Moatez Jrad

Nanotech Master
2020

CEA LIST
17 Avenue des Martyrs, 38000 Grenoble, France

Design and optimization of Computational SRAM architectures dedicated to highly data-centric applications

from 17/02/20 to 31/07/20

Confidentiality: yes

Under the supervision of:

- **Company supervisor: Jean-Philippe NOEL, jean-philippe.noel@cea.fr**

  Present at the defense: yes

- **Academic Tutor: Lorena ANGHEL, lorena.anghel@grenoble-inp.fr**

Abstract

This report details the work that I have done, during my graduation internship at CEA Grenoble as a member of the memory design team of the LIST laboratory, on designing and optimizing Computational SRAM architectures dedicated to data-centric applications. It is divided into four main sections: (1) A general introduction presenting my personal motivations for this work as well as the host organization (2) The scientific context leading to this project (3) The theoretical background necessary to understand my work (4) Presentation of the initial C-SRAM design that I have dealt with as well as my personal contribution in order to optimize it.

Résumé

Ce rapport détaille le travail que j'ai réalisé, lors de mon stage de fin d'études au CEA Grenoble en tant que membre de l'équipe de conception mémoire du laboratoire LIST, sur la conception et l'optimisation d'architectures de calcul SRAM dédiées aux applications centrées sur la data. Il est divisé en quatre sections principales: (1) Une introduction générale décrivant mes motivations personnelles pour ce travail ainsi que l'organisme d'accueil (2) Le contexte scientifique menant à ce projet (3) Les concepts théoriques nécessaires pour comprendre mon travail (4) Présentation du design initial de la C-SRAM que j'ai traité ainsi que ma contribution personnelle afin de l'optimiser.

Acknowledgments

I dedicate this work to my parents, my brothers, my lovely girlfriend and my small family for their continuous and unconditional support. I could not be what I am today without you. Thank you.

First, I would like to thank Didier Lattard, the head of laboratory, for allowing me to seize this great opportunity and to be part of the amazing memory design team at LFIM.

A special thank goes to Jean-Philippe Noel, my tutor, for supervising me and guiding me through this internship. I also thank Valentin Egloff and Roman Gauchi for their technical support since the beginning of this work and all my other colleagues for their warm welcome and making me feel comfortable among the team.

Lastly, I offer my gratitude to Liliana Buda-Prejbeanu, co-coordinator of Nanotech Master, for letting me work on this project, Lorena Anghel, my academic supervisor, who accompanied me during this internship and to any person who helped me during my studies.

Table of Contents

Glossary

LFIM: Laboratoire des Fonctions Innovantes et Mixtes
EPFL: École Polytechnique Fédérale Lausanne
IC: Integrated Circuit
CEA: Commissariat à l'énergie atomique et aux énergies alternatives
DSCIN: Département des Systèmes et Circuits Intégrés numériques
C-SRAM: Computational SRAM
IMC: In-Memory Computing
CPU: Central Processing Unit
NMC: Near-Memory Computing
DRAM: Dynamic Random Access Memory
SRAM: Static Random Access Memory
RTL: Register Transfer Level
GF: Global-Foundries
PIM: Processing-In-Memory
CMOS: Complementary Metal Oxide Semiconductor
LIM: Logic-In-Memory
CIM: Computing-In-Memory
ALU: Arithmetic & Logic Unit
FSM: Finite State Machine
WL: Word Line
BL: Bit Line
IMPACT: In-Memory Power Aware CompuTing
ISA: Instruction Set Architecture
TCL: Tool Command Language
FSM: Finite State Machine
FDSOI: Fully Depleted Silicon On Insulator
SoC: System on Chip
WNS: Worst Negative Slack
DRC: Design Rule Checking
ECO: Engineering change order
VCD: Value Change Dump
FSDB: Fast Signal DataBase
IP: Intellectual Property

List of Figures

List of tables

General introduction

This Master Thesis at CEA Grenoble is a perfect conclusion to my 3-year engineering studies at Grenoble-INP Phelma including my international Nanotech Master of Science degree that I carried out during the last two years in three high level institutions: Politecnico di Torino, Grenoble-INP Phelma and EPFL. It has been a very formative and prosperous period that has allowed me to consolidate my multidisciplinary engineering skills and the scientific knowledge I accumulated during my 3-years of studies, mostly in applied physics and electronics in the benefit of micro and nano systems.

a- Personal motivations

During my last academic semester at EPFL, I developed an interest for the design of integrated systems with its both categories: digital and analog design. The scope of having millions of transistors included in a single complex chip providing high technical performances, intrigued me always. Even though I had been learning a variety of courses in different fields during the previous years, I was more attracted to those related to designing modern Integrated Circuits (ICs). It was mainly due to how much reliable, compact, fast, efficient and optimized these circuits are. Since ICs are included in all electronic devices that surround us, I truly believe that, by working in this domain, I can contribute to shaping the human's future; make it better and brighter.

On the other hand, I have always kept in mind that, in such a connected world, the exponential growth in the volume of information regarding people and objects is an urgent matter that must be processed quickly. It has sparked an explosion in the volumes of digital data, causing a huge power consumption necessary to manipulate this flow of data. Global data centers around the world used roughly 416 terawatts last year, nearly 40% more than the entire United Kingdom and this power consumption will double every four years [1]. These numbers were shocking to me but so relevant at the same time. I have been wondering since I knew this if the new emerging computing architectures, more energy-efficient with much faster processing are not THE big deal today?

These motives had clearly a big influence on my graduation internship research and I feel more than privileged to have found such an interesting and innovative project at CEA Grenoble, that not only meets my professional expectations but represents also a perfect step and an amazing learning opportunity before starting officially my engineering career.

b- Presentation of the host organization

CEA (Commissariat à l'énergie atomique et aux énergies alternatives), short for the Alternative Energies and Atomic Energy Commission, is a public, scientific and technological research organization funded mostly by the French government. Known as a major player in research, development and innovation, CEA operates in four different areas: (1) defense and security, (2) nuclear and renewable energies, (3) technological research for industrial applications and (4) fundamental research. It contributes also to diverse collaboration projects with its academic and industrial partners.

