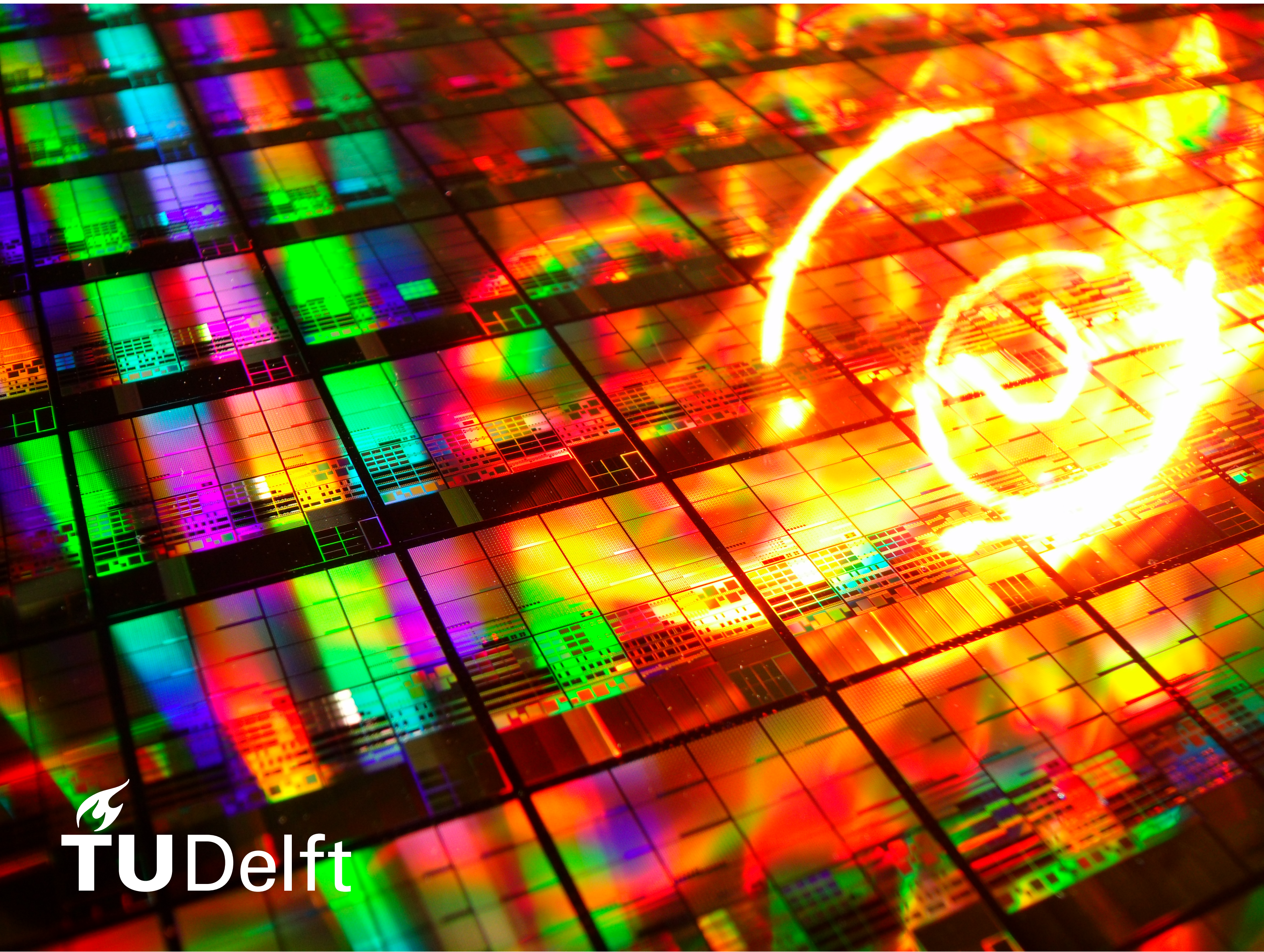
# Address decoders with FeFET-based Content-Addressable Memories

Master Thesis

Thomas I. Makryniotis



# Address decoders with FeFET-based Content-Addressable Memories

by

Thomas I. Makryniotis

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday, 27th of August 2024 at 14:00 PM.

| | | |
|---|---|---|
| Student number: | 4625501 | |
| Project duration: | July 2023 – August 2024 | |
| Thesis committee: | Prof. dr. ir. G. Gaydadjiev, | TU Delft, supervisor |
| | Prof. dr. ir. M. Taouil, | TU Delft , co-supervisor |
| | Prof. dr. ir. M. Babaie, | TU Delft |

Cover: 300 mm test wafer under magnification (2.5 mm side size of the chip) lightened with visible light. Creative Commons CC0 1.0
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

# Acknowledgements

It is been a long time since I started this journey; definitely not an easy one, but not a difficult one, either. It was a proper adventure, with all the necessary ingredients an adventure has: excitement, success, failure, commitment, trouble and eventually, relief. As with every journey, in the end, what worth remembering is the people that helped you go through it.

So, this is a public “thank you” to everyone who stood by me during this journey.

First of all, I would like to express my deepest gratitude to my supervisor Georgi and my co-supervisor Motta, for all the valuable help and guidance they provided, often regardless of the time or the day of the week. Without them it would not be possible to complete this project.

A big ”thank you” also goes to the members of the Q&CE group who generously shared their expertise, answered my questions and provided their ideas.

Then, I would like to thank all my friends, old and new, who were tolerating my tantrums and celebrating my successes. Every cup of coffee, every beer, every walk and every conversation we had, was one more little push towards the finish line.

Finally, I would like to thank my family; my parents Giannis and Georgia and my brother Konstantinos for all their unconditional love, support and motivation throughout the years.

*Thomas I. Makryniotis*
*Delft, August 2024*

# Summary

Emerging, non-volatile memories are promising solutions to contemporary computing problems. These include In-Memory Computing, Neuromorphic Computing, and Machine Learning. We believe that these are not the only possible applications of non-volatile emerging memory devices and that these can be used effectively for tackling several challenges of the ”conventional” computer architecture.

For instance, address decoders are an integral part of random access memories. They are typically implemented using fast logic optimised for low latency. They are, however, difficult to test, while their repair is considered to be impossible. In this thesis we research the possibility of a highly scalable and testable address decoder solution, based on Content-Addressable Memories build with ferroelectric transistors (FeFET).

This solution can offer numerous advantages including transistor count close to state of the art designs, while outperforming them in terms of latency. In addition, a key advantage could emerge during the testing of this decoder; due to its regular 2D structure, it’s testability is comparable to that of conventional memory arrays. Moreover, it can enable higher production yields, considering that adding a few spare rows will enable end-of-production repair, in the presence of manufacturing defects.

By additionally increasing the number of address bits stored in a single FeFET CAM cell, further potential area reductions of 30% - compared to the traditional dynamic NAND decoders - can be achieved.

# Contents

# List of Figures

# List of Tables

# Nomenclature

*If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.*

## Abbreviations

| Abbreviation | Definition |
|---|---|
| ISA | International Standard Atmosphere |
| ACAM | Analog Content-Adressable Memory |
| AI | Artificial Intelligence |
| BCAM | Binary Content-Addressable Memory |
| CAM | Content-Addressable Memory |
| CMOS | Complementary Metal-Oxide Semiconductor |
| CNTFET | Carbon NanoTube Field-Effect Transistor |
| DRAM | Dynamic Random Access Memory |
| FCAM | Ferroelecric Content-Addressable Memory |
| FPGA | Field-Programmable Gate Array |
| FRAM | Ferroelectric Random Access Memory |
| FeFET | Ferroelectric Field-Effect Transistor |
| HKMG | High-K Metal Gate |
| HVT | High $V_{TH}$ (Threshold Voltage) |
| HZO | Hafnium Zirconate |
| LUT | Look-Up Table |
| LVT | Low $V_{TH}$ (Threshold Voltage) |
| MCAM | Multi-level Content Addressable Memory |
| MFIS | Metal-Ferroelectric-Insulator-Semiconductor |
| MFMIS | Metal-Ferroelectric-Metal-Insulator-Semiconductor |
| MLSA | Match Line Sense Amplifier |
| MOSFET | Metal-Oxide Semiconductor Field-Effect Transistor |
| NVFF | Non-Volatile Flip-Flop |
| PCM | Phase-Change Memory |
| PZT | Lead Zirconate Titanate |
| PTM | Predictive Technology Model |
| RAM | Random Access Memory |
| ReRAM | Resistive Random Access Memory |
| SNN | Spiking Neural Network |
| SRAM | Static Random Access Memory |
| SSD | Solid-State Disk |
| ScL | Source Line |
| TCAM | Ternary Content-Addressable Memory |
| TLB | Translation Lookaside Buffer |

# 1

# Introduction

## 1.1. Motivation and Research Questions

Computer engineering is one of the youngest fields of engineering and arguably, the one with the greatest impact in the modern world. The advancements in the field are so rapid and groundbreaking that within 70 years, computers went on from being enormous, power-hungry, unreliable machines that only a few institutes could afford experimenting with, to tiny, elegant and powerful gadgets that play a vital role in our day-to-day lives.

Several times over the years, the computing paradigms have shifted: from a central processing unit responsible for everything, to separate accelerators that communicate over fast buses, to the most recent advancements of Near- and In-Memory Computing. Undoubtedly, these advancements would not be possible without the constant research and development in the microelectronics field. Vacuum tubes gave their place to several different types of transistors, like Metal-Oxide Semiconductor Field-Effect Transistors (MOSFET), which then became the building blocks for Integrated Circuits (IC). Complementary Metal-Oxide Semiconductor (CMOS) technology dominates the IC industry the last decades and it seems that it will keep on doing so for the foreseeable future.

Despite the success of CMOS, novel, emerging technologies are gaining traction, fueled by the increased demand for data processing, and the applications in the fields of Machine Learning (ML), Neural Networks (NN) and Artificial Intelligence (AI). These technologies include among others Memristors, Ferroelectric FET (FeFET), Phase-Change Memory (PCM), Spintronics and Carbon Nanotube FET (CNTFET). A big part of the recent research on these devices is devoted in their storage capabilities and their potential to be used in building memory arrays that could also perform logic operations. Despite the significant potential of these novel, non-von Neumann architectures, we believe that emerging devices can give solutions to many more problems in the field of Computer Engineering.

Address decoders, for instance, are an integral part of Random Access Memories (RAM) and low latency address decoding remains a challenge in high-performance memory designs ranging from cache memories up to large main memory arrays. This typically results in highly customised logic implementations for the targeted memory array that are fast, but difficult to test. Detecting address decoder faults is challenging since the tests have to ”recover” the effects from the expected values obtained from different locations in the memory array [1], [2]. Moreover, detecting linked address decoder faults requires long March tests [3]. Implementing testable and ideally, repairable address decoders will not only help reducing end of production test times but it will also improve the overall yield.

In this thesis we research the feasibility of a highly scalable and testable address decoder solution, based on Content-Addressable Memories (CAM) build with FeFET. During the course of our research we tried to answer the following two key questions:

- Is it feasible to leverage the FeFET device to create a CAM array, which then can be used to decode a memory address?
- If yes, how is it possible to test the functionality of this structure in a simple and effective way?

## 1.2. FeFET-based Address Decoders

*Hardwired dynamic NAND* address decoders are widely used in random access memories to decode parts of the address. The function of a dynamic NAND decoder is quite simple: each row of the decoder corresponds to a row of the decoder's truth table and a certain number of transistors are hardwired to either the address signals, or their inverted ones. Based on which transistors are ON or OFF, the appropriate line of the address decoder is activated.

Practically, the same can be achieved by using a content-addressable memory which has all the available addresses stored: an input address is used as a query and if a CAM row contains it, it signals a Match Line (ML) which then activates a row in the memory array. An implementation was suggested for the first time in [4] where the proposed CAM array was based on a ReRAM array. This design was discussed in a more elaborate way in [5] where a few applications were also proposed such as fully associative TLBs and virtually addressable memory.

FeFET-based address decoders constitute a special category of CAM-based decoders, where the CAM cell is built with FeFETs. Compared to cells built with other technologies, FCAM cells demonstrate lower search energy, significantly reduced area (especially for multi-level cells) and switching times similar to those of a ”traditional” MOSFET. In addition, FeFET manufacturing is fully compatible with the existing manufacturing processes, with only minor changes being deemed necessary. This leads to a much easier integration with ”conventional” CMOS circuits and a rapid design-prototype-manufacturing cycle, compared with other technologies like ReRAM or PCM.

## 1.3. Contribution

In this paper we propose a novel, low latency address decoder built using an array of two-FeFET Content Adressable Memory (CAM) cells. Each FeFET CAM cell stores at least two bits in order to reduce the overall array size. Our address decoder was simulated in PTM14nm CMOS technology and outperforms the state of the art in latency, by at least a factor of 1.7x. At the same time, its area stays on par with the baseline solutions. When more than two bits per CAM cell are used, additional area gains can be achieved. Moreover, due to its memory-like regular organization, our decoder is easy to test and by adding few spare rows end-of-production-line repair can be facilitated.

The main contributions of this paper are:

- a fast FeFET CAM based address decoder design;
- a careful investigation of the possible partial-open and bridge defects in the proposed two-transistor CAM cells;
- a set of march tests able to detect the above defects.

## 1.4. Thesis Organisation

The remainder of the dissertation is organised as follows:

*Chapter 2* provides the necessary background on FeFETs. In this chapter we discuss the physics, the materials and the applications of FeFETs. Key concepts are explained, as well as a general introduction to the possibilities of this emerging technology.

*Chapter 3* provides an overview of address decoders. Several different types are being discussed, including static, dynamic and column decoders. Also, key concepts are introduced, like decoder stages and predecoding.

*Chapter 4* is an extensive overview of Content-Addressable memories. Starting from established designs and technology (like CMOS), we discuss designs and applications, as well as past and present advancements, pitfalls and edge cases. The chapter also introduced the new generation of CAM cells built with emerging memory technologies, showcasing their advantages over previous designs.

*Chapter 5* introduces and discusses the FCAM row decoder. Combining the insight from the previous chapters, we present the idea of a memory address decoder based on CAMs built with FeFET devices.

*Chapter 6* presents the experimental platform and the simulation results. In this chapter we also discuss the results and the challenges of the project and we provide our solution for testing this new type of decoder based on simple march tests. We also compare certain metrics of the design with other state-of-the-art conventional decoder designs.

In the appendices, extra information can be found with regards to this research. *Appendix A* includes a conference paper to be published in VLSI-SoC 24' which constitutes a ”condensed” version of this thesis, while *Appendix B* includes additional material from the simulations, such as plots and figures of measurements on different defect sites.

# 2

# Introduction to FeFETs

## 2.1. Ferroelectricity and Ferroelectric Materials

*Ferroelectricity* is the property of certain materials (crystals) to demonstrate a spontaneous, yet reversible, electric polarization. We call these materials *ferroelectrics* and their outstanding feature is that, this polarization reversal can occur by applying a strong electric field in the opposite direction. Therefore, the polarization is dependent not only on the current electric field but also on its history, effectively yielding hysteresis loops (Figure 2.1). This phenomenon was discovered and reported for the first time in 1920 in Rochelle salt by J. Valasek [6]. The observed similarities with ferromagnetism lead to the use of the prefix ferro-, meaning iron, despite the fact that most ferroelectric materials do not contain iron.

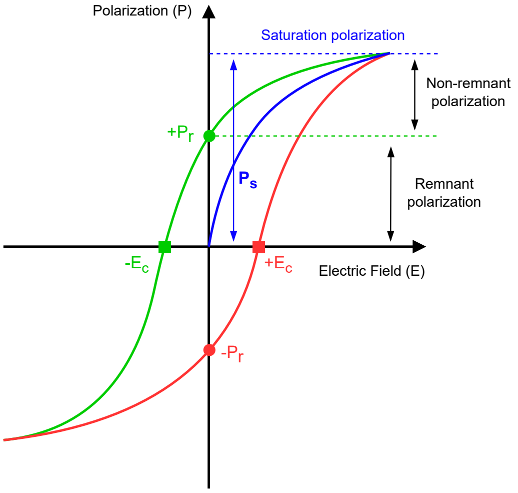
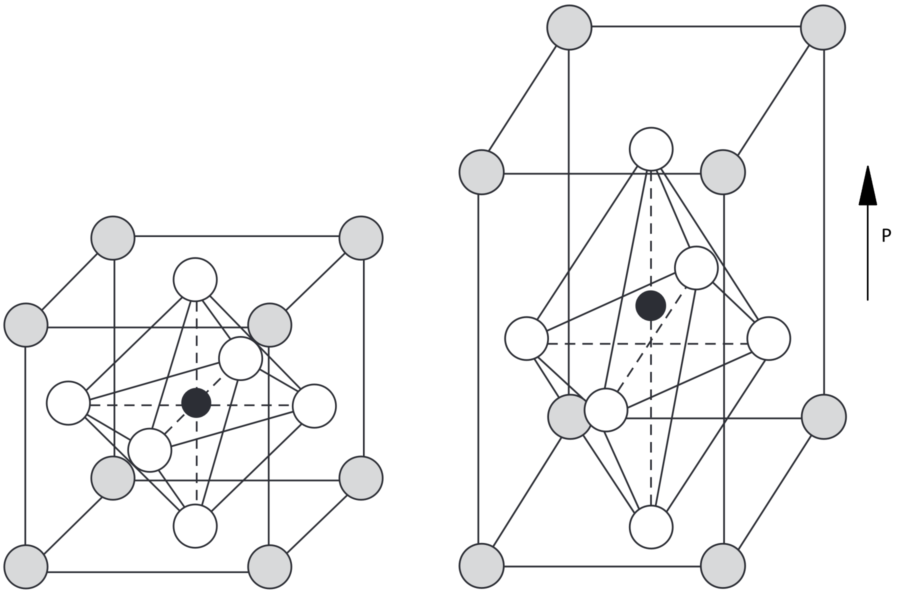


**Figure 2.1:** The ferroelectric hysteresis loop

In a way similar to *ferromagnetism*, this spontaneous polarization is the consequence of a specific arrangement within the crystal lattice of the material. A good example of a characteristic ferroelectric material is lead titanate ($PbTiO_3$); its ferroelectric properties arise from the displacement of titanium atom in the center of the $TiO_6$ octahedron. In general, the spontaneous polarization present in these materials stems from the breaking of symmetry in the crystal structure along a unique axis, specifically, a non-centrosymmetric lattice. Figure 2.2 shows the perovskite structure of the lead titanate, in two different phases the paraelectric and the ferroelectric. Similarly structured materials demonstrate similar ferroelectric properties like lead zirconate titanate and barium titanate. Despite all of them sharing the same pervskite structure, it is important to note that most perovskites do not show any ferroelectric properties.

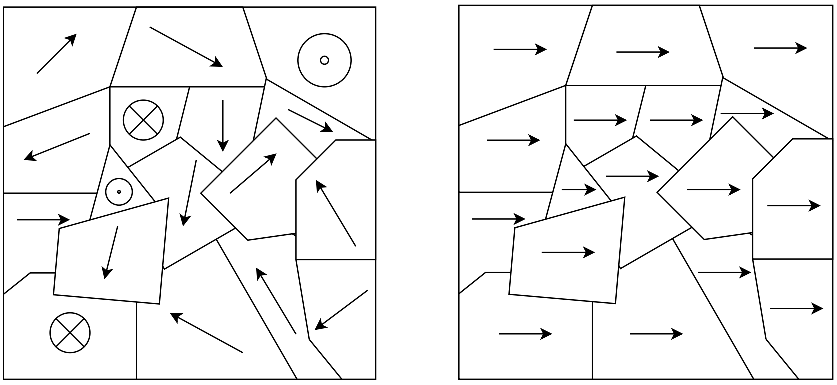
Macroscopically, ferroelectric materials are not homogeneous and have an incoherent structure with small areas of local homogeneity. These areas are called *domains* and they have discrete boundaries called *domain walls*. The size of the domains can vary between a few nanometers, up to one micron, and the most important property is that each domain has its own ferroelectric orientation. Although each domain has its own spontaneous polarization, eventually the cumulative polarization of all the domains cancels out and the material, as a whole, is neutral. Figure 2.3 shows the difference between



**(a)** Paraelectric phase

**(b)** Ferroelectric phase

**Figure 2.2:** The two phases of lead titanate and of similar materials.



**(a)** Domain polarization before external field

**(b)** Domain polarization after external field

**Figure 2.3:** Domain polarizations before and after the application of an external field

a material without ferroelectric homogeneity, against one where ferroelectric polarization is observable.

Having described the domain structure and behaviour, it is easier to understand the phases of the hysteresis loop as shown in Figure 2.1. Starting with a pristine piece of ferroelectric material, no electric polarization is observable, hence both the polarization and the applied electric field are zero. By increasing the intensity of the externally applied electric field, the polarization of the material starts to rise, in the opposite direction of the applied electric field. Once all the domains have been re-oriented, the material has reached the *saturation polarization*; regardless of the intensity of the external electric field, there are no domains left to be re-oriented. At this point, if the electric field is completely removed, the polarization of the material will drop since some of the domains will not be able to retain their polarization. However, the majority of them will do so, and the material will keep its overall net polarization. This is called *remnant polarization* ($P_r$), while the polarization ”lost” due to the fallback of the domains, is called *non-remnant polarization*. At this point, if a reverse external field is applied, the polarization of the material will start to drop again, until it reaches a negative ”peak” value, ie. a negative saturation polarization. The behaviour is exactly the same as the one we described, but since the orientation of the electric field is reversed, the polarization of the material is also reversed. The polarization shows a reaction delay (or hysteresis) with regards to the electric field, hence the name hysteresis loop.

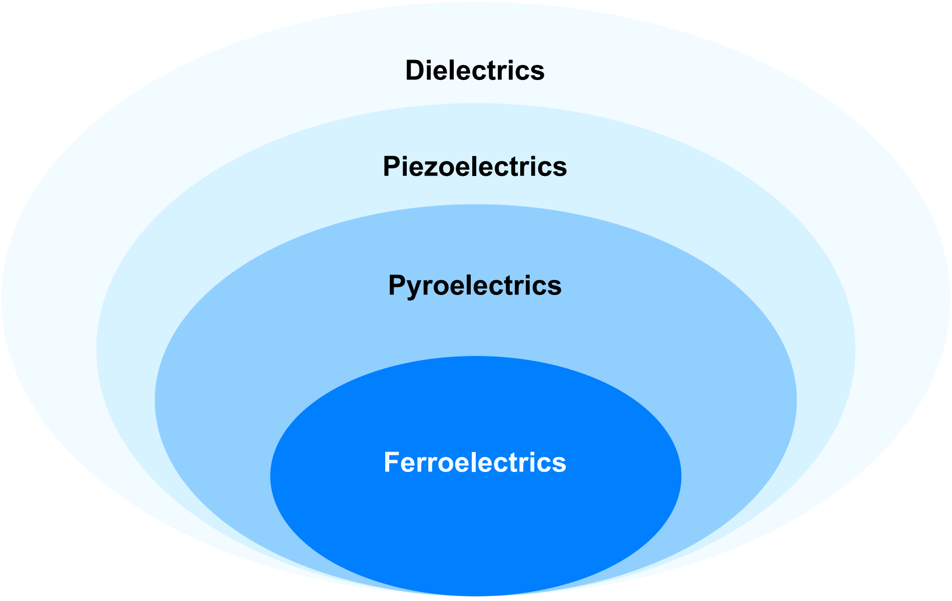


**Figure 2.4:** Relationship between dielectric, piezoelectric, pyroelectric and ferroelectric materials.

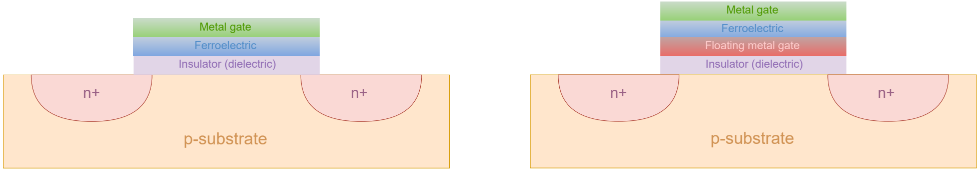
Ferroelectrics are a subset of the broader family of materials called *pyroelectrics*, which in turn are a subset of the *piezoelectric* family of materials. Figure 2.4 shows this relationship.

Until recently, the use of ferroelectric materials was limited in applications like FRAM, due to its significant advantages over Flash and EEPROM memories. FRAM technology employed a ferroelectric capacitor utilizing lead zirconate titanate (PZT) in the back-end-of-the-line (BEOL) configuration, which was linked to the drain of a front-end metal-oxide-semiconductor field-effect transistor (MOSFET). However, wide-scale adoption of ferroelectric technologies remained unrealized due to the difficulties associated with incorporating perovskite oxides into front-end semiconductor manufacturing processes. These challenges primarily encompassed issues related to perovskite etching, sensitivity to hydrogen, as well as limitations in thickness and cell size scaling beyond the 130 nm technology node [7].

The resurgence of research in ferroelectric devices was ignited by the discovery of ferroelectricity in binary oxides like silicon doped hafnia (Si:HfO$_2$) in 2011 [8]. Hafnia has played a pivotal role as a key facilitator of *high-K-metal-gate* (HKMG) technology for advanced logic transistors since the mid-2000s. Its compatibility with modern complementary metal-oxide-semiconductor (CMOS) technology, along with its scalability, holds the potential to unlock the promise of ferroelectric memories in high-volume semiconductor manufacturing, opening up opportunities for a wide array of commercial products [9]. Ferroelectric memories offer numerous prospects across various levels of the memory hierarchy, including embedded memory, main memory, storage-class memory, and long-term storage. In addition, many proposals are out for using ferroelectric devices for alternative computing paradigms such as non-von Neumann and neuromorphic computing [10], [11], [12].

## 2.2. An overview on FeFETs

The main concept of a FeFET refers to using a layer of ferroelectric material that is deposited on a semiconductor, in order to tune its surface conductivity. While this idea was discussed for the first time in 1957 [13, 14], as a device it was realized for the first time in 1974 by We et al., at the Westinghouse Research Laboratory [15]. This first prototype was an n-type MOSFET using a Bismuth Titanate film (Bi$_4$Ti$_3$O$_{12}$) as a ferroelectric material. In practice, the structure of a FeFET is straightforward; an additional ferroelectric (FE) layer is used in the gate stack, forcing the MOSFET to showcase two different threshold voltages, under two different polarization states. The two architectures used in the gate stack are the *Metal-Ferroelectric-Insulator-Semiconductor* (MFIS) and the *Metal-Ferroelectric-Metal-Insulator-Semiconductor* (MFMIS). Figure 2.5 shows these structures.



**(a)** MFIS structure

**(b)** MFMIS structure

**Figure 2.5:** The two different gate stack structures for FeFETs

FeFETs demonstrate negative capacitance (NC) and hysteresis in their I-V curves [16]. As we already mentioned, a switching of the ferroelectric polarization results in the shifting of the FeFET's $V_{th}$ and this is the key characteristic we aim to exploit. Apart from this fact, the FeFET effectively behaves identically to any other conventional MOSFET. The implications of such a structure are obvious; by controlling the polarization of the ferroelectric layer (and as a consequence, the $V_{th}$ of the transistor), it is possible to store binary values in a FeFET which then can be read afterwards.

The polarization of the ferroelectric layer can be controlled by applying appropriate pulses on the gate terminal. Two characteristics of these pulses can be controlled: amplitude and duration. The designer is responsible for deciding the values of these factors and tune them according to the application. FeFETs demonstrate two states, the first one called HVT (High $V_{th}$) and the other LVT (Low $V_{th}$). The circuit equivalent of a FeFET is shown in Figure 2.7 and it is essentially a ferroelectric capacitor in series with the gate of a conventional MOSFET. The $V_{GS}$ equivalent would be derived by the equation:

$$V_{GS} = (V_{FE} + V_{MOS}) \tag{2.2.1}$$

When a strong positive $V_{GS}$ is applied, it causes the ferroelectric layer to switch its polarity in such a way that it now manifests a positive voltage towards the MOSFET structure (Figure 2.7). This attracts electrons towards the gate and thus lowers the voltage needed to fully create the conductive channel (Low $V_{th}$). On the contrary, if a strong negative pulse is applied, the ferroelectric layer will be polarized in such way that a negative voltage will be manifested towards the MOSFET structure, hence, a repulsion of electrons will occur and the $V_{th}$ will be raised (High $V_{th}$).

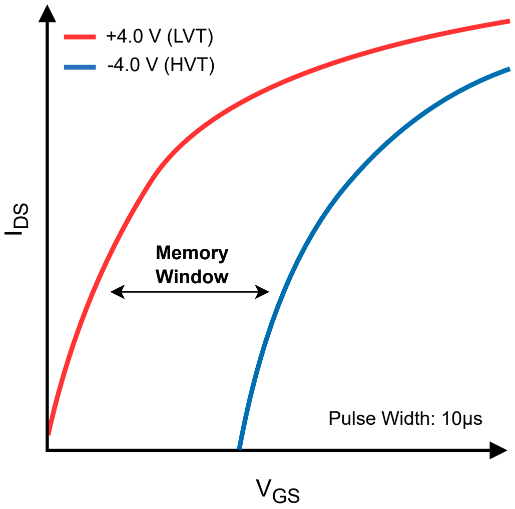


**Figure 2.6:** The I-V characteristics for the two FeFET states

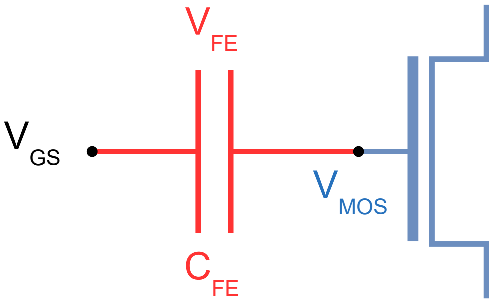


**Figure 2.7:** Circuit equivalent of FeFET device

Figure 2.6 shows the different I-V characteristic curves between the LVT and HVT states. The voltage window between these two is also called *memory window* and it is the voltage range within which a non-destructive read can take place. FeFET is a voltage-driven device, but current sensing is used to retrieve the value stored. As with regular MOSFETs, FeFETs' drain-source current is given by the following equation:

$$I_D = \frac{1}{2} \cdot \mu \cdot C_{OX} \frac{W}{L} (V_{GS} - V_{TH})^2 \tag{2.2.2}$$

FeFETs have demonstrated several advantages such as high density, low power, non-volatility, high read/write speeds, random access, high endurance and non-destructive readout. However, relatively short retention times is the major disadvantage of FeFETs which happens due to gate leakage current, depolarization field and carrier trapping in the gate-dielectric stack [17, 18].

## 2.3. FeFET Advancements and Materials

The previous section was an overview of the basic FeFET functional principles. However, factors like memory window, retention time and endurance, are defined by the materials used and the unique properties associated with them. As we already mentioned, in 1974, the first n-channel FeFET on silicon wafer with ($Bi_4Ti_3O_{12}$) as FE-material was reported. After one year, K. Sugibuchi et al., [19] reported the first p-channel FeFET on silicon wafer with same FE-material and structure. Both of them used a Metal-Ferroelectric-Semiconductor (MFS) structure.

Coercive field ($\varepsilon_c$) and remnant polarization ($P_r$) are the critical factors of FE-film used in FeFETs; both of them can be modulated with a capping layer, annealing conditions and doping concentration [20]. Dielectric breakdown and gate leakage are the serious issues associated with MFIS-structure based FeFETs, therefore, the gate stack must be carefully designed so that the program/erase voltage can be effectively distributed between the FE and insulator layers. Both gate leakage current and depolarization field ($E_{dp}$) degrade the retention time of FeFETs; the $E_{dp}$ reduces the polarization in the FE- layer which then leads to the reduction in memory retention time [21]. $E_{dp}$ can be calculated by the following formula:

$$E_{dp} = \frac{PC_{FE}}{\varepsilon(C_{IS} + C_{FE})} \tag{2.3.1}$$

$P$, $C_{FE}$, $\varepsilon$ and $C_{IS}$ represent polarization, FE-capacitance, permittivity of FE-film and semiconductor capacitance respectively.

Quite often, an insulating layer will be used to prevent the chemical reaction between Si-wafer and the FE-material. However, this will also create a depolarization field in the gate stack which reduces the memory retention time. In general, FeFETs with an MFIS structure demonstrate poor memory window and retention time. The MFMIS-structure has emerged as an excellent solution to mitigate the limitations of MFIS-structure based FeFETs [22]. FeFETs with MFMIS structure can be operated with lower voltages compared with FeFETs having MFIS structure due to the enhancement of FE capacitance.

The introduction of insulating layer leads to the increase of writing voltage due to the voltage drop across it. In order to reduce this voltage drop, a high-k material is usually preferred, however, usage of high-k dielectrics (e.g. $HfO_2$) creates a poor interface between Si-wafer and the high-k material that severely degrades the memory performance of FeFETs. These problems can be effectively mitigated by employing an intermediate electrode in between insulating and FE-layers, which leads to the MFMIS structure [23]. That is, FeFETs with MFMIS structure showcase perfect non-destructive readout and disturb free write operation compared with FeFETs with MFIS structure.

