ns), but an extra clock cycle is necessary for operations that adjust the signals VDD3, VSS1, VSS2, and VSS3 to change the mode. We calculate the average time to reach stable levels (low to high and high to low) as 0.34ns for VDD3, 0.47ns for VSS1, 0.45ns for VSS2 and 0.13ns for VSS3. It is important to note that operations that need an additional cycle are rare because the number of accesses that only use the main cells is much larger than the accesses that make use of the exchange circuits and/or secondary cells, which moreover usually happen in small bursts when executing speculatively for optimistic concurrency (in the case of HTM, this case happens inside a so-called “atomic” region). Thus, the timing overheads incurred by this additional cycle can be considered negligible, because as shown in Table 3.3, the probability of accessing to the main cells is dominant.

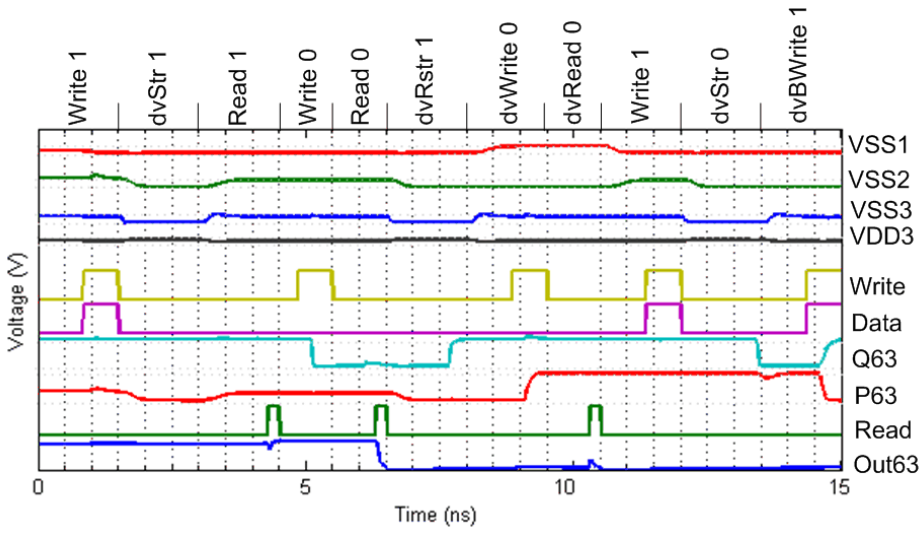


Figure 3.5: Simulation waveform for a sequence of eleven operations for the first cell of the last raw of a sub-array. The operations are shown on top. Max and min voltage levels for VSS1 (VSS2) are 0.3V and 0.05V, for VSS3 are 0.2V and 0.025V and for VDD3 are 0.910V and 0.85V respectively.

3.7 Instability problems of un-accessed cells

In this section, we discuss an important instability problem that may occur in a dvSRAM array during execution time and show how our proposed SAB-Control circuit alleviates it. As mentioned in Section 3.4, the main cells are active much more often than the secondary cells. Main and secondary cells are connected to each other via exchange circuits; thus they are not completely separated, so we should pay attention to any undesired changes in the preserved data of secondary cells while they are not accessed. This can be solved by our proposed noise sensitive bias control circuit that we apply to dvSRAM array.

We explain the sensitivity problem with a simple example: Assume P=1 and that WL-En2 is low for some cycles in Figure 2.1 (Chapter 2). When dvStr and dvRstr are low, the data is retained in the secondary cell. Also, assume Q=1 and that WL-En1 is low for some cycles too; hence the gate and the drain of transistor MNS1 are high. Any noise injection may increase the subthreshold leakage current of MNS2 and discharge node P

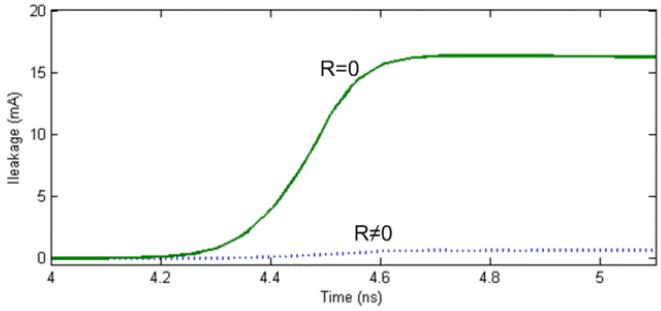


Figure 3.6: The leakage current passes through the transistors MNS1 and MNS2 when exchange circuits are un-accessed for R1=0 and R1≠0.

and change the secondary and then the main cell values (remember WL-En1 and WL-En2 are low for some cycles).

In the SAB-Control circuit, that additional leakage current of MNS2 increases virtual VSS (VSS3) voltage level (the voltage drop across R1), helping to reduce the additional leakage current that passes through MNS2 and decreases the probability of changing the saved data. Figure 3.6 shows the leakage current passing through transistors MNS1 and MNS2 in two modes; when R1=0 and when R1≠0. It demonstrates that R1 has a considerable effect in leakage current reduction. Similarly, when PB=0 and QB=0 for some cycles, our proposed SAB-Control circuit alleviates the instability problems.

3.8 Summary

In this thesis chapter, we propose two novel bias control circuits, DB-Control circuit and SAB-Control circuit, and as a use case we apply them to our recently proposed dual-versioning L1 data cache to manage the energy consumption of un-selected SRAM cells. We present the design details and calculate the energy consumption for each operation and the total energy consumption for tested benchmarks. Our simulations demonstrate the usefulness of our proposed circuits to reduce the energy consumption with acceptable area increase and negligible delay overhead. We also show how our proposed techniques can reduce the instability problems that might appear, especially in drowsy mode.

Chapter 4

Circuit Design of a Novel Adaptable and Reliable L1 Data Cache

4.1 Introduction

As energy becomes one of the key design concerns for computer systems, a dramatic improvement in the energy efficiency of microprocessors is required in order to keep the power under control. A very effective approach in reducing the energy consumption is to reduce the supply voltage (Vcc) close to the transistor's threshold. However, the energy reduction in the low-power mode comes with a drastic increase in the number of memory cell failures especially in large memory structures such as on-chip SRAM memories (such as L1 and L2 caches) [8], [9]. This motivates us to design a cache which is resilient to large number of cell failures and operates at lower supply voltages.

These memory failures can be persistent (i.e. yield loss or hard errors) or non-persistent (i.e. soft errors or erratic bits) while rates of both failures increase as the Vcc is decreased. Moreover transistor scaling increases the vulnerability of transistors to radiation events since it increases the likelihood of having multibit soft errors on adjacent bits [16]. Thus, it is essential to implement reliability solutions addressing both persistent and non-persistent failures in caches in order to reduce the Vcc and provide reliable cache operation for future technology nodes.

There are two main techniques to deal with high fault rates stemming from the above issues: 1) Coding techniques such as parity or ECC, 2) In-cache replication. While they are effective, both mechanisms have issues. Error Correction Codes (ECCs) are the most widely used techniques for detecting and correcting both persistent and non-persistent failures with additional area, power and encoding/decoding time overhead [58], [59], [60]. However, the increase in the error correction capability of ECC is much lower than the increase in power and area consumption. For example, in 8-byte data, correcting a double-bit error costs 19% area overhead while three-bit error correction requires a stronger and a more complex ECC with 100% area overhead [16]. Intel's latest 22nm 15-core Xeon processor uses Double Error Correction, Triple Error Detection (DECTED), a very strong ECC, for its L3 cache data array; however, the computational cost of DECTED ECC impacts the L3 data accesses, whose latency is variable, thus

significantly complicating the micro-architecture [61]. Due to the diminishing benefits of stronger ECCs, providing reliability in an environment with a very high fault rate such as when the processor is operating in a very low power mode, is not trivial. Thus, only a few ECC solutions address large-scale multibit errors in a line [58], [60]. However, they require a complex encoder/decoder with a high energy consumption which diminishes the energy saving potential of the low-power mode execution.

The second mechanism, in-cache replication such as triplication, is a conventional way of providing high reliability with minimum fault recovery latency. More specifically, triplication approach not only detects errors but also has the ability to correct them. This approach triplicates the data, executes the operation on each of them simultaneously and passes the results to a voter, which decides the correct result that will be passed on [62], [63]. At each access, there are three simultaneously active modules, instead of a single module; which leads to considerable area and energy overhead. However, replication schemes have two main problems: (1) Writing/reading more than one cache line increases access latency and energy consumption. (2) When processors operate with a very low Vcc, the number of uncorrectable lines increases due to the multiple failures in the same bit-position.

In this chapter, we design a level one cache which can tolerate very high bit failure rates of ultra-low voltage execution with minimum overhead, and without harming the cache capacity in the nominal mode. To this end, we present Flexicache, a new cache design which avoids the problematic aspects of coding and in-cache replication through a two-tiered approach [64], [65]. First, Flexicache proposes a circuit-driven solution that duplicates/triplicates all the available cache lines and achieves read/write accesses to multiple lines without increasing access latency and with a minimum increase in the access energy. Second, Flexicache divides each cache line into single-parity-protected partitions to increase the error correction capability of replication schemes.

Flexicache automatically adapts itself for different supply voltages in order to provide as much reliability as possible with minimum energy consumption and access time. It operates in three modes: (1) Single Version Mode (SVM), (2) Double Version Mode

(DVM) or (3) Triple Version Mode (TVM). When the supply voltage is relatively high, i.e. the nominal voltage, Flexicache operates in SVM to satisfy high performance execution. In this state, Flexicache provides reliability based only on single-bit interleaved parity similar to many typical L1 caches. DVM is utilized in medium low supply voltage ranges (low voltage but not near threshold) in which error rate starts to increase. Flexicache writes data to two selected cache-lines simultaneously. At every read instance, Flexicache compares the values from both cache lines through XOR circuits to check their equality. In this approach, the utilization of parity bits enables Flexicache to decide the correct value upon an error. When the supply voltage is at near-threshold and the error rate increases drastically, Flexicache operates in TVM. Flexicache writes data to three selected cache-lines simultaneously to provide very high reliability. At every read instance, the correct value is decided through bit-wise majority voters similar to conventional triplication technique but the error correction capability is improved by parity bits participation in majority voting. The main contributions of this study are the following:

-We design and simulate a 64-KB a novel, reliable L1 data cache (Flexicache) with 45-nm Predictive Technology Model [22] at 2GHz processor frequency, which configures itself for different supply voltages from the nominal to the near threshold voltage levels in order to duplicate or triplicate each data line when higher reliability is required.

-Flexicache provides significantly higher cache capacity with less error correction energy compared to OLSC [58] and conventional triplication.

-Flexicache allows cache operating down to 320 mV (10% failure rate) by presenting, on average, 63% energy reduction in cache operations. The area overhead of Flexicache is only 12% compared to a typical L1 cache.

4.2 Background and Related Work

In this section, we first explain the nomenclatures of failures in memory structures. Then we present the previous schemes used for scaling Vcc.

4.2.1 Memory Failures

Bit failures are classified into two broad categories [58]: persistent failures and non-persistent failures. Here we explain these two categories.

4.2.1.1 Persistent Failures

The random variation in the number and location of dopant atoms in the channel region of the device leads to the random variations in transistor threshold voltage. It causes threshold voltage mismatch between the transistors close to each other. In a SRAM cell, a mismatch in the strength between the neighbouring transistors caused by intra-die variations can result in the failure of the cell [66]. A cell failure can occur due to: (1) An increase in the cell access time, (2) unstable read operation, (3) unstable write operation, (4) failure in the data holding capability of the cell. Further details can be found in [67].

On the other side, open or short circuits cause irreversible physical changes in the semiconductor devices. These permanent failures tend to occur early in the processor lifetime due to manufacturing faults (called the infant mortality), or late in the lifetime due to thermal and process related stress. The location of a persistent failure is random and independent of whether the neighboring bit is faulty or not [68]. The locations of persistently defective bits can be detected by performing built-in self-test (BIST) [69]. Figure 4.1 shows the mapping between the bit failure rate and the Vcc calculated for 32nm technology (Miller et al [60]). As can be seen here, the bit failure rate increases exponentially whenever Vcc is lowered. We refer to this figure for our evaluations.

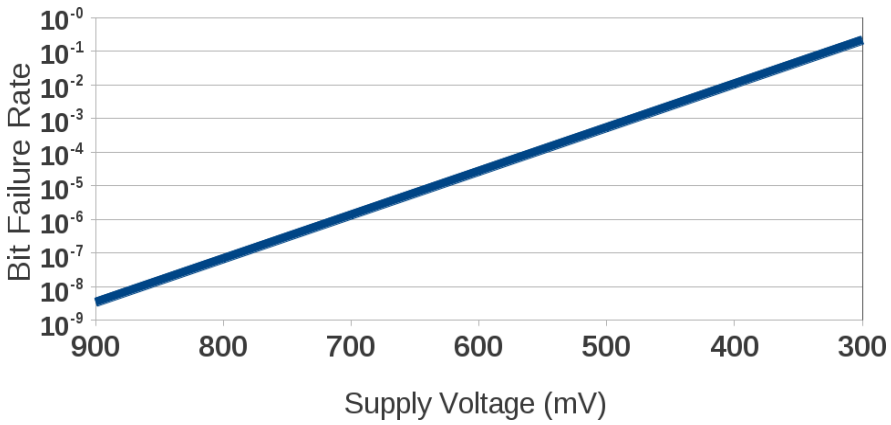


Figure 4.1: Bit Failure Rate vs Supply Voltage

4.2.1.2 Non-Persistent Failures

Radiation events or power supply noise can cause a bit flip and corrupt a data stored in a device until a new data is written [70]. As transistor dimensions and operating voltages shrink, sensitivity to radiation events increases drastically. On the other side, process variation or in-progress wear-out, combined with voltage and temperature fluctuations might cause correlated faults of short duration. They are termed intermittent faults (or erratic failures), that last from several cycles to several seconds [71]. Diagnosing an intermittent fault by BIST is hard since it does not persist and conditions that cause the fault are hard to regenerate. As Vcc decreases, the bit failure rate increases rapidly for both intermittent faults and persistent failures [60], [58].

4.2.2 Related Work

In this section, we discuss architecture-based error correction schemes utilized under scaling voltage and compare their main characteristics with Flexicache in Table 4.1.

	Segmented ECC	2D ECC	Disabling/ Bit-Fix	Flexicache
PersistentFailures	**Yes**	**Yes**	**Yes**	**Yes**
Non-Persistent Failures	**Yes**	**Yes**	No	**Yes**
Minimum Vcc	375 mV	-	400 mV	**320 mV**
Latency in the Low-Power Mode	1 cycle	1 cycle	**0 cycle**	1 cycle
Other Latency	**No**	read-modify –write	**No**	**No**

Table 4.1: Comparison of Flexicache with Architecture Based Error Correction Schemes for Scaling Vcc (Bold is better).

Orthogonal Latin Square Code (OLSC) [59] is a state of the art ECC scheme used for level-one caches when the supply voltage is lower than the safe margin. OLSC which is designed based on the orthogonal latin squares concept [59] can be decoded using majority voting. An OLSC encodes orthogonal groups of bits to form check bits. The final value is generated by each data bit through a voting process and from a group of orthogonally coded data

and check bits [17]. Multi-Bit Segmented ECC (MS-ECC) [58] utilizes OLSC at a finer granularity in order to increase the error correction capability of OLSC to be used for ultra-low voltage level. Thus MS-ECC can reduce the supply voltage until 350 mV in 35nm technology by providing 6.5% useful cache capacity. We define useful cache capacity as the portion of the cache which is not disabled [60].

Kim, et al. [72], propose two-dimensional (2D) ECC to correct multi-bit errors with a minimum area overhead in check bits. However, the correction capability of this scheme is strongly dependent upon the location of defective bits; for this reason, it is not convenient to use it in low-power mode when failures are random. Also, it requires a read-modify-write operation for all stores and for every cache miss which increases the delay and power consumed by all write operations.

Miller et al. [60] propose Parichute which utilizes Turbocodes for reducing Vcc of the second and higher level caches. Although this scheme provides a very high error correction rate supporting a voltage reduction significantly, its error correction latency can be more than 5 cycles [60] in the near-threshold voltage level. Thus, Parichute is not convenient to be used in time-critical L1 caches.

Several disabling schemes have been proposed for tolerating only persistent failures [67], [73], [66]. Wilkerson et al. [67] disables the faulty words in order to combine two consecutive cache lines to form a single cache line where only non-failing words are used. Although the area overhead of word-disable in high-power mode is only 8%, in the low power mode the available cache size shrinks to half when the error rate is lower than 0.01%.

Abella, et al. [73] propose to disable sub-blocks instead of words in order to utilize more capacity in the low-power mode. Both disabling schemes need to access a fault map in parallel. Zereh Cache [66] employs fine granularity re-mapping of faulty bits and relies on solving a graph coloring problem to find the best re-mapping of defects to spare lines. Bit-fix [67] stores the location of defective bits and their correct values to the quarter of cache ways.

Besides architectural approaches, circuit-based approaches have also been proposed such as using 8T SRAM cells [72] and 10T SRAM cells [74]. 8T SRAM cell is more stable against parameter variations than 6T cell, and well suited for noisy ambient. However, in spite of its effectiveness for low Vdd modes, it presents high area overhead in the nominal voltage. Kulkarni et al. [68] propose a Schmidt-trigger based 10T SRAM cell with inherent tolerance towards process variation using a feedback-based mechanism. However, this SRAM cell requires a 100% increase in area and about 42% increase in access time for nominal voltage operation.

In this thesis chapter, we propose Flexicache, a circuit-driven solution that duplicates or triplicates all the available lines in the cache with no increase in the access latency. In this study, besides elaborating the circuit design, we present the details of the address decoder and the architectural extensions of Flexicache.

4.3 Architecture of Flexicache

Flexicache allows three modes of error protection according to the resilience level of the applied Vcc: Single Copy State (SVM), Double Copy State (DVM) and Triple Copy State (TVM). Flexicache divides each cache line into parity protected-partitions akin to many commercial L1 caches protected by single-bit parity in block, word or byte granularity [75]. Figure 4.2 presents the design of DVM and TVM for a hypothetical 8-bit partition. SVM, which is not presented in the figure, provides reliability solely based on single bit interleaved parity. In this study, Flexicache runs in SVM in the nominal voltage when the failure rate is minimal in order to provide full cache capacity. Note that instead of parity, a stronger code can also be utilized to provide a higher reliability for mission critical applications.

Flexicache runs in DVM when the Vcc is medium-low and writes data to two cache lines. Note that the circuit design allows writing/reading multiple lines simultaneously (i.e. without increasing the access time) as we explain in the following section. In a read, DVM compares two duplicated, parity-protected partitions through the XORs to check if there is any fault. In case of equality, Flexicache dispatches one of the partitions to the output buffer. Otherwise, Flexicache calculates the parity of each partition and sends out

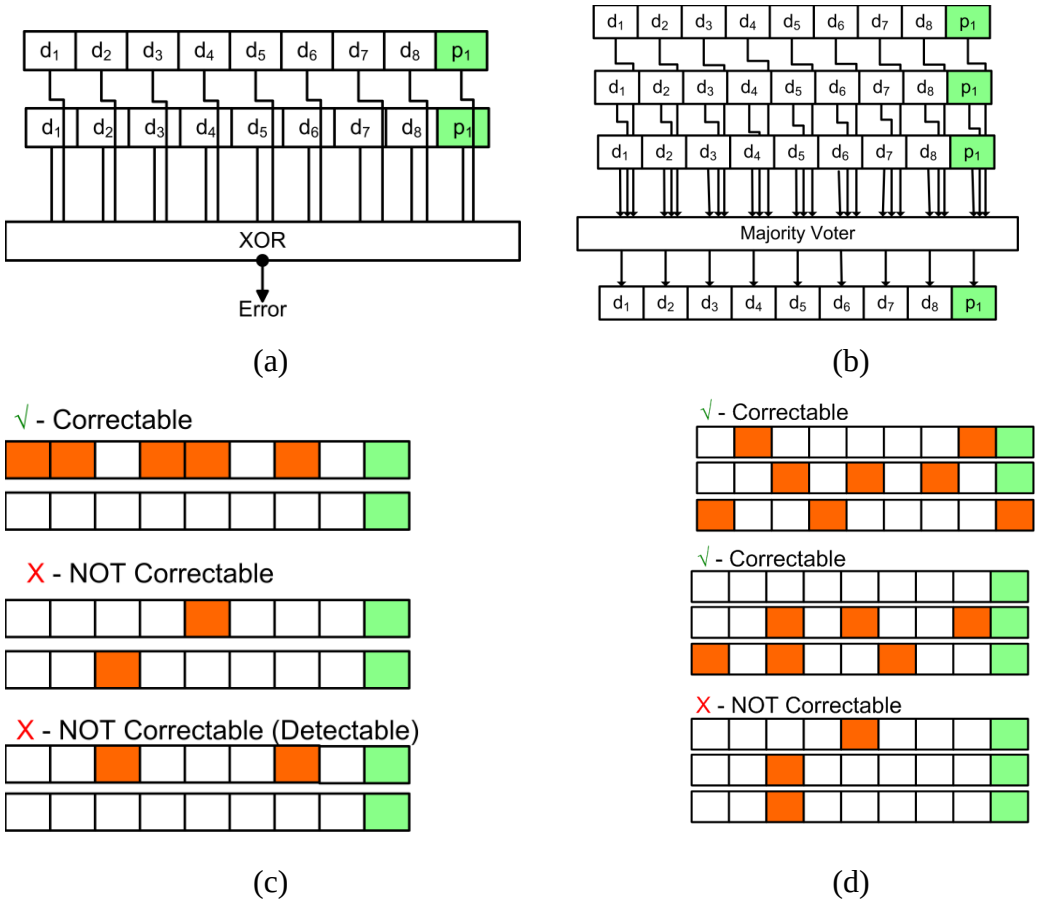


Figure 4.2: The figure presents the behavior of Flexicache for DVM (a), and TVM (b) for 8-bit partitions. Also, it presents examples for correctable and non-correctable faults (c and d).

the partition which has the correct parity. DVM provides a backup copy for each partition. For instance, when a particle strike affects several adjacent bits in a line, the correct value is read from its replica without requiring any decode and correct time. In order to decrease the probability of a strike affecting both coupled lines, Flexicache couples the lines with spatially distant locations. (e.g 0th and 62th lines.)

When the Vcc is near threshold, in order to tolerate the drastically increased error rate, Flexicache runs in TVM by writing the data to three cache lines simultaneously. On a read, Flexicache uses bitwise majority voting to obtain the correct data and calculates the parity of the data. Unless parity confirms that the result is correct, Flexicache calculates the parities of three partitions and sends out the correct partition. In TVM, the whole cache should be divided into three which is not trivial for a cache having 2^n lines. One

solution can be manually connecting lines by taking into account that the lines in the same group should be in distinct positions (e.g. 0th, 42th and 84th lines for a 128-line cache). However, this considerably increases the complexity of the address decoder. Instead, we add spare lines to make the cache dividable into three. Note that using spare lines for tolerating yield loss is a common approach [66], [76] and the area overhead due to the extra lines is similar to this approach.

DVM can correct an odd number of errors if they affect only one copy of the data (Figure 4.2c). However, if the faults are in different copies of the data, DVM can only detect the bit-positions of the faults without correcting them. Similarly, if there is an even number of faults in one partition DVM cannot correct them, either. TVM (Figure 4.2d), on the other hand, can correct errors easily unless they are affecting the bits in the same position (which has a significant possibility in very high bit failure rate). Otherwise, after calculating the parity, TVM detects that the result of majority voter is not correct, and it can correct errors if one of the three copies is error-free. If all three copies are erroneous, and some errors are in the same bit position, TVM cannot correct the partition.

When there is an uncorrectable partition in a line, we utilize a partition-x mechanism in DVM and TVM to avoid wasting the correct partitions. Partition-x is similar to the bit-x proposed by Wilkerson et al [67]. It uses a quarter of the cache ways to store locations and the correct values of defective partitions. This reduces both the cache size and associativity in the low-power mode. Thus, we utilize partition-x mechanism only for the lines which have uncorrectable partitions. Note that, our partition-x mechanism is different from the bit-x for a non-persistent bit failure correction. In bit-x, the cache lines are not protected by any other means, they only rely on memory tests and fixing the detected failures. In Flexicache, the fixed partitions are also protected by DVM or TVM which can still correct non-persistent failures.