Created by General De Gaulle in 1945, headed currently by François Jacq as general administrator and advised by Patrick Landais the high commissioner, the CEA have 9 centers all over France, as shown in Figure 1. With a yearly budget of 4.7 billion euros and a permanent staff counting more than 16 000 people, CEA has irrigated several fields of research directly or indirectly linked to the atomic sciences, transferring its technological innovations, scientific knowledge and expertise to the industrial world.

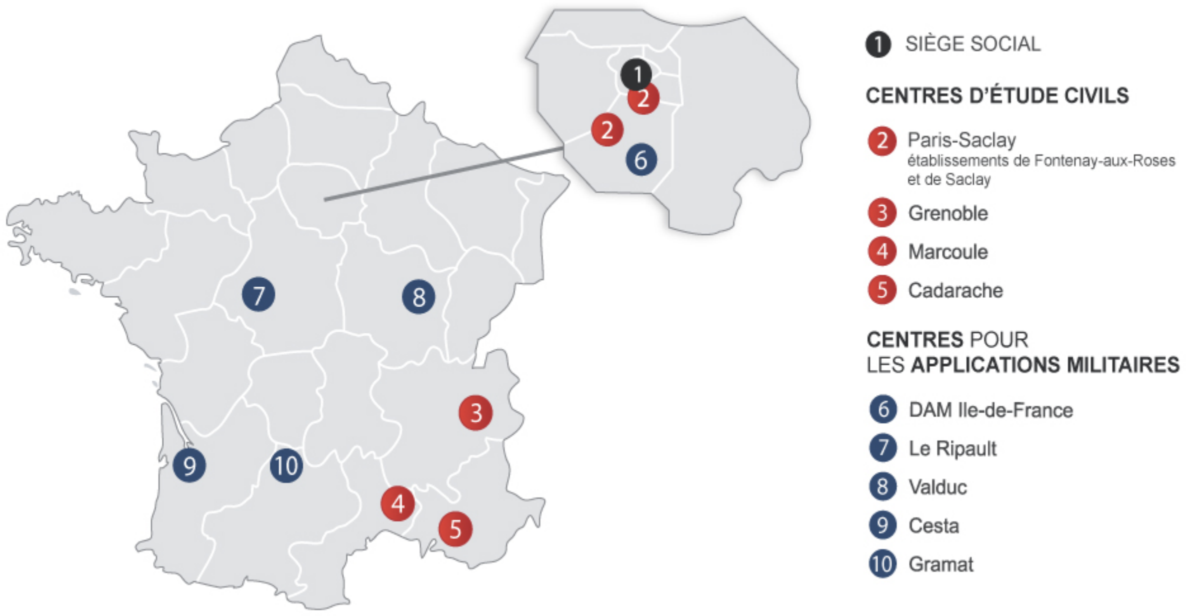


Figure 1: CEA facilities in France

CEA is based on 4 major divisions:

- The division of nuclear energy (DEN).
- The division of technological research (DRT), also called CEA Tech.
- The division of fundamental research (DRF).
- The division of military applications (DAM).

The CEA Tech division is distributed between two sites of CEA: Saclay (Paris) and Grenoble. It can be further divided into three institutes: LETI, LIST and LITEN. The CEA-List institute, in Grenoble, is where I carried out this 6-month internship.

In the heart of a very rich scientific, industrial and academic environment, the CEA facility in Grenoble devotes most of its research activities to the development of new technologies in the fields of energy, health, information and communication. From material sciences, through biotechnologies to nanotechnologies, it is the forefront of the technological research in the region.

Moreover, the CEA-LIST works mainly on intelligent digital systems and software-intensive technology, specialized in embedded systems, sensors-big data and advanced manufacturing. Counting around 200 industrial collaborations per year, its main mission is to carry out excellent technological developments on behalf of its strategic partners.

During this internship, I was a member of a 20-person memory design team of the LFIM laboratory belonging to the DSCIN department.

c- Planning

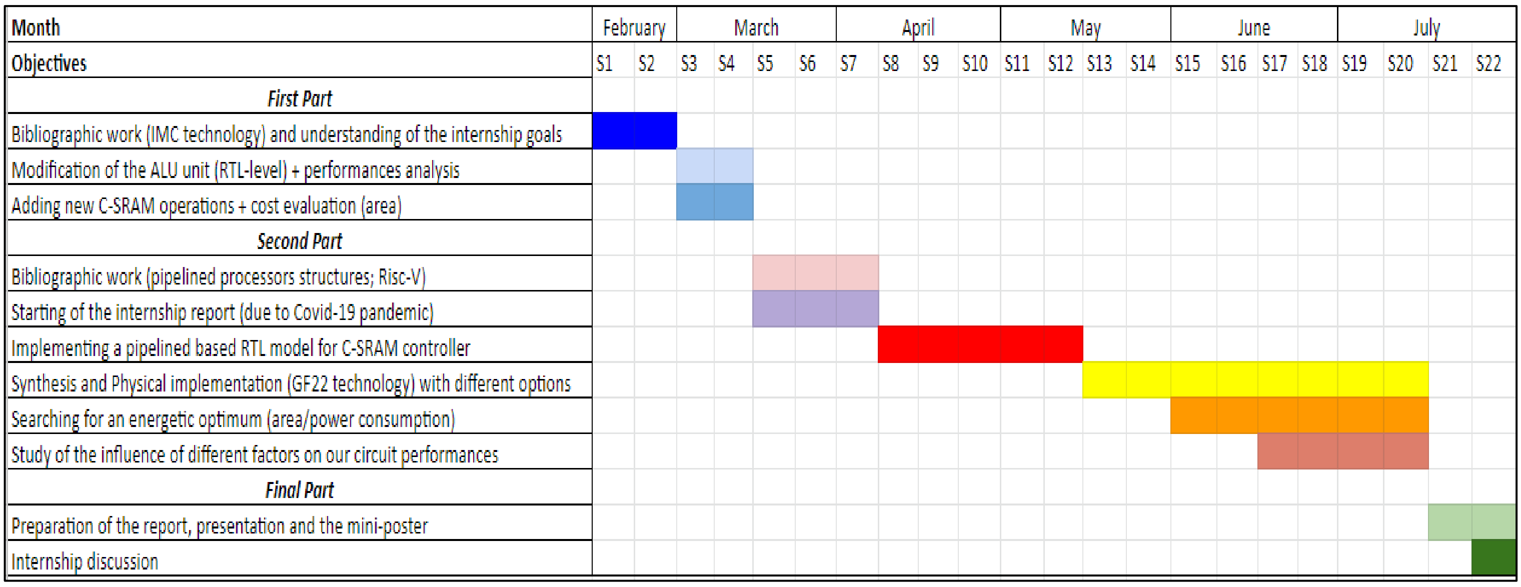
This Master Thesis took place from 17/02/2020 to 31/07/2020.