The memory performance of FeFETs is highly sensitive to temperature; it is reported that an increase in temperature leads to the increase of leakage current and the decrease of $I_{ON}/I_{OFF}$ and memory window [24], [25], [26], [27]. In 2011, Tang et al., [25] experimentally demonstrated an MFIS-FeFET using HfTaO buffer layer and $SrBi_2Ta_2O_9$ FE-film that showcased a memory window, $I_{ON}/I_{OFF}$ and endurance of 0.9 V, $10^7$ and $2 \times 10^11$ cycles respectively. In the same year, Böscke et al., [24], [26] discovered ferroelectricity in $HfO_2$, a major breakthrough in the development of FeFETs due to the fact that $HfO_2$ provides better scalability compared with strontium-bismuth titanate FE-material. In 2013, Mueller et al., [28] experimentally demonstrated a FeFET using Si-doped $HfO_2$ (Si:$HfO_2$) FE-film.

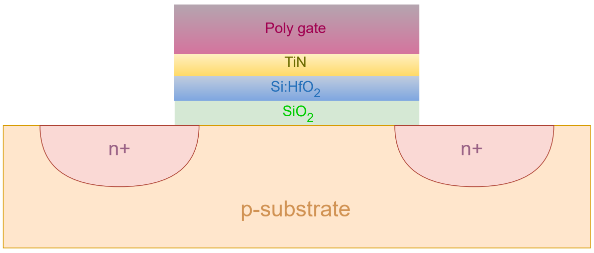


**Figure 2.8:** FeFET structure with silicon-doped hafnia as a FE layer.

In 2017, Chatterjee et al., [20] successfully fabricated a FD-SOI FeFET using gate last, self aligned process featuring $Hf_{0.8}Zr_{0.2}O_2$ FE-film that showcased a memory window, ION/IOFF, retention time and endurance of 0.5 V, $10^6$, 10 years and $10^7$ cycles respectively.

FeFETs are a field of active research and new structures and proposals are constantly published. A few indicative examples of their promising future can be found in the research by Zeng et al., [29] a FeFET was successfully fabricated, featuring HfN top electrode and $Hf_{0.5}Zr_{0.5}O_2$ (HZO) FE-film that showcased a MW and endurance of 0.92 V and $10^4$ cycles, respectively. In 2019, Huan Liu et al., [30] were the first to demonstrate a p-channel FeFET on Ge wafer featuring Ge-channel, TaN gate electrode and $ZrO_2$ FE-film that exhibited a memory window and endurance of 0.78 V and $10^7$ cycles respectively. S. Ohmi et al., [31] reported a FeFET having metal-FE-semiconductor architecture that exhibited a memory window and endurance of 1.2 V and $10^10$ cycles, respectively. It was also reported that forming $HfO_2$ FE-film with Kr/O2-plasma sputtering, offers superior polarization, reduced leakage current, improved ferroelectric capacitance, improved the drain current and provided superior memory window and endurance characteristics compared with $HfO_2$ formed by using Ar/$O_2$-plasma sputtering. Finally, in 2022, Dutta et al., [32] successfully fabricated a interfacial layer-free FeFET featuring $Hf_{0.5}Zr_{0.5}O_2$ FE-film and indium tungsten oxide (IWO) semiconductor channel that offers high speed, low voltage memory operation and exhibited a MW and endurance of 1.6 V and $10^{11}$ cycles, respectively. The elimination of interfacial layer efficiently minimizes the charge traps which in turn lead to the reduction in write voltage and improvement in endurance.

Large write voltage remains a serious issue in FeFETs, mainly due to the voltage drop across the interfacial layer. To reduce write voltage, the electric field across the interfacial layer should be reduced, however, it has been shown that this causes a severe degradation of the memory window. Reducing write voltage without degrading memory window is a ongoing challenge in the development of FeFETs [33].

For this thesis, the experiments took place using a FeFET model calibrated with data from fabricated FeFETs. The FeFETs were build based on an MFMIS structure and using an (HZO) $Hf_{0.4}Zr_{0.}O_2$ ferroelectric layer with 10nm thickness [34].

## 2.4. Challenges of $HfO_2$ Integration

Despite significant progress since its discovery in 2011, the implementation of ferroelectric $HfO_2$ still presents certain key non-ideal properties that affect its observed behavior. This section addresses some of the most commonly discussed challenges, according to [35].

### 2.4.1. Depolarization Field

The polar nature of ferroelectrics inevitably leads to the generation of a depolarization field, which significantly impacts the characteristics of thin films. This depolarization field is caused by uncompensated charges at the interfaces. From device fabrication, the presence of a dielectric layer at the ferroelectric/metal interface, often referred to as a dead layer, is commonly present [36]. This dielectric layer hinders the screening of surface charges, as the counter charges in the metal electrode are spatially separated from the surface charge. The unscreened charges create a depolarization field across the ferroelectric, reducing the effective polarization and limiting thickness scaling, with its impact becoming more pronounced in thinner films [37–39]. Even in the case of having no dielectric interface layer, a depolarization field will still be present, since even metal electrodes will have a non-zero screening length. However, the larger coercive field in $HfO_2$-based ferroelectrics, compared to the traditional perovskite ferroelectrics, should enable $HfO_2$ to withstand higher depolarization fields [36, 40]. The depolarization field is the cause of reduced memory state retention for FeFETs. The understanding and control of depolarization fields in ferroelectrics is of key importance for their use in technological applications.

### 2.4.2. Retention

In ferroelectric devices, *retention* refers to the ability of a ferroelectric material to maintain its polarization over time, in the absence of an external electric field. Retention is an important characteristic because it determines how long the ferroelectric state can be maintained, once the external electric field is removed. Retention performance is influenced by various factors, including material properties, device architecture, and operating conditions [35, 41]. It is of great importance to design FeFETs in a way that ensures the reliability and longevity of the device, especially when it comes to applications where stored information needs to be preserved for extended periods.

### 2.4.3. Wake-up

It is common in $HfO_2$-based ferroelectrics to observe what is called a ”wake-up” effect, which is a gradual increase of the $P_r$, during voltage cycling. It is typical that the wake-up effect is especially strong during the first, approximately, 1000 cycles. Initially, it was believed that the wake-up effect only stemmed from the redistribution of oxygen vacancies [35, 42–44], causing domains in the film to de-pin which results in an elevated $P_r$. However, recent evidence suggests that the main contribution is a phase transition from the t- to the o-phase, or even m-phase to o-phase during voltage cycling, offering an alternative explanation for the observed increase in $P_r$. Currently, substantial efforts are underway to mitigate the presence of the wake-up effect in $HfO_2$-based ferroelectrics by engineering the thermal treatment [35, 45], introducing La dopants [46], and optimizing the deposition conditions.

### 2.4.4. Endurance

The ability of a ferroelectric material to undergo repeated cycles of polarization switching without significant degradation, is called endurance. However, $HfO_2$-based ferroelectrics, often experience hard breakdown before any significant degradation of the polarization occurs. For this reason, the ”cycles until breakdown” is typically what is reported in the literature. Reported endurance values for $HfO_2$ ferroelectrics span a range from $10^4$ to $10^{11}$ cycles. Comparison of different devices with regards to endurance should be done carefully, since endurance is dependent on both frequency and field magnitude. A better metric could be the time until break down, which, as demonstrated in the work by Toprasertpong et al., is constant, independently of the cycling frequency used [47]. Additionally, thinner films are expected to have improved endurance compared to thicker films, since the breakdown field decreases with thickness [35, 47].

### 2.4.5. Fatigue

A ferroelectric material that goes over repeated switching of its polarization under the influence of an external electric field, it can experience *fatigue*. These phenomenon is manifested as the reduction of the material’s ability to maintain a stable and well-defined polarization state. Several mechanisms contribute to fatigue in ferroelectrics, with one common factor being the accumulation of defects in the material, during the polarization-switching process. Over time, these defects can impede the movement of domain walls (pinning) and eventually hinder the ability of the material to maintain a stable polarization state. As a result, fatigue can play a major role in the material’s reliability and performance.

## 2.5. Structures and Applications

Ferroelectric FET technology is emerging in many application domains ranging from non-volatile storage, to reconfigurable hardware. Some examples of promising FeFET based applications are emulating atomic neuromorphic operations [48], hyper-dimensional encoding [49], multiply-accumulate operation crossbars [50] and energy/area efficient FPGA fabrics [51]. Clearly, both Academia and Industry consider using FeFET devices in promising solutions for various limitations faced by contemporary computing technology.

### 2.5.1. Memories

FeFETs are by definition a very promising candidate for new types of memories.

**FeRAM**

Ferroelectric random-access memory (FeRAM) aims to replace dynamic random-access memory (DRAM) that constitutes the main memory of today's devices. The FeRAM architecture, resembles DRAM, but with one key difference, that the insulator in the capacitor is now a ferroelectric film, allowing for non-volatile charge storage. The read and write procedures are similar to DRAM and as the access transistor is opened, a voltage is applied across the ferroelectric capacitor. Then, if polarization reversal occurs, the sensed current will be significantly larger than a non-switched one. The non-volatile nature of FeRAM removes the need for refresh, which makes it significantly more power efficient. Recently a non-destructive read process for FeRAM was demonstrated, focusing on the difference in capacitance between the two polarization states, allowing for read endurance beyond $10^11$ cycles [52]. Moreover, the charge density offered by ferroelectrics enables the integration of smaller capacitors. Despite these advantages, challenges such as limited endurance, and high write voltage still require further improvements before consumer products can be realized [53].

**FeFET Crossbar Memories**

As already mentioned above, FeRAM aims to replace DRAM [35] and with regards to non-volatile storage, similar arguments can be made for the case of FeFETs aiming to be used as a replacement for Flash memory. Flash memory is a non-volatile semiconductor storage technology widely used in electronic devices such as USB drives, solid-state drives (SSDs), and memory cards. Its operation is based on a threshold voltage shift to realize two different memory states. It achieves this through the principle of storing charge on a floating gate buried between the gate electrode and transistor channel. As charges are trapped in the floating gate, the electric field of the gate electrode is screened. This increases the threshold voltage of the transistor. Although flash memory is well established, it suffers from limited endurance (typically < $10^5$) and very high write energy consumption (~1 nJ/bit) [35]. FeFETs, offers improved endurance, significantly improved energy efficiency and have the potential to replace flash memory in non-volatile applications. Using their two-level threshold voltages, they can act in a similar fashion to Flash, for memory applications. Another concept has been introduced where instead of integrating the ferroelectric directly in the gate stack of the transistor, a ferroelectric capacitor which shares one of its electrodes with the transistor gate, is connected in series with a traditional non-ferroelectric transistor. This device approach is called a ferroelectric-metal-field-effect-transistor (FeMFET) and allows for reduced programming voltages and generally better reliability when compared to the standard FeFETs[54].

### 2.5.2. Neurons

Neuromorphic computing (like *spiking neural networks* - (SNNs)) is a relatively novel computing paradigm which attempts to mimic the human brain in the way computations are done. However, it is fundamentally different that the more popular, computer science driven, *machine learning* and its variants (such as *deep learning*) [55]. This attempt to capture brain-like features such as computation using spikes, holds the promise of improving the energy efficiency of computing platforms. In order to achieve an energy efficiency significantly better than the current CMOS-based neuronal circuits, researchers have turned to the emerging NVM devices.

The general working principle of a SNN can be described as follows: when a synapse receives a spike (i.e. an action potential), from its pre-synaptic neuron, it emits a post-synaptic potential (PSP). The PSP in turn stimulates the membrane potential of the post-synaptic neuron and the neuronal membrane potential exhibits temporal evolution where it integrates the PSPs. When the membrane potential crosses a threshold, the post-synaptic neuron fires, emitting an output spike.

FeFETs are an excellent candidate for implementing a hardware spiking neuron, due to their partial polarization characteristics. This feature enables the polarization change of the FeFET layer through a sequence of smaller potentiation pulses, instead of a single longer pulse. It is evident that, the inherent ferroelectric polarization switching dynamics, closely resemble the neuronal membrane dynamics of the spiking neuron and can be used to emulate one. Figure 2.9 shows the proposal of spiking neuron based on a FeFET and three conventions transistors [56].

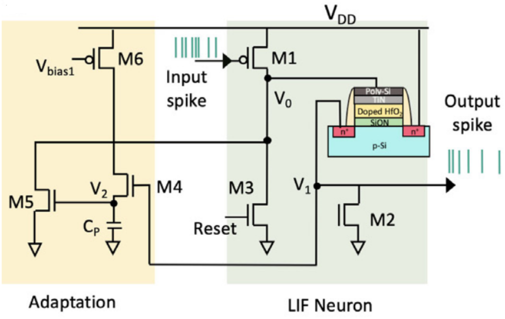


**Figure 2.9:** FeFET-based spiking neuron

### 2.5.3. Miscellaneous

**Multiply-accumulate operation crossbar**

In-Memory Computing is a computational paradigm that uses memory cells organized in crossbar arrays to perform parallel, in-situ execution of operations such as Matrix-Vector Multiplications. While SRAM and DRAM cells could be used, challenges like static leakage, continuous refreshing and complex peripheral circuitry, are significant drawbacks. Emerging devices, such as ReRAM and FeFETs have reduced size and improved data retention, and can thus be attractive alternatives. ReRAMs, which were among the first devices used for building IMC crossbars. Despite having significantly smaller memory footprint compared to SRAMs and DRAMs, they come with some drawbacks too; sneak path effects and crossbar parasitics can significantly complicate their use. FeFET-based crossbars, can be used as an alternative due to their minimal leakage currents, compactness, compatibility with CMOS technology, and the elimination of selector devices, enabling the creation of larger and more efficient crossbar arrays.

Figure 2.10 shows such a FeFET crossbar, used for IMC operations. The method proposed by [57] occurs in two phases. During the first phase, the BLs are precharged to a potential denoted as $V_{precharge}$. In the second phase, the WLs representing the input activations are enabled simultaneously. At the same time, the source lines (ScLs) are connected to a lower potential ($V_{SS}$). This configuration allows the BLs to discharge linearly based on the stored data (programmed FeFET state) and input activations. Consequently, this arrangement facilitates a MAC operation through integration along the BL.

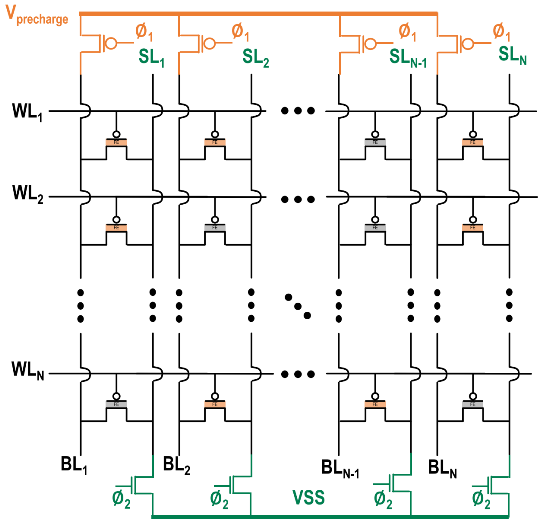
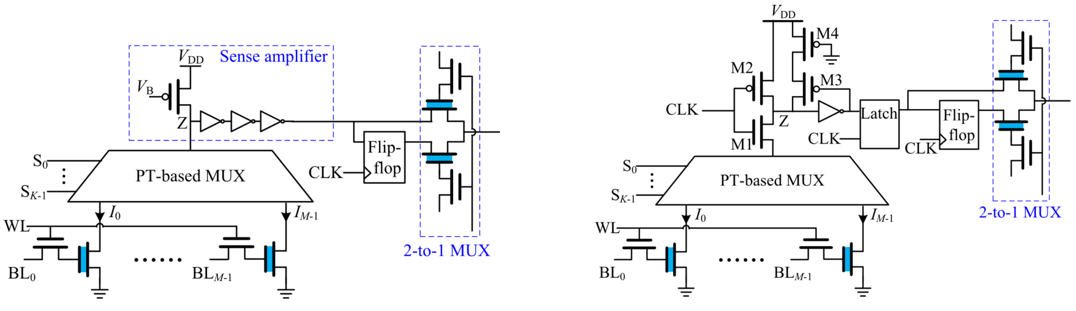


**Figure 2.10:** FeFET crossbar for IMC operations

**FPGAs**

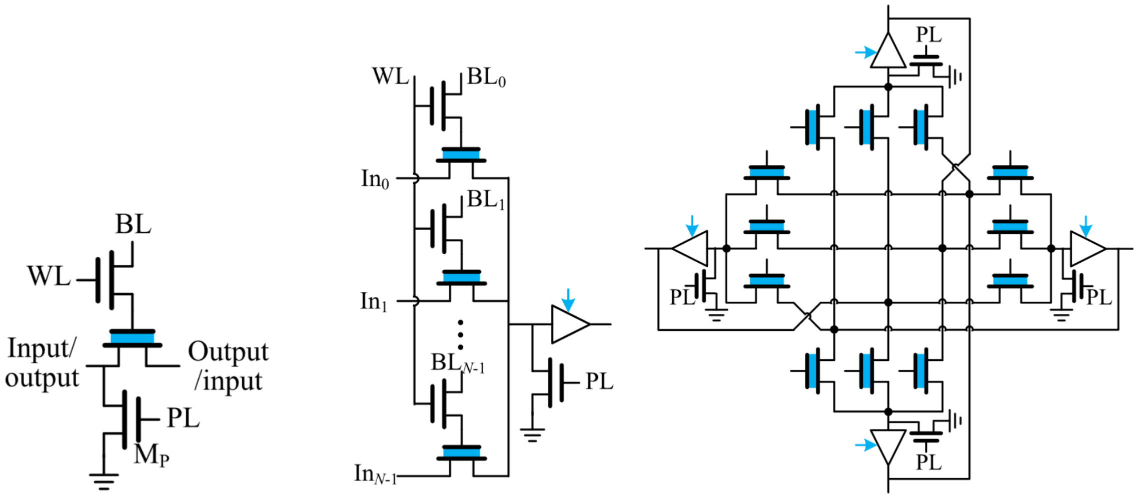
Since ferroelectric field-effect transistors (FeFETs) have the potential to reduce the power and area by integrating non-volatile storage elements with logic, this fact can be exploited to design lookup tables (LUTs) and routing switches. These elements have obvious utility in Field-Programmable Gate Arrays (FPGAs). It has been proposed in [51] that basic building blocks for FPGAs



**(a)** Static LUT

**(b)** Dynamic LUT

**Figure 2.11:** Two FeFET-based FPGA LUTs



**(a)** Routing switch unit

**(b)** Routing multiplexer

**(c)** Routing crosspoint

**Figure 2.12:** Routing elements with FeFETs

like lookup tables (LUTs) and routing elements can be beneficial in terms of area, energy, or both. Figure 2.11 shows two FeFET-based FPGA LUT structures.

It is also proposed that FPGA routing units can be built using FeFETs. Figure 2.12 shows the routing elements built with FeFETs, from a simple routing switch unit, up to a routing crosspoint. The research towards this direction is very promising: despite the fact that programming the FeFETs consumes about $35\times$ more energy and is over four orders of magnitude slower than programming an SRAM cell, much faster programming times (<50 ns) have been reported [48] with some reports as low as 10ns [58]. However, FeFET-based LUTs are reported to have a advantage over many aspects against competing technologies, when it comes to reading/accessing them. In the same work, the authors reported a 55% reduction in the number of transistors needed for a LUT, when compared to traditional CMOS, while the power-delay product (PDP) was reduced up to $16\times$.

### Gates, Latches, Flip-Flops

Considering the gate of a FeFET as one input and the stored value as the second, a FeFET can be used as a 1-transistor, 2-input AND gate, which will produce the result in two phases. In the first phase, the input ’B’ will have to be stored in it, while in the second, the input ’A’ will be asserted on the gate pin. Considering a $V_r = 0.6V$, an LVT (logic 1) value of 0,4V and an HVT (logic 0) value of 0.8V the following truth table can be realized:

Batteryless devices leveraging energy-harvesting methods (such as solar panels, vibrations, and radio frequency (RF)) can frequently lose their computing progress in case of abrupt power outages. In these cases, the devices can be benefited by a mechanism that stores their current computing state, which

| A | B | OUT |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

**Table 2.1:** Truth table with representing voltages for a 1FeFET AND gate

| VG | Vth | IDS |
|---|---|---|
| 0V | 0.8V | - |
| 0.6V | 0.8V | - |
| 0V | 0.4V | - |
| 0.6V | 0.4V | YES |

**Table 2.2:** Truth table with representing voltages for a 1FeFET AND gate

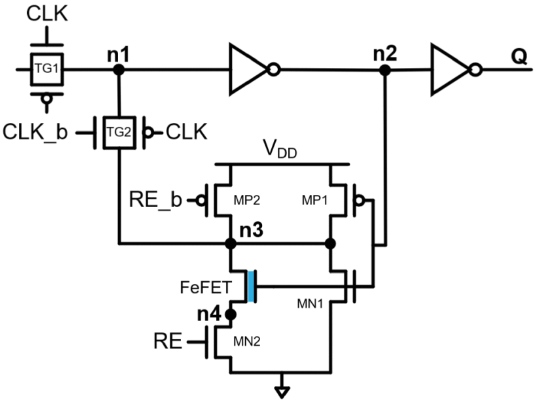


**Figure 2.13:** FeFET-based non-volatile flip-flop

can then be resumed after the device receives enough power again. In this, ”intermittent computing” paradigm, FeFETs can play a crucial role by being used in a new type of non-volatile latches and flip-flops (NVFFs). Several researchers have worked towards this direction and promising results have been reported. An indicative example is the NVFF designs reported by Kim et al [59], where it offers a per-cycle backup operation while it requires only 19% more area compared to a conventional flip-flop. Figure 2.13 shows the slave latch which is build around a FeFET device.

## 2.6. Simulation of FeFETs

FeFET devices are modelled using two main approaches: the Landau-Khalatnikoff (L-K) model which is based on time-dependent equations that describe the relationship between the polarization ($P$) and the electric field ($E$) [60] and the Preisach model, built upon the fact that a FE thin-film consists of multiple independent domains, with a distribution of coercive fields [61], [34]. While L-K model is useful in specific cases, it also has a few drawbacks. The first being that it assumes a single domain FE material, while in practice it is usually poly-crystalline with multiple domains. The second limitation is that the $V_{GS}$ applied to the FeFET can either keep the device polarization or switch it, as reported in [62]. It is impossible to turn the transistor completely off and maintain polarization, which means that the model cannot reproduce the basic non-volatile operation of FeFET [62]. There are several models that have been proposed for both approaches, both compact and physical [34, 63–65].

# 3

# Overview of Memory Decoders

In this chapter we will discuss about memory decoders and we will give the necessary background in order to understand the concepts described later. Memory decoders are an integral part of almost any type of memory, regardless of the level, type or the use case for it. Since memories are built in an array fashion, having a system that enables locating a specific area within that matrix, is needed. Depending on the usage of the decoders, they can be classified into two large categories, row decoders and column decoders. In this chapter we will focus on decoders for random-access memories (RAM) like SRAM or DRAM.

## 3.1. Row Decoders

A decoder is a circuit with many inputs and many outputs. Its job is to reduce the number of select signals by $log_2$, so if the number of inputs is N, then the number of outputs is $2^N$.

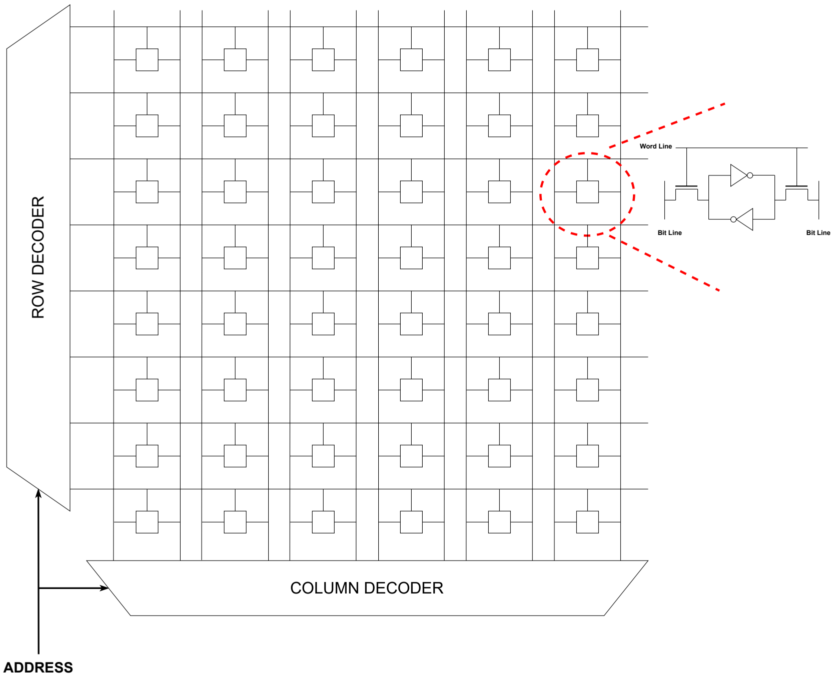


**Figure 3.1:** An SRAM array with decoders

Before we describe the specifics of row decoders it is important to review how exactly a memory array is built and how the decoders fit into the peripheral circuitry. Figure 3.1 depicts an SRAM array together with the row and the column decoder. The address signal is split into two parts, and being fed to the decoders. The row decoder is responsible for activating the correct word line, while the column decoder should activate the correct bitlines.

As a standard decoder design we can consider the use of multiple AND gates, one for every word line (WL). Each one of these AND gates has a unique combination of address inputs, or their inverted values. This means, that if we want to decode an 8-bit address (for a hypothetical $256 \times 256$ SRAM array), this would effectively lead to a total of 256 lines and consequently, to 256 8-input AND gates. Eventually the logic function for activating the word lines would be like this:

$$WL_0 = \bar{A}_7\bar{A}_6\bar{A}_5\bar{A}_4\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0 \tag{3.1.1}$$

$$WL_{255} = A_7A_6A_5A_4A_3A_2A_1A_0 \tag{3.1.2}$$

Of course it is possible to use NOR logic:

$$WL_{255} = \overline{A_7 + A_6 + A_5 + A_4 + A_3 + A_2 + A_1 + A_0} \tag{3.1.3}$$

### 3.1.1. Static Decoders

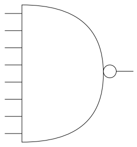


**Figure 3.2:** 8-input NAND gate. Each gate can drive a wordline.

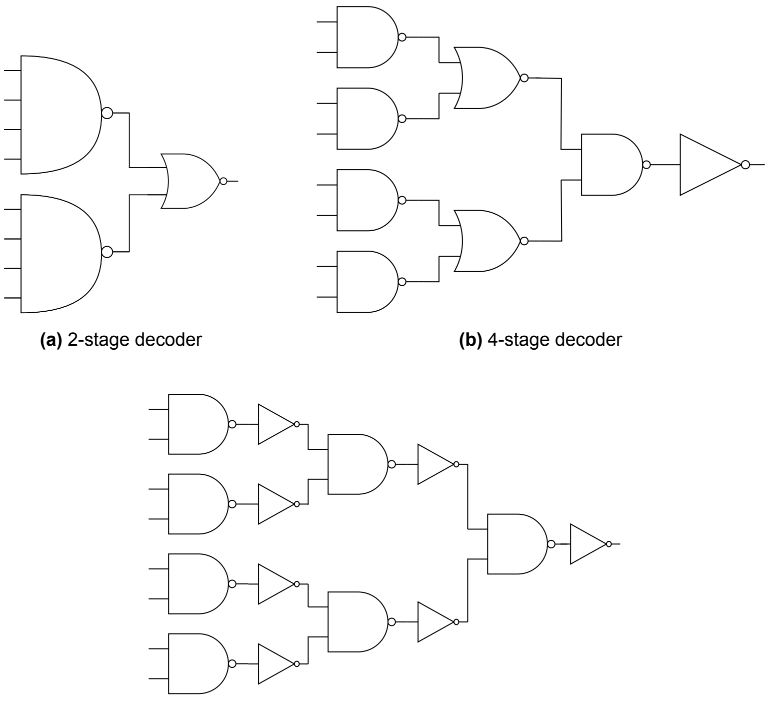
The first way to implement an address decoder is by using static logic. This means that regular combinational logic is used without any clock signals or precharge phases. There are many ways to implement a static decoder and the most obvious is by using a sequence of AND (or NAND) gates, so that each one of them activates only with a specific combination of the input (address) signals. We have already shown this possibility in the equations 3.1.1 and 3.1.2. Figure 3.2 shows this NAND gate.

Of course, the problem with a gate like this is the large fanout, 256 bit cells, which can be prohibitive in terms of gate size and/or delay. Another approach could involve splitting the inputs into two smaller NAND gates and then wire them in a 2-input NOR, effectively having two stages. Splitting them to even more stages is also feasible and a few of these designs are shown in Figure 3.3. At this point the design with the best logical effort should be considered, which happens to be the 6-stage decoder. The implementation of such a decoder however, doesn't come without problems: each of the addresses is essentially driving 128 gates which is an enormous fanout, using long wires with overall high capacitance. On top of that, there will be problems with the bitcell pitch, essentially not having enough pitch space to fit all the signal wires.

**Predecoding**

A solution to the aforementioned problems is implementing a predecoder in the architecture. The idea of a predecoder comes from the fact that neighbouring addresses differ only by one bit, hence a lot of the decoding hardware can be shared and reused. By grouping together inputs from the boolean expression, smaller decoders can be created; their outputs can be ANDed producing the final output. An alternative predecoding architecture for the example we already mentioned could be using 2 4:16 predecoders (depicted in figure 3.6). This architecture allows us to decode the 4 LSBs separately from the 4 MSBs and then AND the output of the predecoder circuits effectively producing the word line signal. This requires 256 AND gates, however, these AND gates are now smaller since they are having only two inputs, in addition, each address driver is driving only 8 gates which is far less than with the naive approach (128 gates) and thus the capacitance is significantly reduced. This architecture leads also to enormous area reductions, since less gates are needed (due to sharing) and the final stage 2-input AND gates are easily placed in the tight bit cell pitch.

Using small predecoders as opposed to bigger ones does not produce better results in terms of area or performace. Figure 3.6 depicts a rough schematic comparison between two different approaches, the one using two 4:16 predecoders and the second using four 2:4 predecoders. The consequences of choosing the one over the other can be of great impact; the design using the smaller predecoders suffers from inefficiencies such as pitch fitting issues (due to the large AND gates that drive the word lines), switching higher capacitances due to the length/number of wires and demonstrating worse logical effort for the predecoder stages. Hence, it is important to remember that predecoding should be done with as large predecoders as possible.

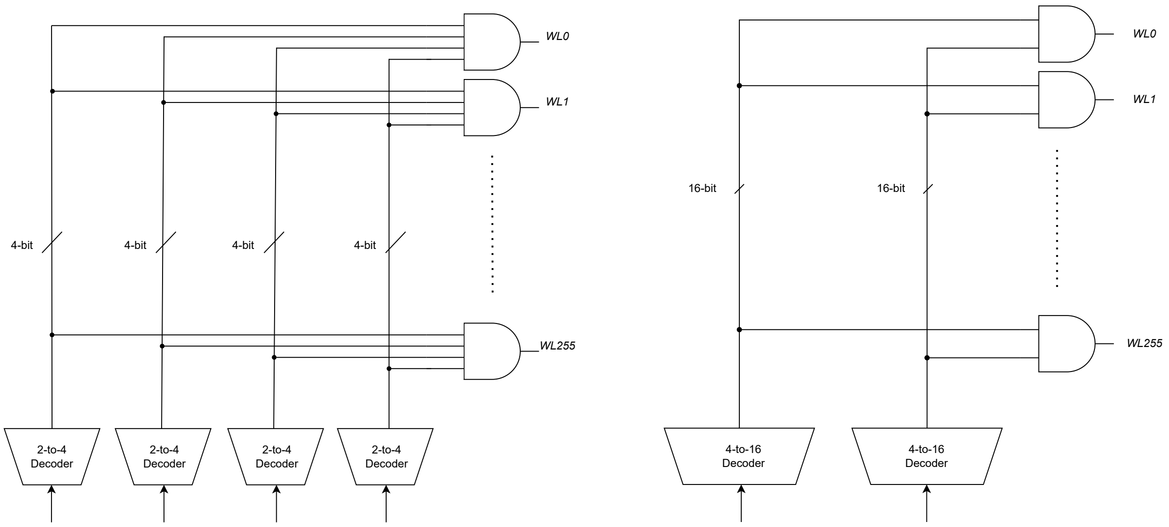


**(a)** 2-stage decoder

**(b)** 4-stage decoder

**(c)** 6-stage decoder

**Figure 3.3:** Decoders with different number of stages.



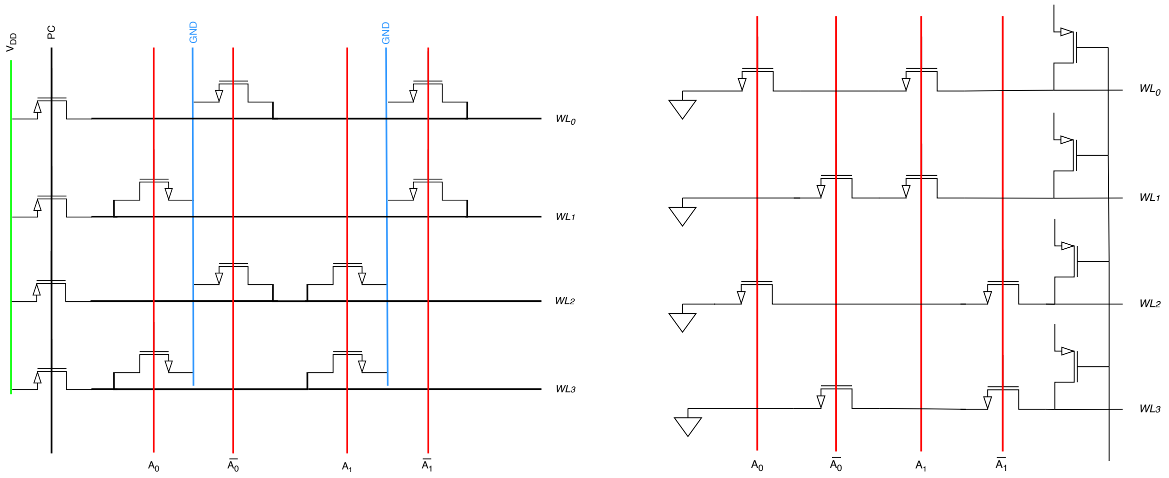
**(a)** Decoder architecture with four 2-to-4 predecoders.