Previous triplication schemes [62], [63] write data to three cache lines and read the correct value from the majority voter. In Flexicache, partitioning and parity protection of each partition present higher error correction capability. Persistent-fault tolerating

proposals perform BIST [69] either post-manufacturing or at boot time to determine the uncorrectable cache lines at each voltage level [73], [67], [60]. These lines are stored in on-chip ROM or main memory and loaded before the processor transitions into near-threshold. For non-persistent failures, if the system cannot correct a fault in L1 cache, either the correct value is re-fetched from L2 cache if the write-through cache is utilized or the system issues a machine check exception unless other means are utilized. Flexicache performs BIST test as in previous proposals to determine faulty partitions in order to x them or disable the cache ways/lines including them. At runtime, Flexicache can detect and correct non-persistent failures, as well. For uncorrected non-persistent failures, Flexicache can utilize lightweight, global checkpointing such as SafetyNet [77].

4.4 Circuit Design of Flexicache

In this section we describe the block diagram of level one data Flexicache and explain the details of each sub-array and address decoder circuits. Even though we propose the circuit design for the level-one data cache, we also suggest, given its characteristics, to apply it for the level-one instruction cache or the level-two cache.

4.4.1 Block Diagram of Flexicache

Our sub-bank design is based on Cacti suggestions and the structure that we propose in Chapter 2. In Chapter 2, we propose a novel level-one data cache design with dual versioning SRAM cells (dvSRAM) where each dvSRAM cell keeps two versions of the same data. Also in Chapter 2, we present a reconfigurable cache which can switch between a 64KB general purpose data cache and a 32KB special purpose data cache with a little decoder circuit modification. The proposed control circuits and word-line buffers give this opportunity to access to two adjacent cache-lines at each access time with acceptable energy consumption overhead. Our design in this chapter is inspired from those proposed structures; Flexicache needs to access replicated data within the cache access time with minimum energy. Also, Flexicache requires a reconfigurable cache design so that it can provide three different execution modes (i.e. SVM, DVM, TVM) to not sacrifice the cache capacity in the high-performance execution mode.

In this section, we elaborate how we can design the circuit of Flexicache cache for a level-one data cache so that it can replicate cache lines without increasing access latency and with minimal energy overhead. Note that it is straightforward to extend the design for the instruction cache and the Level-two cache. Flexicache can also be designed orthogonally to dvSRAM so that it can support both optimistic concurrency and near-threshold voltage execution; we leave it out of the scope of this study.

Figure 4.3 shows the block diagram of Flexicache. We use Cacti [28] to determine the optimal number and size of Flexicache components. For the level-one data cache configuration we assume 64KB, four-way, 64B lines, and two clock cycle access time. For a one bank array, Cacti suggests two identical sub-banks, one mat for each sub-bank and four sub-arrays in each mat. We divide each sub-array to eight equal slices each containing 16 sub-array lines; each slice has its own pre-charged circuit, write circuit, sense amplifier circuit and input and output buffers (64KB = 4 bank×8 sub-bank/bank×8 slice/sub-bank×16 lines/slice×128 bits/lines). Also, we extend each sub-array with extra slices (one for each sub-array) to be able to make it divisible by three for Triple Copy State (TVM); these extra slices are identical to other slices.

Considering the cache structure of our system, we need eight address bits to address a line. Also, to address the extra slice lines, we need one extra bit; so those eight bits, this extra bit and some necessary control signal bits enter to the up side of sub-banks. 512 data-in and 512 data-out bits are routed from the up side as well.

During an access, only one of the two sub-banks is activated; hence the number of bits that produce word-line addresses is seven. The address decoder and control signal generator units are placed in the middle part of the array. 128+16 word-line addresses and 13 control signals are generated in the middle part of the array and enter to the sub-bank from the left side as shown in Figure 4.3. The four identical sub-arrays of a mat are activated during an access; therefore each sub-array holds a part of the cache-line, so 128 of the 512 data bits (64B) are distributed to each sub-array. We place necessary optimized drivers (chain of two series inverters) in paths to reach their related loads in the sub-arrays. We show the necessary address and data drivers only for one sub-array in the figure. Eight interleaved parity-bits extend each sub-array line to detect odd numbers of failures (each interleaved parity-bit is the XOR of 16 non-adjacent bits) [78].

Necessary circuits to generate these parity bits are located at the bottom of each sub-array slice.

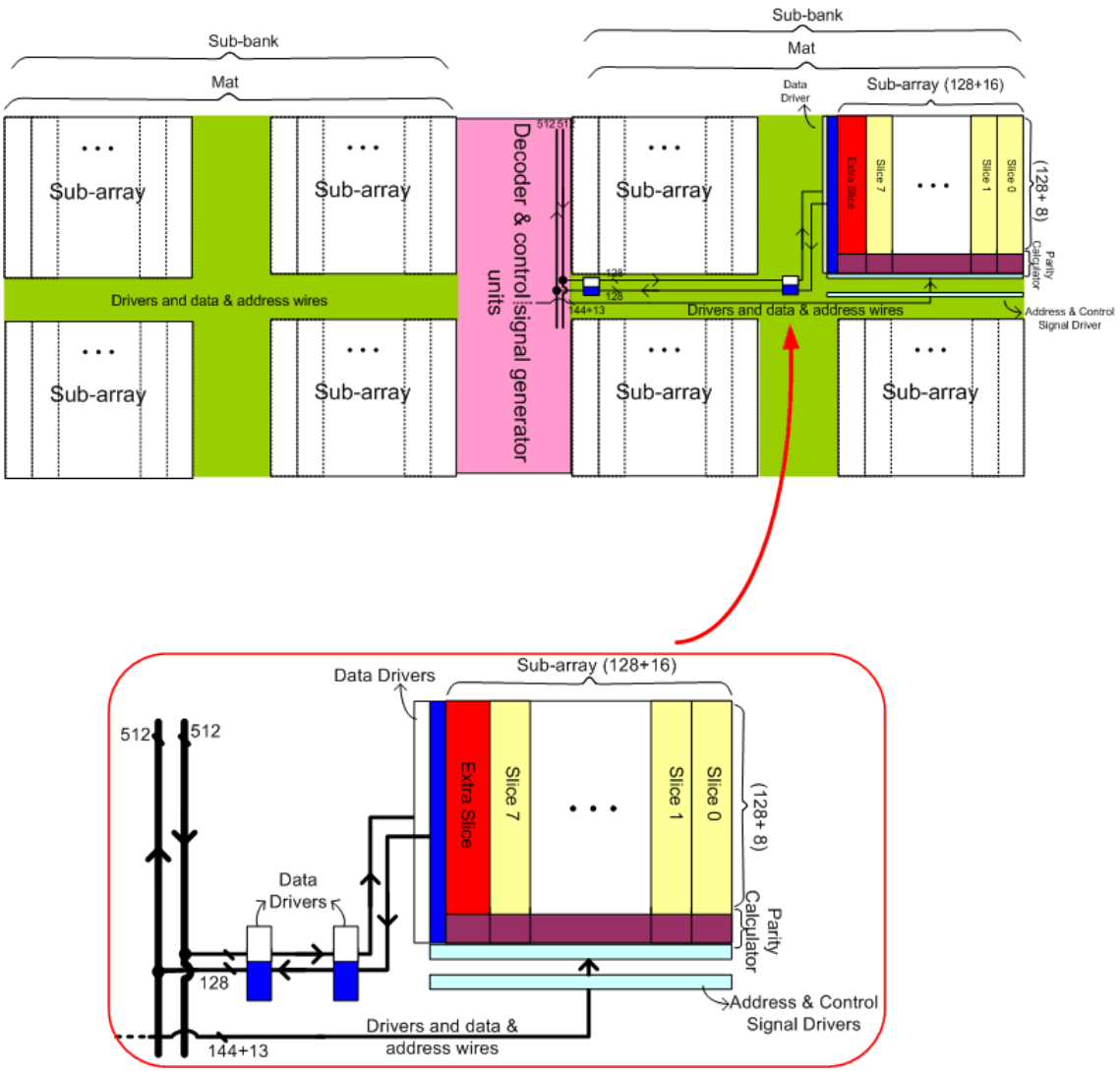


Figure 4.3: Flexicache block diagram

4.4.2 Circuit Details of Each Flexicache Sub-Array

Figure 4.4 presents the main components of Flexicache for one sub-array. According to the decoded addresses and the Vcc level, one, two or three slice(s) are activated and the data coming from the bus is written to the enabled slice(s). Figure 4.5 shows the complete details for the block diagram of each sub-array structure. The figure presents the necessary buffers, comparators, parity calculators and control and data lines in detail. We describe the details for each TVM, DVM and SVM mode in this section.

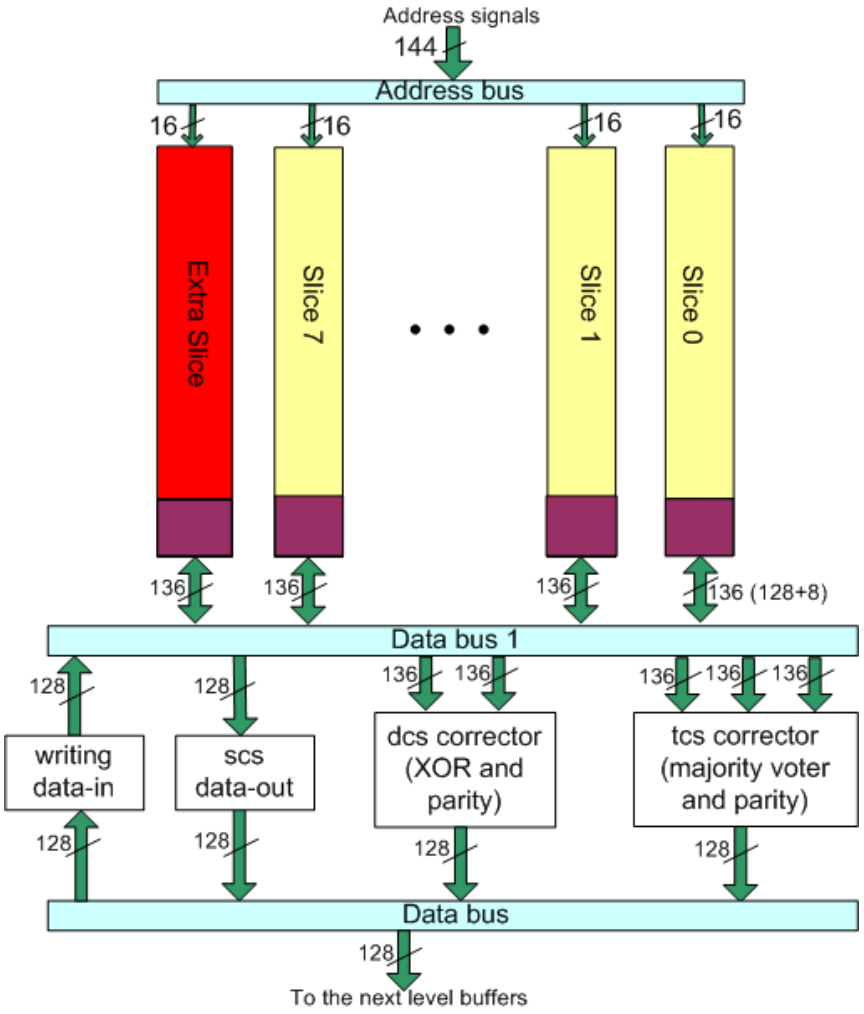


Figure 4.4: The figure presents the main components of Flexicache such as buses and decoder.

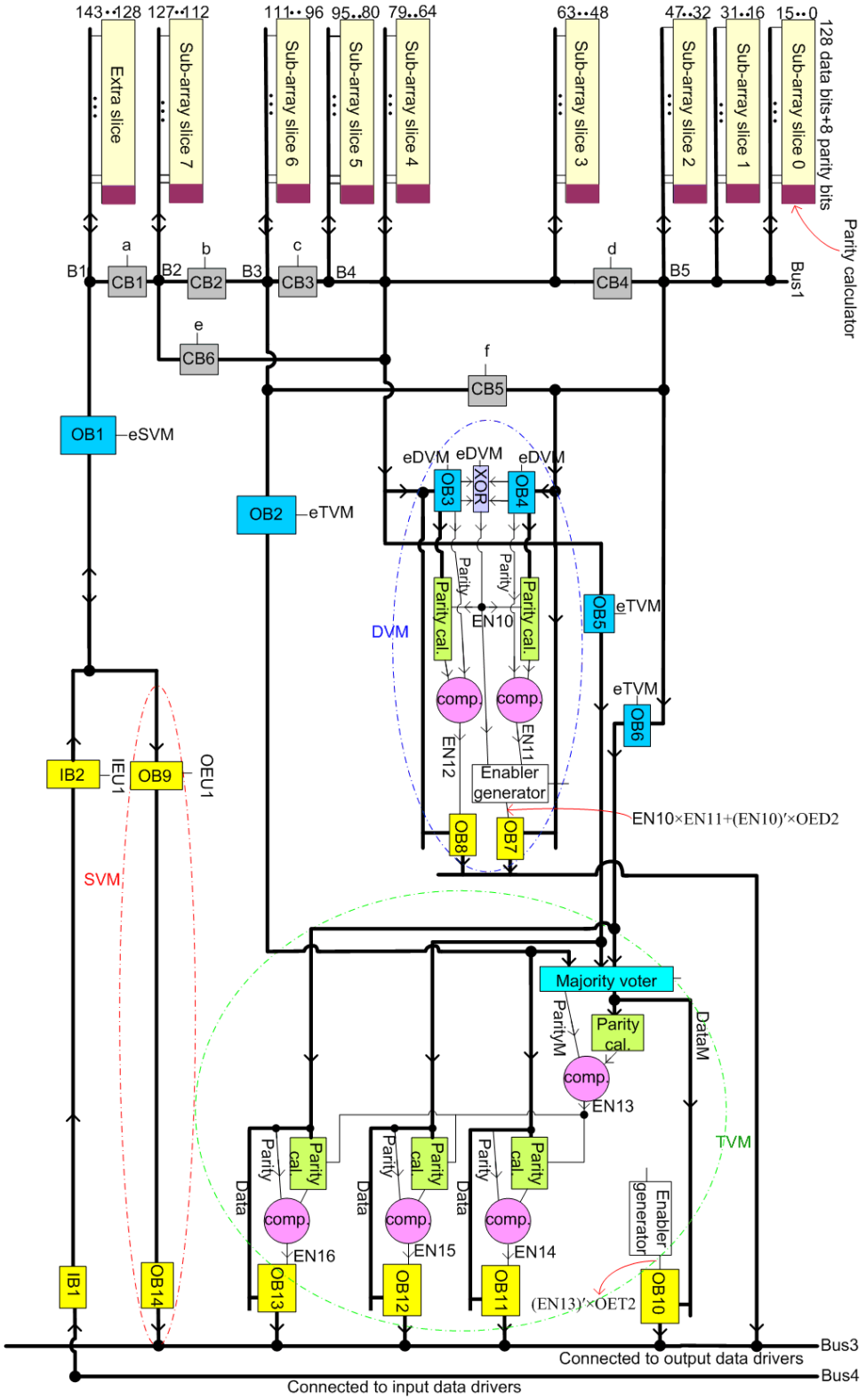


Figure 4.5: Sub-array block diagram; sub-arrays are much bigger but for clarity we show like that.

4.4.3 Circuit Details in SVM

Figure 4.6 shows the block diagram of one sub-array during writing the data. For writing the selected cache-line, signal IEU1 is high and activates input buffers IB1 and IB2 and data can be transferred to the selected cache-line via Bus4 and Bus1. Bus4 is connected to input data drivers which are located close to each sub-array. Furthermore, Figure 4.7 shows the block diagram of a read access to one sub-array when being in SVM. In SVM, only one cache-line is activated at each access time. For reading the selected line, signal OEU1 is high and output buffers OB9 and OB14 are active and data is transferred from Bus1 to Bus3. Bus3 is connected to output data drivers which are located close to each sub-array. For both figures, at each access time, the enabler signals (a, b, c, d, e, f) are high and activate connector buffers, CB1, CB2, CB3, CB4, CB5 and CB6 in order to connect nodes B1, B2, B3, B4 and B5 to each other (Each connector buffer contains two series inverters with enablers).

Similar to many typical level-one caches, its error protection is based on bit-parity calculation in order to achieve higher speed but less accuracy. We divide each cache-line into 8 partitions; each contains 16 bits where each interleaved parity protects one partition. At each reading time the parity bit is calculated and compared with the original parity bit. The access time for SVM is two clock cycles. Whenever an error is detected the line is fetched from the higher level caches.

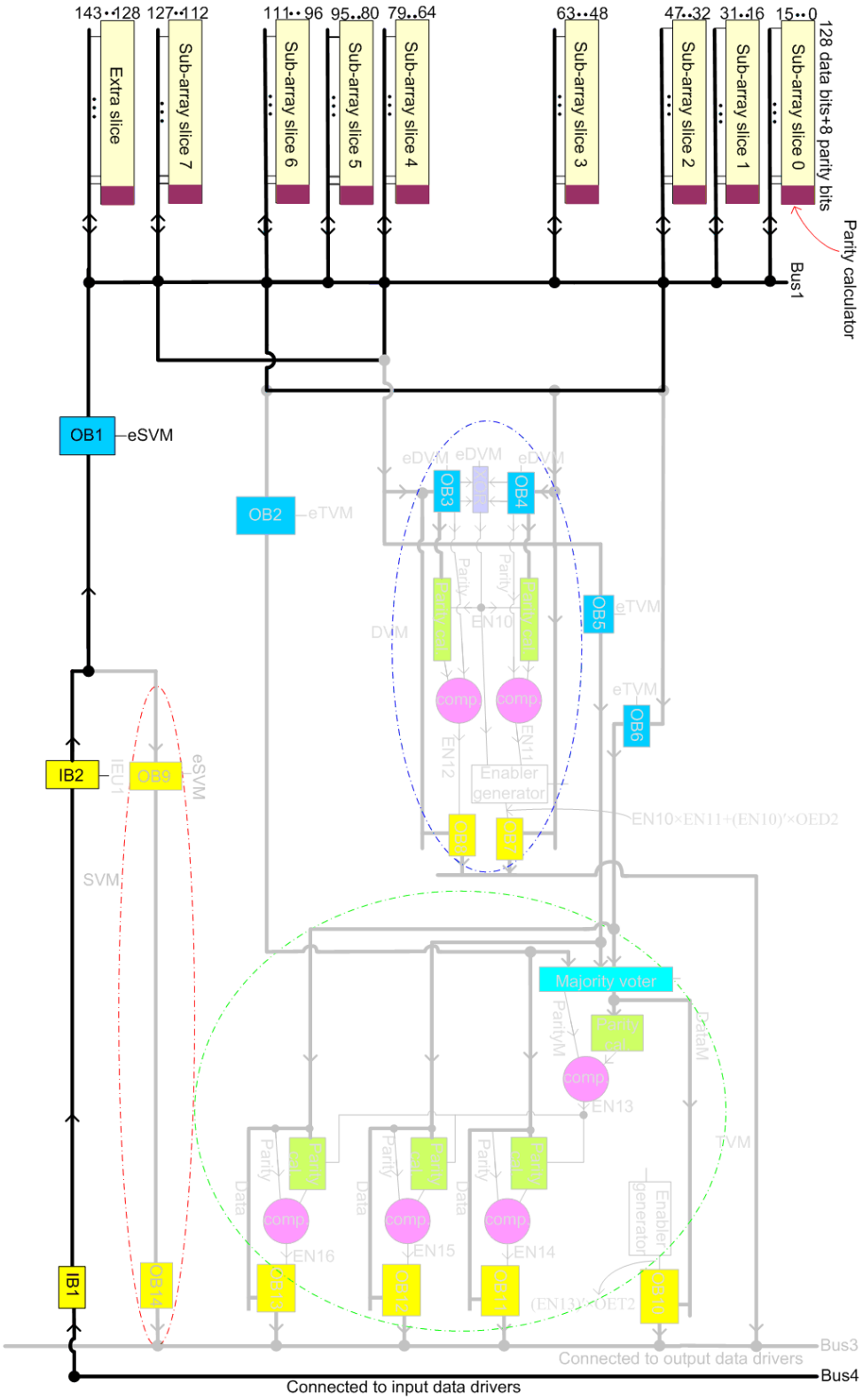


Figure 4.6: Sub-array block diagram during a write access in SVM mode.

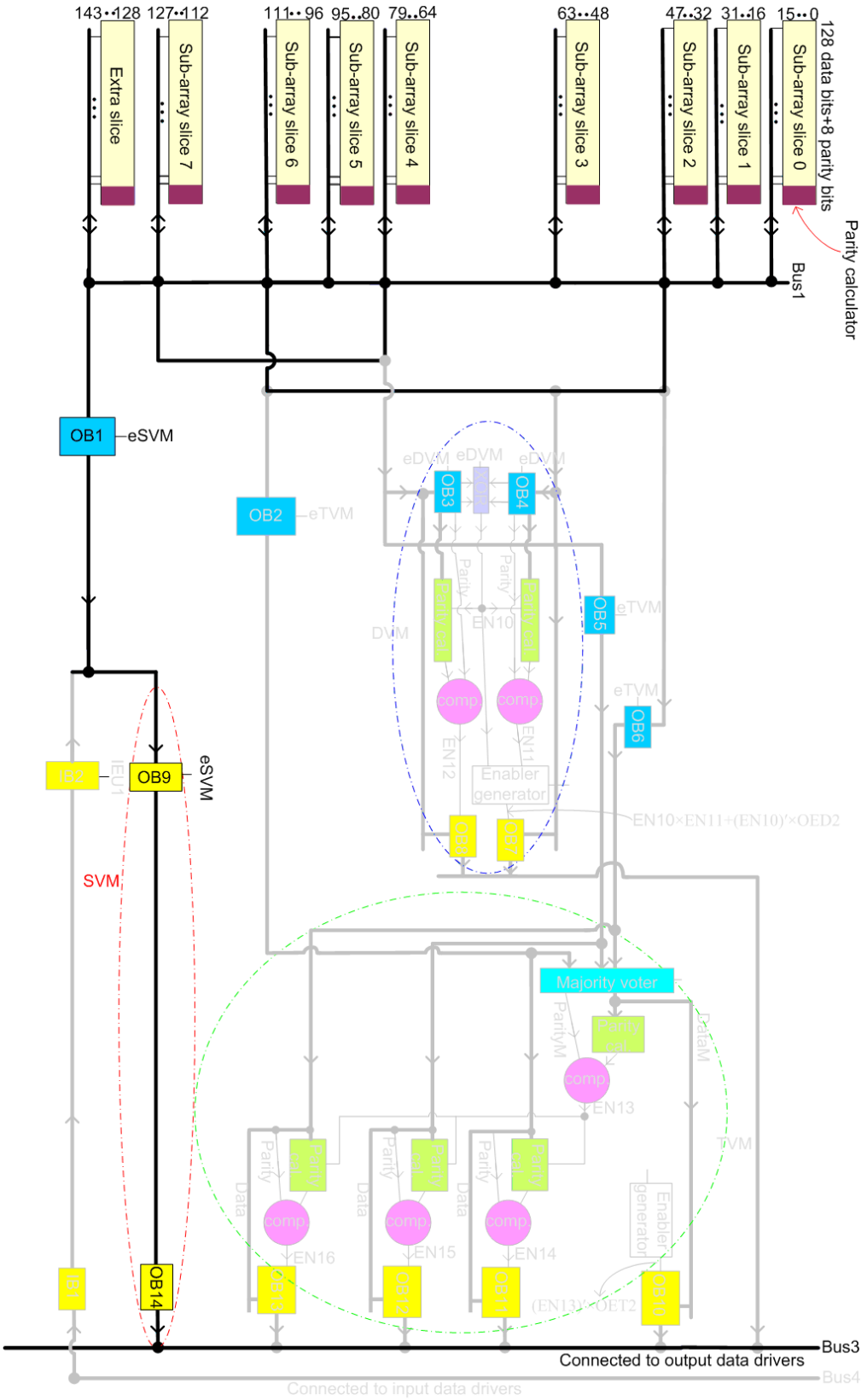


Figure 4.7: Sub-array block diagram during a read access in SVM mode.