It can be divided in the following major steps:

- Bibliographic work and understanding of the internship goals
- Getting started with the existing design flow
- Definition of new C-SRAM functionalities
- Description of a pipelined-RTL model for the C-SRAM controller
- Synthesis and physical implementation of C-SRAM with different options on an advanced process technology node (22nm FD-SOI of Globalfoundries).
- Searching for an performance optimum points (area/energy consumption)

Once these tasks are achieved, the same work can be adapted later to another process technology node (analog-oriented 130nm CMOS process HCMOS9A of STMicroelectronics) for the sake of performance comparison.

Figure 2 presents a Gantt diagram that summarizes the most important milestones of this internship.



Month	February		March				April				May				June				July			
Objectives	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22
First Part																						
Bibliographic work (IMC technology) and understanding of the internship goals	🟦	🟦																				
Modification of the ALU unit (RTL-level) + performances analysis			🟦	🟦																		
Adding new C-SRAM operations + cost evaluation (area)			🟦	🟦																		
Second Part																						
Bibliographic work (pipelined processors structures; Risc-V)					🟥	🟥	🟥															
Starting of the internship report (due to Covid-19 pandemic)					🟪	🟪	🟪															
Implementing a pipelined based RTL model for C-SRAM controller								🟥	🟥	🟥	🟥	🟥										
Synthesis and Physical implementation (GF22 technology) with different options													🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨		
Searching for an energetic optimum (area/power consumption)															🟧	🟧	🟧	🟧	🟧	🟧		
Study of the influence of different factors on our circuit performances																	🟥	🟥	🟥	🟥		
Final Part																						
Preparation of the report, presentation and the mini-poster																					🟩	🟩
Internship discussion																						🟩

Figure 2: Gantt diagram

I- Scientific context

a- Need for both energy-efficient and low-latency computing architectures

The last decade has revealed an unmatched growth in the amount of data being stored, moved, processed and then displayed. Not to forget that, from the end-user devices such as computers, tablets or smartphones to the gigantic data centers, all these connected structures are powered by electricity. In fact, the Information-Communication Technologies (ICT) ecosystem consumes an estimated 10% of the world electricity and accounts for nearly 2% of the global carbon emission [2].
It is clearly becoming imperative to trust furthermore the emerging energy-efficient computing techniques. This may lead to a much-needed sustainable and reasonable energy consumption, reducing the environmental issues of new technologies in an energy-constrained world. Hence, power consumption criteria has become one of the important design constraints in all computing systems.

As described also by a program called "Energy-Efficient Computing: from Devices to Architectures" launched by the National Science Foundation (NSF), an inevitable consensus across the major industries involved in computing infrastructures, is that the future performance optimizations are strictly limited by the amount of energy consumed to critically manipulate data. We are reaching a proven performance-energy crisis with several trade-offs depending on the range of applications considered and the technological approaches admitted to address this challenge.
Several proposals of new device concepts, computing circuits and architectures has been rising lately all over the world, extending the engineering limits of energy-efficient computation. Thus, different circuit architectures are being reconsidered including digital circuits but also analog, memory and more sophisticated devices lending themselves to unconventional processing architectures.

In compliance with the global trending explained previously, my internship comes as a contribution to the work devoted, by CEA-LIST, to novel computing architectures involving memory technologies with a particular focus on optimizing functional and energy consumption aspects. However, it is known that these technologies do not scale as fast as the computing performances which leads to a famous problem known as the memory wall. Hence, bringing the computing part as close to memory as possible is the solution proposed to overcome this challenge.

b- Memory wall

"Memory Wall" is a famous concept originally theorized in 1994 by Wulf and McKee [3], mentioning that processing units, such as CPUs, are advancing so fast that will leave the DRAM-based main memory stagnant. It describes the implications of a processor/memory performance gap growing steadily over the last decades, as shown in Figure 3. Besides, the processing units will be forced to stall waiting on memory if the memory latency and bandwidth are insufficient to provide the processor with its set of instructions and data in time to proceed the computations [3]. Thus, placing more cores in a chip will only aggravate the situation since it uses a narrower channel to memory resources. The performance limitations of modern architectures, ranging from embedded systems to supercomputers, come mostly from memory accesses, power and delay [4].