**(b)** Decoder architecture with two 4-to-16 predecoders.

**Figure 3.4:** Comparison of predecoder architectures.

### 3.1.2. Dynamic Decoders

An alternative to the static decoder approach we described earlier would be the use of dynamic logic. Decoders designed with dynamic logic could have either a NOR or a NAND structure and their main advantage is speed, due to the limited usage of pMOS transistors and the lower capacitances being driven. However, the static power dissipation is higher and the circuit is more complex to design. For these reasons dynamic decoders are preferred in high speed designs and are considered as the state-of-the-art of the industry. Figure 3.5 shows the 2 different architectures of dynamic decoders. The advantage of a NOR topology is that it is faster, since only one transistor is required to open for the line to be discharged, however, this also requires that each transistor is connected to the ground as shown in Figure 3.5a. Even though the use of ground lines to each transistor looks trivial for small decoders, for larger ones it imposes significant complexity. On the other hand, NAND decoders are slower due to the fact that all transistors on the line need to open for the discharge to occur; their manufacturing is much simpler though, since each one of the transistors is just diffusion connected to their neighbour.



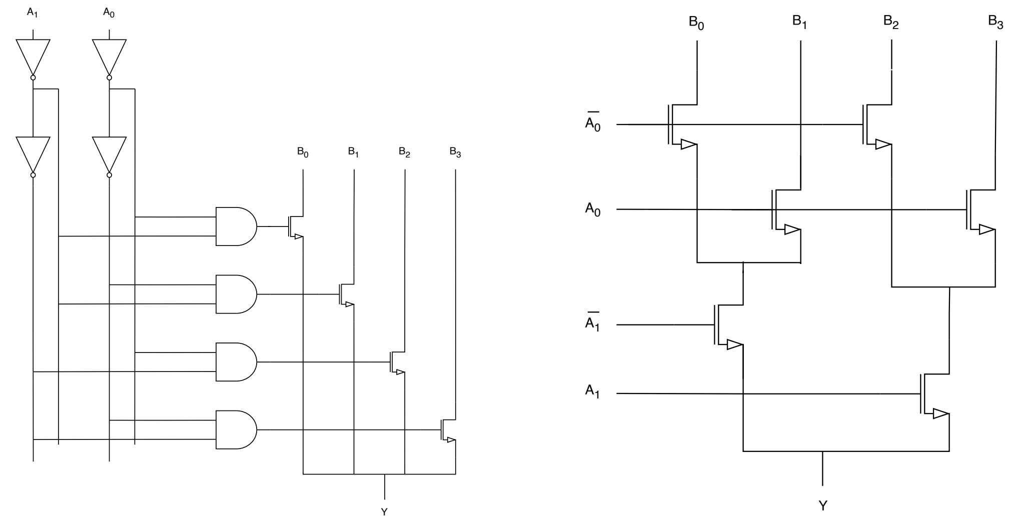
**(a)** NOR-topology dynamic decoder.

**(b)** NAND-topology dynamic decoder.

**Figure 3.5:** 2-input dynamic decoders

## 3.2. Column Decoders

The column decoder is another essential part of RAM memories, which is responsible for multiplexing the signals from the bitlines. Again, there are a few options for the architecture of a column decoder, and those are dictated by the goals of the designer. The first (and straightforward) solution would be the use of a pass-transistor logic multiplexer coupled with a decoder. Figure 3.6a show the schematic of this architecture. It consists of a 2-to-4 decoder that each one of its outputs drives a single transistor, opening the correct bitline.



**(a)** NOR-topology dynamic decoder.

**(b)** NAND-topology dynamic decoder.

**Figure 3.6:** 2-input dynamic decoders

A second option would be to use a 4-to-1 tree decoder. With this approach, assuming a $2^k : 1$ multiplexer, the result is a design of $k$ transistors, wired in series. Figure 3.6b depicts this design. The removal of any external decode logic causes a significant reduction in the total circuit area. However, the delay increases quadratically with the number of address bits and thus, the number of transistors.

## 3.3. Address Decoder Faults

In the last section of this chapter we will briefly discuss address decoder faults. These faults concern faults in the address decoder structure. According to [66], there are four main types of address decoder faults:

1. Given a certain address, no cell is accessed
2. There is no address with which a certain cell can be accessed, thus, a certain cell is *never* accessed
3. Given an address, multiple channels are being accessed simultaneously
4. A certain cell can be accessed through multiple addresses

A particular type cannot occur alone, but it has to co-exist in combination with at least one more type. For instance, if a type 1 fault occurs, then inevitably either a type 2 or a type 3 fault occur too. Address decoder faults are also divided in two more types: *simple memory cell array* faults and *coupling faults*, i.e. between memory cells. In this work we will examine both single-cell and coupled faults, however, without doing any of the classification of the previous categories.

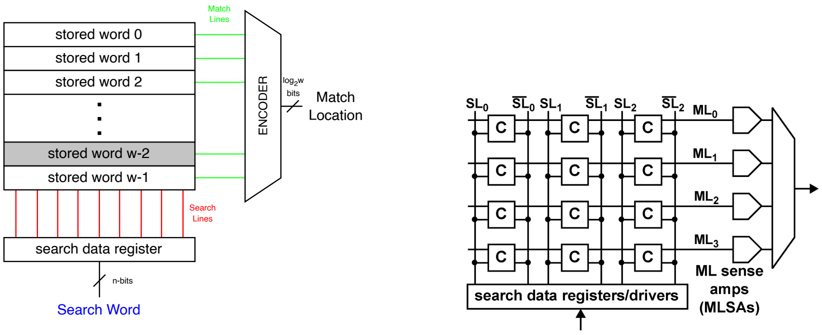
# 4

# Content-Addressable Memories

In this chapter we are briefly discussing the concept of Content-Addressasble Memories (CAMs), which lies in the core of this thesis. We are looking into the different technologies used to implement them, the applications that can be benefited from the use of them, as well as the different types of CAMs.

## 4.1. CMOS-based Content-Addressable Memories

A content-addressable memory is a type of memory which accepts a set of data as an input, compares it to the data it has already stored and returns one or more boolean values as output, based on whether the data was found within the memory or not. It essentially functions in the opposite way from a regular memory; in a conventional memory, data is accessed by providing the memory address that should be read from, while in a content-addressable memory, (a portion of) data values are provided as a query to the memory. The memory then compares the query against all stored entries and returns the location of the match. This process is highly parallelizable, allowing a CAM to search through thousands of entries simultaneously. The read time is comparable to that of a conventional memory, however, the power consumption is considerably higher, since all memory rows are activated in parallel. This fundamental difference in their operation principle makes CAMs useful for applications that require high-speed table lookup, such as networking routers (for packet forwarding and classification), databases and cache memory in computers. Figure 4.1a shows the basic concept of a CAM array. The search word matches location $w-2$ as indicated by the shaded box. The MLs provide the row match results. The encoder outputs an encoded version of the match location using $\log_2 w$ bits.



**(a)** Conceptual view of a CAM containing $w$ words.

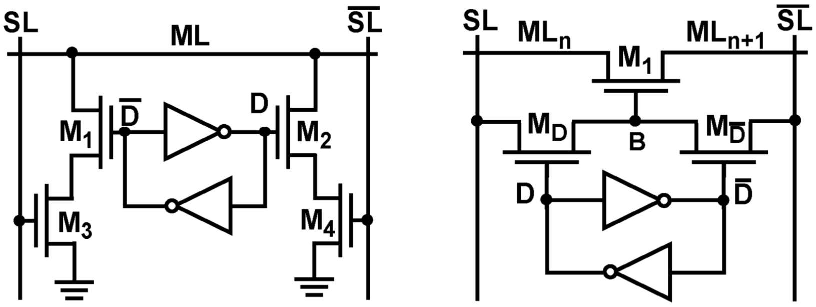
**(b)** Architecture of a CAM array, accompanied by the peripheral circuitry.

**Figure 4.1**

The main variants of CAMs are *binary* CAMs and *ternary* CAMs. Binary CAMs work by storing and comparing binary values, which make their structure simple but it also limits their possible applications to those that rely on exact data matches (e.g. caches). Ternary CAMs extend the functionality of binary CAMs by allowing three different values to be stored and queried, i.e. 0, 1 or X - the latter one being interpreted as ”don’t care”. TCAM cells are significantly more complex to design and manufacture, but they offer flexibility which can be useful in applications like network equipment, where wildcard searches are essential.

Figure 4.1b allows us to have a closer look into the architecture. The figure depicts a CAM storing 4 words, each word containing 3 bits arranged horizontally (each one of them corresponding to a CAM cell). Each word has an associated match line (e.g., $\mathrm{ML}_0$, $\mathrm{ML}_1$, etc.) that drives a match line sense amplifier (MLSA). In addition to this, there is a differential search line pair for each bit of the search word (e.g., $SL_0$ - $\overline{SL}_0$, $SL_1$ - $\overline{SL}_1$, etc.).

A CAM search operation is being initiated by loading the query into the search-data registers, precharging all match lines to a high voltage, temporarily setting them in the match state. Subsequently, the search line drivers broadcast the search word onto the differential search lines. Each CAM core cell then compares its stored bit against the bit on its corresponding search lines. Match lines (MLs) with all bits matching remain in the precharged-high state, while MLs with at least one mismatched bit discharge to ground. The MLSA then detects whether its ML indicates a match or miss condition. Finally, the encoder converts the ML of the matching location into its encoded address.



**(a)** 10-T NOR CAM cell

**(b)** 9-T NAND CAM cell

**Figure 4.2:** Binary CAM cells

There are a few different ways to build a CAM cell using conventional CMOS technology. Figure 4.2a shows the structure of a 10-transistor NOR CAM cell, while Figure 4.4b show a CAM cell based on a 9 transistor NAND structure. In the first one, the pull-down paths connect in parallel resembling the pull-down path of a CMOS NOR logic gate (hence the characterization NOR). Similarly, in the second one, the serial nMOS chain of all $M_i$ the transistors resembles the pull-down path of a CMOS NAND logic gate, thus a match condition for the entire word occurs only if every cell in a word is in the match condition. To simplify the schematic, the nMOS access transistors and bitlines which are used to read and write the SRAM storage bit, are omitted.

### 4.1.1. A special case: TCAMs

The NOR and NAND cells we have just described are binary CAM cells, which means that they can store either a logic ”0” or a logic ”1”. Ternary CAM cells, as we mentioned beforehand, can store on top of these values, also the value ”X” which signifies ”don’t care”. This essentially allows the value to be used as a wildcard, causing a match regardless of the input bit. This can be a very useful for packet forwarding in network routers.

The ternary state can be encoded using two bits, D and $\bar{D}$. In total, 4 symbols can be represented with the two available bit. This gives us an extra combination which is not used (since ternary uses only three symbols). In order to store a ternary value in a NOR CAM cell, the cell needs to be be extended with a second SRAM cell.

The two inputs, D and $\bar{D}$ connect to the left and the right pull-down paths, respectively, making the pull-down paths independently controlled.

The ”don’t care” value is being stored by setting both D and $\bar{D}$ to logic ”1”; this disables both pull-down paths and causes a cell match regardless of the values of the inputs. A logic “1” is stored by programming $D = 1$ and $\bar{D} = 0$. To store a logic “0” we program D=0 and $\bar{D}$=1. In addition to storing an “X”, the cell allows searching for an “X” by setting both $SL$ and $\bar{SL}$ to logic “0”. This acts as an external ”don’t care” that causes a match regardless of the stored bit. Table 4.1 shows the relationship between the values that are stored and the values need to be read or written.

| | **Stored Bits** | | **Search Values** | |
|---|---|---|---|---|
| **State Stored** | **D** | **$\overline{\text{D}}$** | **D** | **$\overline{\text{D}}$** |
| Logic ”0” | 0 | 1 | 0 | 1 |
| Logic ”1” | 1 | 0 | 1 | 0 |
| Don’t Care ”X” | 1 | 1 | 0 | 0 |

**Table 4.1:** The NOR CAM values for storing and reading data. Note that, even though D and $\bar{D}$ are not complementary, the complementary notation is preserved for consistency with the previous description of the binary cell architecture.

In cases where ternary operation is needed but only binary CAMs are available, it is possible to emulate the ternary operation by using two binary cells per ternary symbol [67].

In order to modify the ternary NOR cell to accommodate the ternary operation (Figure 4.3a), it has been proposed implementing the pull-down transistors using pMOS devices and complementing the logic levels of the search lines and match lines accordingly [68]. In a similar fashion, a NAND cell can be modified for ternary storage by adding storage for a mask bit at node M, as depicted in 4.3 [69], [70]. When storing an “X”, the mask bit is set to “1” which forces the transistor to open , regardless of the value of D resulting to an always matching cell. In addition to storing an “X”, the cell allows searching for an “X” by setting both $SL$ and $\bar{SL}$ to logic “1”. Table 4.2 lists the stored encoding and search-bit encoding for the ternary NAND cell.

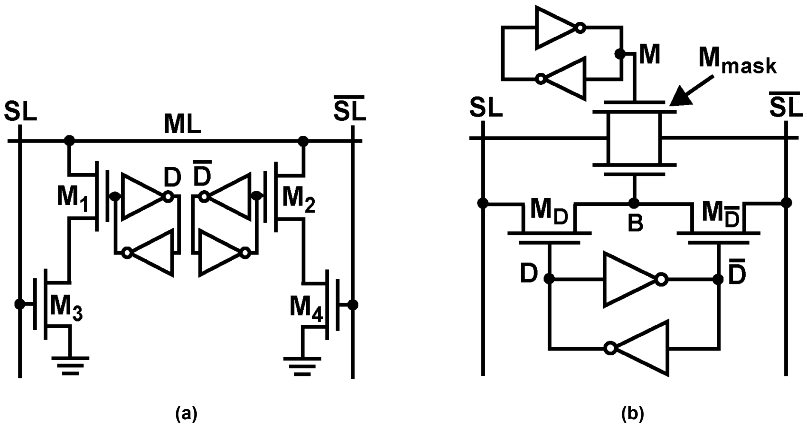


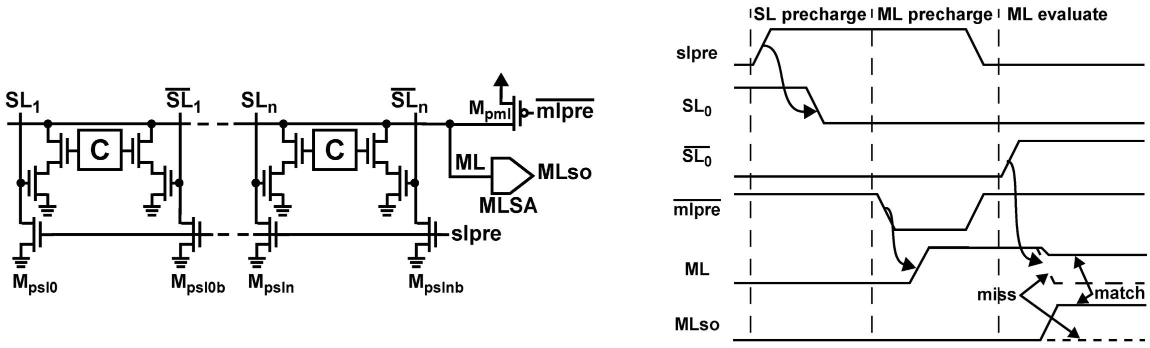
**Figure 4.3:** TCAM cells - (a) NOR-type, (b) NAND-type

| | **Stored Bits** | | **Search Values** | |
|---|---|---|---|---|
| **State Stored** | **D** | **M** | **SL** | **$\overline{\text{SL}}$** |
| Logic ”0” | 0 | 0 | 0 | 1 |
| Logic ”1” | 1 | 0 | 1 | 0 |
| Don’t Care ”X” | 0 | 1 | 1 | 1 |
| Don’t Care ”X” | 1 | 1 | 1 | 1 |

**Table 4.2:** The NAND CAM values for storing and reading data.

## 4.1.2. Matching Line Sensing Schemes

This part is devoted to an important part of the whole CAM concept which is the sensing schemes used on the matching lines, which are responsible for registering the actual match or mismatch of the CAM wordline. The most important scheme is the conventional, precharge-high scheme; most of the other schemes are built upon this with power-saving modifications. The basic operation of this scheme (circuit shown in Figure 4.4a) can be described as follows: the ML are being precharged to high through the pMOS precharge transistors. Then, the NOR cells will pull-down the ML in case of a miss (the transistors will open, thus create a discharge path), or the ML will be left high in case of a match. Figure 4.4 shows the signal timings for precharge and read.



**(a)** Schematic with the precharge circuitry for the precharge-high scheme

**(b)** Timing diagram showing signal transitions for the precharge-high scheme

**Figure 4.4**

In order to perform an analysis on the ML sensing schemes, a simple yet effective model is needed. There are two different models for the two states: match and mismatch. For the match state a simple capacitor $C_{ML}$ is sufficient, while the model for the mismatch state of the ML should include the same capacitor, but also connected in parallel with a pull-down resistor $\frac{R_{ML}}{m}$, where $m$ is the number of mismatched (and thus discharging) cells on the ML. $C_{ML}$ comprises the ML wire capacitance, the NOR-cell diffusion capacitance of the transistors M1 and M2, the diffusion capacitance of the precharge transistors, and the input capacitance of the MLSA. For mismatches, the equivalent $R_{ML}$ resistance varies with the number of bits that are mismatched in the word; however, for the purpose of analysis we use the worst-case (largest) resistance, which occurs when there is a 1-bit mismatch (that is $m = 1$).

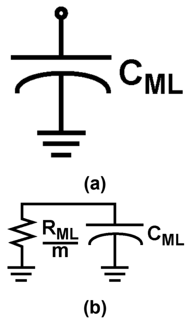


**Figure 4.5:** The equivalent models for match and mismatch for the precharge-high sensing scheme.

Based on the simple match line model we have described, it is possible to deduct the time required to precharge and evaluate the ML. The precharge time is defined as a 10%-to-90% rise time of the ML. From the capacitor charging equation we have:

$$V(t) = V_0 \cdot (1 - e^{-t/\tau}) \tag{4.1.1}$$

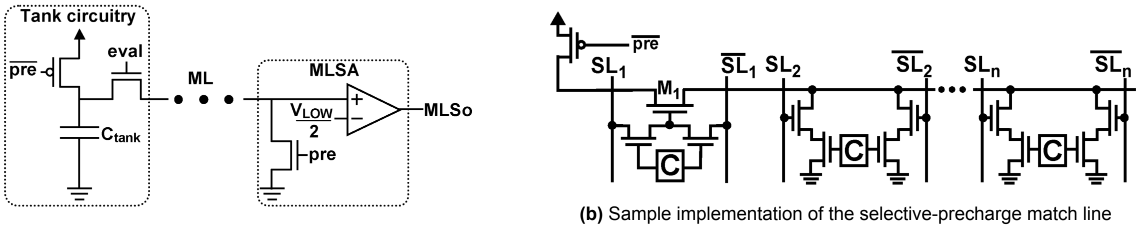
Hence, the equation giving the charge time is:

$$t_{ML_{pre}} = 2.2 R_{eq_{pre}} C_{ML} \tag{4.1.2}$$

The time needed for evaluation depends on the capacitance of the ML and it is defined as a drop to 50% of the ML precharge voltage:

$$t_{ML_{eval}} = 0.69 R_{ML} C_{ML} \tag{4.1.3}$$

The dynamic power consumed by a single mismatching ML is given by the equation 4.1.4 where $f$ is the frequency of search operations. In the case of a match, the power consumption associated with



**(a)** The low-swing sensing scheme proposed in [71]

**(b)** Sample implementation of the selective-precharge match line technique

**Figure 4.6:** Examples of the two sensing schemes discussed

a single ML depends on the previous state of the ML; however, since typically there is only a small number of matches, we can neglect this power consumption. Consequently, the overall ML power consumption of a CAM block with $w$ match lines is given by equation 4.1.5

$$P_{miss} = C_{ML}V_{DD}^2 \cdot f \tag{4.1.4}$$

$$P_{ML} = w \cdot P_{miss} = w \cdot C_{ML}V_{DD}^2 \cdot f \tag{4.1.5}$$

We have already mentioned that there are numerous other sensing schemes, most of them trying to address the high power issues of the one precharge-high. Here we will briefly address two more schemes which are interesting not only for their simplicity but also for their effectiveness.

*Low-swing schemes* are tackling the power consumption issue and potentially the sensing speed by reducing the ML voltage swing [71], [72]. This reduces the power proportionally to the reduction of the voltage swing which is governed by the following equation:

$$P_{ML} = w \cdot C_{ML}V_{DD}V_{ML_{swing}} \cdot f \tag{4.1.6}$$

A characteristic example of such a sensing scheme comes from [71] where the voltage swing is reduced by setting the $V_{LOW} = 300mV$. The precharge to $V_{LOW}$ is taking place by connecting a *tank capacitor* to the ML as per 4.6a. The ML precharge voltage is given by:

$$V_{ML_{pre}} = V_{DD}\frac{C_{tank}}{C_{ML} + C_{tank}} \tag{4.1.7}$$

The *selective precharge* scheme is another sensing scheme used with the main goal to reduce power consumption. This scheme performs first a small match operation on a few bits before fully activate the ML [73]. In long, $m$-bit words, the scheme checks the first $n$-bits for match and then searches the remaining $m-n$ bits only for $n$-bit matches. Assuming a random, uniform data distribution, this matching scheme can exclude almost 87,5% of the searches from proceeding to the second stage, thus saving equal percentage of power. The two main disadvantages of this approach are the possible higher power per bit for the first searches [74] and a data distribution which does not adhere to the random, uniform pattern we previously assumed. Figure 4.6b shows the proposed structure. It is one of the most common power saving methods on match lines [68], [75–79] due to its simplicity and effectiveness in many applications. The first cell of the structure is a NAND-cell, which allows precharge of the rest only if there is a match with the data stored in it. If there is a mismatch, the precharge transistor is disconnected from the match line, resulting in reduced power consumption. The rest of the cells are NOR-type.

## 4.2. CAMs based on Emerging Memory Technologies

At this point the reader should have a clear view about what a Content-Addressable Memory is, what are its strengths and weaknesses and how these structures are conventionally designed, using CMOS logic. Due to their advantages, CAMs are an area of active research, especially within the emerging devices field. Several proposals have been made for smaller, faster and more efficient CAM cells, based on one or more emerging device technologies. Here, a few of them will be introduced, together with a discussion about their advantages and disadvantages, as well as the solutions that they might offer to specific problems.

Almost all of the recent efforts aiming to realize MCAMs are based on memory technologies such as FeFETs and ReRAM [80–83], while in a few cases, floating-gate FLASH cells are being proposed [84]. Some of these cells, compared to the CMOS cells we discussed before, offer dramatic improvements over area, while the improvements in power consumption and speed can be significant too. Since these technologies are not yet quite as established and optimised as conventional CMOS, the potential future advancements are even more promising. The so called nvCAM cells (non-volatile CAM), are leveraging the fact that a single device such as a FeFET or a memristor can store information in a non-volatile fashion. Despite the advantage of the non-volatility, the improvements on area are also significant, compared to the traditional SRAM cells used in conventional CAMs.

### 4.2.1. Resistive CAM cells

ReRAM-based CAM cells are mentioned in [80], [81] and [85]; in the first paper the researchers present a high-speed memristive Ternary CAM (TCAM) cell build with 9 transistors and 2 ReRAM memristors (9T-2R). The size of this cell is comparable to these mentioned in the previous paragraph, so there are no improvements with regards to area. However the is significantly faster even from state-of-the-art CMOS 16T CAM cells [86], with a search delay in the range of 200ps, compared with the delay of a few nanoseconds reported in [86]. Figure 4.7 shows the schematic of the cell.

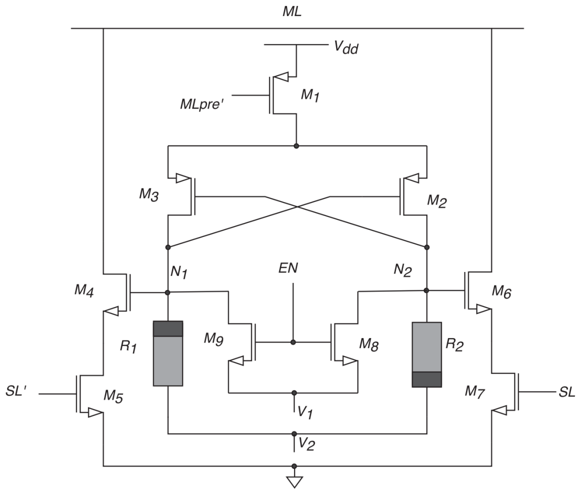


**Figure 4.7:** A 9T-2R memristive CAM cell

Another ReRAM-based cell worth discussing has been reported on [85]. The main advantage of this cell is its area; with only three CMOS transistors and one memristor is more than two times smaller that the aforementioned ReRAM-based cell. Its search energy is also considerably less: 2.24fJ/bit compared to 3.69fJ/bit for the previous ReRAM design, although this comes at the cost of significantly increased search delay: the authors report a search delay of 960ps, comparable to that of the conventional CMOS designs. The schematic of the CAM cell is shown in Figure 4.8.

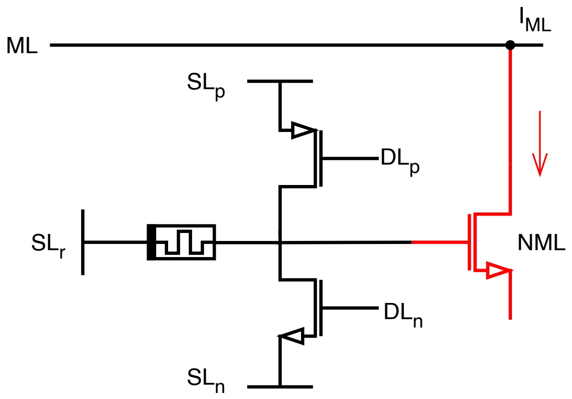


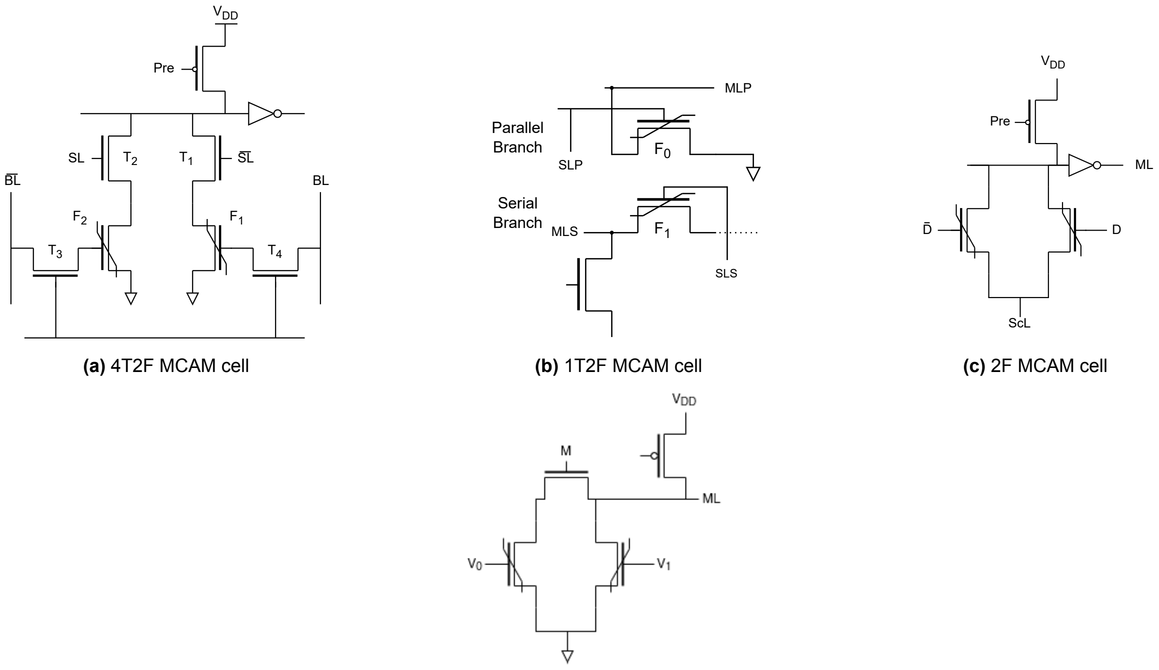
**Figure 4.8:** A 3T1R memristive CAM cell

### 4.2.2. FeFET-based CAM cells

When it comes to FeFETs there are three popular CAM designs: the 4T-2FeFET [62], the 2FeFET [62], [87] and the 2FeFET-1T [88]. The above designs promise a much better energy efficiency as compared to their memristor counterparts and marginally better than the 16T CMOS implementations [62], while they all require significantly less area (see Figure 4.9). In addition to these designs, a few others have been proposed; the TC-MEM design proposed in [89] extends the 2-FeFET cell so it can be accessed either by content or by address (Figure 4.9-d), while the authors of [90] introduced a complementary FeFET CAM design utilizing a pair of p- and n- FeFET devices and reporting significantly reduced search latency and failure probability. Another novel design was also proposed in [91], which uses two n-FeFET devices connected in serial topology; the authors reported promising results with regards to search energy, compared to other designs. A summary of the basic metrics of the most popular FeFET designs is provided in Table 4.3. This table also provides comparison with CMOS and memristive designs. Despite the depth of the literature review, it was impossible to find accurate power and delay information for each and every design, so the table includes only verified work which provided this data.

| **Reference** | [74] | [92] | [93] | [94] | [95] | [96] | [97] | [98] | [98] |
|---|---|---|---|---|---|---|---|---|---|
| **CAM Type** | TCAM | TCAM | TCAM | TCAM | TCAM | BCAM | BCAM | TCAM | TCAM |
| **Technology** | CMOS | ReRAM | ReRAM | FeFET | CMOS | CMOS | CMOS | FeFET | FeFET |
| **Transistors/cell** | 16T | 2T-2R | 3T-1R | 2F | 14T | 10T | 10T | 2F-1T | 4F-2T (Hybrid) |
| **Process node** | 45nm | 90nm | 90nm | 45nm | 45nm | 45nm | 45nm | 45nm | 45nm |
| **Sensing scheme** | Precharge | Precharge | Precharge | Precharge | Non-precharge | Non-precharge | Non-precharge | Precharge | Precharge |
| **Search delay [ns]** | 0.58 | 0.35 | 0.96 | 0.34 | 20 | 1.25 | N/A | 0.25 | 1.23 |
| **Search energy [fJ/bit/search]** | 0.59 | 0.55 | 0.51 | 0.35 | 0.18 | 2.1 | 0.66 | 0.195 | 0.0026 |

**Table 4.3:** Comparison summary of CAM cell designs



**(a)** 4T2F MCAM cell

**(b)** 1T2F MCAM cell

**(c)** 2F MCAM cell

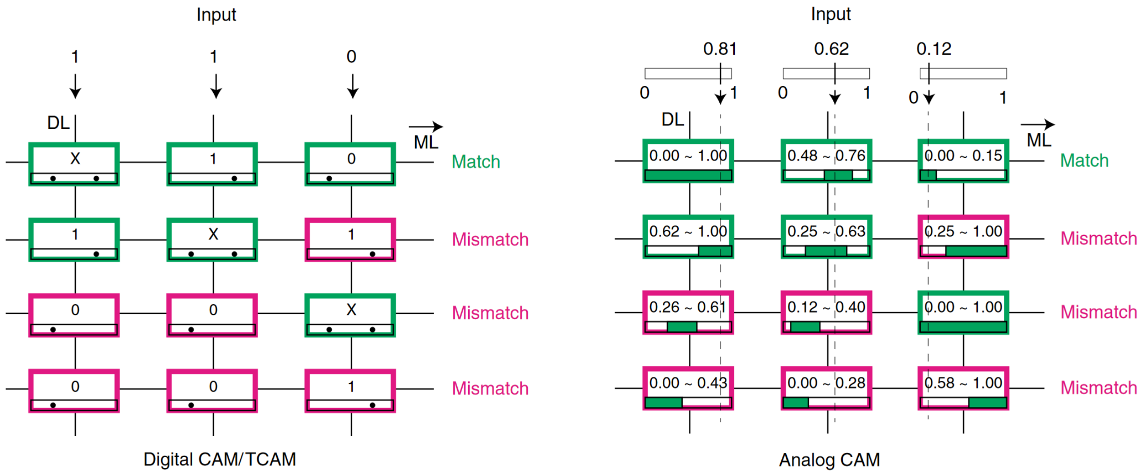
**(d)** TC-MEM CAM cell

**Figure 4.9:** Four different designs of FeFET-based CAM cells.

## 4.3. Analog and Multi-level CAMs

Two different types of CAMs with enormous potential in many different applications are analog CAMs and multi-level CAMs. Starting with the simplest concept, multilevel CAMs are CAMs that are capable of storing and searching multi-bit values on the same cell. While this is impossible for SRAM-based implementations, it is feasible for those based on emerging memories. Multi-level storage is a well-understood feature of certain memory technologies and it has been reported numerous times in the bibliography, both for emerging devices such as ReRAM [99, 100], FeFETs [83, 88, 91] and STT (Spin Torque Transfer devices) [101], as well as for conventional technologies like FLASH [84]. Works like [83] have reported FeFET-based CAMs with a storage capability of up to eight levels (3-bit cell). These cells can be used either for high density storage, or for certain types of operations, like nearest-neighbour search operations, due to the difference in the discharge time. This type of in-memory processing is used mostly in the fields of machine learning and neural networks.