4.4.4 Circuit Details in DVM

Figure 4.8 shows the block diagram of one sub-array when being in DVM. In medium supply voltages, Flexicache operates in DVM where at each cache-access two cache-lines are activated simultaneously. At writing time, signal IEU1 is high and data is transferred from Bus4 to Bus1 via IB1 and IB2 and is written to two selected lines at the same time (this is not shown in the figure). Parity calculator circuits generate parity bits and write them in the parity bit cells as well.

For reading the two selected lines, signals e and f are high and a, b, c, d are low, connector buffers, CB1, CB2, CB3 and CB4 are disconnected while connector buffers, CB6 and CB5 connect B2 with B4 and also B3 with B5. With this method, Bus1 is divided into two parts; sub-array slices 0, 1, 2, 6 are connected to the first part and sub-array slices 3, 4, 5, 7 are connected to the second part.

Signal eDVM is high; output buffers OB2 and OB4 transfer two selected data to the XOR circuit to check their equality. If signal EN10 is high, there is at least one bit flip (an error occurrence) in two data. Signal EN10 activates two parity calculator circuits to calculate parity bits of selected lines. Then the result of these parity calculator circuits are compared with the original parity bits of each selected-lines. These two comparators generate two enable signals EN11 and EN12. Whenever one of the comparator shows equality (when EN11 or EN12 is high), the related output buffer transfers its data to Bus3. If two compared data are equal, signal EN10 is low and output buffer OB7 transfers data to BUS3.

There are some points that we mention here: 1-We define some preferences for the cache-line addresses; if two data are identical, the line with lower address will be transferred to the output driver. 2- Each cache-line is activated with its dual cache-line; we pre-define their addresses in Section 4.5.2. 3- It takes two clock cycles to write data to selected cache-lines and three clock cycles for reading data from selected cache-lines. 4- DVM can correct the errors even if the size of the error is as long as the partition and if errors are affecting only one of both selected cache-lines. However, if the faults are in both selected cache-lines, DVM can only detect the faults with their bit-positions, but it cannot correct them.

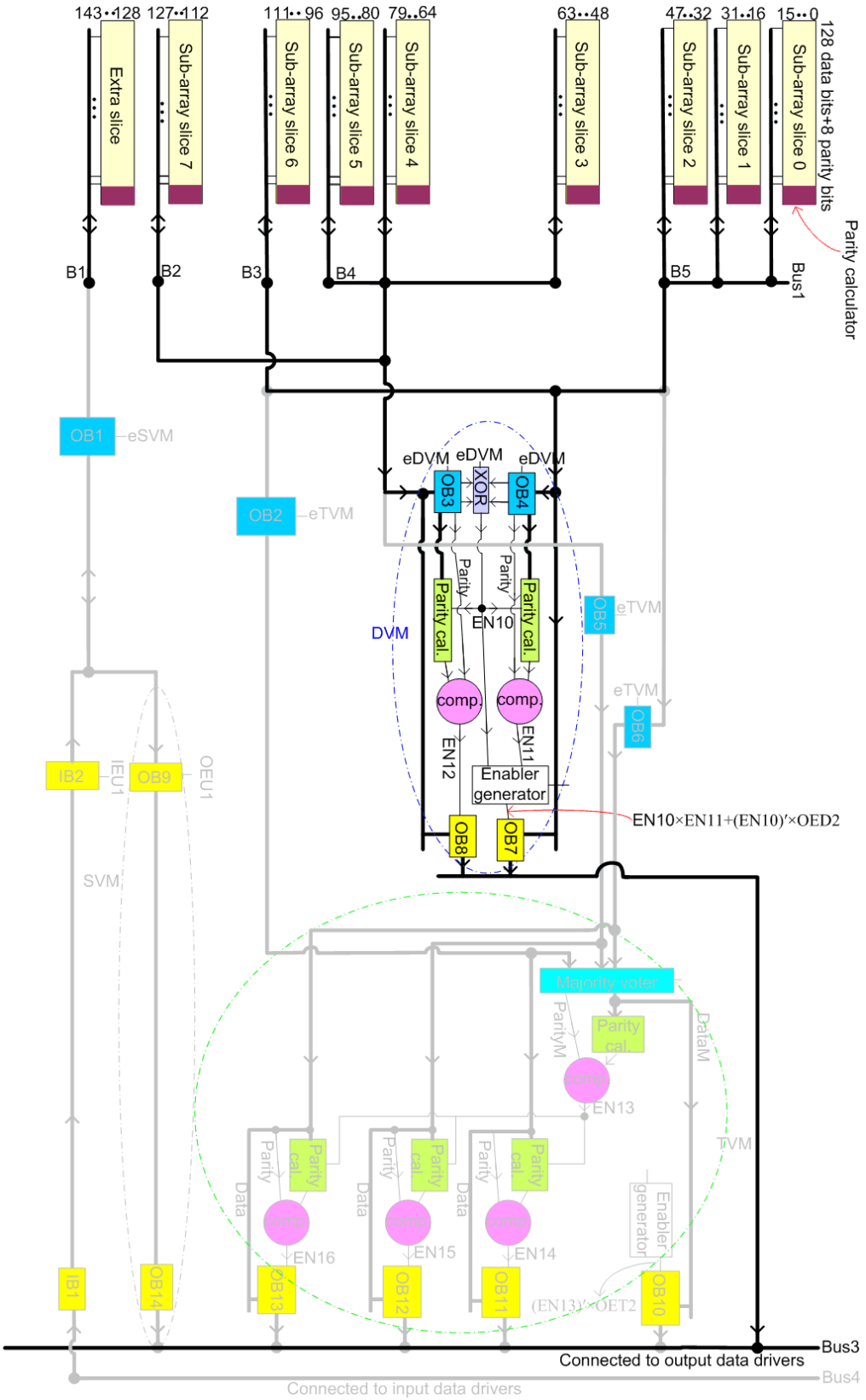


Figure 4.8: Sub-array block diagram in DVM mode.

4.4.5 Circuit Details in TVM

Figure 4.9 shows the block diagram of one sub-array when being in TVM. In TVM, IEU1 is high for writing the selected cache-lines and data is transferred from Bus4 to Bus1 via IB1 and IB2 and is written in three selected lines simultaneously (this is not shown in the figure). Similar to SVM and DVM, enabler signals are high to connect separated Bus1 nodes with each other.

At reading time, a and b are high; c, d, e, and f are low so that Bus1 is divided into three parts; sub-array slices 0, 1, 2 are connected to the first part and sub-array slices 3, 4, 5 are connected to the second part and sub-array slices 6, 7 and extra slice are connected to the third part. CB3 and CB4 circuits disconnect B3 with B4 and B4 with B5. Also CB6 and CB5 circuits disconnect B2 with B4 and B3 with B5. For reading the three selected lines, each from separate sub-array slice group, signal eTVM is high, OB2, OB5 and OB6 are active and data including parities are transferred to the majority voter (the correct value is decided by bit-wise majority voter). Signal DataM is the majority voter output for cache-lines and signal ParityM is their parity-bits.

The parity calculator circuit calculates the parity bits of DataM. One comparator circuit compares these results (the parity bits of DataM) with ParityM. If there are any differences, signal En13 will be high and the parity bits of selected lines will be calculated and compared with their original parity bits. Whenever one of the parity comparators shows equality, En14, En15 or En16 is high; the related output buffer (OB11, OB12 or OB13) is active and transfers data to Bus3. If signal En13 is low, DataM will be transferred to Bus3 via OB10.

There are some points that we mention here: 1- We define some preferences for the cache-lines: If two of En14, En15 and En16 are high, the cache-line with lower address will be transferred to Bus3; for example OB11 has a preference to transfer data compared to OB12 and OB13. 2- TVM can correct errors easily if they are affecting the bits in different positions regardless of effecting the same partition or different partitions. If a fault affects bits in the same position (there is a high possibility in very high error rates such as 1%), the majority voter takes the incorrect value for the result. Fortunately, after parity calculation, TVM can detect that the result of majority voter is not correct. Thus, it can correct errors that affect the bits in the same position if one of the three

cache-lines is error-free. The worst case scenario for TVM is that all three cache-lines are erroneous, and some errors are in the same bit position in which Flexicache cannot correct the failure. 3- Each cache-line is activated with two other cache-lines and we pre-define their addresses in Section 4.5.2. 4- It takes two clock cycles to write data to selected cache-lines and three clock cycles for reading data from selected cache-lines.

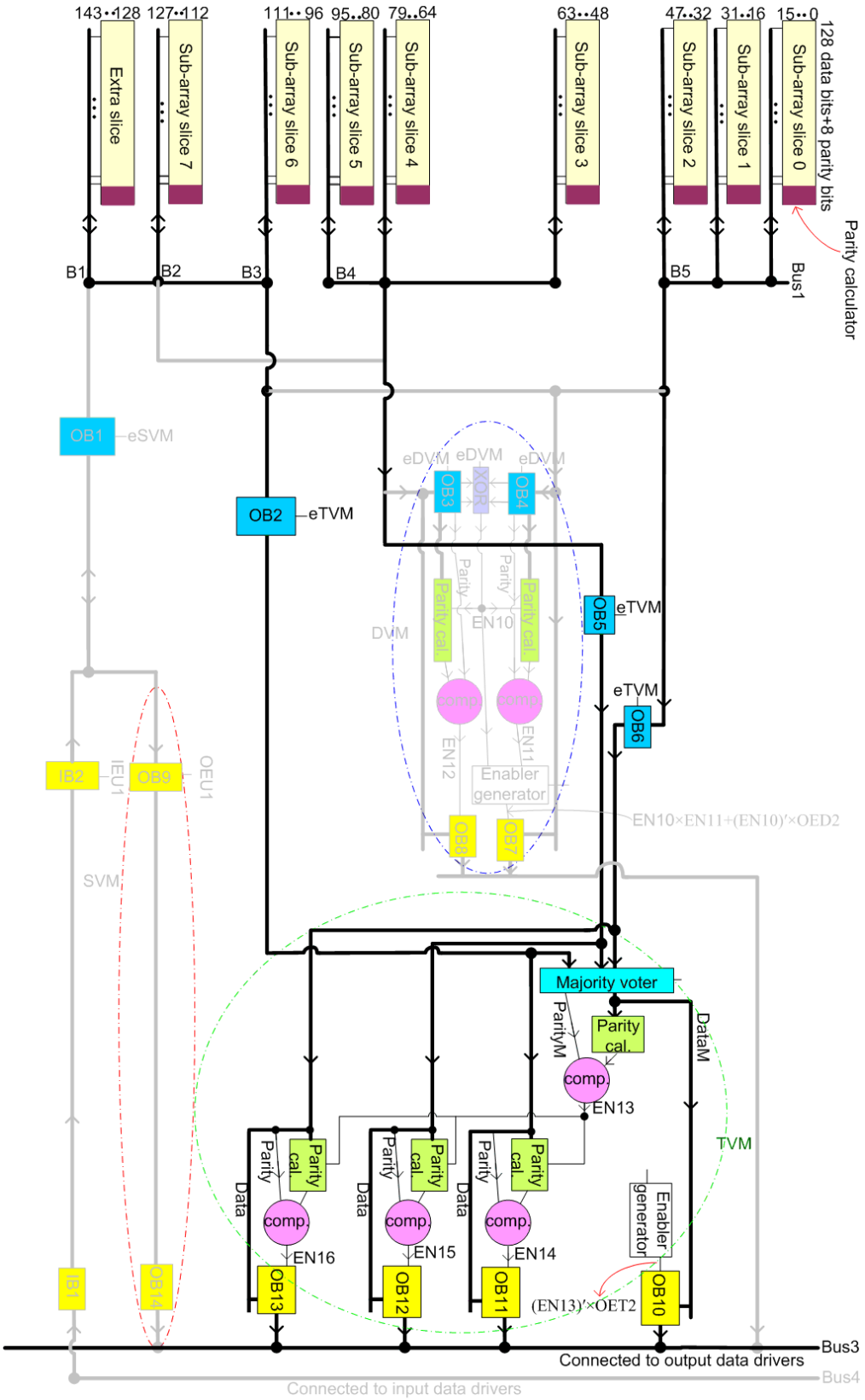


Figure 4.9: Sub-array block diagram in TVM mode.

4.5 Address Decoder Circuits of Flexicache

4.5.1 Abstract View of the Address Decoder

Figure 4.10 presents an abstract view of the address decoder. In the figure, A0 to A7 represent the address bits. The decoder uses the 4 least significant bits (i.e. A0 to A3) in order to address the line number within a slice. Also, it uses A7 to activate either the left sub-bank or the right sub-bank. Voltage Level Detector activates either SVM, DVM or TVM. These three signals together with A4 to A6 generate enable signals (EN0 to ENex) which activate slice(s). At each time, depending on the mode, one, two or three enable signals are high and data is written to (and read from) one two or three cache lines simultaneously.

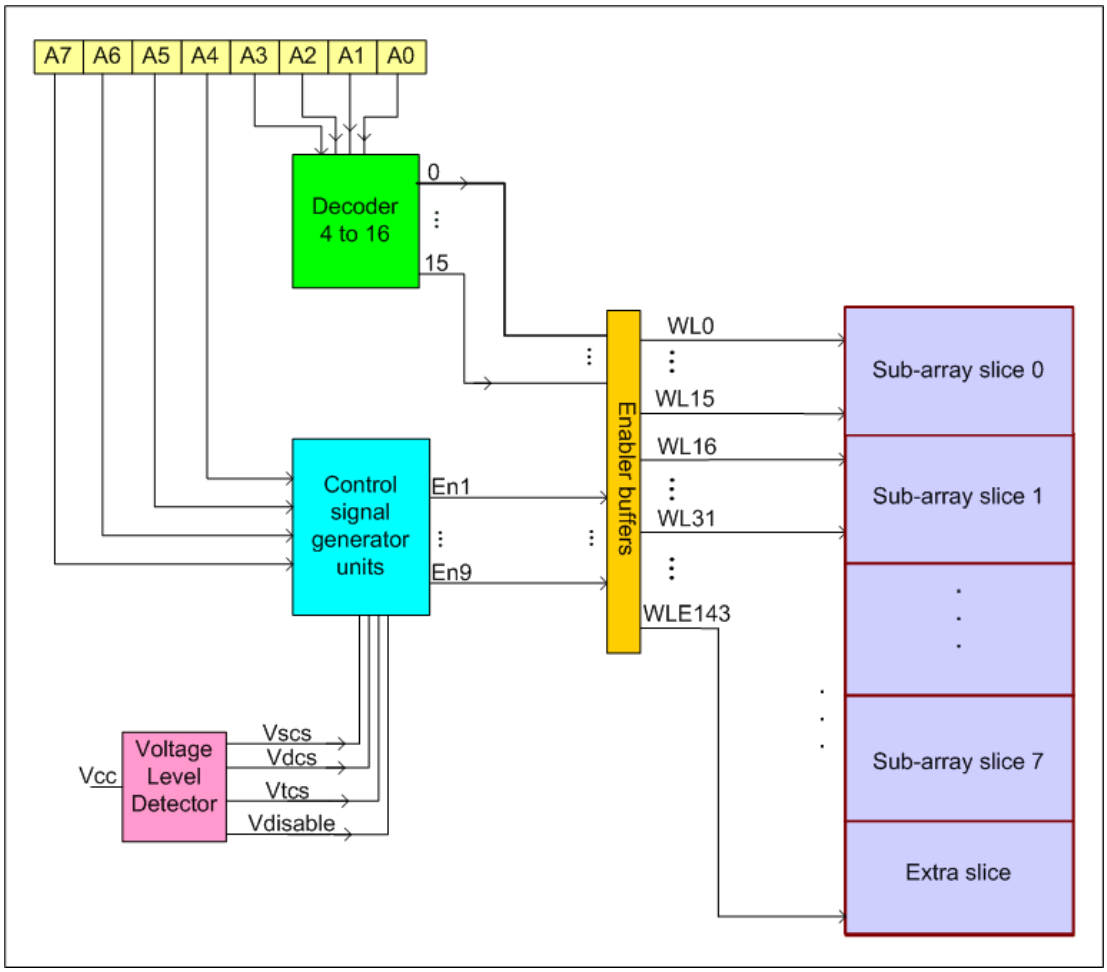


Figure 4.10: Abstract view of the address decoder of Flexicache.

4.5.2 Details of the Decoder Circuits of Flexicache

Middle part of the cache generates all word-line addresses and necessary control signals. In this Section, we explain all units of this part. The details of the address decoder are presented in Figure 4.11. Voltage level detector circuit generates four output signals V1, V2, V3 and V4 according to the supply voltage, Vcc; if V1 is high, the cache is in SVM and only one word-line address is activated at each access time; if V2 is high, the cache is in DVM and two word-line addresses are activated at each access time; if V3 is high, the cache is in TVM and three word-line addresses are activated at each access time ; if V4 is high, the supply voltage level is lower than two threshold voltages and the memory cells operate in sub-threshold mode which is beyond our work in this thesis chapter and we leave it for future; so the cache-lines will be deactivated in this state.

Pre-decoder 2, control signal generator unit 2 and control signal generator unit 3 and control signal generator unit 4 generate enabler signals, En1, En2 . . . and En9 to activate 144 word-line addresses. The logical details are depicted in Table 4.2. There are two buffers group located in the right and left side of the pre-decoder 1. Each buffers group contains 9 sub-groups, and each sub-group has 16 buffers. The outputs of pre-decoder 1 are connected to the buffers of each sub-group and generate 144 word-line addresses. All buffers of each sub-group are activated with one enabler signal. For example all buffers of sub-group 1 are enabled by signal En1. When partial address 0 and En1 are high, WL0 will be generated. In this way, all word-line addresses from 0 to 143 are generated. If A7 is high, the left part of each cache way is activated; similarly, if A7 is low, the right part of each cache way is activated.

At each access time, depending on the mode, one, two or three enable signals are high and data is written to (read from) one, two or three cache-lines simultaneously. In order to reduce the probability of a strike affecting the selected lines, they should be far from each other. The minimum 16 bit-line distance is enough for our purpose [79]. For example in DVM, WL0 and WL48 are activated simultaneously; whenever one of the addresses 0 or 48 are activated, X1 or X4 are high and Vn1 is high (look at the equations in Figure 4.12); so En1 and En4 are high. En1 and En4 are enablers for buffers and let partial address 0 pass and generates WL0 and WL48. In this way whenever one of these two addresses is activated, the other one is activated as well. in TVM, for instance, WL0,

WL48 and WL96 are activated at the same time; whenever one of the following addresses 0, 48 or 96 are activated, X1, X4 or X7 are high and Vm1 is high; so En1, En4 and En7 become high. En1, En4 and En7 are enablers for buffers and let partial address 0 pass and generate WL0 and WL48 and WL96. In this way, activated lines are far enough from each other and their distances meet our prerequisite assumption [79].

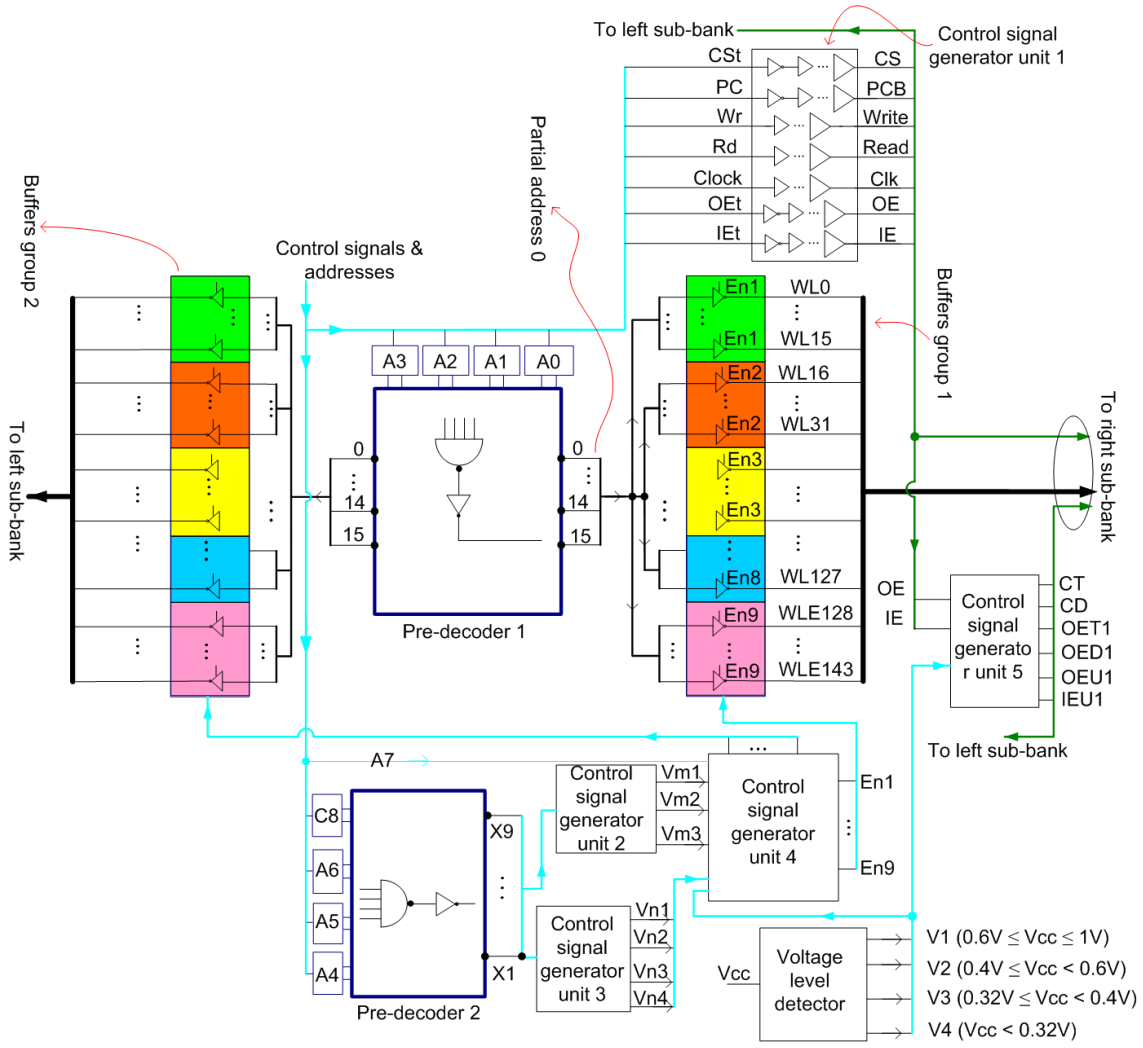


Figure 4.11: Necessary decoders and control signal generators.