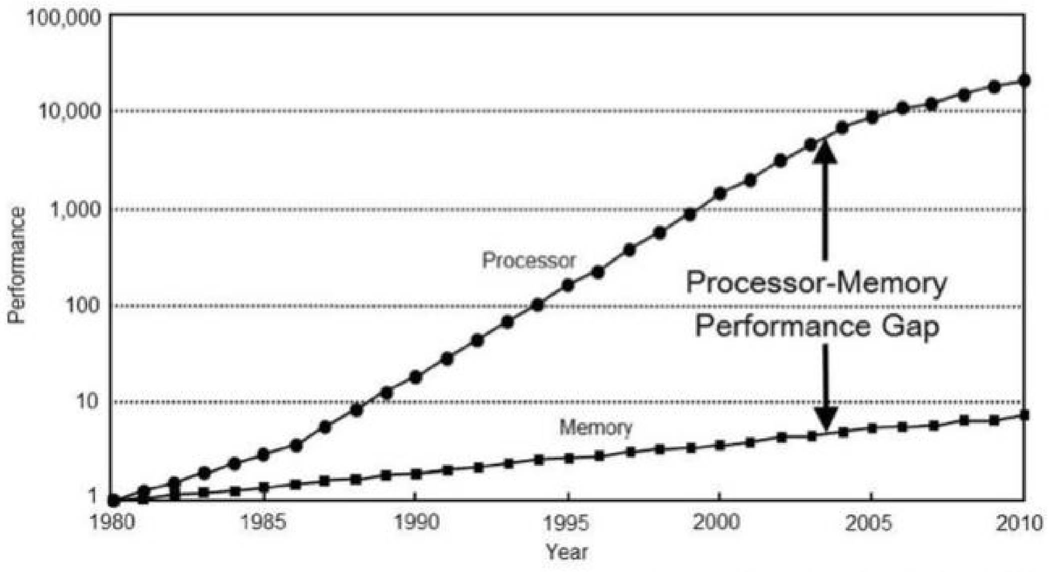


Figure 3: Processor-Memory performance gap (Memory bottleneck) [5]

The fast growing gap over the years between memory and computing technologies has led to several emerging researches on joining memory circuits and alternative computing architectures trying to break this wall [3]. Thus, the Von Neumann architecture commonly used for computing devices, which is a concept consisting in having the instructions and the data in the same memory, is getting structurally changed.

Therefore, the concept of processing in memory (PIM) has then reignited interest among industry and academic communities, largely driven by the recent advances in related technology (die stacking, emerging nonvolatile memory..) [6]. It also presents a solution to overcome the 'Memory wall' by pushing the computing units close to the memory which optimizes the data movement, in so called data-centric architectures [7]. Related works have shown that specific techniques such as In-Memory Computing (IMC) and Near-Memory Computing (NMC) [8] contributes not only to increasing memory throughput and applications performances, but also to reducing energy consumption through computing in the memory boundary. These techniques will be detailed later in this report.

c- Internship objectives

My main role during this internship was to participate in an undergoing project at CEA-LIST regarding an innovative design of a C-SRAM architecture, its structure will be explained in the upcoming sections, performing specific operations directly within the memory boundary. My primary mission consisted in optimizing its different design steps in order to reduce the energy consumption.

The first task to begin with was to familiarize with the C-SRAM controller model already developed by the lab as well as the entire design flow, then evaluate its performances and lastly fix the specific aspects to optimize. New C-SRAM functionalities had to be added after that through its RTL model. Furthermore, a pipeline-based controller model was implemented and evaluated.

Once the functional model was fixed, the following step was to synthesize it, place and route it, using EDA tools, with different options in a GF 22-nm technology in order to obtain a performance optimum point between area and energy consumption. We tried to study the influence of different design factors on the performances of our circuit. Then, we adapted the outcome of this study to a final C-SRAM design with multiple memory instances.

II- Background

a- In-Memory Computing (IMC)

Being outpaced by the spectacular growth of data-intensive applications we are witnessing today, the limitations of computing systems are reaching a certain point where a new paradigm is much needed [6]. As the traditional benefits for expanding the processing capability of computing devices through technology scaling have diminished with the end of Dennard scaling [6], new techniques have been explored based on Processing in-Memory (PIM), Logic in-Memory (LIM) and IMC architectures with a common aim of reducing memory latency and increasing the data transfer bandwidth while improving the energy efficiency [9].

In fact, a wide variety of applications are being researched for in-memory computing, as shown in Figure 4. They are classified in three main categories with respect to the degree of computational precision required, as function of data access and computational complexity [10].