Analog CAMs (ACAMs) have been used as TCAM alternatives with increased data density, reduced operational energy and area, for in-memory processing circuits [81]. An ACAM cell that stores multiple, narrow, non-overlapping ranges, can be viewed as a high-density digital CAM referred here as Multi-bit Content Addressable Memories (MCAM), with each range signifying a distinct state. The primary disparity between MCAMs and ACAMs lies in their search capabilities: every MCAM cell exclusively seeks within a restricted set of input values, each corresponding to a state, while ACAMs search across an infinite range of inputs. In MCAMs, the stored ranges maintain a one-to-one correlation with the inputs. Therefore, if there are four narrow, non-overlapping ranges and specific inputs, the ACAM would manifest as a 4-state or 2-bit MCAM. Consequently, MCAM can be viewed as a discretised and robust case of ACAM [83]. Figure 4.10 [102] shows the correlation between these two types. The majority of the FeFET cells we discussed previously are both TCAM/MCAM cells.



**(a)** Representation of a digital CAM. Each cell holds only one value and the input is being evaluated against that.

**(b)** Representation of an analog CAM. Each cell holds a value range and the input is being evaluated against that.

**Figure 4.10**

# 5

# The FCAM Row Decoder

In this chapter we will describe in full detail the FCAM Row decoder. We will make use of the topics discussed in the previous chapter and we will further extend them towards the direction of FeFET-based CAM cells, the peripheral circuitry used, and the challenges in designing such structures.

## 5.1. Concept and motivation

We have already discussed in Chapter 3 about address decoders and their importance on memory architectures. Depending on the application, different types of memories could use different types of decoders based on cost, area or delay considerations. Although there is no public knowledge about the types of decoders used in state-of-the-art memory products, it could be deducted that using a dynamic NAND decoder would be rational due to its speed and size (the pitch in thin-cell SRAM arrays offers little real estate for large decoding stages). Dynamic NAND decoders are customised for each circuit and each of their decoding lines is hardwired to one of the memory wordlines. Using a structure like this, despite its effectiveness, can also lead to serious challenges; testing it is difficult and requires complicated algorithms, while fixing any possible faults it is considered as virtually impossible [1].

The idea of an FCAM decoder arises from two observations: the first one being that FeFET-based CAM cells (FCAM) are significantly smaller than their counterparts, regardless of technology. It has been shown at least in two separate works that it is possible to have fully functional TCAM cells with only two FeFETs, connected either in series or in parallel. The second observation comes from the basic principle of CAM cells: a CAM cell outputs only the result of a search operation. The implication of this observation is immediate; a CAM cell fulfills the function of a decoder, since it receives a long, complex signal as input and produces a simple and short signal as an output. By leveraging these two facts, it is possible to store address data in a CAM array, which then can be queried in exactly the same way a regular address decoder does. If the address is found, the CAM array will set the appropriate match line which simultaneously will act as the driver for the word line of a memory array (like SRAM).

### 5.1.1. Single-cell Structure and Principles of Operation

After taking into consideration the results from the literature study, it was decided to use a 2-FeFET cell for implementing the design. The main reasons were the size and the robustness of the cell, as well as the extend of the research that has been done around it. The basic FCAM cell is shown in 5.2a. It consists of two n-FeFET transistors connected in parallel. Their drain terminals are connected to the match line, while their source terminals to a common line which can be connected either to the ground when reading from the cell, or to a voltage when writing. The ScL plays significant role in the *inhibition bias scheme* which will be discussed later.

There are two more components in the depicted cell, a pMOS precharge transistor and an inverter used as a sense amplifier. These two are not components of a single cell, but rather of the entire row of the CAM array. As we discussed also in Chapter 4, there are several sensing schemes; here we are using

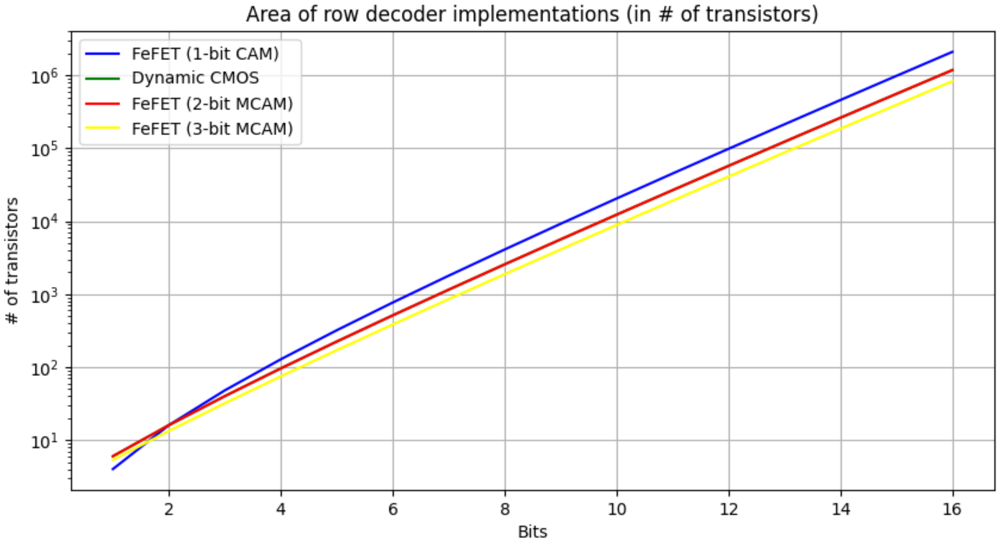
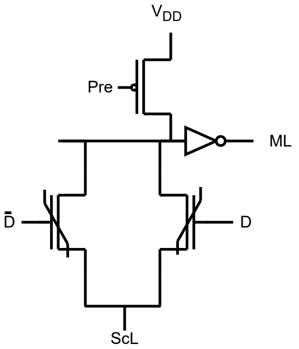


**Figure 5.1:** Area consumed by different implementations of decoders, compared to the decoded address width.

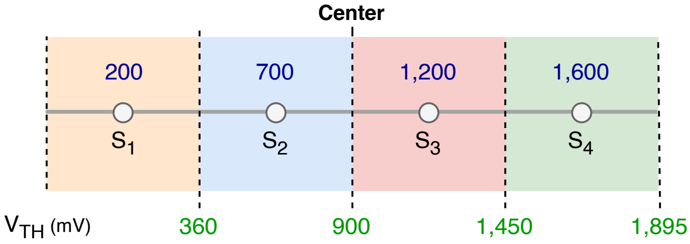
a ”traditional” precharge scheme with a series of two inverters acting as a sense amplifier and driving the wordlines of the main memory.

A great challenge in building this FCAM decoder was to gain the advantage over the conventional dynamic NAND decoders in metrics like area, delay and power consumption. Especially when it comes to area and delay, it can be quite difficult to compete with a well-established design such this. The plot 5.1 shows the size of several decoder implementations, number of transistors, against the width of the address. In the diagram the single-level FCAM implementation consumes the most area per decoded bit, which makes it a weak competition against dynamic NAND implementations. In comparison, a 2-bit FCAM cell is on par with the dynamic NAND implementation, while a 3-bit FCAM can have almost a 30% better area utilization. Thus, it was decided that the FCAM decoder that would be designed and simulated would be based on a 2-bit (4-level) cell.

To be able to work with a multi-level cell the levels need first to be properly defined. As we discussed in Chapter 2, FeFETs are leveraging the polarization of the gate’s ferroelectric layer to manipulate the device’s threshold voltage. Based on the different threshold voltages, different states can be distinguished. To define these states we need to define four different threshold voltages that will act as boundaries for these states. Figure 5.2b depicts a map of the states that can be stored in the cell, together with the threshold voltages that define them. Each state is defined by two threshold voltages, that separate it from the others, as well as one read voltage (in its centre) which is the voltage that we need to apply to the inputs, to query the cell for it.



**(a)** The FCAM cell used in the array and decoder



**(b)** The four levels of the proposed MCAM cell. The white dots are read voltages ($V_R$) while the dotted lines represent the threshold voltages.

**Figure 5.2**

The programming scheme for the cell is quite straight-forward: the right FeFET transistor is programmed to the defined threshold voltage on the right side of the state to be stored. Then, the left FeFET transistor is programmed to the analog-inverse of the threshold voltage on the left side of the state. The analog-inverse of a signal is defined as the one that has the same distance from the centre line as the original, hence, they are symmetrical. For instance, in the state map shown in figure 5.2b, the analog inverse of the 1.450mV threshold is 360mV, while the analog inverse of the $S_4$ reading voltage (1.600mV) is the reading voltage for $S_1$ (200mV). Reading (or querying) the cell follows the same principle; on the right side ($D$ terminal) we apply the read voltage for the state we want to query for, while on the left side ($\overline{D}$ terminal) we apply the analog inverse of that voltage.

The simplest way to define the boundary $V_{th}$ is to start with an erased FeFET device, apply pulses on its gate and fine tune them until the programming pulse give the desired $V_{th}$ value. However for reasons that will be explained in chapter 6 the results were not the ones we would expect and it created serious issues during the functional simulations. Eventually, the states were defined by enforcing different $V_{th}$ values and evaluating the I-V characteristics of the device for these values (Figure 5.3).

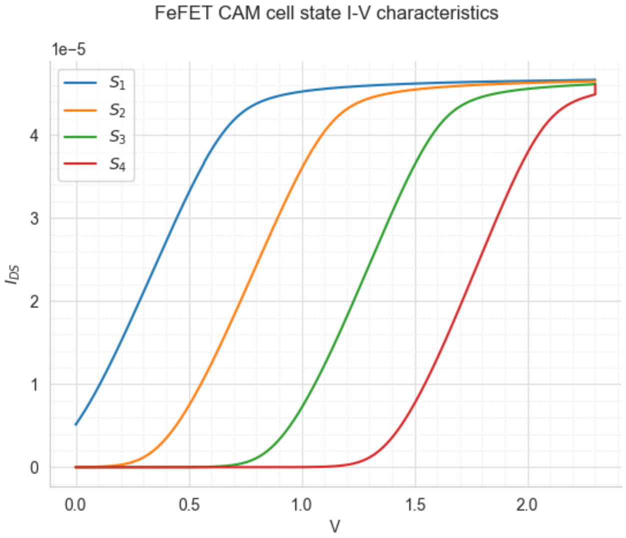


**Figure 5.3:** The I-V characteristics of the FeFET for the four $Vth$ used to define the four CAM states.

### 5.1.2. Scaling to an array

As we have already mentioned, we chose to base our design on the ultra-compact 2FeFET MCAM cell. This compact design however, comes at the cost of increased complexity for the writing and reading scheme, as well as parasitics from neighbouring cells. In order to deal with these challenges, we implemented a programming scheme similar to those proposed in [83], [94] and [103]. The programming scheme is based on analog-inverse voltages applied on each of the FeFETs. The map of the different states is shown in Figure 5.2b.

A simple $2 \times 3$ FCAM array is depicted in Figure 5.4 showing the array organization. Bitlines are shared among cells in the same column, while match lines and source lines (ML and ScL) are shared by all cells in the same row. All match lines are precharged and retain their charges in case of a match. However, if a mismatch occurs, at least one of the FeFETs in one of the CAM cells from that row will open, leading to a discharge of the corresponding ML resulting in a '0' output.

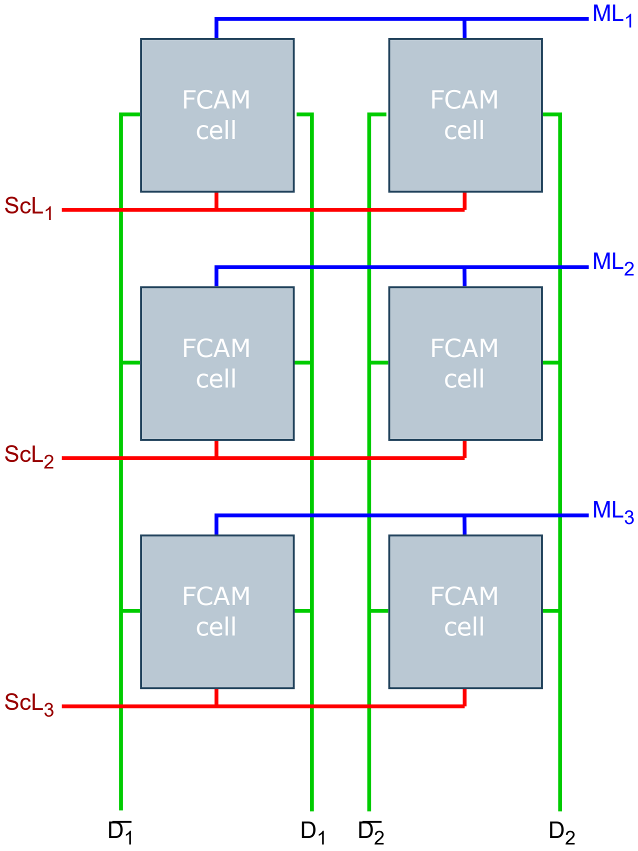


**Figure 5.4:** The high-level architecture of the $2 \times 3$ FCAM array.

### 5.1.3. Program/Read Scheme

The program scheme remains the same as the one described a few paragraphs before, regarding the single cell. However, despite this scheme working well for a single cell, applying it to an array introduces certain difficulties. On each column, the gates of the FeFETs are connected to the same $D$ and $\bar{D}$ bitlines. This creates problems during subsequent program (or erase) of a cell, since the write voltage will be applied to every other cell on the same column. To mitigate this, it is necessary to implement an inhibition bias scheme with which a $V_W/2$ voltage is applied on the ScL lines of the cells not being programmed. The same inhibition bias scheme has been used in the aforementioned works and is derived by the results reported at [103].

# 6 Simulation and Results

In this chapter we will discuss the experimental methodology and simulations and present the results in detail. The results will be discussed and eventually we will be able to answer the research questions we defined in the introduction.

## 6.1. Experimental Methodology

For such a complex system an iterative, incremental procedure was the best course to follow. For this reason we started by implementing a single FCAM cell, which then was progressively scaled up to an array of cells. The size of the array might be considered small, but it is sufficient to assess certain characteristics of the design such as its feasibility and testability, as well as metrics like speed and area. For these reasons we will provide data for two different cases. The first case will be an assessment at a cell level, while the second case will take place on the array level.

In all cases the methodology was straight-forward and followed the following steps:

- Complete an initial design
- Test functionality of the design by using a predetermined testbench
- Introduce faults
- Gather and evaluate data
- Compare experimental data and suggest mitigation techniques

## 6.2. Experimental Platform and FeFET Model

The simulations were conducted with Cadence Spectre, using the FeFET model described in [34]. The model is based on the Preisach approach and it is a compact model which can be readily used in circuit simulations. We have been provided two versions of the model, one based on the PTM 14nm FinFET process and the other based on the PTM 45nm planar process. We used the 14nm process for both our FCAM cells and the sensing circuitry. The precharge pMOS was designed with the 45nm TSMC process and was used throughout the entire sequence of experiments. Admittedly, this was a misstep, since it disturbs the homogeneity of the design; however, after careful consideration, we decided that it shouldn’t affect the reliability of our results and the effects from using a larger precharge transistor would likely be negligible.

Another aspect of the design is the dimensions of the transistors (both FeFETs and conventional CMOS).The FeFET dimensions were set to $75 \times 30$ nm, for the $W \times L$. The width reported here is the effective width $W_{eff}$ which is calculated based on the dimensions of the Fin according to the equation:

$$W_{eff} = 2 \times Fin_{Height} + Fin_{Width} \tag{6.2.1}$$

The sensing circuitry consists of two inverters connected in series, with each of their transistors sized $80 \times 30$ nm. The supply voltage used in the circuit was 0.8V.

In the set of simulations that were run on the array design, it was attempted to model the line capacitance by calculating the capacitance of a single cell using the Elmore delay model [104], and then adding the wire capacitance of the transmission line. The data for the last one was retrieved from the TSMC documentation (since no other resource was available) and was calculated at $0.38fF$, while the capacitance for each memory cell, when a mismatch has occurred, is $0.25fF$. Based on our calculations, the total capacitance of the ML should not be more than a couple fF. In practice, that was impossible to result in a working circuit. The reason is that due to the low capacitance, even the smallest leakage current was sufficient to discharge the ML and produce erroneous results.

With regards to the inputs, all FeFET programming pulses had a constant duration of 200ns, while the duration of the read pulses was 2.5ns. Finally, figure 6.1 depicts the circuit schematic of the cell, while figure 6.2, depicts the circuit schematic of the simulated array.

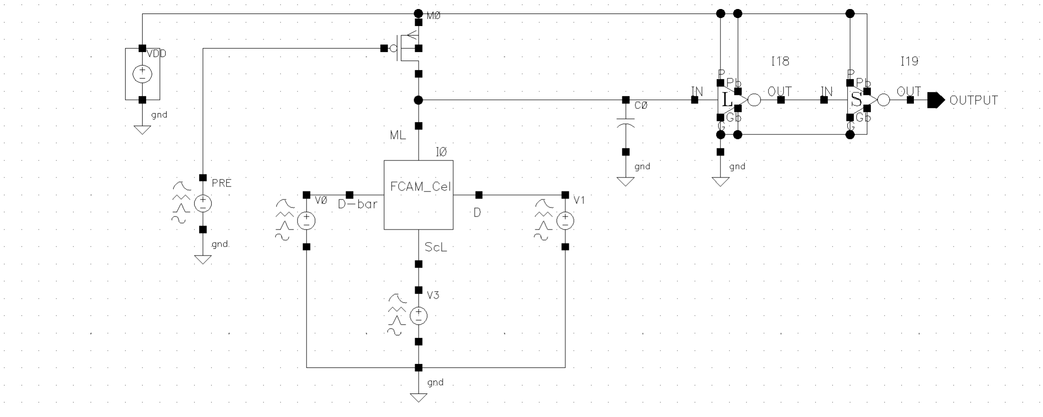


**Figure 6.1:** Schematic of a single FCAM cell with the sense amplifier (SA).

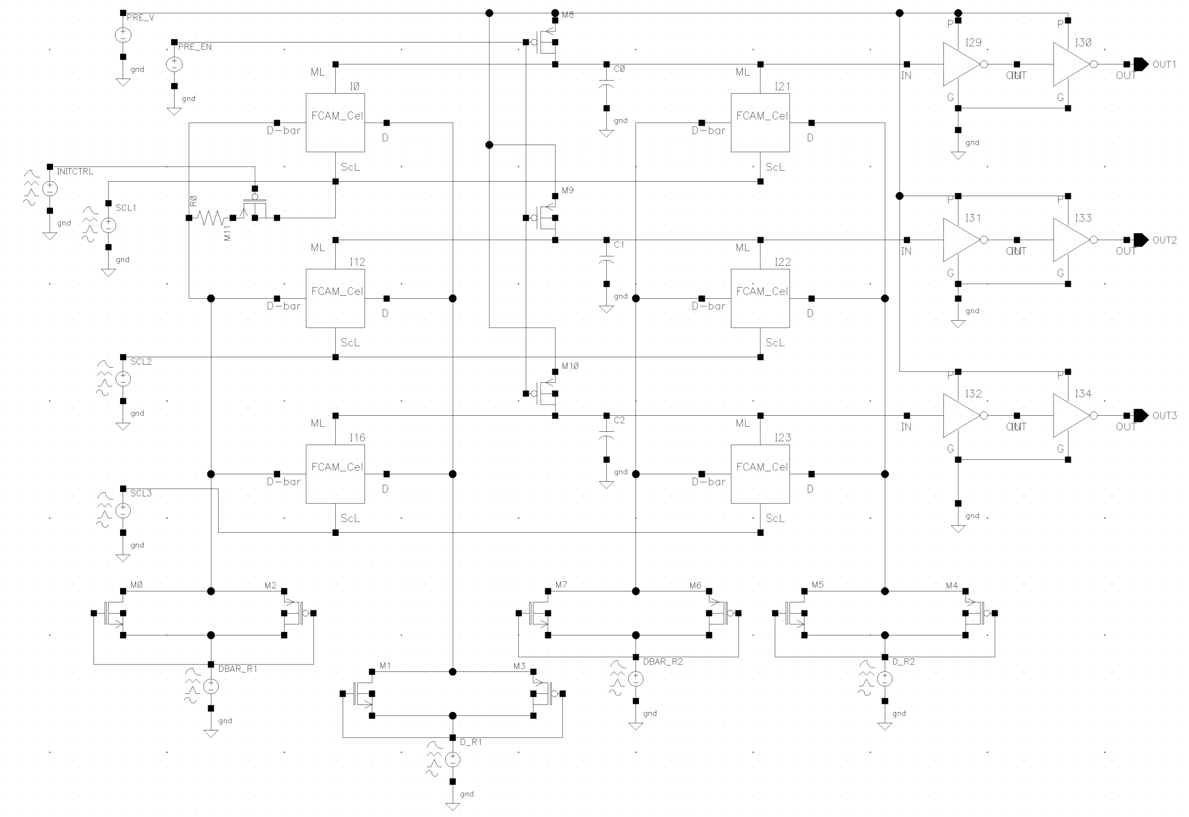


**Figure 6.2:** The circuit schematic of the $2 \times 3$ FeFET decoder. The 2-inverter SA could also be seen as well as one of the simulated resistive defects.

## 6.3. Simulating the Defects

One of the most important parts of the experimental process was simulating the defects. For this work it was decided that we will focus on resistive defects, thus leaving capacitive or other types of defects out of scope. For the resistive defects, we focused on two types: open defects and bridging defects. Both of these categories were first simulated on the single cell and then proceeded to simulate on the array. The defects were simulated one at a time, considering the effect of only one of them. Multiple defects with confounded effects were not simulated.

Figure 6.3 shows the schematic of the modelled defective FCAM cell. In total, six resistors emulated the six possible failure points of the cell. To accelerate the simulation/evaluation iterations and be able to run multiple simulations in parallel, we wrote a netlist generator which was able to produce the Spectre netlist code for the cell using custom parameters. Each one of these parameters would change a small part of the circuit, producing netlist files with slightly different values for resistors, or other components. We tested 20 different values for each one of the defect emulating resistors, with the values were generated in an exponential way starting from 1 $\Omega$ (practically no defect) up to 1 M$\Omega$. Eventually, 120 simulations were run for the open defects and 40 more simulations for the bridge defects.

We followed the same recipe for conducting the array simulations, testing 20 different values per defect for 7 partia-open defect sites. This resulted in a total of 140 simulations on partial-open defects, while we ran 80 more on 4 sites of bridge defects.

To assess the functionality of the cell and the effect the faults have to it, we devised a simple testbench for the simulation. The testbench consists of 4 cell writes, each one followed by 4 cell reads. Each one of the write operations stores one of the four states in the cell, which is then immediately followed by 4 queries. Consequently, there will be only one match which will result to the ML staying high, while the rest of the queries will cause a mismatch and thus, the ML will discharge. These operations are being done sequentially, so, first the cell is erased (resetted), using a -5V pulse. Then, the $S_1$ state is stored,

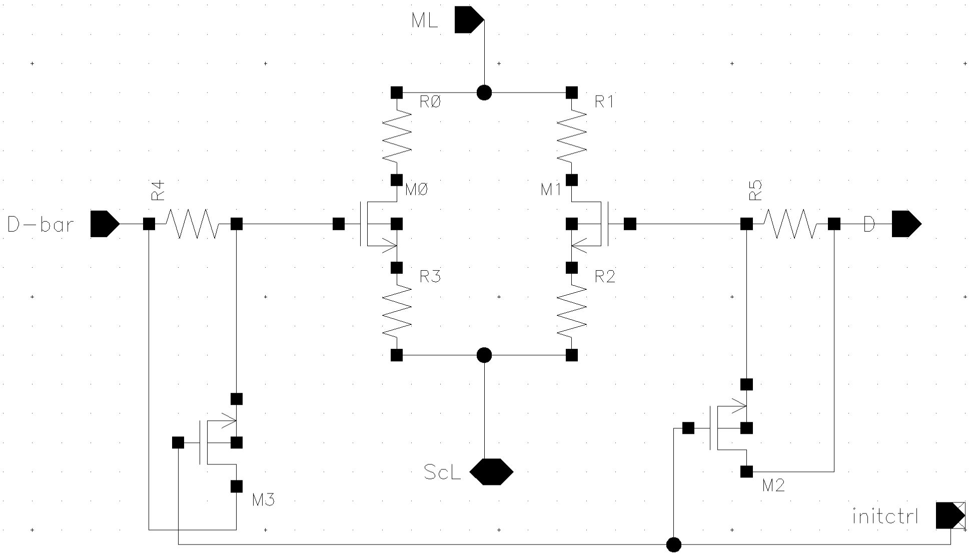


**Figure 6.3:** Schematic of a single, defective FCAM cell. Each resistor simulates one possible open defect.

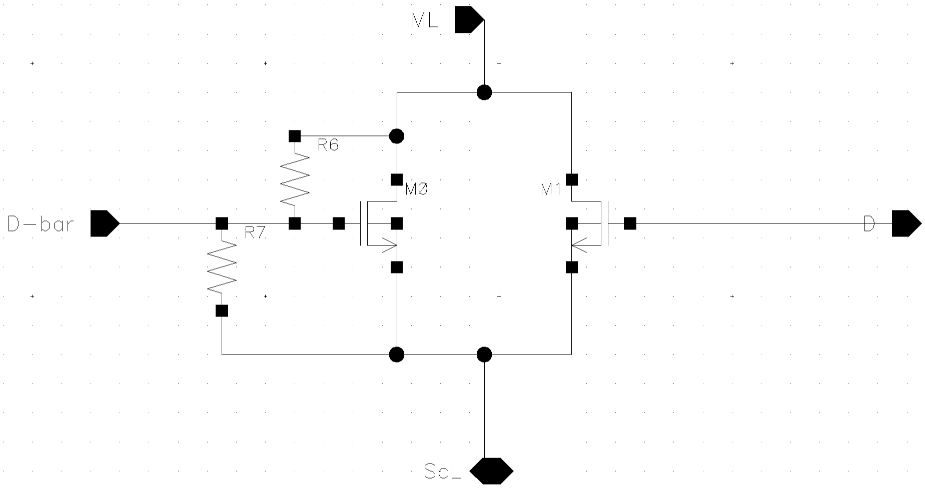


**Figure 6.4:** Schematic of another defective FCAM cell. Each resistor simulates one possible bridge defect.

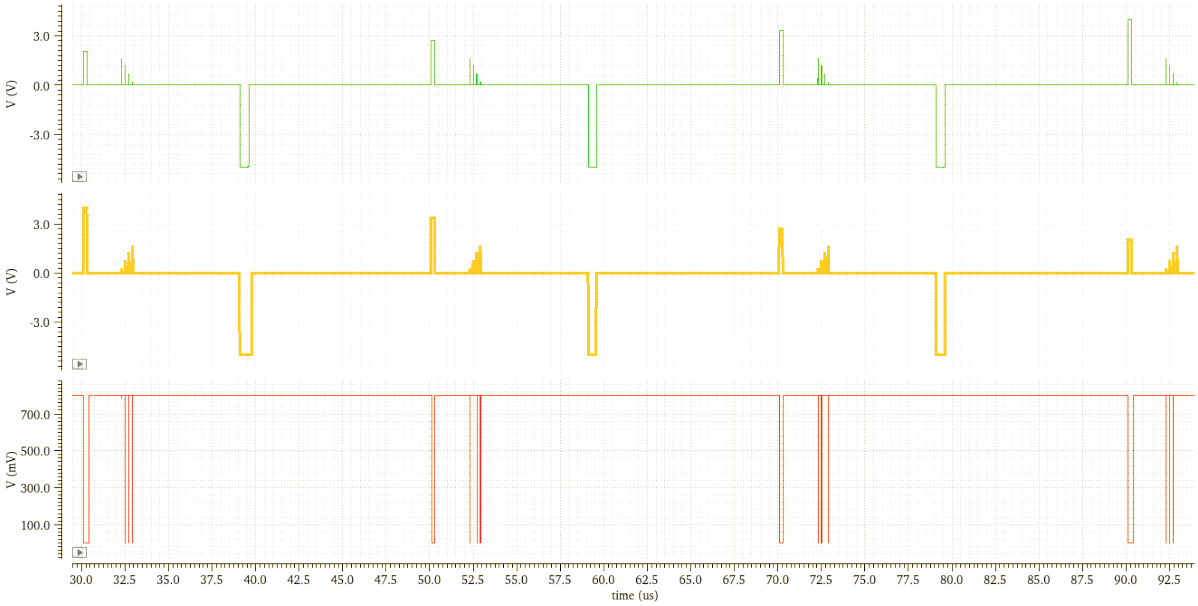


**Figure 6.5:** Pulse plot during simulation. The first two plots are the write/read pulses, while the last one is the output from the SA. The negative pulses are erase operations, followed by a relatively high positive pulse which programs the cell. The subsequent short pulses are the read operations. Notice the differential signals between the input terminals.

which is followed by 4 queries: $S_1$, $S_2$, $S_3$, $S_4$. This sequence of operations is repeated for the other 3 states: erase - write - query (read). This testbench gives us detailed information about the behaviour of the cell under every possible write/read scenario. In total 24 operations are taking place: 4 erases, 4 writes and 16 reads. Figure 6.5 shows these pulses during the course of one simulation round.

## 6.4. Presentation of the Results

This section will present the results of the simulations, which then will be discussed and evaluated. Starting with the single-cell simulations, figures 6.6, 6.7 and 6.8 show the results from the bridge defect simulations (please refer to Appendix B for the plots from the other defect sites). Each one of the dots represents one simulation, with a different value for the resistor emulating the defect. Each of these plots correspond to only one point of defect, in the resistors $R_0$ to $R_5$, respectively. Figure 6.9 and B.4 refer to bridge defects as shown in figure 6.4.