Available Enablers	Word-line addresses	C8	A6	A5	A4	A3	A2	A1	A0
En1	**WL0**	0	0	0	0	0	0	0	0
	⋮	⋮				⋮			
	WL15	0	0	0	0	1	1	1	1
En2	**WL16**	0	0	0	1	0	0	0	0
	⋮	⋮				⋮			
	WL31	0	0	0	1	1	1	1	1
En3	**WL32**	0	0	1	0	0	0	0	0
	⋮	⋮				⋮			
	WL47	0	0	1	0	1	1	1	1
En4	**WL48**	0	0	1	1	0	0	0	0
	⋮	⋮				⋮			
	WL63	0	0	1	1	1	1	1	1
En5	**WL64**	0	1	0	0	0	0	0	0
	⋮	⋮				⋮			
	WL79	0	1	0	0	1	1	1	1
En6	**WL80**	0	1	0	1	0	0	0	0
	⋮	⋮				⋮			
	WL95	0	1	0	1	1	1	1	1
En7	**WL96**	0	1	1	0	0	0	0	0
	⋮	⋮				⋮			
	WL111	0	1	1	0	1	1	1	1
En8	**WL112**	0	1	1	1	0	0	0	0
	⋮	⋮				⋮			
	WL127	0	1	1	1	1	1	1	1
En9	**WL128E**	1	0	0	0	0	0	0	0
	⋮	⋮				⋮			
	WL143E	1	0	0	0	1	1	1	1

Table 4.2: Logical truth tables to generate word-line addresses and enablers.

$$X1 = \overline{C8} \times \overline{A6} \times \overline{A5} \times \overline{A4}$$

$$X2 = \overline{C8} \times \overline{A6} \times \overline{A5} \times A4$$

$$X3 = \overline{C8} \times \overline{A6} \times A5 \times \overline{A4}$$

$$X4 = \overline{C8} \times \overline{A6} \times A5 \times A4$$

$$X5 = \overline{C8} \times A6 \times \overline{A5} \times \overline{A4}$$

$$X6 = \overline{C8} \times A6 \times \overline{A5} \times A4$$

$$X7 = \overline{C8} \times A6 \times A5 \times \overline{A4}$$

$$X8 = \overline{C8} \times A6 \times A5 \times A4$$

$$X9 = C8 \times \overline{A6} \times \overline{A5} \times \overline{A4}$$

$$Vm1 = X1 + X4 + X7$$
$$Vm2 = X2 + X5 + X8$$
$$Vm3 = X3 + X6 + X9$$
$$Vn1 = X1 + X4$$
$$Vn2 = X2 + X5$$
$$Vn3 = X3 + X6$$
$$Vn4 = X7 + X8$$
$$CDE = \overline{V2} \times (V1 \oplus V3)$$
$$CTE = \overline{V3} \times (V1 \oplus V2)$$
$$OEU1 = OE \times V1$$
$$OED1 = OE \times V2$$
$$OET1 = OE \times V3$$
$$IEU1 = IE \times V1$$

$$En1 = A7 \times \overline{V4} \times [V1 + V2 \times Vn1 + V3 \times Vm1]$$
$$En2 = A7 \times \overline{V4} \times [V1 + V2 \times Vn2 + V3 \times Vm2]$$
$$En3 = A7 \times \overline{V4} \times [V1 + V2 \times Vn3 + V3 \times Vm3]$$
$$En4 = A7 \times \overline{V4} \times [V1 + V2 \times Vn1 + V3 \times Vm1]$$
$$En5 = A7 \times \overline{V4} \times [V1 + V2 \times Vn2 + V3 \times Vm2]$$
$$En6 = A7 \times \overline{V4} \times [V1 + V2 \times Vn3 + V3 \times Vm3]$$
$$En7 = A7 \times \overline{V4} \times [V1 + V2 \times Vn4 + V3 \times Vm1]$$
$$En8 = A7 \times \overline{V4} \times [V1 + V2 \times Vn4 + V3 \times Vm2]$$
$$En9 = A7 \times \overline{V4} \times [V1 + V3 \times Vm3]$$

(b)

Figure 4.12: Related equations which are necessary to generate word-line addresses and enablers.

4.6 Switching Between Modes

The Vcc can be increased or decreased at runtime, thus, Flexicache needs to switch between modes. In a naive approach, before mode switching, the whole cache is flushed which presents a cache warm-up performance overhead immediately after switch. In this section, we present a more efficient approach. We organized the activated slices in each mode in order to ease the switching. In Figure 4.13, we present the activated slices at a time during the read/write operation of DVM and TVM.

In order to switch TVM→DVM→SVM, it is adequate to flush the slices in the last column of the old mode in the tables shown in Figure 4.13. In other words, when Flexicache switches from DVM to SVM, slices 3, 4, 5 and 7 are flushed from the cache. Similarly, when Flexicache switches from TVM to DVM, slices 6, 7 and Extra slice are flushed. Also, if Flexicache switches from TVM to SVM (although many systems do not allow this fast voltage increase), combination of both columns (i.e. slices 3, 4, 5, 6, 7, Ex) are flushed. Obviously, before this flushing operation, the slices which are not flushed (i.e staying slices) should be corrected with the old mode. One option can be stopping the execution of the application right after the voltage increase, using the to-be-flushed lines for correcting the staying lines by utilizing the old mode and continuing the application execution after all staying lines are corrected. In the second option, in order to avoid this stopping overhead, all staying lines are traced after changing the mode. When a line is read for the first time after the mode change (or a dirty line is evicted from the cache), this line is corrected by using the old mode. The second or the third replica of the line can be flushed after this correction. If a line is written without reading after changing the mode, the flushing can be done without requiring any correction.

However switching SVM→DVM→TVM is not that trivial since the correct data should be updated in the second or third replica before reducing the supply voltage. Thus, for instance, when Flexicache switches from DVM to TVM, before reducing the supply voltage, lines in the slices 6, 7 and Ex are first evicted from the cache. Then, these lines are updated as the third copy. As an example, lines in slice 6 should be updated by reading the lines in slices 0 and 3 and obtaining the correct data via DVM circuit. It is only safe to reduce the supply voltage after that. Although switching SVM→DVM→TVM present the performance overhead of a runtime barrier for

updating the second or third copies, it is not a show-stopper since this switch operation is required when going towards low-power mode from the high-performance mode meaning that the application can trade off the performance for power.

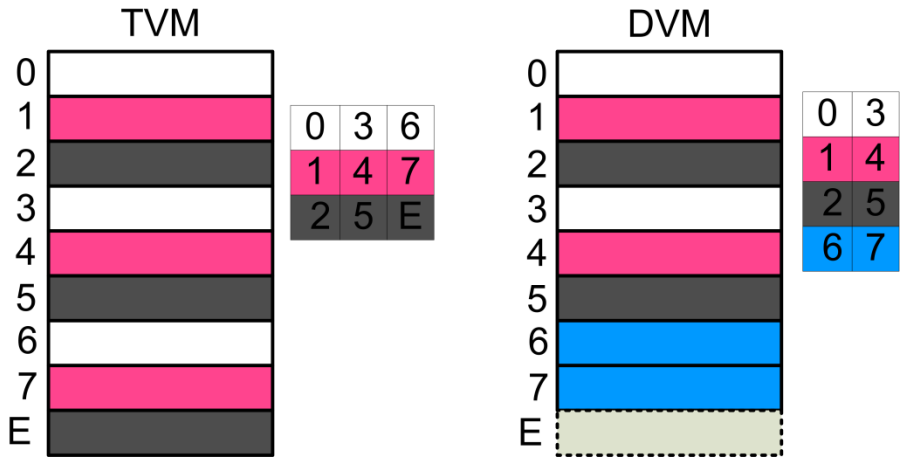


Figure 4.13: Slices activated at a time in DVM and TVM.

4.7 Methodology and Evaluation

In this section, we compare Flexicache against a conventional triplication scheme (TMR) and MS-ECC [58]. We use 4-way set associative, 64KB L1 cache with 2-cycle access time, 64B line size. We divide each line into 32 partitions for both OLSC and Flexicache with the partition size of 16 bits. Miller et al [60] examined the persistent bit failure rate in the given Vcc for 32nm technology (Figure 4.1). As Vcc is lowered, the bit failure rate increases exponentially. Flexicache targets to tolerate ultra-high bit failure rates occurring in the near-threshold voltage level without harming the performance of the cache in the low error rate. For the calculation of the Vcc at which Flexicache can operate reliably, we refer to Figure 4.1. We inject persistent faults into random locations according to bit failure rate (i.e. probability that a single bit fails) given in [60].

We calculate the useful cache capacity as the portion of the cache which is not disabled. For non-persistent failure such as soft errors, we inject multi-bit failures varying between 1 to 10 bits. We present the experimental results for the aspects of 1) useful cache capacity, 2) error correction latency, 3) energy reduction of cache operations, and 4) reliability against non-persistent faults (mean time to failure), 5) uniform view of the cache and 6) area overhead.

4.7.1 Useful Cache Capacity

Figure 4.14 compares the cache capacities. We extend Flexicache with extra slices in order to make it divisible to three, and we normalized the useful cache capacity to the non-extended capacity for fair comparison. First, when the Vcc is high, Flexicache does not sacrifice the useful cache capacity due to its flexible circuit design which dynamically switches its configuration to 64KB general purpose data cache (i.e. SVM). Second, due to the partitioning and partition-fix mechanism of Flexicache, it provides higher cache capacity than the conventional triplication schemes even in the low-power mode. Third, Flexicache can operate until the persistent bit failure rate is 12% while TMR can operate until 6% bit failure rate and MS-ECC can operate until 2% bit failure rate (bit failure rates are not shown in the graph). Therefore, TMR and MS-ECC can provide more than 20% of the cache capacity when the supply voltage is as low as 400mV while Flexicache can provide a similar amount of useful cache capacity when the supply voltage is 320 mV [17].

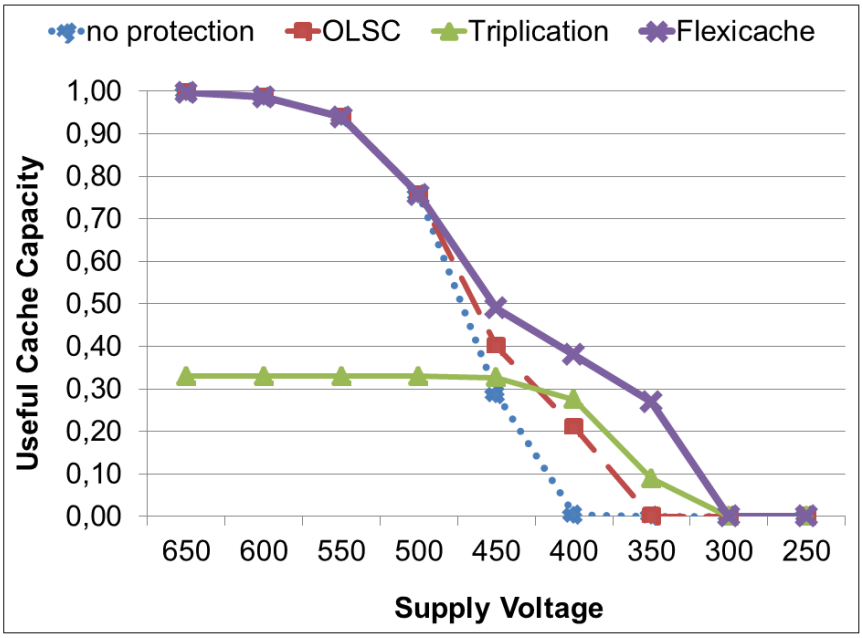


Figure 4.14: Useful cache capacity provided by Flexicache after disabling uncorrectable lines under this bit failure rate.

4.7.2 Error Correction Latency

In Table 4.3, we compare the area overhead and the latency presented by encoders and decoders in Flexicache and OLSC. We first present the number of gates in the critical paths. Although, in Flexicache, the number of gates in the critical path is higher than in MS-ECC, both encoding and decoding in each scheme can be accomplished in one cycle. Note that the decoding latency can be tolerated since decoding is done simultaneously with writing. On the other hand, total number of gates in the encoder and decoder of MS-ECC is much higher than the one in Flexicache which presents higher overhead in both read/write energies (4th row) and area (5th row). Both Flexicache and MS-ECC require changes in the address decoder of the cache to be able to write more than one line simultaneously. The overhead of these address decoders is similar in both schemes.

	Flexicache		MS-ECC	
	Encoder	Decoder	Encoder	Decoder
Number of Gates in the Critical Path	4 XORs	7 XORs + 2 ANDs + 2 ORs	2 XORS	2 XORs + 2 ANDs + 4 ORs
Total Number of Gates	480 XORs	3K XORs + 1,5K ORs + 3,5K ANDs	1,5K XORs	6K XORs + 4,5K ORs + 10K ANDs
Latency	1 cycle	1 cycle	1 cycle	1 cycle
Energy Overhead (In the nominal voltage)	2,5%	20%	5,5%	50%
Area Overhead (Encoder+Decoder)	0.06%		0.12%	

Table 4.3: The area overhead and latency presented by encoders and decoders in Flexicache and OLSC.

4.7.3 Energy Reduction

We simulate a 64-KB L1 data Flexicache with Hspice 2003.03 using HP 45-nm Predictive Technology Model [22] at 2GHz processor frequency. Figure 4.15 presents the energy consumption of cache operations (i.e. read/write energy and static energy). For read and write energies, TMR allocates three cache ways in a non-modified cache which triplicates the energy consumption. Similarly MS-ECC allocates two cache ways (1 for data and the other for parity bits) when the supply voltage is lower than 700 mV, thus at this point MS-ECC also roughly duplicates reading and writing energies. This is mainly because the size of the in/out data is duplicated (or triplicated). Also, the energy consumption of the OLSC decoder is very high (i.e. 50%). Thus, which diminish the energy saving of scaling voltage for read energy as it can be seen at 600mV when OLSC is activated in MS-ECC (Figure 4.15a and Figure 4.15b). On the other hand, Flexicache accomplishes replication and fault recovery within a way without increasing the size of the data in/out bus coming to the way. Thus, reading and writing energies of Flexicache is much lower than MS-ECC and triplication.

For the static energies (i.e. energy spent in one cycle when the cache is idle), Flexicache presents slightly higher energy consumption than an unmodified cache mainly due to the additional extra slices (Figure 4.15c). Note that these additional slices also increase the cache capacity that we excluded this increased capacity in our previous results. The static energy consumption of MS-ECC is negligibly higher than a non-modified cache due to OLSC encoder/decoder. It has been shown that dynamic energies are only the 30% of cache energy consumptions and among them they are mostly (two out of three) read operations.

By considering that, in Figure 4.15d, we present the average energy consumption of a cache at a time. The figure shows that only Flexicache can operate when Vcc is 320 mV by presenting 39% reduction in the energy consumption of the cache compared to a non-modified cache when it executes in the high-performance mode with the minimum safe Vcc (i.e. 700 mV). MS-ECC can reduce the energy consumption by only 5% compared to the same minimum safe voltage level.

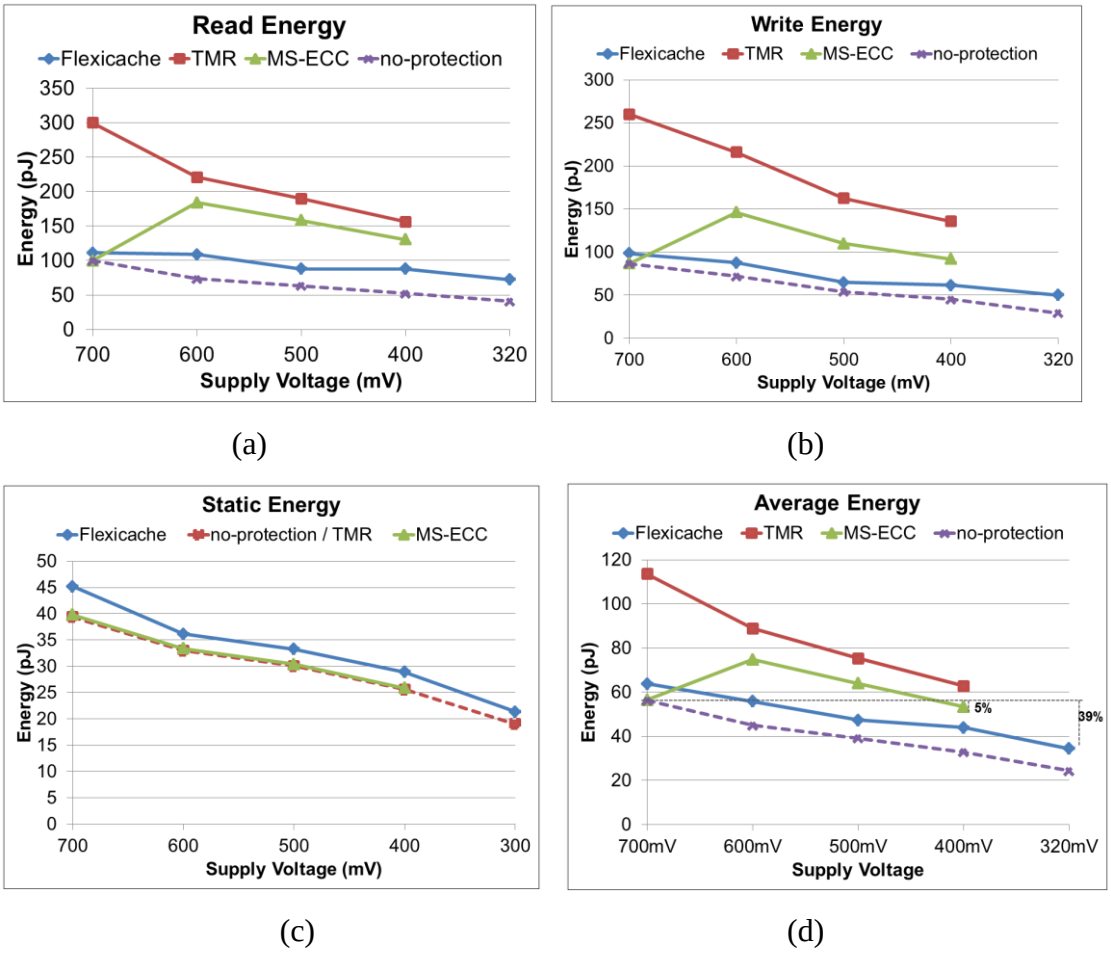


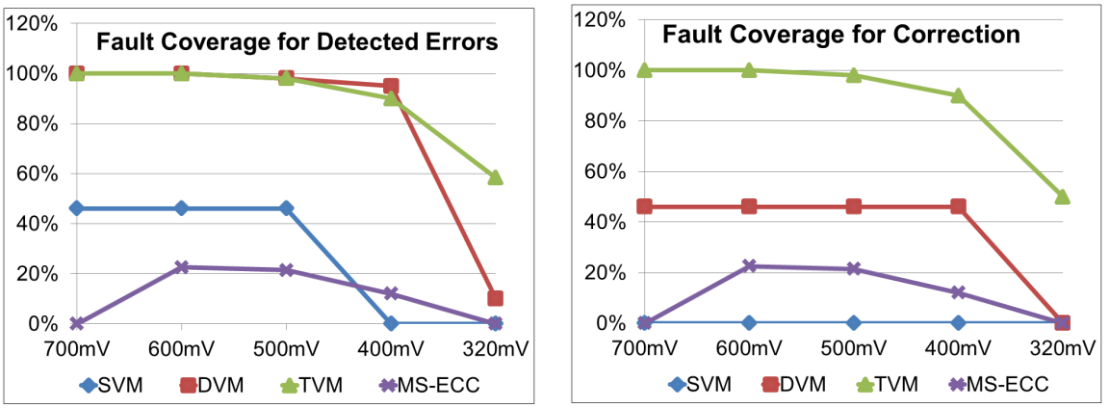
Figure 4.15: Energy reduction in cache operations

Cosemans et al. [80] evaluate the energy consumption of the cache elements during the read or the write operations in a design based on 90nm technology. For instance, during the read operation, timing components (including delay elements and control wires) is the most energy consuming element (i.e. 30%).

Similarly, address decoder consumes around 25% of the read/write energy. Since Flexicache still uses the most of the energy-hungry components (e.g. buses, data drivers and the address decoder) only once in DVM and TVM, it slightly increases the energy consumption of timing elements and the address decoder. On the other side, Flexicache only duplicates (triplicates) the energy consumption of cells and sense amplifiers which consume less than 15% of the read/write energy. Thus, Flexicache presents modest additional energy consumption in DVM and TVM.

4.7.4 Reliability Against Particle Strike

In Figure 4.16, we inject non-persistent, multi-bit faults (i.e. sizes of the faults are between n=1-10 bits which means n adjacent bit become faulty due to a particle strike) to the non-disabled cache portion and, we present the fault coverage (i.e. the percentage of the injected faults) for error detection (Figure 4.16a) and error correction (Figure 4.16b). In the high-performance mode, MS-ECC cannot detect or correct non-persistent faults since it does not extend the cache lines with ECC codes. On the other hand, each cache line is extended with ECC protection in the low-power mode when the persistent fault rate is very high. At this point, additional multi-bit non-persistent faults leads the total number of faults in the cache line higher than OLSC can correct. Thus, non-persistent fault correction capability of MS-ECC is around 20% or less. Note that error detection capability and error correction capability of MS-ECC are identical since OLSC intends to produce the correct data without trying to detect if there was a fault or not. In SVM, Flexicache cannot correct faults, but it can detect half of the injected faults (i.e. when the size of the fault is odd). In DVM, it can correct half of the injected faults since it uses parity for the error correction while it can detect more than 90% of the injected faults. TVM can provide more than 90% error correction capability until Vcc is 400mV. When Vcc is 320mV, only TVM can provide useful cache capacity. At this point, it can detect 58% of the injected non-persistent faults and can correct half of the injected faults. In this study, we switch from SVM to DVM when the Vcc is 600mV. One can decide to utilize DVM for higher Vccs for reliability critical applications or systems in faulty environments in order to provide higher reliability with the cost of useful cache capacity.



(a) (b)

Figure 4.16: Non-persistent fault injection

4.7.5 Area Overhead

Figure 4.17 shows the layouts of one sub-bank [23] for both Flexicache and typical cache arrays; the second symmetric sub-bank is omitted for simplicity. After adding parity bits, parity calculators, extra slices, XORs, majority voters, buffers and peripheral circuits, Flexicache presents 12% area overhead compared to the typical cache without any protection. The biggest portion of this overhead belongs to the extra slices which we add to make the cache dividable by three, therefore, actually increasing the size of the cache. This layout allows Flexicache dynamically switch between SVM, DVM and TVM which provides maximum 100%, 50% and 33% useful cache capacity as we presented in Figure 4.14.

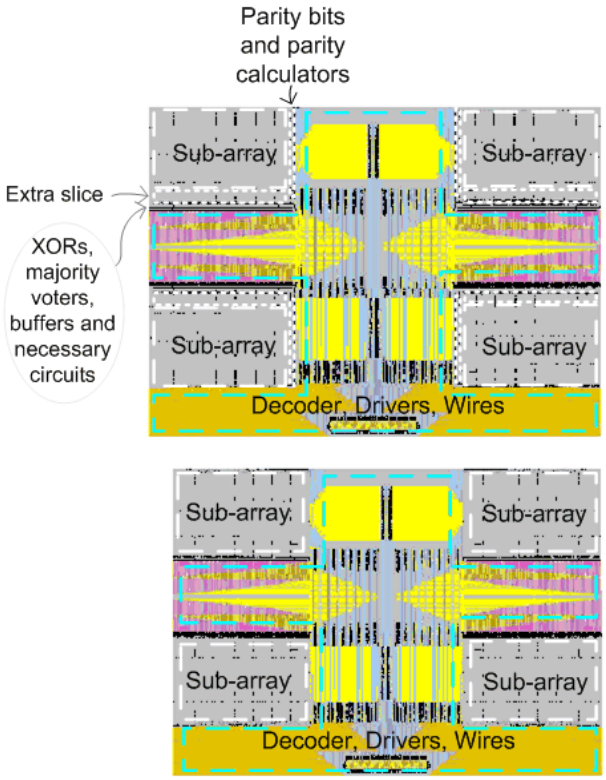


Figure 4.17: The figure shows the layout of a sub-bank and address decoder of Flexicache (top) and a typical cache (down).

4.8 Conclusion

In this chapter, we present the circuit design of Flexicache, a novel, reliable cache design which configures itself for different supply voltages from the nominal to the near threshold voltage levels in order to duplicate or triplicate each data line if higher reliability is required. Flexicache can continue to operate reliably up to 10% bit failure rate. Therefore, it alters the possibility to operate in 320 mV. Compared to the state of art techniques, Flexicache provides a higher capacity in low-power mode with significantly less energy consumption. Also, Flexicache can provide higher reliability against non-persistent faults [17].