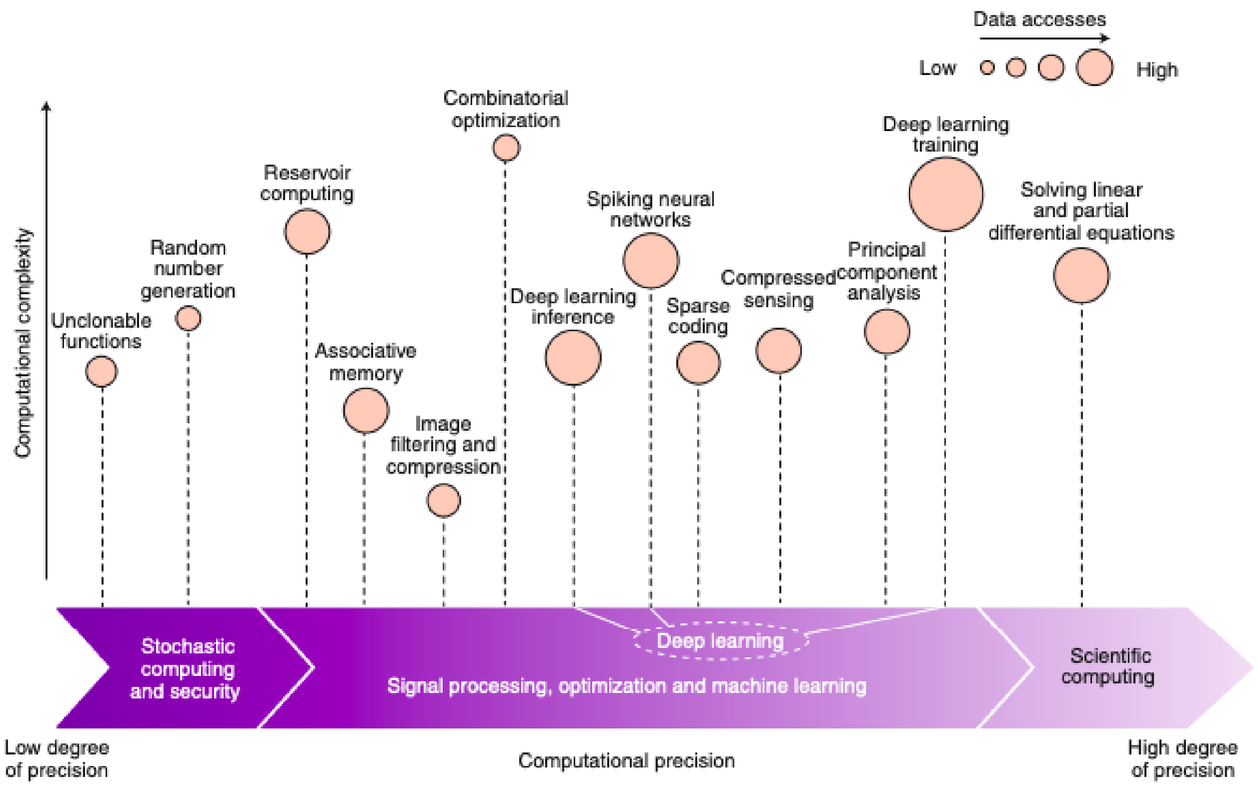


Figure 4: applications landscape for in-memory computing [10]

IMC technique can be adopted to reduce the computational issues of an application as well as the amount of data being accessed by the processing units, through performing computations within the memory arrays [10]. Data-centric applications and scientific computing are then the most rewarded from the IMC approach since it is extremely beneficial for high data access applications such as deep learning applications, neural networks and artificial intelligence (AI) related works.

Besides, there are several emerging computing techniques existing today that are basically differentiated in many ways. For example, PiM consists in bringing the computing unit next to the memory while keeping them dissociated [11], mostly implemented in memories fabricated with a DRAM process. While, LiM and IMC architectures are based on embedded memories fabricated with a CMOS process [9]. Figure 5 summarizes the main differences between these techniques.

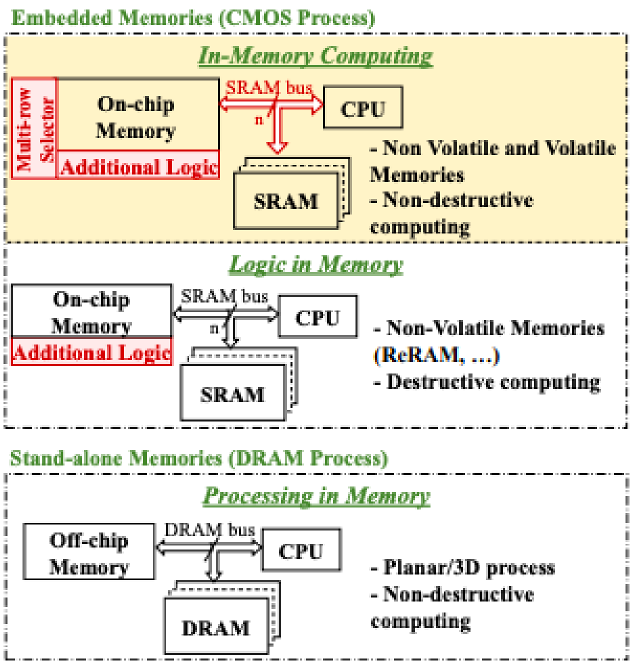


Figure 5: Differences between IMC, LIM and PIM architectures [9]

In this work, we are more interested in IMC-like architectures based on integrating computation units inside the memory boundary [9]. Unlike LIM for example, an important advantage for an IMC architecture is that it enables non-destructive computing in the memory [9], in a way that prevents the operand data loss after computation. However, a communication protocol has to be defined between the memory and the processor in order to perform Computing In Memory (CIM), as well as a specific set of instructions which makes it more complicated to have a complete IMC system. Other solutions like NMC technique can be preferred.

b- From In-Memory Computing (IMC) to Near-Memory Computing (NMC)

As explained previously, IMC technique has proven its benefits when it comes to increasing the performances of data-intensive applications by relaxing the memory throughput [12], through performing computations in the memory. Several works [13-14] show a change in the memory's structure in order to perform a pre-computing when the system accesses data. Computations are carried out on the periphery of the memory without going through the processor [12].

Hence, IMC is commonly used as a bitwise vector-based accelerator, as in compute caches, executing different arithmetic operations in the memory [13]. A part of these operations is pre-computed within the memory between two or more word-lines via the bit-lines. In addition, an Arithmetic Logic Unit (ALU) and a Finite State Machine (FSM) take care of the remaining boolean, logical and arithmetic operations at the bottom of the columns [12]. This working principle will be detailed later in this report.

The main goal, as in this work, is to maximize the number of possible computations at IMC level in order to minimize data movements, much-needed for a variety of applications such as: security and cryptographic applications, bioinformatic applications and binary neural network systems [12]. However, IMC struggles when the amount of memory needed is increased or when applications are scaled up. Adopting NMC technique, through integrating an accelerator as close as possible to memory [8], is a reasonable solution allowing a computing system to perform more operations within the memory periphery, to control data movement between multiple memory instances, as in a multi-IMC architecture, and to reduce its energy cost [12]. It can also be dedicated to aligning vectors while being processed.