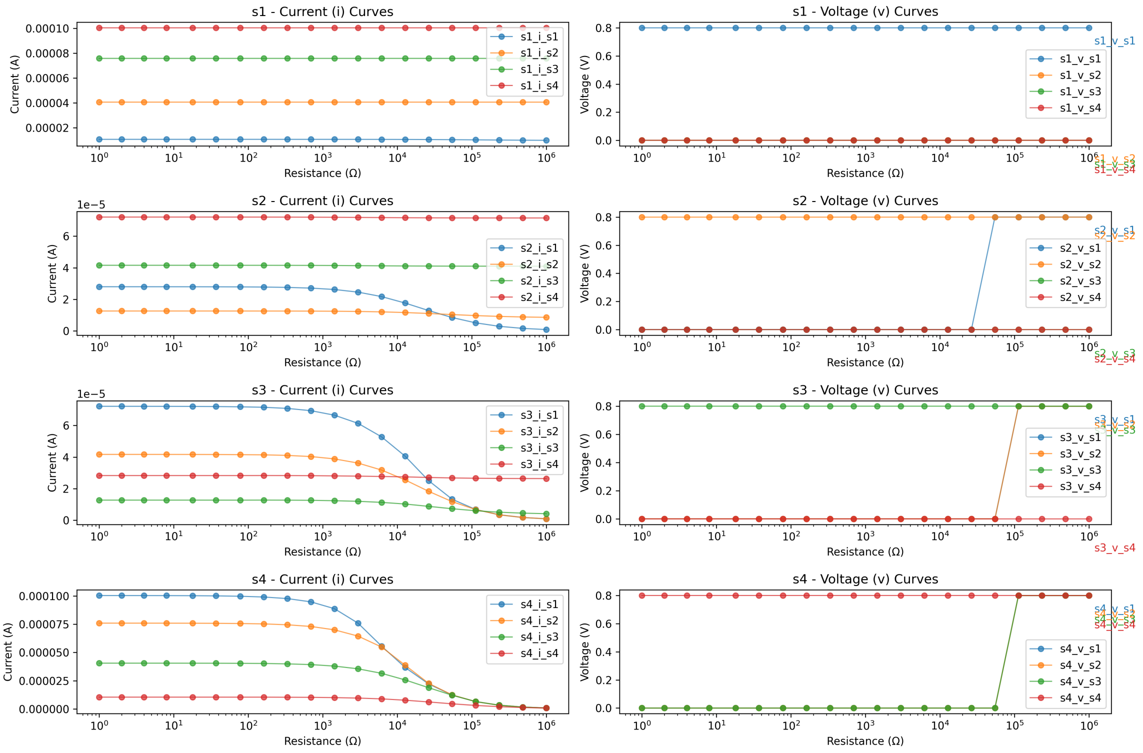


**Figure 6.6:** Voltage and current plots for the defect point on R0.

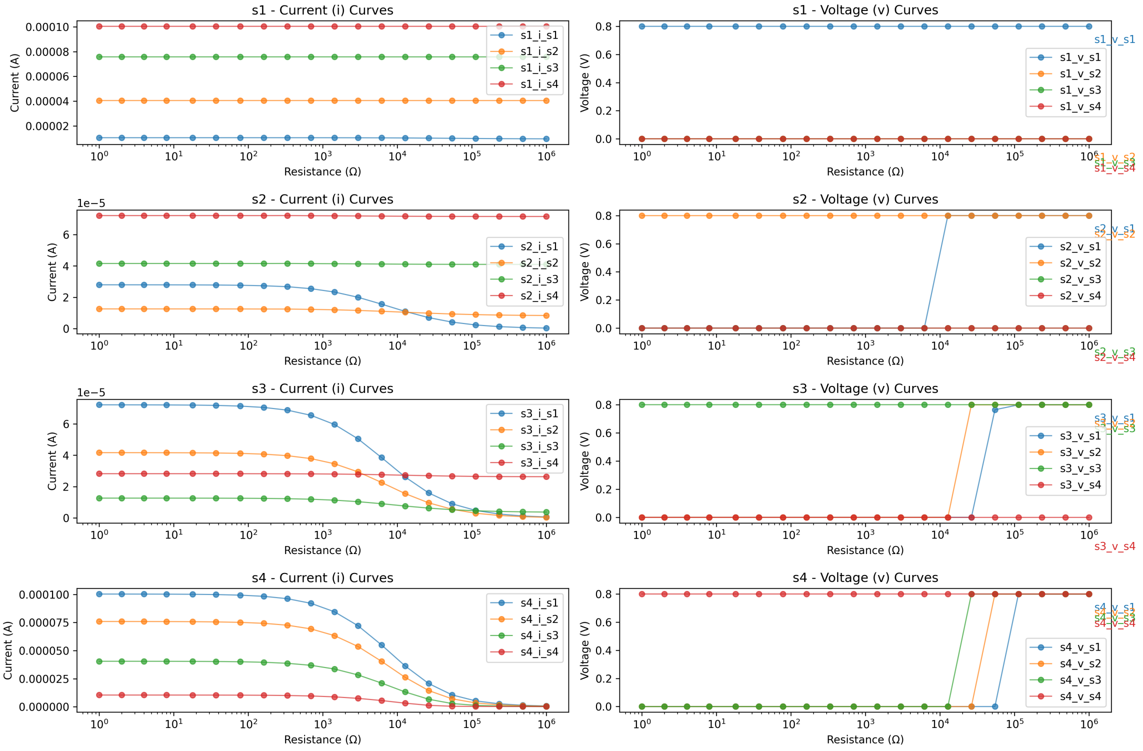


**Figure 6.7:** Voltage and current plots for the defect point on $R_3$.

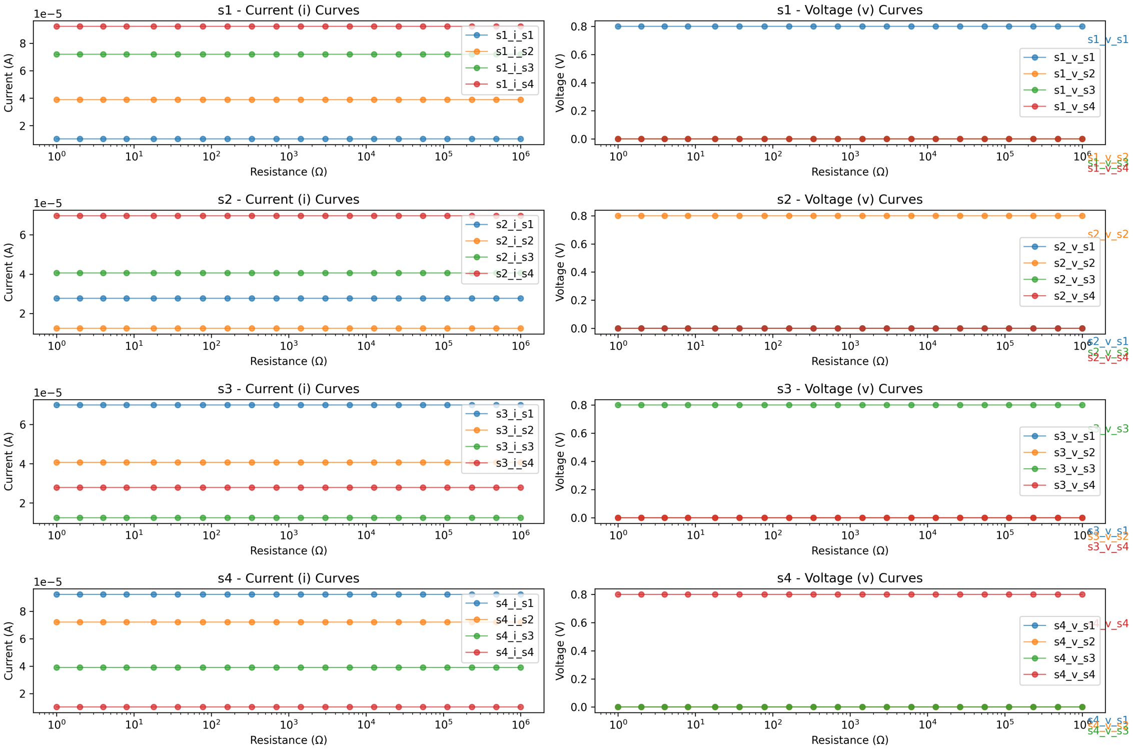


**Figure 6.8:** Voltage and current plots for the defect point on $R_5$.

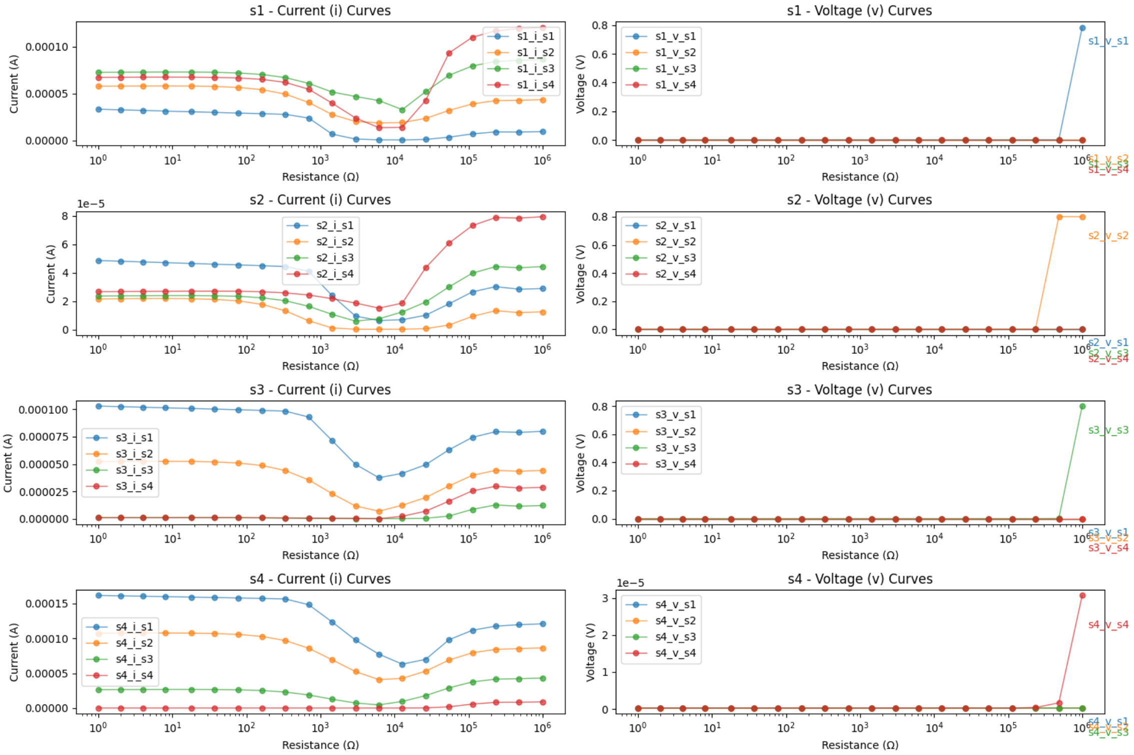
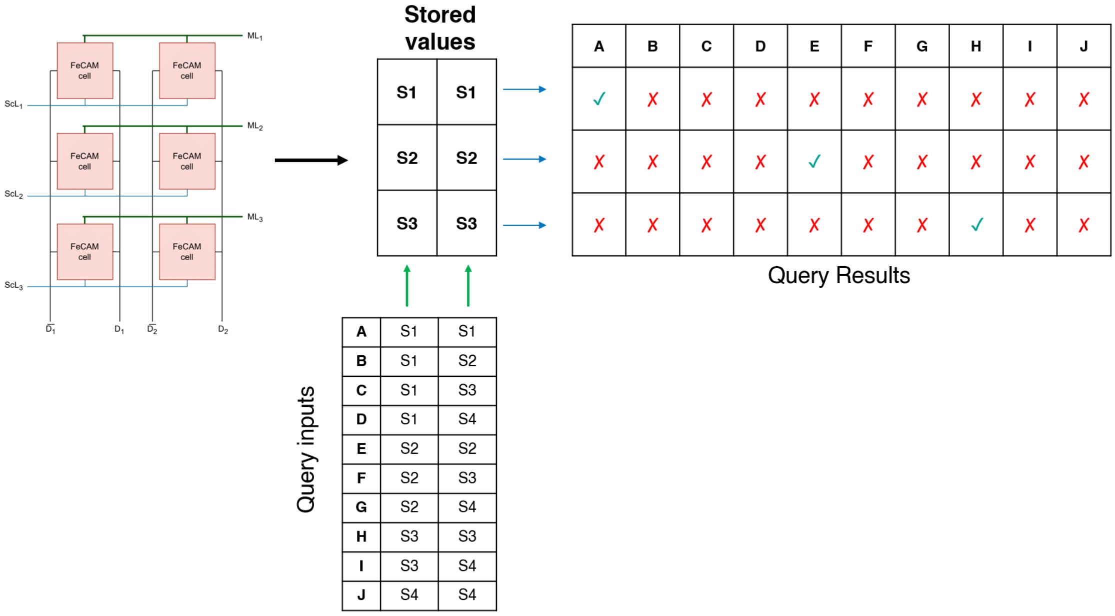


**Figure 6.9:** Voltage and current plots for the defect point on $R_6$.



Stored values

| | |
|---|---|
| S1 | S1 |
| S2 | S2 |
| S3 | S3 |

Query Results

| A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |

Query inputs

| | | |
|---|---|---|
| A | S1 | S1 |
| B | S1 | S2 |
| C | S1 | S3 |
| D | S1 | S4 |
| E | S2 | S2 |
| F | S2 | S3 |
| G | S2 | S4 |
| H | S3 | S3 |
| I | S3 | S4 |
| J | S4 | S4 |

**Figure 6.10:** Schematic overview of the testbench used in the array simulations.

Similar simulations were run on the array level. The testbench we used to evaluate the functionality had to be different due to the increased complexity of the simulated circuit. For this set of simulations, we structured the testbench as follows: a pair of similar states will be stored on each row, i.e $S_1 - S_1$ in the first row, $S_2 - S_2$ on the second, etc. Then, for each row, we request 10 read operations, going through almost all the combinations, avoiding however the symmetric ones, i.e. we will not read $S_1 - S_3$ and then again $S_3 - S_1$; because both cells store the same state, these two reads would yield the same results. Figure 6.10 shows the design of the testbench and figure 6.11 shows the results of a simulation run with this testbench. With regards to the simulation of defects, figures B.1 - B.4, show the results from these runs.

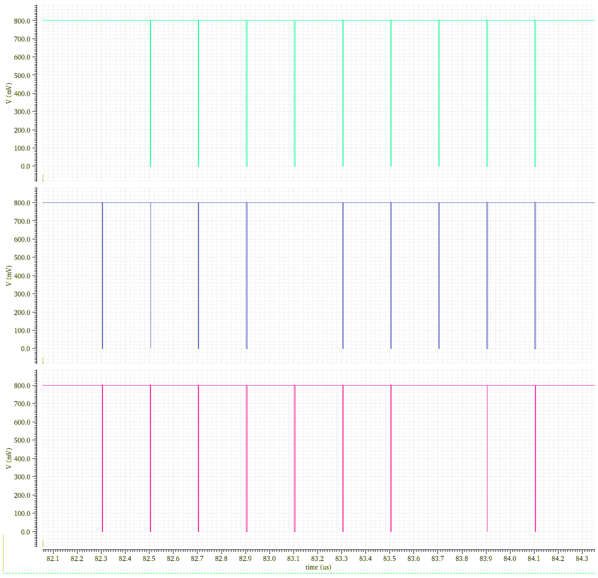


**Figure 6.11:** The simulation output for each of the three match lines of the array. The first row produces a match for the first query ($S_1 - S_1$), the second row gives a match for the fifth query ($S_2 - S_2$) and the third row for the eighth query ($S_3 - S_3$)

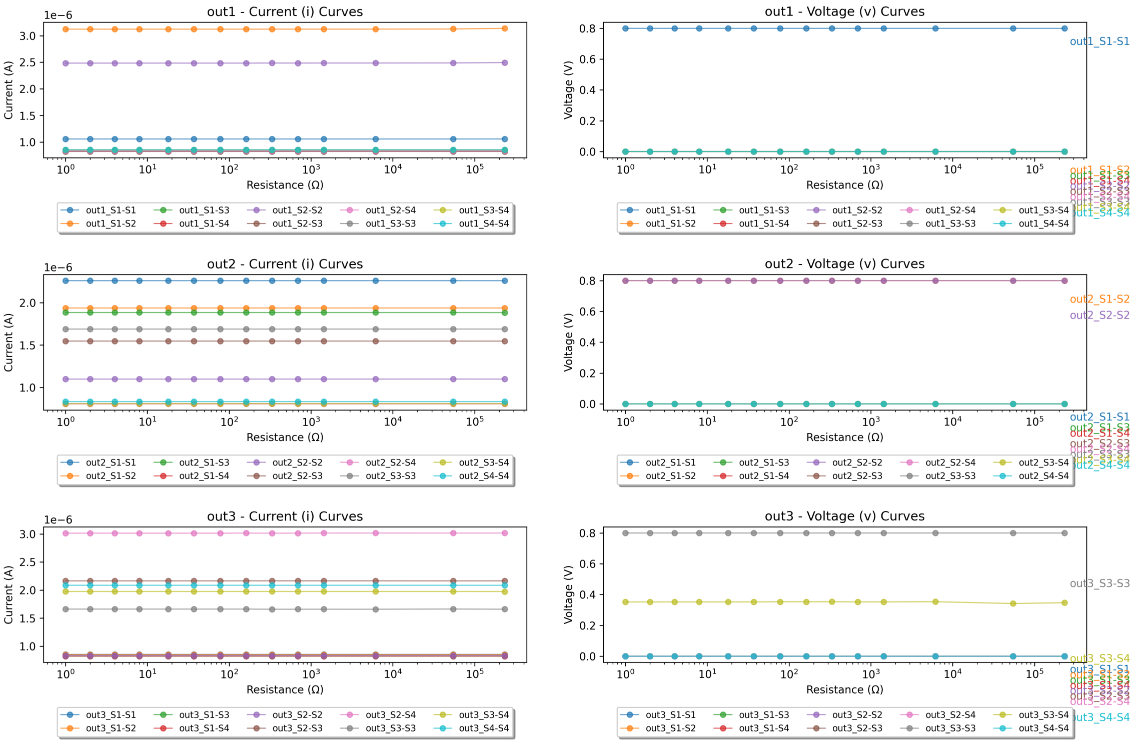


**Figure 6.12:** Voltage and current plots for the partial-open array defect point on $R_1$.

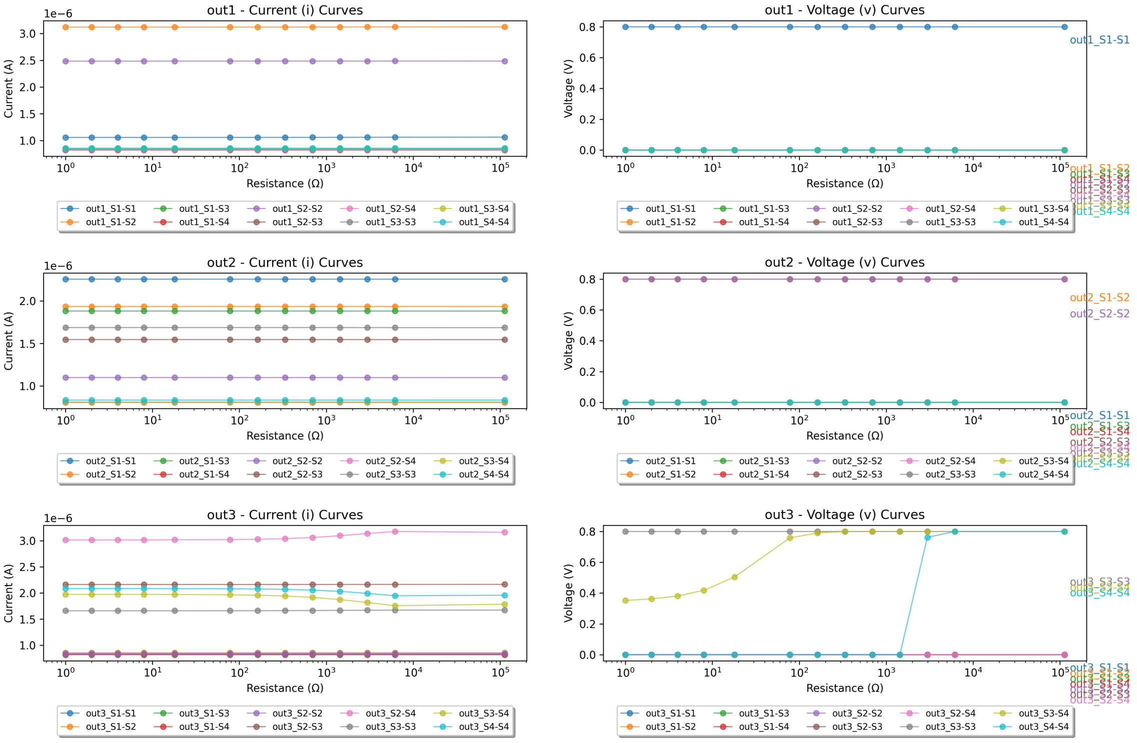


**Figure 6.13:** Voltage and current plots for the partial-open array defect point on $R_4$ (match line).

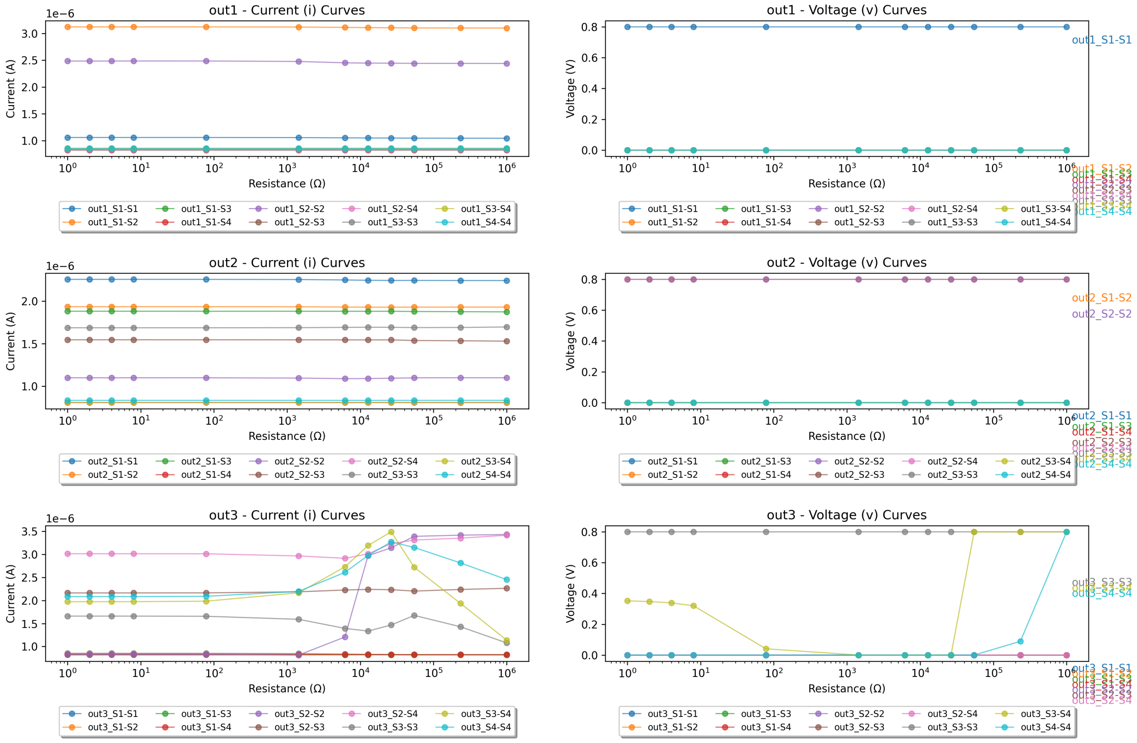


**Figure 6.14:** Voltage and current plots for the partial-open array defect point on $R_6$ (source line).

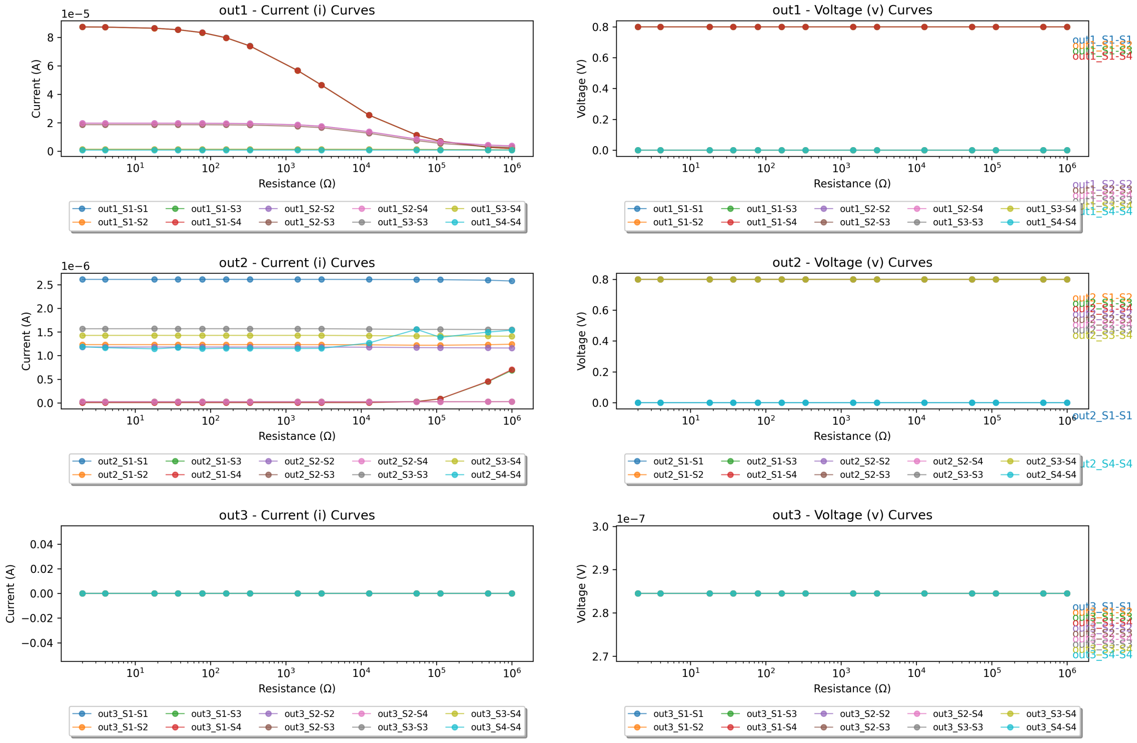


**Figure 6.15:** Voltage and current plots for the array bridge defect shown in 6.2 (between bitline - source line).

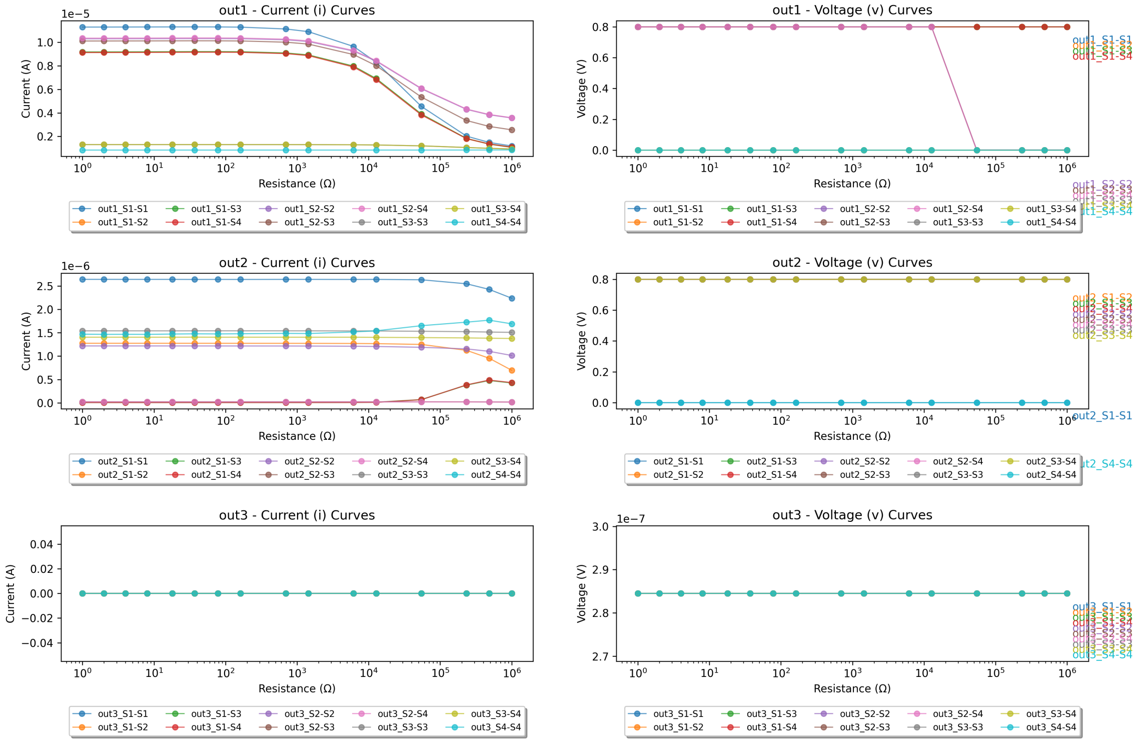


**Figure 6.16:** Voltage and current plots for the array bridge defect between bitline - match line

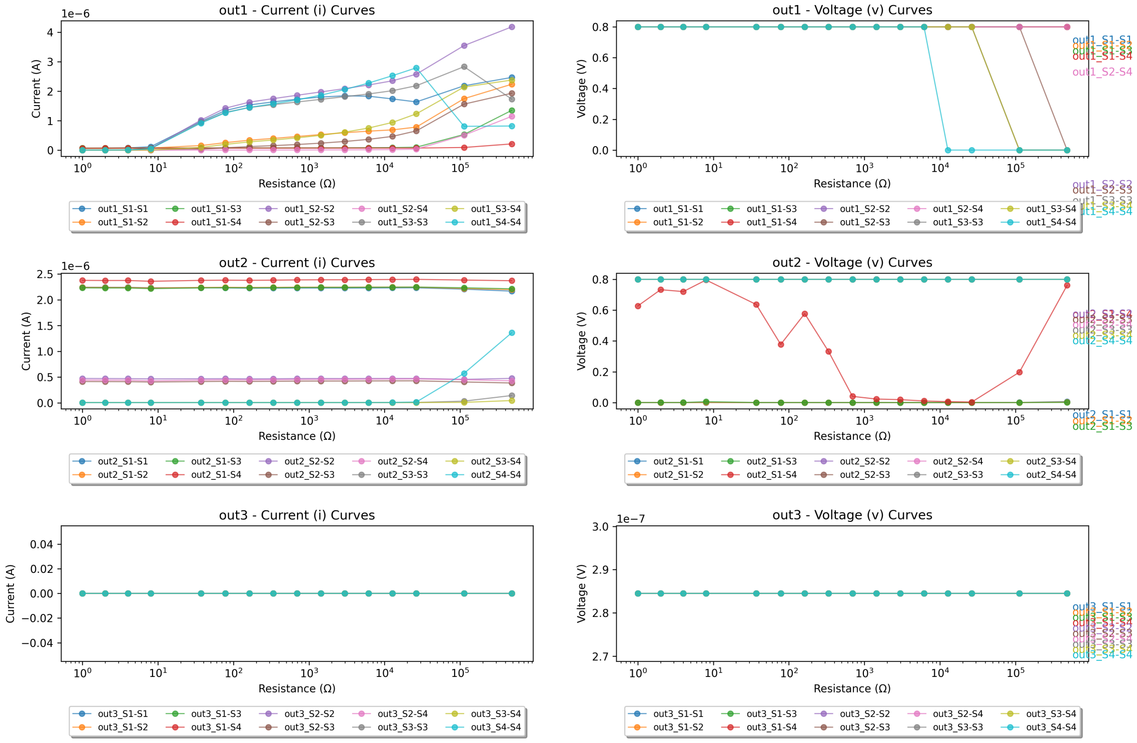


**Figure 6.17:** Voltage and current plots for the array bridge defect between bitline - bitline

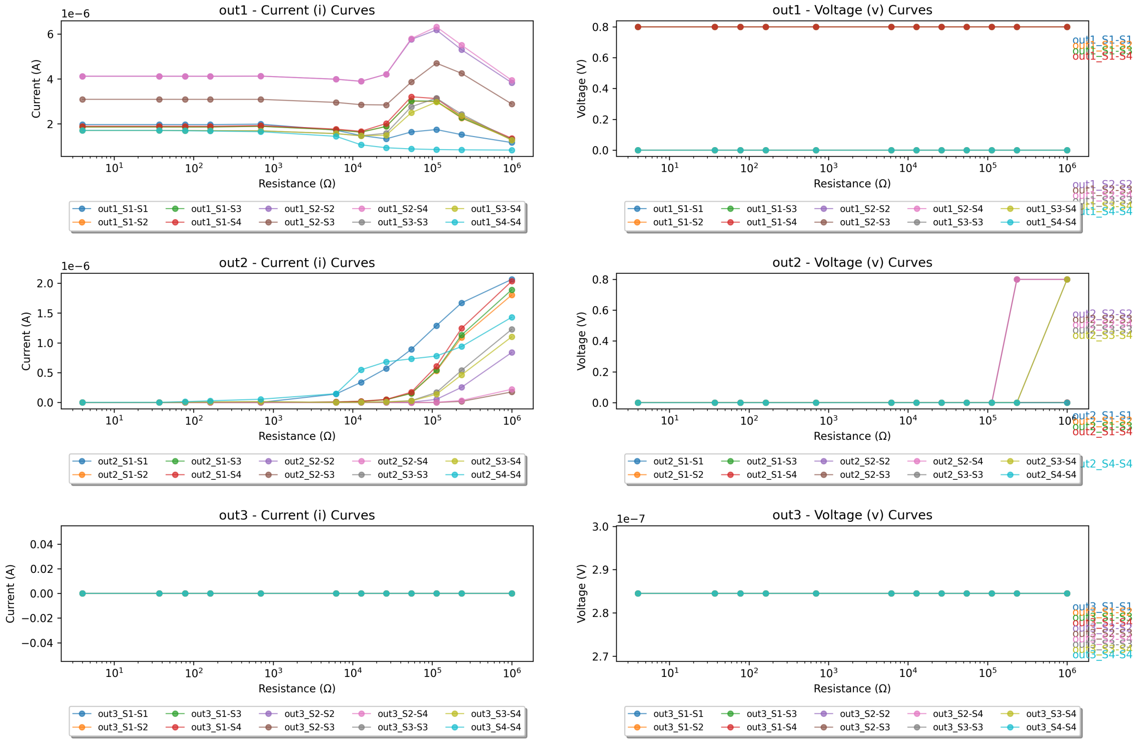


**Figure 6.18:** Voltage and current plots for the array bridge defect between bitline - source line

## 6.5. Discussion and Evaluation

In this section we will discuss the results presented in the plots, together with other metrics such as area and delay. Whenever possible, our design will be compared against the state of the art. We will be referring to the fault sites by using the resistor numbers (e.g. $R_1$, $R_4$, $R_6$, etc.).

### 6.5.1. Evaluating the single-cell simulation results

Figure 6.3 shows the defect sites for a single cell where each resistor represents a resistive defect. We call these defects partial-opens. For the single cell the results were clear from the beginning: as the defects were getting progressively more serious (i.e. larger resistor values), the impact was greater. That impact here was manifested in false-matches, where a query erroneously produces a match result, while it should result in a mismatch. Defects in sites R0 to $R_3$ (figures 6.6 to 6.7) are disrupting either the connection of the drain to the match line, or the connection of the source to the source line. Both of them can affect the way the queries are being interpreted since they disrupt the discharge path. The defects eventually result in service degradation when their value exceeds $8-10k\Omega$. Before that point, the FCAM cell shows significant fault tolerance. In many cases, like in those of the defect sites R0 and $R_1$, this value can be much higher in the range of $40-50k\Omega$.