CHAPTER 5

NOVEL CAM CELL BASED ON NANO-ELECTRO-MECHANICAL SWITCH AND CMOS FOR HIGHLY EFFICIENT TLBS

5.1 Introduction

Nowadays, CMOS generation scaling is approaching profound limits due to the exponential increase of the leakage current during technology scaling. The subthreshold leakage power is mainly affected by the subthreshold-swing (S) of a device, which defined as the amount of gate voltage reduction to reduce the subthreshold current by one decade (S=dVgs/dlogId) [81]. For bulk CMOS, the subthreshold swing has a fundamental lower limit of 60 mV/decade, which leads to a large increase in the power density [82]. In order to maintain the scaling ability, a significant amount of research is ongoing to explore potential alternatives for CMOS technology which motivate us, in this dissertation, to exploit one of the most promising emerging technologies (Nano-Electro-Mechanical switches) to improve the processor performance.

Nano-Electro-Mechanical (NEM) switches have been suggested as a promising candidate for replacing the CMOS technology [83]. NEM switches provide some unique characteristics which are not available in conventional MOS, such as near-zero leakage current and infinite subthreshold slope (less than 0.1 mV/decade [84]). Such characteristics make them ideal for designing highly energy-efficient structures. However, NEMs have relatively long mechanical switching delay [83] compared to the intrinsic delay of CMOS devices, and to this date, they suffer from low endurance (10^{11} write cycles) [85]. Also, NEMs do not offer high turn on current like CMOS transistors.

In spite the large mechanical delay and limited number of reliable cycles, NEMs have been useful for a wide range of applications such as FPGAs (used as programmable routing switches) [86], [87], [88], [89], adders [90], [91], flip-flops [92], memories [93], [82], DACs [90], [91], and ADCs [91] where the long switching time and limited number of hits are not important issues. Most of the mentioned circuits use the benefits of combining NEMs and CMOS technology in order to highlight the advantage points of each technology and alleviate the disadvantages to achieve low-power and high performance operation for some critical components. Motivated by these observations, in this thesis, we propose the integration of NEMs and CMOS cells to design low-power

processor structures like Content Addressable Memories (CAM) where writes are relatively rare, for example the Translation Look-aside Buffers (TLB).

Content Addressable Memories (CAM) have been widely adapted for applications that depend on fully-associative and high-speed search operations, such as translation look aside buffers (TLBs), network routers, and data compression [94]. Since the search operation requires fully-parallel and fast comparisons, CAMs introduce high energy consumption and area constraints. Previous works have explored the design of CAMs with emerging memory technologies to mitigate these issues [95], [96], [97], [98], [99]. However, those CAMs suffer mainly from increased search latency due to the employed technology which prevents them from building performance critical structures such as TLBs.

In this thesis chapter, we introduce a novel CAM cell based on NEMs and CMOS, called NEMsCAM [100]. The memory part of the NEMsCAM cell is constructed with two vertical complementary non-volatile NEM switches, while the comparison circuits are designed with CMOS transistors to allow fast search operation. Novel structure of NEMsCAM along with unique characteristics of NEMs considerably reduces the layout area as well as the energy consumption. Also, in this design, both Out and OutB are available simultaneously, which is essential to design a CAM cell.

As a use case, we leverage the NEMsCAM cell to build fully associative TLBs. The TLB has been pointed out as a critical component of energy and performance in modern processors [101]. We design first-level TLBs with 16nm technology at 2GHz for both data and instruction accesses that complete the search operation in one clock cycle. Our analysis shows that the NEMsCAM DTLB reduces the energy per search and write operation and standby mode by 27%, 41.9% and 53.7% respectively, and the area by 40.5% compared to a CMOS-only TLB. Furthermore, we show that the NEMs' increased write latency introduces minimal performance overhead (0.27% on average).

The main contributions of this chapter are:

_ We design the NEMsCAM cell based on complementary non-volatile NEM switches and CMOS transistors.

_ We design highly efficient first-level TLBs for data and instruction accesses based on the NEMsCAM cell.

_ We evaluate the proposed designs at both circuit and system level, and compare to CMOS-only TLBs.

Section 5.2 provides background information while Section 5.3 and 5.4 describe the design of the NEMsCAM cell and the NEMsCAM TLB respectively. In Section 5.5 we present our evaluation methodology and the obtained results. Finally, in 5.6 we discuss the related work and in Section 5.7 we summarize the chapter.

5.2 Background

This section provides background information related to this work. First, we briefly review recent available emerging technologies. Then, we describe NEM switches in detail and the prior art in their use as memory. Finally, we describe CMOS-only CAM cells and fully-associative TLB structures.

5.2.1 Emerging technologies

In order to continue the trend of Moore's Law, many emerging technologies have been employed such as Phase-Change memory (PCRAM) [102], [103], Magnetoresistive RAM (MRAM) [104], Spin-Torque Transfer Magnetoresistive RAM (STT-RAM) [79], Ferroelectric RAM (FeRAM) [105], Memristor [95], and Nanomechanical Memory (NEM) [84], [89]. Typical performance parameters of mentioned memory technologies are presented in Table 5.1 [106], [95].

Ferroelectric RAM (FeRAM or FRAM) is one promising memory which employs a ferroelectric gate cell instead of a poly-silicon cell and has been considered as a replacement for flash memory [105]. In spite of some disadvantages like lower density, higher cost and poor scalability, it has some advantages over flash like faster programming time, lower power usage, and higher endurance.

Magneto-Resistive RAM (MRAM) which has been designed with magnetic storage elements is another emerging technology [104]. The elements are made of two ferromagnetic plates, each of them holds a magnetic field and an insulating layer separates them. One of plates has a permanent magnetic field and the other has a variable one to be able to store data. MRAM is as fast as SRAM and as dense as DRAM.

	Traditional Technologies				Emerging Technologies				
			Improved Flash						
	DRAM	SRAM	NOR	NAND	FeRAM	MRAM	PCRAM	Memristor	NEMS
Cell Elements	1T1C	6T	1T		1T1C	1T1R	1T1R	1M	1T1N
Half pitch (F) (nm)	50	65	90	90	180	130	65	3-10	10
Smallest cell area (F^2)	6	140	10	5	22	45	16	4	36
Read time (ns)	< 1	< 0.3	< 10	< 50	< 45	< 20	< 60	< 50	0
Write/Erase time (ns)	< 0.5	< 0.3	10^5	10^6	10	20	60	< 250	1ns(140ps-5ns)
Retention time (years)	Seconds	N/A	> 10	> 10	> 10	> 10	> 10	> 10	> 10
Write op. Voltage (V)	2.5	1	12	15	0.9-3.3	1.5	3	< 3	< 1
Read op. Voltage (V)	1.8	1	2	2	0.9-3.3	1.5	3	< 3	< 1
Write endurance	10^{16}	10^{16}	10^5	10^5	10^{14}	10^{16}	10^9	10^{15}	10^{11}
Write energy (fJ/bit)	5	0.7	10	10	30	1.5×10^5	6×10^3	< 50	< 0.7
Density ($Gbit/cm^2$)	6.67	0.17	1.23	2.47	0.14	0.13	1.48	250	48
Voltage scaling	Fairly scalable					No	poor	promising	promising
Highly scalable	Major technological barriers				Poor		promising	promising	promising

Table 5.1: Traditional and emerging memory technologies characteristics [106], [95].

Also it has non-volatility characteristic similar to flash and high endurance. However, it suffers from large cell size, high write current and poor scalability which greatly forbids being widely commercialized.

Another non-volatile memory is Phase Change memory (PCRAM) [102], [103]. Similar to optical storage devices, it stores data into chalcogenide glass. The state of the glass is changed to crystalline or amorphous whenever an electric current passes through a heating element and generates heat or quenches the glass. The main limitations of PCRAM are its high programming current and relatively long write/read time.

Memristor is considered to be one of the best candidates for future memory technologies. A Memristor is a two-terminal non-volatile memory and has been designed based on resistance switching effects [95]. Memristor has low write energy and high density due to multi-layer crossbar architecture. Since the Memristor crossbar-based architecture is highly scalable, it is predicted to be the selected candidate to use for future ultra-high density memories. As no tunnel oxide is used in this device, Memristor

has higher endurance than flash memory. In spite of high density and endurance, read time is considerably high. Memristors may easily replace flash memories; however, they are not an appropriate option to employ in extremely fast system components.

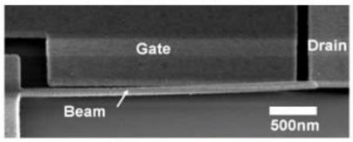
Another promising candidate as a replacement for CMOS devices is the Nano-Electro-Mechanical switch (NEM) [84], [89]. On/off state of NEM devices is determined by both electrical and mechanical forces between gate (a movable beam) and source terminals. Unique characteristics of NEM relays which are not available in conventional CMOS, such as zero leakage current and near infinite sub-threshold slope, make them ideal for designing highly energy-efficient applications. In spite of low leakage current, NEM relays do not offer high turn-on current like CMOS transistors; moreover, they suffer from low endurance.

5.2.2 NEM Switches

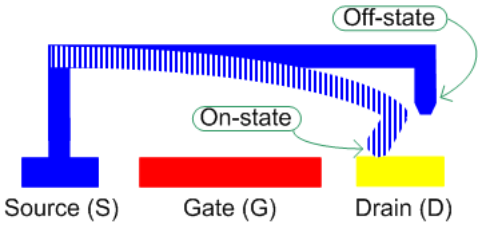
A NEM switch is a device in which a mechanically moving part makes or breaks a conductive path, typically in response to a voltage difference applied over its terminals. There are many different implementations of NEM switches [107], [108], [109].

Figure 5.1(a) and Figure 5.1(b) shows the simplest one, 3T NEM, that consists of three terminals, a cantilever beam (which connected to source terminal), a gate, and a drain. The voltage difference between the gate and the movable terminal (VGS) controls the position of the beam and state of operation. When VGS goes higher than a certain threshold value, called the pull in voltage (Vpi), the electrostatic force exerted by the gate exceeds the elastic force of the beam and pulls the beam toward the drain until the beam collapses to the drain and form a conductive channel from the beam (source) to drain, thus closing the switch (on-state; Figure 5.1(c)). In order to release the beam from the drain, VGS decreases to a voltage smaller than Vpi, called pull out voltage (Vpo) where the electrostatic force is not higher than the elastic force of the beam; at this moment the beam is disconnected from the drain (off-state; Figure 5.1(c)). Due to such sharp on/off transition, NEMS have zero off-state leakage as there is no path for current to flow. Moreover, because of the surface adhesion force between the two contacting regions, Vpo is usually smaller than Vpi and the IV characteristics of NEMS exhibit

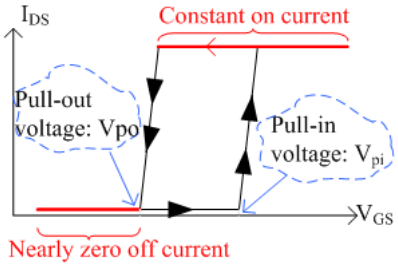
such a hysteresis characteristic which enables NEM relays to be used as memory elements (Figure 5.1(c)).



(a)



(b)



(c)

Figure 5.1: (a) SEM image of three-terminal NEM relay [110]. (b) Schematic of a 3T NEM relay consisting of source, gate, and drain; a cantilever beam connected to the source. The beam collapses to the drain when $V_{GS} \geq V_{pi}$ and is released when $V_{GS} < V_{po}$ (c) Typical I_{DS}-V_{GS} characteristics of 3T NEM array.

Figure 5.2(a) and Figure 5.2(b) illustrates the geometry of the 5-Terminal (5T) NEM switch we are considering in this work. A suspended beam is anchored at the source. Two gate terminals (Gate1 and Gate2) are positioned in close proximity to the beam. The beam can connect to one of two output nodes (Drain1 and Drain2). A voltage difference between Gate1/Gate2 and the beam causes the beam to move towards Gate1/Gate2 because of the electro-static attraction (Figure 5.2(b)). The beam will then connect to Drain1/Drain2, creating a conductive path between the source and this drain. An advantage of using two gates is that the electrostatic force can be used as both pull-out and pull-in voltages of the beam. Hence, one does not have to rely on only the elastic restoring force of the beam. This significantly improves the operational margins and the

scalability of the device. As mentioned before, the mechanical movement of the beam is fairly slow, depending on the device technology. Consequently, write operations in NEM switches take multiple clock cycles [93].

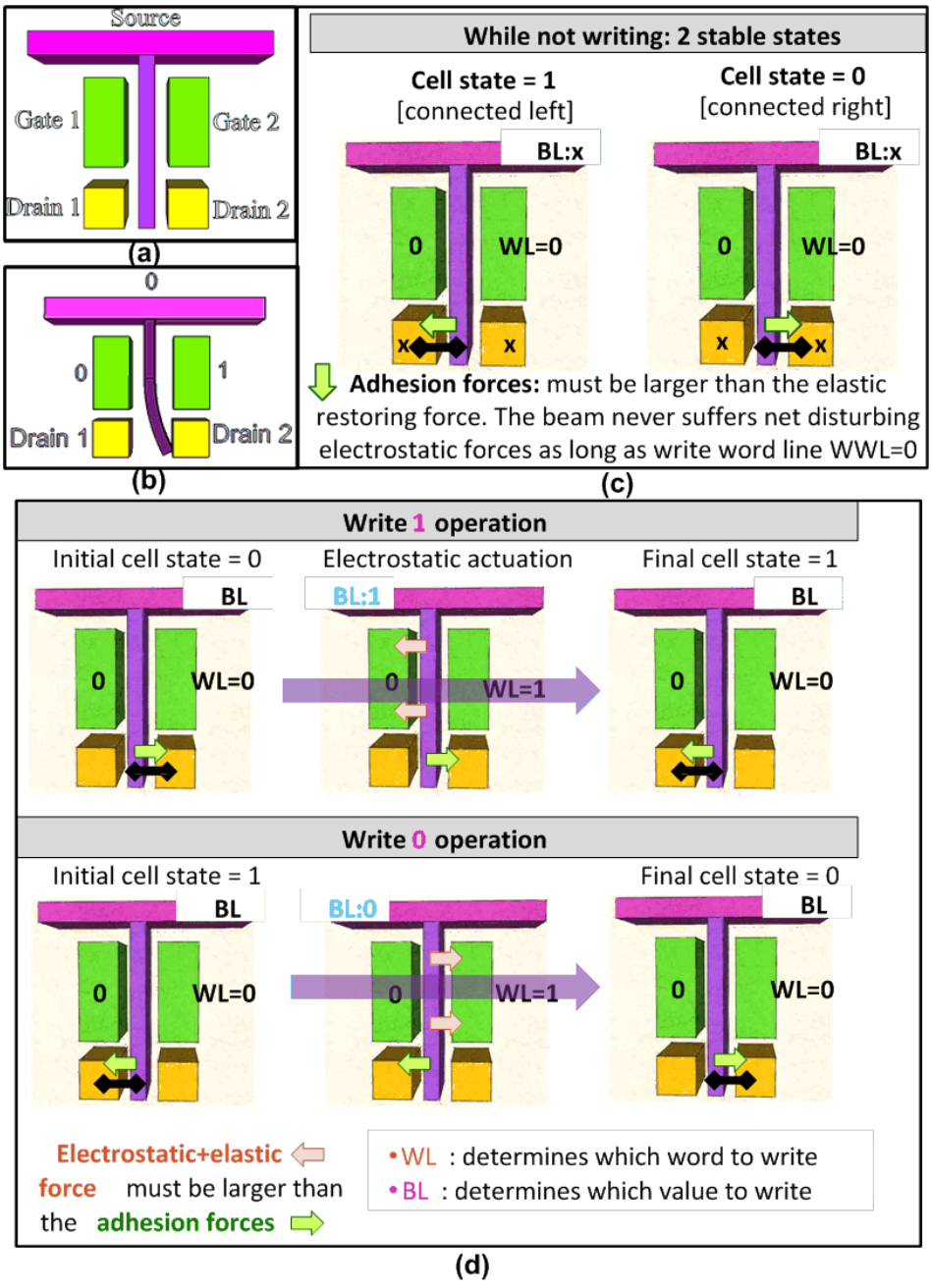


Figure 5.2: (a) Schematic of a 5T NEM switch. (b) VGS1 (VGate1- VSource) is 0 and VGS2 is 1, so the beam collapses to Drain2. (c) The non-volatile NEM assumed in this work has two stable states. (d) Biasing scheme used for writing the non-volatile NEM switching. WL selects the cells for writing, while the BL determines the value that will be written.

5.2.3 Non-volatile NEM switches

For our application, the NEM switches must exhibit non-volatile behavior: once they are connected to a drain, they must stay in that position until the beam is pulled out by electrostatic forces from the opposite gate. To achieve this we use the NEM switch described in [111]. Figure 5.2(c) shows the two stable states of this NEM switch. As long as both gates are at the same potential (wordline WL=0), the beam will never suffer a net disturbing electrostatic force. Figure 5.2(d) illustrates the write operation for this switch.

5.2.4 Memory arrays based on NEM switches

Previous studies have looked into using NEMs for memory applications [112], [93], [113], [110], [114], [115], [116]. Some of them address the use of NEM switches to replace normal memory arrays, such as SRAM. Chong et al. [93] replace the two pull-down transistors in a 6T SRAM cell with NEM switches to reduce leakage and area. Some of these studies also discuss non-volatile memory arrays [113], [115], [110], [114]. The memory array structure disclosed in [114] is of particular interest to our proposal as we explain in Section 5.3.

5.2.5 Configuration elements based on NEM switches

NEM switches have been proposed for configuration tasks. Dong et al. [88] used 3T NEM switches as configuration memory elements in FPGAs, replacing a routing switch by one NEMs, or a LUT cell by two NEMs. This structure could potentially be used for designing a CAM cell; however, the presented cell has several shortcomings. It suffers half-select conditions and relies only on the elastic restoring force for pull-out, it is volatile, and it outputs only Out, not the complementary OutB. The configuration of the storage part of NEMsCAM which we present in Section 5.3 addresses these shortcomings.

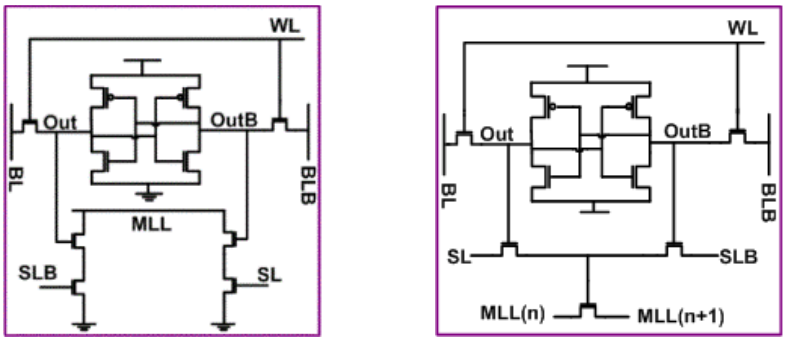
5.2.6 Content Addressable Memory

A Content-Addressable Memory (CAM) concurrently compares the input search data with all of its stored data and returns the address of the matching location in a single

clock cycle [109]. A typical CMOS-only CAM cell incorporates an SRAM cell to store the data bit and additional XOR circuits to compare the stored bit with the search data. CAMs form a popular solution for a wide variety of applications that require high search speeds such as data compressions, network routers, and lookup tables [110]. However, the search operation in a CAM requires fully parallel comparison circuits to meet timing requirements. This results in high energy consumption and poses constraints in the number of entries affecting directly the effectiveness of the CAM.

CAM cells can be implemented into two types, namely NAND or NOR CAM cells [94]. The circuit of a binary NOR CAM cell can be seen in Figure 5.3. A NAND-type CAM architecture is useful for low power application but it is slow; in contrast, NOR-based CAM cells provide much faster speed which comes at a cost of higher power consumption. Also, each CAM cell can be a binary cell and stores logic "0" or "1" or a ternary CAM cell and which operates with a "don't care" value (X) in addition to logic “0” or “1” [94].

Many techniques have been proposed to reduce CAM power consumption in circuit and also in architecture levels [117], [118], [119]]. At the circuit level, many techniques have focused on saving energy by reducing the ML and SL power consumption [117]… [120]. Most of these techniques are proposed for NOR CAM cells [117], [120], [121]. However, these techniques increase latency as well as circuit complexity, while the power dissipation of large-capacity CAMs is still too high.



(a) (b)

Figure 5.3: (a) Typical 10-T NOR type CAM cell, (b) Typical 9T NAND type CAM [94]**.**

5.2.7 Translation Lookaside Buffer

Virtual memory simplifies programming by abstracting and managing the available physical memory in pages. To accelerate virtual memory, processors employ the Translation Lookaside Buffer (TLB) that holds recently used virtual-to-physical translations [122]. The processor searches the TLB on every memory operation using the virtual page number. In case of a hit, the TLB returns the physical page number so that the memory operation can further proceed with accessing the memory hierarchy. However, in case of a miss, the memory operation will not complete until the address translation is retrieved from the memory (page walk) which might take up to hundreds of cycles. The TLB is hence a crucial component for the performance of the processor [122].

A simplified version of a TLB can be seen in Figure 5.4. Figure 5.4 shows the block diagram of two main components of TLB, typical CAM and RAM structures; each consists of CAM cells and SRAM cells respectively.

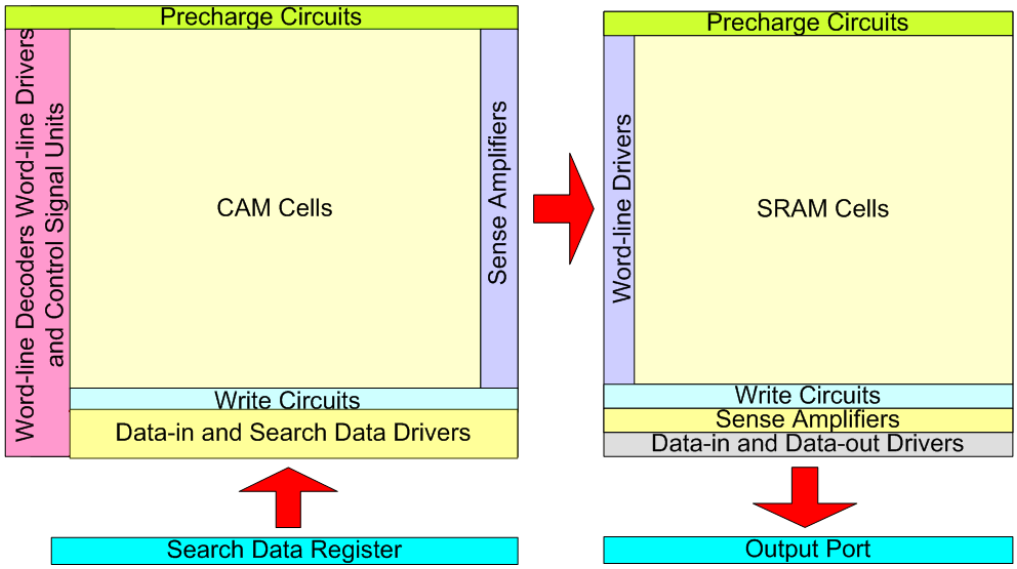
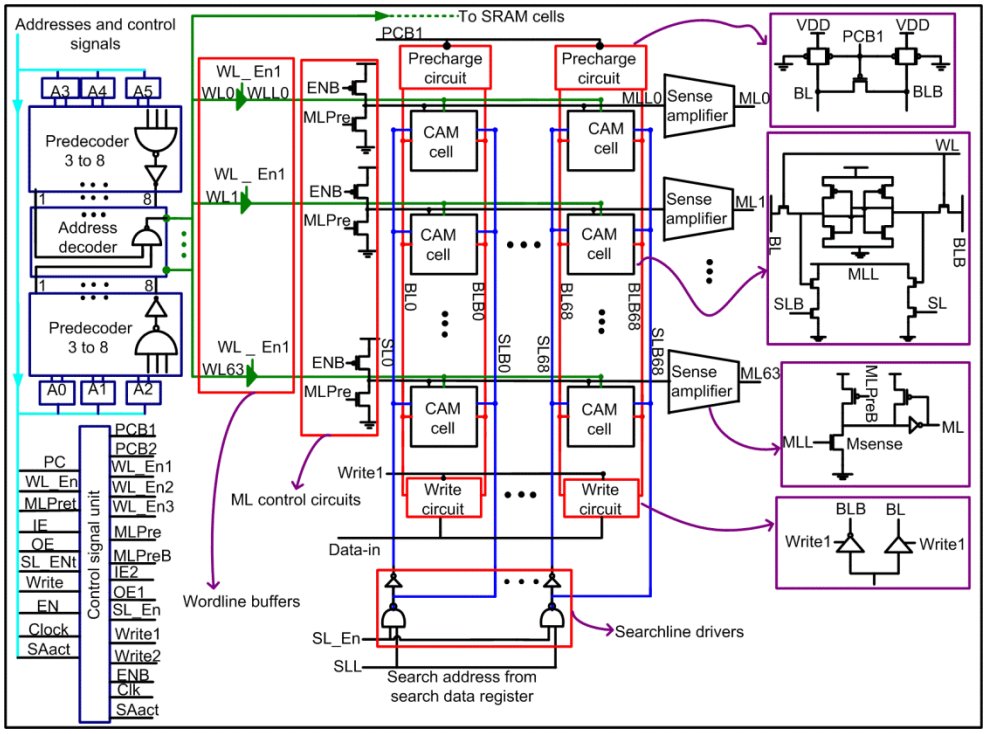
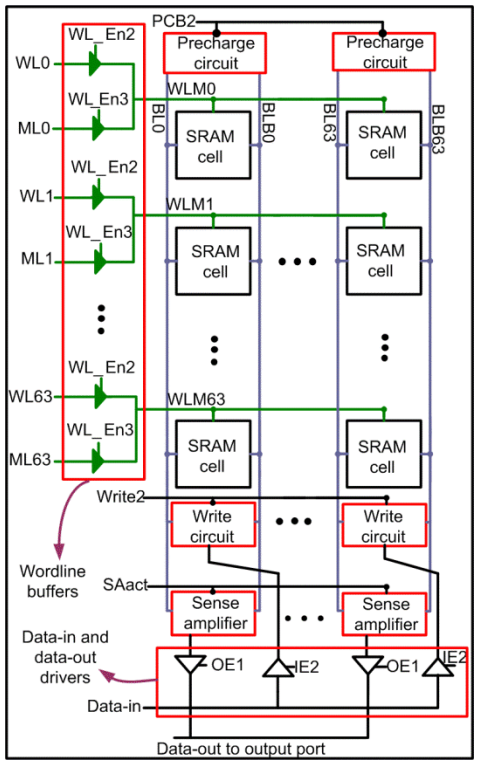


Figure 5.4: Simplified view of a full associative translation look aside buffer.