To comply with these IMC prospects, several aspects of the memory device should be addressed such as: the physical structure, the array-level organization, the peripheral circuitry and the control logic in order to perform specific operations [10].

As illustrated in Figure 6, in a conventional computing system, data D has to be moved into a processing unit, leading to significant loss in latency and energy. While in the case of in-memory computing, an operation f(D) is performed within a computational memory through exploiting the physical attributes of the memory without moving data D to the processing unit [10].

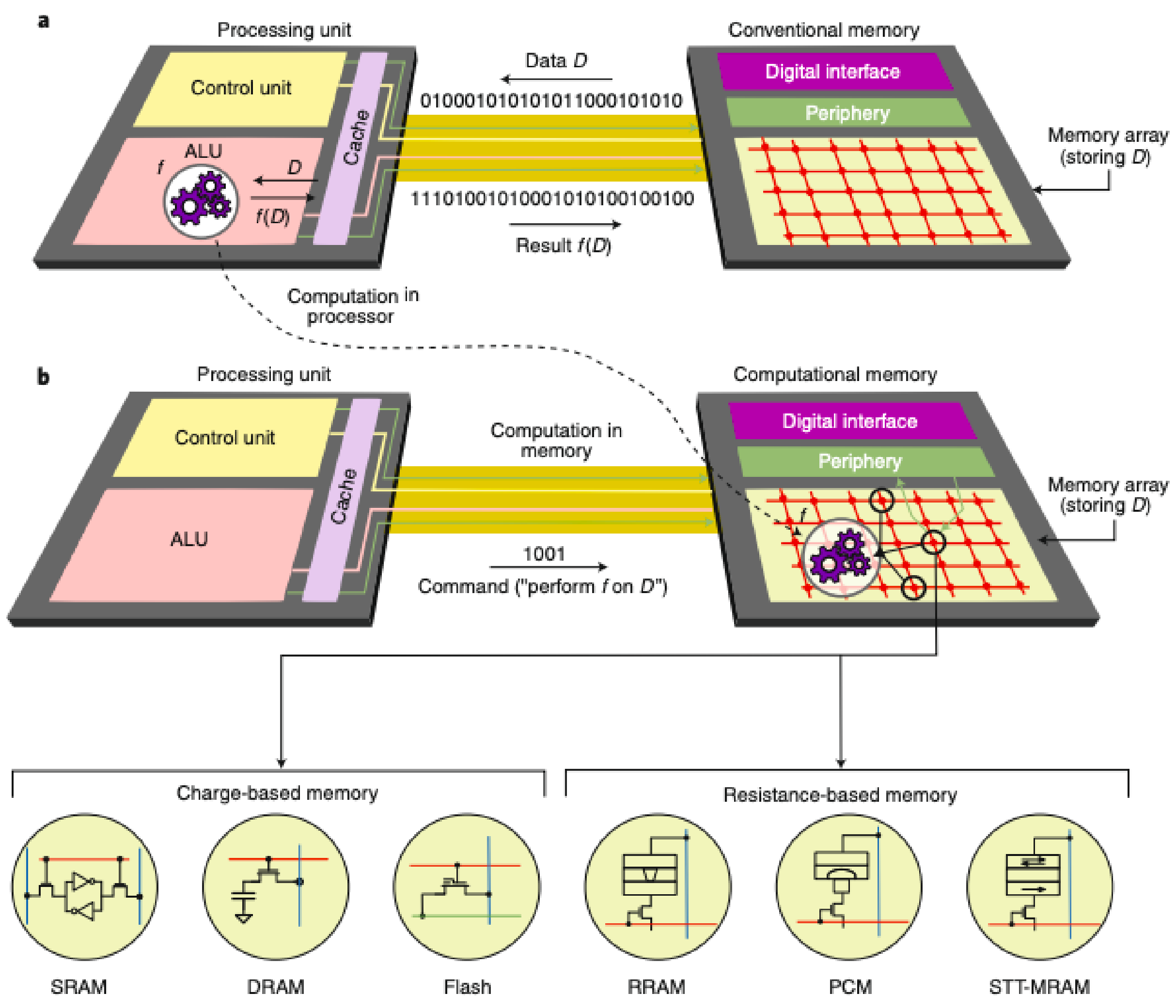


Figure 6: a- conventional data processing , b- computational memory processing [10]

The computational operations are performed within the boundary of the memory array but also at its peripheral circuitry without losing the content of each memory element. Both charge-based memory technologies and resistance-based memory technologies can serve as elements of a computational memory unit.

Considering that the memory hierarchy consists generally of several levels of cache, a main memory and a storage memory, the conventional method that precedes processing data is moving it from the storage memory to caches. Besides, NMC technique, which we will be adopting during this work, aims at processing data close to where it is stored. This principle can be applied to any level of memory subsystem by varying the related technology options [8]. Figure 7 shows how these computing techniques exploit the memory hierarchy.

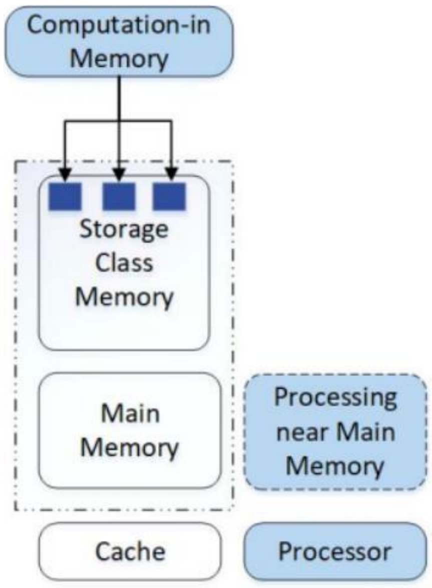


Figure 7: Processing techniques and memory hierarchy [8]

In the literature, different memory technologies have been investigated to perform computations within the memory, ranging from DRAM, SRAM to caches and emerging non-volatile memories. For our case of study, we focused our work on SRAM-based technology as it will be explained in the coming section.

c- SRAM-based computing architectures

Among the different memory technologies, this work, as many recent others, exploits SRAM-based architectures aiming for faster computing capabilities and less power consumption. Nevertheless, these architectures need a specific re-configurability for their physical structures to overcome the expensive cost challenge for similar chip products nowadays [14]. Thus, the redesign of SRAM macros, which has been neglected for a long time, is encouraged again to comply with the new desired computing standards.