With regards to the partial-open defects that affect the gates of the FeFETs (figures B.3 and 6.8), the results show that are not affecting the functionality of the cell in any possible way. Unless there is a complete open defect (which will prohibit any application of voltage on the gate) the cells continue to work without problems.

The last faults we simulated for the single cell were the bridge faults, as they are shown in figures 6.9 and B.4. These bridge faults are essentially short-circuits, either between the gate and drain (and consequently the match line), or between the gate and the source (thus between the gate and source line). The results here are also clear when looking to the voltage plots. There is an overall disruption for low resistance values, which leads all of the queries to return a mismatch as a result. As the resistance value increases, the effect of the short begins to fade and for resistor values above $500k\Omega$ the cell recovers, and returns to normal functionality.

### 6.5.2. Evaluating the array simulation results

The results from the array simulations, as anticipated, showcase a much more complex and diverse nature. Figure 6.19 shows the array together with all the resistive, partial-open defects. We decided to introduce resistive defects to only one row, and only one bitline. The reason for this is, that, due to the symmetry of the structure, any disruptions cause by a defect in one row or bitline, they could be generalised to the rest.

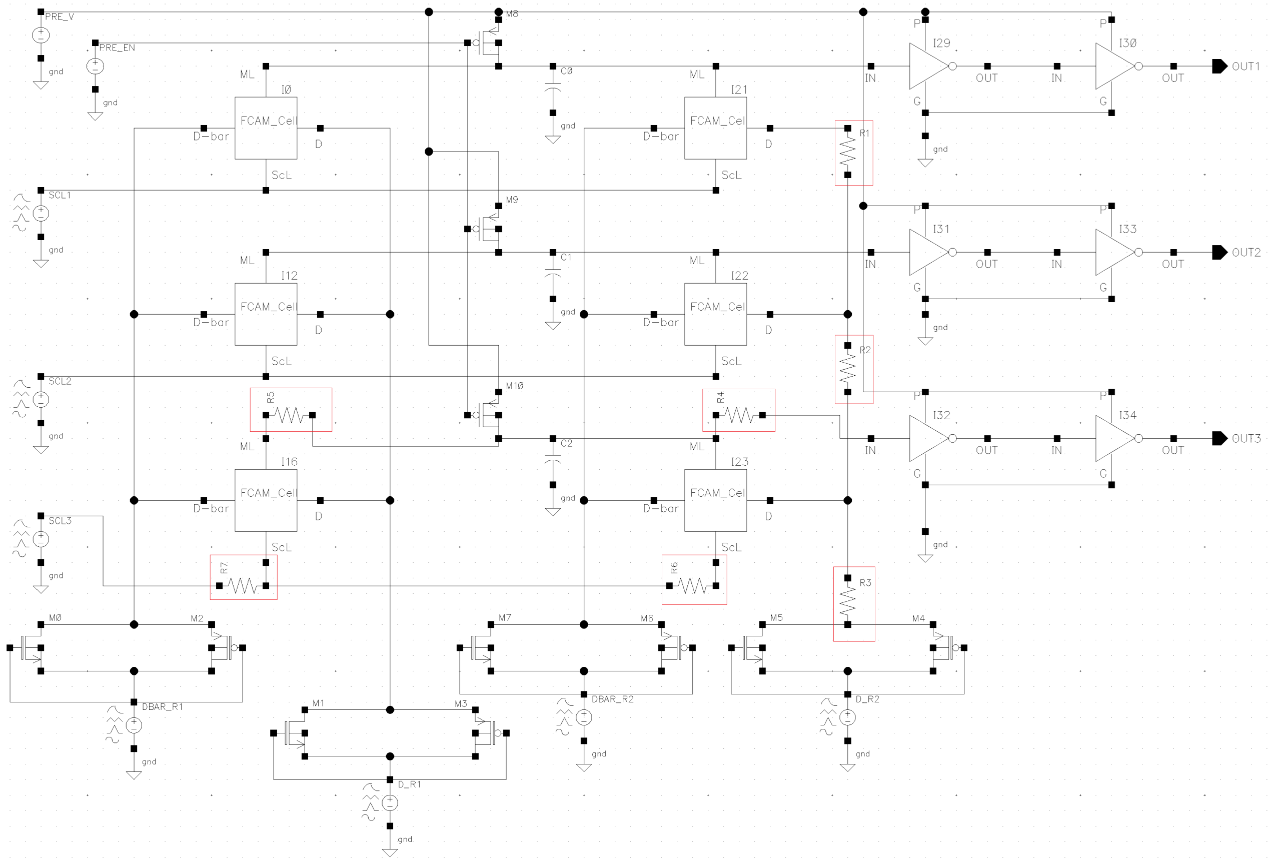


**Figure 6.19:** The array structure, showing the partial-open, resistive defects (marked with red squares)

**Bitline open faults**

The first three plot sets include the results from the simulations on three different defect sites: $R_1$, $R_2$, and $R_3$. The plots depict generally anticipated results, especially if we consider the similar results from the single-cell simulations; resistive defects on the lines driving the FeFET gates do not constitute a serious problem unless they are completely open.

Looking at the corresponding plots, there are a few peculiarities that must be discussed: the first row shows a clear distinction between the matched and mismatched states. That be, $S_1 - S_1$ state is a clear match (output of sense amplifiers is a logic-high voltage) and everything else is a mismatch (voltage dropped to zero). For the second row this is not exactly the same; although there is a match for the query $S_2 - S_2$, there is a second, erroneous match for the query $S_1 - S_2$. After careful review of the results, the simulation itself and the data processing method, it was clear that the cause of the problem was sampling the signal at a wrong time. Figure 6.20 depicts that. What can also be deducted from this plot though is, that, there is a deeper reason for this behaviour, since the discharge of the second ML is quite delayed and short in duration.

A theory that could explain this abnormality has two parts; the first one is that, due to the physical proximity between the stored states $S_2 - S_2$ and the query $S_1 - S_2$, ML cannot discharge fast enough, while the second one is that the ML capacitance might need to be lowered. As we mentioned in the beginning of the chapter, the ML capacitance should not be more than a few femto Farads. We simulated this capacitance by connecting a capacitor in parallel to the ML. The issue with this was, that, very small values for that capacitance were causing problems by discharging the ML even when the FeFETs were barely ON.

We attempted to mitigate this issue by reducing the capacitance value, as well as the read pulse duration, with no success. For reasons unknown yet to us, the simulations were failing for capacitances less than 35fF either by not producing a converged solution, or by running infinitely, stuck at extremely small step sizes (in the range of femto seconds). Eventually, we decided that we will accept the simulation data and take this fluke into account during the interpretation of the results. Taking into account the short discharge depicted in 6.20, we will consider that the $S_1 - S_2$ query produces a proper mismatch in the second row.

The same abnormality takes place in the third row, this time with the $S_3 - S_4$ state; the signal sampling takes place closer to the actual pulse and thus its value is closer to zero. Nevertheless, as the experiments proceed and their parameters change, the ML discharge might take place slighlty different and this is enough to throw out any attempt for accurate sampling. Again, since it was not possible to devise a solution, we can consider that the ML is properly discharging. A last remark about these and the following plots would be that some experiments failed completely to produce results and thus, they are missing from the plots. We believe that the culprit for this behaviour lies in the FeFET model it is being used. We will address this in further detail at a later section.

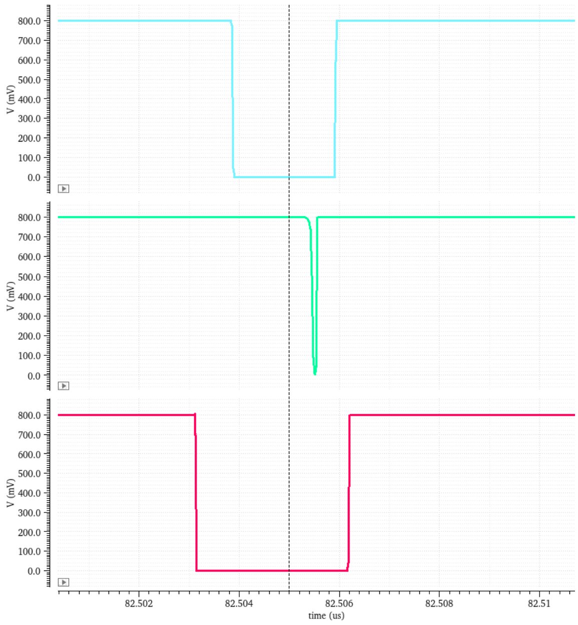


**Figure 6.20:** The plot shows the voltage drop during a mismatch. Rows 1 and 3 show what would be considered, normal, discharge pulse, following the duration of the input. The discharge pulse of the row 2 is much shorter and delayed, which created issues during the sampling of the signals. The marker (dashed vertical line), shows exactly where the sampling point is.

### 6.5.3. Match line open faults

We introduced two more resistive faults in the match line of one row of the array. The faults were put next to each one of the CAM cells, so we could observe difference of behaviour depending on the position of the fault. The results are showing some degradation of functionality affecting at least two queries ($S_3 - S_4$ and $S_4 - S_4$) when the defect is closer to the sense amplifier and severe loss of functionality when the defect is located on the $R_5$ site. In both cases, the erroneous functionality begins to manifest when the severity of the defect exceeds $1k\Omega$ while for one case it starts manifesting itself after only $100\Omega$. Despite the missing points for some experiments, it is easy to identify a pattern which help us interpolate the missing values (figures 6.13 and B.7).

### 6.5.4. Source line open faults

Figures 6.14 and B.8 show the results from simulating the last two partial-open, resistive defects on the array. The defect sites were put on the source line of the third row, next to each one of the FCAM cells, in a fashion similar to the match line faults (see figure 6.19). While the fault on the $R_6$ site cause some degradation in functionality by disrupting a couple of the queries, the second defect - $R_7$, disrupted 100% of the queries for values over $10k\Omega$. This is of course, expected, since the second defect prevents both cells from creating a current path and discharge the ML.

### 6.5.5. Array bridge defects

In total we tried four different sites for bridge defects. We concluded that simulating these would offer enough insight and a more general overview of how bridge defects affect the functionality of the array. Figure 6.21 shows the sites of these defects.

The results show that bridge faults between key lines of the array result in major functional disruptions of the array. Figures 6.15 and 6.16 show bridge faults between a bitline and source line / match line, respectively. Despite the faults occuring only on one cell of the first row, it's impact is significant throughout the array. Since these lines are shared across cells, the contents and behaviour of numerous cells can be disturbed by a defect in a seemingly unrelated site.

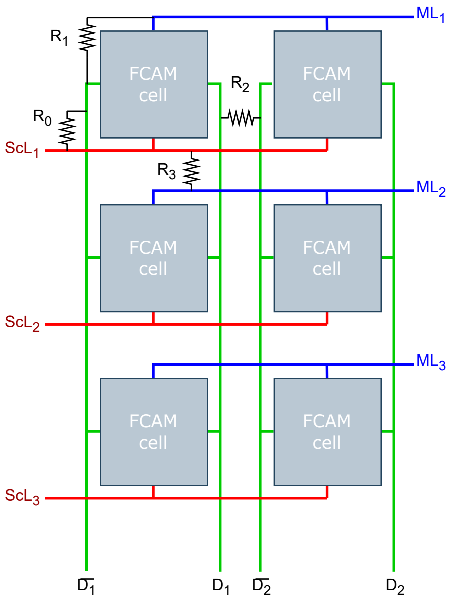


**Figure 6.21:** An overview of the sites of the bridge defects we simulated on the array.

The same situation arises in the case of bridge defects between neighbouring bitline and match line as well as between neighbouring bitlines. Figures 6.17 and 6.18 depict this reality. Despite the severity of these faults it is easier to detect due to their immediate manifestation and the fact that they hinder almost every attempt to read correctly from a cell.

### 6.5.6. Testing for defects

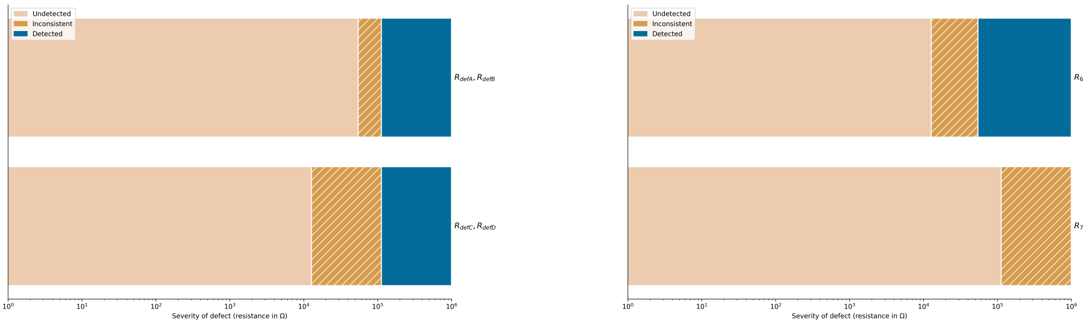
The results from the simulations provide enough information to understand the nature of some of the defects of the FCAM decoder. What is more important now though, is finding a way to test for these defects, so they could be detected and mitigated on time.

| Defect site | Tests |
|---|---|
| $R_{defA}$ | $\{(W_{S2}, R_{S1})\}$ |
| $R_{defB}$ | $\{(W_{S3}, R_{S4})\}$ |
| $R_{defC}$ | $\{(W_{S3}, R_{S4})\}$ |
| $R_{defD}$ | $\{(W_{S2}, R_{S1})\}$ |

**Table 6.1:** An example of a march test used to detect faults in FCAM single cells.

Detecting faulty FCAM cells is easy requiring a short march test, consisting of a single, two-operation, march element per defect. A fail in any of these tests can immediately flag the cell under test as defective and corrective actions, e.g., row repair when spare rows are available, can be taken. Table B.3 lists the tests for every defect considered in our study. The first operation writes a certain state to the cell, e.g., $W_{S3}$, and the second operation queries another state, e.g., $R_{S1}$. The reason for reading back a different state than the one written, is that partial open faults are affecting the cell's ability to properly discharge the matching line (ML) in case of a mismatch, thus leading to "false-match" errors. By querying a state different from the one written, we ensure that a match indicates the cell is definitely defective.

Figure 6.22a shows the ranges within which the FCAM cell operates correctly (the light areas on the left), and thus a partially open defect will not disturb the correct cell operation. Moreover, it also shows the areas where a defect will cause all of the reading operations to fail (the dark areas on the right). The dashed area in the middle shows the range where a defect may or may not cause incorrect operation depending on the specific read operation used. Obviously we have to look for the 'most sensitive' read operation that will ensure detection of defects in the two concatenated faulty areas on the right side. Please note the FCAM design tolerance to partially open defects. Values from zero up to few kilo Ohms will not have any impact on its correct operation.



**(a)**

**(b)**

**Figure 6.22:** Detectability ranges for cell and array faults

Array faults can be detected in a similar manner. Figure 6.22b shows the fault detectability range for the resistive faults of the array in the sites $R_6$ and $R_7$. Despite the simplicity of the testing procedure, careful consideration is needed when choosing which query cases will test against. For instance, the same test cases that will reliably detect a fault in the $R_7$ fault case, will not produce reliable - if any - results, in detecting resistive faults like on in the $R_6$ site.

### 6.5.7. Comparison to conventional decoder designs

Evaluating the decoder delay included determining the total capacitance of the decoder row. As it was mentioned before we set this value to 35 fF. Under these above conditions, our simulations resulted in a propagation delay of approximately 870 ps.

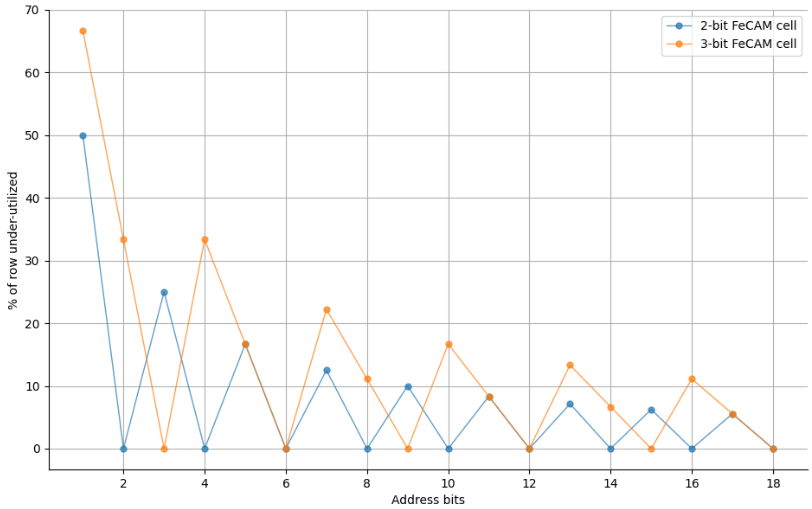


**Figure 6.23:** Under-utilization of 2- and 3-bit FCAM cells with regards to the address length.

Compared to related work as depicted in Table 6.2 [105] [106] this implementation performs 1.68× faster than the fastest, hybrid logic implementation and up to 2.6× faster than the slowest static implementation. With regards to area, this 2-bit/cell solution performs slightly better than the static AND logic based CMOS decoder.

A downside of using a multi-bit approach for address decoding is that the address must be aligned to the number of the bits each cell is able to store. When this is not the case, under-utilization of the hardware occurs, which might put the proposed CAM solution at a disadvantage against other solutions. This reality is reflected in Figure 6.23, where the percentage of row under-utilization is shown, in relation to the address length. For shorter addresses ($\leqslant 8$ bits) the under-utilization can be significant; as the address width increases, any under-utilised cells have lesser impact. On the plot we are considering 2-bit and 3-bit cells; the 2-bit FCAM decoder is more efficiently used with even-sized addresses, while the 3-bit decoder is more efficient when the address width is a multiple of 3.

| Design (5-to-32 decoder) | Transistor Count | Delay (ns) |
|---|---|---|
| Static CMOS (AND Logic) | 308 | 2.25 |
| Static CMOS (NOR Logic) | 220 | 2.16 |
| Pseudo-nMOS Logic | 176 | 1.69 |
| Mixed Logic [106] | 154 | 1.58 |
| Hybrid Logic | 132 | 1.46 |
| FeFET CAM (2-bits / cell) | 224 | 0.87 |

**Table 6.2:** Comparison of the transistor count and delay of different implementations.

# 7 Conclusion

This chapter provides a summary with all the achievements of this dissertation and highlights some potential future research directions

## 7.1. Summary

*Chapter 1* of this thesis introduced the concept and the main research questions that would drive our research on novel memory address decoders. We highlighted the importance of emerging memory technologies not only in fields like In-Memory Computing, where the research is currently intense, but also in the attempt of solving problems pertain to ”conventional” Computer Architecture. One of these problems was identified in the field of memory address decoders; although hardwired, dynamic NAND decoders are widely used in random access memories for decoding parts of the address - due to their speed and simplicity - their testing remains complex and cumbersome. Content-addressable memories can provide similar decoding functionality, however, the current CMOS technology restricts the usage of CAMs only to specialised applications, due to their high cost, high power consumption and large size. Emerging memory technologies like FeFETs, have managed to make a breakthrough in the field of CAMs by enabling the manufacturing of smaller, cheaper and more energy efficient cells. In the light of these advancements, using a CAM array for its decoding functionality is now within reach and can provide significant advantages over the established CMOS technology.

*Chapter 2* provided the background on FeFETs. In this chapter we discussed the physics the materials and the applications of FeFETs. Key concepts were explained, as well as a general introduction to the possibilities of this emerging technology.

*Chapter 3* provided an overview of address decoders. Several different types were discussed, including static, dynamic and column decoders. Also key concepts were introduced like decoder stages and predecoding.

*Chapter 4* was an extensive overview of Content-Addressable memories. Starting from established designs and technology (like CMOS), we discussed designs and applications. The chapter also introduced the new generation of CAM cells built with emerging memory technologies, showcaseing their advantages over previous designs.

*Chapter 5* introduced and discussed the FCAM row decoder. Combining the insight from the previous chapters, we presented the idea of a memory address decoder based on CAMs built with FeFET devices.

*Chapter 6* presented the experimental platform and the simulation results. In this chapter we also discussed the results and the challenges of the project and we provided the solution for testing this new type of decoder based on simple march tests. We were able to also compare certain metrics of the design, with other state-of-the-art conventional decoder designs. Our simulation results show an improved delay against state of the art designs, by at least $1.68\times$, while the transistor count remains on par with those implementations. We also demonstrated that a cell which is able to store more than four levels can provide significant area advantage over conventional designs.

## 7.2. Future Work

This thesis was built upon an idea which was as novel as largely unexplored. On top of that, the area of FeFETs is still under intense research, with new materials and structures being constantly proposed, while older ones are being revised and improved. Hence, what we propose in this thesis is far from perfect or complete. We would mostly describe it as one small brick out of an entire building with countless parameters and possibilities that must be further researched. For this reason, in this section we provide a list of directions that have research interest and can act as points of further development.

We have showed that the testability of this decoder is a significant advantage over competing solutions. In addition, the FCAM-based address decoder showed a notable resiliency in mild defects. However, we have not exhausted all possible kinds of defects - such as capacitive defects - and despite simulating the resistive defects in a small array, further investigation is needed especially with larger arrays and confounded effects from several defects.

Conventional CAMs are inherently power-hungry circuits, however, this seems to change with emerging memory technologies, so energy consumption is also a topic that needs to be investigated and compared with other state-of-the-art designs. At this point, continuous advancements in the fields of ferroelectric materials should be taken into consideration, since different materials (and the compact models originating from them) should be reviewed.

Another area for improvement lies in the maximum number of levels a FCAM cell is able to store. We showcased that a four-level FCAM is possible and there are already reports for an eight-level one. Such an increase it will further reduce the area needed to store a multi-bit address and will make the FCAM decoder significantly more area-efficient compared to other designs.

Finally, the structure of the array itself needs to be reviewed and revised, as further lowering the ML capacitance will consequently improve the overall delay of the decoder. Researching the robustness of the multi-bit cell with regards to process variation and also assessing the reliability limitations compared to other technologies is also crucial in establishing the FCAM decoder as an alternative design.

# References

1. Van de Goor, A. J. *Testing Semiconductor Memories: Theory and Practice* (J. Wiley & Sons, 1991).
2. Van de Goor, A. J. & Gaydadjiev, G. March U: A test for unlinked memory faults. *IEE Proceedings on Circuits, Devices and Systems, IEE Proceedings* **144,** 155–160 (June 1997).
3. Van de Goor, A. J., Gaydadjiev, G. N., Yarmolik, V. N. & Mikitjuk, V. G. *March LA: A test for linked memory faults* in *Proceedings of the 1997 European Design and Test Conference, pp. 627-632, Mar. 1997, issn* (Mar. 1997), 1066–1409.
4. Yavits, L., Weiser, U. & Ginosar, R. Resistive Address Decoder. *IEEE Computer Architecture Letters* **16,** 141–144 (July 2017).
5. Yavits, L., Kaplan, R. & Ginosar, R. Enabling Full Associativity with Memristive Address Decoder. *IEEE Micro* **38,** 32–40 (Sept. 2018).
6. J.Valasek. Minutes of the Washington Meeting, April 23 and 24, 1920. *Phys. Rev.* **15,** 505–564 (6 June 1920).
7. Gundlapudi Ravikumar, P. & Khan, A. in *Emerging Ferroelectric Materials and Devices* (eds Heron, J. & Mi, Z.) 137–164 (Elsevier, 2023).
8. Böscke, T. S., Müller, J., Bräuhaus, D., Schröder, U. & Böttger, U. Ferroelectricity in hafnium oxide thin films. *Applied Physics Letters* **99,** 102903. ISSN: 0003-6951. eprint: https://pubs.aip.org/aip/apl/article-pdf/doi/10.1063/1.3634052/13708834/102903_1_online.pdf (Sept. 2011).
9. Mikolajick, T., Park, M. H., Begon□Lours, L. & Slesazeck, S. From Ferroelectric Material Optimization to Neuromorphic Devices. *Advanced Materials* **35** (Feb. 2023).
10. Lehninger, D. *et al. Enabling Ferroelectric Memories in BEoL - towards advanced neuromorphic computing architectures* in *2021 IEEE International Interconnect Technology Conference (IITC)* 2021 IEEE International Interconnect Technology Conference (IITC) (IEEE, Kyoto, Japan, July 6, 2021), 1–4. ISBN: 978-1-72817-632-1. https://ieeexplore.ieee.org/document/9537346/ (2023).
11. Yayla, M. *et al.* Reliable Binarized Neural Networks on Unreliable Beyond Von-Neumann Architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers* **69,** 2516–2528. ISSN: 1549-8328, 1558-0806. https://ieeexplore.ieee.org/document/9732897/ (2023) (June 2022).
12. Jot, S. *et al. FeFET-Based Neuromorphic Architecture with On-Device Feedback Alignment Training* in *2020 21st International Symposium on Quality Electronic Design (ISQED)* 2020 21st International Symposium on Quality Electronic Design (ISQED) (IEEE, Santa Clara, CA, USA, Mar. 2020), 317–322. ISBN: 978-1-72814-207-4. https://ieeexplore.ieee.org/document/9137035/ (2023).
13. Looney, D. H. *Semiconductive translating device* US Patent 2,791,758. May 1957.
14. Morton, J. A. *Electrical switching and storage* US Patent 2,791,761. May 1957.
15. Wu, S.-Y. A new ferroelectric memory device, metal-ferroelectric-semiconductor transistor. *IEEE Transactions on Electron Devices* **21,** 499–504 (1974).
16. Obradovic, B., Rakshit, T., Hatcher, R., Kittl, J. A. & Rodder, M. S. Modeling Transient Negative Capacitance in Steep-Slope FeFETs. *IEEE Transactions on Electron Devices* **65,** 5157–5164 (2018).
17. Ni, K., Li, X., Smith, J. A., Jerry, M. & Datta, S. Write Disturb in Ferroelectric FETs and Its Implication for 1T-FeFET AND Memory Arrays. *IEEE Electron Device Letters* **39,** 1656–1659 (2018).

18. Huang, S. *et al.* A Retention Model for Ferroelectric-Gate Field-Effect Transistor. *IEEE Transactions on Electron Devices* **58,** 3388–3394 (2011).

19. Sugibuchi, K., Kurogi, Y. & Endo, N. Ferroelectric field□effect memory device using Bi4Ti3O12 film. *Journal of Applied Physics* **46,** 2877–2881. ISSN: 0021-8979 (July 1975).

20. Chatterjee, K. *et al.* Self-Aligned, Gate Last, FDSOI, Ferroelectric Gate Memory Device With 5.5-nm Hf0.8Zr0.2O2, High Endurance and Breakdown Recovery. *IEEE Electron Device Letters* **38,** 1379–1382 (2017).

21. Ma, T. & Han, J.-P. Why is nonvolatile ferroelectric memory field-effect transistor still elusive? *IEEE Electron Device Letters* **23,** 386–388 (2002).

22. Yoon, S.-J. *et al.* Improvement in Long-Term and High-Temperature Retention Stability of Ferroelectric Field-Effect Memory Transistors With Metal–Ferroelectric–Metal–Insulator–Semiconductor Gate-Stacks Using Al-Doped HfO2 Thin Films. *IEEE Transactions on Electron Devices* **67,** 499–504 (2020).

23. Horita, S. & Trinh, B. N. Q. Nondestructive Readout of Ferroelectric-Gate Field-Effect Transistor Memory With an Intermediate Electrode by Using an Improved Operation Method. *IEEE Transactions on Electron Devices* **55,** 3200–3207 (2008).

24. Böscke, T. S., Müller, J., Bräuhaus, D., Schröder, U. & Böttger, U. Ferroelectricity in hafnium oxide thin films. *Applied Physics Letters* **99,** 102903. ISSN: 0003-6951 (Sept. 2011).

25. Tang, M., Xu, X., Ye, Z., Sugiyama, Y. & Ishiwara, H. Impact of HfTaO Buffer Layer on Data Retention Characteristics of Ferroelectric-Gate FET for Nonvolatile Memory Applications. *IEEE Transactions on Electron Devices* **58,** 370–375 (2011).

26. Böscke, T. *et al.* Phase transitions in ferroelectric silicon doped hafnium oxide. *Applied Physics Letters* **99** (2011).

27. Mulaosmanovic, H. *et al.* Investigation of Accumulative Switching in Ferroelectric FETs: Enabling Universal Modeling of the Switching Behavior. *IEEE Transactions on Electron Devices* **67,** 5804–5809 (2020).

28. Mueller, S. *et al.* From MFM Capacitors Toward Ferroelectric Transistors: Endurance and Disturb Characteristics of $\mathrm{HfO}_2$-Based FeFET Devices. *IEEE Transactions on Electron Devices* **60,** 4199–4205 (2013).

29. Zeng, B. *et al.* Compatibility of HfN Metal Gate Electrodes With Hf0.5Zr0.5O2 Ferroelectric Thin Films for Ferroelectric Field-Effect Transistors. *IEEE Electron Device Letters* **39,** 1508–1511 (2018).

30. Liu, H. *et al.* ZrO2 Ferroelectric FET for Non-volatile Memory Application. *IEEE Electron Device Letters* **40,** 1419–1422 (2019).

31. Ohmi, S., Kim, M. G., Kataoka, M., Hayashi, M. & Mailig, R. M. D. Effect of Kr/O2-Plasma Reactive Sputtering on Ferroelectric Nondoped HfO□ Formation for MFSFET With Pt Gate Electrode. *IEEE Transactions on Electron Devices* **68,** 2427–2433 (2021).

32. Dutta, S. *et al.* Logic Compatible High-Performance Ferroelectric Transistor Memory. *IEEE Electron Device Letters* **43,** 382–385 (2022).

33. Mallick, A. & Shukla, N. Evaluation of Bulk and SOI FeFET Architecture for Non-Volatile Memory Applications. *IEEE Journal of the Electron Devices Society* **7,** 425–429 (2019).

34. Ni, K., Jerry, M., Smith, J. A. & Datta, S. *A Circuit Compatible Accurate Compact Model for Ferroelectric-FETs* in *2018 IEEE Symposium on VLSI Technology, Honolulu, HI, USA* (2018), 131–132.

35. Athle, R. *Ferroelectric memristors - materials, interfaces and applications: doctoral thesis* 253 pp. ISBN: 9789180399784 (Department of Electrical and Information Technology, Lund University, Lund, 2024).

36. Gong, N. & Ma, T.-P. Why Is FE–HfO2 More Suitable Than PZT or SBT for Scaled Nonvolatile 1-T Memory Cell? A Retention Perspective. *IEEE Electron Device Letters* **37,** 1123–1126 (2016).

37. Stengel, M. & Spaldin, N. A. Origin of the dielectric dead layer in nanoscale capacitors. *Nature* **443,** 679–682. https://api.semanticscholar.org/CorpusID:4370147 (2006).

38. Müller, J., Polakowski, P., Mueller, S. & Mikolajick, T. Ferroelectric Hafnium Oxide Based Materials and Devices: Assessment of Current Status and Future Prospects. *ECS Journal of Solid State Science and Technology* **4.** https://api.semanticscholar.org/CorpusID:98281129 (2015).

39. Mehta, R. R., Silverman, B. D. & Jacobs, J. T. Depolarization fields in thin ferroelectric films. *Journal of Applied Physics* **44,** 3379–3385. https://api.semanticscholar.org/CorpusID:121487299 (1973).

40. Lomenzo, P. D. *et al.* Ferroelectric Hf1-xZrxO2 memories: device reliability and depolarization fields. *2019 19th Non-Volatile Memory Technology Symposium (NVMTS),* 1–8. https://api.semanticscholar.org/CorpusID:214623837 (2019).

41. Chouprik, A. *et al.* Origin of the retention loss in ferroelectric Hf0.5Zr0.5O2-based memory devices. *Acta Materialia* **204,** 116515. ISSN: 1359-6454. https://www.sciencedirect.com/science/article/pii/S1359645420309435 (2021).

42. Starschich, S., Menzel, S. & Böttger, U. Evidence for oxygen vacancies movement during wake-up in ferroelectric hafnium oxide. *Applied Physics Letters* **108,** 032903. ISSN: 0003-6951. eprint: https://pubs.aip.org/aip/apl/article-pdf/doi/10.1063/1.4940370/14476085/032903_1_online.pdf. https://doi.org/10.1063/1.4940370 (Jan. 2016).

43. Jiang, P. *et al.* Wake-Up Effect in HfO2-Based Ferroelectric Films. *Advanced Electronic Materials* **7,** 2000728. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aelm.202000728. https://onlinelibrary.wiley.com/doi/abs/10.1002/aelm.202000728 (2021).

44. Kim, H. J. *et al.* A study on the wake-up effect of ferroelectric Hf0.5Zr0.5O2 films by pulse-switching measurement. *Nanoscale* **8,** 1383–1389. http://dx.doi.org/10.1039/C5NR05339K (3 2016).

45. Shekhawat, A. *et al.* Effect of furnace annealing on the ferroelectricity of Hf0.5 Zr0.5O2 thin films. *Thin Solid Films.* https://api.semanticscholar.org/CorpusID:139797190 (2019).

46. Kozodaev, M. G. *et al.* Mitigating wakeup effect and improving endurance of ferroelectric HfO2-ZrO2 thin films by careful La-doping. *Journal of Applied Physics.* https://api.semanticscholar.org/CorpusID:128292300 (2019).