(a)



(b)

Figure 5.5: The circuit details of a typical (a) CAM and (b) RAM architecture in a TLB structure.

5.2.7.1 The Circuit Design of CMOS-only TLB

Figure 5.5 depicts the circuit designs of CMOS-only TLB consists of typical CMOS CAM and RAM architectures. It is similar to the NEMsCAM TLB structure, except that its CAM cells are implemented with CMOS only cells. Also, NEMsCAM TLB does not need any BL precharge circuits for the CAM array. Here in this thesis, we only describe the NEMsCAM TLB structure details.

5.3 Design of NEMsCAM Cell

In this section, we present the circuit details of our proposed NEMsCAM cell. We use the memory structure proposed in [114] to implement the storage part of NEMsCAM. That memory structure provides full-select behavior which are essential to design a CAM cell; it also uses electrostatic pull-out and pull-in and does not require a cell selector device in the write path. The non-volatile memory design is based on the NEM switch described in [111] which has the ability to eliminate net-disturbing electrostatic force. Figure 5.6(a) shows the schematic of the NEMsCAM cell. The outputs, Out and OutB, are connected to the transistors of the comparison circuit. We choose CMOS for the comparison circuit to avoid the long delay of the NEMs that occurs due to the mechanical movement of the beams, and that would slow down the search operation. Figure 5.7(a) represents the schematic of our NEM memory cell when it is programming to state ¨1¨. Figure 5.7(b) shows its switch model and Figure 5.7(c) shows a simplified NEM Verilog-A model between BL (source), Out (drain) and WL nodes which we employ in our circuit analysis [93]. Other switches of NEM memory cell comply with similar switch modeling.

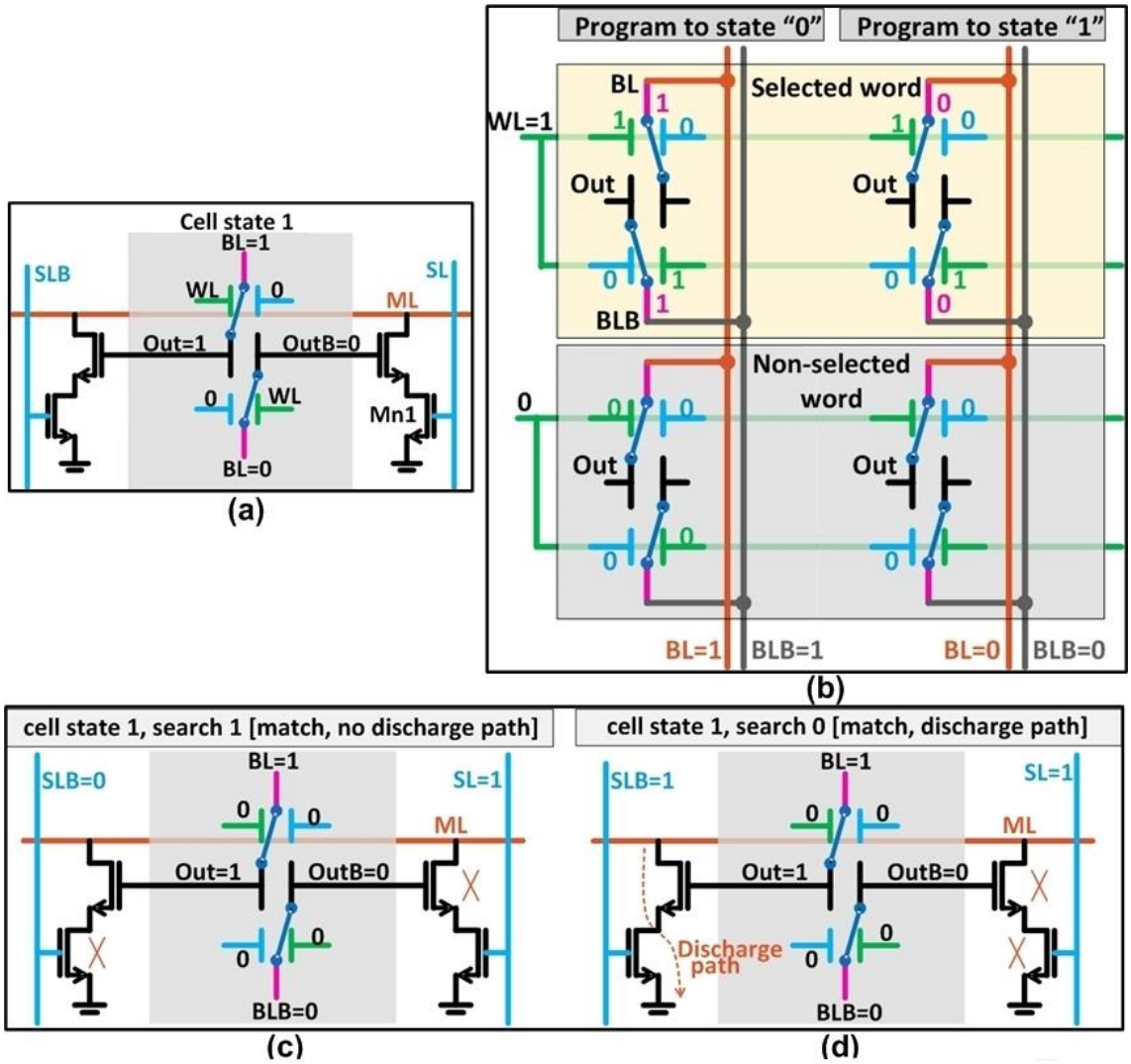


Figure 5.6: (a) Schematic of the proposed NEMsCAM cell. (b) NEMsCAM storage array organization and writing scheme. (c) When the NEMsCAM cell content matches the value being searched, there is no discharge path through the cell. (d) In case of a mismatch, the cell tries to discharge the match line ML.

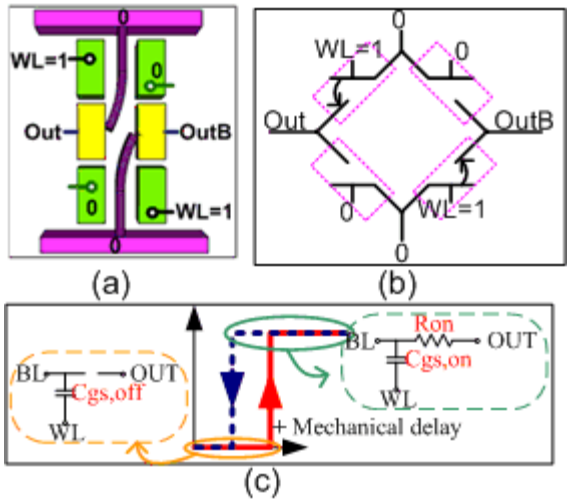


Figure 5.7: (a) Schematic of our NEM memory cell. (b) Simplified switch model of NEM memory cell (c) A simple NEM Verilog-A model between BL (source), Out (drain) and WL nodes [93].

5.3.1 Circuit Operations

5.3.1.1 Write Biasing Scheme

Figure 5.6(b) shows how the storage part of the NEMsCAM cell is written. Once the wordline (WL) is activated, all beams on this row are sensitized. For those columns whose cells are to be programmed to 0, the bitlines are set to high (BL=BLB=1), while for those whose cells are to be programmed to 1, BL=BLB=0 is applied.

No cell suffers half-disturb conditions, and there is also no risk of short-circuit current running through the switches, as BL and BLB are always at the same potential during beam switching. This is important, because high currents through contact between the beam and the drain can be a source of device failure. During normal operation, BL is put at 1 and BLB at 0. Cells whose beams are in state 0 hence have Out=0 and OutB=1. Note that there is no separate read operation in this NEM memory design because there is no mechanical switch latency in the read path.

5.3.1.2 Search operation

Figure 5.6(c) and Figure 5.6(d) illustrate a cell that matches/mismatches the search data respectively. If a cell(s) that is connected to a wordline causes a mismatch, that cell tries to discharge the entire ML which is associated with that wordline, indicating an overall mismatch.

5.3.2 Cell Architecture

Figure 5.8 presents the three-dimensional view of two adjacent NEMsCAM cells located in the same column index of the CAM array. Since NEMs have the potential to be fully integrated with CMOS devices [123], we place them on top of the CMOS layer and substantially reduce the layout area. The SL wires run parallel to the BL wires, while the matchlines (ML) and wordlines (WL) are orthoghonal to the BLs. By employing vertical NEM switches [109], the requirement of a long beam has little impact on the layout area, as it is out-of-plane. Two Gate1s are aligned and connected to their corresponding WL while the two Gate2s are connected to 0.

The drains are connected from the opposite sides and form a cross shape. Note that the BL and WL wires can be merged with the actual device terminals, resulting in a compact layout. Finally, the Vias connect Out and OutB to the CMOS layer which is located below the NEMs. Due to this organization, our proposed NEMsCAM cell reduces the wire lengths which considerably reduces the energy consumption along with the near-zero leakage characteristic of NEMs.

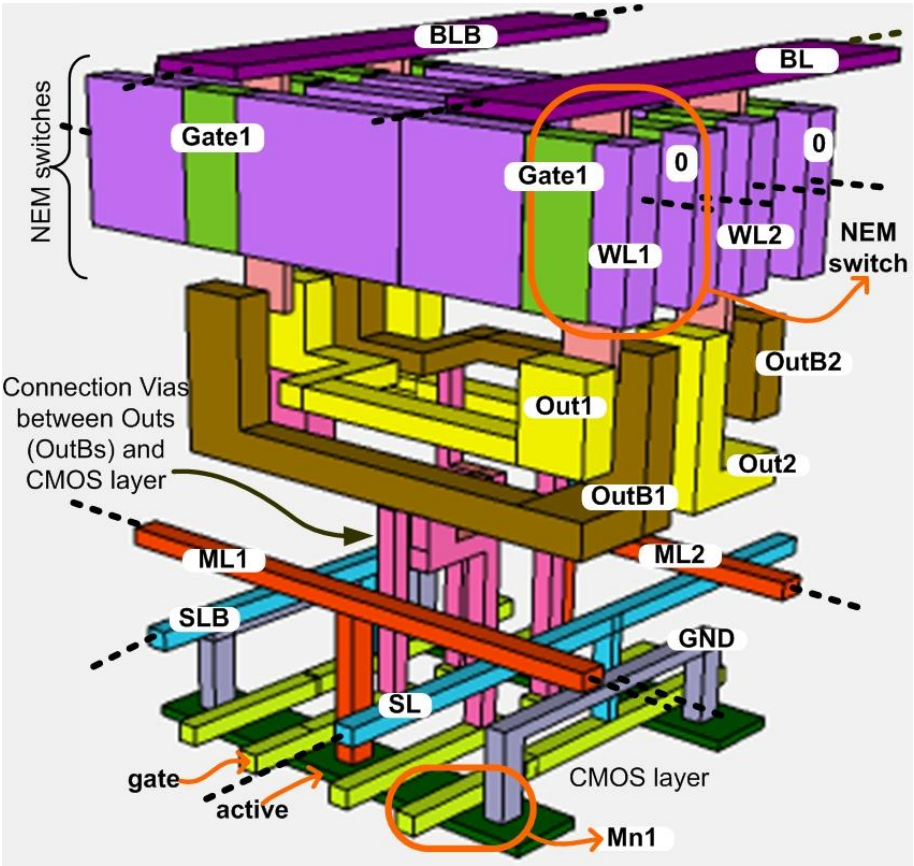


Figure 5.8: Three-dimensional view of two adjacent NEMsCAM cells in a CAM array.

5.4 A Use Case for NEMsCAM: TLB

As mentioned before, we leverage the design of the proposed NEMsCAM cell to build a fully associative Translation Lookaside Buffer (TLB), called NEMsCAM TLB. In this section, we first elaborate on the motivation behind it and then we describe the design details and the circuit operations.

5.4.1 Motivation

Due to the criticality of the TLB in the system's performance, processor vendors have employed a two-level TLB organization [124]. The first-level TLB is small, fully-associative and features a very fast search operation, while the second level TLB is large and aims at holding as many translations as possible. To boost the system's performance further, processors provide separate TLBs for data and instructions [124].

The TLB hierarchy has been reported to account for an important percentage of the energy spent in the chip [125], [126]. Intel recently reported that 13% of the total core power comes from the TLBs for memory-intensive workloads [101]. Based on our evaluation infrastructure (Section 5.5), we find that the TLB energy is overwhelmingly dominated by the first-level TLBs in terms of accesses across the TLB hierarchy (Figure 5.9). Moreover, by breaking down the energy consumption in the first-level TLBs, we find that the CAM part contributes by 94%. To reduce this source of energy consumption without affecting the performance, we leverage the NEMsCAM cell to design a highly energy-efficient first level TLB.

Per-core TLB Organization		
Level 1	Data (DTLB)	64 entries, fully assoc.
	Instruction (ITLB)	48 entries, fully assoc.
Level 2	Data (L2-DTLB)	1024 entries, 4-way assoc.
	Instruction (L2-ITLB)	512 entries, 4-way assoc.

Table 5.2: TLB organization of a modern processor [124].

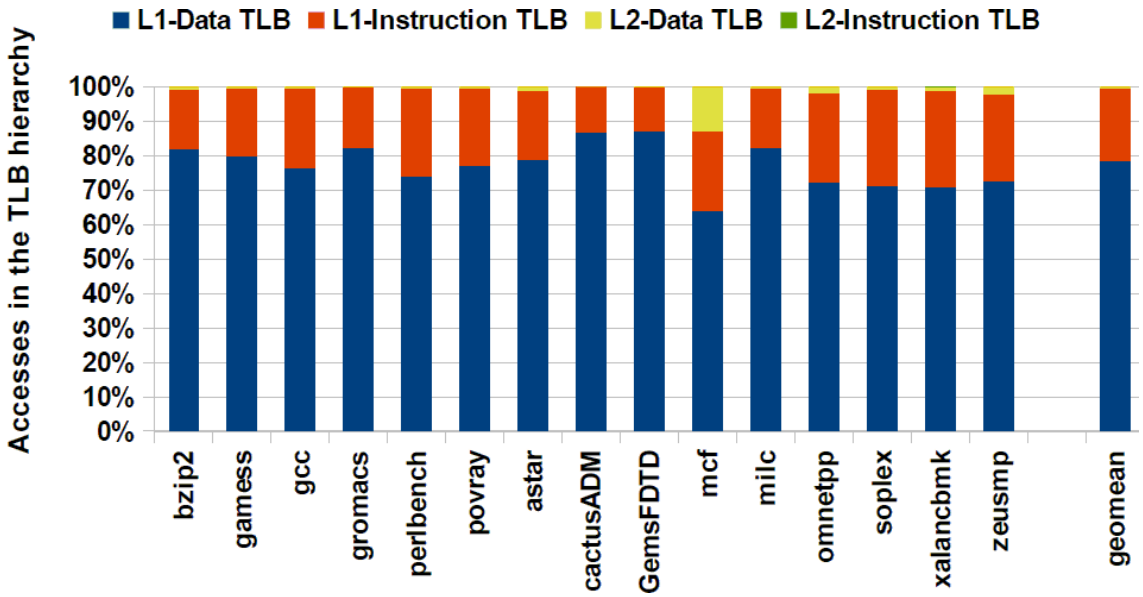


Figure 5.9: Breakdown of accesses in the TLB hierarchy.

5.4.2 Design

We design the NEMsCAM TLB with our proposed CAM cell and with typical SRAM memory circuits (Figure 5.11). The CAM part (Figure 5.11(a)) consists of the NEMsCAM cells and the necessary peripheral circuitry optimized for both search and write operations. Similarly, the SRAM cells (Figure 5.11(b)) and the associated circuits are designed with CMOS technology. The control signal unit consists of the necessary inverter chains that generate the signals to control the TLB circuits so that the search and the write operations are performed correctly.

The address decoder, the write circuits, and the data-in drivers are used only for the write operation; however, the rest of the circuits are designed to be used during the search operation as well. BL and BLB are driven with predefined signals according to the operations. The control circuit unit is added to generate the necessary Gate1 and Gate2 signals during the search and write operations.

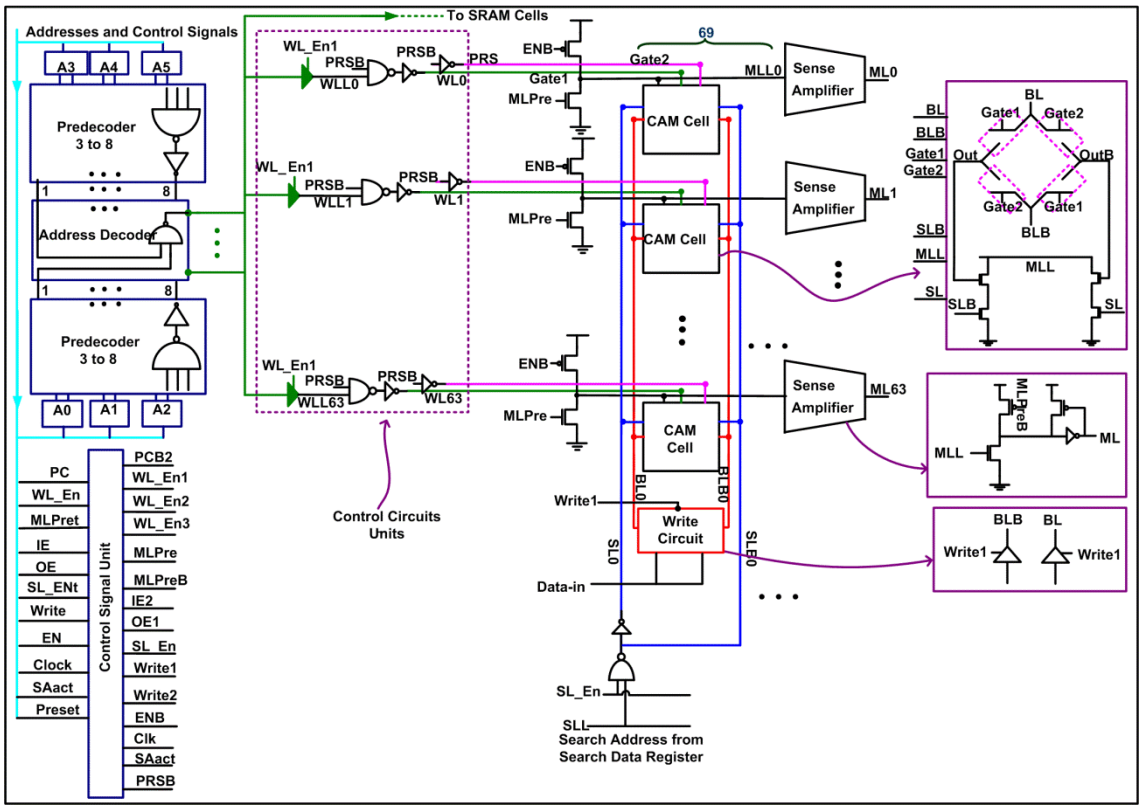
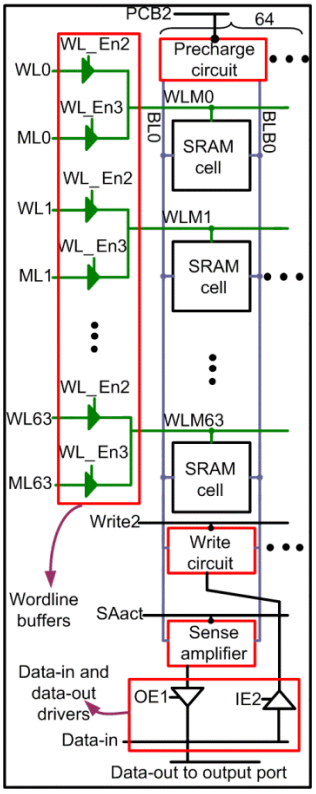


Figure 5.10 (a)



(b)

Figure 5.11: The circuit detail of (a) the proposed NEM-CMOS CAM and (b) a typical SRAM architecture in the proposed TLB structure.

5.4.2.1 Search Operation

During the search operation, WLen3 is high and the ML lines are connected to the WLM lines. At the beginning of each search operation, all MLs are put temporarily in the precharged state as in a CMOS-only TLB. The search cycle starts when the precharge signal (MLpre) is high driving the ML to low. Concurrently, the SLs are charged to their data value; with this method there is no need for a separate SL precharge phase. After this (completion of the precharge phase), the ENB signal becomes low and supplies the ML with the current source. During the evaluation phase, the stored bits of the CAM cells are compared against the data provided on the corresponding SLs.

In case of a match (TLB hit) the current source enabled by ENB pulls ML up and the ML voltage changes to high state. The state of each ML row is sensed and improved by a ML sense amplifier. Our used sense-amplifier can be seen in this Figure: an nMOS

transistor, and also a half-latch circuit which store the output data (Figure 5.11(a)). We choose the current-race scheme among various matchline sensing techniques due to its simplicity and the average-low ML energy consumption [94]. Alternatively, in case of a mismatch (TLB miss) the cell(s) that cause a mismatch counteract the current source and keep ML close to ground level. In the match case, matchline trips the half-latch circuit when it is charged to a voltage slightly higher than threshold voltage of Msense; whereas, it leaves the latch in its initial state in the mismatch case when remains at a much lower voltage [109], [83]. Finally, the ML sense amplifiers feed the wordline buffers mapping the match location to its corresponding encoded address as stored in the SRAM cells (Figure 5.11(a)). Figure 5.13 summarizes the signal behavior of the matching case for a cell of the NEMsCAM TLB.