The memory characteristics are then a key factor for in-memory computing. Among the primary ones, we can cite the access time, representing how fast data can be written and retrieved, and the cycling endurance, referring to the number of memory switches from one state to another [10]. Since we are interested in SRAM-based technology in this work, an SRAM bit-cell has a fast access time (sub-nanosecond) and a very high endurance [16] which makes it suitable for IMC technology. Its structure consists in a bi-stable transistor made of two CMOS inverters connected back to back with two additional field-effect transistors (FETs) serving as selectors, yielding a standard 6 transistor (6T) SRAM cell [10] as shown in Figure 8.

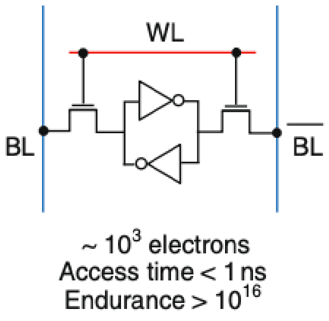


Figure 8: A 6T SRAM cell [10]

To be able to perform operations within the memory, several rows should be selected, in an SRAM bit-cell array in our case, at the same time during a single clock cycle [14]. Indeed, the computational memory structure is quite similar to a conventional memory, consisting in a two-dimensional array with horizontal wires, known as word lines (WLs) and vertical wires, referred to as bit lines (BLs). The main differences appear in the read/write circuitry, the data format and the control logic [10] allowing to select multiple WLs and sense BLs in parallel.

For bit-wise logical operations for example, a basic working principle is the following, see Figure 9: BLs are initially pre-charged to VDD. Then, both WLs are simultaneously activated and BLs are discharged at different rates depending on the data stored in the bit-cells. AND and NOR operations can be performed by sensing the BL or its complement voltages respectively.

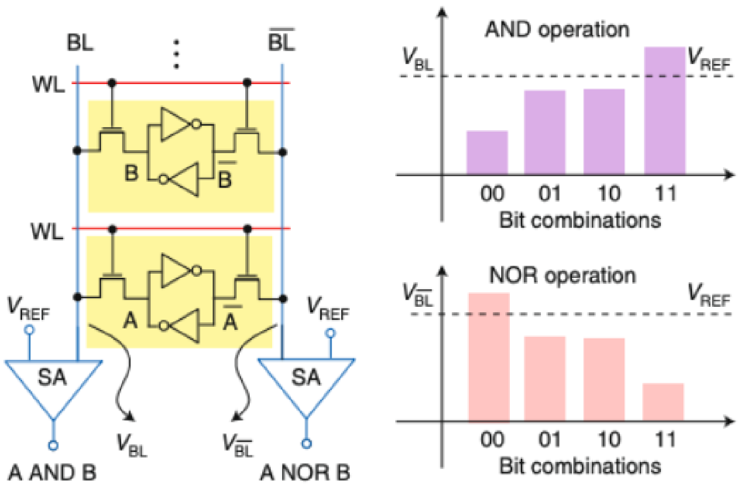


Figure 9: Bit-wise logical operations using an SRAM array [10]

This method can be also used efficiently to cascade this kind of operation. With additional cloning or duplication steps, it is possible to construct in-memory full adders and multipliers through the ability of processing bit lines in parallel [10]. Other operations can also be performed in-situ by changing the peripheral circuitry in a convenient way.

To sum up this section, even if there are several existing computing architectures nowadays that are introduced as solutions to overcome the memory bottleneck, explained previously, they still show serious vulnerabilities particularly related to their lack of industrial compatibility; they require a lot of structural modifications which is always unwelcome from this point of view. A need for an industrial-ready system with high computing capabilities is the main reason for our interest in a C-SRAM architecture during this work.

III- Computational SRAM macro

IV- Design and Optimization

Summary and outlook

It is fair to say that all the work we conducted on the initial design of C-SRAM led to several interesting results that could be hugely valuable for choosing the best design options and reach optimum energetic points as aimed in the beginning of this internship. Hence, we can summarize them in the upcoming points:

- ❖ Adding new functionalities to the C-SRAM instructions set (shifts and HSWAP).
- ❖ Implementing a 5-stage pipeline in the RTL model helped us reach better power, performance and area (PPA) tradeoffs. We were able to increase the operating clock frequency while maintaining a proper function of the system. In addition to that, we gained 31% in total chip area and 42% in power at a frequency of 500MHz and for a 1R1W memory.
- ❖ We completed a detailed power analysis of C-SRAM that gave us an idea on the participation of each power attribute in the total consumption. We did characterize also the energetic cost of different C-SRAM operations and even of every pipeline stage executed.
- ❖ We determined that it is better to use the largest memory size available instead of assembling smaller macros together to reach the same size capacity. However, this second option was really useful to reach memory capacities beyond what we had at our disposition.
- ❖ Increasing the clock frequency is rewarding when it comes to reducing the energy cost of C-SRAM operations. This could be extremely favorable for future integration of C-SRAM. Yet, it generated more timing violations and power consumed which should be also analyzed carefully.
- ❖ Memory shapes with higher MUX inputs consumed less power and energy.
- ❖ We managed to successfully implement a C-SRAM design with several memory instances assembled together to reach high size capacities. This represented a chip version that could be sufficient for targeted applications to perform their computations efficiently.