47. Toprasertpong, K. *et al.* Low Operating Voltage, Improved Breakdown Tolerance, and High Endurance in Hf0.5Zr0.5O2 Ferroelectric Capacitors Achieved by Thickness Scaling Down to 4 nm for Embedded Ferroelectric Memory. *ACS Applied Materials & Interfaces* **14.** PMID: 36319949, 51137–51148. https://doi.org/10.1021/acsami.2c15369 (2022).

48. Beyer, S. *et al. FeFET: A versatile CMOS compatible device with game-changing potential* in *2020 IEEE International Memory Workshop (IMW), Dresden, Germany: IEEE* (May 2020), 1–4.

49. Huang, Q. *et al.* FeFET Based In-Memory Hyperdimensional Encoding Design. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,* 1–1 (2023).

50. De, S. *et al.* Demonstration of Multiply-Accumulate Operation With 28 nm FeFET Crossbar Array. *IEEE Electron Device Lett.* **43,** 2081–2084 (Dec. 2022).

51. Chen, X. *et al.* Power and Area Efficient FPGA Building Blocks Based on Ferroelectric FETs. *IEEE Trans. Circuits Syst. I* **66,** 1780–1793 (May 2019).

52. Zhou, Z. *et al. Non-Destructive-Read 1T1C Ferroelectric Capacitive Memory Cell with BEOL 3D Monolithically Integrated IGZO Access Transistor for 4F2 High-Density Integration* in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)* (2023), 1–2.

53. Mikolajick, T., Park, M. H., Begon-Lours, L. & Slesazeck, S. From Ferroelectric Material Optimization to Neuromorphic Devices. *Advanced Materials* **35,** 2206042. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.202206042. https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.202206042 (2023).

54. Ni, K. *et al. SoC Logic Compatible Multi-Bit FeMFET Weight Cell for Neuromorphic Applications* in *2018 IEEE International Electron Devices Meeting (IEDM)* (2018), 13.2.1–13.2.4.

55. Pei, J. *et al.* Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572,** 106–111. ISSN: 0028-0836, 1476-4687. https://www.nature.com/articles/s41586-019-1424-8 (2024) (Aug. 2019).

56. Dutta, S. *et al.* Supervised Learning in All FeFET-Based Spiking Neural Network: Opportunities and Challenges. *Frontiers in Neuroscience* **14,** 634. ISSN: 1662-453X. https://www.frontiersin.org/article/10.3389/fnins.2020.00634/full (2024) (June 24, 2020).

57. Yu, E. Ferroelectric capacitors and field-effect transistors as in-memory computing elements for machine learning workloads. *Scientific Reports* (2024).

58. Dunkel, S. *et al. A FeFET based super-low-power ultra-fast embedded NVM technology for 22nm FDSOI and beyond* in *2017 IEEE International Electron Devices Meeting (IEDM)* 2017 IEEE International Electron Devices Meeting (IEDM) (IEEE, San Francisco, CA, USA, Dec. 2017), 19.7.1–19.7.4. ISBN: 978-1-5386-3559-9. https://ieeexplore.ieee.org/document/8268425/ (2023).

59. Kim, S. K., Oh, T. W., Lim, S., Ko, D. H. & Jung, S.-O. High-Performance and Area-Efficient Ferroelectric FET-Based Nonvolatile Flip-Flops. *IEEE Access* **9,** 35549–35561 (2021).

60. Aziz, A., Ghosh, S., Datta, S. & Gupta, S. K. Physics-based circuit compatible spice model for ferroelectric transistors. *IEEE Electron Device Letters* **37,** 805–808 (2016).

61. Ni, K. *et al.* Critical role of interlayer in hf 0.5 zr 0.5 o 2 ferroelectric fet nonvolatile memory performance. *IEEE Transactions on Electron Devices* **65,** 2461–2469 (2018).

62. Yin, X., Reis, D., Niemier, M. & Hu, X. S. *Ferroelectric FET Based TCAM Designs for Energy Efficient Computing* in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Miami, FL, USA* (2019), 437–442.

63. Tung, C.-T., Pahwa, G., Salahuddin, S. & Hu, C. A Compact Model of Ferroelectric Field-Effect Transistor. *IEEE Electron Device Letters* **43,** 1363–1366 (2022).

64. Lu, D., De, S. & Qiu, B.-H. *Ferroelectric FET Compact Model for Neuromorphic* 2020. https://dx.doi.org/10.21227/86qq-5n18.

65. Aziz, A., Ghosh, S., Datta, S. & Gupta, S. K. Physics-Based Circuit-Compatible SPICE Model for Ferroelectric Transistors. *IEEE Electron Device Letters* **37,** 805–808 (2016).

66. Van De Goor, A. & Gaydadjiev, G. *An analysis of (linked) address decoder faults* in *Proceedings. International Workshop on Memory Technology, Design and Testing (Cat. NO.97TB100159)* Proceedings. International Workshop on Memory Technology, Design and Testing (Cat. NO.97TB100159) (IEEE Comput. Soc, San Jose, CA, USA, 1997), 13–20. ISBN: 978-0-8186-8099-1. http://ieeexplore.ieee.org/document/619389/ (2024).

67. Ramirez-Chavez, S. Encoding don't cares in static and dynamic content-addressable memories. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **39,** 575–578 (1992).

68. Roth, A., Foss, D., McKenzie, R. & Perry, D. *Advanced ternary CAM circuits on 0.13 /spl mu/m logic process technology* in *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference (IEEE Cat. No.04CH37571)* (2004), 465–468.

69. Choi, S. *et al. A 0.7fJ/bit/search, 2.2ns search time hybrid type TCAM architecture* in *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)* (2004), 498–542 Vol.1.

70. Choi, S., Sohn, K. & Yoo, H.-J. A 0.7-fJ/bit/search 2.2-ns search time hybrid-type TCAM architecture. *IEEE Journal of Solid-State Circuits* **40,** 254–260 (2005).

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

72. Khellah, M. & Elmasry, M. *Use of charge sharing to reduce energy consumption in wide fan-in gates* in *1998 IEEE International Symposium on Circuits and Systems (ISCAS)* **2** (1998), 9–12 vol.2.

73. Zukowski, C. & Wang, S.-Y. *Use of selective precharge for low-power content-addressable memories* in *1997 IEEE International Symposium on Circuits and Systems (ISCAS)* **3** (1997), 1788–1791 vol.3.

74. Pagiamtzis, K. & Sheikholeslami, A. Content-addressable memory (CAM) circuits and architectures: a tutorial and survey. *IEEE Journal of Solid-State Circuits* **41,** 712–727 (2006).

75. Hsiao, I. Y.-L., Wang, D.-H. & Jen, C.-W. *Power modeling and low-power design of content addressable memories* in *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)* **4** (2001), 926–929 vol. 4.

76. Efthymiou, A. & Garside, J. *An adaptive serial-parallel CAM architecture for low-power cache blocks* in *Proceedings of the International Symposium on Low Power Electronics and Design* (2002), 136–141.

77. Efthymiou, A. & Garside, J. A CAM with mixed serial-parallel comparison for use in low energy caches. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **12,** 325–329 (2004).

78. Mohan, N. & Sachdev, M. *Low power dual matchline ternary content addressable memory* in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)* **2** (2004), II–633.

79. Cheng, K., Wei, C. & Jiang, S. Static divided word matching line for low-power content addressable memory design. *Proceedings - IEEE International Symposium on Circuits and Systems* **2.** 2004 IEEE International Symposium on Cirquits and Systems - Proceedings ; Conference date: 23-05-2004 Through 26-05-2004, II629–II632. ISSN: 0271-4310 (2004).

80. Gnawali, K. P. & Tragoudas, S. High-Speed Memristive Ternary Content Addressable Memory. *IEEE Transactions on Emerging Topics in Computing* **10,** 1349–1360 (2022).

81. Li, C., Graves, C. E., Sheng, X., *et al.* Analog content-addressable memories with memristors. *Nat Commun* **11,** 1638 (2020).

82. Ni, K. *et al.* Ferroelectric ternary content-addressable memory for one-shot learning. *Nature Electronics* (2019).

83. Kazemi, A. *et al.* FeFET Multi-Bit Content-Addressable Memories for In-Memory Nearest Neighbor Search. *IEEE Transactions on Computers* **71,** 2565–2576 (Oct. 1, 2022).

84. Kazemi, A. *et al. A Flash-Based Multi-Bit Content-Addressable Memory with Euclidean Squared Distance* in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA* (2021), 1–6.

85. Chang, M.-F. *et al.* A 3T1R Nonvolatile TCAM Using MLC ReRAM for Frequent-Off Instant-On Filters in IoT and Big-Data Processing. *IEEE Journal of Solid-State Circuits* **52,** 1664–1679 (2017).

86. Shim, M., Jeong, J., Song, J., Kim, Y. & Choi, W. Segmented Match-Line and Charge-Sharing Based Low-Cost TCAM. *IEEE Transactions on Circuits and Systems II: Express Briefs* **69,** 5104–5108 (2022).

87. Yin, X. *et al.* FeCAM: A Universal Compact Digital and Analog Content Addressable Memory Using Ferroelectric. *IEEE Transactions on Electron Devices* **67,** 2785–2792 (July 2020).

88. Li, C. *et al. A Scalable Design of Multi-Bit Ferroelectric Content Addressable Memory for Data-Centric Computing* in *2020 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2020, pp. 29.3..3.4* (2020), 1–29.

89. Marchand, C. *et al.* A FeFET-Based Hybrid Memory Accessible by Content and by Address. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* **8,** 19–26 (2022).

90. Bekdache, O. *et al. Scalable Complementary FeFET CAM Design* in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)* (2023), 1–5.

91. Jin, C. *et al.* A Multi-Bit CAM Design With Ultra-High Density and Energy Efficiency Based on FeFET NAND. *IEEE Electron Device Letters* **44,** 1104–1107 (2023).

92. Li, J., Montoye, R. K., Ishii, M. & Chang, L. 1 Mb 0.41 µm² 2T-2R Cell Nonvolatile TCAM With Two-Bit Encoding and Clocked Self-Referenced Sensing. *IEEE Journal of Solid-State Circuits* **49,** 896–907 (2014).

93. Chang, M.-F. *et al. 17.5 A 3T1R nonvolatile TCAM using MLC ReRAM with Sub-1ns search time* in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers* (2015), 1–3.

94. Yin, X. *et al.* An Ultra-Dense 2FeFET TCAM Design Based on a Multi-Domain FeFET Model. *IEEE Transactions on Circuits and Systems* **66,** 1577–1581 (Sept. 2019).

95. Venkata Mahendra, T., Wasmir Hussain, S., Mishra, S. & Dandapat, A. Energy-Efficient Precharge-Free Ternary Content Addressable Memory (TCAM) for High Search Rate Applications. *IEEE Transactions on Circuits and Systems I: Regular Papers* **67,** 2345–2357 (2020).

96. Venkata Mahendra, T., Mishra, S. & Dandapat, A. Self-Controlled High-Performance Precharge-Free Content-Addressable Memory. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **25,** 2388–2392 (2017).

97. Mahendra, T. V., Hussain, S. W., Mishra, S. & Dandapat, A. Low discharge precharge free matchline structure for energy-efficient search using CAM. *Integration* **69,** 31–39. ISSN: 0167-9260. `https://www.sciencedirect.com/science/article/pii/S0167926018305418` (2019).

98. Yin, X. *et al.* Ferroelectric Ternary Content Addressable Memories for Energy-Efficient Associative Search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **42,** 1099–1112. ISSN: 0278-0070, 1937-4151. `https://ieeexplore.ieee.org/document/9852752/` (2023) (Apr. 2023).

99. Aziza, H. *et al.* Multi-Level Control of Resistive RAM (RRAM) Using a Write Termination to Achieve 4 Bits/Cell in High Resistance State. *Electronics* **10.** ISSN: 2079-9292. `https://www.mdpi.com/2079-9292/10/18/2222` (2021).

100. Sheng, X. *et al.* Low-conductance and multilevel cmos-integrated nanoscale oxide memristors. *Advanced Electronic Materials* (2019).

101. Liu, L., Chi, P., Li, S., Cheng, Y. & Xie, Y. *Building energy-efficient multi-level cell STT-RAM caches with data compression* in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)* (2017), 751–756.

102. Li, C. *et al.* Analog content-addressable memories with memristors. *Nature Communications* **11,** 1638. ISSN: 2041-1723. `https://www.nature.com/articles/s41467-020-15254-4` (2023) (Apr. 2, 2020).

103. Ni, K., Li, X., Smith, J. A., Jerry, M. & Datta, S. Write Disturb in Ferroelectric FETs and Its Implication for 1T-FeFET AND Memory Arrays. *IEEE Electron Device Letters* **39,** 1656–1659 (Nov. 2018).

104. Weste, N. & Harris, D. *CMOS VLSI Design: A Circuits and Systems Perspective* 4th. ISBN: 0321547748 (Addison-Wesley Publishing Company, USA, 2010).

105. Reddy, M. M., Challa, K. V. & Raja, B. S. *An Energy Efficient Static Address Decoder for High-Speed Memory Applications* in *2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India* (2022), 50–53.

106. Balobas, D. & Konofaos, N. Design of low-power high-performance 2-4 and 4-16 mixed-logic line decoders. *IEEE transactions on circuits and systems-II: express briefs* **64,** 176–180 (2017).

# A

# Publication: FeFET-based CAM Address Decoders

# FeFET-based CAM Address Decoders

Thomas Makryniotis, Georgi Gaydadjiev, Said Hamdioui and Mottaqiallah Taouil
EEMCS, Department of Quantum and Computer Engineering
Delft University of Technology, the Netherlands
T.Makryniotis@student.tudelft.nl, G.N.Gaydadjiev@tudelft.nl, S.Hamdioui@tudelft.nl, M.Taouil@tudelft.nl

***Abstract*—Address decoders are an integral part of random access memories. They are typically implemented using fast logic optimised for low latency. The latter, however, are difficult to test, while their repair is considered to be impossible. In this work we propose a highly scalable and testable address decoder solution, based on Content-Addressable Memories build with ferroelectric transistors (FeFET). Our solution has a transistor count close to the state of the art, while outperforms it in terms of latency. Due to its regular 2D structure, our proposal's testability is comparable to that of memory arrays. Moreover, adding a few spare rows will enable end-of-production repair, in the presence of manufacturing defects. By additionally increasing the number of address bits stored in a single FeFET CAM cell, potential area reductions of 30% - compared to the traditional dynamic NAND decoders - can be achieved.**

***Index Terms*—FeFET, decoders, ferroelectric, memory, testing**

## I. Introduction

Low latency address decoding remains a challenge in high-performance memory designs ranging from cache memories up to large main memory arrays [1]. This typically results in highly customised logic implementations for the targeted memory array that are fast but difficult to test [2]. Detecting address decoder faults is challenging since the tests have to "recover" the effects from the expected values obtained from different locations in the memory array [3, 4]. Moreover, detecting linked address decoder faults requires long march tests [2]. Implementing testable and ideally, repairable address decoders will not only help reducing end of production test times but will also improve yield.

Ferroelectric FET (FeFET) technology is emerging in many application domains ranging from non-volatile storage, up to reconfigurable hardware. Some examples of promising FeFET based applications are emulating atomic neuromorphic operations [5], hyper-dimensional encoding [6], multiply-accumulate operation crossbars [7] and energy/area efficient FPGA fabrics [8]. Clearly, both Academia and Industry consider using FeFET devices in promising solutions for various limitations faced by contemporary computing technology.

In this paper we propose a novel, low latency address decoder built using an array of two-FeFET Content Adressable Memory (CAM) cells. Each FeFET CAM cell stores at least two bits in order to reduce the overall array size. Our address decoder was simulated in 14nm CMOS technology and outperforms the state of the art in latency, by at least a factor of 1.7x. At the same time, its area stays on par with the baseline solutions. When more than two bits per CAM cell are used, additional area gains can be achieved. Moreover, due to its memory-like regular organization, our decoder is easy to test and by adding few spare rows end-of-production-line repair can be facilitated.

The main contributions of this paper are:

- a fast FeFET CAM based address decoder design;
- a careful investigation of the possible partial open defects in the proposed two-transistor CAM cells;
- a set of march tests able to detect the above defects.

The remainder of the paper is organized as follows. Section II introduces the necessary background. Section III describes the proposed solution, while Section IV presents our experimental results and the comparison against state of the art. Finally, Section V concludes the paper and provides some directions for future work.

## II. Background and Related Work

First, we introduce the Ferroelectric FETs and the two main compact models used by the research community. Next, we provide the necessary background on Multi-bit Content Addressable Memories (MCAMs) and describe the most relevant proposals using FeFETs. At the end of this section we describe related work on CAMs usage in TLBs.

### *A. Ferroelectric FETs*

Ferroelectric Field-Effect Transistors (FeFETs) are created by integrating a layer of a ferroelectric material (FE layer) into the gate stack of a regular MOSFET. This layer introduces an additional capacitance, coupled with the gate capacitance of the MOSFET device, while its polarization can be manipulated by an appropriate voltage pulse applied on the gate of the transistor. The implications of such a device are obvious; by controlling the polarization (and as a consequence, the voltage) of the FE layer, it is possible to store binary values in a FeFET and read them afterwards, in a non-destructive way, as $I_{ON}$ and $I_{OFF}$ will be associated with a logic '1' or '0'. Since the polarization of the FE layer within a FeFET is retained, the device behaves as a non-volatile storage element.

FeFETs offer a way to realize a MOSFET with a "programmable" threshold voltage. An invaluable property is the partial polarization switching of the FE layer; by utilizing that, a FeFET storing different voltage levels is possible. The partial polarization of the FE layer can be achieved by controlling the duration and/or the amplitude of the programming pulses. In this work we are using constant-width, variable-amplitude pulses for both programming and reading.

FeFET devices are modelled using two main approaches: the Landau-Khalatnikoff (L-K) model which is based on time-dependent equations that describe the relationship between the polarization ($P$) and the electric field ($E$) [9] and the Preisach model, built upon the fact that a FE thin-film consists of multiple independent domains, with a distribution of coercive fields [10], [11]. While L-K model is useful in specific cases, it also has a few drawbacks. The first being that it assumes a single domain FE material, while in practice it is usually poly-crystalline with multiple domains. The second limitation is that the $V_{gs}$ applied to the FeFET can either keep the device polarization or switch it, as reported in [12]. It is impossible to turn the transistor completely off and maintain polarization, which means that the model cannot reproduce the basic non-volatile operation of FeFET [12]. In our work, we make use of a Preisach compact model, that can well mimic the aforementioned behaviour [11].

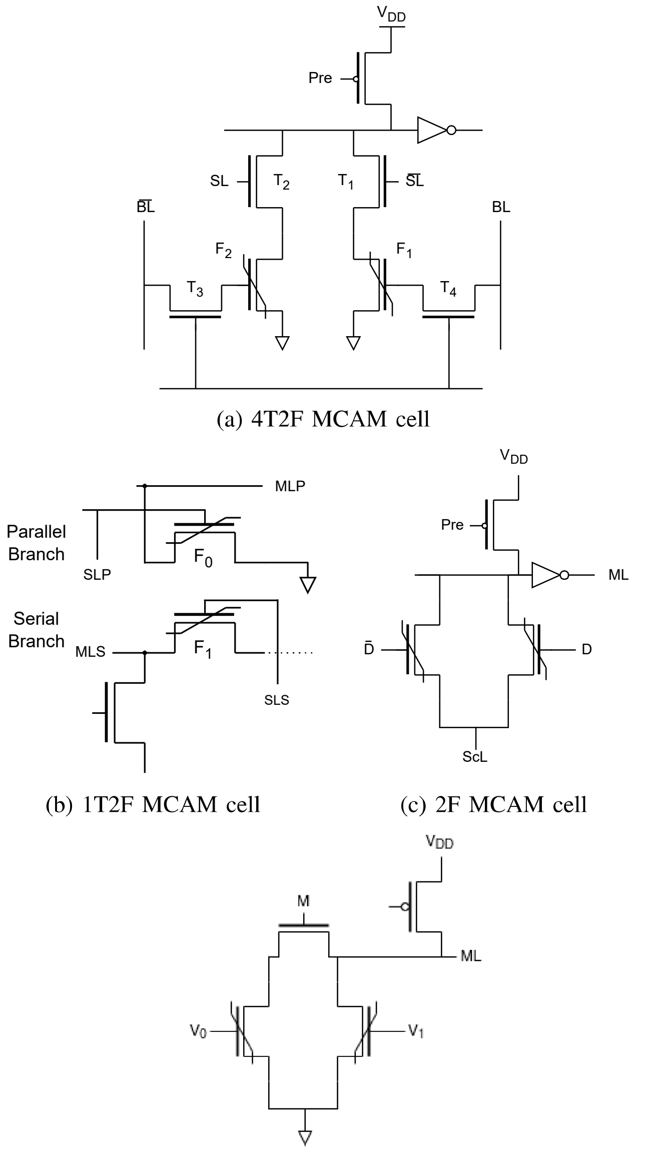
### B. Multi-bit Content Addressable Memories

Content-addressable memories (CAMs) function as associative memories, comparing input data with stored data and yielding addresses for matching data [13]. Their parallel processing capability across multiple data rows renders CAM-based searches notably faster compared to other alternatives. CAMs find particular appeal in applications requiring rapid and energy-efficient search operations, such as network packet forwarding [14] and wide database searches [15]. Ternary CAMs (TCAMs) enjoy widespread adoption and are being researched for, among others, neuromorphic computing [16], while Analog CAMs (ACAMs) have been used as TCAMs alternatives with increased data density, reduced operational energy and area for in-memory processing circuits [17].

An ACAM cell that stores multiple, narrow, non-overlapping ranges, can be viewed as a high-density digital CAM referred here as Multi-bit Content Addressable Memories (MCAM), with each range signifying a distinct state. The primary disparity between MCAMs and ACAMs lies in their search capabilities: every MCAM cell exclusively seeks within a restricted set of input values, each corresponding to a state, while ACAMs search across an infinite range of inputs. In MCAMs, the stored ranges maintain a one-to-one correlation with the inputs. Therefore, if there are four narrow, non-overlapping ranges and specific inputs, the ACAM would manifest as a 4-state or 2-bit MCAM. Consequently, MCAM can be viewed as a discretised and robust case of ACAM [18].

### C. FeFET-based MCAM designs

Almost all of the recent efforts aiming to realize MCAMs are based on emerging memory technologies such as FeFETs and ReRAM [16], [17], [18], while in a few cases, floating-gate FLASH cells are being proposed [19]. When it comes to FeFETs there are three main designs: the 4T-2FeFET [12], the 2FeFET [12], [20] and the 2FeFET-1T [21]. The above designs promise a much better energy efficiency as compared to their memristor counterparts and marginally better than the 16T CMOS implementations [12], while they all require significantly less area (see Figure 1). In addition to these designs, a few others have been proposed; the TC-MEM design proposed in [22] extends the 2-FeFET cell so it can be accessed either by content or by address (Figure 1-d), while the authors of [23] introduced a complementary FeFET CAM design utilizing a pair of p- and n- FeFET devices and reporting significantly reduced search latency and failure probability. Another novel design was also proposed in [24], which uses two n-FeFET devices connected in serial topology; the authors reported promising results with regards to search energy, compared to other designs.



(a) 4T2F MCAM cell

(b) 1T2F MCAM cell

(c) 2F MCAM cell

(d) TC-MEM CAM cell

Fig. 1: Four different designs of FeFET-based CAM cells.

In this work we use the most compact design, which consists of only two FeFETs.

### D. *CAM-based Address Decoders*

Hardwired dynamic NAND address decoders are widely used in random access memories to decode parts of the address. The function of a dynamic NAND decoder is quite simple: each row of the decoder corresponds to a row of the decoder's truth table and a certain number of transistors are hardwired to either the address signals, or their inverted ones. Based on which transistors are ON or OFF, the appropriate line of the address decoder is activated. Practically, the same can be achieved by using a content-addressable memory which has all the available addresses pre-programmed: an address is used as a query and if a CAM row contains it, it signals a Match Line (ML). An implementation was suggested for the first time in [25] where the proposed CAM array was based on ReRAM. This design was discussed in a more elaborate way in [26] where a few applications were also proposed such as fully associative TLBs and virtually addressable memory.

## III. Proposed FCAM-based solution

Our work combines carefully chosen elements and ideas from the aforementioned works. The FeFET-based CAM cell provides a viable alternative to the originally proposed ReRAM devices. On the other hand, the idea of replacing a logic based address decoder with a CAM array is appealing not only because of the potential performance advantages but also from a testability perspective.

Testing a decoder implemented with CMOS logic can be challenging, while repairing such structure could turn virtually impossible [3]. Replacing the decoder with a CAM array immediately shifts the address decoder testing challenges to the realm of memory array testing.

Despite the focus on testing, we could not overlook other possible advantages, such as area and power consumption. While the previous works suggest a single-level resistive CAM cell, we are leveraging the benefits of the multi-bit CAM cells in order to reduce the overall decoder area. Combining this with the virtually effortless integration with existing CMOS processes, the intrinsic advantage of FeFETs against ReRAM on search energy [16] and the comparable search delay, the FCAM address decoder becomes an appealing solution.

### A. *Structure of the array*

As we have already mentioned, we chose to base our design on the ultra-compact 2FeFET MCAM cell. This compact design however, comes at the cost of increased complexity for the writing and reading scheme and parasitics from neighbouring cells. In order to deal with these challenges, we implemented a programming scheme similar to those proposed in [18], [27] and [28]. The programming scheme is based on analog-inverse voltages applied on each of the FeFETs. The map of the different states is shown in Figure 3.

A simple $2 \times 3$ FCAM array is depicted in Figure 2 as a motivational example of the array organization. Bitlines are shared among cells in the same column, while match lines and source lines (ML and ScL) are shared by all cells in the same row. All match lines are precharged and retain their charges in case of a match. However, if a mismatch occurs, at least one of the FeFETs in one of the CAM cells from that row will open, leading to a discharge of the corresponding ML resulting in a '0' output.

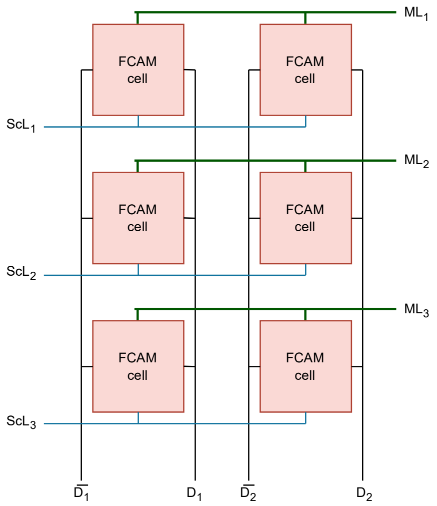


Fig. 2: A motivational example of a $2 \times 3$ FCAM array.

### B. *Program/Read Scheme*

The Program/Read scheme idea is straight-forward; the right transistor is programmed to the defined threshold voltage on the right side of the state to be stored. Then, the left transistor is programmed to the analog-inverse of the threshold voltage on the left side of the state. The analog-inverse of a signal is defined as the one that has the same distance from the centre line as the original, hence, they are symmetrical. Despite this scheme working well for a single cell, applying it to an array introduces certain challenges. On each column, the gates of the FeFETs are connected to the same $D$ and $\bar{D}$ bitlines. This creates problems during subsequent program (or erase) of a cell, since the write voltage will be applied to every other cell on the same column. To mitigate this, it is necessary to implement an inhibition bias scheme with which a $V_W/2$ voltage is applied on the ScL lines of the cells not being programmed. The same inhibition bias scheme has been used in the aforementioned works and is derived by the results reported at [28].

### C. *Testing and Fault Tolerance*

One of the main advantages of the proposed FCAM address decoder is its testability. In order to demonstrate this, we introduced partial open defects inside a FCAM cell and evaluated both its functional margins and a simple cell failure detection approaches. We simulated the different defects in isolation, by introducing a variable resistor in four different locations inside the FCAM cell. Figure 4 depicts the sites where the resistors emulating partial open defects were placed. Figure 5 shows the ranges within which the FCAM cell operates correctly (the light areas on the left), and thus a partially open defect

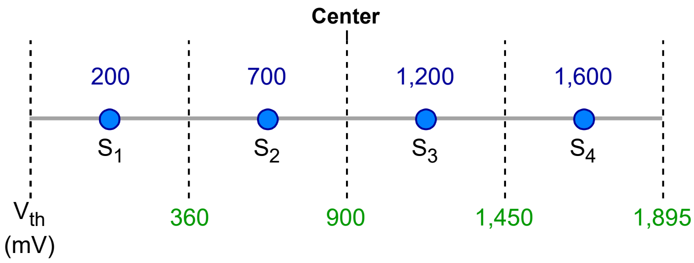


Fig. 3: The four levels of the proposed MCAM cell. The blue dots are read voltages ($V_R$) while the dotted lines represent the threshold voltages.

| Defect site | Tests |
|---|---|
| $R_{defA}$ | $\{(W_{S2}, R_{S1})\}$ |
| $R_{defB}$ | $\{(W_{S3}, R_{S4})\}$ |
| $R_{defC}$ | $\{(W_{S3}, R_{S4})\}$ |
| $R_{defD}$ | $\{(W_{S2}, R_{S1})\}$ |

TABLE I: March tests used to detect faulty FCAM cells.

will not disturb the correct cell operation. Moreover, it also shows the areas where a defect will cause all of the reading operations to fail (the dark areas on the right). The dashed area in the middle shows the range where a defect may or may not cause incorrect operation depending on the specific read operation used. Obviously we have to look for the 'most sensitive' read operation that will ensure detection of defects in the two concatenated faulty areas on the right side. Please note the FCAM design tolerance to partially open defects. Values from zero up to few kilo Ohms will not have any impact on its correct operation.

Detecting faulty FCAM cells is easy requiring a short march test, consisting of a single, two operations, march element per defect. A fail in any of these tests can immediately flag the cell under test as defective and corrective actions, e.g., row repair when spare rows are available, can be taken. Table I lists the tests for every defect considered in our study. The first operation writes a specific state to the cell, e.g., $W_{S3}$, and the second operation queries another state, e.g., $R_{S1}$. The reason for attempting to read back a different state than the one written, is that partial open faults are affecting the cell's ability to properly discharge the matching line (ML) in case of a mismatch, thus leading to "false-match" errors. By querying a state different from the one written, we ensure that a match indicates the cell is definitely defective.

## IV. Evaluation and Methodology

We conducted the simulations of our 5-to-32 FCAM address decoder with Cadence Spectre, using the model described in [11]. We were provided two versions of the model, one based on the PTM 14nm FinFET process and the other based on the PTM 45nm planar process. We used the 14nm process for both our FCAM cells and the sensing circuitry. The FeFET dimensions were set to $W \times L$ of $75 \times 30$ nm, respectively. The width reported here is the effective width $W_{eff}$ which

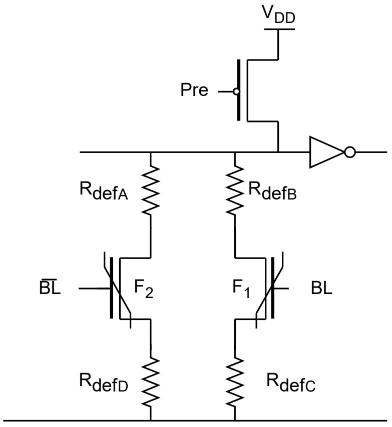


Fig. 4: The sites where resistive defects in a cell are inserted (high resistor values mimic partially open defects).

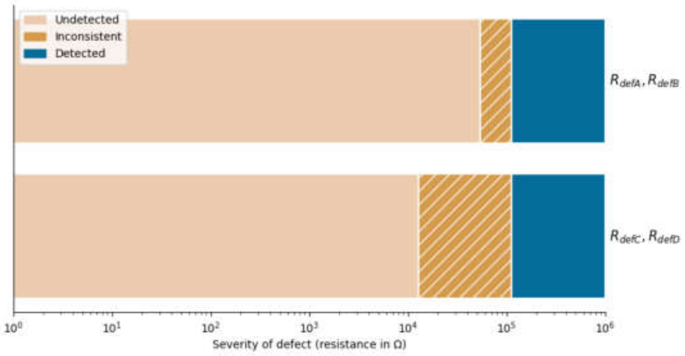


Fig. 5: Detectability ranges of resistive defects located on the ($R_{defA}$-$R_{defD}$) sites.

is calculated based on the dimensions of the Fin according to the equation $W_{eff} = 2 \times Fin_{Height} + Fin_{Width}$. The sensing circuitry consists of two inverters connected in series, with each of their transistors sized $80 \times 30$ nm. The supply voltage used in our circuit simulation was 0.8V.

The FeFET programming pulses had constant duration of 200ns, while the read pulses used were 2.5 ns. Evaluating the decoder delay included determining the total capacitance of the decoder row. For a 5-to-32 decoder, assuming an average of three cells per decoder row, together with the peripheral circuitry (the pre-charge PMOS and the two inverters used for sensing the output signal), as well as the capacitance of the transmission line itself, we estimated the total capacitance as $35fF$. Under the above conditions, our simulations resulted in a propagation delay of approximately 870 ps.