5.4.2.2 Write Operation

During the write operation, the WLen1 and WLen2 are high, the WL which is generated in address decoder is routed to the CAM and SRAM parts, and the data is written into the corresponding cells.

5.5 Experimental Evaluation

In this section, we first describe our methodology to evaluate the NEMsCAM TLB, and then we present the results.

5.5.1 Methodology

We design NEMsCAM-based TLBs for data (DTLB) and instruction (ITLB) accesses based on [94] using the TLB organization of a modern AMD server-oriented processor [124] (Table 5.2). For both NEMsCAM and CMOS-only TLB, we construct the transistor level netlists with all the necessary circuitries, equivalent capacitances and resistance of wires. We simulate and optimize both TLB structures with Cadence Spectre using 16nm Predictive Technology Model [22] at T=25°C targeting 2GHz processor frequency. As indicated before, for the NEM switches, we use a simple Verilog-A model (Figure 5.7) with the following parameters: Vpi=0.8V, Vpo=0.2V, Cgs-off =15aF, Cgs-on=20aF, tmech=3ns [93]. We optimize our circuits to minimize the energy consumption. We verified that the search and write operations are executed correctly

fulfilling the timing requirements. We also calculate the energy consumption per search and write operation, and standby mode. Furthermore, we draw the layouts [23], and measure the wire lengths and optimize the wire capacitances in the netlists. Finally, to estimate the impact of the NEMsCAM TLBs at system's performance, we use the Sniper simulator [127] with the configuration of Table 5.2, and run the TLB-intensive workloads from Spec2006 with the reference input set and execute for one billion instructions.

5.5.2 Results

5.5.2.1 Energy & Area

Table 5.3 shows the simulation results for both NEMsCAM and CMOS-only TLBs. First, we observe that the area reduces by 40.5% for the DTLB (Figure 5.12). The reason for this improvement is thanks to the novel structure of the NEMsCAM cell. Second, we observe that the energy per search and write operation and standby mode reduces by 27%, 41.9% and 53.7% respectively for the DTLB. This happens due to the lower dimensions of the circuit leading to lower parasitic wire capacitances and resistances on the searchlines and the matchlines which in turn require fewer driving buffers. Moreover, the energy consumption further reduces due to the near-zero leakage current that NEMs provide. Similar results hold for the ITLB as well.

Structure	Metric	CMOS	NEMs	Benefit (%)
DTLB 64 entries	Search oper. (pJ)	4.529	3.308	27.0
	Write oper. (pJ)	0.148	0.086	41.9
	Standby mode (pJ)	0.141	0.065	53.7
	Area (normalized)	1.000	0.595	40.5
ITLB 48 entries	Search oper. (pJ)	3.658	2.805	23.3
	Write oper. (pJ)	0.187	0.107	42.8
	Standby mode (pJ)	0.106	0.046	55.9
	Area (normalized)	1.000	0.661	33.9

Table 5.3: Energy consumption for search, write operations and standby mode and normalized area footprint.

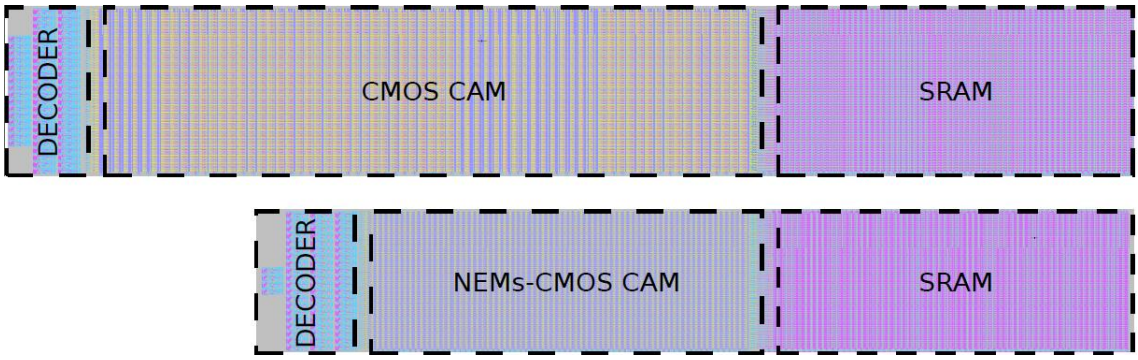


Figure 5.12: Layout of the DTLB implemented with CMOS-only CAM cells (top), and NEM-based CAM cells (bottom).

5.5.2.2 Latency

Figure 5.13 shows the simulation waveform of the matching state for a cell of the NEMsCAM DTLB during the search operation. The waveform verifies that our design meets the target time requirement of one clock cycle per search operation. On the other hand, the write operation takes 6 cycles in the NEMsCAM TLB (based on [93]) while it takes 2 cycles in the CMOS-only TLB. This slowdown is due to the long mechanical delay of the NEM switches. However, this latency barely affects the processor performance as shown next.

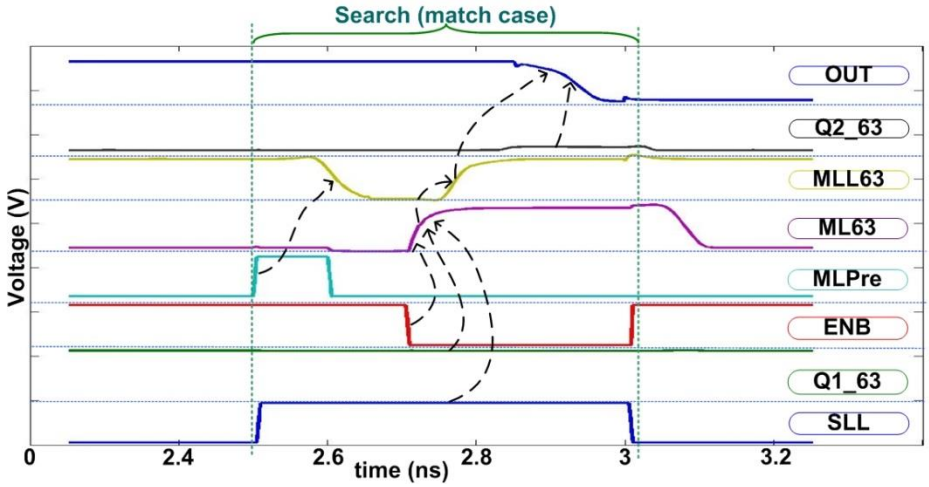


Figure 5.13: Simulation waveform of matching state for the first cell at the last row of the NEMsCAM DTLB.

5.5.2.3 System

Figure 5.14 shows the energy reduction in the first level TLBs due to the employment of NEMsCAM for various workloads. We find that the search operation dominates in the energy breakdown for both DTLB and ITLB, and that the NEMsCAM TLBs reduce the energy spent by 28.7% on average. Taking into account that 13% of the total core power comes from the TLBs [101], the NEMsCAM cell can significantly help in reducing the total chip's energy efficiency. Figure 5.15 shows the estimated execution overhead due to the employment of the NEMsCAM TLBs. This overhead comes from the increased latency of the write operation in NEMsCAM. However, the write operation: (i) takes place only after TLB misses which occur rarely compared to TLB hits, and (ii) adds latency to an already slow operation, i.e. L2-TLB access (~7 cycles [128]), including potentially the penalty of L2-TLB miss (~100s cycles [122]). Consequently, the NEMsCAMs TLB have negligible impact on the execution time for most applications (0.27% on average) while reducing significantly the energy spent in the TLB hierarchy.

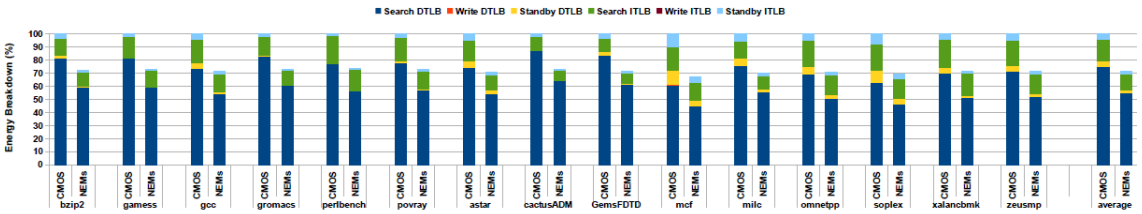


Figure 5.14: Dynamic energy consumption of the CMOS-only and the NEMsCAM DTLB and ITLB.

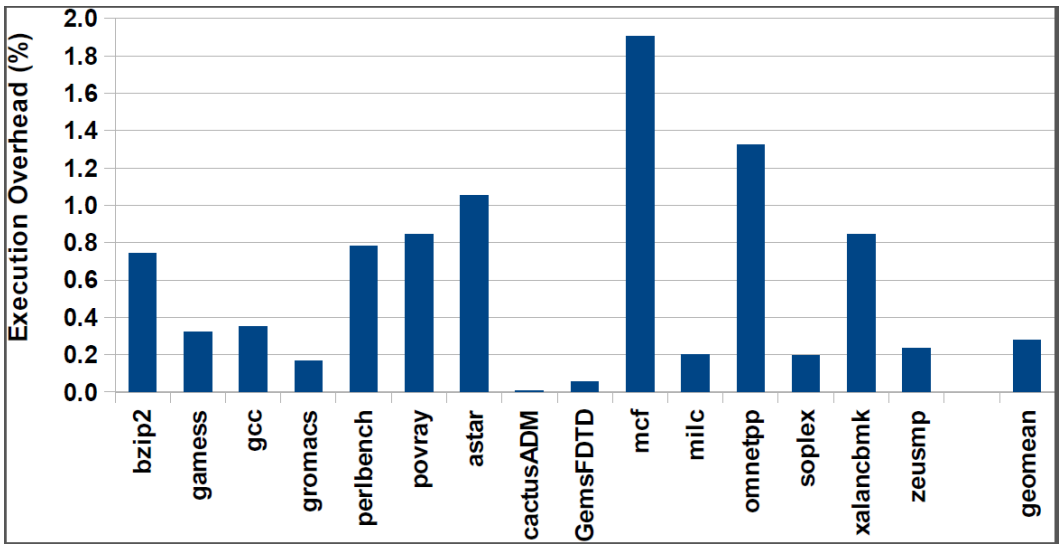


Figure 5.15: Execution time overhead due to NEMsCAM TLBs.

5.6 Related Work

In this section we focus our discussion on leveraging NEMs compared to other state of the art technologies in order to design a CAM cell for performance critical structures such as first-level TLBs. Eshraghian et al. proposed a CAM design based on memristors [95]. The memristor provides high density capability. However, it suffers from increased search latency -compared to NEMs -and active energy consumption because it requires a higher voltage for search and write operations. Therefore, we conclude that memristors appear to be inappropriate for designing power-hungry and time critical components like CAM for TLBs.

Rajendran et al. designed a CAM cell with PCRAM [98]. PCRAM exhibits relatively high area density but -similar to memristors- at the cost of higher search latency, higher write energy consumption and lower endurance. CAM cells based on tunnel junction (MTJ) devices have been also proposed [96], [97], [99]. These cells provide high density thanks to the ability of MTJs to stack over the MOS device, similar to our NEMsCAM cell. However, MTJs suffer from high write power consumption and difficulties in scalability compared to NEMs.

Finally, Liu et al. [129] proposed STT-RAM-based TLB for GPUs. The authors target a set-associative TLB that does not require any CAM cells. The STT-RAM technology is used for designing the data part of the TLB. In contrast, we design NEM-based CAM cells and we leverage them to build the tag part of fully-associative TLBs. Their design provides greater capacity, similar read performance and lower energy consumption compared to CMOS only TLBs. Furthermore, STT-RAM TLB imposes longer delay per search operation while our proposed NEMsCAM TLB accomplishes each search operation in one clock cycle. The authors suggest getting the benefits of differential sensing techniques to enhance the read speed of TLB by trading off the capacity. In one sense, they can mitigate the read delay, but on the other side, the write dynamic energy has doubled and capacity has halved, because the data is written into two adjacent cells instead of one cell during high performance mode. Also, the extra circuitries added to fulfill differential sensing techniques consume extra energy and occupy more area footprint. In contrast to these proposals, NEMsCAM exhibits high improvement in terms

of low search latency, area, and off-state leakage current thanks to the outstanding characteristics of NEMs allowing the design of CAM cells for performance critical structures.

5.7 Summary

In this thesis chapter, we propose the NEMsCAM cell that combines both NEMs and CMOS to design low power and highly efficient processor structures such as TLBs. Our analysis shows that the NEMsCAM TLB exhibits significant benefits over the CMOS-only TLB in terms of energy consumption and area. However, the limited write endurance of current NEMs may delay their adoption until technology improves.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The desire to achieve the super-fast and minimal-size microprocessors has led to the considerable progress in computer systems. Many new alternative devices for CMOS generation have been exploited that allow transistors scale much further and let Moore's Law continue more. Furthermore, multicore processors have been invented to enhance the performance and reduce the power consumption. However, these mentioned progresses suffer from unwanted problems such as power consumption increase, failure rate increase and performance issues which entail further investigation to mitigate these problems. In this thesis, we face with these challenges and try to handle them in the circuit level by proposing some techniques in order to improve the reliability, speed, and energy consumption.

In Chapter 2, we introduce a novel memory cell (dvSRAM) which has the ability to keep two versions of the same logical data simultaneously. Then, we propose a new level-one cache design based on our proposed cell. Each dvSRAM cell consists of two cell, main cell and secondary cell which keep two versions of the same data, new value and the old value. These values can be accessed, modified, moved back and forth between the main and secondary cells within the access time of the cache using available operations. We introduce three well known use cases and evaluate one of them, Hardware Transactional Memory, to show our design impact in terms of performance and energy consumption. Our experiments show that the dvSRAM allows for significant performance gains with acceptable costs in power, delay and area. Also we show that with a few modifications, the dvSRAM array can be presented as a reconfigurable array which provides two execution modes, a general purpose mode and optimistic concurrency mode.

In Chapter 3, we propose two novel bias control circuits to manage the power consumption of inactive cache cells in data retention mode. The first proposed circuit (DB-Control circuit) dynamically tracks the reference current and sets the bias voltage of cells, while the second (SAB-Control circuit) has a self-adjust property to set the bias voltages and also alleviates the instability problems appear due to noise injection. In

order to show their positive impact on performance improvement, we add them to a dvSRAM cache and simulate and optimize the entire cache circuit. The simulations demonstrate the effectiveness of our proposed circuits to considerably reduce the energy consumption compared to the typical dvSRAM cache with a modest area increase and negligible delay overhead. We also show that instability problems are alleviated by using the SAB-Control circuit.

In Chapter 4, we propose a novel Flexible and reliable L1 data cache circuit design (Flexicache) with the unique capability of automatically adapting itself for different supply voltage levels and providing the highest reliability. Depending on the supply voltage level, Flexicache defines three operating modes: In high supply voltages, Flexicache provides reliability through single-bit parity. In middle range of supply voltages, Flexicache writes data to two separate cache-lines simultaneously in order to use one line for error recovery when the other line is faulty. In near threshold supply voltages, Flexicache writes data to three separate cache-lines simultaneously in order to provide the correct data based on bitwise majority voter. According to our simulation results, Flexicache provides a cache with a higher capacity in low-power mode with significantly less energy consumption compared to the state of the art proposals.

In Chapter 5, we propose a novel CAM cell called NEMSCAM which is designed based on both NEMs and CMOS technologies. The memory part of the NEMsCAM cell is constructed with two vertical complementary non-volatile NEM switches, while the comparison circuits are designed with CMOS transistors. As a use case, we leverage the NEMsCAM cell to build first-level TLBs for both data and instruction accesses that complete the search operation in one clock cycle. The TLB has been pointed out as a critical component of energy and performance in modern processors. Our analysis shows that the NEMsCAM TLB exhibits significant benefits over the CMOS-only TLB in terms of energy consumption per search/write operation, standby mode and also area. However, the limited write endurance of current NEMs may delay their adoption until technology improves.

6.2 Future Work

Some of the subjects investigated in this dissertation can be further pursued in future, more specifically, the topics described in Chapter 2, Chapter 3, and Chapter 5 propose clear future research directions that might be worth exploring.

In Chapter 2 we have proposed a L1 data cache designed with dvSRAM cells that applies in optimistic concurrency proposals. In spite of being effective in some recent architecture implementations such as Transactional Memory, it is not applicable in low scale technologies because of CMOS scalability limitation. As a future research topic, we can design our mentioned dual versioning cell with appropriate non-volatile devices like PCRAM which have the ability to save more than one bit at each memory cell. Even with proper manipulation and well design circuitries, it is possible to save three or more bit at each cell at the expense of latency and power consumption. We will propose to design such a dual and triple versioning cell with nonvolatile memories and alleviate the latency and power consumption limitation by applying some available circuit techniques. As an opportunity for future work, the DB-Control circuit which has been introduced in Chapter 3 could be applied to each cache line instead of the entire sub-array, which would lead to a finer granularity power management control. The DB-Control circuits designed for each cache line would have much smaller size than the one we propose for the entire sub-array, keeping the total area overhead low.

Our analysis in Chapter 5 shows that the NEMsCAM TLB exhibits significant benefits over the CMOS-only TLB in terms of energy consumption and area. However, the limited write endurance of current NEMs may delay their adoption until technology improves. For future work, we consider on investigating techniques at both the circuit and architecture levels to improve the expected life-time of such structures under near-future technology constraints.

A possible solution to mitigate reliability issues is to maximize the endurance by reducing the number of writes to each non-volatile cells; we suggest to use some existing gating technique at fine-grain circuit level to eliminate unnecessary turning on and off during write time: By adding each nonvolatile memory word-line to an enabler transistor

which is located in series with all cells, it is possible to control the number of accesses to each word-line. We can turn off each word-line whenever the number of writes is approaching the endurance limitation. Thus, we can optimize the memory capacity usage and get benefit of new nonvolatile technologies.

CHAPTER 7

PUBLICATION LIST AND BIBLIOGRAPHY

The content of this thesis led to following publications:

A. Seyedi, A. Armejach, A. Cristal, O. S. Unsal, I. Hur, and M. Valero, *Circuit Design of a Dual-versioning L1 Data Cache for Optimistic Concurrency*, in Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI, 2011, pp. 325–330.

A. Seyedi, A. Armejach, A. Cristal, O. S. Unsal, I. Hur, and M. Valero, *Circuit Design of a Dual-versioning L1 Data Cache*, Integration on VLSI Journal, vol. 45, no. 3, pp. 237–245, 2012.

A. Armejach, A. Seyedi, R. Titos-Gil, I. Hur, A. Cristal, O. S. Unsal, and M. Valero, *Using a Reconfigurable L1 Data Cache for Efficient Version Management in Hardware Transactional Memory*, in Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on, 2011, pp. 361–371.

Azam Seyedi, Adria Armejach, Adrián Cristal, Osman S. Unsal, Mateo Valero, *Novel SRAM Bias Control Circuits for a Low Power L1 Data Cache* , The 30th NORCHIP conference, Nov 2012

Azam Seyedi, Gulay Yalcin, Osman S. Unsal, Adrián Cristal, *Circuit Design of a Novel Adaptable and Reliable L1 Data Cache*, In 23rd Great Lakes Symposium on Very Large Scale Integration (GLSVLSI'13) - May 2013.

Gulay Yalcin, Azam Seyedi, Osman Unsal, Adrián Cristal, *Flexicache: Highly Reliable and Low Power Cache under Supply Voltage Scaling*, CARLA Latin American High Performance Computing Conference - Oct 2014.

G Yalcin, A Seyedi, OS Unsal, A Cristal, *Flexicache: Highly Reliable and Low Power Cache under Supply Voltage Scaling*, High Performance Computing, 173-190., 2014.

Azam Seyedi, Vasileios Karakostas, Stefan Cosemans, Adrián Cristal, Mario Nemirovsky , Osman Unsal, *NEMsCAM: A Novel CAM Cell based on Nano-Electro-Mechanical Switch and CMOS for Energy Efficient TLBs*, IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH´15) - Jul 2015

References:

[1] K. Olukotun and L. Hammond, "The Future of Microprocessors," ACM, New York, NY, USA, 2005.

[2] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[3] S. G. Narendra and A. P. Chandrakasan, *Leakage in nanometer CMOS technologies*. Springer Science & Business Media, 2006.

[4] B. Jacob and T. Mudge, "Virtual Memory: Issues of Implementation," *Computer (Long. Beach. Calif).*, vol. 31, no. 6, pp. 33–43, 1998.

[5] M. Herlihy and J. E. B. Moss, "Transactional Memory: Architectural Support for Lock-free Data Structures," in *Proceedings of the 20th Annual International Symposium on Computer Architecture*, 1993, pp. 289–300.

[6] T. Harris, J. Larus, and R. Rajwar, *Transactional Memory, 2Nd Edition*, 2nd ed. Morgan and Claypool Publishers, 2010.

[7] J. Warnock, Y. Chan, W. Huott, S. Carey, M. Fee, H. Wen, M. J. Saccamango, F. Malgioglio, P. Meaney, D. Plass, Y. Chan, M. Mayo, G. Mayer, L. Sigal, D. Rude, R. Averill, M. Wood, T. Strach, H. Smith, B. Curran, E. Schwarz, L. Eisen, D. Malone, S. Weitzel, P. Mak, T. McPherson, and C. Webb, "A 5.2GHz microprocessor chip for the IBM zEnterprise #x2122; system," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, pp. 70–72.

[8] G. K. Chen, D. Blaauw, T. Mudge, D. Sylvester, and N. S. Kim, "Yield-driven Near-threshold SRAM Design," in *Proceedings of the 2007 IEEE/ACM International Conference on Computer-aided Design*, 2007, pp. 660–666.

[9] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proc. IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[10] International Technology Roadmap for Semiconductors, "Emerging Research

Devices, 2013 Edition.” .

[11] H. Chafi, J. Casper, B. D. Carlstrom, A. McDonald, C. C. Minh, W. Baek, C. Kozyrakis, and K. Olukotun, “A Scalable, Non-blocking Approach to Transactional Memory,” in *HPCA*, 2007, pp. 97–108.

[12] L. Yen, J. Bobba, M. R. Marty, K. E. Moore, H. Volos, M. D. Hill, M. M. Swift, and D. A. Wood, “LogTM-SE: Decoupling Hardware Transactional Memory from Caches,” in *Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture*, 2007, pp. 261–272.

[13] C. H. Kim, J.-J. Kim, S. Mukhopadhyay, and K. Roy, “A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations,” *Very Large Scale Integr. Syst. IEEE Trans.*, vol. 13, no. 3, pp. 349–357, 2005.

[14] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, “Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-submicron Cache Memories,” in *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, 2000, pp. 90–95.

[15] M. Khellah, D. Somasekhar, Y. Ye, N. S. Kim, J. Howard, G. Ruhl, M. Sunna, J. Tschanz, N. Borkar, F. Hamzaoglu, G. Pandya, A. Farhang, K. Zhang, and V. De, “A 256-Kb Dual-VCC SRAM Building Block in 65-nm CMOS Process With Actively Clamped Sleep Transistor,” *Solid-State Circuits, IEEE J.*, vol. 42, no. 1, pp. 233–242, 2007.

[16] M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F. Witulski, J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N. Damoulakis, “Models and Algorithmic Limits for an ECC-Based Approach to Hardening Sub-100-nm SRAMs,” *Nucl. Sci. IEEE Trans.*, vol. 54, no. 4, pp. 935–945, 2007.

[17] G. Yalcin, “Designs for increasing reliability while reducing energy and increasing lifetime,” Universitat Polit{è}cnica de Catalunya, 2014.

[18] V. Krishnan and J. Torrellas, “A Chip-Multiprocessor Architecture with Speculative Multithreading,” *IEEE Trans. Comput.*, vol. 48, no. 9, pp. 866–880, 1999.

[19] R. Rajwar and J. R. Goodman, “Speculative Lock Elision: Enabling Highly Concurrent Multithreaded Execution,” in *Proceedings of the 34th Annual*

ACM/IEEE International Symposium on Microarchitecture, 2001, pp. 294–305.