Thus, the study realized on the impact of various factors on C-SRAM performances revealed to be very helpful in the pursuit of optimum energetic points. Nevertheless, there were other possible optimizations that we thought about and which could be implemented later. This included:

- Using registers to store data instead of writing it in the memory when possible, especially when this data will be used for other operations. It will significantly reduce power consumption.
- Using multiple clock domains in our design, since the maximum clock frequency for memory is always higher than the digital wrapper's (DW) [for example: f(memory)= 2*f(DW) or even 3*f(DW)].
- Increasing even more the operating frequency (beyond 1GHz), determining the critical paths responsible for the timing violations and fixing them at the synthesis step.
- Reaching higher total memory size through increasing the number of words (memory addresses) instead of the memory line's width in a way that we will not have to deal with very long routing connections in the Place&Route step and so we can increase the density of our design and reduce the area and power consumed. However, this will affect the parallel computing ability of the system.

Conclusion

To sum up, I can say that this internship conducted at the LIST laboratory at CEA Grenoble, was overall a magnificent experience on a personal level as well as professionally. I am grateful for seizing this opportunity and being able to work on such an innovative topic that could start a revolution when it comes to low power computing architectures. Indeed, the enormous volume of digital data that we are manipulating nowadays is causing an unbelievable energy consumption and C-SRAM-like emerging architectures may represent an efficient solution for this problem in the near future.

On another hand, I should also mention that the studies that I have followed during Nanotech Master of Science's program helped me a lot through this internship and made me extremely comfortable while facing multiple technical problems. Not to forget that I have also acquired a significant experience dealing with multiple software and new programming languages throughout this adventure.

Despite some major disturbances due to the COVID-19 pandemic that emerged at the beginning of my internship, I fortunately managed to overcome efficiently the difficulties related to this issue with the help of my supervisor and through adapting quickly to the new work conditions such as working from home. Reaching almost all the main objectives set at the beginning was then a huge achievement for me that I am really proud of.

References

[1]- Radoslav Danilak, “Why Energy Is A Big And Rapidly Growing Problem For Data Centers”, Forbes, 2017.

[2]- Tulika Mitra, “Energy-Efficient Computing with Heterogeneous Multi-Cores”, in 2014 International Symposium on Integrated Circuits (ISIC), 2014.

[3]- Sally A. McKeeRobert and W. Wisniewski, “Memory Wall”, Encyclopedia of Parallel Computing. Springer, Boston, 1994.

[4]- M. Horowitz, “1.1 computing’s energy problem (and what we can do about it)”, in 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014, pp. 10–14.

[5]- J.L. Hennessy and D.A. Patterson, “Computer Architecture: a Quantitative Approach”, Morgan-Kaufman, 2011, pp. 856.

[6]- Soroosh Khoram, Yue Zha, Jialiang Zhang and Jing Li, “Challenges and Opportunities: From Near-memory Computing to In-memory Computing”, in 2017 ACM, 2017.

[7]- T. Agerwala and M. Perrone, “Data centric systems, the next paradigm in computing,” 2014.

[8]- G. Singh, L. Chelini, S. Corda, A. J. Awan, S. Stuijk, R. Jordans, H. Corporaal, and A. Boonstra, “A review of near-memory computing architectures: Opportunities and challenges”, in 2018 21st Euromicro Conference on Digital System Design (DSD), 2018, pp. 608–617.

[9]- M. Kooli, H. P. Charles, C. Touzet, B. Giraud and J. P. Noel, “Smart Instruction Codes For In-Memory Computing Architectures Compatible With Standard Sram Interfaces”, in 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018.

[10]- Manuel Le Gallo, Riduan Khaddam-Aljameh and Evangelos Eleftheriou, “Memory devices and applications for in-memory computing Abu Sebastian”, in 2020 Nature Nanotechnology, 2020.

[11]- M. Gokhale, B. Holmes, and K. Iobst, “Processing in memory: The Terasys massively parallel PIM array”, Computer Volume 28, 1995.

[12]- M. Kooli, P. Vivet, J.-P. Noel, E. Beigné, S. Mitra and H.-P. Charles, “Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture”, in 2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC), 2019.

[13]- S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, and R. Das, “Compute caches”, in 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017.

[14]- K. C. Akyel, H. P. Charles, J. Mottin, B. Giraud, G. Suraci, S. Thuries, and J. P. Noel, “DRC2: Dynamically reconfigurable computing circuit based on memory architecture”, in 2016 IEEE International Conference on Rebooting Computing (ICRC), 2016.

[15]- Richard S. Walton, Laura Peters, Ron Bowman, William J. McClean, Eric Bogatin, Brian Matas, James Griffin, Jim Griffin, Terri McGrath, Integrated Circuit Engineering Corporation, "SRAM technology", 1997, pp. 8-24.
[16]- Maha Kooli, Henri-Pierre Charles, Clément Touzet, Bastien Giraud and Jean-Philippe Noel, "Software Platform Dedicated for In-Memory Computing Circuit Evaluation", in 2017 International Symposium on Rapid System Prototyping (RSP), 2017.
[17]- J. P. Noel, V. Egloff, M. Kooli, R. Gauchi, J.-M. Portal2 , H.-P. Charles1 , P. Vivet and B. Giraud, "Computational SRAM Design Automation using Pushed-Rule Bitcells for Energy-Efficient Vector Processing", in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020.
[18]- Wikipedia, Logic synthesis.
[19]- Wikipedia, Reduced instruction set computer.
[20]- Wikipedia, Instruction pipelining.
[21]- Wikipedia, Physical design.
[22]- Sadiq M. Sait & Habib Youssef, "Introduction to VLSI Physical Design", 2003, pp 16-18.
[23]- GlobalFoundries Website, Technology and solutions, CMOS, 22FDX.