Compared to related work as depicted in Table II [29] [30] our implementation performs 1.68× faster than the fastest, hybrid logic implementation and up to 2.6× faster than the slowest static implementation. With regards to area, our 2-bit/cell solution performs slightly better than the static AND logic based CMOS decoder. However, as it was shown before [18], an 8-state, 3-bit FCAM cell is feasible; by exploiting this denser design we can further reduce the number of transistors

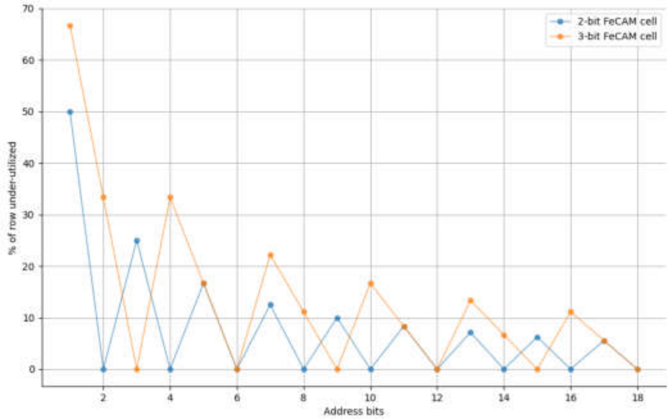


Fig. 6: Under-utilization of 2- and 3-bit FCAM cells with regards to the address length.

| Design (5-to-32 decoder) | Transistor Count | Delay (ns) |
|---|---|---|
| Static CMOS (AND Logic) | 308 | 2.25 |
| Static CMOS (NOR Logic) | 220 | 2.16 |
| Pseudo-nMOS Logic | 176 | 1.69 |
| Mixed Logic [30] | 154 | 1.58 |
| Hybrid Logic | 132 | 1.46 |
| FeFET CAM (2-bits / cell) | 224 | 0.87 |

TABLE II: Comparison of the transistor count and delay of different implementations.

down to 160 for the considered 5-to-32 address decoder. This will result in a significant improvement over most of the designs shown in Table II.

A downside of using a multi-bit approach for address decoding is that the address must be aligned to the number of the bits each cell is able to store. When this is not the case, under-utilization of the hardware occurs, which might put the proposed CAM solution at a disadvantage against other solutions. This reality is reflected in Figure 6, where the percentage of row under-utilization is shown, in relation to the address length. For shorter addresses ($\leq 8$ bits) the under-utilization can be significant; as the address width increases, any under-utilised cells have lesser impact. On the plot we are considering 2-bit and 3-bit cells; the 2-bit FCAM decoder is more efficiently used with even-sized addresses while the 3-bit when the address width is a multiple of 3.

## V. Discussion, Conclusion and Future Work

In this paper we proposed a novel address decoding circuit built upon FeFET based Content Addressable Memory cells. We, (i) demonstrate a FeFET CAM based address decoder design with superior latency, (ii) conduct a thorough investigation of the partial open defects in the proposed CAM cells and (iii) define a set of march tests able to quickly detect the aforementioned defects. Our simulation results show an improved delay against state of the art designs, by at least 1.68×, while the transistor count remains on par with those implementations. The proposed address decoder structure can be tested for defects, using simple march tests consisting only of a single two-operations element.

Research in ferroelectrics has yielded promising results in recent years, especially when compared to other technologies such as ReRAM and traditional CMOS. Specifically for CAM designs, the authors of [27] have reported 3.5× to 3200× reduced write energy compared to CMOS and ReRAM designs, respectively, while the reductions on the search EDP were between 4.1× and 2.8×, respectively. The authors of [24] reported a further reduced number for the search energy, by more than 60%, using their in-series design. Considering the several advancements in materials and their compatibility with the existing CMOS manufacturing processes, FeFETs are the most promising candidate for CAM cells from all the emerging device technologies.

However, there are still important aspects being worth of further research such as the robustness of multi-bit cell with regard to process variation, assessing the reliability limitations compared to other technologies and evaluating other, device specific defects. These topics will be explored as continuation of the current research.

Future work will also include completing the design and the simulation of the decoder version based on 3-bit FCAM cells, while exploring its energy consumption and comparing it to other state of the art decoders is a natural part of this.

## VI. Acknowledgements

We are grateful to Dr. Kai Ni, from University of Notre Dame, for providing us the FeFET compact model used in our simulations.

## References

[1] Leonid Yavits et al. "Dual Mode Logic Address Decoder". In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2020, pp. 1–5. DOI: 10.1109/ISCAS45731.2020.9180587.

[2] A. J. van de Goor et al. "March LA: A test for linked memory faults". In: *Proceedings of the 1997 European Design and Test Conference, pp. 627-632, Mar. 1997, issn*. Mar. 1997, pp. 1066–1409.

[3] A. J. van de Goor. *Testing Semiconductor Memories: Theory and Practice*. J. Wiley & Sons, 1991.

[4] A. J. van de Goor and Georgi Gaydadjiev. "March U: A test for unlinked memory faults". In: *IEE Proceedings on Circuits, Devices and Systems, IEE Proceedings* 144 (June 1997), pp. 155–160. DOI: 10.1049/ip-cds:19971147.

[5] S. Beyer et al. "FeFET: A versatile CMOS compatible device with game-changing potential". In: *2020 IEEE International Memory Workshop (IMW), Dresden, Germany: IEEE*. May 2020, pp. 1–4. DOI: 10.1109/IMW48823.2020.9108150.

[6] Q. Huang et al. "FeFET Based In-Memory Hyperdimensional Encoding Design". In: *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* (2023), pp. 1–1. DOI: 10.1109/TCAD.2023.3253766.

[7] S. De et al. “Demonstration of Multiply-Accumulate Operation With 28 nm FeFET Crossbar Array”. In: *IEEE Electron Device Lett.* 43.12 (Dec. 2022), pp. 2081–2084. DOI: 10.1109/LED.2022.3216558.

[8] X. Chen et al. “Power and Area Efficient FPGA Building Blocks Based on Ferroelectric FETs”. In: *IEEE Trans. Circuits Syst. I* 66.5 (May 2019), pp. 1780–1793. DOI: 10.1109/TCSI.2018.2874880.

[9] A. Aziz et al. “Physics-based circuit compatible spice model for ferroelectric transistors”. In: *IEEE Electron Device Letters* 37 (2016), pp. 805–808.

[10] K. Ni et al. “Critical role of interlayer in hf 0.5 zr 0.5 o 2 ferroelectric fet nonvolatile memory performance”. In: *IEEE Transactions on Electron Devices* 65 (2018), pp. 2461–2469.

[11] K. Ni et al. “A Circuit Compatible Accurate Compact Model for Ferroelectric-FETs”. In: *2018 IEEE Symposium on VLSI Technology, Honolulu, HI, USA*. 2018, pp. 131–132.

[12] X. Yin et al. “Ferroelectric FET Based TCAM Designs for Energy Efficient Computing”. In: *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Miami, FL, USA*. 2019, pp. 437–442.

[13] K. Pagiamtzis et al. “Content-addressable memory (cam) circuits and architectures: A tutorial and survey”. In: *JSSC* (2006).

[14] H. J. Chao. “Next generation routers”. In: *Proceedings of the IEEE* (2002).

[15] J. P. Wade et al. “A ternary content addressable engine”. In: *JSSC* (1989).

[16] K. Ni et al. “Ferroelectric ternary content-addressable memory for one-shot learning”. In: *Nature Electronics* (2019).

[17] C. Li, C. E. Graves, X. Sheng, et al. “Analog content-addressable memories with memristors”. In: *Nat Commun* 11 (2020), p. 1638.

[18] A. Kazemi et al. “FeFET Multi-Bit Content-Addressable Memories for In-Memory Nearest Neighbor Search”. In: *IEEE Transactions on Computers* 71.10 (Oct. 1, 2022), pp. 2565–2576.

[19] A. Kazemi et al. “A Flash-Based Multi-Bit Content-Addressable Memory with Euclidean Squared Distance”. In: *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA*. 2021, pp. 1–6.

[20] X. Yin et al. “FeCAM: A Universal Compact Digital and Analog Content Addressable Memory Using Ferroelectric”. In: *IEEE Transactions on Electron Devices* 67.7 (July 2020), pp. 2785–2792.

[21] C. Li et al. “A Scalable Design of Multi-Bit Ferroelectric Content Addressable Memory for Data-Centric Computing”. In: *2020 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2020, pp. 29.3..3.4*. 2020, pp. 1–29.

[22] Cédric Marchand et al. “A FeFET-Based Hybrid Memory Accessible by Content and by Address”. In: *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* 8.1 (2022), pp. 19–26. DOI: 10.1109/JXCDC.2022.3168057.

[23] O. Bekdache et al. “Scalable Complementary FeFET CAM Design”. In: *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2023, pp. 1–5. DOI: 10.1109/ISCAS46773.2023.10181788.

[24] Chengji Jin et al. “A Multi-Bit CAM Design With Ultra-High Density and Energy Efficiency Based on FeFET NAND”. In: *IEEE Electron Device Letters* 44.7 (2023), pp. 1104–1107. DOI: 10.1109/LED.2023.3277845.

[25] L. Yavits, U. Weiser, and R. Ginosar. “Resistive Address Decoder”. In: *IEEE Computer Architecture Letters* 16.2 (July 2017), pp. 141–144.

[26] L. Yavits, R. Kaplan, and R. Ginosar. “Enabling Full Associativity with Memristive Address Decoder”. In: *IEEE Micro* 38.5 (Sept. 2018), pp. 32–40.

[27] X. Yin et al. “An Ultra-Dense 2FeFET TCAM Design Based on a Multi-Domain FeFET Model”. In: *IEEE Transactions on Circuits and Systems* 66.9 (Sept. 2019), pp. 1577–1581.

[28] K. Ni et al. “Write Disturb in Ferroelectric FETs and Its Implication for 1T-FeFET AND Memory Arrays”. In: *IEEE Electron Device Letters* 39.11 (Nov. 2018), pp. 1656–1659.

[29] M. Madhusudhan Reddy, K. V. Challa, and B. Srinivasa Raja. “An Energy Efficient Static Address Decoder for High-Speed Memory Applications”. In: *2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India*. 2022, pp. 50–53.

[30] Dimitrios Balobas and Nikos Konofaos. “Design of low-power high-performance 2-4 and 4-16 mixed-logic line decoders”. In: *IEEE transactions on circuits and systems-II: express briefs* 64 (2017), pp. 176–180.

# B

# Additional plots and figures

## B.1. Partial-open single cell defects

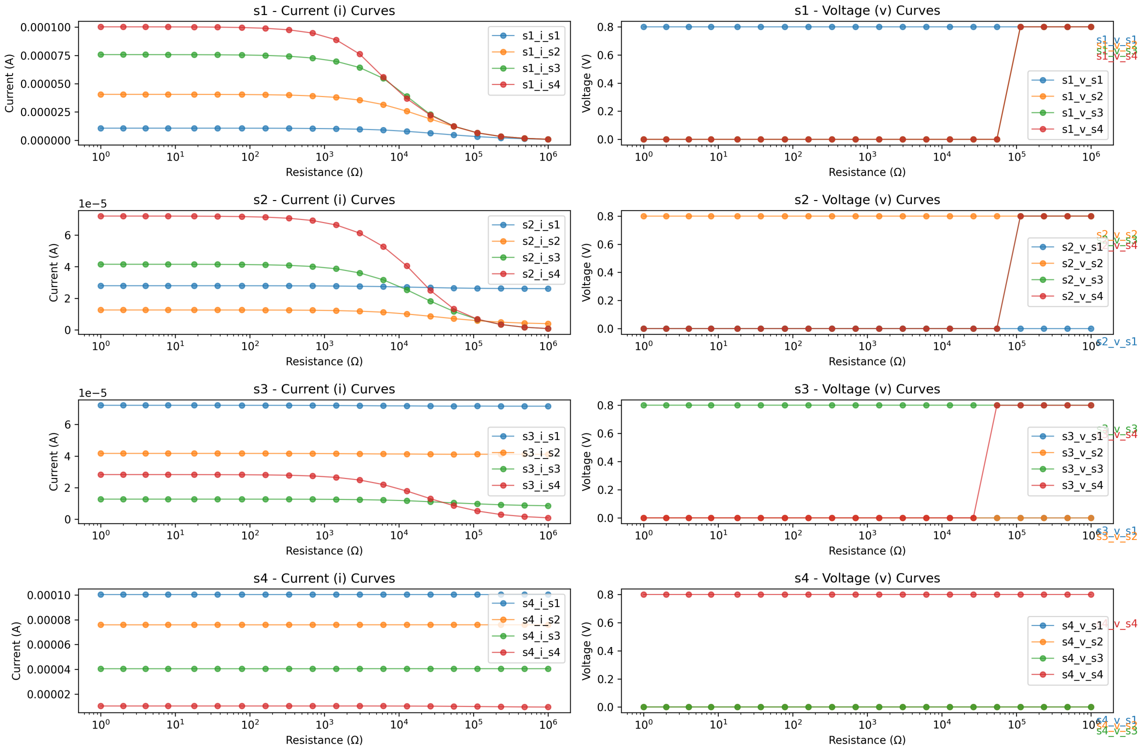


**Figure B.1:** Voltage and current plots for the single cell, partial-open defect point on $R_1$.

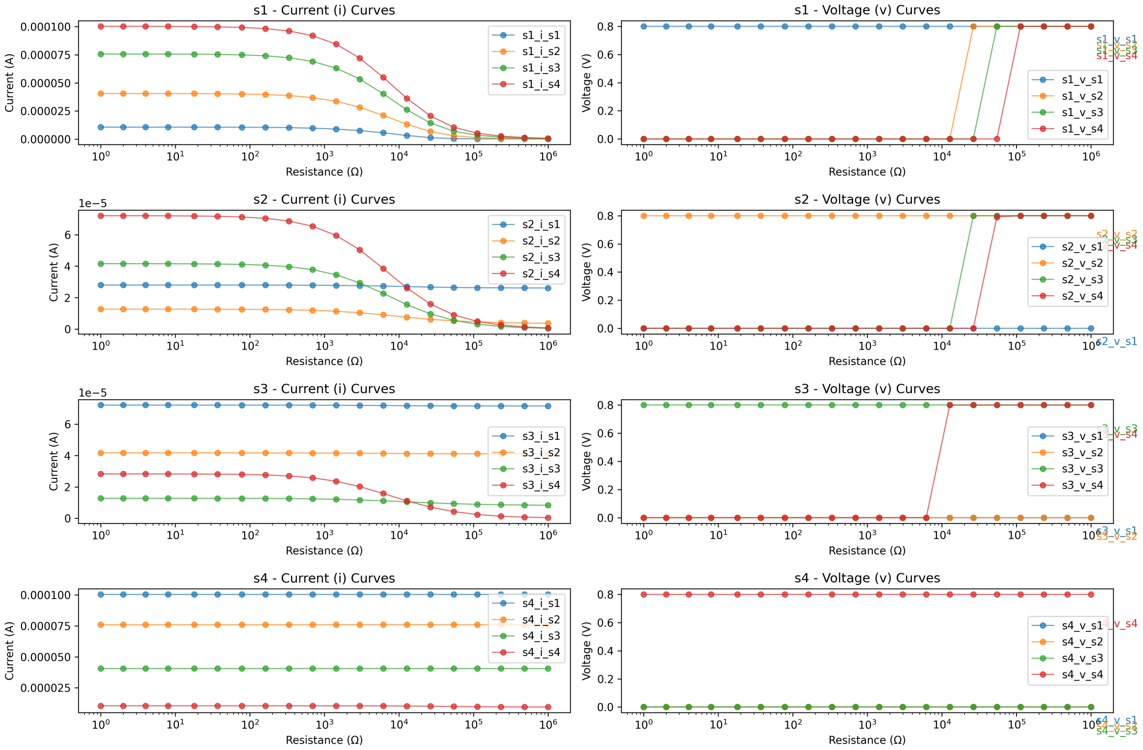


**Figure B.2:** Voltage and current plots for the single cell, partial-open defect point on $R_2$.

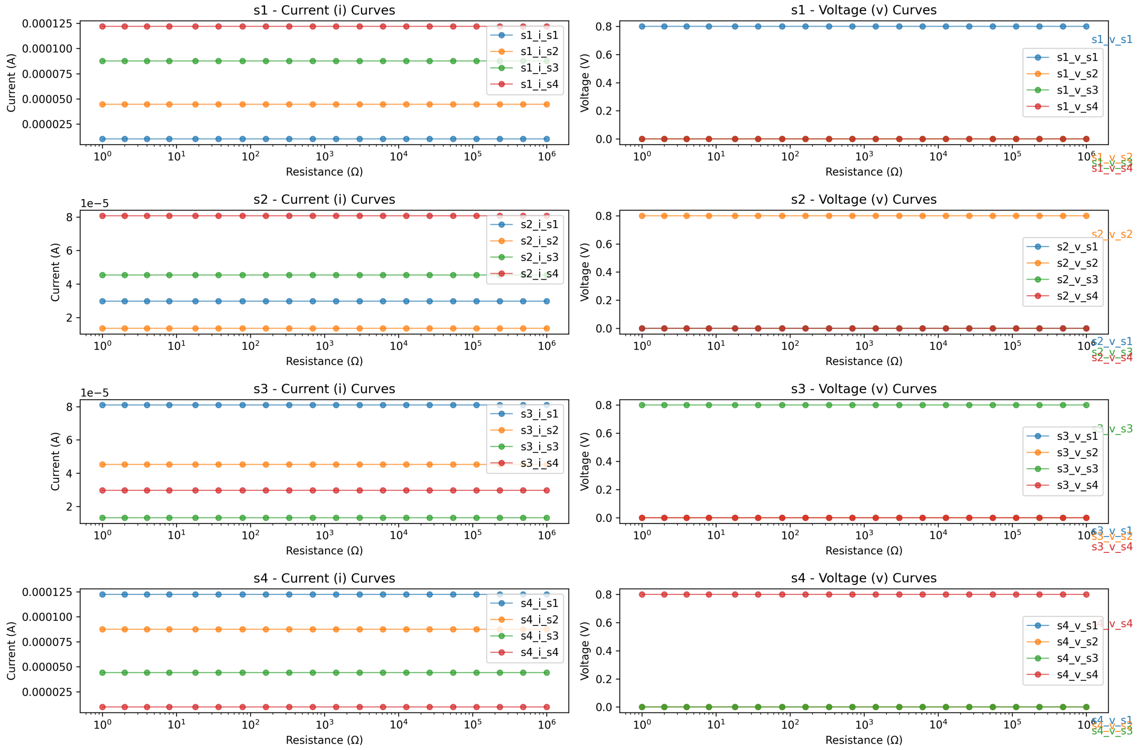


**Figure B.3:** Voltage and current plots for the single cell, partial-open defect point on $R_4$.

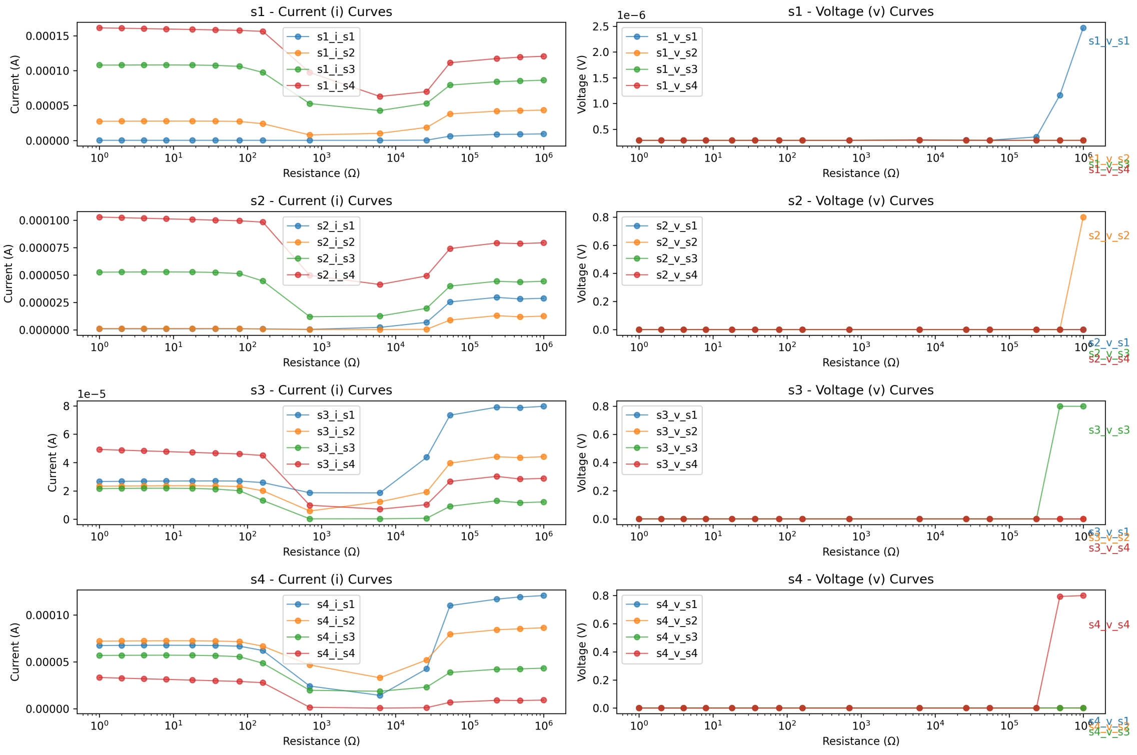


**Figure B.4:** Voltage and current plots for the single cell, bridge defect point on $R_7$.

## B.2. Partial-open coupled defects

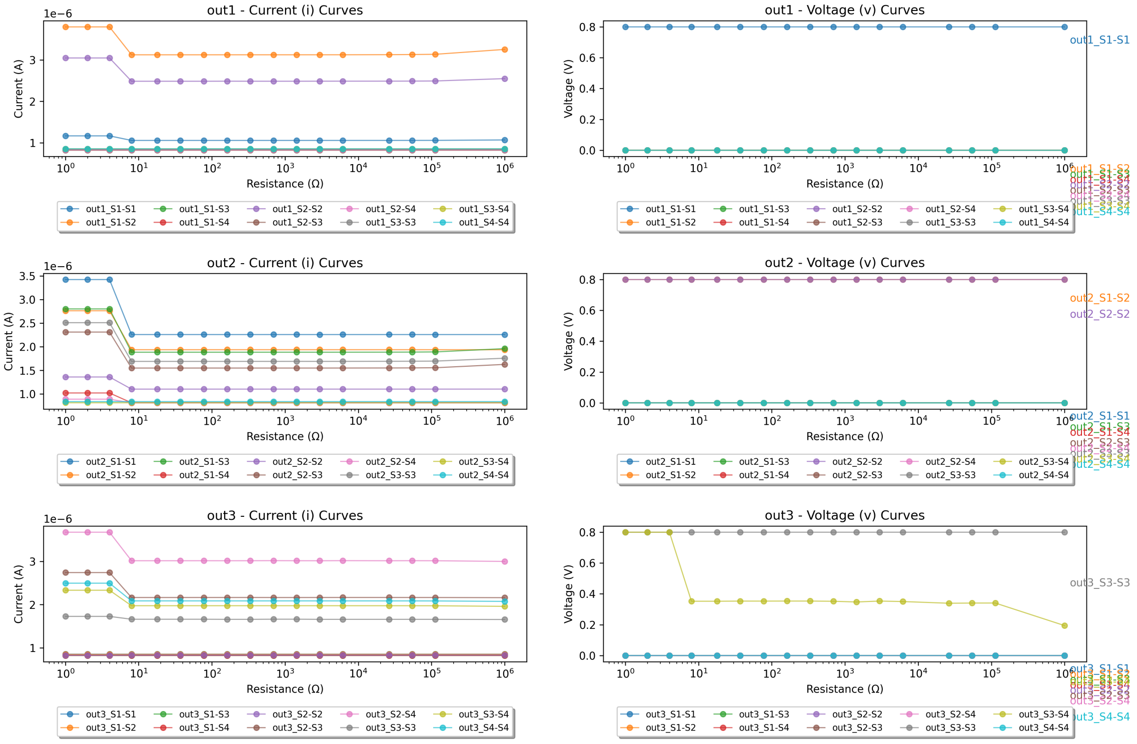


**Figure B.5:** Voltage and current plots for the partial-open array defect point on $R_2$.

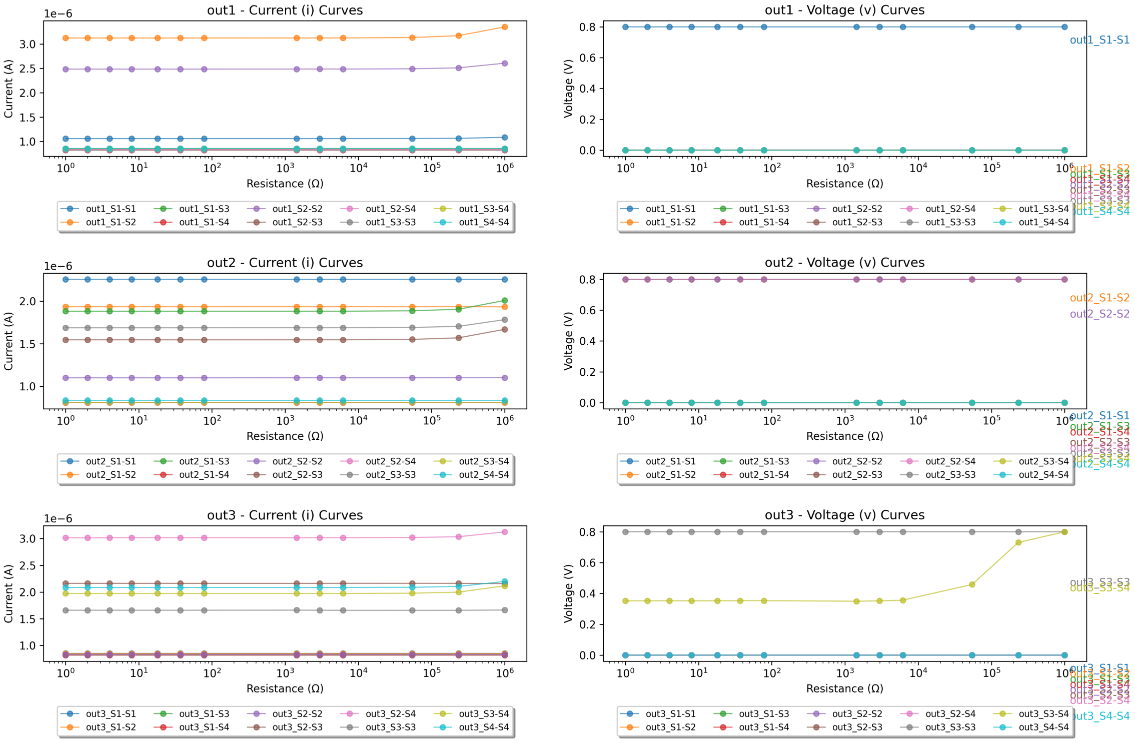


**Figure B.6:** Voltage and current plots for the partial-open array defect point on $R_3$.

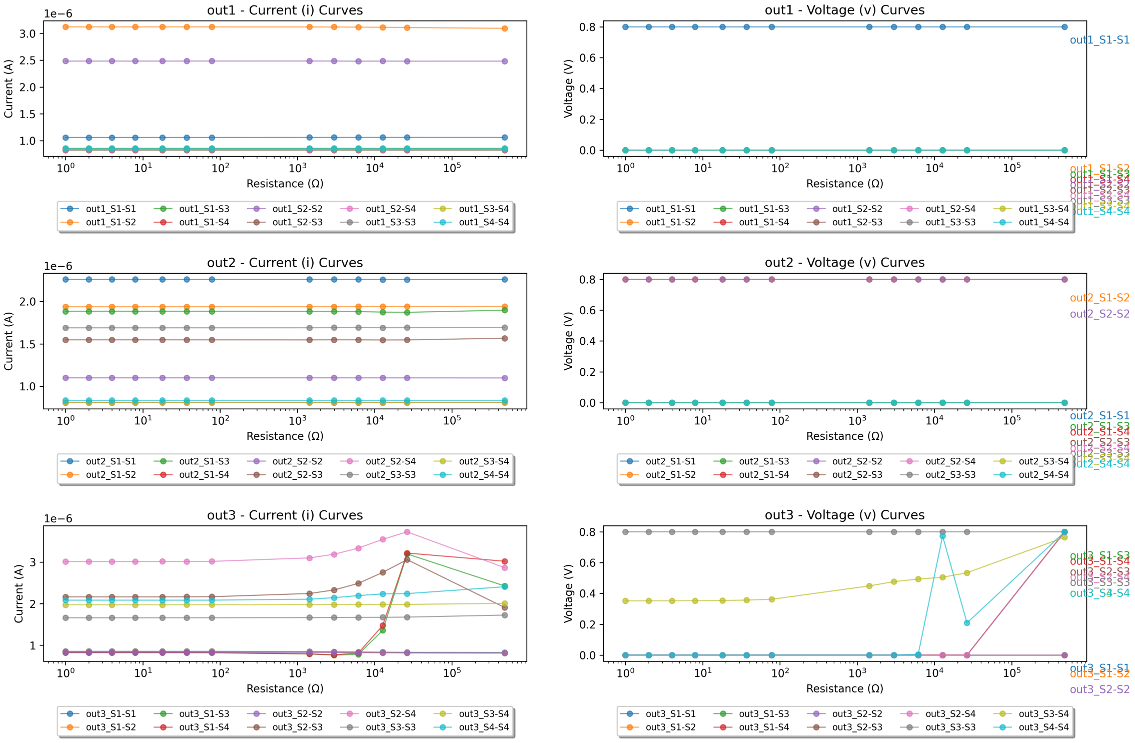


**Figure B.7:** Voltage and current plots for the partial-open array defect point on $R_5$ (match line).

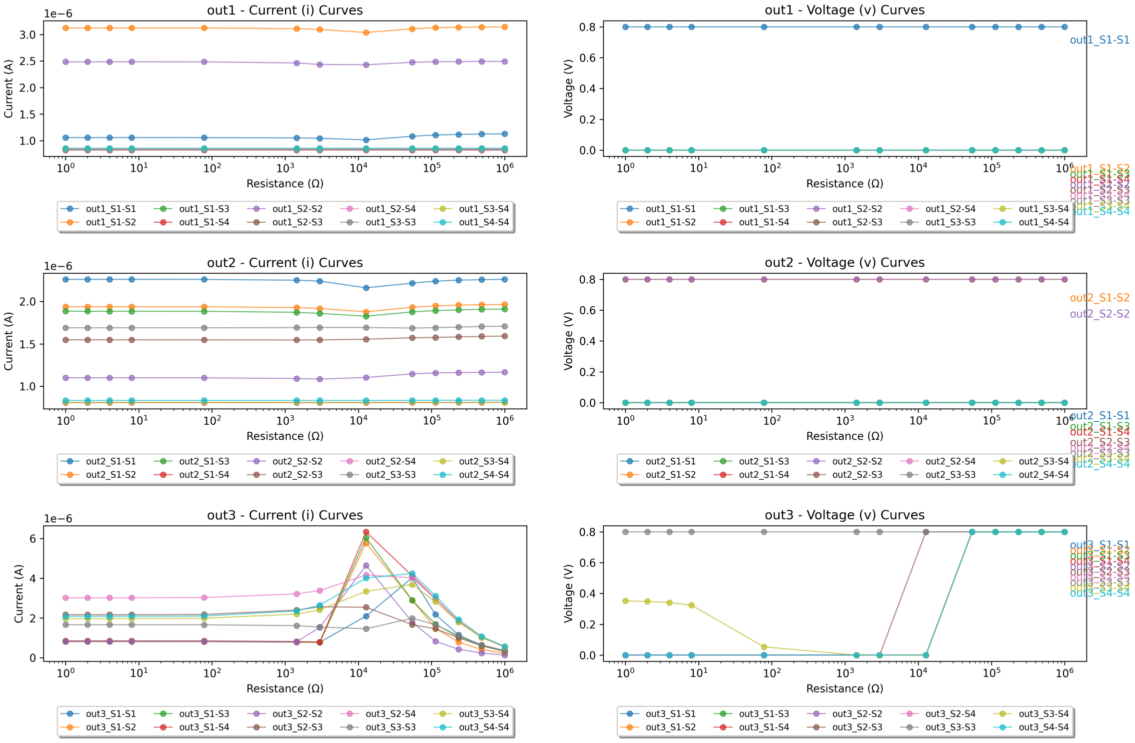


**Figure B.8:** Voltage and current plots for the partial-open array defect point on $R_7$ (source line).

## B.3. Additional March Tests

| Defect site | Tests |
|---|---|
| $R_6$ | $\{(W_{S1}, R_{S1})\}$ |
| $R_7$ | $\{(W_{S1}, R_{S1})\}$ |

**Table B.1:** An example of a March test to detect single-cell bridge defects

| Defect site | Tests |
|---|---|
| $R_4$ | $\{(W_{S3-S3}, R_{S3-S4})\}$ |
| $R_5$ | $\{(W_{S3-S3}, R_{S4-S4})\}$ |
| $R_6$ | $\{(W_{S3-S3}, R_{S3-S4})\}$ |
| $R_7$ | $\{(W_{S3-S3}, R_{S2-S3})\}$ |

**Table B.2:** An example of a March test used to detect partial-open faults in FCAM arrays.

| Defect site | Tests |
|---|---|
| $R_4$ | $\{(W_{S1-S1}, R_{S1-S4})\}$ |
| $R_5$ | $\{(W_{S1-S1}, R_{S1-S4})\}$ |
| $R_6$ | $\{(W_{S1-S1}, R_{S1-S4})\}$ |
| $R_7$ | $\{(W_{S1-S1}, R_{S1-S4})\}$ |

**Table B.3:** An example of a March test used to detect bridge faults in FCAM arrays.