[20] A. Seyedi, A. Armejach, A. Cristal, O. S. Unsal, I. Hur, and M. Valero, "Circuit Design of a Dual-versioning L1 Data Cache for Optimistic Concurrency," in *Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI*, 2011, pp. 325–330.

[21] A. Seyedi, A. Armejach, A. Cristal, O. S. Unsal, I. Hur, and M. Valero, "Circuit Design of a Dual-versioning L1 Data Cache," *Integr. VLSI J.*, vol. 45, no. 3, pp. 237–245, 2012.

[22] "Predictive Technology Model." .

[23] "The Electric VLSI Design System." .

[24] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.

[25] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar, "Scaling of Stack Effect and Its Application for Leakage Reduction," in *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, 2001, pp. 195–200.

[26] Y. Ye, S. Borkar, and V. De, "A new technique for standby leakage reduction in high-performance circuits," in *VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on*, 1998, pp. 40–41.

[27] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[28] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "CACTI 5.1," *HP Lab. April*, vol. 2, p. 24, 2008.

[29] "CACTI5.3." .

[30] Y. Wang, H. J. Ahn, U. Bhattacharya, Z. Chen, T. Coan, F. Hamzaoglu, W. M. Hafez, C.-H. Jan, P. Kolar, S. H. Kulkarni, J.-F. Lin, Y.-G. Ng, I. Post, L. Wei, Y. Zhang, K. Zhang, and M. Bohr, "A 1.1 GHz 12 #956;A/Mb-Leakage SRAM

Design in 65 nm Ultra-Low-Power CMOS Technology With Integrated Leakage Reduction for Mobile Applications," *Solid-State Circuits, IEEE J.*, vol. 43, no. 1, pp. 172–179, 2008.

[31] B. S. Amrutur and M. A. Horowitz, "Fast low-power decoders for RAMs," *Solid-State Circuits, IEEE J.*, vol. 36, no. 10, pp. 1506–1515, Oct. 2001.

[32] Technical Report IBM Application Notes, IBM, "Understanding static RAM operation,1997." .

[33] B. S. Amrutur, "Design and analysis of fast low power SRAMs," Stanford University, 1999.

[34] A. F. Yeknami, "Design and evaluation of a low-voltage, process-variation-tolerant SRAM cache in 90nm CMOS technology," *Master's Thesis, Linkoping Univ. Sweden*, 2008.

[35] S. Cosemans, W. Dehaene, and F. Catthoor, "A Low-Power Embedded SRAM for Wireless Applications," *Solid-State Circuits, IEEE J.*, vol. 42, no. 7, pp. 1607–1617, 2007.

[36] J. Pille, D. Wendel, O. Wagner, R. Sautter, W. Penth, T. Froehnel, S. Buettner, O. Torreiter, M. Eckert, J. Paredes, D. Hrusecky, D. Ray, and M. Canada, "A 32kB 2R/1W L1 data cache in 45nm SOI technology for the POWER7TM processor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 344–345.

[37] A. Armejach, A. Seyedi, R. Titos-Gil, I. Hur, A. Cristal, O. S. Unsal, and M. Valero, "Using a Reconfigurable L1 Data Cache for Efficient Version Management in Hardware Transactional Memory," in *Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on*, 2011, pp. 361–371.

[38] V. Krishnan and J. Torrellas, "A chip-multiprocessor architecture with speculative multithreading," *Comput. IEEE Trans.*, vol. 48, no. 9, pp. 866–880, Sep. 1999.

[39] C. C. Minh, J. Chung, C. Kozyrakis, and K. Olukotun, "STAMP: Stanford transactional applications for multi-processing," in *Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*, 2008, pp. 35–46.

[40] O. Ergin, D. Balkan, D. Ponomarev, and K. Ghose, “Early Register Deallocation Mechanisms Using Checkpointed Register Files,” *Comput. IEEE Trans.*, vol. 55, no. 9, pp. 1153–1166, Sep. 2006.

[41] 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*, 2009, pp. 213–221.

[42] W. S. Yu, R. Huang, S. Q. Xu, S.-E. Wang, E. Kan, and G. E. Suh, “SRAM-DRAM Hybrid Memory with Applications to Efficient Register Files in Fine-grained Multi-threading,” in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, 2011, pp. 247–258.

[43] Y. S. Yu, H. W. Kye, B. N. Song, S. J. Kim, and J. B. Choi, “A new multi-valued static random access memory (MVSRAM) with hybrid circuit consisting of single-electron (SE) and MOSFET,” in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, p. 4 pp.–4578.

[44] Y. S. Yu, H. W. Kye, B. N. Song, S.-J. Kim, and J.-B. Choi, “Multi-valued static random access memory (SRAM) cell with single-electron and MOSFET hybrid circuit,” *Electron. Lett.*, vol. 41, no. 24, pp. 1316–1317, Nov. 2005.

[45] A. Seyedi, A. Armejach, A. Cristal, O. S. Unsal, and M. R. Valero, “Novel SRAM bias control circuits for a low power L1 data cache,” in *NORCHIP, 2012*, 2012, pp. 1–6.

[46] W.-F. Wong, C.-K. Koh, Y. Chen, and H. Li, “VOSCH: Voltage scaled cache hierarchies,” in *Computer Design, 2007. ICCD 2007. 25th International Conference on*, 2007, pp. 496–503.

[47] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge, “Circuit and microarchitectural techniques for reducing cache leakage power,” *Very Large Scale Integr. Syst. IEEE Trans.*, vol. 12, no. 2, pp. 167–184, Feb. 2004.

[48] A. Agarwal, H. Li, and K. Roy, “A single-Vt low-leakage gated-ground cache for deep submicron,” *Solid-State Circuits, IEEE J.*, vol. 38, no. 2, pp. 319–328, Feb. 2003.

[49] K. Zhang, F. Hamzaoglu, and Y. Wang, “Low-Power SRAMs in Nanoscale

CMOS Technologies," *Electron Devices, IEEE Trans.*, vol. 55, no. 1, pp. 145–151, 2008.

[50] A. J. Bhavnagarwala, S. V Kosonocky, M. Immediato, D. Knebel, and A.-M. Haen, "A pico-joule class, 1 GHz, 32 KByte/spl times/64 b DSP SRAM with self reverse bias," in *VLSI Circuits, 2003. Digest of Technical Papers. 2003 Symposium on*, 2003, pp. 251–252.

[51] Y. Takeyama, H. Otake, O. Hirabayashi, K. Kushida, and N. Otsuka, "A low leakage SRAM macro with replica cell biasing scheme," *Solid-State Circuits, IEEE J.*, vol. 41, no. 4, pp. 815–822, 2006.

[52] M. Yamaoka, Y. Shinozaki, N. Maeda, Y. Shimazaki, K. Kato, S. Shimada, K. Yanagisawa, and K. Osada, "A 300-MHz 25- mu;A/Mb-leakage on-chip SRAM module featuring process-variation immunity and low-leakage-active mode for mobile-phone application processor," *Solid-State Circuits, IEEE J.*, vol. 40, no. 1, pp. 186–194, 2005.

[53] K. Zhang, U. Bhattacharya, Z. Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Y. Wang, B. Zheng, and M. Bohr, "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *Solid-State Circuits, IEEE J.*, vol. 40, no. 4, pp. 895–901, 2005.

[54] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2001.

[55] J.-M. Redoute and M. Steyaert, "Current mirror structure insensitive to conducted EMI," *Electron. Lett.*, vol. 41, no. 21, pp. 1145–1146, Oct. 2005.

[56] B. G. Song, O. J. Kwon, I. K. Chang, H. J. Song, and K. D. Kwack, "A 1.8 V self-biased complementary folded cascode amplifier," in *ASICs, 1999. AP-ASIC '99. The First IEEE Asia Pacific Conference on*, 1999, pp. 63–65.

[57] H. Yoshida, K. De, and V. Boppana, "Accurate pre-layout estimation of standard cell characteristics," in *Design Automation Conference, 2004. Proceedings. 41st*, 2004, pp. 208–211.

[58] Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu, and S.-L. Lu, "Improving cache lifetime reliability at ultra-low voltages," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, 2009, pp. 89–

99.

[59] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, “Orthogonal Latin Square Codes,” *IBM J. Res. Dev.*, vol. 14, no. 4, pp. 390–394, 1970.

[60] T. N. Miller, R. Thomas, J. Dinan, B. Adcock, and R. Teodorescu, “Parichute: Generalized Turbocode-Based Error Correction for Near-Threshold Caches,” in *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, 2010, pp. 351–362.

[61] S. Rusu, H. Muljono, D. Ayers, S. Tam, W. Chen, A. Martin, S. Li, S. Vora, R. Varada, and E. Wang, “5.4 Ivytown: A 22nm 15-core enterprise Xeon #x00AE; processor family,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, 2014, pp. 102–103.

[62] A. Chakraborty, H. Homayoun, A. Khajeh, N. Dutt, A. Eltawil, and F. Kurdahi, “E < MC2: Less Energy Through Multi-copy Cache,” in *Proceedings of the 2010 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, 2010, pp. 237–246.

[63] W. Zhang, “Replication cache: a small fully associative cache to improve data cache reliability,” *Comput. IEEE Trans.*, vol. 54, no. 12, pp. 1547–1555, 2005.

[64] A. Seyedi, G. Yalcin, O. S. Unsal, and A. Cristal, “Circuit Design of a Novel Adaptable and Reliable L1 Data Cache,” in *Proceedings of the 23rd ACM International Conference on Great Lakes Symposium on VLSI*, 2013, pp. 333–334.

[65] G. Yalcin, A. Seyedi, O. S. Unsal, and A. Cristal, “Flexicache: Highly Reliable and Low Power Cache under Supply Voltage Scaling,” in *High Performance Computing*, Springer Berlin Heidelberg, 2014, pp. 173–190.

[66] A. Ansari, S. Gupta, S. Feng, and S. Mahlke, “ZerehCache: Armoring Cache Architectures in High Defect Density Technologies,” in *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 100–110.

[67] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, “Trading off Cache Capacity for Reliability to Enable Low Voltage Operation,” in *Proceedings of the 35th Annual International Symposium on Computer*

Architecture, 2008, pp. 203–214.

[68] J. P. Kulkarni, K. Kim, and K. Roy, "A 160 mV Robust Schmitt Trigger Based Subthreshold SRAM," *Solid-State Circuits, IEEE J.*, vol. 42, no. 10, pp. 2303–2313, Oct. 2007.

[69] M. Franklin and K. K. Saluja, "Built-in self-testing of random-access memories," *Computer (Long. Beach. Calif).*, vol. 23, no. 10, pp. 45–56, Oct. 1990.

[70] R. Baumann, "Soft errors in advanced computer systems," *Des. Test Comput. IEEE*, vol. 22, no. 3, pp. 258–266, May 2005.

[71] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," *Micro, IEEE*, vol. 23, no. 4, pp. 14–19, 2003.

[72] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. Hoe, "Multi-bit Error Tolerant Caches Using Two-Dimensional Error Coding," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, 2007, pp. 197–209.

[73] J. Abella, J. Carretero, P. Chaparro, X. Vera, and A. González, "Low Vccmin Fault-tolerant Cache with Highly Predictable Performance," in *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 111–121.

[74] B. H. Calhoun and A. Chandrakasan, "A 256kb Sub-threshold SRAM in 65nm CMOS," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, 2006, pp. 2592–2601.

[75] C. McNairy and D. Soltis, "Itanium 2 Processor Microarchitecture," *IEEE Micro*, vol. 23, no. 2, pp. 44–55, 2003.

[76] C. McNairy and R. Bhatia, "Montecito: A Dual-Core, Dual-Thread Itanium Processor," *IEEE Micro*, vol. 25, no. 2, pp. 10–20, 2005.

[77] D. J. Sorin, M. M. K. Martin, M. D. Hill, and D. A. Wood, "SafetyNet: improving the availability of shared memory multiprocessors with global checkpoint/recovery," in *Computer Architecture, 2002. Proceedings. 29th Annual International Symposium on*, 2002, pp. 123–134.

[78] M. Manoochehri, M. Annavaram, and M. Dubois, "CPPC: Correctable Parity Protected Cache," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, 2011, pp. 223–234.

[79] D. F. Heidel, P. W. Marshall, J. A. Pellish, K. P. Rodbell, K. A. LaBel, J. R. Schwank, S. E. Rauch, M. C. Hakey, M. D. Berg, C. M. Castaneda, P. E. Dodd, M. R. Friendlich, A. D. Phan, C. M. Seidleck, M. R. Shaneyfelt, and M. A. Xapsos, "Single-Event Upsets and Multiple-Bit Upsets on a 45 nm SOI SRAM," *Nucl. Sci. IEEE Trans.*, vol. 56, no. 6, pp. 3499–3504, 2009.

[80] S. Cosemans, W. Dehaene, and F. Catthoor, "A 3.6 pJ/Access 480 MHz, 128 kb On-Chip SRAM With 850 MHz Boost Mode in 90 nm CMOS With Tunable Sense Amplifiers," *Solid-State Circuits, IEEE J.*, vol. 44, no. 7, pp. 2065–2077, 2009.

[81] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.

[82] H. F. Dadgour and K. Banerjee, "Design and Analysis of Hybrid NEMS-CMOS Circuits for Ultra Low-Power Applications," in *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, 2007, pp. 306–311.

[83] K. Akarvardar, D. Elata, R. Parsa, G. C. Wan, K. Yoo, J. Provine, P. Peumans, R. T. Howe, and H.-S. P. Wong, "Design Considerations for Complementary Nanoelectromechanical Logic Gates," in *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, 2007, pp. 299–302.

[84] R. Nathanael, V. Pott, H. Kam, J. Jeon, and T.-J. K. Liu, "4-terminal relay technology for complementary logic," in *Electron Devices Meeting (IEDM), 2009 IEEE International*, 2009, pp. 1–4.

[85] R. Gaddi, C. Schepens, C. Smith, C. Zambelli, A. Chimenton, and P. Olivo, "Reliability and performance characterization of a mems-based non-volatile switch," in *Reliability Physics Symposium (IRPS), 2011 IEEE International*, 2011, pp. 2G.2.1–2G.2.6.

[86] F. Chen, M. Spencer, R. Nathanael, C. Wang, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, D. Markovic, V. Stojanovic, and E. Alon, "Demonstration of integrated micro-electro-mechanical switch circuits for VLSI

applications," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 150–151.

[87] X. Wang, S. Narasimhan, A. Krishna, F. G. Wolff, S. Rajgopal, T.-H. Lee, M. Mehregany, and S. Bhunia, "High-temperature (#x003E;500 #x00B0;C) reconfigurable computing using silicon carbide NEMS switches," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, 2011, pp. 1–6.

[88] C. Dong, C. Chen, S. Mitra, and D. Chen, "Architecture and Performance Evaluation of 3D CMOS-NEM FPGA," in *Proceedings of the System Level Interconnect Prediction Workshop*, 2011, pp. 2:1–2:8.

[89] C. Chen, W. S. Lee, R. Parsa, S. Chong, J. Provine, J. Watt, R. T. Howe, H.-S. P. Wong, and S. Mitra, "Nano-Electro-Mechanical relays for FPGA routing: Experimental demonstration and a design technique," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, 2012, pp. 1361–1366.

[90] M. Spencer, F. Chen, C. C. Wang, R. Nathanael, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, and others, "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI applications," *Solid-State Circuits, IEEE J.*, vol. 46, no. 1, pp. 308–320, 2011.

[91] F. Chen, H. Kam, D. Markovic, T.-J. K. Liu, V. Stojanovic, and E. Alon, "Integrated circuit design with NEM relays," in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, 2008, pp. 750–757.

[92] J.-W. Han, J.-H. Ahn, M.-W. Kim, J.-B. Yoon, and Y.-K. Choi, "Monolithic integration of NEMS-CMOS with a Fin Flip-flop Actuated Channel Transistor (FinFACT)," in *Electron Devices Meeting (IEDM), 2009 IEEE International*, 2009, pp. 1–4.

[93] S. Chong, K. Akarvardar, R. Parsa, J.-B. Yoon, R. T. Howe, S. Mitra, and H.-S. P. Wong, "Nanoelectromechanical (NEM) relays integrated with CMOS SRAM for improved stability and low leakage," in *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, 2009, pp. 478–484.

[94] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: a tutorial and survey," *Solid-State Circuits, IEEE J.*, vol. 41, no. 3, pp. 712–727, 2006.

[95] K. Eshraghian, K.-R. Cho, O. Kavehei, S.-K. Kang, D. Abbott, and S.-M. S. Kang, "Memristor MOS Content Addressable Memory (MCAM): Hybrid Architecture for Future High Performance Search Engines," *Very Large Scale Integr. Syst. IEEE Trans.*, vol. 19, no. 8, pp. 1407–1417, 2011.

[96] S. Matsunaga and T. Hanyu, "Quaternary 1T-2MTJ Cell Circuit for a High-Density and a High-Throughput Nonvolatile Bit-Serial CAM," in *Multiple-Valued Logic (ISMVL), 2012 42nd IEEE International Symposium on*, 2012, pp. 98–103.

[97] R. Nebashi, N. Sakimura, Y. Tsuji, S. Fukami, H. Honjo, S. Saito, S. Miura, N. Ishiwata, K. Kinoshita, T. Hanyu, T. Endoh, N. Kasai, H. Ohno, and T. Sugibayashi, "A content addressable memory using magnetic domain wall motion cells," in *VLSI Circuits (VLSIC), 2011 Symposium on*, 2011, pp. 300–301.

[98] B. Rajendran, R. W. Cheek, L. A. Lastras, M. M. Franceschini, M. J. Breitwisch, A. G. Schrott, J. Li, R. K. Montoye, L. Chang, and C. Lam, "Demonstration of CAM and TCAM Using Phase Change Devices," in *Memory Workshop (IMW), 2011 3rd IEEE International*, 2011, pp. 1–4.

[99] W. Wang, "Magnetic Content Addressable Memory Design for Wide Array Structure," *Magn. IEEE Trans.*, vol. 47, no. 10, pp. 3864–3867, Oct. 2011.

[100] A. Seyedi, V. Karakostas, S. Cosemans, A. Cristal, M. Nemirovsky, and O. Unsal, "NEMsCAM: A novel CAM cell based on nano-electro-mechanical switch and CMOS for energy efficient TLBs," in *Nanoscale Architectures (NANOARCH), 2015 IEEE/ACM International Symposium on*, 2015, pp. 51–56.

[101] S. A., "Race to Exascale: Opportunities and Challenges," in *MICRO Keynote*, 2011.

[102] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting Phase Change Memory As a Scalable Dram Alternative," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, 2009, pp. 2–13.

[103] C. Lam, "Cell Design Considerations for Phase Change Memory as a Universal Memory," in *VLSI Technology, Systems and Applications, 2008. VLSI-TSA 2008. International Symposium on*, 2008, pp. 132–133.

[104] J.-G. Zhu, "Magnetoresistive Random Access Memory: The Path to Competitiveness and Scalability," *Proc. IEEE*, vol. 96, no. 11, pp. 1786–1798,

Nov. 2008.

[105] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory." 2008.

[106] "International technology roadmap for semiconductors: Process integration, devices, and structures, 2010." .

[107] A. M. Ionescu, V. Pott, R. Fritschi, K. Banerjee, M. J. Declercq, P. Renaud, C. Hibert, P. Fluckiger, and G. A. Racine, "Modeling and design of a low-voltage SOI suspended-gate MOSFET (SG-MOSFET) with a metal-over-gate architecture," in *Quality Electronic Design, 2002. Proceedings. International Symposium on*, 2002, pp. 496–501.

[108] J. M. Kinaret, T. Nord, and S. Viefers, "A carbon-nanotube-based nanorelay," *Appl. Phys. Lett.*, vol. 82, no. 8, 2003.

[109] R. Parsa, W. S. Lee, M. Shavezipur, J. Provine, R. Maboudian, S. Mitra, H. P. Wong, and R. T. Howe, "Laterally Actuated Platinum-Coated Polysilicon NEM Relays," *Microelectromechanical Syst. J.*, vol. 22, no. 3, pp. 768–778, 2013.

[110] R. Vaddi, V. Pott, G. L. Chua, J. T. M. Lin, and T. T. Kim, "Design and Scalability of a Memory Array Utilizing Anchor-Free Nanoelectromechanical Nonvolatile Memory Device," *Electron Device Lett. IEEE*, vol. 33, no. 9, pp. 1315–1317, Sep. 2012.

[111] S. Cosemans, "Data storage cell and memory arrangement." Google Patents, 2015.

[112] K. Akarvardar and H.-S. P. Wong, "Ultralow Voltage Crossbar Nonvolatile Memory Based on Energy-Reversible NEM Switches," *Electron Device Lett. IEEE*, vol. 30, no. 6, pp. 626–628, 2009.

[113] J. K. Gopal, A. T. Do, P. Singh, G. L. Chua, and T. T.-H. Kim, "A Cantilever-Based NEM Nonvolatile Memory Utilizing Electrostatic Actuation and Vibrational Deactuation for High-Temperature Operation," *Electron Devices, IEEE Trans.*, vol. 61, no. 6, pp. 2177–2185, 2014.

[114] R. P. Van Kampen, “Four-terminal multiple-time programmable memory bitcell and array architecture.” Google Patents, 2009.

[115] M.-S. Kim, W. W. Jang, J.-M. Lee, S. Kim, E.-J. Yun, K.-H. Cho, S.-Y. Lee, I.-H. Choi, J.-B. Y. Yong, D.-W. Kim, and D. Park, “NEMS switch with 30 nm thick beam and 20 nm high air gap for high density non-volatile memory applications,” in *Semiconductor Device Research Symposium, 2007 International*, 2007, pp. 1–2.

[116] W. Young Choi, H. Kam, D. Lee, J. Lai, and T.-J. K. Liu, “Compact Nano-Electro-Mechanical Non-Volatile Memory (NEMory) for 3D Integration,” in *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, 2007, pp. 603–606.

[117] K. Pagiamtzis and A. Sheikholeslami, “A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme,” *Solid-State Circuits, IEEE J.*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.

[118] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, “200MHz/200MSPS 3.2W at 1.5V Vdd, 9.4Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme,” in *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, 2003, pp. 387–390.

[119] K. Pagiamtzis and A. Sheikholeslami, “Using cache to reduce power in content-addressable memories (CAMs),” in *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, 2005, pp. 369–372.

[120] H. Miyatake, M. Tanaka, and Y. Mori, “A design for high-speed low-power CMOS fully parallel content-addressable memory macros,” *Solid-State Circuits, IEEE J.*, vol. 36, no. 6, pp. 956–968, Jun. 2001.

[121] I. Arsovski, T. Chandler, and A. Sheikholeslami, “A Ternary Content-Addressable Memory (TCAM) Based on 4T Static Storage and Including a Current-Race Sensing Scheme.”

[122] B. Jacob and T. Mudge, “Virtual Memory: Issues of Implementation,” *Comput. Pract.*, no. June, pp. 33–43, 1998.

[123] C. Chen, R. Parsa, N. Patil, S. Chong, K. Akarvardar, J. Provine, D. Lewis, J. Watt, R. T. Howe, H.-S. P. Wong, and S. Mitra, “Efficient FPGAs Using

Nanoelectromechanical Relays," in *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2010, pp. 273–282.

[124] "Advance Micro Devices. Software Optimization Guide for AMD Family 15h Processors. Number 47414.," 2014.

[125] "Intel Strongarm Processor. www.intel.com/design/pca/ applicationsprocessors/1110 brf.htm."

[126] "Sh-3 RISC Processor family. http://www.hitachi-eu.com/hel/ecg/products/micro/32bit/sh 3.html."

[127] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the Level of Abstraction for Scalable and Accurate Parallel Multi-core Simulation," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, pp. 52:1–52:12.

[128] "Intel Corporation, IntelR 64 and IA-32 Architectures Optimization Reference Manual. Number 248966-026."

[129] X. Liu, Y. Li, Y. Zhang, A. K. Jones, and Y. Chen, "STD-TLB: A STT-RAM-based dynamically-configurable translation lookaside buffer for GPU architectures," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, 2014, pp. 355–360.